

Metody numerické integrace s podrobnějším zaměřením na Rombergovu metodu

Numerical integration with focus on Romberg's method

Roman Král

Bakalářská práce

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

Ostrava, 2021

Abstrakt

Mnohdy je příliš náročné daný určitý integrál vypočítat jen pomocí nástrojů matematické analýzy, a proto si musíme pomoci vhodnými numerickými přístupy. Aproximace určitých integrálů jsou obecně založeny na rozdělení původního intervalu na podintervaly, na nichž se integrované funkce nahradí polynomem. Základní metody numerické integrace konvergují jen pomalu, pro rychlejší konvergenci můžeme použít třeba Gaussovu kvadraturu nebo Rombergovu metodu. Rombergova metoda je založena na tzv. Richardsonově extrapolaci; využívá složená integrační pravidla na postupně dvojnásobně zjemňovaných sítích uzlů ke zlepšení řádu konvergence.

Klíčová slova

numerické integrační metody, Lagrangeův polynom, Newtonovy-Cotesovy kvadrurní vzorce, analýza chyb, rekurzivní metody, Richardsonova extrapolace, Rombergova metoda, Gaussovy kvadrurní vzorce, ortogonální polynomy

Abstract

It is often too difficult to calculate a given definite integral only by means of mathematical analysis, therefore, we have to use proper numerical approaches. Approximation of definite integrals are generally based on cutting a given interval into subintervals and on each of them we approximate the function by a polynomial. The very basic methods of numerical integration converge only very slowly, to accelerate the convergence we can use, i.e, Gaussian quadrature or Romberg's method. The latter is based on the so-called Richardson extrapolation and it uses composite integration schemes on gradually refined grids to improve the rate of convergence.

Keywords

numerical integration methods, Lagrange polynomial, Newton-Cotes quadrature formulas, error analysis, recursive methods, Richardson extrapolation, Romberg algorithm, Gaussian quadrature formulas, orthogonal polynomials

Poděkování

Rád bych poděkoval své vedoucí práce Ing. Marii Sadowské, Ph.D. za její cenné připomínky, duchaplné rady a hlavně za skvělé a pravidelné konzultace při tvorbě této bakalářské práce. Taktéž bych rád poděkoval své rodině a přátelům za plnou podporu, která mne přes maturitu dovedla až k této příležitosti.

Obsah

Seznam použitých zkratk a symbolů	6
Seznam obrázků	7
Seznam tabulek	8
1 Úvod	9
2 Newtonovy-Cotesovy kvadrurní formule	11
2.1 Formule nižšího řádu	12
2.2 Odhad chyby	15
2.3 Složená pravidla	19
2.4 Formule vyšších řádů	22
2.5 Numerické experimenty pro Newtonovy-Cotesovy vzorce	23
3 Rekurzivní metody	28
3.1 Dělení subintervalů na třetiny	30
3.2 Numerické experimenty pro rekurzivní metody	33
4 Eulerova-Maclaurinova expanze	34
5 Richardsonova extrapolace	40
5.1 Numerické experimenty pro Richardsonovu extrapolaci	43
6 Rombergova metoda	47
6.1 Numerické experimenty pro Rombergovu metodu	48
7 Adaptivní Simpsonovo pravidlo	51
7.1 Numerické experimenty pro adaptivní Simpsonovo pravidlo	53
8 Gaussova kvadratura	54
8.1 Gaussova-Legendreova kvadratura	56

8.2	Gaussova-Čebyševova kvadratura	60
8.3	Numerické experimenty pro Gaussovy kvadratury	63
9	Závěr	64
	Literatura	65
	Příloha A Ukázky kódů	66

Seznam použitých zkratek a symbolů

■	konec důkazu
△	konec příkladu
\mathbb{N}	množina všech přirozených čísel
\mathbb{N}_0	množina všech nezáporných celých čísel
\mathbb{R}	množina všech reálných čísel
\mathbb{R}^n	vektorový prostor všech uspořádaných n -tic reálných čísel
$\mathbb{R}^{m \times n}$	vektorový prostor všech reálných matic $m \times n$
$C^n(\langle a, b \rangle)$	prostor všech reálných funkcí se spojitými derivacemi na $\langle a, b \rangle$ až do řádu n včetně
$\mathcal{P}_n(\langle a, b \rangle)$	prostor všech reálných polynomů na $\langle a, b \rangle$ až do řádu n včetně
$\deg(q)$	stupeň mnohočlenu q
$\ f\ _\infty$	$\max_{\xi \in \langle a, b \rangle} f(\xi) $
$\mathcal{O}(f)$	Landauova notace, popisuje limitní chování funkce

Seznam obrázků

1.1	Průběh funkce $e^{\cos(\sin(\arctg(x^2)))}$	9
2.1	Vizuální porovnání Newtonových-Cotesových formulí	14
2.2	Rozdělení intervalu $\langle a, b \rangle$ pro složené lichoběžníkové pravidlo	20
2.3	Rozdělení intervalu $\langle a, b \rangle$ pro složené Simpsonovo pravidlo	20
2.4	Konvergence složených pravidel I	24
2.5	Chování pravidel vyšších řádů I	24
2.6	Průběh funkce $x^3 \cos(4\pi x)$	25
2.7	Konvergence složených pravidel II	26
2.8	Chování pravidel vyšších řádů II	27
3.1	Půlení intervalů pro rekurzivní lichoběžníkové pravidlo	28
3.2	Půlení intervalů pro rekurzivní Simpsonovo pravidlo	30
3.3	Rozdělení intervalů na třetiny pro rekurzivní lichoběžníkové pravidlo	31
3.4	Rozdělení intervalů na třetiny pro rekurzivní Simpsonovo pravidlo	32
5.1	Konvergence metody centrálních diferencí	44
5.2	Konvergence Richardsonovy extrapolace I	45
5.3	Konvergence Richardsonovy extrapolace II	46
6.1	Konvergence Rombergovy metody I	49
6.2	Konvergence Rombergovy metody II	50
8.1	Legendreovy polynomy nultého až osmého řádu	57
8.2	Čebyševovy polynomy nultého až osmého řádu	62

Seznam tabulek

2.1	Odhady chyb základních pravidel	19
2.2	Odhady chyb složených pravidel	21
2.3	Newtonovy-Cotesovy formule vyšších řádů	22
2.4	Aproximace pomocí složených Newtonových-Cotesových formulí I	23
2.5	Aproximace pomocí složených Newtonových-Cotesových formulí II	26
3.1	Aproximace integrálu pomocí rekurzivních metod	33
7.1	Aproximace pomocí adaptivního Simpsonova pravidla	53
8.1	Kvadrurní body a váhy pro Newtonovy-Cotesovy metody řádu n	55
8.2	Integrační body a váhy Gaussovy-Legendreovy kvadratury na intervalu $\langle -1, 1 \rangle$	59
8.3	Integrační body a váhy Gaussovy-Čebyševovy kvadratury na intervalu $\langle -1, 1 \rangle$	61
8.4	Aproximace pomocí Gaussovy-Legendreovy kvadratury	63
8.5	Aproximace pomocí Gaussovy-Čebyševovy kvadratury	63

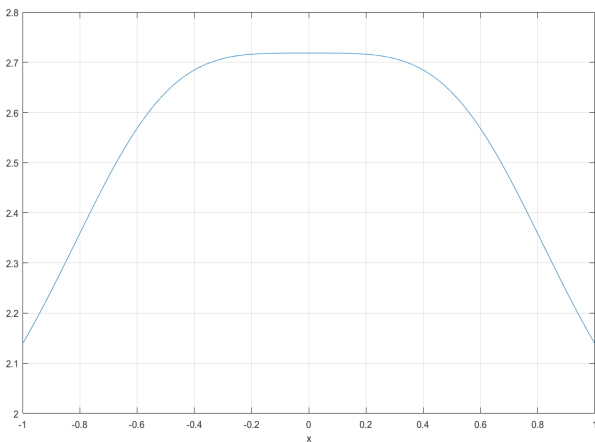
Kapitola 1

Úvod

V mnoha aplikacích matematické analýzy se často setkáváme s problémem, že nejsme schopni daný určitý integrál na intervalu $\langle a, b \rangle$ analyticky spočítat. Problémy nastávají zejména pro složené funkce, které mohou mít i velmi hladký průběh, ale nejsme schopni pomocí známých integračních metod (per partes, první a druhá substituční metoda, ...) určit jejich hodnotu, přestože existuje. Vezměme si ku příkladu

$$\int_{-1}^1 e^{\cos(\sin(\arctg(x^2)))} dx. \quad (1.0.1)$$

Průběh této integrované funkce na intervalu $\langle -1, 1 \rangle$ můžeme vidět na následujícím obrázku:



Obrázek 1.1: Průběh funkce $e^{\cos(\sin(\arctg(x^2)))}$

Vidíme, že se jedná o spojitou funkci (jedná se o složení spojitých funkcí), takže integrál (1.0.1) musí existovat, ovšem analytický výpočet tohoto integrálu je takřka nemožný. V praxi si ovšem vystačíme s dobrými aproximacemi, které nám poskytuje numerická integrace neboli kvadratura. V této práci si předvedeme jedny z nejznámějších metod: obdélníkové, lichoběžníkové a Simpsonovo pravidlo (ty řadíme mezi tzv. Newtonovy-Cotesovy kvadraturní formule). Zjistíme, že můžeme velmi

efektivně zrychlit výpočet díky rekurzivním formám dříve zmíněných pravidel. Ovšem stále se jedná o aproximaci a vznikají zde chyby, které budeme analyzovat. Ukážeme si, že chybu lichoběžníkového pravidla lze vyjádřit Eulerovou-Maclaurinovou formulí, což nás povede k Richardsonově extrapolaci, která obecně vede ke zpřesnění výsledků numerických metod. Ukážeme si, jak můžeme zpřesnit nejen kvadratury, ale třeba i numerické derivování. Jestliže zvolíme pro Richardsonovu extrapolaci lichoběžníkové pravidlo, tak mluvíme o Rombergově metodě. Úkolem této metody je vytvoření dolní trojúhelníkové matice, která má v prvním sloupci aproximace z rekurzivního lichoběžníkového pravidla, a použitím Richardsonovy extrapolace jsou vytvářeny další sloupce, které v sobě nesou přesnější výsledky. V posledním sloupci se pak nachází nejlepší aproximace.

Jednou z podmínek Eulerovy-Maclaurinovy formule je jistá třída hladkosti funkce, tudíž při nesplnění této podmínky není Rombergova metoda spolehlivá. Pro úplnost výčtu základních numerických metod uvedeme Gaussovu kvadraturu, která má své integrační uzly uvnitř intervalu, a adaptivní Simpsonovo pravidlo, které také patří k nejčastěji využívaným kvadraturám.

Kapitola 2

Newtonovy-Cotesovy kvadrurní formule

První myšlenkou, jak bychom mohli numericky spočítat složitější integrály, je nahradit původní funkci na daném intervalu jinou funkcí, která by ji co nejlépe aproximovala, a zároveň, aby její integrace byla jednodušší. Takovéto vlastnosti má například **Lagrangeův interpolační polynom** p_n stupně n , který je důkladně popsán v [2, 5, 7, 8].

Uvažujme $f : \mathbb{R} \rightarrow \mathbb{R}$ spojitou na daném $\langle a, b \rangle$. Označme $x_i \in \langle a, b \rangle$, $i = 0, 1, 2, \dots, n$, jako interpolační uzly, kde $a = x_0 < x_1 < x_2 < \dots < x_n = b$. Lagrangeův interpolační polynom je definován pomocí¹

$$p_n(x) = \begin{cases} f\left(\frac{b+a}{2}\right), & n = 0, \\ \sum_{k=0}^n \ell_k(x)f(x_k), & n \geq 1, \end{cases}$$

kde pro $n \geq 1$ jsou bázové funkce $\ell_0, \ell_1, \ell_2, \dots, \ell_n$ určené vztahem

$$\ell_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{x - x_i}{x_k - x_i}, \quad k = 0, 1, 2, \dots, n. \quad (2.0.1)$$

Jelikož se jedná o polynom, jehož integrace je velmi jednoduchá, tak je dobrým „kandidátem“ pro nahrazení a píšeme

$$\int_a^b f(x) dx \approx \int_a^b p_n(x) dx.$$

Pro jednoduchost budeme dále uvažovat pro $n \geq 1$ ekvidistantní síť uzlů

$$x_i = a + ih, \quad h = \frac{b-a}{n}, \quad i = 0, 1, 2, \dots, n. \quad (2.0.2)$$

¹Obecně můžeme pro $n = 0$ zvolit bod kdekoli v intervalu $\langle a, b \rangle$, ale my uvádíme bod $\frac{b+a}{2}$ z důvodu pozdějšího navázání na tzv. obdélníkové pravidlo, které se s touto volbou bodu nazývá „midpoint rule“ v odborné literatuře.

Jestliže položíme

$$w_k = \begin{cases} \int_a^b 1 \, dx = b - a, & n = 0, \\ \int_a^b \ell_k(x) \, dx, & n \geq 1, \end{cases} \quad k = 0, 1, 2, \dots, n,$$

můžeme psát¹

$$\int_a^b f(x) \, dx \approx \sum_{k=0}^n w_k f(x_k). \quad (2.0.3)$$

Číslo w_k nazveme **integrační váhou**, interpolační uzly $x_0, x_1, x_2, \dots, x_n$ **integračními body** a (2.0.3) **Newtonovou-Cotesovou kvadraturní formulí řádu n** .

Nyní si vysvětlíme teorii formulí do $n = 2$. Určíme jejich numerické kvadraturní vzorce a vzniklou chybu, kterou také popisují [7, 8, 9].

2.1 Formule nižšího řádu

V této kapitole se zabýváme formulí nižších řádů (viz také [8, 4, 7]). Jestliže máme zavedené integrační body, které obsahují krajní prvky $\langle a, b \rangle$ (viz (2.0.2)), mluvíme o **uzavřené formulí**. Pokud máme ekvidistantní síť uzlů takovou, že všechny integrační body jsou uvnitř intervalu, tak se jedná o **otevřenou formulí**, jak zmiňuje třeba [6]. Široký seznam uzavřených i otevřených formulí uvádí [10], my se zaměříme na základní z nich.

Obdélníkové pravidlo

Jestliže $n = 0$, mluvíme o **obdélníkovém pravidle**, které řadíme do otevřených formulí. Tato metoda obsahuje pouze jeden integrační bod. Víme, že

$$p_0(x) = f\left(\frac{b+a}{2}\right).$$

Zintegrujeme-li p_0 na $\langle a, b \rangle$, dostáváme aproximaci

$$\int_a^b f(x) \, dx \approx \int_a^b p_0(x) \, dx = (b-a)f\left(\frac{b+a}{2}\right) \stackrel{\text{ozn.}}{=} O(f, a, b). \quad (2.1.1)$$

Jméno tohoto pravidla je určeno z jeho geometrického vyobrazení (počítáme vlastně obsah obdélníku).

¹V případě $n = 0$ uvažujeme $x_0 = \frac{b+a}{2}$.

Lichoběžníkové pravidlo

Nyní se podívejme na situaci, kdy $n = 1$. Zvolme integrační body $x_0 = a$, $x_1 = b$ a určíme Lagrangeův polynom prvního řádu (jedná se o lineární polynom)

$$p_1(x) = \ell_0(x)f(a) + \ell_1(x)f(b). \quad (2.1.2)$$

Následně vypočteme bázové funkce

$$\ell_0(x) = \frac{x-b}{a-b}, \quad \ell_1(x) = \frac{x-a}{b-a},$$

ty dosadíme do (2.1.2) a dostáváme

$$p_1(x) = \frac{x-b}{a-b}f(a) + \frac{x-a}{b-a}f(b).$$

Nyní zintegrujme p_1 na $\langle a, b \rangle$ a dostáváme předpis pro **lichoběžníkové pravidlo** (počítáme vlastně obsah lichoběžníku), které řadíme mezi uzavřené formule:

$$\int_a^b f(x) dx \approx \int_a^b p_1(x) dx = \frac{b-a}{2} [f(a) + f(b)] \stackrel{\text{ozn.}}{=} L(f, a, b). \quad (2.1.3)$$

Simpsonovo pravidlo

Zvolíme-li $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$ a $n = 2$, tj. $h = (b-a)/2$, tak funkci f aproximujeme kvadratickým polynomem

$$p_2(x) = \ell_0(x)f(x_0) + \ell_1(x)f(x_1) + \ell_2(x)f(x_2).$$

Abychom jej určili, musíme nejdříve získat bázové funkce

$$\ell_0(x) = \frac{x-x_1}{x_0-x_1} \cdot \frac{x-x_2}{x_0-x_2} = \frac{x-x_1}{-h} \cdot \frac{x-x_2}{-2h} = \frac{1}{2h^2}(x-x_1)(x-x_2),$$

$$\ell_1(x) = \frac{x-x_0}{x_1-x_0} \cdot \frac{x-x_2}{x_1-x_2} = \frac{x-x_0}{h} \cdot \frac{x-x_2}{-h} = -\frac{1}{h^2}(x-x_0)(x-x_2),$$

$$\ell_2(x) = \frac{x-x_0}{x_2-x_0} \cdot \frac{x-x_1}{x_2-x_1} = \frac{x-x_0}{2h} \cdot \frac{x-x_1}{h} = \frac{1}{2h^2}(x-x_0)(x-x_1).$$

Nyní stačí zintegrovat p_2 na $\langle a, b \rangle$:

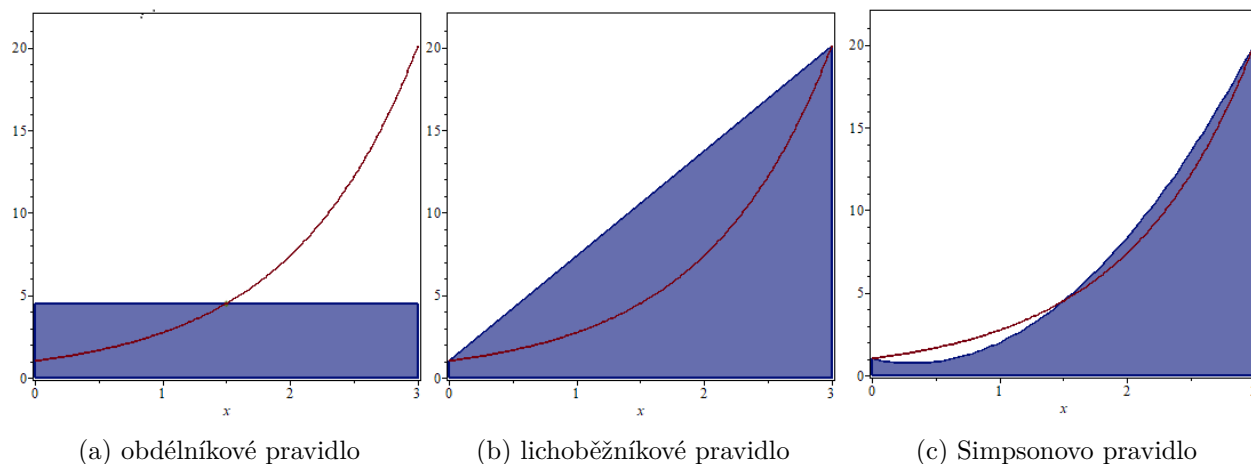
$$\begin{aligned} \int_a^b p_2(x) \, dx &= \int_a^b [\ell_0(x)f(x_0) + \ell_1(x)f(x_1) + \ell_2(x)f(x_2)] \, dx = \\ &= \int_{x_0}^{x_2} \ell_0(x)f(x_0) \, dx + \int_{x_0}^{x_2} \ell_1(x)f(x_1) \, dx + \int_{x_0}^{x_2} \ell_2(x)f(x_2) \, dx = \\ &= \frac{h}{3}f(x_0) + \frac{4h}{3}f(x_1) + \frac{h}{3}f(x_2) = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] \end{aligned}$$

a získáváme předpis pro **Simpsonovo pravidlo**, které taktéž řadíme mezi uzavřené formule.

$$\int_a^b f(x) \, dx \approx \int_a^b p_2(x) \, dx = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \stackrel{\text{ozn.}}{=} S(f, a, b). \quad (2.1.4)$$

Na obr. 2.1 vidíme porovnání těchto metod, které jsou aplikovány na integrál

$$\int_0^3 e^x \, dx.$$



Obrázek 2.1: Vizuální porovnání Newtonových-Cotesových formulí

2.2 Odhad chyby

Zamyslíme-li se teď nad vztahem (2.0.3), mohla by nás napadnout otázka, kdy nastane rovnost. Tak jako u všech aproximací, zde vzniká chyba, kterou můžeme ovlivnit. Označme tuto chybu Newtonovy-Cotesovy formule jako

$$E_n(f, a, b) \stackrel{\text{ozn.}}{=} \int_a^b f(x) dx - \sum_{k=0}^n w_k f(x_k).$$

Věta 2.2.1 (Odhad chyby obdélníkového pravidla). [8, 9] *Nechť $f \in C^2(\langle a, b \rangle)$. Pak*

$$|E_0(f, a, b)| \leq \frac{(b-a)^3}{24} \|f''\|_\infty.$$

Důkaz. (Analogicky jako v [8] pro Simpsonovo pravidlo.) *Zajímá nás vlastně odhad chyby*

$$\int_a^b f(x) dx - (b-a)f\left(\frac{b+a}{2}\right). \quad (2.2.1)$$

Položme

$$x = x(t) = \frac{a+b}{2} + \frac{b-a}{2}t, \quad t \in \langle -1, 1 \rangle \quad (2.2.2)$$

a definujme funkci $F(t) = f(x(t))$. Dosazením do (2.2.1) dostáváme

$$\int_a^b f(x(t)) dx - (b-a)f\left(\frac{b+a}{2}\right) = \frac{b-a}{2} \left(\int_{-1}^1 F(\tau) d\tau - 2F(0) \right).$$

Definujme funkci G předpisem

$$G(t) = \int_{-t}^t F(\tau) d\tau - 2tF(0), \quad t \in \langle -1, 1 \rangle.$$

Všimneme si, že

$$\frac{b-a}{2}G(1) = \frac{b-a}{2} \left(\int_{-1}^1 F(\tau) d\tau - 2F(0) \right).$$

Sestavme nyní funkci H pomocí

$$H(t) = G(t) - t^3G(1), \quad t \in \langle a, b \rangle.$$

Je zřejmé, že $H(1) = H(0) = 0$. Aplikujeme na funkci H Rolleovu větu. Tudiž $\exists \xi_1 \in (0, 1) : H'(\xi_1) = 0$. Zderivujme nyní funkci H :

$$H'(t) = G'(t) - 3t^2G(1) = F(t) + F(-t) - 2F(0) - 3t^2G(1).$$

Vidíme, že $H'(0) = 0$, to ale znamená, že $H'(\xi_1) = H'(0) = 0$, takže použijeme znovu Rolleovu větu,

a tedy $\exists \xi_2 \in (0, \xi_1) : H''(\xi_2) = 0$. Zderivujeme nyní H' a dosadíme $t = \xi_2$:

$$6\xi_2 G(1) = F'(\xi_2) - F'(-\xi_2).$$

Na pravou stranu poslední rovnosti použijeme Lagrangeovu větu o střední hodnotě

$$\exists \xi_3 \in (-\xi_2, \xi_2) : 2\xi_2 F''(\xi_3) = F'(\xi_2) - F'(-\xi_2),$$

a proto

$$G(1) = \frac{1}{3} F''(\xi_3). \quad (2.2.3)$$

S pomocí (2.2.2) určíme hodnotu druhé derivace:

$$F''(\xi_3) = \frac{(b-a)^2}{4} f''(x(\xi_3)), \quad (a, b) \ni x(\xi_3) \stackrel{\text{ozn.}}{=} \xi. \quad (2.2.4)$$

Dosadíme (2.2.4) do (2.2.3), vynásobíme $\frac{b-a}{2}$ a dostáváme námi hledanou rovnost

$$\int_a^b f(x(t)) \, dx - O(f, a, b) = \frac{(b-a)^3}{24} f''(\xi), \quad \xi \in (a, b). \quad (2.2.5)$$

■

Věta 2.2.2 (Odhad chyby lichoběžníkového pravidla). [7, 8, 9] Necht $f \in C^2(\langle a, b \rangle)$. Pak platí

$$|E_1(f, a, b)| \leq \frac{(b-a)^3}{12} \|f''\|_\infty.$$

Důkaz. [7, 8] Připomeňme, že chyba polynomiální interpolace s $(n+1)$ uzly v intervalu $\langle a, b \rangle$ je (viz také [5])

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \pi_{n+1}(x), \quad \xi \in (a, b),$$

kde

$$\pi_{n+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_n).$$

Tudíž

$$E_1(f, a, b) = \frac{f''(\xi)}{2} \int_a^b (x-a)(x-b) \, dx = -\frac{(b-a)^3}{12} f''(\xi). \quad (2.2.6)$$

■

Analogicky bychom pro chybu Simpsonova pravidla dostali

$$\begin{aligned} |E_2(f, a, b)| &\leq \|f'''\|_\infty \frac{1}{6} \int_a^b \left| (x-a) \left(x - \frac{a+b}{2} \right) (x-b) \right| \, dx = \\ &= \frac{(b-a)^4}{192} \|f'''\|_\infty. \end{aligned} \quad (2.2.7)$$

Nyní si zavedeme lepší určení odhadu chyby Simpsonova pravidla (viz následující větu). Jak se píše v [8], je pěkně vidět, že Simpsonovo pravidlo integruje přesně polynomy do 3. stupně včetně, což neukazuje odhad (2.2.7).

Věta 2.2.3 (Odhad chyby Simpsonova pravidla). *Nechť $f \in C^4(\langle a, b \rangle)$. Pak*

$$|E_2(f, a, b)| \leq \frac{(b-a)^5}{2880} \|f^{(4)}\|_\infty.$$

Důkaz. [8] *Zkoumejme chybu*

$$\int_a^b f(x) \, dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]. \quad (2.2.8)$$

Položme

$$x = x(t) = \frac{a+b}{2} + \frac{b-a}{2}t, \quad t \in \langle -1, 1 \rangle,$$

a definujme funkci F pomocí $F(t) = f(x(t))$. Dosazením do (2.2.8) dostáváme

$$\begin{aligned} \int_a^b f(x(t)) \, dx - \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] = \\ = \frac{b-a}{2} \left(\int_{-1}^1 F(\tau) \, d\tau - \frac{1}{3} [F(-1) + 4F(0) + F(1)] \right). \end{aligned}$$

Definujme funkci G předpisem

$$G(t) = \int_{-t}^t F(\tau) \, d\tau - \frac{t}{3} [F(-t) + 4F(0) + F(t)], \quad t \in \langle -1, 1 \rangle.$$

Můžeme vidět, že

$$\frac{b-a}{2} G(1) = \frac{b-a}{2} \left(\int_{-1}^1 F(\tau) \, d\tau - \frac{1}{3} [F(-1) + 4F(0) + F(1)] \right).$$

Sestavme dále funkci H , kde

$$H(t) = G(t) - t^5 G(1), \quad t \in \langle -1, 1 \rangle.$$

Podívejme se na funkční hodnoty $H(0)$ a $H(1)$. Jelikož je zřejmé, že $H(0) = H(1) = 0$ a z předpokladu víme, že $f \in C^4(\langle a, b \rangle)$, tak aplikujeme na H Rolleovu větu. Tudiž $\exists \xi_1 \in (0, 1) : H'(\xi_1) = 0$. Zderivujme nyní funkci H :

$$H'(t) = G'(t) - 5t^4 G(1).$$

Rozepíšeme si a upravíme G' :

$$G'(t) = F(t) + F(-t) - \frac{1}{3} [F(-t) + 4F(0) + F(t)] - \frac{t}{3} [-F'(-t) + F'(t)].$$

Teď, když již máme určenou G' , tak jistě vidíme, že $H'(\xi_1) = H'(0) = 0$, a tudíž opět aplikujeme Rolleovu větu, neboli $\exists \xi_2 \in (0, \xi_1) : H''(\xi_2) = 0$. Zderivujme nyní funkci H' :

$$H''(t) = G''(t) - 20t^3G(1).$$

Rozepíšeme si a upravíme G'' :

$$\begin{aligned} G''(t) &= F'(t) - F'(-t) - \frac{1}{3}[F'(t) - F'(-t)] - \frac{1}{3}[F'(t) - F'(-t)] - \frac{t}{3}[F''(t) + F''(-t)] \\ &= \frac{1}{3}[F'(t) - F'(-t)] - \frac{t}{3}[F''(t) + F''(-t)]. \end{aligned}$$

Teď, když již máme určenou $G''(t)$, tak jistě vidíme, že $H''(\xi_1) = H''(0) = 0$, a tudíž ještě jednou aplikujeme Rolleovu větu, neboli $\exists \xi_3 \in (0, \xi_2) : H'''(\xi_3) = 0$. Zderivujme nyní funkci H'' :

$$H'''(t) = G'''(t) - 60t^2G(1). \quad (2.2.9)$$

Rozepíšeme si a upravíme G''' :

$$\begin{aligned} G'''(t) &= \frac{1}{3}[F''(t) + F''(-t)] - \frac{1}{3}[F''(t) + F''(-t)] - \frac{t}{3}[F'''(t) - F'''(-t)] = \\ &= -\frac{t}{3}[F'''(t) - F'''(-t)]. \end{aligned}$$

Upravený vztah dosadíme do (2.2.9):

$$H'''(t) = -\frac{t}{3}[F'''(t) - F'''(-t)] - 60t^2G(1).$$

Jelikož víme, že $H'''(\xi_3) = 0$, dostaneme

$$60\xi_3^2G(1) = -\frac{\xi_3}{3}[F'''(\xi_3) - F'''(-\xi_3)].$$

Na hranatou závorku na pravé straně poslední rovnosti použijeme Lagrangeovu větu o střední hodnotě

$$\exists \xi_4 \in (-\xi_3, \xi_3) : 2\xi_3 F^{(4)}(\xi_4) = F'''(\xi_3) - F'''(-\xi_3)$$

a získáváme

$$\begin{aligned} 60\xi_3^2G(1) &= -\frac{2\xi_3^2}{3}F^{(4)}(\xi_4), \\ 90G(1) &= -F^{(4)}(\xi_4), \\ G(1) &= -\frac{1}{90}F^{(4)}(\xi_4). \end{aligned}$$

Zbytek důkazu leží v určení $F^{(4)}(\xi_4)$:

$$F^{(4)}(\xi_4) = \frac{(b-a)^4}{16} f^{(4)}(x(\xi_4)), \quad (a, b) \ni \xi_4 \stackrel{\text{ozn.}}{=} \xi,$$

odkud

$$\frac{b-a}{2} G(1) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi). \quad (2.2.10)$$

■

Uvedme ještě přehled odhadů chyb pro uvedené metody:

pravidlo	odhad chyby
obdélníkové (2.1.1)	$ E_0(f, a, b) \leq \frac{(b-a)^3}{24} \ f''\ _\infty$
lichoběžníkové (2.1.3)	$ E_1(f, a, b) \leq \frac{(b-a)^3}{12} \ f''\ _\infty$
Simpsonovo (2.1.4)	$ E_2(f, a, b) \leq \frac{(b-a)^5}{2880} \ f^{(4)}\ _\infty$

Tabulka 2.1: Odhady chyb základních pravidel

2.3 Složená pravidla

Pro většinu funkcí námi dříve zmiňované metody nedokáží uspokojivě aproximovat jejich integrál na daném intervalu. Rozložíme-li ovšem interval $\langle a, b \rangle$ následovně:

$$\langle a, b \rangle = \bigcup_{i=1}^m \langle x_{i-1}, x_i \rangle,$$

kde $m \geq 2$ je počet subintervalů stejné délky, a na každý subinterval aplikujeme dané pravidlo, získáváme **složené formule**, o kterých pojednáváme v této kapitole (viz také [7, 8]).

Nyní se podíváme na složenou formu obdélníkového a lichoběžníkového pravidla. Předpokládejme, že $f \in C(\langle a, b \rangle)$. Poté

$$\int_a^b f(x) \, dx = \sum_{i=1}^m \int_{x_{i-1}}^{x_i} f(x) \, dx.$$

Délku každého subintervalu $\langle x_{i-1}, x_i \rangle$ určíme jako

$$h = \frac{b-a}{m}.$$

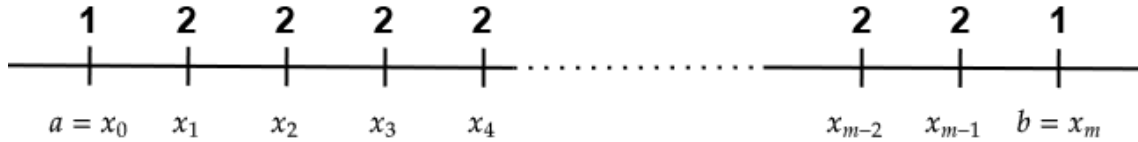
Aplikováním obdélníkového pravidla na každý subinterval dostáváme **složené obdélníkové pra-**

vidlo

$$\int_a^b f(x) dx \approx h \sum_{i=1}^m f\left(\frac{x_{i-1} + x_i}{2}\right) \stackrel{\text{ozn.}}{=} O(f, h). \quad (2.3.1)$$

Zaměříme se na **složené lichoběžníkové pravidlo**. Jelikož body s indexy $i = 1, 2, \dots, m - 1$ počítáme dvakrát, jak je vyobrazeno na obr. 2.2, dostáváme:

$$\int_a^b f(x) dx \approx h \left[\frac{f(a)}{2} + \sum_{i=1}^{m-1} f(x_i) + \frac{f(b)}{2} \right] \stackrel{\text{ozn.}}{=} L(f, h). \quad (2.3.2)$$



Obrázek 2.2: Rozdělení intervalu $\langle a, b \rangle$ pro složené lichoběžníkové pravidlo

Předpokládejme nyní, že jsme interval $\langle a, b \rangle$ rozdělili na $2m$ subintervalů tak, aby

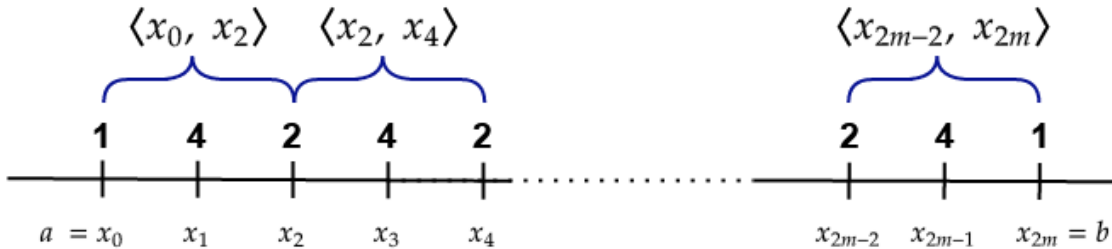
$$x_i = a + ih, \quad h = \frac{b-a}{2m}, \quad i = 0, 1, 2, \dots, 2m, \quad m \geq 1.$$

Poté po aplikaci Simpsonova pravidla na každý subinterval $\langle x_{2i-2}, x_{2i} \rangle$, $i = 1, 2, \dots, m$, dostáváme

$$\int_a^b f(x) dx = \sum_{i=1}^m \int_{x_{2i-2}}^{x_{2i}} f(x) dx \approx \frac{h}{3} \sum_{i=1}^m [f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})].$$

Jak ukazuje obr. 2.3, vnitřní body intervalu $\langle a, b \rangle$ se sudými indexy počítáme dvakrát a body s lichými indexy čtyřikrát. Těto skutečnosti využijeme a dostáváme předpis pro **složené Simpsonovo pravidlo**:

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[f(a) + f(b) + 2 \sum_{i=1}^{m-1} f(a + 2ih) + 4 \sum_{i=1}^m f(a + (2i-1)h) \right] \stackrel{\text{ozn.}}{=} S(f, h). \quad (2.3.3)$$



Obrázek 2.3: Rozdělení intervalu $\langle a, b \rangle$ pro složené Simpsonovo pravidlo

Podívejme se nyní na chyby, kterých jsme se při skládání pravidel dopustili. Vezměme si ku příkladu **chybu složeného obdélníkového pravidla**, které si analogicky odvodíme z důkazu chyby složeného lichoběžníkového pravidla uvedeném například v [7] s pomocí (2.2.5).

$$\begin{aligned} E_0(f, h) &\stackrel{\text{ozn.}}{=} \int_a^b f(x) \, dx - O(f, h) = \sum_{i=1}^m \int_{x_{i-1}}^{x_i} f(x) \, dx - O(f, x_{i-1}, x_i) = \\ &= \sum_{i=1}^m \frac{(x_i - x_{i-1})^3}{24} f''(\xi_i) = \frac{h^3}{24} \sum_{i=1}^m f''(\xi_i), \quad \xi_i \in (x_{i-1}, x_i). \end{aligned}$$

Vynásobme poslední výraz jedničkou ve speciálním tvaru a dostaneme:

$$E_0(f, h) = \frac{h^3}{24} \cdot \underbrace{\frac{b-a}{h} \cdot \frac{1}{m}}_1 \sum_{i=1}^m f''(\xi_i). \quad (2.3.4)$$

Zkoumejme nyní

$$\frac{1}{m} \sum_{i=1}^m f''(\xi_i).$$

Po krátkém rozmyšlení zjistíme, že se jedná o průměr hodnot $f''(\xi_i)$. Zajisté poté $\exists \xi \in (a, b)$ takové, že $f''(\xi)$ je rovno tomuto průměru. Dosadíme do (2.3.4):

$$E_0(f, h) = \frac{b-a}{24} f''(\xi) \cdot h^2, \quad \xi \in (a, b), \quad (2.3.5)$$

a získáváme tak **odhad chyby složeného obdélníkového pravidla**

$$|E_0(f, h)| \leq \frac{b-a}{24} \|f''\|_{\infty} h^2.$$

Odhady chyb složeného lichoběžníkového a Simpsonova pravidla dostaneme analogicky jako (2.3.5) pomocí (2.2.6) a (2.2.10).¹ V tab. 2.2 je uveden přehled odhadů chyb pro složené metody:

pravidlo	odhad chyby
složené obdélníkové (2.3.1)	$ E_0(f, h) \leq \frac{b-a}{24} \ f''\ _{\infty} h^2$
složené lichoběžníkové (2.3.2)	$ E_1(f, h) \leq \frac{b-a}{12} \ f''\ _{\infty} h^2$
složené Simpsonovo (2.3.3)	$ E_2(f, h) \leq \frac{b-a}{180} \ f^{(4)}\ _{\infty} h^4$

Tabulka 2.2: Odhady chyb složených pravidel

¹Pro Simpsonovo pravidlo si jen musíme uvědomit, že $h = (b-a)/(2m)$.

2.4 Formule vyšších řádů

Nyní si předvedeme Newtonovy-Cotesovy formule řádu $n > 2$ až do $n = 10$. Připomeňme, že námi zavedená ekvidistantní síť uzlů splňuje

$$x_i = a + ih, \quad h = \frac{b-a}{n}, \quad i = 0, 1, 2, \dots, n.$$

Pro získání příslušných Lagrangeových polynomů jsme použili software *Maple*, pomocí nativní funkce *PolynomialInterpolation()*. Po následném zintegrování získáváme výrazy (viz také [10]) uvedené v následující tabulce:¹

n	kvadraturní formule
3	$\frac{3}{8}h(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3))$
4	$\frac{2}{45}h(7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4))$
5	$\frac{5}{288}h(19f(x_0) + 75f(x_1) + 50f(x_2) + 50f(x_3) + 75f(x_4) + 19f(x_5))$
6	$\frac{1}{140}h(41f(x_0) + 216f(x_1) + 27f(x_2) + 272f(x_3) + 27f(x_4) + 216f(x_5) + 41f(x_6))$
7	$\frac{7}{17280}h(751f(x_0) + 3577f(x_1) + 1323f(x_2) + 2989f(x_3) + 2989f(x_4) + 1323f(x_5) + 3577f(x_6) + 751f(x_7))$
8	$\frac{4}{14175}h(989f(x_0) + 5888f(x_1) - 928f(x_2) + 10496f(x_3) - 4540f(x_4) + 10496f(x_5) - 928f(x_6) + 5888f(x_7) + 989f(x_8))$
9	$\frac{9}{89600}h(2857(f(x_0) + f(x_9)) + 15741(f(x_1) + f(x_8)) + 1080(f(x_2) + f(x_7)) + 19344(f(x_3) + f(x_6)) + 5778(f(x_4) + f(x_5)))$
10	$\frac{5}{299376}h(16067(f(x_0) + f(x_{10})) + 106300(f(x_1) + f(x_9)) - 48525(f(x_2) + f(x_8)) + 272400(f(x_3) + f(x_7)) - 260550(f(x_4) + f(x_6)) + 427368f(x_5))$

Tabulka 2.3: Newtonovy-Cotesovy formule vyšších řádů

Ačkoliv se jedná o polynomy vyšších řádů a měly by na původní funkci „přilehnout“ a aproximovat blízké body lépe než polynomy do řádu $n < 3$, tak uvidíme, že je mnohem spolehlivější aproximovat integrál pomocí složených pravidel nižších řádů.

¹Pro tyto vzorce v naší práci neuvádíme vzniklé chyby aproximace (ani jejich složené formy), protože s nimi dále v tomto textu nepracujeme.

2.5 Numerické experimenty pro Newtonovy-Cotesovy vzorce

Příklad 2.5.1. Pomocí Newtonových-Cotesových vzorců aproximujme integrál

$$\int_0^4 x^3 \cos(4\pi x) dx.$$

Analytickou hodnotu tohoto integrálu bychom jednoduše určili opakovaným použitím per partes, ovšem my si pomůžeme matematickým softwarem Maple:

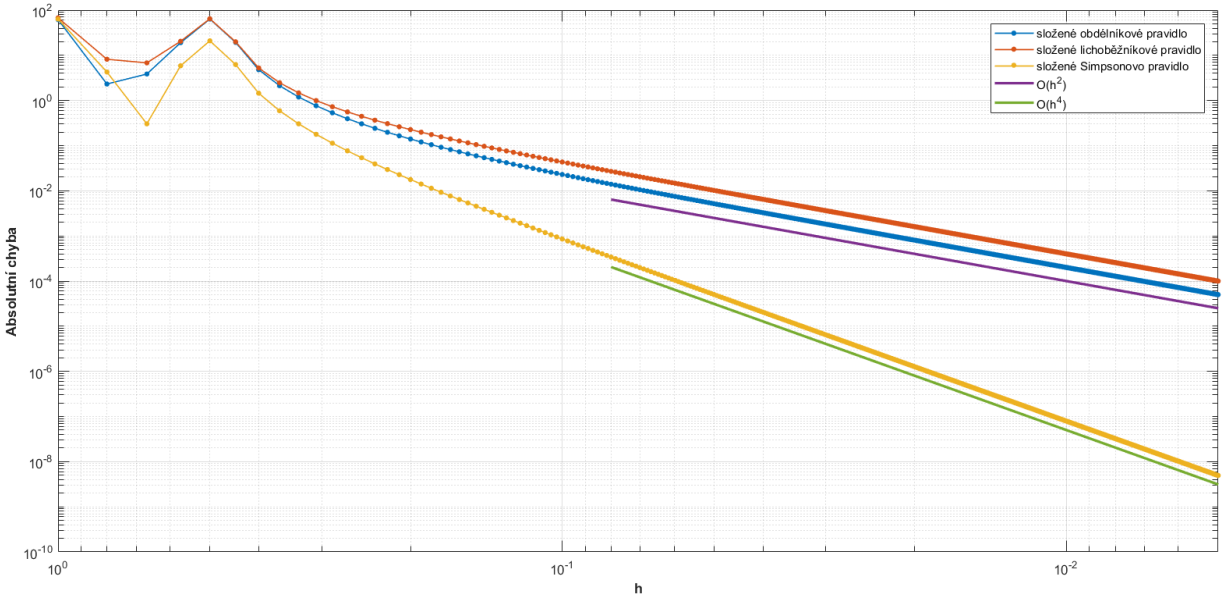
$$\int_0^4 x^3 \cos(4\pi x) dx = \frac{3}{\pi^2}.$$

Označme $f(x) = x^3 \cos(4\pi x)$. Zkoumejme nyní aproximaci integrálu pomocí složených pravidel s daným počtem subintervalů. Vypočtené hodnoty uvádíme v následující tabulce:

m	h	$ O(f, h) - 3/\pi^2 $	$ L(f, h) - 3/\pi^2 $	m	h	$ S(f, h) - 3/\pi^2 $
2	2	55,69603	79,69603	2	1	63,69603
4	1	61,69604	67,69604	4	0,5	63,69604
8	0,5	63,80396	64,69604	8	0,25	20,97063
16	0,25	0,30396	0,44603	16	0,125	0,05396
32	0,125	0,03880	0,07104	32	0,0625	0,00219
64	0,0625	0,00825	0,01612	64	0,03125	$1,2 \cdot 10^{-4}$
128	0,03125	0,00198	0,00394	128	0,01563	$7,6 \cdot 10^{-6}$
256	0,01562	$4,9 \cdot 10^{-4}$	$9,8 \cdot 10^{-4}$	256	0,00781	$4,7 \cdot 10^{-7}$
512	0,00781	$1,2 \cdot 10^{-4}$	$2,4 \cdot 10^{-4}$	512	0,00391	$2,9 \cdot 10^{-8}$
1024	0,00391	$3,1 \cdot 10^{-5}$	$6,1 \cdot 10^{-5}$	1024	0,00195	$1,8 \cdot 10^{-9}$

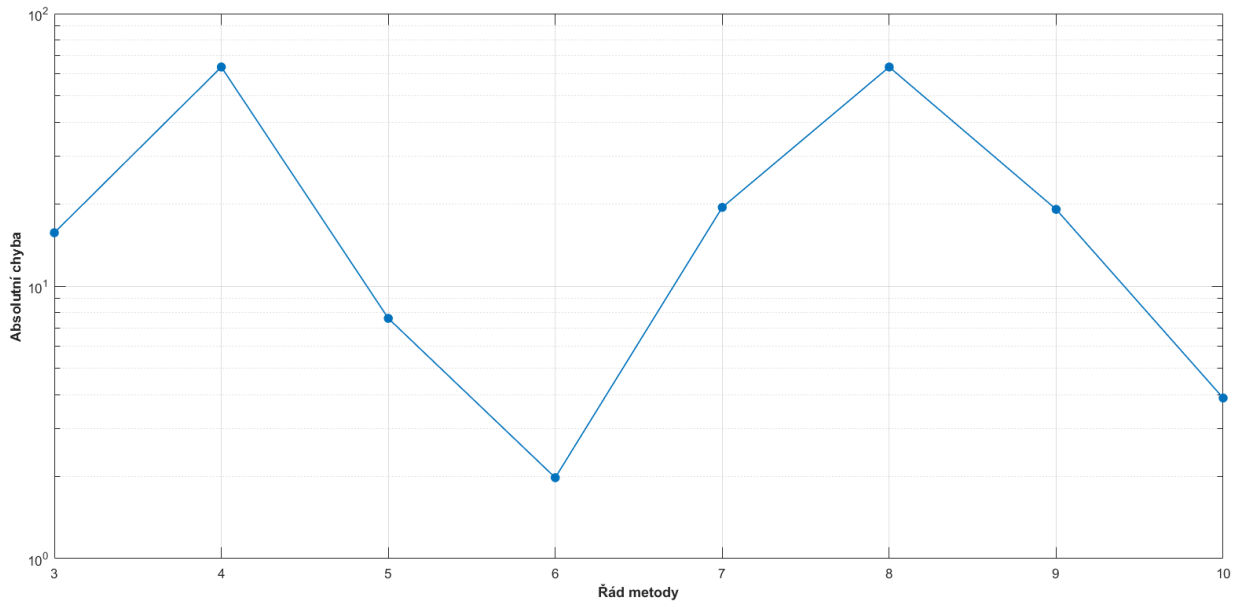
Tabulka 2.4: Aproximace pomocí složených Newtonových-Cotesových formulí I

Pro grafické znázornění konvergence metod nám slouží obr. 2.4, kde absolutní chybou rozumíme absolutní hodnotu rozdílu výsledku dané metody a analytické hodnoty pro dané h .



Obrázek 2.4: Konvergence složených pravidel I

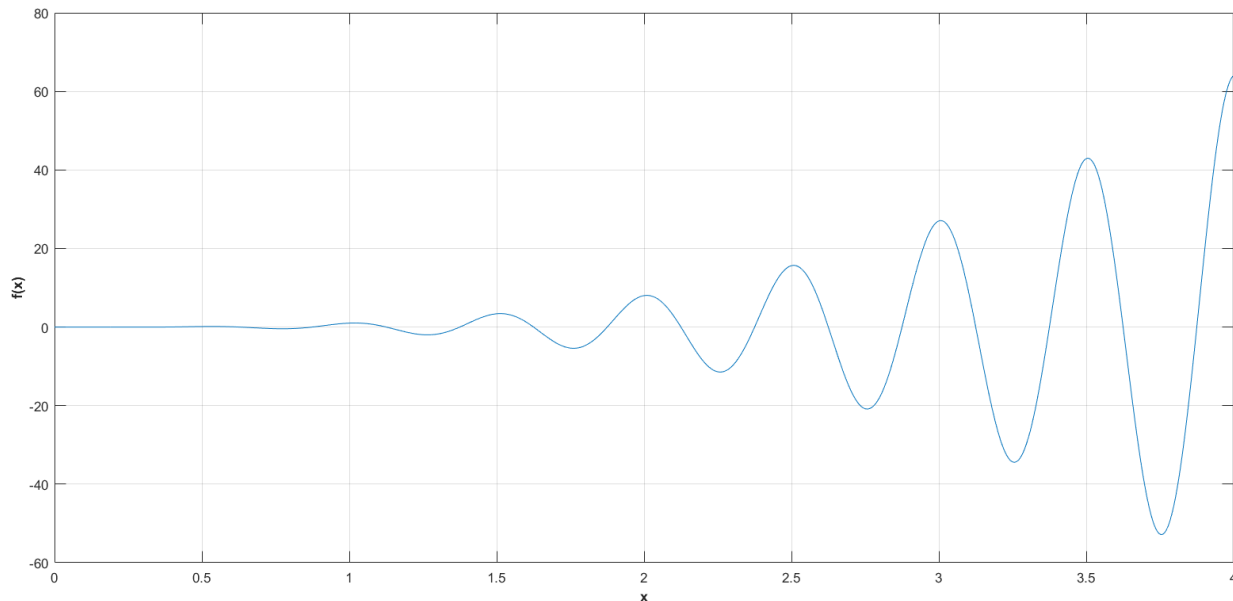
Na obr. 2.4 vidíme, že složené obdélníkové a složené lichoběžníkové pravidlo drží trend $O(h^2)$, zatímco složené Simpsonovo pravidlo konverguje s přesností $O(h^4)$. Průběh aproximace pro Newtonovy-Cotesovy vzorce vyšších řádů jsme nasimulovali na obr. 2.5.



Obrázek 2.5: Chování pravidel vyšších řádů I

Předpokládali bychom, že s větším řádem bude chyba menší, ovšem jak můžeme vidět, tak vyšší

řády nekonvergují. Důvod vidíme na obr. 2.6. Funkce má rychle rostoucí amplitudy, tudíž námi vytvořený Lagrangeův interpolační polynom vyššího řádu „nesedne“ na funkci. Zajisté by pomohlo, kdybychom vytvořili složené tvary těchto metod, ovšem to by mohlo stát za samostatnou bakalářskou prací.



Obrázek 2.6: Průběh funkce $x^3 \cos(4\pi x)$

△

Příklad 2.5.2. Pomocí Newtonových-Cotesových formulí aproximujeme nyní integrál z Rungeovy funkce

$$\int_{-1}^1 \frac{2}{1+x^2} dx.$$

Analytickou hodnotu určíme jednoduchým postupem:

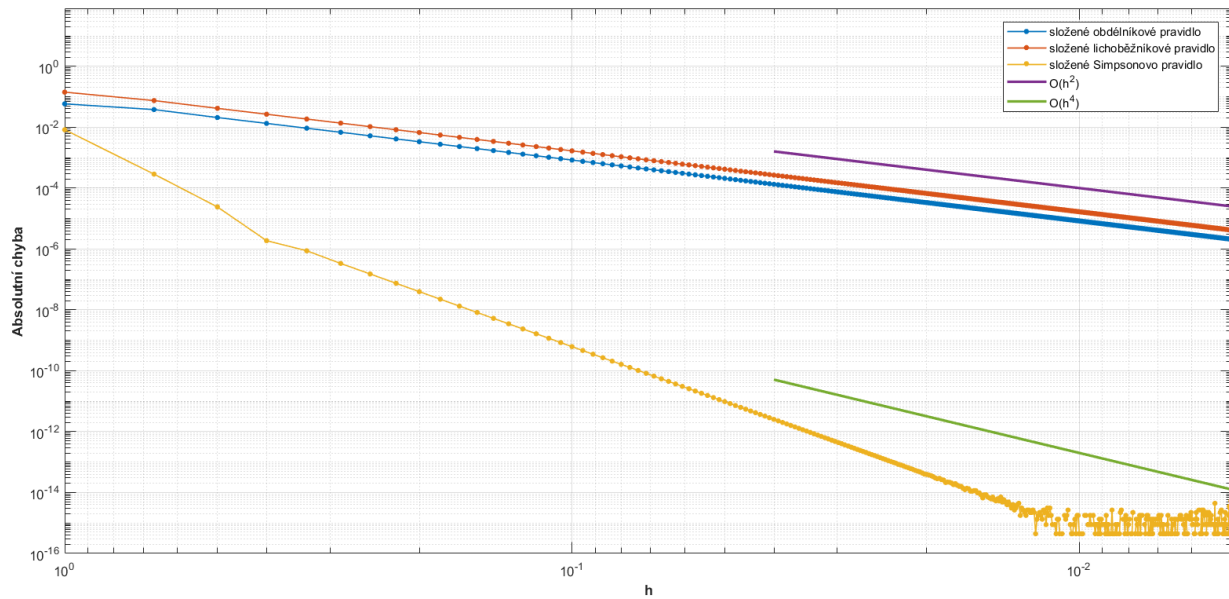
$$\int_{-1}^1 \frac{2}{1+x^2} dx = 2[\operatorname{arctg}(x)]_{-1}^1 = 2\left(\frac{\pi}{4} + \frac{\pi}{4}\right) = \pi.$$

Označme $f(x) = \frac{2}{1+x^2}$ a stejně jako v předchozím příkladu se podívejme na chyby složených pravidel, které máme popsány v následující tabulce:

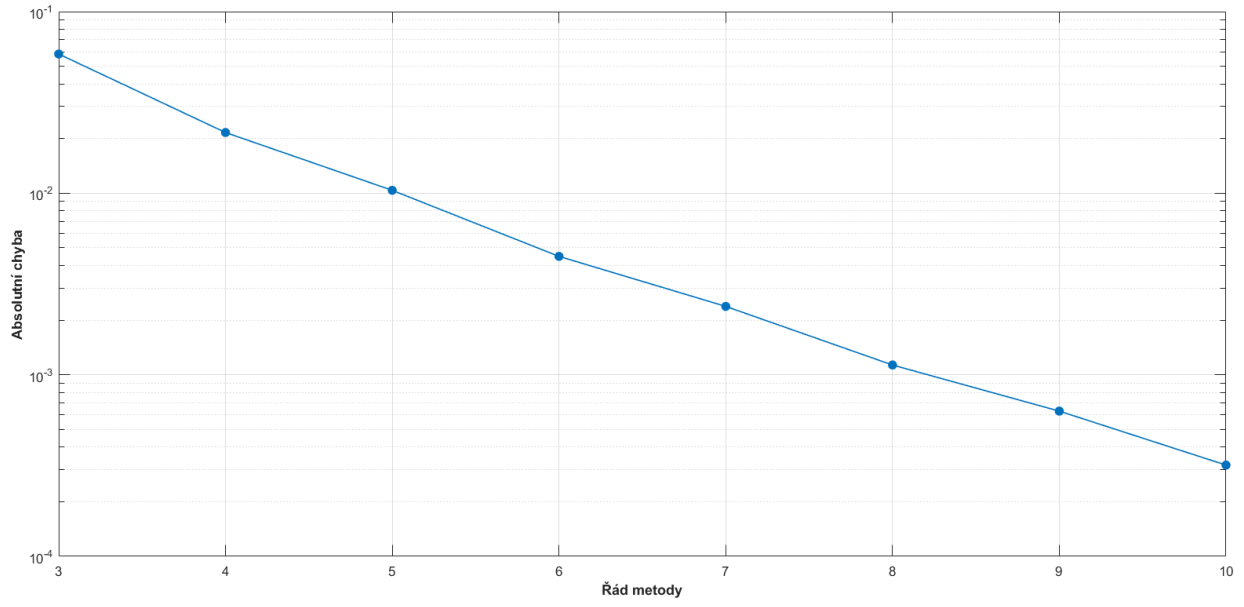
m	h	$ O(f, h) - \pi $	$ L(f, h) - \pi $	m	h	$ S(f, h) - \pi $
2	1	0,05841	0,14159	2	0,5	0,00826
4	0,5	0,02076	0,04159	4	0,25	$2,4 \cdot 10^{-5}$
8	0,25	0,00520	0,01041	8	0,125	$1,5 \cdot 10^{-7}$
16	0,125	0,00130	0,00260	16	0,0625	$2,3 \cdot 10^{-9}$
32	0,0625	$3,3 \cdot 10^{-4}$	$6,5 \cdot 10^{-4}$	32	0,03125	$3,6 \cdot 10^{-11}$
64	0,03125	$8,1 \cdot 10^{-5}$	$1,6 \cdot 10^{-4}$	64	0,01563	$5,7 \cdot 10^{-13}$
128	0,01563	$2,0 \cdot 10^{-5}$	$4,1 \cdot 10^{-5}$	128	0,00781	$8,4 \cdot 10^{-15}$
256	0,00781	$5,1 \cdot 10^{-6}$	$1,0 \cdot 10^{-5}$	256	0,00391	$4,4 \cdot 10^{-16}$
512	0,00391	$1,3 \cdot 10^{-6}$	$2,5 \cdot 10^{-6}$	512	0,00195	$4,4 \cdot 10^{-15}$
1024	0,00195	$3,2 \cdot 10^{-7}$	$6,4 \cdot 10^{-7}$	1024	0,00098	$3,1 \cdot 10^{-15}$

Tabulka 2.5: Aproximace pomocí složených Newtonových-Cotesových formulí II

Konvergenzi jednotlivých pravidel můžeme vidět na obr. 2.7. Důvodem oscilujících hodnot u Simpsonova složeného pravidla je to, že v těchto h dosahujeme již počítačové přesnosti. Oproti minulému příkladu vidíme na obr. 2.8, že chyba klesá s rostoucím řádem pravidel.



Obrázek 2.7: Konvergence složených pravidel II



Obrázek 2.8: Chování pravidel vyšších řádů II

△

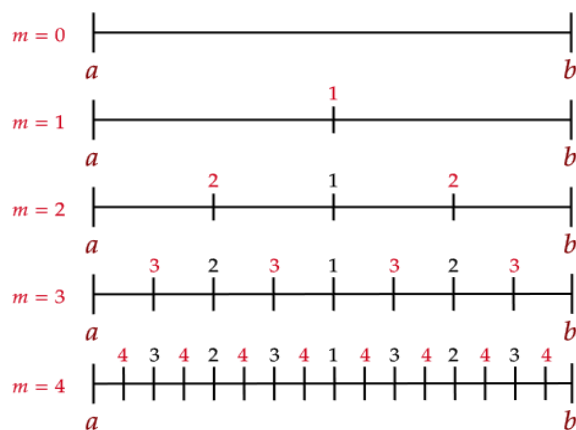
Kapitola 3

Rekurzivní metody

Teď, když již známe složenou formuli pro lichoběžníkové pravidlo, zamysleme se nad tím, jak bychom mohli ještě efektivněji aproximovat daný integrál. Zabývejme se případem, kdy interval $\langle a, b \rangle$ rozdělíme na 2^m subintervalů

$$h_m = \frac{b-a}{2^m}, \quad h_{m+1} = \frac{h_m}{2}, \quad m \in \mathbb{N}_0.$$

Rozložení subintervalů pro různá m jsme vyobrazili na obr. 3.1.



Obrázek 3.1: Půlení intervalů pro rekurzivní lichoběžníkové pravidlo

Můžeme tedy psát, že pro dané m má složená formule lichoběžníkového pravidla tvar

$$L^{[2]}(f, h_m) = h_m \left[\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{i=1}^{2^m-1} f(a + ih_m) \right], \quad (3.0.1)$$

kde horní index $()^{[2]}$ nám říká, že dělíme subintervaly na půl. Pro úroveň $(m + 1)$, $m \geq 0$, píšeme

$$L^{[2]}(f, h_{m+1}) = h_{m+1} \left[\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{i=1}^{2^{m+1}-1} f(a + ih_{m+1}) \right].$$

Předpokládejme, že jsme vypočetli (3.0.1). Všimneme si, že body z m -té úrovně jsou body se sudými indexy na $(m + 1)$ -ní úrovni, tudíž

$$L^{[2]}(f, h_{m+1}) = \frac{h_m}{2} \left[\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{i=1}^{2^m-1} f(a + ih_m) + \sum_{i=0}^{2^m-1} f(a + (2i + 1)h_{m+1}) \right],$$

z čehož získáváme

$$L^{[2]}(f, h_{m+1}) = \frac{L^{[2]}(f, h_m)}{2} + h_{m+1} \sum_{i=0}^{2^m-1} f(a + (2i + 1)h_{m+1}). \quad (3.0.2)$$

Vztah (3.0.2) nazveme **rekurzivním lichoběžníkovým pravidlem s dělením na poloviny** (viz také [7]). Jak vidíme, tato metoda využívá již dříve vypočtené body, což dělá formuli velmi efektivní co se výpočetního času týče.

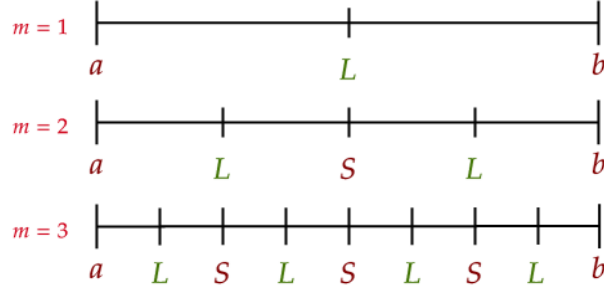
Inspirujme se a obdobně zavedme **rekurzivní Simpsonovo pravidlo s dělením na poloviny**. Nesmíme zapomenout, že u Simpsonova pravidla máme $2m$ subintervalů, ale aproximujeme na intervalech délky $2h_m$, tudíž volíme $m \geq 1$. Víme (viz (2.3.3)), že

$$S^{[2]}(f, h_m) = \frac{h_m}{3} \left[f(a) + f(b) + 2 \sum_{i=1}^{2^{m-1}-1} f(a + 2ih_m) + 4 \sum_{i=1}^{2^{m-1}} f(a + (2i - 1)h_m) \right]$$

a

$$S^{[2]}(f, h_{m+1}) = \frac{h_{m+1}}{3} \left[f(a) + f(b) + 2 \sum_{i=1}^{2^m-1} f(a + 2ih_{m+1}) + 4 \sum_{i=1}^{2^m} f(a + (2i - 1)h_{m+1}) \right].$$

Na obr. 3.2 jsme nasimulovali dělení intervalů pro různé úrovně m . Můžeme vidět, že sudé body (ozn. S) a liché body (ozn. L) z minulé úrovně se v následné iteraci stávají sudými body.



Obrázek 3.2: Půlení intervalů pro rekurzivní Simpsonovo pravidlo

Tudíž platí, že

$$\sum_{i=1}^{2^{m-1}} f(a + 2ih_{m+1}) = \sum_{i=1}^{2^{m-1}-1} f(a + 2ih_m) + \sum_{i=1}^{2^{m-1}} f(a + (2i-1)h_m),$$

a proto

$$2 \sum_{i=1}^{2^{m-1}} f(a + 2ih_{m+1}) = 2 \sum_{i=1}^{2^{m-1}-1} f(a + 2ih_m) + 4 \sum_{i=1}^{2^{m-1}} f(a + (2i-1)h_m) - 2 \sum_{i=1}^{2^{m-1}} f(a + (2i-1)h_m).$$

Dosazením do $S^{[2]}(f, h_{m+1})$ získáváme

$$S^{[2]}(f, h_{m+1}) = \frac{h_{m+1}}{3} \left[f(a) + f(b) + 2 \sum_{i=1}^{2^{m-1}-1} f(a + 2ih_m) + 4 \sum_{i=1}^{2^{m-1}} f(a + (2i-1)h_m) - 2 \sum_{i=1}^{2^{m-1}} f(a + (2i-1)h_m) + 4 \sum_{i=1}^{2^m} f(a + (2i-1)h_{m+1}) \right]$$

a úpravou námi hledaný rekurzivní vztah pro Simpsonovo pravidlo s dělením na poloviny:

$$S^{[2]}(f, h_{m+1}) = \frac{S^{[2]}(f, h_m)}{2} + \frac{h_{m+1}}{3} \left[4 \sum_{i=1}^{2^m} f(a + (2i-1)h_{m+1}) - 2 \underbrace{\sum_{i=1}^{2^{m-1}} f(a + (2i-1)h_m)}_{\text{vypočteno v } m\text{-té úrovni}} \right].$$

3.1 Dělení subintervalů na třetiny

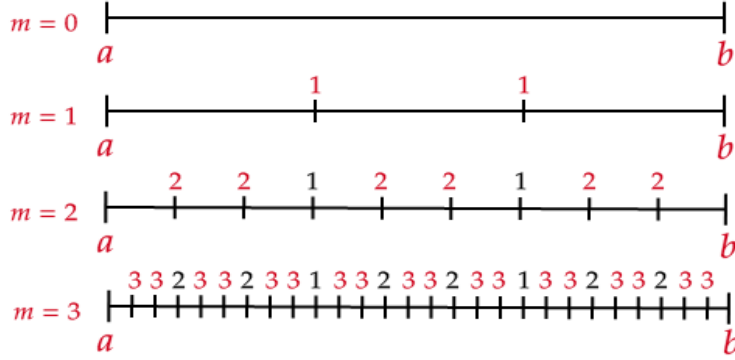
Ač-li je dělení na poloviny pro nás přirozenější, můžeme obecně dělit subintervaly na libovolný počet dílků. Uvedme ku příkladu dělení na třetiny. Položme

$$h_m = \frac{b-a}{3^m}, \quad h_{m+1} = \frac{h_m}{3}, \quad m \in \mathbb{N}_0.$$

Hledáme **lichoběžníkové pravidlo s dělením na třetiny**. Zajisté platí, že

$$L^{[3]}(f, h_m) = h_m \left[\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{i=1}^{3^m-1} f(a + ih_m) \right]. \quad (3.1.1)$$

Na obr. 3.3 vidíme, že při dělení subintervalu na třetiny nám jej rozdělí vždy dva nové body. Hodnoty v nových bodech bychom tudíž mohli efektivně spočítat tak, že každý třetí bod přeskočíme.



Obrázek 3.3: Rozdělení intervalů na třetiny pro rekurzivní lichoběžníkové pravidlo

Tudíž můžeme psát, že

$$\begin{aligned} L^{[3]}(f, h_{m+1}) &= h_{m+1} \left[\frac{f(a)}{2} + \frac{f(b)}{2} + \sum_{i=1}^{3^m-1} f(a + ih_m) + \sum_{\substack{i=1 \\ i \bmod 3 \neq 0}}^{3^{m+1}-1} f(a + ih_{m+1}) \right] = \\ &= \frac{L^{[3]}(f, h_m)}{3} + h_{m+1} \sum_{\substack{i=1 \\ i \bmod 3 \neq 0}}^{3^{m+1}-1} f(a + ih_{m+1}), \end{aligned}$$

což je námi hledaný předpis.

Změňme nyní velikost h_m na

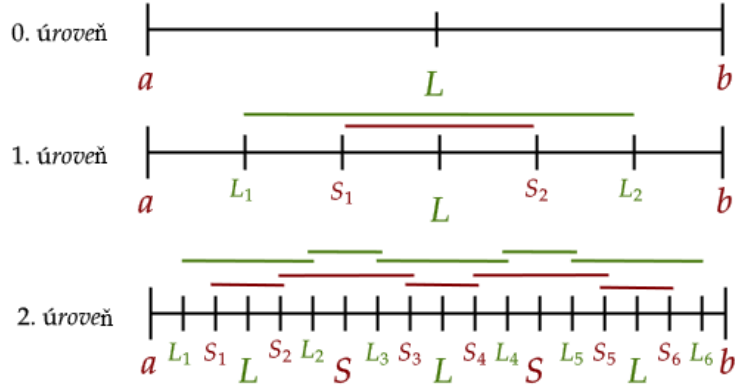
$$h_m = \frac{b-a}{2 \cdot 3^m}, \quad h_{m+1} = \frac{h_m}{3}, \quad m \in \mathbb{N}_0$$

a hledejme **rekurzivní Simpsonovo pravidlo s dělením na třetiny**. Víme, že

$$S^{[3]}(f, h_m) = \frac{h_m}{3} \left[f(a) + f(b) + 2 \sum_{i=1}^{3^m-1} f(a + 2ih_m) + 4 \sum_{i=1}^{3^m} f(a + (2i-1)h_m) \right].$$

Pro určení další iterace se zamyslíme nad obr. 3.4. Všimněme si, že v každém dalším kroku nám vznikne sudý počet bodů s ozn. L_i (nové liché body) a sudý počet bodů s ozn. S_i (nové sudé body). Z obrázku si není těžké uvědomit, jakým způsobem jsou body rozloženy. Body L_i (resp. S_i) jsou od

sebe střídatavě vzdáleny o $4h_m$ (resp. o $2h_m$) a následně o $2h_m$ (resp. p $4h_m$), jak zobrazují úsečky korespondující barvy.



Obrázek 3.4: Rozdělení intervalů na třetiny pro rekurzivní Simpsonovo pravidlo

Můžeme tudíž psát, že pro $(m + 1)$ -ní, $m \geq 0$, úroveň platí

$$L_i = \begin{cases} a + h_{m+1}, & i = 1, \\ L_{i-1} + 4h_{m+1}, & i \bmod 2 = 0, \\ L_{i-1} + 2h_{m+1}, & i \bmod 2 = 1, \end{cases} \quad i = 1, 2, \dots, 2 \cdot 3^m,$$

resp.

$$S_i = \begin{cases} a + 2h_{m+1}, & i = 1, \\ S_{i-1} + 2h_{m+1}, & i \bmod 2 = 0, \\ S_{i-1} + 4h_{m+1}, & i \bmod 2 = 1, \end{cases} \quad i = 1, 2, \dots, 2 \cdot 3^m.$$

S touto informací se nyní zaměříme na

$$S^{[3]}(f, h_{m+1}) = \frac{h_{m+1}}{3} \left[f(a) + f(b) + 2 \sum_{i=1}^{3^{m+1}-1} f(a + 2ih_{m+1}) + 4 \sum_{i=1}^{3^m} f(a + (2i - 1)h_{m+1}) \right].$$

Z obr. 3.4 je vidět, že

$$\sum_{i=1}^{3^{m+1}-1} f(a + 2ih_{m+1}) = \underbrace{\sum_{i=1}^{3^m-1} f(a + 2ih_m)}_{\text{vypočteno v } m\text{-té úrovni}} + \sum_{i=1}^{2 \cdot 3^m} f(S_i)$$

a

$$\sum_{i=1}^{3^{m+1}} f(a + (2i - 1)h_{m+1}) = \underbrace{\sum_{i=1}^{3^m} f(a + (2i - 1)h_m)}_{\text{vypočteno v } m\text{-té úrovni}} + \sum_{i=1}^{2 \cdot 3^m} f(L_i).$$

Pomocí elementárních úprav tudíž dostáváme

$$S^{[3]}(f, h_{m+1}) = \frac{S^{[3]}(f, h_m)}{3} + \frac{h_{m+1}}{3} \left[2 \sum_{i=1}^{2 \cdot 3^m} f(S_i) + 4 \sum_{i=1}^{2 \cdot 3^m} f(L_i) \right].$$

3.2 Numerické experimenty pro rekurzivní metody

Příklad 3.2.1. *Se znalostí rekurzivních metod aproximujme*

$$\int_0^1 4\sqrt{1-x^2} dx.$$

Analytickou hodnotu tohoto integrálu vypočteme následovně:

$$\int_0^1 4\sqrt{1-x^2} dx = 4 \underbrace{\int_0^1 \sqrt{1-x^2} dx}_{\text{obsah čtvrtiny jednotkové kružnice}} = \pi = 3,141592653589793 \dots$$

Označme aproximaci z m -té úrovně jako A_m a aproximace z $(m-1)$ -ní úrovně jako A_{m-1} , poté definujme relativní chybu jako

$$\frac{|A_{m-1} - A_m|}{|A_m|}. \quad (3.2.1)$$

Toto zavedení využijeme jako ukončovací podmínku: rekurze bude trvat do té doby, dokud bude relativní chyba (3.2.1) menší než 10^{-5} . Získáváme výsledky uvedené v následující tabulce:

metoda	výsledná aproximace	max. dosažená úroveň	absolutní chyba
$L^{[2]}$	3,141579965411448	11	$1,2 \cdot 10^{-5}$
$L^{[3]}$	3,141590440782387	8	$2,2 \cdot 10^{-6}$
$S^{[2]}$	3,141578637812139	9	$1,4 \cdot 10^{-5}$
$S^{[3]}$	3,141591066012415	7	$1,5 \cdot 10^{-6}$

Tabulka 3.1: Aproximace integrálu pomocí rekurzivních metod

Vidíme, že nejlepší aproximace získáváme díky $L^{[3]}$ a $S^{[3]}$. Jestliže porovnáme $L^{[2]}$ a $S^{[3]}$, tak vidíme, že $S^{[3]}$ dosáhlo o řád lepší aproximace skoro již za polovinu maximální dosažené úrovně.

△

Kapitola 4

Eulerova-Maclaurinova expanze

Zkoumejme znovu chybu složeného lichoběžníkového pravidla (viz tab. 2.2 na str. 22)

$$\left| \int_a^b f(x) dx - L(f, h) \right| \leq \frac{b-a}{12} \|f''\|_{\infty} h^2.$$

Všimněme si, že se jedná o metodu 2. řádu, nebo-li že chyba je třídy $\mathcal{O}(h^2)$. V této kapitole se zabýváme **Eulerovou-Maclaurinovou expanzí**, která tuto chybu vyjadřuje jako řadu mocnin h^2 , a umožňuje další zpřesnění pomocí extrapoláčnických technik (viz také [2, 8]).

Definice 4.0.1. Uvažujme posloupnost polynomů q_r , $r \in \mathbb{N}$, definovanou pomocí:

- (i) $\deg(q_r) = r$,
- (ii) $\forall r \in \mathbb{N} : q'_{r+1} = q_r$,
- (iii) q_r je $\begin{cases} \text{lichá funkce,} & r \text{ je liché,} \\ \text{sudá funkce,} & r \text{ je sudé,} \end{cases}$
- (iv) pro $r > 1$ liché platí: $q_r(-1) = q_r(1) = 0$,
- (v) $q_1(t) = -t$.

Pro ukázkou uveďme výpočet prvních pěti polynomů. Z definice víme, že

$$q_1(t) = -t.$$

Nyní si stačí uvědomit, že

$$q_2(t) = \int q_1(t) dt = -\frac{t^2}{2} + A_2, \quad A_2 \in \mathbb{R},$$
$$q_3(t) = \int q_2(t) dt = -\frac{t^3}{6} + A_2 t + A_3, \quad A_3 \in \mathbb{R}.$$

Jelikož z definice plyne, že q_3 je lichá funkce, tak nutně musí z lichosti platit, že

$$q_3(0) = 0 \quad \Rightarrow \quad A_3 = 0.$$

Dále vidíme, že

$$q_3(1) = -\frac{1}{6} + A_2 = 0 \quad \Rightarrow \quad A_2 = \frac{1}{6}.$$

Tedy:

$$q_2(t) = -\frac{t^2}{2} + \frac{1}{6}, \quad q_3(t) = \frac{t^3}{6} + \frac{t}{6}.$$

Obdobným způsobem vypočteme další dva polynomy:

$$\begin{aligned} q_4(t) &= -\frac{t^4}{24} + \frac{t^2}{12} + A_4, \quad A_4 \in \mathbb{R}, \\ q_5(t) &= -\frac{t^5}{120} + \frac{t^3}{36} + A_4 t + A_5, \quad A_5 \in \mathbb{R}, \\ q_5(0) &= 0 \quad \Rightarrow \quad A_5 = 0, \\ q_5(1) &= -\frac{1}{120} + \frac{1}{36} + A_4 = 0 \quad \Rightarrow \quad A_4 = -\frac{7}{360}. \end{aligned}$$

Tedy:

$$q_4(t) = -\frac{t^4}{24} + \frac{t^2}{12} - \frac{7}{360}, \quad q_5(t) = -\frac{t^5}{120} + \frac{t^3}{36} - \frac{7t}{360}.$$

Věta 4.0.1. *Nechť $g \in C^{2k}((-1, 1))$, poté*

$$\begin{aligned} \int_{-1}^1 g(t) dt - (g(-1) + g(1)) &\stackrel{(*)}{=} \int_{-1}^1 -tg'(t) dt \stackrel{(\circ)}{=} \\ &\stackrel{(\circ)}{=} \sum_{r=1}^k q_{2r}(1) \left(g^{(2r-1)}(1) - g^{(2r-1)}(-1) \right) - \int_{-1}^1 q_{2k}(t) g^{(2k)}(t) dt. \end{aligned}$$

Důkaz. [8] *Všimněme si, že rozdíl*

$$\int_{-1}^1 g(t) dt - (g(-1) + g(1))$$

je chyba lichoběžníkového pravidla pro daný integrál. Podívejme se nyní na rovnost (). Je zřejmé, že*

$$\begin{aligned} \int_{-1}^1 -tg'(t) dt &= \left| \begin{array}{l} \text{per partes} \\ u = -t, \quad u' = -1 \\ v' = g', \quad v = g \end{array} \right| = [-tg(t)]_{-1}^1 + \int_{-1}^1 g(t) dt = \\ &= -(g(1) + g(-1)) + \int_{-1}^1 g(t) dt. \end{aligned}$$

Nyní dokažme rovnost (\diamond):

$$\begin{aligned}
\int_{-1}^1 q_1(t)g'(t) dt &= \left| \begin{array}{l} \text{per partes} \\ u = g', \quad u' = g'' \\ v' = q_1, \quad v = q_2 \end{array} \right| = [g'(t)q_2(t)]_{-1}^1 - \int_{-1}^1 g''(t)q_2(t) dt = \\
&= q_2(1)(g'(1) - g'(-1)) - \int_{-1}^1 g''(t)q_2(t) dt = \\
&= \left| \begin{array}{l} \text{per partes} \\ u = g'', \quad u' = g''' \\ v' = q_2, \quad v = q_3 \end{array} \right| = \\
&= q_3(1)(g''(1) - g''(-1)) - [g''(t)q_3(t)]_{-1}^1 + \int_{-1}^1 g'''(t)q_3(t) dt = \\
&= \left| \begin{array}{l} \text{per partes} \\ u = g''', \quad u' = g^{(4)} \\ v' = q_3, \quad v = q_4 \end{array} \right| = \\
&= q_4(1)(g'''(1) - g'''(-1)) - \int_{-1}^1 g^{(4)}(t)q_4(t) dt = \\
&= \sum_{r=1}^2 q_{2r}(1) \left(g^{(2r-1)}(1) - g^{(2r-1)}(-1) \right) - \int_{-1}^1 g^{(2\cdot 2)}(t)q_{2\cdot 2}(t) dt.
\end{aligned}$$

Pokud bychom metodu per partes udělali $(k+1)$ -krát, tak získáváme požadovanou rovnost

$$\int_{-1}^1 -tg'(t) dt = \sum_{r=1}^k q_{2r}(1) \left(g^{(2r-1)}(1) - g^{(2r-1)}(-1) \right) - \int_{-1}^1 g^{(2k)}(t)q_{(2k)}(t) dt.$$

■

Věta 4.0.2 (Eulerova-Maclaurinova expanze). [8] Necht $f \in C^{2k}(\langle a, b \rangle)$,

$$\langle a, b \rangle = \bigcup_{i=1}^m \langle x_{i-1}, x_i \rangle, \quad x_i = a + ih, \quad i = 0, 1, \dots, m, \quad h = \frac{b-a}{m}, \quad m \geq 1$$

a

$$I = \int_a^b f(x) dx.$$

Pak

$$I - L(f, h) = \sum_{r=1}^k c_r h^{2r} \left(f^{(2r-1)}(b) - f^{(2r-1)}(a) \right) - \left(\frac{h}{2} \right)^{2k} \sum_{i=1}^m \int_{x_{i-1}}^{x_i} q_{2k}(t) f^{2k}(x) dx,$$

kde

$$t = t(x) = -1 + \frac{2}{h}(x - x_{i-1}), \quad x \in \langle x_{i-1}, x_i \rangle, \quad c_r = \frac{q_{2r}(1)}{2^{2r}}, \quad r = 1, 2, \dots, k.$$

Důkaz. Platí

$$\int_{x_{i-1}}^{x_i} f(x) dx = \left| \begin{array}{l} \text{substituce} \\ x = x_{i-1} + \frac{h}{2}(t+1) \\ dx = \frac{h}{2} dt \\ x = x_{i-1} \Rightarrow t = -1 \\ x = x_i \Rightarrow x_i = x_{i-1} + \frac{h}{2}(t+1) \Rightarrow t = 1 \end{array} \right| = \frac{h}{2} \int_{-1}^1 g(t) dt, \quad (4.0.1)$$

kde $g(t) \stackrel{\text{ozn.}}{=} f(x(t))$, a zabýváme se chybou na jednom subintervalu

$$\int_{x_{i-1}}^{x_i} f(x) dx - \frac{h}{2} (f(x_{i-1}) + f(x_i)) = \frac{h}{2} \left[\int_{-1}^1 g(t) dt - (g(-1) + g(1)) \right]. \quad (4.0.2)$$

Z věty 4.0.1 plyne, že pravou stranu (4.0.2) můžeme přepsat do tvaru

$$\frac{h}{2} \sum_{r=1}^k q_{2r}(1) \left(g^{(2r-1)}(1) - g^{(2r-1)}(-1) \right) - \frac{h}{2} \int_{-1}^1 q_{2k}(t) g^{(2k)}(t) dt.$$

To ale znamená, že

$$\begin{aligned} I - L(f, h) &= \sum_{i=1}^m \int_{x_{i-1}}^{x_i} f(x) dx - \frac{h}{2} (f(x_{i-1}) + f(x_i)) = \\ &= \frac{h}{2} \sum_{r=1}^k q_{2r}(1) \sum_{i=1}^m \left(g^{(2r-1)}(1) - g^{(2r-1)}(-1) \right) - \frac{h}{2} \sum_{i=1}^m \int_{-1}^1 q_{2k}(t) g^{(2k)}(t) dt. \end{aligned} \quad (4.0.3)$$

Nyní se podívejme na první člen za poslední rovností. Uvědomme si, že

$$g^{(p)}(t) = f^{(p)}(x(t)) \cdot \left(\frac{h}{2} \right)^p, \quad p \in \mathbb{N}, \quad (4.0.4)$$

protože se jedná o derivaci složené funkce (viz substituci v (4.0.1)). Můžeme tedy psát:

$$\begin{aligned} \frac{h}{2} \sum_{r=1}^k q_{2r}(1) \sum_{i=1}^m \left(f^{(2r-1)}(x_i) \left(\frac{h}{2} \right)^{2r-1} - f^{(2r-1)}(x_{i-1}) \left(\frac{h}{2} \right)^{2r-1} \right) = \\ \frac{h}{2} \sum_{r=1}^k q_{2r}(1) \left(\frac{h}{2} \right)^{2r-1} \sum_{i=1}^m \left(f^{(2r-1)}(x_i) - f^{(2r-1)}(x_{i-1}) \right). \end{aligned}$$

Dále vidíme, že

$$\sum_{i=1}^m \left(f^{(2r-1)}(x_i) - f^{(2r-1)}(x_{i-1}) \right) = f^{(2r-1)}(b) - f^{(2r-1)}(a),$$

protože se odečtou derivace v uzlech x_1, x_2, \dots, x_{m-1} (jednou jsou zde tyto derivace s kladným a jednou se záporným znaménkem). Tudíž

$$\frac{h}{2} \sum_{r=1}^k q_{2r}(1) \sum_{i=1}^m \left(g^{(2r-1)}(1) - g^{(2r-1)}(-1) \right) = \sum_{r=1}^k c_r h^{2r} \left(f^{(2r-1)}(b) - f^{(2r-1)}(a) \right), \quad (4.0.5)$$

kde

$$c_r = \frac{g_{2r}(1)}{2^{2r}}.$$

Nyní se zaměříme na druhý člen za poslední rovností v (4.0.3). Zpětnou substitucí

$$t = t(x) = -1 + \frac{2}{h}(x - x_{i-1}), \quad dt = \frac{2}{h} dx$$

dostáváme pomocí (4.0.4)

$$\frac{h}{2} \sum_{i=1}^m \int_{-1}^1 q_{2k}(t) g^{(2k)}(t) dt = \sum_{i=1}^m \left(\frac{h}{2} \right)^{2k} \int_{x_{i-1}}^{x_i} q_{2k}(t) f^{2k}(x) dx, \quad t = t(x). \quad (4.0.6)$$

Odečtením (4.0.6) od (4.0.5) získáváme

$$I - L(f, h) = \sum_{r=1}^k c_r h^{2r} \left(f^{(2r-1)}(b) - f^{(2r-1)}(a) \right) - \left(\frac{h}{2} \right)^{2k} \sum_{i=1}^m \int_{x_{i-1}}^{x_i} q_{2k}(t) f^{2k}(x) dx.$$

■

Podívejme se nyní na hodnoty

$$c_r = \frac{q_{2r}(1)}{2^{2r}}.$$

Připomeňme, že

$$q_2(t) = -\frac{t^2}{2} + \frac{1}{6}, \quad q_4(t) = -\frac{t^4}{24} + \frac{t^2}{12} - \frac{7}{360}.$$

Dostáváme tudíž pro první dvě r hodnoty c_r :

$$\begin{aligned} r = 1 &\Rightarrow q_2(1) = -\frac{1}{2} + \frac{1}{6} = -\frac{1}{3} \Rightarrow c_1 = -\frac{1}{12}, \\ r = 2 &\Rightarrow q_4(1) = -\frac{1}{24} + \frac{1}{12} - \frac{7}{360} \Rightarrow c_2 = \frac{1}{720}. \end{aligned}$$

Tímto postupem bychom dostali i další hodnoty c_r :

$$c_3 = -\frac{1}{30240}, \quad c_4 = \frac{1}{1209600}, \quad c_5 = -\frac{1}{47900160}.$$

Lze navíc ukázat, že

$$c_r = \frac{B_{2r}}{(2r)!}, \quad r \in \mathbb{N},$$

kde B_{2r} jsou tzv. **Bernoulliova čísla** (viz také [8]) se sudými indexy, která se dají získat z Taylorova rozvoje:

$$\frac{x}{2} \coth\left(\frac{x}{2}\right) = \sum_{r=0}^{\infty} \frac{B_{2r} x^{2r}}{(2r)!}.$$

Kapitola 5

Richardsonova extrapolace

Teď, když již víme, jak můžeme pomocí Eulerovy-Maclaurinovy expanze vyjádřit chybu složeného lichoběžníkového pravidla, tak se zamyslíme, zda-li bychom mohli využít této znalosti k zpřesnění aproximace. Připomeňme, že Eulerova-Maclaurinova expanze je ve tvaru

$$\int_a^b f(x) dx - L(f, h) = \sum_{r=1}^k c_r h^{2r} (f^{(2r-1)}(b) - f^{(2r-1)}(a)) - \left(\frac{h}{2}\right)^{2k} \sum_{i=1}^m \int_{x_{i-1}}^{x_i} q_{2k}(t) f^{(2k)}(x) dx,$$

kde

$$f \in C^{(2k)}(\langle a, b \rangle), \quad h = \frac{b-a}{m}, \quad m \geq 1$$

Uvažujme $k = 3$, pak můžeme psát, že

$$\int_a^b f(x) dx - L(f, h) = C_1 h^2 + C_2 h^4 + \mathcal{O}(h^6), \quad (5.0.1)$$

kde

$$C_1 = c_1 (f'(a) - f'(b)), \quad C_2 = c_2 (f'''(a) - f'''(b)).$$

Jak vidíme, (5.0.1) je aproximací 2. řádu, což platí i pro $h/2$:

$$\int_a^b f(x) dx - L\left(f, \frac{h}{2}\right) = C_1 \left(\frac{h}{2}\right)^2 + C_2 \left(\frac{h}{2}\right)^4 + \mathcal{O}(h^6). \quad (5.0.2)$$

Pokusme se přijít na metodu, s kterou bychom dostali lepší přesnost. Abychom aproximaci zlepšili, tak musíme první člen pravé strany rovností (5.0.1) a (5.0.2) odstranit. Pokud tedy (5.0.2) vynásobíme 2^2 a následně odečteme od (5.0.1), tak získáváme

$$(5.0.1) - 2^2 \cdot (5.0.2) : \quad (1 - 2^2) \int_a^b f(x) dx - L(f, h) + 2^2 L\left(f, \frac{h}{2}\right) = \left(C_2 - \frac{C_2}{2^2}\right) h^4 + \mathcal{O}(h^6)$$

a úpravou dostáváme aproximaci následující 4. řádu

$$\int_a^b f(x) dx - \frac{2^2 L\left(f, \frac{h}{2}\right) - L(f, h)}{2^2 - 1} = \bar{C}_2 h^4 + \mathcal{O}(h^6), \quad (5.0.3)$$

kde

$$\bar{C}_2 = \frac{C_2 - \frac{C_2}{2^2}}{1 - 2^2} = -\frac{C_2}{2^2}.$$

Označme

$$L_1(f, h) \stackrel{\text{ozn.}}{=} \frac{2^2 L\left(f, \frac{h}{2}\right) - L(f, h)}{2^2 - 1}$$

a procesu, díky kterému jsme se dostali k (5.0.3), říkáme **Richardsonova extrapolace** (viz také [7, 8]).

Zkusme obdobným způsobem ještě lépe zpřesnit aproximaci:

$$\int_a^b f(x) dx - L_1(f, h) = \bar{C}_2 h^4 + \mathcal{O}(h^6), \quad (5.0.4)$$

$$\int_a^b f(x) dx - L_1\left(f, \frac{h}{2}\right) = \bar{C}_2 \left(\frac{h}{2}\right)^4 + \mathcal{O}(h^6). \quad (5.0.5)$$

Zde si musíme uvědomit, že je třeba vynásobit (5.0.5) číslem 2^4 :

$$(5.0.4) - 2^4 \cdot (5.0.5) : \quad (1 - 2^4) \int_a^b f(x) dx - L_1(f, h) + 2^4 L_1\left(f, \frac{h}{2}\right) = \mathcal{O}(h^6).$$

Úpravou získáváme

$$\int_a^b f(x) dx - \underbrace{\frac{2^4 L_1\left(f, \frac{h}{2}\right) - L_1(f, h)}{2^4 - 1}}_{\text{ozn. } L_2(f, h)} = \mathcal{O}(h^6),$$

z čehož vidíme, že jsme dostali aproximaci 6. řádu. Pro další použití extrapolace bychom potřebovali $k > 3$ (funkce není „dostatečně hladká“). Označíme-li

$$L_0(f, h) \stackrel{\text{ozn.}}{=} L(f, h),$$

tak pro dostatečně hladkou funkci f dostáváme

$$L_k(f, h) = \frac{4^k L_{k-1}\left(f, \frac{h}{2}\right) - L_{k-1}(f, h)}{4^k - 1}, \quad k = 1, 2, 3, \dots, \quad (5.0.6)$$

kde přesnost aproximace je $\mathcal{O}(2^{2k+2})$. Tvar (5.0.6) jsme získali díky lichoběžníkovému pravidlu a Eulerově-Maclaurinově expanzi, ovšem to je již přímá aplikace, která nám poslouží jako základní stavební kámen pro Rombergovu metodu.

Richardsonova extrapolace ovšem má své využití také při aproximování řešení diferenciální rovnice či aproximace derivace (viz také [2, 6]). Obecně tento nástroj slouží ke zpřesnění numerických metod a byla by škoda, kdybychom si neukázali jeho obecný tvar, který si nyní odvodíme (viz také [2]).

Předpokládejme, že chceme získat hodnotu ξ , kterou můžeme aproximovat pomocí nějaké dostatečně hladké funkce Φ , která reprezentuje algoritmus závislý na h , a platí $\Phi(h) \rightarrow \Phi(0) = \xi$ pro $h \rightarrow 0$. Přírozenou cestou pro získání lepší přesnosti je hodnotu h stále zmenšovat a získávat tím nové hodnoty Φ , ovšem může se stát, že je výpočet $\Phi(h)$ s malým h velmi časově náročný či zde mohou nastat chyby v zaokrouhlování v plovoucí řádové čárce, které poškodí přesnost algoritmu. Využijme předpokladu hladkosti Φ a pomocí Taylorovy řady získáváme

$$\Phi(h) = \Phi(0) + h\Phi'(0) + \frac{1}{2}h^2\Phi''(0) + \frac{1}{6}h^3\Phi'''(0) + \dots$$

Víme, že $\Phi(0) = \xi$, a jestliže položíme konstanty $\{c_j\}_1^\infty$ jako $c_j = \frac{1}{j!}\Phi^{(j)}(0)$, tak poté

$$\Phi(h) = \xi + c_1h + c_2h^2 + c_3h^3 + \dots,$$

což je aproximace hodnoty ξ s chybou $\mathcal{O}(h)$. Vezměme nyní h poloviční, tudíž chyba tohoto rozvoje odpovídá taktéž polovině:

$$\Phi(h/2) = \xi + c_1\frac{1}{2}h + c_2\frac{1}{4}h^2 + c_3\frac{1}{8}h^3 + \dots$$

Definujme nyní funkci Ψ jako

$$\begin{aligned} \Psi(h) &= 2\Phi(h/2) - \Phi(h) = \\ &= 2\left(\xi + c_1\frac{1}{2}h + c_2\frac{1}{4}h^2 + c_3\frac{1}{8}h^3 + \dots\right) - \left(\xi + c_1h + c_2h^2 + c_3h^3 + \dots\right) = \\ &= \xi - c_2\frac{1}{2}h^2 - c_3\frac{3}{4}h^3 + \dots \end{aligned}$$

Zajisté $\Psi(h)$ aproximuje ξ , tudíž $\Psi(0) = \xi$ s chybou řádu $\mathcal{O}(h^2)$, oproti $\Phi(h)$, které aproximuje ξ s chybou řádu $\mathcal{O}(h)$. Uvědomme si, že

$$\Psi(h/2) = \xi - c_2\frac{1}{8}h^2 - c_3\frac{3}{32}h^3 + \dots,$$

čehož můžeme využít k dalšímu zlepšení aproximace. Položme

$$\Theta(h) = \frac{4\Psi(h/2) - \Psi(h)}{3} = \xi + c_3\frac{1}{8}h^3 + \dots,$$

odkud vidíme, že $\Theta(h)$ aproximuje ξ s chybou řádu $\mathcal{O}(h^3)$. Pokud bychom tento proces stále opa-

kovali, tak přicházíme na obecný vzorec Richardsonovy extrapolace:

$$T_0(h) = \Phi(h),$$

$$T_k(h) = \frac{2^k T_{k-1}\left(\frac{h}{2}\right) - T_{k-1}(h)}{2^k - 1}, \quad k = 1, 2, 3, \dots, m.$$

Můžeme tak vytvořit následující extrapolační tabulku:

$$\begin{array}{ccccccc} T_0(h) & & & & & & \\ T_0\left(\frac{h}{2}\right) & T_1(h) & & & & & \\ T_0\left(\frac{h}{2^2}\right) & T_1\left(\frac{h}{2}\right) & T_2(h) & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ T_0\left(\frac{h}{2^m}\right) & T_1\left(\frac{h}{2^{m-1}}\right) & T_2\left(\frac{h}{2^{m-2}}\right) & \cdots & T_m(h) & & \\ \uparrow & \uparrow & \uparrow & & \uparrow & & \\ \mathcal{O}(h) & \mathcal{O}(h^2) & \mathcal{O}(h^3) & & \mathcal{O}(h^{m+1}), & & \end{array}$$

kde nejlepší aproximace v daných sloupcích jsou poslední prvky. Zajisté si všimneme, že čím víc se pohybujeme v pravé části tabulky, tím dosahujeme lepší přesnosti. Nejlepší aproximaci dostáváme v prvku $T_m(h)$.

5.1 Numerické experimenty pro Richardsonovu extrapolaci

Příklad 5.1.1. Využijme Richardsonovu extrapolaci k nalezení aproximace první derivace pro nějakou spojitě diferencovatelnou funkci f . Hledáme $\xi = f'(x_0)$, kde x_0 je vnitřním bodem definičního oboru f . Pro aproximaci první derivace použijeme metodu centrálních diferencí, kterou si odvodíme.

Pomocí Taylorovy věty víme, že

$$f\left(x + \frac{h}{2}\right) = f(x) + f'(x)\frac{h}{2} + f''(x)\frac{h^2}{8} + \mathcal{O}(h^3),$$

$$f\left(x - \frac{h}{2}\right) = f(x) - f'(x)\frac{h}{2} + f''(x)\frac{h^2}{8} - \mathcal{O}(h^3).$$

Z těchto rovnic lehce určíme aproximaci první derivace:

$$f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right) = f'(x)h + \mathcal{O}(h^3), \quad \text{tj.}$$

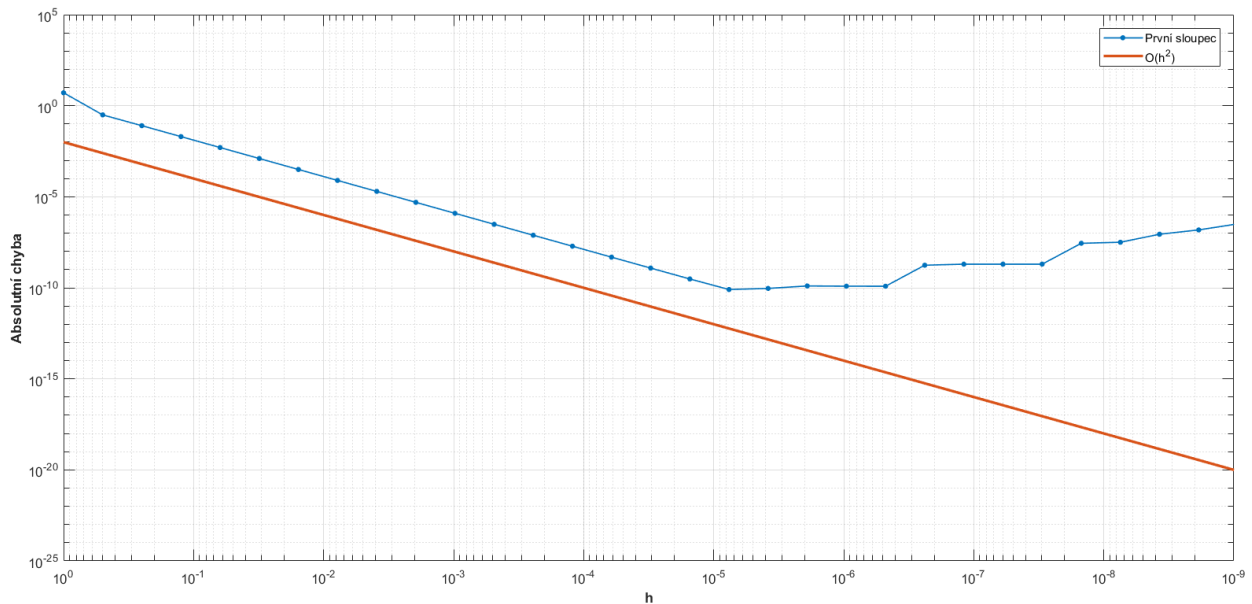
$$f'(x) = \underbrace{\frac{f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)}{h}}_{\text{tzv. centrální diference}} + \mathcal{O}(h^2).$$

Uvažujme $f(x) = \sin(\pi x)$ a zvolme $x_0 = 1$. Platí, že $f'(x_0) = \pi \cos(\pi x_0) = -\pi = -3,1415926535\dots$

Položme $m = 5$ a počáteční $h = 1$, poté dostáváme extrapolační tabulku:

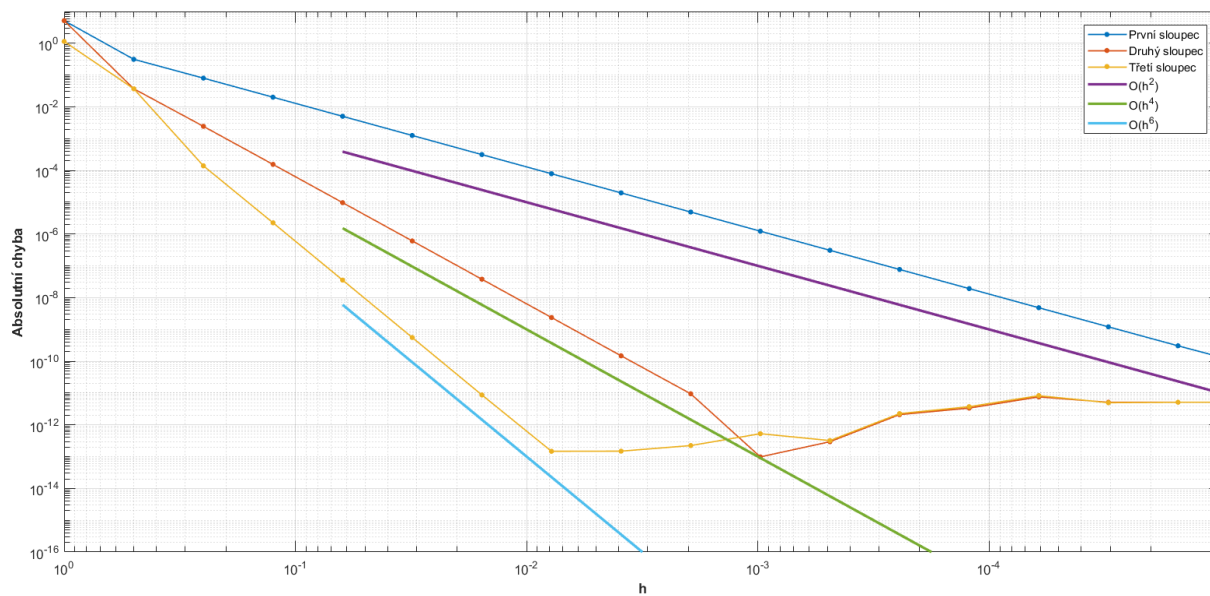
-2,0000000000				
-2,8284271247	-3,1045694996			
-3,0614674589	-3,1391475703	-3,1414527750		
-3,1214451522	-3,1414377167	-3,1415903931	-3,1415925775	
-3,1365484905	-3,1415829366	-3,1415926179	-3,1415926532	-3,1415926535.

Z této tabulky můžeme vidět, že skutečně $T_5(h) = -3,1415926535$ je velmi dobrou aproximací s absolutní chybou $9 \cdot 10^{-11}$. Otázka zní: mohli bychom dostat lepší aproximaci, pokud bychom počítali s větším m (neboli menšími h)? Zaměřme se na první sloupec tabulky, který reprezentuje aproximace pomocí centrálních diferencí pro klesající h . Na obr. 5.1 vidíme, že absolutní chyba je řádu $\mathcal{O}(h^2)$ přibližně do $h \approx 10^{-5}$, poté chyba roste z důvodu zaokrouhlovacích chyb. Můžeme tedy říct, že přesnost ztrácíme okolo $h \approx 10^{-5}$.



Obrázek 5.1: Konvergence metody centrálních diferencí

Na obr. 5.2 vidíme konvergenci ostatních sloupců, které v porovnání s prvním sloupcem konvergují rychleji, ovšem o to rychleji ztrácí svou přesnost.



Obrázek 5.2: Konvergence Richardsonovy extrapolace I

△

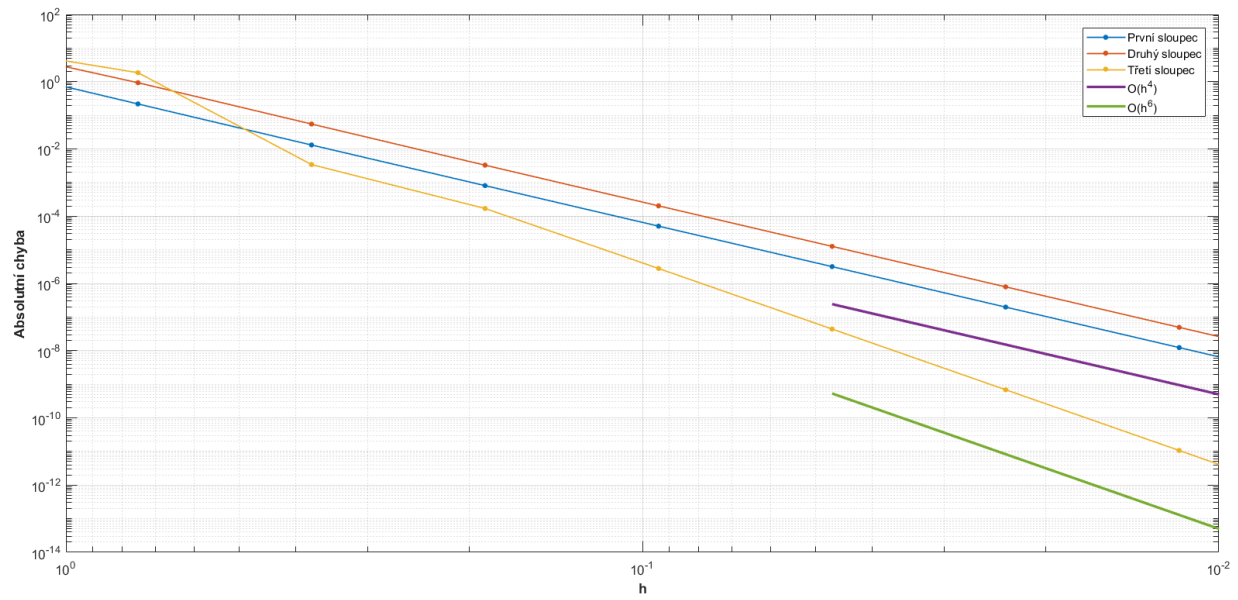
Příklad 5.1.2. Nyní se podíváme, jak můžeme pomocí Richardsonovy extrapolace zpřesnit složené Simpsonovo pravidlo. Aproximujme nyní

$$\int_0^1 \frac{1}{20} x^4 \cos(3\pi x) dx.$$

Analytickou hodnotu určíme pomocí matematického software Maple: $\frac{1}{20}(888 \sin(6) + 720 \cos(6))$, což je přibližně 22,16008219938087... Zvolme $m = 5$ a počáteční $h = 1$. První sloupec vypočteme díky rekurzivní metodě, poté výsledky aproximací vypadají následovně:

46,1811561300				
18,4919131342	9,2621654689			
21,9425015100	23,0926976353	24,0147331130		
22,1469923186	22,2151559215	22,1566531406	22,1271598077	
22,1592724181	22,1633657846	22,1599131088	22,1599648543	22,1600935016.

Na obr. 5.3 můžeme vidět, že první dva sloupce se řídí trendem $\mathcal{O}(h^4)$ a následující $\mathcal{O}(h^6)$.



Obrázek 5.3: Konvergence Richardsonovy extrapolace II

△

Kapitola 6

Rombergova metoda

Richardsonovu extrapolaci můžeme použít pro libovolné dělení h , ale zdvojnásobením počtů uzlů vede na použití rekurzivní lichoběžníkové metody s dělením na poloviny, kde efektivně využijeme již dříve získaných výpočtů, jak si ukážeme na **Rombergově metodě** (viz také [7, 8]).

Úkolem této metody je vytvoření tabulky R ,¹ kde každý prvek této tabulky je aproximací integrálu

$$\int_a^b f(x) dx.$$

Tyto prvky dostaneme pomocí kombinací rekurzivní lichoběžníkové metody s dělením na poloviny a Richardsonovy extrapolace. Uvažujme $h = b - a$, položme

$$R(0, 0) = L_0(f, h)$$

a následující členy prvního sloupce získáme díky rekurzi:

$$\begin{aligned} R(1, 0) &= L_0\left(f, \frac{h}{2}\right), \\ R(2, 0) &= L_0\left(f, \frac{h}{2^2}\right), \\ &\vdots \\ R(m, 0) &= L_0\left(f, \frac{h}{2^m}\right). \end{aligned}$$

Nyní si můžeme zvolit jeden z postupů, jak budeme metodu orientovat. Buďto se budeme snažit výsledek určit výpočtem jednotlivých řádků, nebo, tak jak si předvedeme, získáme výsledek po sloupcích. Využitím Richardsonovy extrapolace postupně vypočteme sloupce $R(:, k)$, $k = 1, 2, 3, \dots, m$.

¹Symbolem $R(i, j)$ budeme značit prvek v i -tém řádku a v j -tém sloupci tabulky R . Symbolem $R(:, j)$ budeme značit j -tý sloupec tabulky R .

Předpokládejme, že sloupce $R(:, k - 1)$ jsme již určili, a tedy:

$$R(j, k) = \frac{4^k R(j, k - 1) - R(j - 1, k - 1)}{4^k - 1}, \quad j, k = 1, 2, 3, \dots, m.$$

Extrapoláčnická tabulka R z Rombergovy metody vypadá následovně:

$$\begin{array}{ccccccc} L_0(f, h) & & & & & & \\ L_0\left(f, \frac{h}{2}\right) & L_1(f, h) & & & & & \\ L_0\left(f, \frac{h}{2^2}\right) & L_1\left(f, \frac{h}{2}\right) & L_2(f, h) & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ L_0\left(f, \frac{h}{2^m}\right) & L_1\left(f, \frac{h}{2^{m-1}}\right) & L_2\left(f, \frac{h}{2^{m-2}}\right) & \cdots & L_m(f, h), & & \end{array}$$

přičemž v posledním řádku se nacházejí nejlepší aproximace z k -tých sloupců, které mají přesnost $\mathcal{O}(h^{2k+2})$, a nejlepší aproximaci nám dává $L_m(f, h)$ (resp. $R(m, m)$).

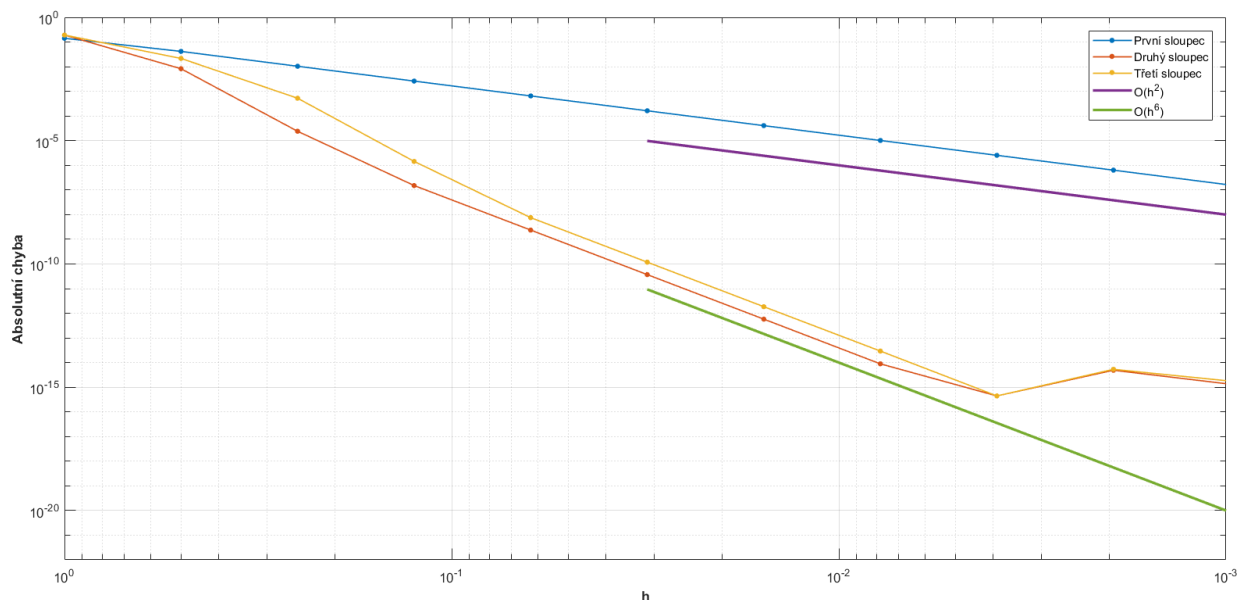
6.1 Numerické experimenty pro Rombergovu metodu

Příklad 6.1.1. *Zkusme nyní využít tento algoritmus pro integrál z Rungeovy funkce. Aproximujme*

$$\int_{-1}^1 \frac{2}{1+x^2} dx,$$

viz příklad 2.5.2 na str. 20. Po zvolení $m = 5$ dostáváme následující výsledky Rombergova algoritmu:

2,0000000000				
3,0000000000	3,3333333334			
3,1000000000	3,1333333334	3,1200000000		
3,1311764706	3,1415686275	3,1421176471	3,1424687208	
3,1389884945	3,1415925025	3,1415940941	3,1415857837	3,1415823213.



Obrázek 6.1: Konvergence Rombergovy metody I

△

Příklad 6.1.2. *Pokusme se nyní aproximovat integrál*

$$\int_0^1 x^{\frac{1}{5}} dx.$$

Jelikož se jedná o tabulkový integrál, tak velmi lehce zjistíme jeho analytickou hodnotu:

$$\int_0^1 x^{\frac{1}{5}} dx = \left[\frac{5x^{\frac{6}{5}}}{6} \right]_0^1 = \frac{5}{6}.$$

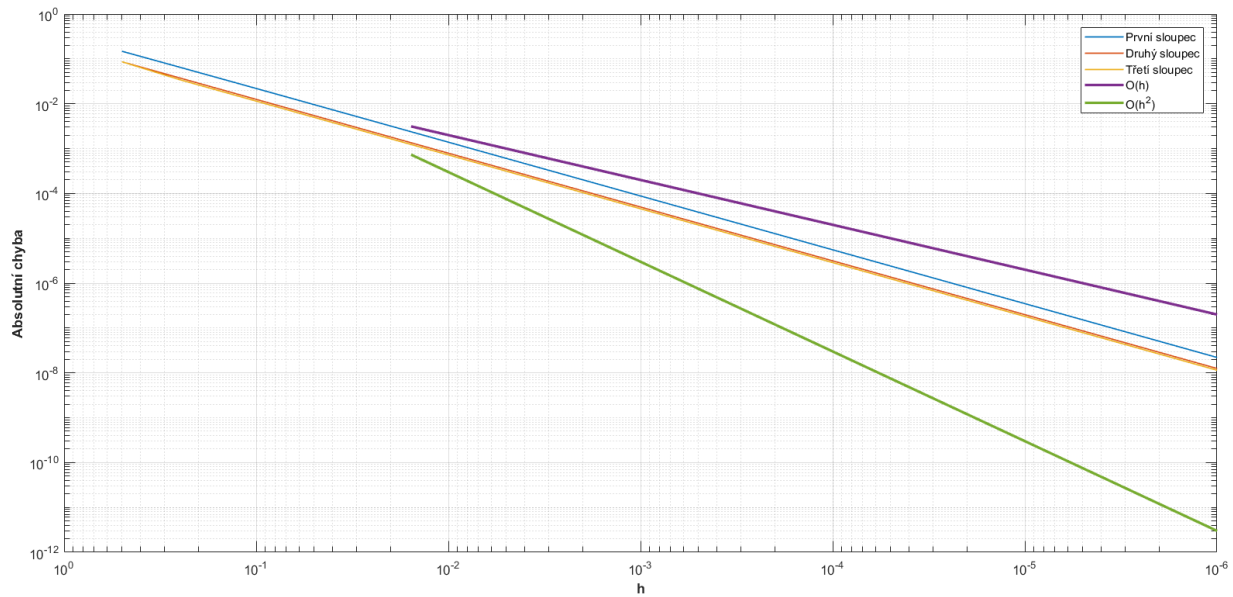
Všimněme si, že funkce $f(x) = x^{\frac{1}{5}}$ není v 0 diferencovatelná, protože funkce má v tomto bodě nevlastní derivaci. Skutečně:

$$\left(x^{\frac{1}{5}}\right)' = \frac{1}{5x^{\frac{4}{5}}}, \quad x \neq 0 \quad \Rightarrow \quad f'(0) = \lim_{x \rightarrow 0^-} \frac{1}{5x^{\frac{4}{5}}} = \lim_{x \rightarrow 0^+} \frac{1}{5x^{\frac{4}{5}}} = \infty.$$

Tudíž nejsou splněny předpoklady pro Eulerovu-Maclaurinovu expanzi (viz str. 39), což může mít za následek chyby extrapolace. Když se podíváme na výsledky z následující tabulky:

0,5000000000				
0,6852752816	0,7470337089			
0,7681240895	0,7957403587	0,7989874687		
0,8047569388	0,8169678885	0,8183830572	0,8186909237	
0,8208465226	0,8262097172	0,8268258391	0,8269598516	0,8269922787.

a graf na obr. 6.2, který znázorňuje konvergenci sloupců, tak vidíme, že Rombergova metoda konverguje pro tento integrál velmi pomalu.



Obrázek 6.2: Konvergence Rombergovy metody II

△

Další problém by nastal například při numerickém řešení nevlastního integrálu

$$\int_0^1 \frac{1}{\sqrt{x}} dx,$$

kde ač-li hodnoty funkce, když se x blíží k nule, jdou do nekonečna, tak výsledek je konečný:

$$\int_0^1 \frac{1}{\sqrt{x}} dx = \lim_{t \rightarrow 0^+} \int_t^1 \frac{1}{\sqrt{x}} dx = \lim_{t \rightarrow 0^+} [2\sqrt{x}]_t^1 = \lim_{t \rightarrow 0^+} (2 - 2\sqrt{t}) = 2. \quad (6.1.1)$$

Námi doposud zmíněné metody (založené na uzavřených formulích) nedokáží efektivně řešit integrály tohoto druhu. Pokud by tyto algoritmy pracovaly jen s vnitřními body intervalu, tak bychom tento problém vyřešili. V kapitole 8 se s tímto postupem seznámíme, ovšem pojďme se ještě předtím podívat na metodu, která nám dá velmi dobré výsledky za použití chytré myšlenky.

Kapitola 7

Adaptivní Simpsonovo pravidlo

Zamysleme se nad složeným Simpsonovým pravidlem. Uvědomíme si, že jestliže máme ekvidistantní síť uzlů, tak lokální chyby na subintervalech se budou s největší pravděpodobností lišit. Víme, že lokální chyba i -tého subintervalu je

$$E_{2_i}(f, a, b) = -\frac{h^5}{90} f^{(4)}(\xi_i), \quad \xi_i \in (x_{i-1}, x_i).$$

Můžeme tedy volit subintervaly různé délky takové, že pokud se derivace chová na daném dílku „divoce“, tak je třeba jej rozdělit na menší části.

Volme h proměnlivé tak, abychom mohli rovnoměrně rozložit chybu přes všechny subintervaly. Toho docílíme tím, že tam, kde f' nemění signifikantně své hodnoty, budeme volit větší subintervaly a tam, kde f' velmi ostře mění své hodnoty, budeme volit subintervaly menší. Tomuto procesu říkáme **adaptivní Simpsonovo pravidlo** (viz také [7]).

Uvažujme interval $\langle a, b \rangle$ a $h = \frac{b-a}{2}$, poté

$$S_1(f, a, b) = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

a

$$E_1(f, a, b) = -\frac{h^5}{90} f^{(4)}(\xi), \quad \xi \in (a, b).$$

Výpočet S_1 není problém, ovšem určení chyby není jednoduché, protože netušíme, jaké ξ máme zvolit. Nyní rozložíme $\langle a, b \rangle$ na $\langle a, c \rangle$ a $\langle c, b \rangle$, kde $c = \frac{a+b}{2}$. Poté

$$S_2(f, a, b) = S_1(f, a, c) + S_1(f, c, b)$$

a

$$\begin{aligned} E_2(f, a, b) &= E_1(f, a, c) + E_1(f, c, b) = \\ &= -\frac{1}{90} \left(\frac{h}{2}\right)^5 f^{(4)}(\xi_1) - \frac{1}{90} \left(\frac{h}{2}\right)^5 f^{(4)}(\xi_2), \quad \xi_1 \in (a, c), \quad \xi_2 \in (c, b). \end{aligned}$$

Předpokládejme, že $f^{(4)} \approx \text{konst.}$ na (a, b) , poté¹

$$E_2 = -\frac{2}{90} \left(\frac{h}{2}\right)^5 f^{(4)}(\xi) = -\frac{h^5}{90 \cdot 2^4} f^{(4)}(\xi) = \frac{1}{2^4} E_1 \quad \Rightarrow \quad E_1 = 16E_2, \quad \xi \in (a, b).$$

Víme, že

$$\int_a^b f(x) dx \approx S_1 + E_1 = S_2 + E_2 \quad \Rightarrow \quad S_1 + 16E_2 = S_2 + E_2 \quad \Rightarrow$$

z čehož vyplývá

$$E_2 = \frac{S_2 - S_1}{15}. \tag{7.0.1}$$

Tudíž lepší aproximaci získáme pomocí

$$A(f, a, b) \stackrel{\text{ozn.}}{=} S_2 + \frac{S_2 - S_1}{15},$$

což je předpis pro adaptivní Simpsonovo pravidlo. Vraťme se k rovnosti (7.0.1). Zajisté jí můžeme přepsat na tvar

$$S_2 - S_1 = 15E_2.$$

Uvažujme, že bychom rádi toto pravidlo naprogramovali rekurzivně, tudíž musíme mít ukončovací podmínku, kterou si zvolíme jako

$$|S_1 - S_2| < 15\varepsilon,$$

kde ε je námi libovolné (ale pevně určené) kladné číslo, které nazveme tolerancí. Rekurzivní metoda bude vypadat tudíž následovně: v další iteraci zvolíme půlku původního intervalu a snížíme toleranci o polovinu. S tímto dělením postupujeme, dokud nesplníme všechny potřebné ukončovací podmínky. Podrobnější vysvětlení najdeme v následujícím pseudokódu:

¹Od teď nebudeme zdůrazňovat závislost E_i ani S_i na f, a a b .

Pseudokód 1 adaptivní Simpsonovo pravidlo

function výsledek = Simpson(f , a , b , ε)vypočteme S_1 , S_2 **if** $|S_2 - S_1| < 15\varepsilon$ výsledek = $S_2 + (S_2 - S_1)/15$ **else** $c = (b + a)/2$ $L = \text{Simpson}(f, a, c, \varepsilon/2)$ $R = \text{Simpson}(f, c, b, \varepsilon/2)$ **end if**

Pro ilustraci výše uvedených myšlenek jsme použili Simpsonovo pravidlo, ale analogicky lze vytvořit také adaptivní obdélníkové a lichoběžníkové pravidlo.

7.1 Numerické experimenty pro adaptivní Simpsonovo pravidlo

Příklad 7.1.1. Pomocí adaptivního Simpsonova pravidla aproximujme integrál

$$\int_0^2 x^{10} e^{4x^3 - 3x^4} dx.$$

K vypočtení jeho analytické hodnoty využijeme matematický software Maple, který udává výsledek 7,258395170614293. V následující tabulce vidíme, jak přesně algoritmus approximoval hodnotu integrálu se zadanou tolerancí:

tolerance	aproximace	absolutní chyba
10^{-3}	7,258376114514226	$1,9 \cdot 10^{-5}$
10^{-4}	7,258399589492167	$4,4 \cdot 10^{-6}$
10^{-5}	7,258395395788935	$2,6 \cdot 10^{-6}$
10^{-6}	7,258395178137319	$2,3 \cdot 10^{-7}$
10^{-7}	7,258395173052513	$7,5 \cdot 10^{-9}$
10^{-8}	7,258395172479220	$1,9 \cdot 10^{-9}$

Tabulka 7.1: Aproximace pomocí adaptivního Simpsonova pravidla

△

Kapitola 8

Gaussova kvadratura

Shrneme si základy, na kterých jsme naše kvadratury doposud tvořili. Připomeňme Newtonovu-Cotesovu kvadraturní formuli řádu n , která je ve tvaru:

$$\int_a^b f(x) dx \approx \sum_{k=0}^n w_k f(x_k).$$

Integrační body $x_0, x_1, x_2, \dots, x_n$ jsme pevně volili jako ekvidistantní a integrační váhy jsme volili jako

$$w_k = \begin{cases} \int_a^b 1 dx = b - a, & n = 0, \\ \int_a^b \ell_k(x) dx, & n \geq 1, \end{cases} \quad k = 0, 1, 2, \dots, n,$$

kde ℓ_k je k -tou kardinální funkcí Lagrangeova polynomu (viz (2.0.1)). Není těžké si uvědomit, že takovýmto způsobem jsme zařídili, že

$$\int_a^b p(x) dx = \sum_{k=0}^n w_k p(x_k), \quad p \in \mathcal{P}_n(\langle a, b \rangle). \quad (8.0.1)$$

Připomeňme, jak vypadají integrační body a váhy k dříve uvedených metodám v tab. 8.1.

Zkusme nyní přijít na způsob, kterým bychom nevolili kvadraturní váhy pomocí Lagrangeova interpolačního polynomu (viz také [3, 7]). Pro kvadraturu

$$\int_{-1}^1 f(x) dx \approx A_0 f(x_0) + A_1 f(x_1) + A_2 f(x_2) + \dots + A_n f(x_n),$$

kde $A_0, A_1, A_2, \dots, A_n$ jsou **obecné kvadraturní váhy**¹ (dále již jen kvadraturní váhy) a body $x_0, x_1, x_2, \dots, x_n$, které jsou voleny náhodně, chceme, aby dávala přesný výsledek pro všechny $p \in$

¹Všimněme si, že při tomto zavedení jsou kvadraturní váhy použité u Newtonových-Cotesových formulí speciálním případem obecných vah.

pravidlo	n	body	váhy
obdélníkové	0	$x_0 = (b + a)/2$	$w_0 = b - a$
lichoběžníkové	1	$x_0 = a$ $x_1 = b$	$w_0 = w_1 = (b - a)/2$
Simpsonovo	2	$x_0 = a$ $x_1 = (b + a)/2$ $x_2 = b$	$w_0 = w_2 = (b - a)/6$ $w_1 = 2(b - a)/3$

Tabulka 8.1: Kvadraturní body a váhy pro Newtonovy-Cotesovy metody řádu $n \leq 2$

$\mathcal{P}_n(\langle -1, 1 \rangle)$. Uvažujme nyní $n = 3$:

$$\int_{-1}^1 p(x) dx = A_0 f(x_0) + A_1 f(x_1) + A_2 f(x_2) + A_3 f(x_3). \quad (8.0.2)$$

Jelikož pro 4 uzly chceme přesně integrovat polynomy až do třetího řádu včetně, tak musí platit, že pro $p(x) = 1, x, x^2, x^3$ nastává rovnost (8.0.2). Budeme potřebovat hodnoty

$$\begin{aligned} \int_{-1}^1 1 dx &= 2, & \int_{-1}^1 x dx &= 0, \\ \int_{-1}^1 x^2 dx &= \frac{2}{3}, & \int_{-1}^1 x^3 dx &= 0. \end{aligned} \quad (8.0.3)$$

Vzniká nám systém n lineárních rovnic s n neznámými integračními váhami, který můžeme vyjádřit pomocí tzv. Vandermondeovy matice:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_0 & x_1 & x_2 & x_3 \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 \\ x_0^3 & x_1^3 & x_2^3 & x_3^3 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ \frac{2}{3} \\ 0 \end{bmatrix}.$$

Tato matice je ovšem špatně podmíněná.¹ Ukažme si nyní způsob, při kterém budeme integrační body volit takovým způsobem, kterým bychom potřebovali pouze n vah a n integračních bodů pro přesnou integraci všech polynomů z prostoru $\mathcal{P}_{2n-1}(\langle -1, 1 \rangle)$. Díky tomuto nebudeme muset řešit velkou soustavu rovnic a zároveň budeme mít výrazně lepší přesnost.

¹Pokud se nám trošku změni pravá strana, tak se může signifikantně změnit řešení této soustavy pro velká n .

8.1 Gaussova-Legendreova kvadratura

Zabývejme se nyní **Legendreovými polynomy n -tého řádu** s označením \mathcal{L}_n . Víme, že tyto polynomy můžeme získat díky Gramovu-Schmidtovu ortogonalizačnímu procesu z báze $\{1, x, x^2, x^3, \dots, x^n\}$ prostoru $\mathcal{P}_n(\langle -1, 1 \rangle)$, vzhledem ke skalárnímu součinu

$$\langle u, v \rangle = \int_{-1}^1 u(x)v(x) dx. \quad (8.1.1)$$

Pro připomenutí si ukážeme první tři Legendreovy polynomy:

$$\mathcal{L}_0 = 1, \quad \mathcal{L}_1 = x - \frac{\langle \mathcal{L}_0, x \rangle}{\langle \mathcal{L}_0, \mathcal{L}_0 \rangle} \mathcal{L}_0 = x, \quad \mathcal{L}_2 = x^2 - \frac{\langle \mathcal{L}_0, x^2 \rangle}{\langle \mathcal{L}_0, \mathcal{L}_0 \rangle} \mathcal{L}_0 - \frac{\langle \mathcal{L}_1, x^2 \rangle}{\langle \mathcal{L}_1, \mathcal{L}_1 \rangle} \mathcal{L}_1 = x^2 - \frac{1}{3}.$$

Tyto Legendreovy polynomy ještě škálujeme tak, aby $\forall n \in \mathbb{N}_0 : \mathcal{L}_n(1) = 1$. Uvedme alespoň prvních pět stupňů škálovaných polynomů:

$$\begin{aligned} \mathcal{L}_0(x) &= 1, \\ \mathcal{L}_1(x) &= x, \\ \mathcal{L}_2(x) &= \frac{1}{2}(3x^2 - 1), \\ \mathcal{L}_3(x) &= \frac{1}{2}(5x^3 - 3x), \\ \mathcal{L}_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3). \end{aligned}$$

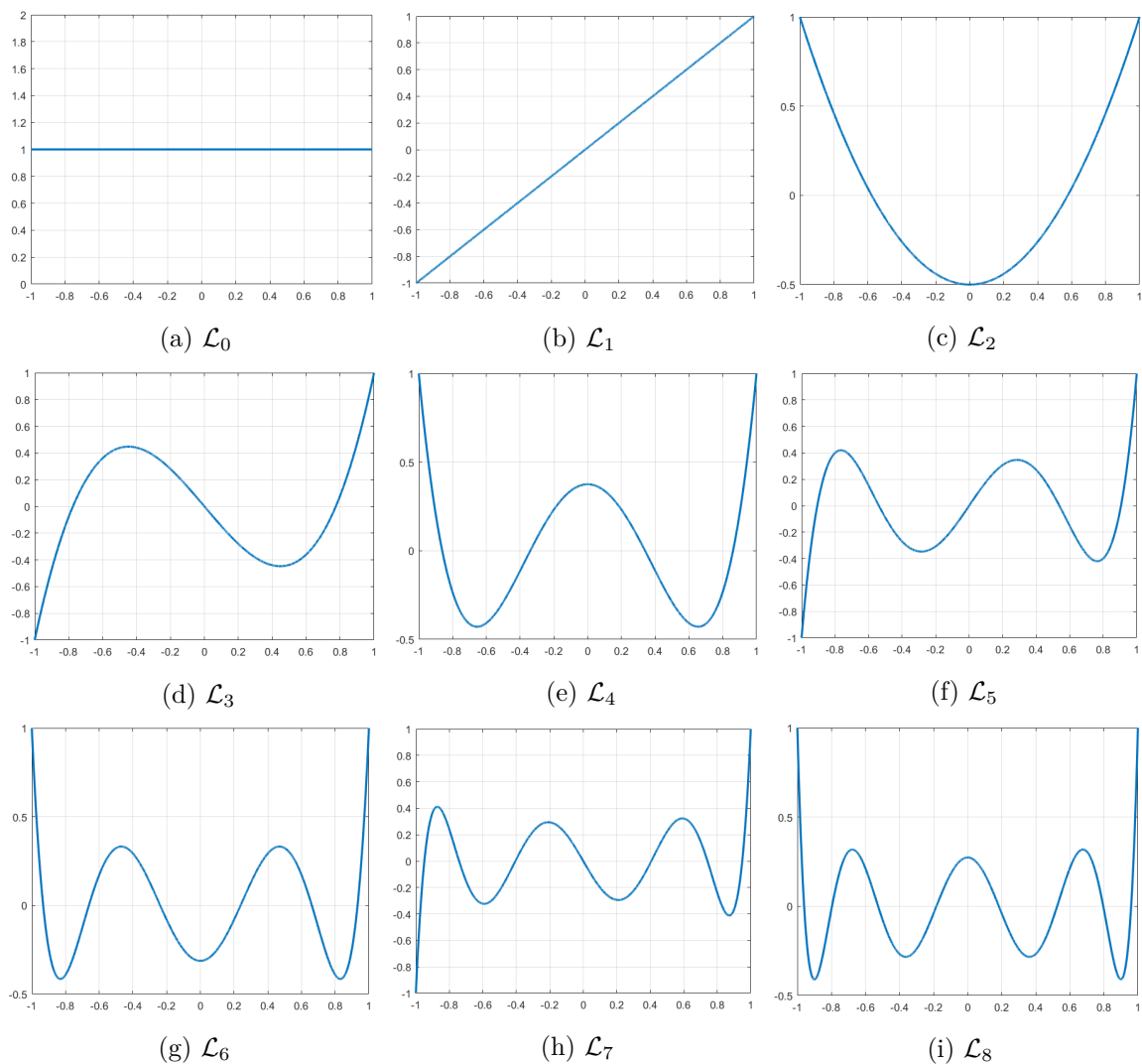
Grafické znázornění těchto polynomů vidíme na obr. 8.1.

Jak popisují [3, 7], tak si všimněme, že takto sestrojený prvek \mathcal{L}_n je nejen ortogonální ke všem předešlým prvkům $\mathcal{L}_0, \mathcal{L}_1, \dots, \mathcal{L}_{n-1}$, ba dokonce i ke všem polynomům, které leží v $\mathcal{P}_{n-1}(\langle -1, 1 \rangle)$, protože Legendreovy polynomy $\mathcal{L}_0, \mathcal{L}_1, \dots, \mathcal{L}_{n-1}$ taktéž tvoří bázi tohoto prostoru. Lze dále ukázat, že \mathcal{L}_n sudého stupně jsou sudé funkce a \mathcal{L}_n lichého stupně jsou liché funkce. Dále má Legendreův polynom n -tého stupně, $n \geq 1$, n různých reálných kořenů ležících v intervalu $(-1, 1)$, které se shlukují ke krajním bodům -1 a 1 a jsou rozloženy symetricky vzhledem k bodu 0 . Těchto vlastností využijeme při tvorbě **Gaussovy-Legendreovy kvadraturní formule**.

Uvažujme nyní $p \in \mathcal{P}_{2n-1}(\langle -1, 1 \rangle)$, $n \in \mathbb{N}$. Pak víme, že pokud vydělíme tento polynom polynommem \mathcal{L}_n , tak dostáváme

$$\frac{p(x)}{\mathcal{L}_n(x)} = q(x) + \frac{r(x)}{\mathcal{L}_n(x)} \quad \Rightarrow \quad p(x) = q(x)\mathcal{L}_n(x) + r(x),$$

kde q a r jsou polynomy z $\mathcal{P}_{n-1}(\langle -1, 1 \rangle)$. Zintegrujeme-li obě strany pravé rovnosti na intervalu



Obrázek 8.1: Legendreovy polynomy nultého až osmého řádu

$\langle -1, 1 \rangle$, získáváme

$$\int_{-1}^1 p(x) dx = \int_{-1}^1 q(x) \mathcal{L}_n(x) dx + \int_{-1}^1 r(x) dx.$$

Všimněme si, že

$$\int_{-1}^1 q(x) \mathcal{L}_n(x) dx = 0,$$

protože víme, že \mathcal{L}_n je ortogonální vzhledem ke skalárnímu součinu (8.1.1) k polynomům nižšího stupně. Tudíž

$$\int_{-1}^1 p(x) dx = \int_{-1}^1 r(x) dx.$$

Zvolme integrační body jako kořeny \mathcal{L}_n (kořenů je n a všechny jsou v intervalu $(-1, 1)$): $-1 < x_0 <$

$x_1 < x_2 < \dots < x_{n-1} < 1$. S takto zvolenými integračními body nám nyní zbývá již jen určit kvadrurní váhy, kterých je n (připomeňme, že $r \in \mathcal{P}_{n-1}(\langle -1, 1 \rangle)$).

$$\int_{-1}^1 p(x) \, dx = \int_{-1}^1 r(x) \, dx = A_0 r(x_0) + A_1 r(x_1) + \dots + A_{n-1} r(x_{n-1}),$$

které určíme pomocí systému n lineárních rovnic (stejně jako na str. 55):

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_0 & x_1 & x_2 & \dots & x_{n-1} \\ x_0^2 & x_1^2 & x_2^2 & \dots & x_{n-1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_0^{n-1} & x_1^{n-1} & x_2^{n-1} & \dots & x_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_{n-1} \end{bmatrix} = \begin{bmatrix} \int_{-1}^1 1 \, dx \\ \int_{-1}^1 x \, dx \\ \int_{-1}^1 x^2 \, dx \\ \vdots \\ \int_{-1}^1 x^{n-1} \, dx \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ \frac{2}{3} \\ \vdots \\ \frac{1}{n}(1 - (-1)^n) \end{bmatrix}.$$

Krása této myšlenky je taková, že exaktní hodnotu integrálu z polynomu stupně $2n - 1$ na $\langle -1, 1 \rangle$ jsme schopni zajistit pomocí n integračních bodů a vah, čímž jsme signifikantně zvětšili efektivitu kvadratury. Celý proces shrneme v následujícím pseudokódu:¹

Pseudokód 2 Gaussova-Legendreova kvadratura

function výsledek = Gaussova-Legendreova kvadratura(f, a, b, n)

- určíme n kořenů Legendreova polynomu n -tého stupně
 - sestavíme Vandermondeovu matici $V \in \mathbb{R}^{n \times n}$
 - sestavíme vektor $c \in \mathbb{R}^n$ obsahující na i -té pozici $\int_{-1}^1 x^i \, dx$
 - n integračních vah ve vektoru $A \in \mathbb{R}^n$ určíme řešením soustavy rovnic $VA = c$
 - transformuje kořeny a integrační váhy pro obecný interval $\langle a, b \rangle$
 - výsledek = $\sum_{i=0}^{n-1} A(i) f(x_i)$
-

V pseudokódu hovoříme o transformaci integračních bodů a kořenů pro obecný interval, o čemž se zmíníme za okamžik. Vyřešme nyní příklad 8.0.2 z motivace této kapitoly. Nalezneme Gaussovu-Legendreovu kvadraturu, která bude počítat exaktně integrály polynomů $p \in \mathcal{P}_3(\langle -1, 1 \rangle)$. Jelikož $3 = 2 \cdot 2 - 1$, je $n = 2$. Píšeme tedy

$$\int_{-1}^1 p(x) \, dx = A_0 p(x_0) + A_1 p(x_1).$$

Kořeny Legendreova polynomu stupně 2 jsou (viz tab. 8.2)

$$x_{0,1} = \pm \frac{1}{\sqrt{3}}.$$

¹Vektory v pseudokódu indexujeme od 0.

Připomeňme rovnosti (8.0.3). Řešíme tudíž soustavu rovnic

$$\begin{aligned} A_0 + A_1 &= 2, \\ A_0 \frac{1}{\sqrt{3}} + A_1 \left(-\frac{1}{\sqrt{3}}\right) &= 0, \end{aligned}$$

jejichž řešení je $A_0 = A_1 = 1$. Řešením tohoto příkladu je tudíž

$$\int_{-1}^1 p(x) \, dx = p\left(\frac{1}{\sqrt{3}}\right) + p\left(-\frac{1}{\sqrt{3}}\right).$$

Tímto způsobem bychom určili Gaussovu-Legendreovu kvadraturu pro jakékoliv n . V následující tabulce vidíme integrační body a váhy do pátého stupně. Je nutno podotknout, že rozdělení integračních vah je symetrické, což plyne ze symetrie rozložení integračních bodů, které jsou symetrické vzhledem k 0.

n	integrační body	integrační váhy
1	0	2
2	$\pm 1/\sqrt{3}$	1
3	$\pm\sqrt{0,6}$	5/9
	0	8/9
4	$\pm\sqrt{(3 - \sqrt{4,8})/7}$	$(18 + \sqrt{30})/36$
	$\pm\sqrt{(3 + \sqrt{4,8})/7}$	$(18 + \sqrt{30})/36$
5	$\pm\sqrt{(5 - \sqrt{40/7})/9}$	$(322 + 13\sqrt{70})/900$
	0	128/255
	$\pm\sqrt{(5 + \sqrt{40/7})/9}$	$(322 - 13\sqrt{70})/900$

Tabulka 8.2: Integrační body a váhy Gaussovy-Legendreovy kvadratury na intervalu $\langle -1, 1 \rangle$

Takto zvolené integrační body a váhy jsou ovšem pouze na intervalu $\langle -1, 1 \rangle$. Uvažujme $t \in \langle -1, 1 \rangle$, poté pro převedení do obecného intervalu $\langle a, b \rangle$ využijeme následující transformaci:

$$\int_a^b f(x) \, dx = \left. \begin{array}{l} \text{substituce} \\ x = \frac{b-a}{2}t + \frac{b+a}{2} \\ dx = \frac{b-a}{2}dt \\ x = a \Rightarrow t = -1 \\ x = b \Rightarrow t = 1 \end{array} \right| = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{1}{2}(b-a)t + \frac{1}{2}(b+a)\right) dt.$$

Takže transformované integrační body na $\langle a, b \rangle$ jsou:

$$x_i^{\langle a, b \rangle} = \frac{1}{2}(b-a)x_i^{\langle -1, 1 \rangle} + \frac{1}{2}(b+a), \quad i = 0, 1, \dots, n-1,$$

a transformované integrační váhy na $\langle a, b \rangle$ jsou:

$$A_i^{\langle a, b \rangle} = \frac{b-a}{2} A_i^{\langle -1, 1 \rangle}, \quad i = 0, 1, \dots, n-1.$$

8.2 Gaussova-Čebyševova kvadratura

Ač-li je Gaussova-Legendreova kvadratura nejčastěji používaná z Gaussových kvadratur, tak si zde předvedeme další z možných způsobů, jak k ní můžeme přistupovat. Zabývejme se nyní **Čebyševovými polynomy n -tého řádu** s označením \mathcal{C}_n . Tyto polynomy můžeme taktéž získat díky Gramovu-Schmidtovu ortogonalizačnímu procesu z báze $\{1, x, x^2, x^3, \dots, x^n\}$ prostoru $\mathcal{P}_n(\langle -1, 1 \rangle)$, vzhledem ke skalárnímu součinu

$$\langle u, v \rangle = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} u(x)v(x) dx.$$

Vytvořme si takto alespoň první tři Čebyševovy polynomy:

$$\mathcal{C}_0 = 1, \quad \mathcal{C}_1 = x - \frac{\langle \mathcal{C}_0, x \rangle}{\langle \mathcal{C}_0, \mathcal{C}_0 \rangle} \mathcal{C}_0 = x, \quad \mathcal{C}_2 = x^2 - \frac{\langle \mathcal{C}_0, x^2 \rangle}{\langle \mathcal{C}_0, \mathcal{C}_0 \rangle} \mathcal{C}_0 - \frac{\langle \mathcal{C}_1, x^2 \rangle}{\langle \mathcal{C}_1, \mathcal{C}_1 \rangle} \mathcal{C}_1 = x^2 - \frac{1}{2}.$$

Tyto Čebyševovy polynomy ještě škálujeme tak, aby $\forall n : \mathcal{C}_n(1) = 1$. Uvedme alespoň prvních pět stupňů škálovaných polynomů:

$$\begin{aligned} \mathcal{C}_0(x) &= 1, \\ \mathcal{C}_1(x) &= x, \\ \mathcal{C}_2(x) &= 2x^2 - 1, \\ \mathcal{C}_3(x) &= 4x^3 - 3x, \\ \mathcal{C}_4(x) &= 8x^4 - 8x^2 + 1. \end{aligned}$$

Grafické znázornění těchto polynomů vidíme na obr. 8.2. Pokud zvolíme n integračních bodů jako n kořenů Čebyševova polynomu n -tého řádu, tak obdobným způsobem, jako jsme ukázali u odvození Gaussovy-Legendreovy kvadratury, bychom zjistili, že takto zvolené integrační body vedou na řešení

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} r(x) dx = A_0 r(x_0) + A_1 r(x_1) + A_2 r(x_2) + \dots + A_{n-1} r(x_{n-1}),$$

kde $r \in \mathcal{P}_{n-1}(\langle -1, 1 \rangle)$. Soustava n lineárních rovnic, ve které určujeme n integračních vah, vypadá následovně:

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_0 & x_1 & x_2 & \dots & x_{n-1} \\ x_0^2 & x_1^2 & x_2^2 & \dots & x_{n-1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_0^{n-1} & x_1^{n-1} & x_2^{n-1} & \dots & x_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_{n-1} \end{bmatrix} = \begin{bmatrix} \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx \\ \int_{-1}^1 \frac{x}{\sqrt{1-x^2}} dx \\ \int_{-1}^1 \frac{x^2}{\sqrt{1-x^2}} dx \\ \vdots \\ \int_{-1}^1 \frac{x^{n-1}}{\sqrt{1-x^2}} dx \end{bmatrix} = \begin{bmatrix} \pi \\ 0 \\ \frac{\pi}{2} \\ \vdots \\ \vdots \end{bmatrix}.$$

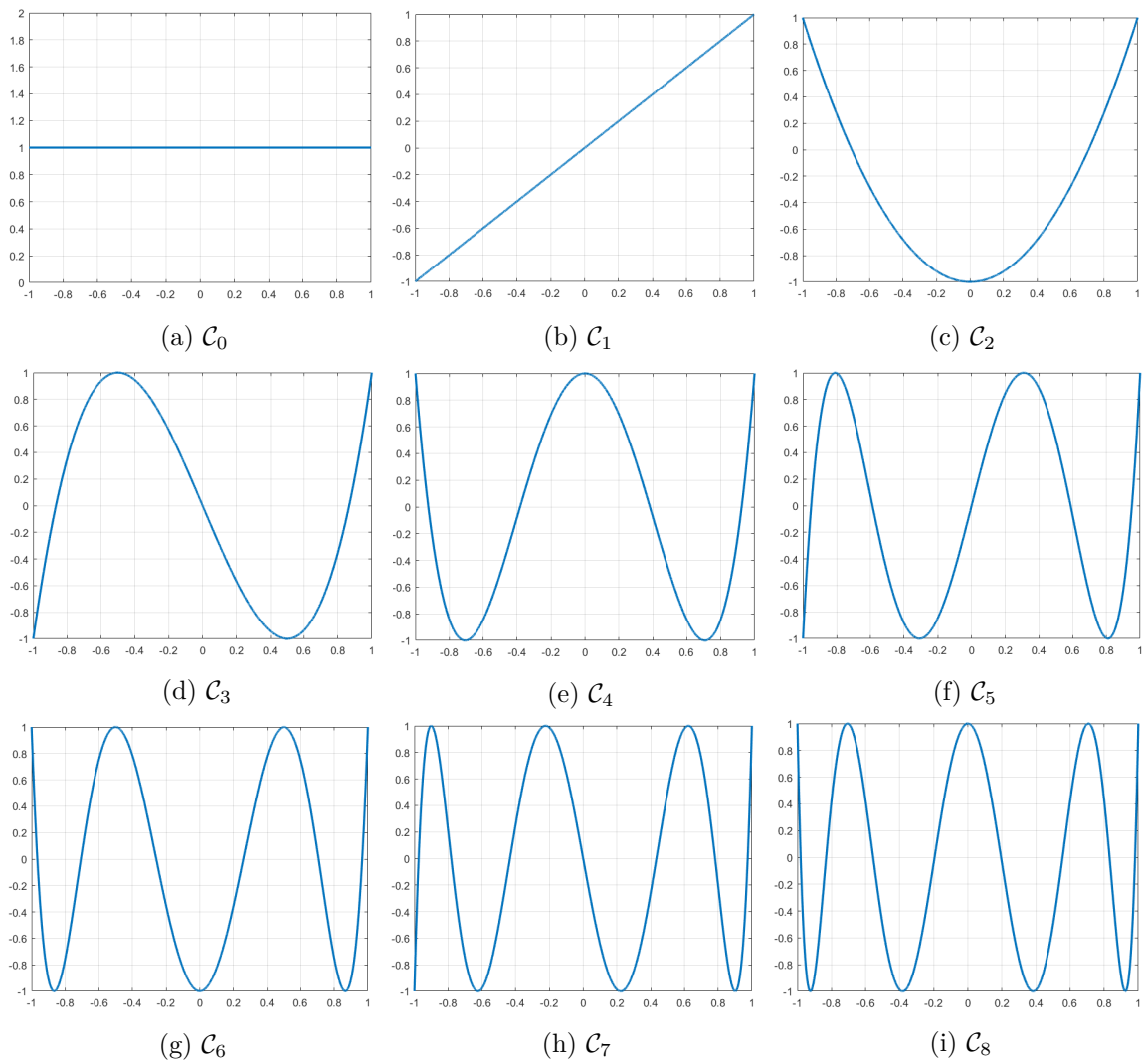
Ukažme si alespoň několik prvních integračních vah a bodů, které v této kvadratuře využíváme:¹

n	integrační body	integrační váhy
2	$\pm\sqrt{2}/2$	$\pi/2$
3	$\pm\sqrt{3}/2,$ 0	$\pi/3$
4	$\pm\sqrt{2+\sqrt{2}}/2,$ $\pm\sqrt{2-\sqrt{2}}/2$	$\pi/4$
5	$\pm\sqrt{(5+\sqrt{5})}/2,$ $\pm\sqrt{(5-\sqrt{5})}/2$ 0	$\pi/5$

Tabulka 8.3: Integrační body a váhy Gaussovy-Čebyševovy kvadratury na intervalu $\langle -1, 1 \rangle$

Mezi další druhy Gaussovy kvadratury, které jsou taktéž založeny na ortogonálních systémech polynomů vůči příslušnému skalárnímu součinu, jsou ku příkladu **Gaussova-Hermiteova** a **Gaussova-Laguerreova** kvadraturní formule, kterou důkladně popisuje třeba [1]. Vzhledem k rozsahu práce nebudeme již zmiňovat vzniklou chybu Gaussovy kvadratury, ale odkážeme se na [1, 2, 8], kde jsou chyby popsány. V praxi využíváme tabelované váhy a body, které pouze přetransformujeme na obecný interval $\langle a, b \rangle$.

¹Pro n integračních vah v Gaussově-Čebyševově kvadratuře platí, že $A_0 = A_1 = A_2 = \dots = A_{n-1}$.



Obrázek 8.2: Čebyševovy polynomy nultého až osmého řádu

8.3 Numerické experimenty pro Gaussovy kvadratury

Příklad 8.3.1. Zkusme nyní pomocí Gaussovy-Legendreovy kvadratury aproximovat nevlastní integrál

$$\int_0^1 \frac{1}{\sqrt{x}} dx.$$

Analytická hodnota je 2, jak jsme vyřešili v (6.1.1) na str. 53. V následující tabulce můžeme vidět výsledky aproximace:

n	absolutní chyba
2	$3,5 \cdot 10^{-1}$
4	$1,9 \cdot 10^{-1}$
8	$1,0 \cdot 10^{-1}$
16	$5,3 \cdot 10^{-2}$
32	$2,7 \cdot 10^{-2}$

Tabulka 8.4: Aproximace pomocí Gaussovy-Legendreovy kvadratury

△

Příklad 8.3.2. Zvolme nyní $f(x) = \frac{10}{4+x^2}$ a aproximuje následující integrál pomocí Gaussovy-Čebyševovy kvadratury

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} f(x) dx.$$

Zde jsme taktéž pomocí programu Maple zjistili analytickou hodnotu tohoto integrálu, která je $\frac{5\pi}{3}$. Výsledky aproximace uvádíme v následující tabulce

n	absolutní chyba
2	$9,3 \cdot 10^{-2}$
4	$4,8 \cdot 10^{-4}$
8	$1,3 \cdot 10^{-8}$
16	$1,7 \cdot 10^{-15}$
32	0

Tabulka 8.5: Aproximace pomocí Gaussovy-Čebyševovy kvadratury

Vidíme, že u $n = 16$ se již pohybujeme při počítačové nule.

△

Kapitola 9

Závěr

V jednotlivých kapitolách jsme měli možnost se seznámit s různými pohledy na numerické integrování. Zjistili jsme, že analýzou chyb můžeme zpřesnit kvadraturu, jako jsme udělali u adaptivního Simpsonova pravidla. Analýza chyb nám taktéž pomohla při tvorbě Eulerovy-Maclaurinovy expanze a s ní spojenou Richardsonovou extrapolací, kterou jsme využili pro zpřesnění hodnot lichoběžníkového pravidla pro různá h . Kapitola o rekurzivních metodách nám pomohla při počítání prvního sloupce Rombergova algoritmu, protože pro vysoká n bychom zbytečně ztráceli mnoho času při počítání funkčních hodnot v bodech, které jsme již vypočetli v předešlé úrovni.

Za zmínku zajisté stojí Gaussova kvadratura a adaptivní Simpsonovo pravidlo, které nám aproximovaly integrály, se kterými si uzavřené metody nevěděly rady.

Všechny uvedené metody jsme implementovali pomocí programu *Matlab*, kde jsme naprogramovali metody, které uvádíme v appendixu. V tomto programu byly taktéž vytvořeny grafy, které se v práci nacházejí. Pro určení některých analytických hodnot jsme využili program *Maple*.

Autor čerpal hlavně z anglických zdrojů, ovšem objevují se zde i čeští autoři, zejména pracovníci VŠB-TUO (vzhledem k roku vydání bakalářské práce), u kterých měl tu čest si vyslechnout cenné informace na přednáškách. Autor zde chce vyzvednout hlavně profesorku Wen Shen z Penn State University, jejíž přednášky ho inspirovaly celé psaní společně s knihou *An introduction to numerical analysis* od E. Süliho a D. F. Mayerse. Zajisté je taktéž nutno vyzvednout Marka Embreeho z Virginia Tech a jeho *Lecture Notes on Numerical Analysis*, který inspiroval autora zejména při tvoření grafů.

Pro autora této práce byla tato bakalářská práce přínosná hlavně tím, že spojuje široké spektrum matematiky bakalářského studia nejen z okruhu matematické analýzy, ale i prvky lineární algebry, algebry a především numerických metod. Práce mu ukázala možnou aplikaci výše zmíněných disciplín a utvrdila jeho pochopení v mnoha pojmech.

Literatura

- [1] M. Bailová. „Gaussovy kvadratury“. Bakalářská práce. VŠB - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2013.
- [2] M. Embree. *Lecture Notes on Numerical Analysis*. učební text na Virginia Tech, 2016. URL: <http://intranet.math.vt.edu/people/embree/math5466/nanotes.pdf>.
- [3] P. Grinfeld. *MathTheBeautiful*. 2014. URL: <https://www.youtube.com/c/MathTheBeautiful/videos>.
- [4] D. Lukáš. *Integrace*. 2020. URL: <https://home1.vsb.cz/~luk76/Teaching/NM1/Prednasky/3-Integrace.pdf>.
- [5] D. Lukáš. *Interpolace*. 2020. URL: <https://home1.vsb.cz/~luk76/Teaching/NM1/Prednasky/1-Interpolace.pdf>.
- [6] Z. Němečková. „Extrapolční metody pro numerické integrování“. Bakalářská práce. Masarykova univerzita, Přírodovědecká fakulta, 2013.
- [7] W. Shen. *An Introduction to Numerical Computation*. World Scientific, 2019.
- [8] E. Süli a D. F. Mayers. *An introduction to numerical analysis*. Cambridge University Press, 2003.
- [9] P. Vodstrčil. *Porovnáváme lichoběžníkové a obdélníkové pravidlo*. 2010. URL: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/prednaska_0.pdf.
- [10] E. W. Weisstein. *Newton-Cotes Formulas*. Dostupné z MathWorld. URL: <https://mathworld.wolfram.com/Newton-CotesFormulas.html>.

(Citace na webové stránky jsou uváděny k datu 30.4.2021.)

Příloha A

Ukázky kódů

V tomto apendixu si ukážeme jednotlivé kódy, které byly pro tuto bakalářskou práci vytvořeny v programu *Matlab*. Každý kód v sobě navíc obsahuje výpis výsledků, který v apendixu neuvádíme.

Složené obdélníkové pravidlo

```
1 function [Vysledek, Chyba] = Slozene_obdelnikove_pravidlo(f, a, b, m)
2 %Aproximace integrálu pomocí složeného obdélníkového pravidla.
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   m - počet podintervalů
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10 %   Chyba - odhad chyby způsobené při integraci
11
12 %výpočet integrálu
13 h = (b-a)/m;
14 x = a+h/2 : h : b-h/2;
15 Vysledek = h*sum(f(x));
16 %výpočet odhadu chyby
17 syms x
18 f2d(x) = diff(f(x),2); %vypočtení druhé derivace
19 f2dMax = fminbnd(matlabFunction(-abs(f2d(x))),a,b);
20 Chyba = (b-a)/24*f2dMax*h^2;
```

Algoritmus A.1: Složené obdélníkové pravidlo

Složené lichoběžníkové pravidlo

```
1 function [Vysledek, Chyba] = Slozene_lichobeznikove_pravidlo(f, a, b, m)
2 %Aproximace integrálu pomocí složeného lichoběžníkového pravidla.
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   m - počet podintervalů
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10 %   Chyba - odhad chyby způsobené při integraci
11
12 %výpočet integrálu
13 h = (b-a)/m;
14 x = a+h : h : b-h;
15 Vysledek = h * ((f(a) + f(b))/2 + sum(f(x)));
16 %výpočet odhadu chyby
17 syms x
18 f2d(x) = diff(f(x),2); %vypočtení druhé derivace
19 f2dMax = fminbnd(matlabFunction(-abs(f2d(x))),a,b);
20 Chyba = (b-a)/12*f2dMax*h^2;
```

Algoritmus A.2: Složené lichoběžníkové pravidlo

Složené Simpsonovo pravidlo

```
1 function [Vysledek, Chyba] = Slozene_Simpsonovo_pravidlo(f, a, b, m)
2 %Aproximace integrálu pomocí složeného Simpsonova pravidla.
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   m - 2*m podintervalů
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10 %   Chyba - odhad chyby způsobené při integraci
11
12 %výpočet integrálu
13 h = (b-a)/(2*m);
14 xodd = a+h : 2*h : b-h;
15 xeven = a+2*h : 2*h : b-2*h;
16 Vysledek = h/3 * (f(a) + f(b) + 4*sum(f(xodd)) + 2*sum(f(xeven)));
17 %výpočet odhadu chyby
18 syms x
19 f4d(x) = diff(f(x),4); %vypočtení čtvrté derivace
20 f4dMax = fminbnd(matlabFunction(-abs(f4d)), a, b);
21 Chyba = (b-a)/180*f4dMax*h^4;
```

Algoritmus A.3: Složené Simpsonovo pravidlo

Rekurzivní lichoběžníkové pravidlo

```
1 function [Vysledek, Chyba] = Rekurzivni_lichobeznikove_pravidlo(f, a, b, e)
2 %Aproximace integrálu pomocí rekurzivního lichoběžníkového pravidla.
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   e - maximální relativní chyba
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10 %   Chyba - odhad chyby způsobené při integraci
11
12 %výpočet integrálu
13 h_m = b-a;
14 T_m = h_m/2*(f(a) + f(b));
15 rel_err = inf;
16 while rel_err >= e
17     T = T_m;
18     h_m = h_m/2;
19     xodd = a+h_m : 2*h_m : b-h_m;
20     T_m = T/2 + h_m*sum(f(xodd));
21     rel_err = abs(T-T_m)/abs(T_m);
22 end
23 Vysledek = T_m;
24 %výpočet odhadu chyby
25 syms x
26 f2d(x) = diff(f(x),2); %vypočtení druhé derivace
27 f2dMax = fminbnd(matlabFunction(-abs(f2d)),a,b);
28 Chyba = (b-a)/12*f2dMax*h_m^2;
```

Algoritmus A.4: Rekurzivní lichoběžníkové pravidlo

Rekurzivní lichoběžníkové pravidlo s dělením na třetiny

```
1 function [Vysledek, Chyba] = Rekurzivni_lichobeznikove_pravidlo_3(f, a, b, e)
2 %Aproximace integrálu pomocí rekurz. lich. pravidla s dělením na třetiny.
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   e - maximální relativní chyba
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10 %   Chyba - odhad chyby způsobené při integraci
11
12 %výpočet integrálu
13 h_m = b-a;
14 T_m = h_m/2*(f(a) + f(b));
15 m = 0;
16 rel_err = inf;
17 while rel_err >= e
18     Suma = 0;
19     T = T_m;
20     h_m = h_m/3;
21     for i = 1:3^(m+1)-1
22         if mod(i,3) > 0
23             Suma = Suma + f(a+i*h_m);
24         end
25     end
26     T_m = T/3 + h_m*Suma;
27     rel_err = abs(T-T_m)/abs(T_m);
28     m = m + 1;
29 end
30 Vysledek = T_m;
31 %výpočet odhadu chyby
32 syms x
33 f2d(x) = diff(f(x),2); %vypočtení druhé derivace
34 f2dMax = fminbnd(matlabFunction(-abs(f2d)),a,b);
35 Chyba = (b-a)/12*f2dMax*h_m^2;
```

Algoritmus A.5: Rekurzivní lichoběžníkové pravidlo s dělením na třetiny

Rekurzivní Simpsonovo pravidlo

```
1 function [Vysledek, Chyba] = Rekurzivni_Simpsonovo_pravidlo(f, a, b, e)
2 %Aproximace integrálu pomocí rekurzivního Simpsonova pravidla
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   e - maximální relativní chyba
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10 %   Chyba - odhad chyby způsobené při integraci
11
12 %výpočet integrálu
13 h_m = (b-a)/2;
14 S_m = h_m/3 * (f(a) + f(b) + 4*f(a+h_m));
15 Suma_minula = f(a+h_m);
16 rel_err = inf;
17 while rel_err >= e
18     S = S_m;
19     h_m = h_m/2;
20     xodd = a+h_m: 2*h_m :b-h_m;
21     Suma = sum(f(xodd));
22     S_m = S/2 + h_m/3*(4*Suma - 2*Suma_minula);
23     Suma_minula = Suma;
24     rel_err = abs(S-S_m)/abs(S_m);
25 end
26 Vysledek = S_m;
27 %výpočet odhadu chyby
28 syms x
29 f4d(x) = diff(f(x),4); %vypočtení čtvrté derivace
30 f4dMax = fminbnd(matlabFunction(-abs(f4d)),a,b);
31 Chyba = (b-a)/180*f4dMax*h_m^4;
```

Algoritmus A.6: Rekurzivní Simpsonovo pravidlo

Rekurzivní Simpsonovo pravidlo s dělením na třetiny

```
1 function [Vysledek, Chyba] = Rekurzivni_Simpsonovo_pravidlo_3(f, a, b, e)
2 %Aproximace integrálu pomocí rekurz. Simps. pravidla s dělením na třetiny
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   e - maximální relativní chyba
8 %   Vystupní hodnoty
9 %   Vysledek - výsledek integrace
10 %   Chyba - odhad chyby způsobené při integraci
11
12 %výpočet integrálu
13 h_m = (b-a)/2;
14 S_m = h_m/3 * (f(a) + f(b) + 4*f(a+h_m));
15 m = 0;
16 rel_err = inf;
17 while rel_err >= e
18     S = S_m;
19     h_m = h_m/3;
20     odd_m = a + h_m;
21     suma_odd = f(odd_m);
22     for i = 2:2*3^m
23         if mod(i,2) == 1
24             odd_m = odd_m + 2*h_m;
25         else
26             odd_m = odd_m + 4*h_m;
27         end
28         suma_odd = suma_odd + f(odd_m) ;
29     end
30     even_m = a + 2*h_m;
31     suma_even = f(even_m);
32     for i = 2:2*3^m
33         if mod(i,2) == 1
34             even_m = even_m + 4*h_m;
35         else
36             even_m = even_m + 2*h_m;
37         end
```



```

38     suma_even = suma_even + f(even_m);
39 end
40 S_m = S/3 + h_m/3 * (4*suma_odd + 2*suma_even);
41 rel_err = abs(S-S_m)/abs(S_m);
42 m = m + 1;
43 end
44 Vysledek = S_m;
45 %výpočet odhadu chyby
46 syms x
47 f4d(x) = diff(f(x),4); %vypočtení čtvrté derivace
48 f4dMax = fminbnd(matlabFunction(-abs(f4d)),a,b);
49 Chyba = (b-a)/180*f4dMax*h_m^4;

```

Algoritmus A.7: Rekurzivní Simpsonovo pravidlo s dělením na třetiny

Aproximace první derivace pomocí Richardsonovy extrapolace

```

1 function Vysledek = Richardsonova_extrapolace_derivace(f, a, n)
2 %Aproximace první derivace pomocí Richardsonovy extrapolace
3 %   Vstupní hodnoty
4 %   f - funkce určená pro první derivaci
5 %   a - bod, ve kterém aproximujeme
6 %   n - počet sloupců a řádků matice
7 %   Výstupní hodnoty
8 %   Vysledek - výsledek derivování
9
10 %inicializace t
11 h = 1;
12 t(1,1) = (f(a + h/2) - f(a - h/2))/h;
13 %Richardsonova extrapolace
14 for j = 2:n
15     h = h/2;
16     t(j,1) = (f(a + h/2) - f(a - h/2))/h;
17     for k = 2:j
18         t(j,k) = (2^(2*(k-1))*t(j,k-1) - t(j-1,k-1)) / (2^(2*(k-1)) - 1);
19     end
20 end
21 Vysledek = t(n,n);

```

Algoritmus A.8: Aproximace první derivace pomocí Richardsonovy extrapolace

Rombergova metoda

```
1 function Vysledek = Rombergova_metoda(f, a, b, n)
2 %Aproximace integrálu pomocí Rombergovy metody
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   n - počet sloupců a řádků matice
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10
11 %inicializace r
12 h = b - a;
13 r(1,1) = h*(f(a) + f(b))/2;
14 %Rombergův algoritmus
15 for j = 2:n
16     h = h/2;
17     xodd = a+h: 2*h :b-h;
18     r(j,1) = r(j-1,1)/2 + h*sum(f(xodd));
19     for k = 2:j
20         r(j,k) = (2^(2*(k-1))*r(j,k-1) - r(j-1,k-1)) / (2^(2*(k-1)) - 1);
21     end
22 end
```

Algoritmus A.9: Rombergova metoda

Adaptivní Simpsonovo pravidlo

```
1 function Vysledek = Adaptivni_Simpsonovo_pravidlo(f, a, b, e)
2 %Aproximace integrálu pomocí adaptivního Simpsonova pravidla
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   e - ukončovací podmínka
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10
11 %výpočet integrálu
12 c = (a+b)/2;
13 S1 = (b-a)/6*[f(a) + 4*f((a+b)/2) + f(b)];
14 S1L = (c-a)/6*[f(a) + 4*f((a+c)/2) + f(c)];
15 S1R = (b-c)/6*[f(c) + 4*f((b+c)/2) + f(b)];
16 S2 = S1L + S1R;
17 %rekurze
18 if abs(S2 - S1) < 15*e
19     Vysledek = S2 + (S2 - S1)/15;
20 else
21     c = (b+a)/2;
22     L = adaptivni_Simpsonovo_pravidlo(f, a, c, e/2);
23     R = adaptivni_Simpsonovo_pravidlo(f, c, b, e/2);
24     Vysledek = L + R;
25 end
```

Algoritmus A.10: adaptivní Simpsonovo pravidlo

Gaussova-Legendreova kvadratura

```
1 function Vysledek = Gaussova_Legendreova_kvadratura(f, a, b, n)
2 %Aproximace integrálu pomocí Gaussovy-Legendreovy kvadratury
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   a - dolní mez intervalu
6 %   b - horní mez intervalu
7 %   n - počet integračních bodů a vah
8 %   Výstupní hodnoty
9 %   Vysledek - výsledek integrace
10
11 %určení kořenů Legendreova polynomu
12 syms x
13 Leg = legendreP(n, x);
14 I_body = vpa(solve(Leg,x,'MaxDegree',n));
15
16 %sestavení soustavy n lineárních rovnic s n neznámými vahami
17 c = zeros(n, 1);
18 for i = 1 : n
19     c(i) = int(x.^(i-1), -1, 1);
20 end
21 Vander = fliplr(vander(I_body))';
22
23 %řešení soustavy a určení výsledku
24 I_vahy = linsolve(Vander,c);
25
26 %transformace na obecný interval a určení výsledku
27 I_body = (b-a)/2*I_body + (b+a)/2;
28 I_vahy = (b-a)/2*I_vahy;
29 Vysledek = sum(f(I_body).*I_vahy);
```

Algoritmus A.11: Gaussova-Legendreova kvadratura

Gaussova-Čebyševova kvadratura

```
1 function Vysledek = Gaussova_Cebysevova_kvadratura(f, n)
2 %Aproximace integrálu pomocí Gaussovy-Čebyševovy kvadratury
3 %   Vstupní hodnoty
4 %   f - funkce určená pro integraci
5 %   n - počet integračních bodů a vah
6 %   Výstupní hodnoty
7 %   Vysledek - výsledek integrace
8 %výpočet integrálu
9
10 %určení kořenů Čebyševova polynomu
11 syms x;
12 Che = chebyshevT(n, x);
13 I_body = vpa(solve(Che,x,'MaxDegree',n));
14
15 %sestavení soustavy n lineráních rovnic s n neznámými vahy
16 c = zeros(n, 1);
17 for i = 1:n
18     c(i) = int(x^(i-1)./(sqrt(1-x.^2)), -1, 1);
19 end
20 V = fliplr(vander(I_body))';
21
22 %řešení soustavy a určení výsledku
23 I_vahy = linsolve(V,c);
24 Vysledek = sum(f(I_body).*I_vahy);
```

Algoritmus A.12: Gaussova-Čebyševova kvadratura