

Obsah

1	Numerické úlohy v analýze a aproximace funkcí	9
1.1	Modelování úloh matematické analýzy na počítači.	9
1.2	Příklad.	10
1.3	Třídy aproximujících funkcí.	11
1.4	Výběr aproximující funkce.	12
1.5	Měření chyby aproximace: spojitý případ.	13
1.6	Příklad.	14
1.7	Měření chyby aproximace: diskrétní případ.	14
1.8	Příklad.	15
1.9	Polynomy jako aproximující funkce.	16
1.10	Věta.	16
1.11	Poznámka.	16
2	Aproximace Taylorovým polynomem	16
2.1	Taylorův polynom.	16
2.2	Příklad.	17
2.3	Chyba aproximace.	17
2.4	Věta.	18
2.5	Odhad chyby aproximace.	18
2.6	Příklad.	18
2.7	Řád chyby aproximace.	19
2.8	Chyba aproximace a celková chyba.	19
2.9	Úlohy.	20
2.9.1	20
2.9.2	20
2.9.3	20
2.9.4	20
2.9.5	21
2.9.6	21
3	Aproximace interpolačním polynomem	21
3.1	Obecně o interpolační aproximaci.	21
3.2	Poznámky.	22
3.2.1	22
3.2.2	22
3.3	Lagrangeův interpolační polynom.	23
3.4	Příklad.	23
3.5	Lineární a kvadratická interpolace.	24
3.6	Příklad.	25
3.7	Vliv nepřesnosti ve vstupních datech.	25
3.8	Příklad.	25

3.9	Chyba aproximace interpolačním polynomem.	26
3.10	Věta.	26
3.11	Příklad.	27
3.12	Odhad chyby aproximace.	27
3.13	Příklad.	27
3.14	Extrapolace.	28
3.15	Příklad. Lineární extrapolace	28
3.16	Nevillův algoritmus.	28
3.17	Příklad.	29
3.18	Použití Nevillova algoritmu.	30
3.19	Příklady.	30
	3.19.1	30
	3.19.2	30
3.20	Newtonův interpolační polynom.	31
3.21	Poměrné diference.	31
3.22	Příklad.	33
3.23	Koeficienty interpolačního polynomu.	33
3.24	Příklad.	34
3.25	Algoritmy.	34
3.26	Příklad.	35
3.27	Ekvidistantní uzly interpolace.	36
3.28	Příklad.	37
3.29	Interpolační polynom s dopřednými diferencemi.	37
3.30	Příklad.	38
3.31	Interpolační polynom se zpětnými derivacemi.	38
3.32	Konvergence interpolačního procesu.	39
3.33	Úlohy.	41
	3.33.1	41
	3.33.2	41
	3.33.3	41
	3.33.4	41
	3.33.5	41
	3.33.6	41
	3.33.7	42
	3.33.8	42
	3.33.9	42
	3.33.10	42
	3.33.11	42
	3.33.12	43
	3.33.13	43

4	Interpolace racionální funkcí a spline-funcí	43
4.1	Interpolace racionální funkcí.	43
4.2	Příklad.	44
4.3	Aproximace funkcemi, které jsou po částech polynomy.	45
4.4	Příklad: po částech lineární interpolace.	45
4.5	Interpolace kubickými spline-funcemi.	46
4.6	Konstrukce interpolační spline-funkce.	47
4.7	Příklad.	50
4.8	Vlastnosti kubických spline-funcí.	51
4.8.1	Věta.	52
4.9	Úlohy.	52
4.9.1		52
4.9.2		52
4.9.3		53
4.9.4		53
4.9.5		53
4.9.6		53
4.9.7		53
4.9.8		54
4.9.9		54
4.9.10		54
5	Ortogonalní systémy funkcí	55
5.1	Lineárně nezávislé systémy funkcí.	55
5.2	Příklady	56
5.2.1		56
5.2.2		56
5.2.3		56
5.3	Ortogonalní systémy funkcí.	56
5.4	Příklady.	58
5.4.1		58
5.4.2		58
5.4.3		59
5.4.4		59
5.4.5		59
5.4.6		60
5.5	Ortogonalní polynomy.	60
5.6	Příklady	61
5.6.1		61
5.6.2		62
5.6.3		62
5.6.4		63
5.6.5		63

5.7	Úlohy	63
5.7.1		63
5.7.2		64
5.7.3		64
5.7.4		64
5.7.5		64
5.7.6		64
6	Metoda nejmenších čtverců	64
6.1	Nejlepší L_2 -aproximace.	64
6.2	Normální rovnice.	65
6.3	Příklady.	66
6.3.1		66
6.3.2		67
6.4	Existence nejlepší L_2 -aproximace.	67
6.5	Věta.	68
6.6	Příklady.	68
6.6.1		68
6.6.2		69
6.7	Řešení normálních rovnic a volba základních funkcí.	69
6.8	Příklad.	71
6.9	Algoritmus metody nejmenších čtverců.	72
6.10	Úlohy	73
6.10.1		73
6.10.2		74
6.10.3		74
6.10.4		74
6.10.5		74
6.10.6		75
6.10.7		75
6.10.8		75
7	Aproximace trigonometrickými polynomy	75
7.1	Fourierovy řady.	75
7.2	Věta. Fourierova analýza – spojitý případ.	76
7.3	Přibližný výpočet Fourierových koeficientů.	77
7.4	Věta. Fourierova analýza – diskrétní případ.	78
7.5	Praktické důsledky věty 7.4.	79
7.6	Příklad.	80
7.7	Rychlá Fourierova analýza.	81
7.8	Algoritmus rychlé Fourierovy analýzy.	82
7.9	Fourierova syntéza.	83
7.10	Úlohy.	83

7.10.1	83
7.10.2	84
7.10.3	84
7.10.4	84
7.10.5	84
7.10.6	84
7.10.7	85
7.10.8	85
8 Čebyševova aproximace	85
8.1 Nejlepší stejnoměrná aproximace.	85
8.2 Věta.	86
8.3 Příklad.	86
8.4 Algoritmus metody minimalizace maximální chyby.	86
8.5 Příklad.	87
8.6 Úlohy	88
8.6.1	88
8.6.2	88
9 Zvyšování přesnosti výpočtů extrapolací	88
9.1 Zvyšování přesnosti přibližných výpočtů.	88
9.2 Příklad.	89
9.3 Richardsonova extrapolace.	90
9.4 Algoritmus opakované Richardsonovy extrapolace.	90
9.5 Příklad.	91
9.6 Varianty extrapoláčního algoritmu.	92
9.7 Úlohy	92
9.7.1	92
9.7.2	92
9.7.3	93
9.7.4	93
10 Obecně o metodách pro výpočet integrálu a derivace	94
10.1 Úvod.	94
10.2 Princip metod pro přibližný výpočet integrálu a derivace.	94
10.3 Příklad.	95
10.4 Kvadrurní vzorce.	95
10.5 Příklad.	95
10.6 Interpolační kvadrurní vzorce.	97
10.7 Složené kvadrurní vzorec.	98
10.8 Úloha.	99

11 Newtonovy-Cotesovy kvadrturní vzorce	99
11.1 Odvození Newtonových-Cotesových vzorců.	99
11.1.1 Obdélníkové pravidlo.	100
11.1.2 Lichoběžníkové pravidlo.	100
11.1.3 Simpsonovo pravidlo.	100
11.2 Příklad.	101
11.3 Algoritmus složeného Simpsonova pravidla.	101
11.4 Příklad.	102
11.5 Chyba obdélníkového, lichoběžníkového a Simpsonova pravidla.	103
11.6 Příklad.	105
11.7 Obecné Newtonovy-Cotesovy vzorce.	105
11.8 Jak dosáhnout požadované přesnosti.	106
11.9 Příklad.	107
11.10 Úlohy.	107
11.10.1	107
11.10.2	108
11.10.3	108
11.10.4	108
11.10.5	108
11.10.6	108
11.10.7	108
11.10.8	109
12 Eulerův-Maclaurinův vzorec.	
Rombergova metoda	109
12.1 Eulerův-Maclaurinův vzorec.	109
12.2 Kvadrturní vzorce s hodnotami derivací.	109
12.3 Výpočet integrálu z periodické funkce přes periodu.	110
12.4 Příklad.	110
12.5 Nekonečný interval integrace.	111
12.6 Příklad.	111
12.7 Rombergova integrační metoda.	111
12.8 Příklad.	112
12.9 Vlastnosti Rombergovy metody.	113
12.10 Příklad.	113
12.11 Úlohy	114
12.11.1	114
12.11.2	114
12.11.3	114
12.11.4	114
12.11.5	115
12.11.6	115

12.11.7	115
12.11.8	115
13 Gaussovy kvadrurní vzorce	116
13.1 Vzorce s neekvidistantními uzly.	116
13.2 Věta.	116
13.3 Vlastností Gaussových kvadrurních vzorců.	116
13.4 Gaussovy kvadrurní vzorce pro interval $\langle -1, 1 \rangle$. Chyba aproximace	117
13.5 Příklady.	118
13.5.1	119
13.6 Použití Gaussových vzorců.	119
13.7 Příklady.	120
13.7.1	120
13.7.2	121
13.8 Přibližný výpočet integrálů s váhovou funkcí.	121
13.9 Úlohy	122
13.9.1	122
13.9.2	122
13.9.3	122
13.9.4	123
13.9.5	123
13.9.6	123
13.9.7	123
14 Několik poznámek k numerické kvadratuře	124
14.1 Adaptivní programy pro výpočet integrálů.	124
14.2 Použití adaptivních programů.	124
14.2.1 Stabilita kvadrurních algoritmů.	125
15 Numerické derivování	126
15.1 Derivování interpolačního polynomu.	126
15.2 Příklad.	126
15.3 Chyba aproximace.	127
15.4 Některé často užívané vzorce.	128
15.5 Příklad.	129
15.6 Podmíněnost numerického výpočtu derivace.	129
15.7 Příklad.	130
15.8 Aproximace derivací diferencemi.	130
15.9 Úlohy	131
15.9.1	131
15.9.2	131
15.9.3	131

15.9.4	131
15.9.5	132
15.9.6	132
15.9.7	132

I. Aproximace funkcí

1 Numerické úlohy v analýze a aproximace funkcí

1.1 Modelování úloh matematické analýzy na počítači.

Předmětem studia matematické analýzy jsou funkce a operace s nimi. Typickými úlohami matematické analýzy jsou např. výpočet integrálu z dané funkce přes interval, výpočet hodnoty derivace, řešení diferenciálních rovnic nebo i pouhé stanovení hodnoty funkce v některém bodě. Pro úlohy matematické analýzy je charakteristické, že zpravidla nejde o numerické úlohy, tj. takové úlohy, kde vstupní i výstupní data jsou vektory o konečném počtu složek [21]. Protože na počítači můžeme přímo modelovat (naprogramovat) pouze úlohy numerické (příkladem mohou sloužit úlohy lineární algebry, viz [21]), bývá při řešení úloh matematické analýzy na počítači nezbytným přípravným krokem přibližné nahrazení (aproximace) dané matematické úlohy úlohou numerickou. Numerické metody matematické analýzy mají proto poněkud jiný charakter než numerické metody algebry.

Jedním ze základních úkolů numerických metod matematické analýzy je studium aproximací funkcí. Při numerickém řešení úloh matematické analýzy totiž často nahrazujeme danou funkci f , vystupující v řešené matematické úloze, jinou funkcí φ , která v nějakém vhodném smyslu napodobuje funkci f a snadno se přitom matematicky zpracovává či modeluje na počítači. Tuto funkci φ pak nazýváme *aproximací* (přiblížením) funkce f .

Oblasti matematiky, v nichž používáme aproximace, jsou značně různorodé. V tomto sešitě se budeme zabývat převážně těmi úlohami matematické analýzy, jejichž vstupními a výstupními daty jsou reálné funkce jedné reálné proměnné. Již pouhý výpočet funkčních hodnot takových funkcí na počítači se provádí užitím aproximací φ , jejichž hodnoty se dají vypočítat pomocí konečného počtu aritmetických a logických operací.¹⁾ Tyto aproximace jsou ovšem pro řadu funkcí již zabudovány do výpočetního systému a uživatel počítače si často ani neuvědomuje, že píše-li ve svém programu např. $Y=\text{SIN}(X)$, nahrazuje výpočet hodnoty funkce $\sin x$ např. výpočtem hodnoty jistého polynomu.

Jiným typickým příkladem jsou numerické metody pro výpočet určitého integrálu z funkce f . Zde nahrazujeme funkci f funkcí φ , která se snadno integruje, např. polynomem.

Další oblastí numerické matematiky založenou na užití aproximací je zpracování výsledků měření. Hledáme tu zpravidla jednoduchý analytický výraz vyjadřující (přibližně) funkční závislost, zadanou tabulkou naměřených hodnot.

Při použití aproximací tedy místo původní úlohy, ve které vystupovala funkce f , řešíme úlohu, v níž místo f vystupuje její aproximace φ . To má ovšem za následek, že výsledkem výpočtu nebude přesné řešení původní úlohy. Úkolem numerických metod analýzy je

¹⁾ To jsou totiž nejobecnější funkce, jejichž hodnoty se dají na počítači přímo vyčíslit.

proto také zkoumat, co můžeme říci o odchylce získaného přibližného řešení od přesného (teoretického) řešení dané úlohy.

1.2 Příklad.

Hledejme na intervalu $\langle 0, \frac{1}{2} \rangle$ řešení počáteční úlohy

$$(1.2.1) \quad y'' = -xy, \quad y(0) = 1, \quad y'(0) = 0.$$

Úloha (1.2.1) není numerická úloha, neboť ani funkci y , ani její derivace y', y'' nemůžeme specifikovat konečným počtem čísel. Tuto matematickou úlohu můžeme aproximovat numerickou úlohou např. tak, že jako výstupní data budeme hledat aproximaci řešení y pro $x = h, 2h, 3h, \dots, \frac{1}{2}$, $h = 1/(2N)$, kde N je přirozené číslo, a rovnici (1.2.1) aproximujeme vhodnou diferenční rovnicí [21]. Výsledkem tohoto postupu bude převedení počáteční úlohy (1.2.1) na výpočet přibližných hodnot řešení v uvedených bodech $x = h, 2h, 3h, \dots$ pomocí jistých rekurentních vzorců, které se dají na počítači bez větších problémů naprogramovat. Nepřesnost takto získaného přibližného řešení se dá odhadnout postupy, o kterých pojednáme v kap. III.

Jiný možný přístup k numerickému řešení úlohy (1.2.1) spočívá v tom, že řešení (1.2.1) aproximujeme spojitou funkcí, která se dá spolu se svými derivacemi specifikovat konečným počtem parametrů. Příkladem takové funkce je polynom.

Hledejme např. polynom $p(x) = a + bx + cx^2 + dx^3$ třetího stupně, který co nejlépe splňuje obě počáteční podmínky a rovnici (1.2.1). Postupným derivováním dostaneme $p'(x) = b + 2cx + 3dx^2$, $p''(x) = 2c + 6dx$. Počáteční podmínky můžeme splnit přesně, stačí položit $a = 1$, $b = 0$. Máme tedy $p(x) = 1 + cx^2 + dx^3$.

Dosaďme $p(x)$ do diferenciální rovnice (1.2.1). Dostaneme $2c + 6dx = -x(1 + cx^2 + dx^3)$ neboli $2c + 6dx = -x - cx^3 - dx^4$. Porovnáním koeficientů u stejných mocnin x^k pro $k = 0, 1, 3$, odtud dostaneme $c = 0$, $d = -\frac{1}{6}$. Diferenciální rovnici tedy nesplníme přesně, ale s chybou $-dx^4 = x^4/6$. Nalezené přibližné řešení je tvaru

$$(1.2.2) \quad p(x) = 1 - \frac{1}{6}x^3.$$

Z našeho postupu vyplývá, že $p(x)$ je přesným řešením počáteční úlohy

$$(1.2.3) \quad y'' = -xy + \frac{1}{6}x^4, \quad y(0) = 1, \quad y'(0) = 0.$$

Toto je rovněž jeden ze způsobů, kterými nás numerická matematika informuje o přesnosti získaného přibližného řešení: Ukáže se, že přibližné řešení je přesným řešením porušené původní úlohy, a udá se nějaká informace o velikosti této poruchy. Je-li původní matematická úloha dobře podmíněná, tj. stabilní vůči takovým poruchám, lze pak usuzovat na velikost odchylky přesného a přibližného řešení dané úlohy. V našem případě pro $x \in \langle 0, \frac{1}{2} \rangle$, platí $x^4/6 \leq \frac{1}{96}$. Z (1.2.2) dostáváme $p(0,5) \doteq 0,979$; dá se ukázat, že odchylka této hodnoty od $y(0,5)$ je menší než $\frac{1}{2} \cdot 10^{-4}$.

1.3 Třídy aproximujících funkcí.

V celé této kapitole se budeme zabývat aproximacemi spojitých funkcí jedné proměnné. Při výběru vhodné aproximace postupujeme v numerické matematice tak, že předem zvolíme tvar aproximující funkce, ve kterém vystupují některé proměnné parametry, a hodnoty parametrů se pak snažíme určit tak, aby získaná aproximace vyhovovala našim požadavkům. Tak např. v odst. 1.2 jsme volili aproximaci řešení ve tvaru polynomu

$$a + bx + cx^2 + dx^3.$$

Všimněme si také toho, že uvedený polynom je lineární kombinací funkcí $1, x, x^2, x^3$.

Obecně se při výběru aproximace velmi často postupuje takto: Zvolíme nejprve pevně systém jednoduchých *základních funkcí* $\varphi_0, \varphi_1, \dots, \varphi_n$ takových, které se snadno matematicky zpracovávají nebo se s nimi dobře pracuje na počítači. Danou funkci f pak aproximujeme lineární kombinací φ základních funkcí:

$$(1.3.1) \quad \varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_n\varphi_n(x).$$

Otázka výběru aproximace k funkci f se tak převádí na určení hodnot parametrů c_0, c_1, \dots, c_n podle nějakého kritéria vhodného pro tu či onu konkrétní úlohu: (K řešení úloze případně přihlídneme také při volbě základních funkcí.) Protože aproximace φ daná výrazem (1.3.1) závisí na parametrech c_0, c_1, \dots, c_n lineárně, říkáme, že (1.3.1) je *aproximace lineárního typu*. Při pevných základních funkcích $\varphi_0, \varphi_1, \dots, \varphi_n$ nazýváme množinu \mathcal{V}_n všech funkcí tvaru (1.3.1) *třídou aproximujících funkcí* (lineárního typu).

Příklady často používaných základních funkcí jsou

$$(1.3.2) \quad 1, x, x^2, \dots, x^n,$$

$$(1.3.3) \quad 1, (x - x_0), (x - x_0)^2, \dots, (x - x_0)^n, x_0 \quad \text{pevně,}$$

$$(1.3.4) \quad 1, \cos x, \sin x, \dots, \cos Lx, \sin Lx \quad (n = 2L),$$

$$(1.3.5) \quad 1, e^{ix}, e^{2ix}, \dots, e^{inx} \quad (i^2 = -1).$$

Jsou-li základní funkce $\varphi_0, \varphi_1, \dots, \varphi_n$ dány (1.3.2) nebo (1.3.3) je \mathcal{V}_n *třída polynomů* nejvýše n -tého stupně. Pro (1.3.4) a (1.3.5) se aproximující funkce φ lineárního typu nazývají *trigonometrické polynomy*.

Příkladem tříd aproximací, které nejsou lineárního typu, jsou *třídy racionálních funkcí* [např. tvaru $\varphi(x) = (a + bx)/(c + dx)$]. Teorie a praxe takových aproximací je - jako je tomu u většiny nelineárních problémů - ve srovnání s aproximacemi lineárního typu poměrně komplikovaná. Pro některé úlohy však aproximace nelineárního typu dávají velmi dobré výsledky.

1.4 Výběr aproximující funkce.

Budiž f daná funkce a φ její aproximace na intervalu $\langle a, b \rangle$. Funkci E danou pro $x \in \langle a, b \rangle$ vztahem

$$E(x) = f(x) - \varphi(x)$$

budeme nazývat *chybou aproximace*. Chceme-li vybrat funkci φ z dané třídy \mathcal{V}_n tak, aby byla dobrou aproximací funkce f , budeme se jistě snažit o to, aby chyba aproximace vyla v nějakém smyslu malá. Ve většině úloh nás přitom prakticky nezajímá znaménko chyby a kvalitu aproximace posuzujeme podle absolutní hodnoty chyby aproximace $|E(x)| = |f(x) - \varphi(x)|$.

Možností, jak posuzovat ("měřit") velikost funkce E na intervalu $\langle a, b \rangle$ je ale celá řada. Můžeme např. požadovat, aby maximální hodnota $|E(x)|$ na intervalu $\langle a, b \rangle$ byla menší než nějaké dané číslo ε . Jindy pro nás může být důležitá velikost $|E(x)|$ pouze v některých bodech intervalu $\langle a, b \rangle$. Můžeme požadovat, aby byl malý integrál z funkce $|E(x)|$ přes $\langle a, b \rangle$ nebo aby byla malá také absolutní hodnota derivací funkce E (pokud ovšem existují). To, jakým způsobem budeme chybu aproximace měřit, závisí tedy do značné míry na naší libovůli. Volbu vhodného kritéria pro měření chyby budeme provádět s přihlédnutím k řešené matematické úloze a způsobu zadání funkce f .

V dalším výkladu budeme rozlišovat dva základní typy kritérií pro měření chyby aproximace podle toho, zda kvalitu aproximace posuzujeme na základě hodnot chyby E (a příp. některých jejích derivací) ve všech bodech $x \in \langle a, b \rangle$ (*spojitý případ*) nebo jen podle hodnot ve vybrané soustavě bodů $x_i \in \langle a, b \rangle$, $i = 0, 1, \dots, m$ (*diskrétní případ*). V příkladu 1.2 jsme vlastně oba přístupy kombinovali. Aproximaci p řešení y úlohy (1.2.1) jsme tam hledali tak, aby veličina

$$\varrho(y, p) = |y(0) - p(0)| + |y'(0) - p'(0)| + \max_{x \in \langle 0, \frac{1}{2} \rangle} |y''(x) - p''(x)|$$

byla pokud možno malá. Chybu aproximace jsme tedy posuzovali podle hodnot E a E' v bodě $x = 0$ a hodnot E'' ve všech bodech $x \in \langle 0, \frac{1}{2} \rangle$. Řadu jiných, často používaných kritérií pro měření chyby aproximace uvádíme v odst. 1.5 a 1.7.

Vrat'me se nyní k problematice výběru aproximující funkce φ z dané třídy \mathcal{V}_n kterou chceme použít jako aproximaci dané funkce f na intervalu $\langle a, b \rangle$. Předpokládejme, že jsme již pevně zvolili nějaké kritérium $\varrho(f, \varphi) \geq 0$ pro posuzování chyby aproximace a že chceme zvolit $\varphi \in \mathcal{V}_n$ tak, aby hodnota $\varrho(f, \varphi)$ byla pokud možno malá. Je pak nasnadě otázka, zda v dané třídě \mathcal{V}_n existuje *nejlepší aproximace*, tj. aproximace φ^* taková, že pro všechny aproximace $\varphi \in \mathcal{V}_n$ platí

$$(1.4.1) \quad 0 \leq \varrho(f, \varphi^*) \leq \varrho(f, \varphi).$$

Dále nás bude zajímat, zda taková aproximace φ^* existuje jediná a jak ji efektivním způsobem sestavit, tj. jak stanovit příslušné hodnoty koeficientů c_0, c_1, \dots, c_n ve vyjádření (1.3.1).

O nejlepších aproximacích pojednává celá jedna oblast matematiky, *teorie aproximací* [20]. V tomto sešitě se některými příklady nejlepších aproximací budeme zabývat také. Je

však třeba mít na paměti, že z praktického hlediska je většinou primárním požadavkem to, aby chyba aproximace měřená veličinou $\varrho(f, \varphi)$ byla menší než nějaké předem dané číslo (požadovaná přesnost). Ke splnění tohoto požadavku není vždy nutné používat nejlepší aproximace, někdy je to dokonce neefektivní nebo nemožné.

1.5 Měření chyby aproximace: spojitý případ.

Jako měřítko kvality aproximace se ve spojitém případě často užívá veličina

$$(1.5.1) \quad \varrho_{\max}(f, \varphi) = \max_{x \in \langle a, b \rangle} |f(x) - \varphi(x)|.$$

Typickým příkladem úloh numerické matematické analýzy, při nichž se používají aproximace vybírané na základě (1.5.1), je přibližný výpočet hodnot funkcí zadaných různými analytickými výrazy na počítači. Nahrazujeme-li pro přibližný výpočet hodnoty takových funkcí na intervalu $\langle a, b \rangle$ hodnotami jejich aproximací, je přirozené požadovat, aby $|E(x)|$ byla malá pro všechna $x \in \langle a, b \rangle$. Tomuto požadavku právě odpovídá (1.5.1) nebo podobné kritérium pro relativní chybu. Nejlepší aproximace podle (1.5.1) se často nazývá *Čebyševova aproximace* nebo *nejlepší stejnoměrná aproximace* funkce f .

Ve spojitém případě se chyba aproximace měřívá rovněž pomocí veličin

$$(1.5.2) \quad \varrho_p(f, \varphi) = \left(\int_a^b |f(x) - \varphi(x)|^p dx \right)^{1/p}, \quad p \geq 1,$$

které (pro malá p) nezdůrazňují maximální absolutní hodnotu chyby aproximace. Ze souboru kritérií (1.5.2) je v praxi nejužívanější

$$(1.5.3) \quad \varrho_2(f, \varphi) = \left(\int_a^b |f(x) - \varphi(x)|^2 dx \right)^{1/2}.$$

Nejlepší aproximace podle (1.5.2) se nazývá *nejlepší L_p -aproximace*; pro (1.5.3) máme pak *nejlepší L_2 -aproximaci*.

Kritérium (1.5.3) je někdy výhodné přizpůsobit řešené úloze tak, že se v něm budou více uplatňovat hodnoty chyby aproximace v některých částech intervalu $\langle a, b \rangle$. Toho se dosáhne užitím veličiny

$$(1.5.4) \quad \varrho_{2,\omega}(f, \varphi) = \left[\int_a^b |f(x) - \varphi(x)|^2 \omega(x) dx \right]^{1/2},$$

kde ω je funkce spojitá a kladná na intervalu (a, b) , která může mít v koncových bodech a, b integrovatelné singularity. Funkce ω se nazývá *váhová funkce*; jako příklad může sloužit funkce $\omega(x) = (1 - x^2)^{-1/2}$ (setkáme se s ní v odst. 5.6.1).

Kritéria spojitého typu (1.5.1) až (1.5.4) se uplatní zejména při práci s funkcemi zadanými analytickým výrazem. Při zpracování výsledků měření a v řadě numerických metod matematické analýzy používáme jejich diskrétní obdoby popsané v odst. 1.7.

1.6 Příklad.

Jako ilustraci vlivu volby kritéria pro měření chyby aproximace na tvar nejlepší aproximace (při pevně dané třídě \mathcal{V}_n) uvedeme dvě nejlepší aproximace funkce $f(x) = \sqrt{x}$ na intervalu $\langle 0, 1 \rangle$ polynomem prvního stupně. Jako základní funkce (viz odst. 1.3) tedy volíme $\varphi_0(x) \equiv 1$, $\varphi_1(x) = x$ a nejlepší aproximaci hledáme ve třídě \mathcal{V}_1 aproximací tvaru $\varphi(x) = c_0 + c_1x$ [srov. (1.3.1)].

Nejlepší stejnoměrná aproximace φ_{max}^* má tvar $\varphi_{max}^*(x) = x + \frac{1}{8}$ (viz příkl. 8.3 a 8.5). Platí pro ni $\varrho_{max}(f, \varphi_{max}^*) = \frac{1}{8} = 0,125$. Nejlepší L_2 -aproximace φ_L^* má tvar $\varrho_{max} = \frac{4}{5}x + \frac{4}{15}$ (viz příkl. 6.6.1). Platí pro ni $\varrho_2(f, \varphi_L^*) = \left(\frac{1}{450}\right)^{1/2} \doteq 0,0471$ (nakreslete si grafy aproximované funkce i obou aproximací).

Měříme-li chybu aproximace φ_{max}^* podle (1.5.3) dostáváme $\varrho_2(f, \varphi_{max}^*) = \left(\frac{7}{960}\right)^{1/2} \doteq 0,0854$. Měříme-li chybu aproximace φ_L^* podle (1.5.1), dostáváme $\varrho_{max}(f, \varphi_L^*) = \frac{4}{15} \doteq 0,267$. (Hodnoty veličiny ϱ_2 se snadno ověří přímým výpočtem; hodnoty ϱ_{max} byly získány vyšetřením průběhu funkce $|E(x)|$.)

1.7 Měření chyby aproximace: diskrétní případ.

Funkce f , kterou chceme aproximovat, může být zadána různými způsoby. V praxi je velmi častý případ, kdy je funkce f dána tabulkou

$$\{(x_i, f(x_i)), i = 0, 1, \dots, m\}$$

funkčních hodnot $f(x)_i$ v bodech $x_i \in \langle a, b \rangle$. V takových případech měříme chybu aproximace diskrétními obdoby kritérií popsaných v odst. 1.5, v nichž vystupuje pouze daný konečný počet hodnot funkce f . Tak např. (1.5.1) přejde v

$$(1.7.1) \quad \varrho_{max}^m(f, \varphi) = \max_{i=0,1,\dots,m} |f(x_i) - \varphi(x_i)|$$

a jako diskrétní obdobu (1.5.2) můžeme užít

$$(1.7.2) \quad \varrho_p^m(f, \varphi) = \left(\sum_{i=0}^m |f(x_i) - \varphi(x_i)|^p \right)^{1/p}, \quad p \geq 1.$$

Při vědeckotechnických výpočtech se v diskrétním případě velmi často chyba aproximace měří pomocí

$$(1.7.3) \quad \varrho_p^m(f, \varphi) = \left(\sum_{i=0}^m |f(x_i) - \varphi(x_i)|^2 \right)^{1/2}$$

nebo

$$(1.7.4) \quad \varrho_{2,\omega}^m(f, \varphi) = \left(\sum_{i=0}^m |f(x_i) - \varphi(x_i)|^2 \omega_i \right)^{1/2},$$

kde váhy ω_i jsou nějaká pevně zvolená kladná čísla.

Nejlepším aproximacím říkáme v diskrétním případě *nejlepší diskrétní aproximace*. Nejlepší diskrétní L_2 -aproximaci se také říká *aproximace metodou nejmenších čtverců* [srov. (1.7.3) nebo (1.7.4)].

Příkladem měřítka chyby aproximace, které je poněkud jiného typu, je veličina

$$(1.7.5) \quad \varrho_T^n(f, \varphi) = \sum_{i=0}^m |f^{(j)}(x_0) - \varphi^{(j)}(x_0)|, \quad x_0 \in \langle a, b \rangle,$$

kde $f^{(j)}$ a $\varphi^{(j)}$ znamená j -tou derivaci funkcí f a φ . Měříme-li chybu aproximace podle (1.7.5) a hledáme-li nejlepší aproximaci ve třídě polynomů n -tého stupně [n stejně jako v (1.7.5)], je touto aproximací Taylorův polynom funkce f známý z matematické analýzy (viz čl. 2).

1.8 Příklad.

Vyšetřujme stejně jako v příkl. 1.6 aproximaci funkce $f(x) = \sqrt{x}$ na intervalu $\langle 0, 1 \rangle$ polynomem prvního stupně. Funkce f nechť je nyní zadána tabulkou.

Máme-li funkci f zadánu tabulkou hodnot v bodech $x_0 = 0$ a $x_1 = 1$ [$f(0) = 0$, $f(1) = 1$], má nejlepší diskrétní L_2 -aproximace φ_L^* tvar $\varphi_L^*(x) = x$. Ze je φ_L^* nejlepší aproximace, je ihned vidět z toho, že platí

$$\varrho_2^1(f, \varphi_L^*) = \left(\sum_{i=0}^1 |f(x_i) - x_i|^2 \right)^{1/2} = 0.$$

Všimněme-li si ještě toho, že o velikosti chyby aproximace E mimo body x_0 a x_1 nám nulová hodnota ϱ_2^1 nic neříká.

Bude-li funkce f zadána tabulkou hodnot ve třech bodech $x_0 = 0$, $x_1 = \frac{1}{2}$, $x_2 = 1$, bude mít nejlepší diskrétní L_2 -aproximace φ_L^* tvar $\varphi_L^*(x) = x + \frac{1}{6}(\sqrt{2} - 1)$ (viz příkl. 6.6.2); nyní již bude $\varrho_2^2(f, \varphi_L) > 0$.

Aproximujeme-li f Taylorovým polynomem, sestaveným z hodnot $f(\frac{1}{2}) = (\sqrt{2})/2$ a $f'(\frac{1}{2}) = (\sqrt{2})/2$, dostaneme aproximaci φ_T^* tvaru

$$\varphi_T^* = [(\sqrt{2})/2](x - \frac{1}{2}) + (\sqrt{2})/2,$$

pro kterou platí $\varrho_T^1(f, \varphi_T^*) = 0$.

1.9 Polynomy jako aproximující funkce.

Jako aproximace funkcí se v numerické matematice velmi často používají polynomy. Zavedeme si proto zvláštní označení pro třídy polynomů, které budeme používat v celém sešitě. Třídu \mathcal{V}_n aproximujících funkcí (viz odst. 1.3), která se skládá ze všech polynomů nejvýše n -tého stupně, budeme nadále značit \mathcal{P}_n .

Důvodů pro používání polynomiálních aproximací je celá řada [26]. Patří sem především to, že se polynomy dají plně popsat konečným počtem údajů (stupeň, koeficienty) a že se jejich hodnoty dají bez problémů počítat konečným počtem aritmetických operací. Jak jsme již uvedli v odst. 1.1, počítáme při užití jiných základních funkcí než polynomiálních hodnoty základních funkcí $\varphi_0, \varphi_1, \dots, \varphi_n$ stejně pomocí aproximace polynomem nebo racionální funkcí. Dalším důvodem pro užívání polynomů je to, že se s nimi snadno matematicky pracuje (derivování, integrování).

Všechny tyto výhody třídy \mathcal{P}_n by však neměly význam, kdybychom nevěděli, že pomocí tříd \mathcal{P}_n můžeme spojitou funkci aproximovat s libovolně vysokou přesností. Toto tvrzení je obsahem věty, jejíž důkaz podal K. Weierstrass (viz [26]).

1.10 Věta.

Budiž f funkce spojitá na intervalu $\langle a, b \rangle$. Pak ke každému $\varepsilon > 0$ existuje přirozené číslo n a polynom $\varphi \in \mathcal{P}_n$ n -tého stupně takový, že

$$\varrho_{\max}(f, \varphi) = \max_{x \in \langle a, b \rangle} |f(x) - \varphi(x)| < \varepsilon.$$

Zcela analogická věta platí, měříme-li chybu aproximace pomocí $\varrho_2(f, \varphi)$.

1.11 Poznámka.

Mnohé výsledky uvedené v této kapitole pro aproximace pomocí polynomů si zachovávají platnost i pro jiné třídy aproximací lineárního typu. Tak např. věta 1.10 i analogická věta pro $\varrho_2(f, \varphi)$ platí také pro aproximaci periodické spojitě funkce s periodou 2π pomocí trigonometrických polynomů.

2 Aproximace Taylorovým polynomem

2.1 Taylorův polynom.

Budiž daná funkce f , která má v daném bodě $x_0 \in \langle a, b \rangle$ alespoň n derivací. Předpokládejme, že známe hodnoty $f^{(j)}(x_0)$, $j = 0, 1, \dots, n$, ($f^{(0)} \equiv f$) a že chceme najít aproximaci

φ funkce f takovou, která "co nejlépe" napodobuje chování funkce f v okolí bodu x_0 . Vzhledem ke způsobu zadání funkce f je přirozené tento požadavek matematicky vyjádřit takto: Hledáme aproximaci φ takovou, aby platilo

$$(2.1.1) \quad \varphi^{(j)}(x_0) = f^{(j)}(x_0), \quad j = 0, 1, \dots, n,$$

nebo aby alespoň byla veličina $\varrho_T^n(f, \varphi)$ z (1.7.5) so nejmenší.

Hledejme φ ve třídě \mathcal{P}_n polynomů nejvýše n -tého stupně a zapišme φ ve tvaru

$$(2.1.2) \quad \varphi(x) = c_0 + c_1(x - x_0) + \dots + c_n(x - x_0)^n$$

(tímto způsobem můžeme při daném x_0 zapsat každý polynom z \mathcal{P}_n). Spočítáme-li j -tou derivaci funkce φ v bodě x_0 , dostaneme $\varphi^{(j)}(x_0) = j!c_j$. Existuje tedy právě jeden polynom $p_n \in \mathcal{P}_n$, pro který platí (2.1.1) neboli $\varrho_T^n(f, \varphi) = 0$. Je to polynom tvaru (2.1.2) s koeficienty $c_j = f^{(j)}(x_0)/j!$, $j = 0, 1, \dots, n$, neboli polynom

$$(2.1.3) \quad p_n(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

Tento polynom známe z matematické analýzy pod názvem *Taylorův polynom n -tého stupně pro funkci f v bodě x_0* . V příkladu 1.8 jsme pro $f(x) = \sqrt{x}$ uvedli Taylorův polynom prvního stupně ($n = 1$) v bodě $\frac{1}{2}$ ($x_0 = \frac{1}{2}$).

2.2 Příklad.

Sestrojme Taylorův polynom čtvrtého stupně pro funkci $f(x) = \sin x$ v bodě $x_0 = 0$. Z matematické analýzy víme, že hledaný polynom je

$$p_4(x) = x - \frac{x^3}{3!}.$$

Zvolíme-li $x_0 = \pi/4$, dostaneme polynom

$$\hat{p}_4(x) = \frac{\sqrt{2}}{2} \left[1 + (x - \pi/4) - \frac{(x - \pi/4)^2}{2!} - \frac{(x - \pi/4)^3}{3!} + \frac{(x - \pi/4)^4}{4!} \right].$$

Všimněme si toho, že je $p_4(0) = 0$, $\hat{p}_4(0) \doteq 0,00196$, $\hat{p}_4(\pi/4) = (\sqrt{2})/2 = \sin(\pi/4) \doteq 0,70711$, $p_4(\pi/4) \doteq 0,70465$ (vysvětlete rozdíly).

2.3 Chyba aproximace.

Z konstrukce Taylorova polynomu vyplývá, že chyba aproximace $E(x) = f(x) - p_n(x)$ je spolu se svými prvními n derivacemi v bodě x_0 rovna nule. Ze samotného tohoto faktu však nemůžeme ještě usuzovat na chování funkce E v bodech $x \neq x_0$. Průběh funkce E na intervalu $\langle a, b \rangle$ budeme moci posoudit teprve tehdy, budeme-li mít podrobnější informaci o chování funkce f a jejích derivací na celém intervalu $\langle a, b \rangle$. Důkaz následující věty o chybě aproximace lze nalézt v učebnicích základů matematické analýzy.

2.4 Věta.

Nechť funkce $f, f', \dots, f^{(n)}$ existují a jsou spojité na intervalu $\langle a, b \rangle$ a necht' pro všechna $x \in \langle a, b \rangle$ existuje derivace $f^{(n+1)}(x)$. Necht' p_n je Taylorův polynom n -tého stupně pro funkci f v bodě x_0 , kde $x_0 \in \langle a, b \rangle$. Pak pro chybu aproximace E platí na intervalu $\langle a, b \rangle$ vzorec

$$(2.4.1) \quad E(x) = f(x) - p_n(x) = \frac{(x - x_0)^{n+1}}{(n + 1)!} f^{(n+1)}(\xi),$$

kde ξ je nějaké (blíže neurčené) číslo, které leží mezi x a x_0 a závisí na intervalu na hodnotě x .

2.5 Odhad chyby aproximace.

Vzhledem k tomu, že hodnotu ξ ve (2.4.1) pro dané x neznáme, dává (2.4.1) jen kvalitativní informaci o průběhu chyby aproximace E na intervalu $\langle a, b \rangle$. Vidíme, že pokud je n a x_0 pevné a $f^{(n+1)}(x)$ se na $\langle a, b \rangle$ příliš nemění, roste hodnota chyby aproximace - vzdalujeme-li se s bodem x od bodu x_0 - asi jako $C(x - x_0)^{n+1}$, kde $C \approx \text{konst}$.

Početně se dá využít snadný důsledek věty 2.4, který udává *odhad chyby aproximace*. Umíme-li totiž stanovit číslo M takové, že $|f^{(n+1)}(x)| \leq M$ pro všechna $x \in \langle a, b \rangle$, pak ze (2.4.1) ihned plyne nerovnost

$$(2.5.1) \quad |E(x)| \leq \frac{M}{(n + 1)!} |x - x_0|^{n+1}.$$

Úloha odhadnout chybu aproximace se tak převádí na úlohu odhadnout $n + 1$ -ní derivaci aproximované funkce.

2.6 Příklad.

Odhadneme chybu aproximace polynomy p_4 a \hat{p}_4 z příkl. 2.2. Protože $|f^{(5)}(x)| = |\cos x| \leq 1$ pro všechna x , dostáváme z (2.5.1)

$$|f(x) - p_4(x)| \leq \frac{1}{120} |x|^5$$

a

$$|f(x) - \hat{p}_4(x)| \leq \frac{1}{120} \left| x - \frac{\pi}{4} \right|^5$$

(porovnejte hodnotu skutečné chyby s odhadem v bodech $x = 0$ a $x = \pi/4$).

2.7 Řád chyby aproximace.

Položíme-li $x = x_0 + h$, můžeme Taylorův polynom p_n zapsat ve tvaru

$$(2.7.1) \quad p_n(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \dots + \frac{h^n}{n!}f^{(n)}(x_0).$$

Podle vzorce (2.7.1) můžeme na základě znalosti hodnot funkce f a jejich derivací v bodě x_0 počítat přibližné hodnoty dané funkce v okolí tohoto bodu. Uvážíme-li, že platí (2.4.1), ihned vidíme, že

$$(2.7.2) \quad f(x_0 + h) = p_n(x_0 + h) + \frac{h^{n+1}}{(n+1)!}f^{(n+1)}(\xi).$$

V dalším výkladu budeme příležitostně používat užitečné symboly $O(h^k)$ apod. (viz též [21]), přičemž zápis $f(x) = O(g(x))$ (při $x \rightarrow x_0$) bude znamenat, že v okolí bodu x_0 je funkce $|f(x)/g(x)|$ ohraničená (x_0 může být i $\pm \infty$). Přesněji (pro x_0 konečně): Existuje $\delta > 0$ a reálné číslo C tak, že $|x - x_0| < \delta \Rightarrow |f(x)| \leq C|g(x)|$. Vztahy (2.7.2) a (2.7.1) tedy dávají pro $h \rightarrow 0$ a funkci f , jejíž $(n+1)$ -ní derivace je v okolí bodu x_0 ohraničená,

$$(2.7.3) \quad f(x_0 + h) = f(x_0) + hf'(x_0) + \dots + \frac{h^n}{n!}f^{(n)}(x_0) + O(h^{n+1});$$

podobně píšeme (zde pro $x \rightarrow x_0$)

$$(2.7.4) \quad f(x) = p_n(x) + O((x - x_0)^{n+1}).$$

Platí-li (2.7.3) a (2.7.4), říkáme, že chyba aproximace je řádu h^{n+1} nebo $(x-x_0)^{n+1}$. V tomto symbolickém vyjádření je uchována podstatná část informace o chování chyby aproximace. Poznamenáváme ještě, že vztahy (2.7.3) a (2.7.4) se často užívají k odvozování různých přibližných vzorců (viz úlohu 2.9.4).

2.8 Chyba aproximace a celková chyba.

Chyba aproximace $E(x)$ funkční hodnoty $f(x)$ hodnotou polynomu $p_n(x)$ nevystihuje (nejen u Taylorových polynomů, ale obecně) všechny nepřesnosti, kterých se při přibližném výpočtu hodnoty $f(x)$ dopouštíme. Udává pouze chybu metody (ve smyslu [21]) vzniklou tím, že jsme matematickou úlohu [výpočet $f(x)$] nahradili jejím jistým modelem [výpočet $p_n(x)$]. Nejsou zde zahrnuty případně vlivy chyb ve vstupních datech [např. u Taylorova polynomu nepřesnosti ve stanovení $f^{(j)}(x_0)$], ani vliv zaokrouhlovacích chyb při výpočtu hodnoty aproximace $p_n(x)$ na počítači. Studium podobných otázek rozsah tohoto sešitu nedovoluje; případného zájemce odkazujeme na učebnice numerické matematiky a monografii [31].

Při zkoumání *celkové chyby*, které se při přibližném výpočtu hodnoty $f(x)$ dopouštíme, můžeme k posouzení citlivosti výstupních dat [$f(x) \approx p_n(x)$] na chyby ve vstupních datech [hodnoty $f^{(j)}(x_0)$] použít *metodu experimentálních perturbací*: Provedeme výpočet vícekrát s porušenými vstupními daty a sledujeme vztah mezi změnami (perturbacemi) vstupních dat a změnami ve výstupních datech. Umělé poruchy ve vstupních datech volíme tak, aby jejich velikost řádově odpovídala nepřesnostem, které v nich předpokládáme. Konkrétní číselné hodnoty poruch je vhodné vybírat náhodně (pomocí tabulky náhodných čísel nebo programu pro generování náhodných čísel na počítači).

2.9 Úlohy.

2.9.1

Sestrojte Taylorovy polynomy čtvrtého stupně v bodě x_0 pro tyto funkce: \sqrt{x} , $\cos x$, $(1-x)^{-1}$, $(1+x)^{1/2}$, $\ln(1+x)$, $\ln(1-x)$.

[Pro \sqrt{x} neexistuje; $\cos x \approx 1 - x^2/2 + x^4/24$; ...; $\ln(1-x) \approx -x - x^2/2 - x^3/3 - x^4/4$.]

2.9.2

Je-li p_n Taylorův polynom n -tého stupně pro funkci e^x v bodě $x_0 = 0$, užití (2.5.1) a stanovte nejmenší n takové, že platí

$$\max_{0 \leq x \leq 1} |e^x - p_n(x)| \leq 10^{-6}.$$

[$n = 9$]

2.9.3

V jak velkém okolí bodu $x_0 = 0$ dávají dále uvedené aproximace zaručeně funkční hodnotu aproximovaných funkcí s přesností na 2D: a) $\sin x \approx x$. b) $\cos x \approx 1$?

[Návod: Ověřte, že jde o aproximace Taylorovými polynomy. Přesnost na 2D (absolutní chyba nejvýše $\frac{1}{2} \cdot 10^{-2}$) je zaručena v obou případech pro $|x| \leq 0,1$ při $M = 1$ ve (2.5.1). Ve skutečnosti platí vztah a) na 2D pro $|x| \leq 0,31$; odhad $|\sin x| \leq 1$ je v okolí $x = 0$ příliš pesimistický.]

2.9.4

Ukažte, že v okolí bodu $x = 0$ platí: a) $(1+x)^{1/2} = 1 + \frac{1}{2}x + O(x^2)$, b) $e^x = 1 + x + O(x^2)$.
[Návod: Aproximujte funkce Taylorovým polynomem.]

2.9.5

Ukažte, že pro dostatečně hladkou funkci f platí pro malá h

$$\frac{f(x+h) - f(x)}{h} = f'(x) + O(h).$$

Co přesně musíme předpokládat o funkci f

[Návod: Vyjádřete $f(x+h)$ pomocí Taylorova polynomu prvního stupně v bodě x . Užijte větu 2.4.]

2.9.6

Rozhodněte, zda je možné splnit (2.1.1) volbou vhodné polynomiální aproximace $\varphi \in \mathcal{P}_N$, kde $N < n$.

[Obecně ne.]

3 Aproximace interpolačním polynomem

3.1 Obecně o interpolační aproximaci.

V odstavci 2.1 jsme aproximovali Taylorovým polynomem n -tého stupně funkci zadanou $n+1$ hodnotami $f(x), f'(x), \dots, f^{(n)}(x)$ v jediném bodě $x = x_0$. Předpokládejme nyní, že je funkce f dána pouze svou hodnotou, ale v $n+1$ bodech x_0, x_1, \dots, x_n , a že chceme najít aproximaci φ takovou, která napodobuje chování funkce f v okolí všech těchto bodů. Typickým příkladem takového způsobu zadání funkce f jsou funkce zadané tabulkou, ať už tato tabulka vznikla jako výsledek nějakého měření či jde o tabulku hodnot některé standardní funkce získanou matematickými výpočty. O tabulkových bodech $x_i, i = 0, 1, \dots, n$, v nichž jsou zadány hodnoty $f(x_i)$, budeme v celém čl. 3 předpokládat, že jsou navzájem různé.

O *interpolaaci* hovoříme tehdy, je-li naším úkolem stanovit hodnotu funkce f dané tabulkou pro nějaké \hat{x} ležící mezi dvěma tabulkovými body (tj. $\hat{x} \neq x_i, i = 0, 1, \dots, n$). S touto úlohou se čtenář setkal již na střední škole, kde prováděl interpolaaci např. v tabulkách logaritmů. Při interpolování aproximujeme určitým způsobem funkci f vhodně vybranou (viz dále) funkcí f a klademe potom $f(\hat{x}) \approx \varphi(\hat{x})$.

Postupujeme tak, že nejprve stanovíme - s přihlédnutím k předpokládaným vlastnostem funkce f - třídu \mathcal{F} aproximujících funkcí, v níž budeme aproximaci φ hledat (např. φ bude polynom nejvýše n -tého stupně, $\mathcal{F} = \mathcal{P}_n$). Konkrétní aproximující funkci φ z této třídy vybereme na základě vstupních dat naší úlohy - funkčních hodnot $f(x_i), i = 0, 1, \dots, n$. V článku 2 jsme aproximující funkci (Taylorův polynom) vybírali tak, že platilo (2.1.1), tj.

$\varphi^{(j)}(x_0) = f^{(j)}(x_0)$, $j = 0, 1, \dots, n$. V souladu se změněnou situací se při interpolaci snažíme volit funkci φ tak, aby platilo

$$(3.1.1) \quad \varphi(x_i) = f(x_i), \quad i = 0, 1, \dots, n.$$

tj. aby přesně nabývala zadaných funkčních hodnot. Podmínkám (3.1.1) bude-me říkat *interpoláční podmínky*.

Z matematického hlediska zde okamžitě vyvstává otázka existence a jednoznačnosti (v dané třídě \mathcal{F}) funkce φ splňující vztahy (3.1.1). Pro aproximace lineárního typu je tato problematika podrobně zpracována v řadě učebnic numerické matematiky (např. [3], [26]). Poznáváme, že funkce φ splňující (3.1.1) (*interpoláční aproximace*) minimalizuje ve třídě \mathcal{F} veličinu

$$\varrho_2^n(f, \varphi) = \left(\sum_{i=0}^n |f(x_i) - \varphi(x_i)|^2 \right)^{1/2};$$

platí pro ni totiž (pokud existuje) $\varrho_2^n(f, \varphi) = 0$. Jde tedy o speciální nejlepší diskrétní L_2 -aproximace pro $n = m$.

V příkladu 1.8 jsme vlastně pro $n = 1$, $x_0 = 0$, $x_1 = 1$ uvedli interpoláční aproximaci funkce \sqrt{x} . Skutečně, nejlepší diskrétní L_2 -aproximaci $\varphi_L^*(x) = x$ z příkl. 1.8 splňuje interpoláční podmínky $x_i = \sqrt{x_i}$, $i = 0, 1$ a je tedy současně interpoláční aproximací ze třídy \mathcal{P}_1 .

Strategie založená na přesném splnění interpoláčních podmínek je při aproximování funkcí zadaných tabulkou oprávněna jen v těch případech, kdy jsou vstupní data (funkční hodnoty) udána s velkou přesností. Zpracováváme-li výsledky měření, je často účelnější nepožadovat přesný souhlas $f(x)$ a $\varphi(x)$ v tabulkových bodech, ale použít strategii jinou (viz metoda nejmenších čtverců, čl. 5).

3.2 Poznámky.

3.2.1

Jak uvidíme v kap. II a III, užití interpoláčních aproximací se zdaleka neomezuje na výpočet hodnot funkcí zadaných tabulkou nebo zpracování výsledků měření. Interpoláční aproximace jsou také základem mnoha efektivních numerických metod pro výpočet integrálů a řešení diferenciálních rovnic.

3.2.2

Úlohu o interpolaci lze dále zobecnit. Jsou-li v některých tabulkových bodech udány rovněž hodnoty derivací funkce f , požaduje se kromě (3.1.1) také splnění podmínek typu $\varphi^{(k)}(x_i) = f^{(k)}(x_i)$. Takové interpolaci se říká *Hermitova interpolace* její speciální případ je vlastně aproximace Taylorovým polynomem. Studium Hermitovy interpolace - jakkoli

je pro inženýra mnohdy užitečná - přesahuje rámec tohoto sešitu. Čtenáře odkazujeme na podrobnější učebnice numerické matematiky, např. [3], [26].

3.3 Lagrangeův interpolační polynom.

Protože interpolační podmínky (3.1.1) představují stejně jako (2.1.1) $n + 1$ podmínek na aproximující funkci, lze očekávat, že jednou z vhodných tříd aproximujících funkcí bude třída \mathcal{P}_n polynomů nejvýše n -tého stupně (takové polynomy mají $n + 1$ koeficientů).

Ukážeme, že pro každou funkci f a každé n skutečně existuje právě jeden polynom $\varphi \in \mathcal{P}_n$, který splňuje podmínky (3.1.1). Tomuto polynomu budeme říkat *interpolační polynom* pro funkci f v bodech x_0, x_1, \dots, x_n a budeme jej značit L_n . Tabulkovým bodům x_0, x_1, \dots, x_n budeme říkat také *uzly interpolace*.

Snadno se ověří (dosaďte $x = x_i$), že interpolační podmínky (3.1.1) splňuje polynom

$$(3.3.1) \quad L_n(x) = \sum_{i=0}^n f(x_i)l_i(x),$$

kde l_i jsou polynomy n -tého stupně takové, že platí

$$(3.3.2) \quad \begin{aligned} l_i(x_k) &= 0 & \text{pro } k \neq i, \\ &= 1 & \text{pro } k = i, \quad i, k = 0, 1, \dots, n. \end{aligned}$$

Polynom l_i má tedy kořeny $x_0, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ a můžeme jej zapsat ve tvaru

$$(3.3.3) \quad l_i(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)}$$

(připomínáme předpoklad, že x_i , $i = 0, 1, \dots, n$, jsou navzájem různé). Interpolačnímu polynomu L_n ve tvaru (3.3.1), kde l_i je dáno vztahem (3.3.3), se říká *Lagrangeův interpolační polynom*. Stupeň polynomu L_n je zřejmě nejvýše n .

Ukázali jsme, že interpolační polynom nejvýše n -tého stupně pro funkci zadanou hodnotami v $n + 1$ navzájem různých bodech skutečně existuje (tím, že jsme jej sestrojili). Dokážeme nyní, že existuje jen jeden takový polynom. Uvažujme libovolný polynom $p_n \in \mathcal{P}_n$, který splňuje interpolační podmínky, tj. $p_n(x_i) = f(x_i)$, $i = 0, 1, \dots, n$. Funkce $p_n - L_n$ je opět polynom nejvýše n -tého stupně $p_n - L_n \in \mathcal{P}_n$, a tento polynom se rovná nule ve všech $n + 1$ bodech x_i , $i = 0, 1, \dots, n$. Ale polynom nejvýše n -tého stupně nemůže mít více než n kořenů, pokud se nerovná identicky nule. V našem případě tudíž nutně $p_n - L_n = 0$, $p_n = L_n$; jednoznačnost interpolačního polynomu je dokázána.

3.4 Příklad.

Nechť je funkce f dána svými hodnotami v bodech ($n = 3$) $x_0 = -1$, $x_1 = 1$, $x_2 = -1$, $x_3 = 3$ a nech $f(0) = 1$, $f(1) = 2$, $f(-1) = 2$, $f(3) = 0$. Vypočítáme přibližnou hodnotu $f(2)$ pomocí Lagrangeova interpolačního polynomu L_3 .

Máme

$$l_0(x) = \frac{(x-1)(x+1)(x-3)}{(0-1)(0+1)(0-3)} = \frac{1}{3}(x-1)(x+1)(x-3),$$

a dále podobně obdržíme

$$l_1(x) = -\frac{1}{4}x(x+1)(x-3),$$

$$l_2(x) = -\frac{1}{8}x(x-1)(x-3),$$

$$l_3(x) = \frac{1}{24}x(x-1)(x+1).$$

Je $l_0(2) = -1$, $l_1(2) = \frac{3}{2}$, $l_2(2) = \frac{1}{4}$, $l_3(2) = \frac{1}{4}$, a tedy

$$L_3(2) = 1 \cdot (-1) + 2 \cdot \frac{3}{2} + 2 \cdot \frac{1}{4} + 0 \cdot \frac{1}{4} = \frac{5}{2} \approx f(2).$$

Samotný polynom L_3 má tvar

$$\begin{aligned} L_3(x) &= 1 \cdot l_0(x) + 2 \cdot l_1(x) + 2 \cdot l_2(x) + 0 \cdot l_3(x) = \\ &= \frac{1}{3}(x-1)(x+1)(x-3) - \frac{1}{2}x(x+1)(x-3) - \frac{1}{4}x(x-1)(x-3) = \\ &= -\frac{5}{12}x^3 + x^2 + \frac{5}{12}x + 1. \end{aligned}$$

3.5 Lineární a kvadratická interpolace.

Interpolujeme-li funkci f na základě jejích hodnot ve dvou bodech, aproximujeme ji polynomem prvního stupně ($n = 1$) a hovoříme o *lineární interpolaci*. Interpolační polynom prvního stupně L_1 má tvar

$$(3.5.1) \quad L_1(x) = \frac{x-x_1}{x_0-x_1}f(x_0) + \frac{x-x_0}{x_1-x_0}f(x_1).$$

Právě lineární interpolace se běžně používá při práci s tabulkami funkcí ve škole

Je-li $n = 2$, hovoříme o *kvadratické interpolaci*. Interpolační polynom L_2 druhého stupně má tvar

$$(3.5.2) \quad \begin{aligned} L_2(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \\ &+ \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2). \end{aligned}$$

3.6 Příklad.

Máme dānu tabulku hodnot funkce $f(x) = \sin x$ na 5D (tab. 1). Vypočítáme pŕibližnou hodnotu $\sin 20^\circ 18'$ lineární interpolací z funkčních hodnot v bodech $x_0 = 20^\circ$ a $x_1 = 21^\circ$.

Tabulka 1

x	20°	21°	22°
$\sin x$	0,342 02	0,358 37	0,374 61

Podle (3.5.1) máme

$$\sin 20^\circ 18' \approx 0,342\,02 \frac{20,3 - 21}{20 - 21} + 0,358\,37 \frac{20,3 - 20}{21 - 20} \doteq 0,346\,93.$$

Pŕesná hodnota $\sin 20^\circ$ zaokrouhlená na 5D je 0,346 94. V tomto pŕípade je tedy lineární interpolace zcela postačující, bylo by zbytečně pracné volit $n > 1$.

3.7 Vliv nepřesnosti ve vstupních datech.

Ze (3.3.1) můžeme snadno odhadnout, jaký vliv mohou mít nepřesnosti ve vstupních datech na hodnotu interpolačního polynomu v bodě x . Předpokládejme, že všechny hodnoty $f(x_i)$ jsou zatíženy chybou, jejíž absolutní hodnota nepřevyší ε . Označme \tilde{L}_n interpolační polynom, který odpovídá hodnotám $\tilde{f}(x_i) = f(x_i) + \varepsilon_i$, $|\varepsilon_i| \leq \varepsilon$. Potom

$$(3.7.1) \quad |\tilde{L}_n(x) - L_n(x)| = \left| \sum_{i=0}^n [\tilde{f}(x_i) - f(x_i)] l_i(x) \right| \leq \\ \leq \sum_{i=0}^n |\tilde{f}(x_i) - f(x_i)| |l_i(x)| \leq \varepsilon \sum_{i=0}^n |l_i(x)|.$$

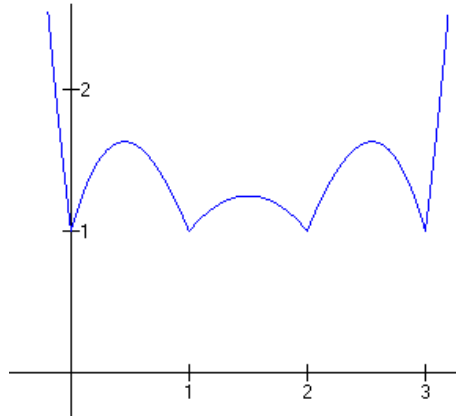
Vliv chyb ve vstupních datech na hodnotu interpolačního polynomu souvisí tedy s chováním funkce

$$F(x) = \sum_{i=0}^n |l_i(x)|;$$

tato funkce závisí pouze na hodnotách x_i , $i = 0, 1, \dots, n$, a volbou tabulkových bodů je proto možno vliv chyb ve vstupních datech na hodnotu interpolačního polynomu redukovat.

3.8 Příklad.

Pro $n = 3$ a $x_i = i$, $i = 0, 1, 2, 3$, je pŕůběh funkce F na obr. 1. Je vidět, že chyby ve vstupních datech se budou silněji projevovat na kraji intervalu vytvářeného body x_i ; mimo tento interval může $L_n(x)$ reagovat na chyby ve vstupních datech již velmi citlivě.



Obrázek 1:

3.9 Chyba aproximace interpolačním polynomem.

Z toho, že při interpolaci pro chybu aproximace $E = f - L_n$ platí $E(x_i) = 0$, $i = 0, 1, \dots, n$ [protože interpolační polynom L_n splňuje (3.1.1)] nemůžeme ještě usuzovat na hodnotu chyby aproximace mimo tabulkové body. Žádná tabulka zřejmě nedefinuje funkci obecně jednoznačně a pro $x \neq x_i$ se daná funkce může chovat zcela libovolně. Čtenář by tedy neměl být překvapen tím, že (podobně jako u Taylorova polynomu) průběh chyby aproximace na intervalu $\langle a, b \rangle$ budeme moci posoudit teprve na základě dalších informací o vlastnostech funkce f , např. informace o tom, že f je na $\langle a, b \rangle$ dostatečně "hladká", tj. má tam určitý počet derivací.

V dalším výkladu budeme symbolem $\text{int}(x_0, x_1, \dots, x_n, x)$ označovat nejmenší otevřený interval takový, že body x_0, x_1, \dots, x_n, x leží uvnitř tohoto intervalu nebo na jeho hranici. Důkaz následující věty o chybě aproximace interpolačním polynomem najde čtenář v učebnicích numerické matematiky (např. [3], [26]).

3.10 Věta.

Nechť $\langle a, b \rangle$ je libovolný interval, který obsahuje všech $n+1$ bodů x_0, x_1, \dots, x_n . Nechť $f, f', \dots, f^{(n)}$ existují a jsou spojité na $\langle a, b \rangle$ a nechť pro všechna $x \in \langle a, b \rangle$ existuje derivace $f^{(n+1)}(x)$. Nechť L_n je interpolační polynom pro funkci f a uzly x_0, x_1, \dots, x_n . Pak pro chybu aproximace funkce f polynomem L_n platí na intervalu $\langle a, b \rangle$ vzorec

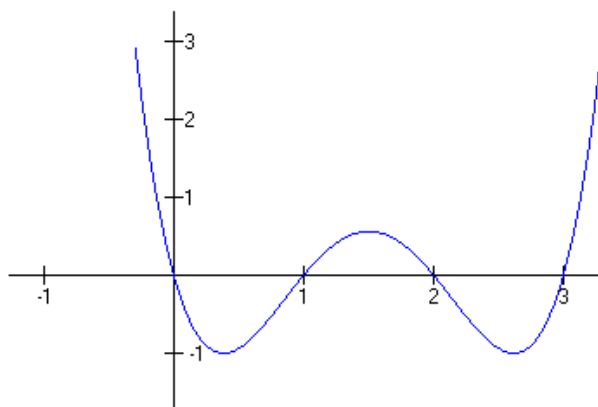
$$(3.10.1) \quad E(x) = f(x) - L_n(x) = (x - x_0)(x - x_1) \dots (x - x_n) \frac{f^{(n+1)}(\xi)}{(n+1)!},$$

kde ξ je nějaké (blíže neurčené) číslo z intervalu $\text{int}(x_0, x_1, \dots, x_n, x)$, které závisí na hodnotě x .

Vidíme, že pokud se $f^{(n+1)}(x)$ na $\langle a, b \rangle$ příliš nemění, je pro průběh chyby aproximace rozhodující průběh polynomu $e_n(x) = (x - x_0)(x - x_1) \dots (x - x_n)$. Tento polynom (stejně jako funkce F v odst. 3.7) nezávisí na interpolované funkci, ale pouze na bodech x_i . Vhodnou volbou uzlů x_i , $i = 0, 1, \dots, n$, můžeme tedy též zmenšit chybu interpolace.

3.11 Příklad.

Průběh funkce e_n v případě $n = 3$, $x_i = i$, $i = 0, 1, 2, 3$ je znázorněn na obr. 2. Opět vidíme, že se dá čekat, že vně intervalu $\text{int}(x_0, x_1, \dots, x_n)$ chyba aproximace interpolačním polynomem prudce poroste.



Obrázek 2:

3.12 Odhad chyby aproximace.

Vzorec (3.10.1) pro chybu interpolace polynomem má podobný tvar jako vzorec (2.4.1) pro chybu aproximace Taylorovým polynomem. Všimněme si také toho, že $e_n(x)$ ve všech uzlech interpolace.

Ze (3.10.1) je také ihned vidět, že je-li aproximovaná funkce f sama polynom nejvýše n -tého stupně, aproximuje ji interpolační polynom n -tého stupně s nulovou chybou, a je pak tedy $L_n = f$ [pro polynom $f \in \mathcal{P}_n$ je totiž $f^{(n+1)}(x) \equiv 0$].

Pro praktické stanovení chyby interpolace v obecném případě však ani zde - stejně jako tomu bylo u Taylorova polynomu - není vztah (3.10.1) použitelný (neznáme ξ). Ze (3.10.1) ale opět plyne odhad chyby aproximace: Existuje-li konstanta M taková, že pro všechna $x \in (a, b)$ je $|f^{(n+1)}(x)| \leq M$, pak

$$(3.12.1) \quad |E(x)| \leq \frac{M}{(n+1)!} |e_n(x)| \leq \frac{M}{(n+1)!} \max_{x \in (a,b)} |e_n(x)|$$

pro všechna $x \in \langle a, b \rangle$.

3.13 Příklad.

Odhadneme chybu aproximace, které jsme dopustili při lineární interpolaci v příkl. 3.6.

Je $n = 1$, $f''(x) = -\sin x$. Tento vztah je ale odvozen za předpokladu, že x měříme v obloukové míře, a proto musíme celý odhad provést na intervalu $\langle \pi/9, 7\pi/60 \rangle$, který v obloukové míře odpovídá intervalu $\langle 20^\circ, 21^\circ \rangle$. Na tomto intervalu máme $|f''(x)| = |\sin x| \leq 0,35837$ (na 5D), a tedy

$$|E(20^\circ 18')| \leq \frac{1}{2} \cdot 0,35837 \cdot \left(\frac{18}{60} \cdot \frac{\pi}{180}\right) \cdot \left(\frac{42}{60} \cdot \frac{\pi}{180}\right) \doteq 1,1 \cdot 10^{-5}.$$

Všimneme si ještě, že pokud bychom užili k odhadnutí druhé derivace pouze vztah $|f''(x)| \leq 1$, dostali bychom odhad chyby aproximace zbytečně pesimistický.

3.14 Extrapolace.

Používáme-li interpolační polynom k výpočtu přibližných hodnot interpolované funkce vně intervalu $\text{int}(x_0, x_1, \dots, x_n)$, říkáme, že provádíme *extrapolaci*. V odstavci 3.11 jsme upozornili na to, že chyba aproximace může být v tomto případě velká; totéž platí i o vlivu nepřesnosti ve vstupních datech na spočítanou přibližnou hodnotu $f(x)$. Pro x velmi blízká koncovým bodům intervalu $\text{int}(x_0, x_1, \dots, x_n)$ lze však přesto dostat dobré výsledky.

3.15 Příklad. Lineární extrapolace

Lineární interpolace z hodnot $\sin 20^\circ$ a $\sin 21^\circ$ (viz příkl. 3.6) dává podle vzorce (3.5.1) pro $\sin 21^\circ 18'$ přibližnou hodnotu 3,36328 s chybou asi $3 \cdot 10^{-5}$, což je zcela přijatelný výsledek. Lineární interpolace s uzly $x_0 = 21^\circ$, $x_1 = 22^\circ$ dává $\sin 21^\circ 18' \approx 0,36324$ s chybou přibližně $1 \cdot 10^{-5}$.

3.16 Nevillův algoritmus.

Tam, kde nás zajímá hodnota interpolačního polynomu jen v jednom bodě $x = \hat{x}$, je zbytečně počítat koeficienty tohoto polynomu. Ukážeme totiž, že hodnotu interpolačního polynomu L_n v bodě $x = \hat{x}$ můžeme spočítat přímo ze vstupních dat úlohy, tj. z čísel x_i , $f(x_i)$, $i = 0, 1, \dots, n$.

Uvažujme interpolační polynom L_{ik} k -tého stupně, který interpoluje funkci f v $k+1$ uzlech $x_{i-k}, x_{i-k+1}, \dots, x_i$ při nějakém daném i , $0 \leq i \leq n$, $i \geq k$. Pro účely tohoto odstavce bude výhodné značit P_{ik} , $i = 0, 1, \dots, n$, $k \leq i$, hodnoty polynomů L_{ik} v bodě $x = \hat{x}$, tj. klademe $P_{ik} = L_{ik}(\hat{x})$, $i = 0, 1, \dots, n$, $k \leq i$. Speciálně bude $P_{i0} = f(x_i)$, $i = 0, 1, \dots, n$, (interpolační polynom je v tomto případě konstanta) a $P_{nn} = L_n(\hat{x})$. Algoritmus pro výpočet P_{nn} , který nyní popíšeme, využívá vzájemný vztah hodnot P_{ik} , $P_{i,k-1}$ a $P_{i-1,k-1}$.

Ze vzorce (3.5.1) vidíme, že pro lineární interpolaci platí např.

$$P_{11} = \frac{(\hat{x} - x_0)f(x_1) - (\hat{x} - x_1)f(x_0)}{x_1 - x_0} = \frac{(\hat{x} - x_0)P_{10} - (\hat{x} - x_1)P_{00}}{x_1 - x_0}.$$

Obecně se dá ukázat (viz např. [3], [29]), že platí

$$(3.16.1) \quad P_{ik} = \frac{(\hat{x} - x_{i-k})P_{i,k-1} - (\hat{x} - x_i)P_{i-1,k-1}}{x_i - x_{i-k}}.$$

Poněkud upravený vzorec (3.16.1) tvoří základ *Nevillova algoritmu* pro výpočet $L_n(\hat{x})$

$$(3.16.2) \quad \text{Vstup : } n, \hat{x}, x_0, x_1, \dots, x_n, f(x_0), f(x_1), \dots, f(x_n).$$

$$\text{Pro } i = 0, 1, \dots, n :$$

$$P_{i0} = f(x_i).$$

$$\text{Pro } k = 1, 2, \dots, i :$$

$$P_{ik} = P_{i,k-1} + (\hat{x} - x_i) \frac{P_{i,k-1} + P_{i-1,k-1}}{x_i - x_{i-k}}.$$

$$\text{Výstup : } L_n(\hat{x}) = P_{nn}.$$

Mezi hodnoty P_{ik} , $i = 0, 1, \dots, n$, $k = 0, 1, \dots, i$ je při ručním výpočtu výhodné uspořádat do tzv. *Nevillova schématu* (tab. 2). Při výpočtu se postupuje po řádcích schématu. Šipky ukazují, z kterých hodnot v levém sloupci schématu se počítají hodnoty v sousedním pravém sloupci.

Tabulka 2

x	$f(x_0)$	$= P_{00}$				
		\searrow				
x_1	$f(x_1)$	$= P_{10}$	\longrightarrow	P_{11}		
\vdots		\vdots		\vdots	\ddots	
x_{n-1}	$f(x_{n-1})$	$= P_{n-1,0}$		$P_{n-1,1}$	\dots	$P_{n-1,n-1}$
		\searrow				\searrow
x_n	$f(x_n)$	$= P_{n0}$	\longrightarrow	P_{n1}	\dots	$P_{n,n-1} \longrightarrow P_{nn}$

3.17 Příklad.

Vypočítáme Nevillovým algoritmem hodnotu $L_3(2)$ pro funkci f z příkl. 3.4. Výpočet (Nevillovo schéma) uspořádáme do tabulky (tab. 3).

Tabulka 3

i	x_i	$\hat{x} - x_i$	$f(x_i) = P_{i0}$	$P_{i1}(k=1)$	$P_{i2}(k=2)$	$P_{i3}(k=3)$
0	0	2	1			
1	1	1	$P_{10} = 2$	3		
2	-1	3	$P_{20} = 2$	$P_{21} = 2$	5	
3	3	-1	0	$\frac{1}{2}$	$\frac{5}{4}$	$\frac{5}{2}$

Je např.

$$P_{21} = P_{20} + \frac{(\hat{x} - x_2)(P_{20} - P_{10})}{x_2 - x_1} = 2 + \frac{3 \cdot 0}{-2} = 2$$

a jako poslední hodnotu počítáme

$$P_{33} = P_{32} + \frac{(\hat{x} - x_3)(P_{32} - P_{22})}{x_3 - x_0} = \frac{5}{4} + \frac{(-1)(\frac{5}{4} - 5)}{3} = \frac{5}{2} = L_3(2) \approx f(2).$$

3.18 Použití Nevillova algoritmu.

Nevillův algoritmus je velmi efektivní způsob výpočtu hodnoty $L_n(x)$ pro dané $x - \hat{x}$. Pokud však chceme znát $L_n(x)$ pro více hodnot x , museli bychom pokaždé sestavovat Nevillovo schéma znovu. Efektivnější metoda pro takovou úlohu bude popsána v odst. 3.20 až 3.26.

Z odvození Nevillova algoritmu vyplývá, že uzly x_i nemusejí být uspořádány monotónně (viz též příkl. 3.17). Je dokonce výhodné uspořádat je podle rostoucí vzdálenosti od bodu \hat{x} , v němž chceme stanovit přibližnou hodnotu funkce f . Při výpočtu po řádcích Nevillova schématu pak podle rozdílů hodnot P_{ii} a $P_{i-1,i-1}$ na diagonále můžeme rozhodnout o tom, zda má smysl brát v úvahu ještě další tabulkový bod a interpolovat polynomem vyššího stupně (tj. počítat další řádek schématu).

3.19 Příklady.

Uvedeme dva příklady výpočtu Nevillovým algoritmem, na nichž chceme ilustrovat tvrzení předcházejícího odstavce. Výsledky uspořádáváme již pouze zkráceně podle tab. 2.

3.19.1

Užitím Nevillova algoritmu vypočítáme $f(0,44)$ pro funkci f , jejíž hodnoty jsou uvedeny v následujícím Nevillově schématu (výpočtu provádíme na 5D):

x_i	$f(x_i)$			
0,3	0,295 5			
0,4	0,389 4	0,426 96		
0,5	0,479 4	0,425 40	0,425 87	
0,6	0,564 6	0,428 28	0,425 98	0,425 91

Čísla na diagonále schématu (jsou tištěna tučně) mají ustálené chování, což nás vede k závěru, že 0,425 9 je dobrá aproximace hodnoty $f(0,44)$.

3.19.2

Užitím Nevillova algoritmu vypočítáme $f(1,8)$ pro funkci f , jejíž hodnoty jsou uvedeny v následujícím Nevillově schématu (výpočty provádíme s pěti platnými číslicemi):

x_i	$f(x_i)$				
2	0,135 34				
1	0,367 88	0,181 85			
3	0,049 787	0,240 64	0,170 09		
0	1,000 0	0,429 87	0,089 260	0,162 01	
0	0,018 316	0,558 24	0,275 83	0,139 01	0,164 31

Chování hodnot na diagonále schématu zde naznačuje, že vstupní data odpovídají funkci, kterou lze na větším intervalu špatně aproximovat polynomem. [Ve skutečnosti je $f(x) \doteq e^{-x}$; $e^{-1,8} \doteq 0,165 30$.] Podobný příklad viz úlohu 3.33.

3.20 Newtonův interpolační polynom.

Chceme-li počítat hodnoty interpolačního polynomu ve více bodech \hat{x} nebo pracovat s interpolačním polynomem v uzavřeném tvaru, je často účelné hledat interpolační polynom k datům $\{(x_i, f(x_i))\}$, $i = 0, 1, \dots, n$ ve tvaru

$$(3.20.1) \quad N_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_n(x - x_0)(x - x_1)\dots(x - x_{n-1}).$$

Interpolační polynom (3.20.1) se nazývá *Newtonův interpolační polynom*. Vzápětí uvidíme, že jsou-li uzly interpolace navzájem různá čísla, určují vstupní data jednoznačně koeficienty a_0, a_1, \dots, a_n takové, že N_n splňuje interpolační podmínky (3.1.1) neboli

$$(3.20.2) \quad N_n(x_i) = f(x_i), \quad i = 0, 1, \dots, n.$$

V odstavci 3.3 jsme ukázali, že podmínkami (3.1.1) a vstupními daty $\{(x_i, f(x_i))\}$, $i = 0, 1, \dots, n$ je interpolační polynom nejvýše n -tého stupně určen jednoznačně. Polynom N_n daný (3.20.1) je nejvýše n -tého stupně, a nebude tedy vlastně ničím jiným než novým způsobem zápisu interpolačního polynomu, který jsme až dosud zapisovali v Lagrangeově tvaru.

Ukažme nyní, že skutečně existuje právě jeden soubor koeficientů a_0, a_1, \dots, a_n takový, že polynom N_n splňuje podmínky (3.20.2). Dosadíme-li do (3.20.2) za $N_n(x_i)$, $i = 0, 1, \dots, n$ dostáváme postupně

$$(3.20.3) \quad \begin{aligned} N_n(x_0) &= a_0 &&= f(x_0), \\ N_n(x_1) &= a_0 + a_1(x_1 - x_0) &&= f(x_1), \\ N_n(x_2) &= a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) &&= f(x_2), \\ &\dots\dots\dots && \\ N_n(x_n) &= a_0 + \dots + a_n(x_n - x_0)(x_n - x_1)\dots(x_n - x_{n-1}) &&= f(x_n). \end{aligned}$$

Díváme-li se na (3.20.3) jako na soustavu rovnic pro výpočet a_0, a_1, \dots, a_n vidíme, že je to soustava lineárních algebraických rovnic s trojúhelníkovou maticí, jejíž diagonální prvky jsou při navzájem různých tabulkových bodech různé od nuly. Taková soustava má právě jedno řešení [21], které se dá vypočítat obdobou zpětného chodu Gaussovy eliminační metody [z první rovnice vypočítáme $a_0 = f(x_0)$, z druhé $a_1 = (f(x_1) - a_0)/(x_1 - x_0)$ atd.].

3.21 Poměrné diference.

Počítat koeficienty a_j , $j = 0, 1, \dots, n$. Newtonova interpolačního polynomu přímo ze soustavy rovnic (3.20.3) není praktické. Abychom je mohli výhodně vyjádřit ve formě vhodné pro výpočet, zavedeme nejprve veličiny, kterým budeme říkat poměrné diference a které se budou počítat jednoduchým rekurentním způsobem z tabulky funkčních hodnot $\{(x_i, f(x_i))\}$, $i = 0, 1, \dots, n$. V dalším výkladu uvidíme, že z poměrných diferencí se koeficienty Newtonova polynomu již snadno získají a že poměrné diference mají v numerické matematice význam i samy o sobě.

Pro jednotnost zápisu nejprve formálně nazveme funkční hodnoty $f(x_i)$ *poměrnými diferencemi nultého řádu* a budeme je v této souvislosti značit $f[x_i]$ [$= f(x_i)$]. Pro danou funkci f a tabulkové body $x_i, i = 0, 1, \dots, n$ utvoříme z diferencí nultého řádu všechny podíly $f[x_i, x_{i+1}]$ tvaru

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} \left(= \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \right),$$

$i = 0, 1, \dots, n-1$, a nazveme je *poměrnými diferencemi prvního řádu*. *Poměrné diference* $f[x_i, x_{i+1}, x_{i+2}]$ *druhého řádu* tvoříme nyní analogicky z diferencí prvního řádu jako podíly

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i},$$

$i = 0, 1, \dots, n-2$. Jestliže jsme již stanovili diference $(k-1)$ -ního řádu, definujeme pro $i+k \leq n$ *poměrné diference* $f[x_i, x_{i+1}, \dots, x_{i+k}]$ *k-tého řádu* rekurentním vztahem

$$(3.21.1) \quad f[x_i, x_{i+1}, \dots, x_{i+k-1}, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k-1}, x_{i+k}] - f[x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

[Diference na pravé straně (3.21.1) jsou $(k-1)$ -ního řádu.] Diference n -tého řádu existuje při n tabulkových bodech pouze pro $i = 0$ (je to $f[x_0, x_1, \dots, x_n]$).

Rekurentní výpočet poměrných diferencí pro danou tabulku hodnot funkce f uspořádáme (podobně jako u Nevillova algoritmu) do *tabulka poměrných diferencí* (tab. 4). Tabulku poměrných diferencí budeme sestavovat po řádcích. Šipky v tabulce ukazují, z kterých hodnot v levém sloupci se podle (3.21.1) počítají diference v sousedním pravém sloupci tabulky.

Tabulka 4

x_0	$f(x_0)$	$= f[x_0]$			
x_1	$f(x_1)$	$= f[x_1]$	\searrow	$f[x_0, x_1]$	
\vdots	\vdots	\vdots		\ddots	
x_{n-1}	$f(x_{n-1})$	$= f[x_{n-1}]$	\searrow	$f[x_{n-2}, x_{n-1}] \dots$	$f[x_0, x_1, \dots, x_{n-1}]$
x_n	$f(x_n)$	$= f[x_n]$	\searrow	$f[x_{n-1}, x_n] \dots$	$f[x_1, x_2, \dots, x_n]$
					\searrow
					$f[x_0, x_1, \dots, x_n]$

Všimněme si ještě té - velmi důležité - vlastnosti poměrných diferencí, že $f[x_i, x_{i+1}, \dots, x_{i+k-1}, x_{i+k}]$ závisí pouze na řádcích $i, i+1, \dots, i+k$ tabulky diferencí, tj. pouze na uzlech $x_i, x_{i+1}, \dots, x_{i+k}$ a na hodnotách funkce f v nich. Speciálně závisí diference $f[x_0, x_1, \dots, x_i]$ pouze na prvních $i+1$ řádcích tabulky. Tato vlastnost je ihned vidět z definice (3.21.1) poměrných diferencí. Dá se také ukázat [3], že poměrné diference jsou symetrické funkce svých argumentů v tom smyslu, že při libovolné permutaci uzlů $x_i, x_{i+1}, \dots, x_{i+k}$ zůstává hodnota $f[x_i, x_{i+1}, \dots, x_{i+k}]$ beze změny.

Poměrné diference mají uplatnění např. při aproximaci derivací funkce f (viz odst. 15.8). Důvodem je skutečnost, že platí [3]

$$(3.21.2) \quad f[x_0, x_1, \dots, x_i] = \frac{f^{(i)}(\xi)}{i!}, \quad i = 0, 1, \dots, n,$$

kde ξ ke nějaký bližší neurčený bod ležící v int (x_0, x_1, \dots, x_i) . Ze (3.12.2) plyne, že je-li polynom N -tého stupně, jsou jeho poměrné diference k -tého řádu pro $k > N$ rovny nule. Z velikosti poměrných diferencí se tak dá v zásadě usuzovat na to, zda se daná funkce bude dobře aproximovat polynomem.

3.22 Příklad.

Sestavíme tabulku poměrných diferencí (tab. 5) pro funkci f zadanou stejně jako v příkl. 3.4.

Tabulka 5

i	x_i	$f(x_i) = f[x_i]$	$k = 1$	$k = 2$	$k = 3$
0	0	1			
1	1	2	$f[x_0, x_1] = 1$		
2	-1	2	$f[x_1, x_2] = 0$	$f[x_0, x_1, x_2] = 1$	
3	3	0	$f[x_2, x_3] = -\frac{1}{2}$	$f[x_1, x_2, x_3] = -\frac{1}{4}$	$f[x_0, x_1, x_2, x_3] = -\frac{5}{12}$

V tabulce 5 např.

$$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2} = \frac{0 - 2}{3 + 1} = -\frac{1}{2}$$

a jako poslední hodnotu počítáme

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{-\frac{1}{4} - 1}{3 - 0} = -\frac{5}{12}.$$

3.23 Koeficienty interpolačního polynomu.

Význam poměrných diferencí pro interpolaci spočívá v tom, že se dá ukázat [3], že pro koeficienty a_j Newtonova interpolačního polynomu platí

$$(3.23.1) \quad a_j = f[x_0, x_1, \dots, x_j], \quad j = 0, 1, \dots, n.$$

Ověřme to alespoň pro $j = 0, 1$. Z první rovnice (3.20.3) máme ihned

$$a_0 = f(x_0) = f[x_0].$$

Vypočítáme-li z druhé rovnice a_1 , dostáváme

$$a_1 = \frac{f(x_1) - a_0}{x_1 - x_0} = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = f[x_0, x_1].$$

Pro $j > 1$ je ověření již technicky náročné a nebudeme je provádět.

Koeficienty Newtonova polynomu jsou tedy čísla na diagonále, tabulky poměrných diferencí (tab. 4, odst. 3.21) a Newtonův polynom můžeme proto zapsat jako

$$(3.23.2) \quad \begin{aligned} N_n(x) = & f[x_0] + f[x_0, x_1](x - x_0) + \\ & + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + \\ & + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1)\dots(x - x_{n-1}). \end{aligned}$$

Interpolační polynom zapsaný ve tvaru (3.23.2) má dvě zásadní výhody oproti Lagrangeovu tvaru. Především je to skutečnost, že koeficienty $f[x_0, x_1, \dots, x_j]$ ve vyjádření (3.23.2) nezávisí na x . Je proto možné vypočítat je (při daných vstupních datech) jednou provždy a hodnoty interpolačního polynomu pro různá x pak počítat postupným dosazováním do (3.23.2). Druhou výhodou je to, že koeficienty Newtonova polynomu mají vlastnost, které se říká *permanence*. Přidáme-li totiž k uzlům interpolace x_0, x_1, \dots, x_n další bod x_{n+1} různý od všech ostatních uzlů, bude mít příslušný interpolační polynom N_{n+1} koeficienty

$$f[x_0], f[x_0, x_1], \dots, f[x_0, x_1, \dots, x_n], f[x_0, x_1, \dots, x_n, x_{n+1}].$$

Prvních $n + 1$ koeficientů se oproti N_n vůbec nezmění, protože jejich hodnoty závisí pouze na x_i a $f(x_i)$ pro $i = 0, 1, \dots, n$. U Newtonova interpolačního polynomu je tak možné pohodlně zvyšovat stupeň interpolačního polynomu tím, že přibíráme do výpočtu další uzly interpolace. Polynom v Lagrangeově tvaru bychom museli v takovém případě sestrojovat celý znovu.

3.24 Příklad.

Newtonův interpolační polynom třetího stupně pro funkci f z příkl. 3.22 můžeme již ihned napsat. Jeho koeficienty jsou na diagonále tabulky poměrných diferencí (tab. 5, odst. 3.22). Máme

$$\begin{aligned} N_3(x) &= 1 + (x - x_0) + (x - x_0)(x - x_1) - \frac{5}{12}(x - x_0)(x - x_1)(x - x_2) = \\ &= 1 + x + x(x - 1) - \frac{5}{12}x(x - 1)(x + 1). \end{aligned}$$

3.25 Algoritmy.

Všimneme si toho, že v s -tém řádku tabulky poměrných diferencí (tab. 4, odst. 3.21), kde $s = 0, 1, \dots, s$, jsou obsaženy vesměs difference tvaru $f[x_{s-k}, x_{s-k+1}, \dots, x_{s-1}, x_s]$, a to pro $k = 0, 1, \dots, s$. Pro algoritmický zápis výpočtu tabulky poměrných diferencí po jejich řádcích bude proto výhodné označit

$$F_{sk} = f[x_{s-k}, x_{s-k+1}, \dots, x_{s-1}, x_s].$$

Při tomto označení budou v s -tém řádku tab. 4. zleva doprava veličiny

$$F_{s0}, F_{s1}, \dots, F_{s,s-1}, F_{ss}.$$

Rekurenci (3.21.1) pro výpočet poměrných diferencí převedeme na rekurenci pro výpočet F_{sk} , $k = 0, 1, \dots, s$, tak, že v ní položíme $i = s - k$, $i + k = s$. Výsledkem je vztah

$$F_{sk} = \frac{F_{s,k-1} - F_{s-1,k-1}}{x_s - x_{s-k}}.$$

Pro výpočet koeficientů Newtonova interpolačního polynomu tak dostáváme algoritmus

$$\begin{aligned}
 (3.25.1) \quad & \text{Vstup : } n, x_0, x_1, \dots, x_n, f(x_0), f(x_1), \dots, f(x_n). \\
 & \text{Pro } s = 0, 1, \dots, n : \\
 & \quad F_{s0} = f(x_s). \\
 & \text{Pro } k = 1, 2, \dots, s : \\
 & \quad F_{sk} = \frac{F_{s,k-1} - F_{s-1,k-1}}{x_s - x_{s-k}}. \\
 & \quad a_s = F_{ss}. \\
 & \text{Výstup : } a_0, a_1, \dots, a_n.
 \end{aligned}$$

Doporučujeme čtenáři, aby si algoritmus prověřil opět na tab. 5 z příkl. 3.22. Hodnota F_{sk} je v tab. 5. na průsečíku řádku s a sloupce k .

Hodnoty interpolačního polynomu vyjádřeného ve tvaru (3.20.1), resp. (3.23.2) můžeme po provedení algoritmu (3.25.1) pohodlně počítat algoritmem podobným Hornerovu algoritmu z [21]:

$$\begin{aligned}
 (3.25.2) \quad & \text{Vstup : } n, x_0, x_1, \dots, x_{n-1}, a_0, a_1, \dots, a_{n-1}, a_n. \\
 & \quad P = a_n. \\
 & \text{Pro } i = n - 1, n - 2, \dots, 1, 0 : \\
 & \quad P = P(x - x_i) + a_i. \\
 & \text{Výstup : } N_n(x) = P.
 \end{aligned}$$

Znaménko rovnosti ve čtvrtém řádku algoritmu (3.25.2) zde používáme ve smyslu přiřazení - tak jako v programovacích jazycích. Veličinu P je třeba chápat jako proměnnou v jazyce FORTRAN, Algol apod. Tuto licenci budeme občas užívat i v dalším výkladu, aniž bychom na to znovu upozorňovali.

3.26 Příklad.

V příkladu 3.24 jsme našli Newtonův interpolační polynom N_3 pro funkci f z příkl. 3.22. Vypočítejme hodnotu tohoto polynomu v bodě $x = 2$.

Užití algoritmu (3.25.2) neznamena nic jiného, než že výraz $N_3(x)$ uzavorkujeme takto:

$$\begin{aligned}
 N_3(x) &= 1 + (x - x_0)1 + (x - x_1)\left[1 - \frac{5}{12}(x - x_2)\right] = \\
 &= 1 + x1 + (x - 1)\left[1 - \frac{5}{12}(x + 1)\right].
 \end{aligned}$$

Skutečně, algoritmus (3.25.2) dává pro $x = 0$, $x_0 = 0$, $x_1 = 1$, $x_2 = -1$, $a_0 = 1$, $a_1 = 1$, $a_2 = 1$, $a_3 = -\frac{5}{12}$ postupně

$$\begin{aligned}
 P &= -\frac{5}{12}, \\
 P &= -\frac{5}{12}(2 + 1) + 1 = -\frac{1}{4} \quad (i = 2), \\
 P &= -\frac{1}{4}(2 - 1) + 1 = \frac{3}{4} \quad (i = 1), \\
 P &= \frac{3}{4} \cdot 2 + 1 = -\frac{5}{2} \quad (i = 0).
 \end{aligned}$$

je tedy $N_3(2) = \frac{5}{2}$. Výsledek je stejný jako hodnota $L_3(2)$ v příkl. 3.17 a 3.4.

3.27 Ekvidistantní uzly interpolace.

Jsou-li uzly interpolace x_i , $i = 0, 1, \dots, n$ uspořádány podle velikosti a *ekvidistantní*, tj. je-li $x_{i+1} - x_i = h$, $i = 0, 1, \dots, n-1$, $h = \text{konst} > 0$, je možné sestavení Newtonova interpolačního polynomu dále zjednodušit. Poměrné diference (3.21.1) se v tomto případě dají vyjádřit pomocí jednodušších veličin, kterým se říká pouze diference nebo dopředné diference.

V souladu s tradicí budeme u ekvidistantních tabulek používat označení $f_i = f(x_i)$. Nechť je tedy dána tabulka $\{(x_i, f_i), i = 0, 1, \dots, n\}$ hodnot funkce f s ekvidistantními uzly $x_i = x_0 + ih$, $i = 0, 1, \dots, n$, $h = \text{konst}$. Konstantně h budeme říkat *krok tabulky*. Dopředné diference se definují obdobně jako diference poměrné. Pro každé $i = 0, 1, \dots, n$ definujeme *dopřednou diferenci nultého řádu* vztahem $\Delta^0 f_i = f_i$ a pro $i = 0, 1, \dots, n-1$ definujeme *dopřednou diferenci prvního řádu* vztahem $\Delta^1 f_i = f_{i+1} - f_i$. Často se místo Δ^1 píše pouze Δ . *Dopřednou diferenci k -tého řádu* $\Delta^k f_i$, $k \geq 2$, definujeme nyní pro $i + k \leq n$ rekurentním vztahem

$$(3.27.1) \quad \Delta^k f_i = \Delta(\Delta^{k-1} f_i) = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i.$$

Pro $i \leq n-2$ tedy např. je

$$\Delta^2 f_i = \Delta(\Delta f_i) = \Delta f_{i+1} - \Delta f_i = (f_{i+2} - f_{i+1}) - (f_{i+1} - f_i) = f_{i+2} - 2f_{i+1} + f_i.$$

Podobně jako tomu bylo u poměrných diferencí, závisí dopředná diference $\Delta^k f_i$ pouze na hodnotách $f_i, f_{i+1}, \dots, f_{i+k}$ (odtud název *dopředná* diference).

Mezi poměrnými diferencemi s ekvidistantními body x_i , $i = 0, 1, \dots, n$ a dopřednými diferencemi platí vztah

$$(3.27.2) \quad f[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{\Delta^k f_i}{k!h^k},$$

který se snadno při pevném i ověří indukcí podle k [indukční krok se provede užitím (3.21.2) a (3.27.1)].

Dopředné diference můžeme počítat opět algoritmem (3.25.1). Za tím účelem označíme $F_{sk} = \Delta^k f_{s-k}$, $s = 0, 1, \dots, n$, $k = 0, 1, \dots, s$, a vzorec pro F_{sk} ve (3.25.1) zaměníme vzorcem

$$(3.27.2) \quad F_{sk} = F_{s,k-1} - F_{s-1,k-1},$$

který je přepisem vztahu (3.27.1) pro $i = s - k$. Při ručním výpočtu založeném na rekurentním použití vztahu (3.27.1) uspořádáváme diference opět do tabulky (tab. 6).

Tabulka 6

x_0	$f(x_0)$	$= f_0$			
x_1	$f(x_1)$	$= f_1$	\searrow	\longrightarrow	Δf_0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
x_{n-1}	$f(x_{n-1})$	$= f_{n-1}$	\searrow	\longrightarrow	$\Delta f_{n-2} \dots \Delta^{n-1} f_0$
x_n	$f(x_n)$	$= f_n$	\searrow	\longrightarrow	$\Delta f_{n-1} \dots \Delta^{n-1} f_1 \longrightarrow \Delta^n f_0$

Šipky ukazují, z kterých hodnot v levém sloupci tab. 6 se podle (3.27.1) počítají diference v sousedním pravém sloupci. Při zápisu diferencí do tabulky se často vynechává desetinná čárka a píšou se pouze *významné číslice* (číslíčky normalizované mantisy při vyjádření čísla v semilogaritmickém tvaru). Místo 0,079 576 tedy do tabu.ky zaneseme 79576.

3.28 Příklad.

Sestrojíme tabulku diferencí pro $f(x) = \sin x$, $x_0 = 0,6$, krok $h = 0,1$ a $n = 4$. Dostáváme (srov. tab. 6):

x_i	f_i			
0,6	0,564 642			
0,7	0,644 218	79576		
0,8	0,717 356	73138	-6438	
0,9	0,783 327	65971	-7167	-729
1,0	0,841 471	58144	-7827	-660 69

Je tedy např.

$$\begin{aligned}\Delta f_2 &= f_3 - f_2 = 0,065\,971, \\ \Delta^4 f_0 &= \Delta^3 f_1 - \Delta^3 f_0 = -0,000\,660 + 0,000\,729 = 0,000\,069 \quad \text{apod.}\end{aligned}$$

3.29 Interpolační polynom s dopřednými diferencemi.

Jsou-li uzly interpolace ekvidistantní s krokem h , můžeme koeficienty Newtonova interpolačního polynomu a_j , $j = 0, 1, \dots, n$, pomocí (3.27.2) snadno stanovit z diferencí $\Delta^j f_0$, $j = 0, 1, \dots, n$, tedy opět z hodnot na diagonále tabulky diferencí (tab. 6). Chceme-li užít k zápisu interpolačního polynomu dopředné diference, je však účelné provést v polynomu N_n záměnu proměnných a položit

$$(3.29.1) \quad x = x_0 + th.$$

Budeme tam měřit v jednotkách h vzdálenost bodu x od x_0 . Zavedeme ještě jedno označení, kterým zobecníme pojem binomického koeficientu $\binom{n}{k}$ pro n reálná. Položíme pro reálné t a přirozené j

$$(3.29.2) \quad \binom{n}{k} = \frac{t(t-1)\dots(t-j+1)}{j!}, \quad \binom{t}{0} \equiv 1.$$

Po substituci (3.29.1) a dosazení za poměrnou diferenci podle (3.27.2) přejde Newtonův interpolační polynom N_n v polynom N_n^+ , kde

$$(3.29.3) \quad \begin{aligned}N_n^+(t) &= f_0 + \binom{t}{1} \Delta f_0 + \dots + \binom{t}{n} \Delta^n f_0 = \\ &= \sum_{j=0}^n \binom{t}{j} \Delta^j f_0 \quad \left(t = \frac{x - x_0}{h} \right).\end{aligned}$$

Polynom N_n^+ se nazývá *Newtonův interpolační polynom s dopřednými diferencemi* nebo také *Newtonův polynom pro interpolaci vpřed* ($t \geq 0$). Protože jde jen o speciální způsob zápisu interpolačního polynomu, platí pro chybu aproximace polynomem N_n^+ stále vzorce (3.10.1) a (3.12.1). Provedeme-li substituci (3.29.1) také např. v (3.12.1), dostaneme odtud odhad chyby ve tvaru

$$(3.29.4) \quad |f(x_0 - th) - N_n^+(t)| \leq Mh^{n+1} \left| \binom{t}{n+1} \right|$$

kde $M \geq |f^{(n+1)}(\xi)|$ pro všechna $\xi \in \text{int}(x_0, x_n, x)$.

3.30 Příklad.

Určeme $\sin 0,63$ z tabulky příkl. 3.28 a odhadněme přesnost výsledku. Použijeme interpolační polynom N_n^+ .

Máme $x_0 = 0,6$, $h = 0,1$, $n = 4$. Zřejmě $t = 0,3$. Hodnoty diferencí, které potřebujeme dosadit do (3.29.3), jsou v tabulce vytištěny tučně (pozor na desetinnou čárku!). Jako aproximaci hodnoty $\sin 0,63$ interpolací s použitím polynomu čtvrtého stupně dostáváme (na 6D) podle (3.29.3)

$$\begin{aligned} N_4^+(0,3) = & 0,564642 + \binom{0,3}{1} \cdot 0,079576 - \binom{0,3}{2} \cdot 0,006438 - \\ & - \binom{0,3}{3} \cdot 0,000729 + \binom{0,3}{4} \cdot 0,000069 \doteq 0,589145. \end{aligned}$$

Pátá derivace funkce $\sin x$ je $\cos x$, a je tedy v absolutní hodnotě menší než 1. Absolutní hodnota chyby aproximace je tedy podle (3.29.4) menší než

$$1 \cdot 10^{-5} \cdot \left| \binom{0,3}{5} \right| = \frac{10^{-5}}{5!} |(0,3) \cdot (-0,7) \cdot (-2,7) \cdot (-3,7)| < 0,3 \cdot 10^{-6}.$$

3.31 Interpolační polynom se zpětnými derivacemi.

V kapitole III o řešení diferenciálních rovnic budeme potřebovat interpolační polynom ve tvaru obdobném (3.29.3), ale vycházejícím z hodnoty funkce f na pravém konci soustavy ekvidistančních uzlů interpolace x_i , tj. z hodnoty f_n v bodě $x_n = x_0 + nh$. V tomto tvaru interpolačního polynomu budou pak vystupovat difference tvořené zpětně z funkčních hodnot funkce f vlevo od bodu x_n . Těmito diferencím se říká zpětně difference.

Zcela obdobně jako jsme v odst. 3.27 zavedli dopředné difference, definujeme *zpětně difference nultého řádu* vztahem $\Delta^0 f_i = f_i$, $i = 0, 1, \dots, n$, *zpětně difference prvního řádu* definujeme pro $i = 1, 2, \dots, n$ vztahem $\Delta^1 f_i = f_i - f_{i-1}$ a zpětně difference k -tého řádu rekurentním vztahem

$$(3.31.1) \quad \Delta^k f_i = \Delta(\Delta^{k-1} f_i - \Delta^{k-1} f_{i-1})$$

pro $2 \leq k \leq i$. (Píšeme zde Δ místo Δ^1 .) Zpětná difference $\Delta^k f_i$ závisí pouze na hodnotách $f_i, f_{i-1}, \dots, f_{i-k}$ odtud název *zpětná difference*).

Při zápisu interpolačního polynomu pomocí zpětných diferencí užíváme stejně jako v odst. 3.29 substituci. Tentokrát však klademe $x = x_n + th$ (měříme vzdálenost bodu x od x_n místo

od x_0). Dá se ukázat [3], [26], že interpolační polynom pro data $(x_i, f(x_i), i = 0, 1, \dots, n)$ je pak možno zapsat ve tvaru

$$(3.31.2) \quad \begin{aligned} N_n^-(t) &= f_n - \binom{-t}{1} \Delta f_n + \dots + (-1)^n \binom{-t}{n} \Delta^n f_n = \\ &= \sum_{j=0}^n \binom{-t}{j} \Delta^j f_n \quad \left(t = \frac{x - x_n}{h} \right). \end{aligned}$$

Polynom N_n^- se nazývá *Newtonův interpolační polynom se zpětnými diferencemi* nebo také *Newtonův polynom pro interpolaci vzad* ($t \leq 0$).

Připomínáme znovu, že ani polynom N_n^- není nic jiného než určitý způsob zápisu interpolačního polynomu, který je vstupními daty jednoznačně určen. Čtenář si možná také již všiml toho, že není nezbytně nutné zavádět oba druhy diferencí, Δ i Δ . Platí např. (ověřte1).

$$\Delta f_i = \Delta f_{i-1}, \quad \Delta^2 f_i = \Delta^2 f_{i-2}.$$

Obecně se dá ukázat, že platí

$$\Delta^k f_i = \Delta^k f_{i-k}.$$

Proto jsou číselné hodnoty položek v tabulce zpětných diferencí (kterou nebudeme popisovat) stejné jako v tabulce dopředných diferencí, jde pouze o jiné označení. Hodnoty diferencí vystupující v Newtonově polynomu (3.31.2) pro interpolace vzad najdeme tudíž v posledním řádku u tabulky diferencí (tab. 6).

Kromě dopředných a zpětných diferencí se zavádějí také it centrální diference definované vztahem

$$\delta f_i = f_{i+1/2} - f_{i-1/2}.$$

a dále opět rekurentně. Přitom klademe $f_{i\pm 1/2} = f_{x_i \pm h/2}$. Pomocí tabulkových hodnot lze tedy vyjádřit pouze centrální diference sudých řádů. Je např.

$$\delta^2 f_i = \delta(\delta f_i) = \delta f_{i+1/2} - \delta f_{i-1/2} = f_{i+1} - 2f_i + f_{i-1}.$$

Centrální diference $\delta^k f_i$ závisí na hodnotách funkce f v tabulkových bodech, které jsou rozloženy symetricky kolem x_i . O interpolačních polynomech s centrálními diferencemi viz např. [3], [26].

3.32 Konvergence interpolačního procesu.

V tomto odstavci se budeme zabývat otázkou, zda při interpolaci funkce f na intervalu $\langle a, b \rangle$ s uzly x_0, x_1, \dots, x_n povede zvyšování počtu uzlů (a tedy aproximace interpolačními polynomy stále vyšších stupňů) ke stále lepší aproximaci funkce f .

Nechť je dána funkce f , která je spojitá na intervalu $\langle a, b \rangle$, a označme $D = \{a = x_0 < x_1 < \dots < x_n = b\}$ soubor uzlů interpolace. Víme již, že ke každému souboru $n + 1$ uzlů D existuje interpolační polynom $p_D(x_i) \in \mathcal{P}_n$ takový, že $p_D(x_i) = f(x_i)$ pro $x_i \in D$. Vezměme nyní posloupnost souborů

$$D_s = \{a < x_0^{(s)} < x_1^{(s)} < \dots < x_n^{(s)} = b\}$$

a ke každému souboru uzlů interpolace D_s přiřadíme interpolační polynom p_{D_s} . Zajímá nás, zda polynomy p_{D_s} budou konvergovat k f za předpokladu, že uzly interpolace volíme tak, aby

$$\max_i |x_{i+1}^{(s)} - x_i(s)| \rightarrow 0$$

pro $s \rightarrow \infty$. Odpověď je v mnoha směrech negativní. At' volíme uzly interpolace jakkoli, vždy existuje spojitá funkce tak, že interpolační proces tuto funkci nekonverguje stejnoměrně.

Zvolíme-li uzly interpolace ekvidistantně, tj.

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

pak např. pro tak elementární funkce jako jsou $(1+x^2)^{-1}$ na $\langle a, b \rangle = \langle -5, 5 \rangle$ nebo \sqrt{x} na $\langle a, b \rangle = \langle 0, 1 \rangle$ nekonverguje interpolační proces ani bodově na celém $\langle a, b \rangle$. Dalším negativním příkladem je interpolace funkce $|x|$ na $\langle -1, 1 \rangle$, kde interpolační proces s ekvidistantními uzly nekonverguje v žádném bodě s výjimkou bodů $x = -1, 0, 1$. Interpolace spojitých funkcí může tedy při použití ekvidistantních uzlů dávat problematické výsledky. Jediný použitelný pozitivní výsledek je tu ten, že interpolační proces s ekvidistantními uzly konverguje stejnoměrně, je-li f celistvá analytická funkce. Interpolace ekvidistantních dat polynomy vyšších stupňů je v některých případech také špatně podmíněná úloha, tj. malé nepřesnosti ve vstupních datech působí velké chyby v hodnotě interpolačního polynomu. Potíže vznikají zejména při aproximování funkce poblíž konců intervalu $\langle x_0, x_n \rangle$, kde se pak navíc interpolační polynomy vyšších stupňů zpravidla značně "vlní".

Chceme-li aproximovat nějakou funkci polynomem na celém intervalu a můžeme-li si vybrat body, v nichž počítáme nebo měříme tabulkové hodnoty, je proto vždy rozumnější nevolit uzly x_i ekvidistantně. Dobrá strategie je volit tabulkové body tak, aby byly rozloženy stejně jako kořeny Čebyševových polynomů (viz odst. 5.6 a [26]), čímž mimo jiné minimalizujeme hodnotu $\max |e_n(x)|$, kde $e_n(x) = (x-x_0)(x-x_1)\dots(x-x_n)$ [viz (3.10.1), (3.12.1) a příkl. 3.11].

Pracujeme-li tedy s $\langle a, b \rangle = \langle -1, 1 \rangle$, volíme

$$(3.32.1) \quad x_i = \cos \left(\frac{2i+1}{n+1} \frac{\pi}{2} \right), \quad i = 0, 1, \dots, n$$

$[x_i, i = 0, 1, \dots, n]$ jsou kořeny Čebyševova polynomu $(n+1)$ -ního stupně, viz příkl. 5.6.1]. Na obecný interval $\langle a, b \rangle$ uzly (3.32.1) zobrazíme jednoduchou lineární transformací

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

Volíme-li uzly interpolace podle (3.32.1), má interpolační proces při $n \rightarrow \infty$ tu vlastnost, že vzniklé interpolační polynomy konvergují na $\langle a, b \rangle$ stejnoměrně k aproximované funkci už při velmi mírných požadavcích na tuto funkci. Postačující podmínkou pro stejnoměrnou konvergenci je např. již existence spojitě první derivace f' na $\langle a, b \rangle$. Tuto podmínku lze ještě dále zeslabit. Podobná konvergenční věta platí také pro uzly volené jako kořeny Legendrových polynomů (viz příkl. 5.6.3).

Podrobné poučení o konvergenci interpolačních procesů (pro interpolaci polynomy) najde zájemce např. v [3].

3.33 Úlohy.

3.33.1

Funkce f je dána svými hodnotami v uzlech $x_0 = 1$, $x_1 = 5$, $x_2 = 3$, $x_3 = 0$. Je $f(1) = 2$, $f(5) = 30$, $f(3) = -4$, $f(0) = 5$. Sestrojte Lagrangeův interpolační polynom třetího stupně pro funkci f .

$$[L_3(x) = x^3 - 4x^2 + 5]$$

3.33.2

Funkce $\ln x$ je kvadraticky interpolována v bodech $x = 10, 11, 12$.

a) Odhadněte chybu aproximace interpolačním polynomem v bodě $x = 11, 1$.

$$[|E(11, 1)| \leq 3,3 \cdot 10^{-5}.]$$

b) Jak závisí znaménko této chyby na x ?

$[E(x) > 0$ pro $x \in (10, 11)$, znaménko se mění v uzlech interpolace.]

3.33.3

Sestrojte tabulku hodnot funkce $f(x) = 2x + 1$ pro $x_i = 0, 1i$, $i = 0, 1, \dots, 5$. Stanovte $f(0, 26)$ lineární interpolací a ověřte si, že výsledek je přesný $[E(0, 26) = 0]$. Vysvětlete

[Návod: Užijte (3.10.1).]

3.33.4

Je dána ekvidistantní tabulka hodnot funkce e^x na intervalu $\langle 0, 1 \rangle$ s krokem $h = 0,01$. Odhadněte největší možnou chybu, které se dopustíme při lineární interpolaci v této tabulce.

$[|E(x)| \leq (e/2) \cdot 10^{-4} \leq 1,36 \cdot 10^{-4}$ pro $x \in \langle 0, 1 \rangle$; při přesnějším odhadu $(x - x_0)(x - x_1)$ dostaneme dokonce $|E(x)| \leq (e/8) \cdot 10^{-4}$.]

3.33.5

Přepište Nevillův algoritmus (3.16.2) pro funkcí f danou ekvidistantní tabulkou s krokem h ($x_{i+1} - x_i \equiv h$).

[Návod: Položte např. $x = x_0 + th$.]

3.33.6

Funkce f je dána těmito hodnotami, které jsou zaokrouhleny na 3D: $f(0, 2) = 0,962$, $f(0, 4) = 0,862$, $f(0, 6) = 0,735$, $f(0, 8) = 0,610$.

a) Sestavte Nevillovo schéma pro tato vstupní data a $\hat{x} = 0,5$. Stanovte přibližnou hodnotu $f(0,5)$ při interpolaci kubickým polynomem.

$$[f(0,5) \approx P_{33} \doteq 0,8001.]$$

b) Odhadněte chybu aproximace kvadratickým interpolačním polynomem s uzly $0,2; 0,4; 0,6$. Použijte přitom odhad $|f'''(x)| < 4,67$ pro $x \in \langle 0, 1 \rangle$. Porovnejte se skutečnou chybou, víte-li, že $f(x) = (1 + x^2)^{-1}$.

$$[|E(0,5)| \leq 0,5 \cdot 10^{-3} \cdot 4,67 \leq 2,34 \cdot 10^{-3}; \text{skutečná chyba je } -0,001875.]$$

3.33.7

Funkce f je dána tab. 7. Nalezněte Nevillovým algoritmem přibližnou hodnotu $f(1,5) \approx P_{66}$ a posuďte výsledek.

[Návod: Sestrojte cele Nevillovo schéma a posuďte chování hodnot na diagonále. $P_{66} \doteq 0,3573$.]

Tabulka 7

x	0,1	0,2	0,5	1,0	2,0	5,0	10,0
$f(x)$	10,0	5,0	2,0	1,0	0,5	0,2	0,1

3.33.8

Sestrojte Newtonův interpolační polynom N_3 pro funkci f danou hodnotami $f(1) = 0$, $f(2) = 2$, $f(4) = 12$, $f(5) = 20$.

$$[N_3(x) = 2(x-1) + (x-1)(x-2) = (x-1)x.]$$

3.33.9

Máte informaci, že data v tab. 8 jsou hodnoty jistého kubického polynomu. Užijte poměrné difference a ověřte difference a ověřte toto tvrzení. Sestrojte příslušný polynom.

$$[f(x) = x^3 + x^2 - 2x + 3.]$$

Tabulka 8

x	-3	-1	0	2	3
$f(x)$	-9	5	3	11	33

3.33.10

Sestrojte tabulku diferencí pro funkci f , která je dána svými hodnotami v ekvidistantních uzlech $2 + ih$, $i = 0, 1, 2, 3$, $h = 0, 5$: $f(2) = -1$, $f(2,5) = 5$, $f(3) = 15$, $f(3,5) = 35$. Napište Newtonův polynom pro interpolaci vpřed N_3^+ a pro interpolaci vzad N_3^- .

[Návod: Užijte toho, že $\Delta^k f_i = \Delta^k f_{i-k}$. Kontrola: $\Delta^3 f_0 = 6$.]

3.33.11

Sestrojte tabulku diferencí pro data z tab. 9. Napište Newtonův polynom pro interpolaci vpřed N_3^+ a pro interpolaci vzad N_3^- . Jaký je nejnižší stupeň polynomu, který interpoluje uvedená data?

[Dva. Kontrola: $\Delta^k f_i \equiv 0$, $\Delta^k f_i \equiv 0$ pro $k > 2$.]

Tabulka 9

x	0,0	0,1	0,2	0,3	0,4	0,5
$f(x)$	1,00	1,32	1,68	2,08	2,52	3,00

3.33.12

Sestrojte uzly interpolace na intervalu $\langle 0, 1 \rangle$, které jsou rozloženy podle (3.32.1) s $n = 2$. [$x_0 \doteq 0,933\,01$; $x_1 = \frac{1}{2}$; $x_2 \doteq 0,066\,987$ na 5 platných číslic.]

3.33.13

Napište program, který realizuje algoritmy (3.25.1) a (3.25.2) a umožňuje tak při dané tabulce hodnot funkce f počítat hodnoty příslušného Newtonova interpolačního polynomu. Položte $f(x) = (25x^2 + 1)^{-1}$ a pomocí tohoto programu:

a) Sestrojte interpolační polynomy stupně $n = 5, 10, 15, 20$ pro ekvidistantní uzly interpolace $x_i = -1 + (2/n)i$, $i = 0, 1, \dots, n$. Tabelujte hodnotu aproximace v bodech $(x_i + x_{i+1})/2$, příp. znázorněte f i všechny interpolační polynomy graficky.

b) Proveďte totéž, jako v bodě a), ale s uzly podle (3.32.1). Porovnejte výsledky a) a b).

4 Interpolace racionální funkcí a spline-funcí

4.1 Interpolace racionální funkcí.

Ne každá spojitá funkce se dá dobře aproximovat polynomem, i když důvodů pro to, abychom aproximující funkce používali polynomy, je celá řada (odst. 1.9). V příkladu 3.19.2 a úloze 3.33.7 lze nalézt typické příklady vstupních dat, která polynom aproximuje špatně; v obou případech pocházejí data z funkcí, které mají asymptoty. U takových funkcí dostaneme často lepší výsledky, aproximujeme-li je vhodnou racionální funkcí.

Teoretické problémy interpolace funkcemi ze třídy \mathcal{R}_{pq} racionálních funkcí, které mají v čitateli polynom nejvýše p -tého stupně a ve jmenovateli polynom nejvýše q -tého stupně, jsou podstatně komplikovanější, než tomu bylo u polynomů. Jde tu o aproximace nelineárního typu (hodnota aproximující funkce nezávisí na volitelných parametrech - koeficientech racionální funkce - lineárně). Většinou je však i zde určena $p + q + 1$ vstupními daty $\{(x_i, f(x_i)), i = 0, 1, \dots, p + q\}$ právě jedna racionální funkce $\varphi \in \mathcal{R}_{pq}$,

$$(4.1.1) \quad \varphi(x) = \frac{a_0 + a_1x + \dots + a_px^p}{b_0 + b_1x + \dots + b_qx^q},$$

která splňuje interpolační podmínky

$$(4.1.2) \quad \varphi(x_i) = f(x_i), \quad i = 0, 1, \dots, p + q.$$

Poznamenáváme, že ve všech funkcích z \mathcal{R}_{pq} vystupuje vlastně pouze $p + q + 1$ nezávislých parametrů, neboť jejich koeficienty jsou vždy určeny pouze až na nenulový společný faktor.

Podrobnější studium interpolace racionální funkcí přesahuje rámec tohoto sešitu. Dobré poučení najde čtenář v [29]. Zde se omezíme na to, že pro informaci uvedeme analog Nevillova algoritmu založený na interpolaci racionální funkcí. Tento algoritmus pochází od J. Stoera [29] a je určen k výpočtu hodnoty $\varphi(\hat{x})$ interpolující racionální funkce φ pro vstupní data $\{(x_i, f(x_i)), i = 0, 1, \dots, n\}$. Konstruovaná funkce φ přitom patří do třídy \mathcal{R}_{pq} , kde $n = p + q$ a $p = q$ nebo $p = q - 1$ podle toho, zda n je sudé nebo liché číslo.

(4.1.3) Vstup : $n, \hat{x}, x_0, x_1, \dots, x_n, f(x_0), f(x_1), \dots, f(x_n)$.

Pro $i = 0, 1, \dots, n$:

$$R_{i,-1} = 0; R_{i0} = f(x_i).$$

Pro $k = 1, 2, \dots, i$:

$$R_{ik} = R_{i,k-1} + (\hat{x} - x_i) \frac{R_{i,k-1} + R_{i-1,k-1}}{(x_i - x_{i-k}) - (\hat{x} - x_{i-k}) \frac{R_{i,k-1} + R_{i-1,k-1}}{R_{i,k-1} + R_{i-1,k-2}}}.$$

Výstup : $\varphi(\hat{x}) = R_{nn}$.

Vzhledem k rozdílům ve jmenovatelích může algoritmus (4.1.3) selhat (dělení nulou). Praxe však ukazuje, že se to stává jen velmi zřídka. Stejně jako u Nevillova algoritmu jsou i zde mezivýsledky R_{ik} hodnotami jistých racionálních interpolačních funkcí.

Ruční výpočet podle (4.1.3) můžeme uspořádat do tabulky zcela obdobné Nevillovu schématu (tab. 2, odst. 3.1.6). Rozdíl bude v tom, že hodnota R_{ik} se nyní nepočítá ze dvou hodnot v sousedním levém sloupci schématu, ale ze tří hodnot $R_{i,k-1}, R_{i-1,k-1}, R_{i-1,k-2}$, takže k výpočtu R_{ik} používáme dva sousední sloupce vlevo od sloupce k .

4.2 Příklad.

Aplikujme Stoerův algoritmus (4.1.3) na úlohu z příkl. 3.19.2, kde nám interpolace polynomem nedala při vynaloženém objemu výpočtu dobrý výsledek. Máme stanovit přibližnou hodnotu $f(1, 8)$. Výpočet uspořádáme do tabulky (tab. 10). Šipky v tab. 10 opět naznačují, které hodnoty se použijí při výpočtu. Je např.

$$R_{11} = R_{10} + 0,8 \frac{R_{10} - R_{00}}{-1 - (-0,2)(R_{10} - R_{00})/(R_{10} - R_{0,-1})},$$

tj.

$$R_{11} = 0,36788 + 0,8 \frac{0,23254}{-1 - (0,2) \cdot (0,23254)/(0,36788)} \doteq 0,15493.$$

Vzhledem k ustálenému chování hodnot na diagonále tabulky (jsou tištěny tučně) klademe $f(1, 8) \doteq 0,165$ ($\doteq R_{44}$). Připomínáme, že $e^{-1,8} = 0,16530$. Hodnota R_{44} je hodnota interpolační funkce $\varphi \in \mathcal{R}_{22}$ v bodě $x = 1, 8$.

Tabulka 10

i	x_i	$R_{i,-1}$	$R_{i0} = f(x_i)$	R_{i1}	R_{i2}	R_{i3}	R_{i4}
0	2	0	0,13534				
1	1	0	0,36788	→ 0,15493			
2	3	0	0,049787	0,10346	1,16291		
3	0	0	1,0000	0,080313	0,18623	→ 0,16463	
4	4	0	0,018316	0,039811	0,12462	0,15758	0,165157

4.3 Aproximace funkcemi, které jsou po částech polynomy.

Metodami popsanými v čl. 3 jsme aproximovali funkci f na intervalu $\langle a, b \rangle$ interpolačním polynomem. Je-li interval $\langle a, b \rangle$ velký, nemožňují často interpolační polynomy nižších stupňů dosáhnout požadované přesnosti. Snaha zmenšit velikost chyby aproximace tím, že použijeme interpolační polynom vyššího stupně, však nevede vždy k cíli. Interpolační polynomy vysokých stupňů se zpravidla v intervalech mezi uzly interpolace značně "vlní". Rovněž úvahy o konvergenci interpolačního procesu (odst. 3.32) ukazují, že zejména v případě ekvidistantních uzlů interpolace nemusí zvyšování stupně interpolačního polynomu přinést zmenšení velikosti chyby aproximace.

Osvědčenou cestou, jak dosáhnout u interpolační aproximace požadované přesnosti a vyhnout se přitom interpolačním polynomům vyšších stupňů, je neaproximovat funkci f jedním polynomem na celém intervalu $\langle a, b \rangle$, ale rozdělit $\langle a, b \rangle$ na několik částí na každém intervalu použít polynom jiný. Budou-li dílčí intervaly dostatečně malé, bude přitom možné aproximovat na nich funkci f polynomy nízkých stupňů. Výsledná aproximující funkce φ je pak na $\langle a, b \rangle$ funkce po částech polynomiální. Snažíme-li se zkonstruovat funkci φ tak, aby byla na celém intervalu $\langle a, b \rangle$ spojitá, měla na $\langle a, b \rangle$ co největší počet spojitých derivací a aby její křivost na intervalu $\langle a, b \rangle$ byla co nejmenší, vedou nás tyto požadavky k užití interpolačních spline-funkcí.

4.4 Příklad: po částech lineární interpolace.

Jako ilustraci postupu naznačeného v odst. 4.3 popíšeme interpolační aproximaci funkce f z příkl. 3.4 funkcí φ , která je po částech polynom prvního stupně. Budeme přitom požadovat, aby funkce φ byla na intervalu $\langle a, b \rangle = \langle -1, 3 \rangle$ spojitá. Grafem takové funkce je lomená čára.

Uzly interpolace z příkl. 3.4 uspořádáme podle velikosti, takže bude platit $x_i < x_{i+1}$, $i = 0, 1, 2, 3$, a označíme $h_i = x_{i+1} - x_i$. Údaje o funkci f z příkl. 3.4 sestavíme do tabulky (tab. 11), ve které budou navíc vystupovat některé další veličiny, jež použijeme zčásti v tomto příkladu, zčásti (λ_i, μ_i, g_i) v příkl. 4.7. Body x_i , $i = 0, 1, 2, 3$, dělí interval $\langle -1, 3 \rangle$ na $N = 3$ dílčí intervaly. Na každém z těchto intervalů $\langle x_i, x_{i+1} \rangle$ budeme funkci f aproximovat polynomem φ_i prvního stupně. Budeme přitom žádat, aby hledaná aproximace φ splňovala v uzlech x_i , $i = 0, 1, 2, 3$, interpolační podmínky $\varphi(x_i)$, $i = 0, 1, 2, 3$.

Tabulka 11

i	x_i	h	$f(x_i)$	$f(x_{i+1}) - f(x_i)$	λ_i	μ_i	g_i
0	-1	1	2	-1			
1	0	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	6
2	1	2	2	-2	$\frac{1}{3}$	$\frac{2}{3}$	-4
3	3		0				

Na každém intervalu $\langle x_i, x_{i+1} \rangle$, $i = 0, 1, 2$, sestrojíme jako interpolační polynom prvního stupně $\varphi_i = N_1$ pro data $(x_i, f(x_i)), (x_{i+1}, f(x_{i+1}))$. Podle (3.23.2) pak pro $x \in \langle x_i, x_{i+1} \rangle$ platí (klademe $n = 1$ a zaměníme $x_0 \rightarrow x_0, x_1 \rightarrow x_{i+1}$)

$$(4.4.1) \quad \begin{aligned} \varphi(x) &\approx \varphi_i = f[x_i] + f[x_i, x_{i+1}](x - x_i) = \\ &= f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{h_i}(x - x_i). \end{aligned}$$

Jediné možné body nespojitosti funkce φ v intervalu $\langle -1, 3 \rangle$ jsou $x_1 = 0$ a $x_2 = 1$, v nichž přechází φ_0 ve φ_1 a φ_1 ve φ_2 . Ale protože platí $\varphi_i(x_{i+1}) = f(x_{i+1}) = \varphi_{i+1}(x_{i+1})$, $i = 0, 1$, je funkce φ zřejmě spojitá na celém intervalu $\langle -1, 3 \rangle$.

Na základě hodnot v tab. 11 dostáváme podle (4.4.1)

$$\varphi(x) = \begin{cases} \varphi_0 = 2 - (x + 1) = 1 - x, & -1 \leq x \leq 0, \\ \varphi_1 = 1 - (x - 0) = 1 + x, & 0 \leq x \leq 1, \\ \varphi_2 = 2 - (x - 1) = 3 - x, & 1 \leq x \leq 3, \end{cases}$$

Graf funkce φ je na obr. 3 (v odst. 4.7) spolu s grafem funkce L_3 z příkl. 3.4.

4.5 Interpolace kubickými spline-funkcemi.

Danou funkci f můžeme na intervalu $\langle a, b \rangle$ v zásadě aproximovat funkce-mi, které jsou po částech polynomy libovolného stupně. Ukazuje se však, že z různých důvodů jsou k tomuto účelu obzvláště vhodné aproximace funkce-mi, které jsou po částech polynomy třetího stupně [32].

Rozdělíme interval $\langle a, b \rangle$ na N dílčích intervalů body x_0, x_1, \dots, x_N takovými, že $a = x_0 < x_1 < \dots < x_{N-1} < x_N = b$. V tomto odstavci se budeme zabývat interpolačními aproximacemi pomocí funkcí φ , které budou mít tyto dvě vlastnosti:

- (4.5.1) a) $\varphi \in \mathcal{C}^2 \langle a, b \rangle$, tj. φ a její derivace jsou spojitě funkce definované na intervalu $\langle a, b \rangle$.
 b) Na každém dílčím intervalu $\langle x_i, x_{i+1} \rangle$, $i = 0, 1, \dots, N - 1$, splývá φ s jistým polynome-m třetího stupně.

Funkce φ , která má vlastnosti (4.5.1), se nazývá *kubická spline-funkce*.²⁾ Kubická spline-funkce se

²⁾ Čti splajn-funkce; ustálený český ekvivalent tohoto termínu zatím neexistuje. Anglické slovo spline znamená elastické pravítko užívané k sestrojování velkých křivek při stavbě lodí. Upevní-li se toto pravítko vhodným způsobem tak, aby procházelo body $(x_i, f(x_i))$, $i = 0, 1, \dots, N$, má jeho průhybová křivka v prvním přiblížení stejný tvar jako graf kubické spline-funkce, která interpoluje tato data [32].

tedy skládá z kubických polynomů, pospojovaných dohromady tak, že jejich hodnoty a hodnoty jejich prvních dvou derivací v bodech x_i , $i = 1, 2, \dots, N - 1$, splývají. Těmto bodům se také říká *uzly spline-funkce* φ . Triviální příklad kubické spline-funkce je samozřejmě každý polynom třetího stupně na $\langle a, b \rangle$.

Předpokládáme nyní, že v bodech x_i , $i = 0, 1, \dots, N$, máme udány funkční hodnoty $f(x_i)$, funkce f a že je naším úkolem sestrojít *interpolační kubickou spline-funkci* φ , pro kterou bude platit

$$(4.5.2) \quad \varphi(x_i) = f(x_i), \quad i = 0, 1, \dots, N.$$

Na rozdíl od Lagrangeova interpolačního polynomu není podmínkami (4.5.2) interpolační spline-funkce φ určena jednoznačně. Je to vidět z této úvahy: Podle (4.5.1b) je φ na každém dílčím intervalu $\langle x_i, x_{i+1} \rangle$ polynom třetího stupně. Takový polynom má 4 neznámé koeficienty, dílčích intervalů je N , a φ je tedy na $\langle a, b \rangle$ určena celkem $4N$ parametry. Tyto parametry máme stanovit tak, aby platilo (4.5.1a) a (4.5.2). Podmínky spojitosti funkcí φ , φ' , a φ'' v uzlech x_i , $i = 1, 2, \dots, N - 1$, dávají celkem $3(N - 1) = 3N - 3$ podmínek, interpolační podmínky (4.5.2) představují dalších $N + 1$ podmínek. Máme tedy celkově $4N - 2$ podmínek pro $4N$ neznámých parametrů. Chybějící dvě podmínky můžeme dodat v zásadě podle své libovůle. Naším cílem je přidat takové dvě podmínky, aby příslušná interpolační spline-funkce existovala a byla již určena jednoznačně.

Dá se ukázat, že vhodnými podmínkami jsou např.

$$(4.5.3) \quad \begin{array}{ll} \text{a)} & \varphi'(a) = f'_0, \quad \varphi'(b) = f'_N; \\ \text{b)} & \varphi''(a) = f''_0, \quad \varphi''(b) = f''_N; \\ \text{c)} & \varphi''(a) = 0, \quad \varphi''(b) = 0; \end{array}$$

Podmínkám (4.5.3) říkáme *okrajové podmínky*. Čísla f'_0, f'_N v okrajových podmínkách (4.5.3a) jsou buď hodnoty první derivace funkce f v krajních bodech intervalu $\langle a, b \rangle$ [$f'_0 = f'(a)$, $f'_N = f'(b)$], nebo nějaké jejich předem dané aproximace. Podobně jsou čísla f''_0, f''_N hodnoty nebo aproximace hodnot druhé derivace funkce f v bodech a a b . Interpolační kubické spline-funkci, která splňuje okrajové podmínky (4.5.3c), se říká *přirozená spline-funkce*.

V tomto sešitě popíšeme podrobně pouze konstrukci interpolačních kubických spline-funkcí, které splňují podmínky (4.5.3b) nebo (4.5.3c). O spline-funkcích, které jsou určeny podmínkami (4.5.3a), a jiných vhodných okrajových podmínkách se lze poučit v [7], [29], [32].

4.6 Konstrukce interpolační spline-funkce.

Mějme dánu tabulku funkčních hodnot $(x_i, f(x_i))$, $i = 0, 1, \dots, N$ takovou, že $x_{i+1} > x_i$ a hledejme interpolační kubickou spline-funkci φ , která splňuje interpolační podmínky (4.5.2) a některé z okrajových podmínek (4.5.3), ve tvaru

$$(4.6.1) \quad \varphi(x) = \varphi_i(x) \quad \text{pro} \quad x \in \langle x_i, x_{i+1} \rangle, \quad i = 0, 1, \dots, N - 1,$$

kde

$$(4.6.2) \quad \varphi(x_i) = a + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

(tento způsob zápisů dílčích kubických polynomů nám značně zjednoduší další výpočty). Naším úkolem je tedy stanovit hodnoty koeficientů a_i, b_i, c_i, d_i , $i = 0, 1, \dots, N - 1$.

Vzhledem k tomu, že φ_i jsou polynomy třetího stupně, jsou jejich druhé derivace φ_i'' polynomy prvního stupně. Druhá derivace hledané spline-funkce φ má být podle (4.5.1a) spojitá, je tedy φ'' spojitá funkce, která je po částech polynom prvního stupně. Taková funkce je planě určena svými hodnotami v bodech x_i , $i = 0, 1, \dots, N$. Její konstrukci jsme popsali v příkl. 4.4. Vyjdeme proto i zde z konstrukce druhé derivace hledané spline-funkce φ a z hodnot $\varphi''(x_i)$, $i = 0, 1, \dots, N$, pak stanovíme hodnoty koeficientů a_i, b_i, c_i, d_i , ve (4.6.2).

Stejně jako v příkl. 4.4 budeme značit $h_i = x_{i+1} - x_i$, $i = 0, 1, \dots, N - 1$. Předpokládejme nejprve, že již známe hodnoty druhé derivace hledané spline-funkce φ v bodech x_i ,

$$(4.6.3) \quad \varphi''(x_i) = M_i, \quad i = 0, 1, \dots, N - 1$$

Hodnotám M_i se říká *momenty spline-funkce* φ . K problému, jak najít hodnoty momentů M_i z tabulky hodnot funkce f , se vrátíme později. Naším prvním cílem je ukázat, že hledaná spline-funkce φ je svými momenty plně určena a odvodit vzorce pro výpočet a_i, b_i, c_i, d_i , $i = 0, 1, \dots, N - 1$, z čísel M_i , $i = 0, 1, \dots, N$.

Užijeme vztah (4.4.1) z příkl. 4.4, který ještě poněkud upravíme, a zapíšeme $\varphi''(x_i)$ ve tvaru

$$(4.6.4) \quad \begin{aligned} \varphi_i''(x) &= M_i + \frac{M_{i+1} - M_i}{h_i}(x - x_i) = \\ &= M_i \frac{x_{i+1} - x}{h_i} + M_{i+1} \frac{x - x_i}{h_i}, \quad x \in \langle x_i, x_{i+1} \rangle. \end{aligned}$$

[Převodli jsme vlastně interpolační polynom N_1 pro data $\{(x_i, M_i), (x_{i+1}, M_{i+1})\}$ na Lagrangeův tvar, srov. (3.5.1).] Integrací vztahu (4.6.4) dostaneme výraz pro první derivaci φ_i' ,

$$(4.6.5) \quad \varphi_i'(x) = -M_i \frac{(x_{i+1} - x)^2}{2h_i} + M_{i+1} \frac{(x - x_i)^2}{2h_i} + A_i,$$

kde A_i je zatím neznámá integrační konstanta. Podobně odvodíme

$$(4.6.6) \quad \varphi_i(x) = M_i \frac{(x_{i+1} - x)^3}{6h_i} + M_{i+1} \frac{(x - x_i)^3}{6h_i} + A_i(x - x_i) + B_i,$$

kde B_i je další neznámá konstanta (integrační konstantu C_i jsme tentokrát rozepsali jako $C_i = -A_i x_i + B_i$). Hodnoty konstant A_i, B_i , $i = 0, 1, \dots, N - 1$, určíme z toho, že podle interpolačních podmínek (4.5.2) a na základě spojitosti funkce φ musí platit $\varphi_i(x_i) = f(x_i)$, $\varphi_i(x_{i+1}) = f(x_{i+1})$, $i = 0, 1, \dots, N - 1$. Dosazením $x = x_i$ a $x = x_{i+1}$ do (4.6.6) se snadno ověří, že funkce φ budou tyto podmínky splňovat, bude-li

$$(4.6.7) \quad \begin{aligned} A_i &= \frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{h_i}{6}(M_{i+1} - M_i), \\ B_i &= f(x_i) - M_i \frac{h_i^2}{6}. \end{aligned}$$

Tím máme pomocí momentů plně určeny funkce φ_i , $i = 0, 1, \dots, N - 1$ ve tvaru (4.6.6). Pro

koefficienty a_i, b_i, c_i, d_i ve vyjádření (4.6.2) pak ze (4.6.2), (4.6.5), (4.6.4) a (4.6.7) dostáváme

$$(4.6.8) \quad \begin{aligned} a_i &= \varphi_i(x_i) = \varphi(x_i) = f(x_i), \\ b_i &= \varphi'_i(x_i) = -\frac{M - ih_i}{2} + A_i = \\ &= \frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{2M_i + M_{i+1}}{6}h_i, \\ c_i &= \frac{1}{2}\varphi''_i(x_i) = \frac{1}{2}M_i, \\ d_i &= \frac{1}{6}\varphi'''_i(x_i+) = \frac{M_{i+1} - M_i}{6h_i}, \end{aligned}$$

kde $\varphi'''_i(x_i+)$ znamená derivaci v bodě x_i zprava [vypočtenou z momentů derivováním (4.6.4)].

Zatím nevyužité okrajové podmínky (4.5.3) a nevyužitý požadavek spojitosti první derivace φ' použijeme nyní k odvození rovnic pro momenty. V dalším výkladu se omezujeme na okrajové podmínky (4.5.3b) a (4.5.3c). Všimneme si toho, že okrajové podmínky pro přirozenou spline-funkci jsou speciálním případem podmínek (4.5.3b), kde klademe $f''_0 = 0, f''_N = 0$. Stačí tedy další konstrukci popsat pro okrajové podmínky (4.5.3b).

Z podmínek (4.5.3b) především ihned dostáváme

$$(4.6.9) \quad M_0 = f''_0, \quad M_N = f''_N.$$

Spojitost funkce φ' v uzlu $x_i, i = 1, 2, \dots, N-1$, znamená, že má platit $\varphi'_{i-1}(x_i-) = \varphi'_i(x_i+)$ kdy symboly $-$ a $+$ znamenají derivaci v bodě x_i zleva a zprava. Pro $i = 1, 2, \dots, N-1$ dostaneme podle (4.6.5) a (4.6.7)

$$\begin{aligned} \varphi'_{i-1}(x_i-) &= \frac{f(x_i) - f(x_{i-1})}{h_{i-1}} + \frac{h_{i-1}}{3}M_i + \frac{h_{i-1}}{6}M_{i-1}, \\ \varphi'_i(x_i+) &= \frac{f(x_{i+1}) - f(x_i)}{h_i} + \frac{h_i}{3}M_i + \frac{h_i}{6}M_{i+1}, \end{aligned}$$

Tyto dvě veličiny se mají na sobě rovnat, a máme proto

$$(4.6.10) \quad \begin{aligned} \frac{h_{i-1}}{6}M_{i-1} + \frac{h_{i-1} + h_i}{3}M_i + \frac{h_i}{6}M_{i+1} &= \\ = \frac{f(x_{i+1}) - f(x_i)}{h_i} + \frac{f(x_i) - f(x_{i-1})}{h_{i-1}}, \quad i = 1, 2, \dots, N-1, \end{aligned}$$

což je $N-1$ rovnic pro $N-1$ neznámých momentů M_1, M_2, \dots, M_{N-1} . Položíme-li pro $i = 1, 2, \dots, N-1$ ještě

$$(4.6.11) \quad \frac{h_{i-1}}{h_{i-1} + h_i} = \lambda_i, \quad \frac{h_i}{h_{i-1} + h_i} = 1 - \lambda_i = \mu_i,$$

a užijeme (4.6.9), můžeme (4.6.10) zapsat ve tvaru

$$(4.6.12) \quad \begin{array}{rcl} 2M_1 + & \mu_1 M_2 & = g_1 - \lambda_1 f''_0, \\ \lambda_2 M_1 + & 2M_2 & + \mu_2 M_3 = g_2, \\ \vdots & \vdots & \vdots \\ \lambda_{N-2} M_{N-3} + & 2M_{N-2} & + \mu_{N-2} M_{N-1} = g_{N-2}, \\ & \lambda_{N-1} M_{N-2} + & 2M_{N-1} = g_{N-1} - \mu_{N-1} f''_N. \end{array}$$

Matice soustavy lineárních algebraických rovnic (4.6.12) je třídiagonální (viz [21]) a má tu vlastnost, že pro její prvky λ_i, μ_i platí $\lambda_i + \mu_i < 2$ (neboť $\lambda_i + \mu_i = 1$). Dá se ukázat, že taková matice (tzv. ostře diagonálně dominantní matice) je vždy Gaussovou eliminační metodou pro třídiagonální matice [21].

Závěrem shrňme postup při výpočtu interpolační spline-funkce v algoritmické formě:

- (4.6.13) Vstupní data: $N, x_0, x_1, \dots, x_N, f(x_0), f(x_1), \dots, f(x_N)$.
1. Vypočítáme $h_i = x_{i+1} - x_i, i = 0, 1, \dots, N - 1$.
 2. Vypočítáme $\lambda_i, \mu_i, g_i, i = 1, 2, \dots, N - 1$, podle (4.6.11).
 3. Ze soustavy (4.6.12) stanovíme M_1, M_2, \dots, M_{N-1} . Položíme $M_0 = f_0'', M_N = f_N''$.
 4. a) Buď vypočítáme $a_i, b_i, c_i, d_i, i = 0, 1, \dots, N - 1$, podle (4.6.8),
b) nebo vypočítáme $A_i, B_i, i = 0, 1, \dots, N - 1$, podle (4.6.7).
- Výstupní data: Koeficienty spline-funkce φ ve vyjádření (4.6.2) nebo (4.6.6).

Výpočet hodnoty $\varphi(\hat{x})$ spline-funkce φ v bodě \hat{x} provádíme ve dvou krocích:

1. Stanovíme interval $\langle x_i, x_{i+1} \rangle$ takový, že $\hat{x} \in \langle x_i, x_{i+1} \rangle$.
2. Vypočítáme $\varphi(\hat{x}) = \varphi_i(\hat{x})$ podle (4.6.2) nebo (4.6.6).

4.7 Příklad.

Sestrojíme podle algoritmu (4.6.13) přirozenou interpolační spline-funkce pro data z tab. 11 (příkl. 4.4). Je $N = 3, f_0'' = 0, f_3'' = 0$.

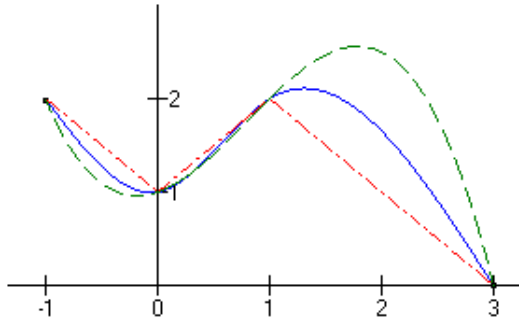
1. Čísla $h_i, i = 0, 1, 2$, jsou již vypočítána v tab. 11.
2. Vypočítáme $\lambda_i, i = 1, 2$, podle (4.6.11) a $\mu_i = 1 - \lambda_i$. Z hodnot v tab. 11 vypočítáme podle (4.6.11) $g_i, i = 1, 2$ (výsledky viz tab. 11).
3. Podle (4.6.12) a tab. 11 dostáváme pro $M_i, i = 1, 2$, soustavu dvou rovnic

$$\begin{aligned} 2M_1 + \frac{1}{2}M_2 &= 6, \\ \frac{1}{3}M_1 + 2M_2 &= -4. \end{aligned}$$

Řešení této soustavy je $M_1 = \frac{84}{23}, M_2 = -\frac{60}{23}$. Dále klademe $M_0 = 0, M_3 = 0$.

4. Stanovíme koeficienty $a_i, b_i, c_i, d_i, i = 0, 1, 2$, ve vyjádření (4.6.2). Užijeme (4.6.8). Dostáváme postupně

$$\begin{aligned} a_0 &= 2, & a_1 &= 1, & a_2 &= 2; \\ b_0 &= \frac{f(x_1) - f(x_0)}{h_0} - \frac{2M_0 + M_1}{6}h_0 = -\frac{37}{23}, \\ b_1 &= \frac{5}{23}, & b_2 &= \frac{17}{23}; \\ c_0 &= 0, & c_1 &= \frac{42}{23}, & c_2 &= -\frac{30}{23}; \\ d_0 &= \frac{14}{23}, & d_1 &= \frac{M_2 - M_1}{6h_1} = -\frac{24}{23}, & d_2 &= \frac{5}{23}. \end{aligned}$$



Obrázek 3:

Hledaná spline-funkce je nyní na intervalu $\langle -1, 3 \rangle$ dána kubickými polynomy $\varphi_0, \varphi_1, \varphi_2$ s koeficienty, které jsme právě určili. Je tudíž

$$\varphi(x) = \begin{cases} 2 - \frac{37}{23}(x+1) + \frac{14}{23}(x+1)^3, & -1 \leq x \leq 0, \\ 1 + \frac{5}{23}x + \frac{42}{23}x^2 - \frac{24}{23}x^3, & 0 \leq x \leq 1, \\ 2 + \frac{17}{23}(x-1) - \frac{30}{23}(x-1)^2 + \frac{5}{23}(x-1)^3, & 1 \leq x \leq 3. \end{cases}$$

Vypočítejme např. $\varphi(2)$. Je $1 \leq 2 \leq 3$, takže

$$\varphi(2) = 2 + \frac{17}{23} - \frac{30}{23} + \frac{5}{23} = \frac{38}{23}.$$

Na obr. 3 je průběh interpolačního polynomu třetího stupně k daným podle příkl. 3.4 (čárkovaná čára), průběh interpolační funkce z příkl. 4.4 (čerchovaná čára) a průběh právě sestrojené přirozené spline-funkce (plná čára).

4.8 Vlastnosti kubických spline-funkcí.

V odstavci 3.32 jsme uvedli, že interpolační polynomy se nemusí blížit k funkci f , jejíž hodnoty interpolujeme, ani když dělení intervalu $\langle a, b \rangle$ body x_i libovolně zjemňujeme. Potíže při interpolaci s více uzly vznikají zejména při ekvidistantním dělení intervalu $\langle a, b \rangle$. O interpolačních kubických spline-funkcích se naproti tomu dá dokázat, že za velmi slabých předpokladů o funkci f a způsobu, jak rozdělujeme interval $\langle a, b \rangle$ na dílčí intervaly, konvergují stejnoměrně k f při $h = \max h_i$ jdoucím k nule. Postačuje k tomu např. existence spojitě první derivace f' na intervalu $\langle a, b \rangle$, či spojitost funkce f a dělení intervalu $\langle a, b \rangle$ ekvidistantním způsobem. Pro dostatečně hladké funkce platí navíc např. toto tvrzení [29], [32]:

4.8.1 Věta.

Nechť $f \in C^4\langle a, b \rangle$ (tj. f a její první čtyři derivace jsou na intervalu $\langle a, b \rangle$ spojité). Budiž K konstanta taková, že při dělení intervalu $\langle a, b \rangle$ na dílčí intervaly délky h_i , $i = 0, 1, \dots, N - 1$, platí $\max h_i / \min h_i \leq K$. Nechť φ je interpolační spline-funkce pro funkci f a nechť φ splňuje okrajové podmínky (4.5.3a) nebo (4.5.3b), kde hodnoty derivací funkce f jsou udány přesně.

Pak pro všechna $x \in \langle a, b \rangle$ a $k = 0, 1, 2, 3$ platí

$$(4.8.1) \quad |f^{(k)}(x) - \varphi^{(k)}(x)| \leq CKh^{4-k},$$

kde $h = \max h_i$ a C je konstanta, která nezávisí na x ani na způsobu dělení intervalu $\langle a, b \rangle$.

Přesnost aproximace spline-funkcemi, v jejichž okrajových podmínkách vystupují pouze přibližné hodnoty derivací, bude ovšem - zejména u krajů intervalu $\langle a, b \rangle$ - nižší. Přirozená spline-funkce sice nevyžaduje žádnou informaci o derivacích interpolované funkce, nedává však u krajů intervalu $\langle a, b \rangle$ aproximaci funkce f s chybou řádově lepší než $O(h^2)$ (a druhé derivace již neaproximuje obecně vůbec). Podrobnější výklad vybočuje v rámci tohoto sešitu a lze jej nalézt v [29], [32].

Kubické spline-funkce se velmi dobře hodí ke grafickému zpracování dat. Mají totiž tu vlastnost, že se málo vlní a dávají křivky, které danými body procházejí "hladce". Přesněji platí [29], [32]: Přirozená interpolační spline-funkce minimalizuje integrál

$$\int_a^b |\psi''(x)|^2 dx$$

ve třídě funkcí ψ , pro něž tento integrál existuje, jejichž první derivace ψ' je na $\langle a, b \rangle$ absolutně spojitá a které splňují interpolační podmínky. Protože křivost křivky $y = \psi(x)$ je při malých hodnotách $\psi'(x)$ přibližně dána hodnotou $\psi''(x)$, je uvedený integrál přibližnou mírou celkové křivosti funkce ψ na intervalu $\langle a, b \rangle$. V tomto smyslu je tedy přirozená spline-funkce "nejhladší" interpolační aproximací. Interpolační spline-funkce splňující jiné okrajové podmínky minimalizují uvedený integrál na jiných, užších třídách funkcí.

4.9 Úlohy.

4.9.1

Přepište Stoerův algoritmus (4.1.3) pro funkci f danou ekvidistantní tabulkou hodnot s krokem h ($x_{i+1} - x_i \equiv h$).

[Návod: Položte $x = x_0 + th$.]

4.9.2

Funkce f je dána daty v tab. 7 (úloha 3.33.7). Nalezněte Stoerovým algoritmem (4.1.3) přibližnou hodnotu $f(1, 5)$. Výsledky porovnejte s výsledkem úlohy 3.33.7.

[Již $R_{11} = \frac{2}{3}$, což je přesná hodnota. Dále $R_{ik} = \frac{2}{3}$, dokud nedojde k dělení nulou při výpočtu R_{33} . Data v tab. 7 jsou funkční hodnoty racionální funkce $f \in \mathcal{R}_{11}$].

4.9.3

a) Sestrojte kubický polynom, který interpoluje funkci $x^{1/4}$ v bodech $x = 0, 1, 16, 81$. Vypočítejte hodnotu tohoto polynomu v bodě $x = 64$. Porovnejte s přesnou hodnotou, která je $2\sqrt{2}$. Pokuste se vysvětlit velikost chyby aproximace.

[Návod: $N_3(x) \doteq x - \frac{7}{120}x(x-1) + 7,1225 \cdot 10^{-4}x(x-1)(x-16)$; ve vzorci pro chybu vystupuje čtvrtá derivace.]

b) Užijte Stoerův algoritmus (4.1.3) k nalezení přibližné hodnoty funkce $x^{1/4}$ v bodě $x = 64$ interpolací v bodech $x = 0, 1, 16, 81$. Porovnejte s výsledkem úlohy a).

[$R_{33} \doteq 2,688033$]

4.9.4

Funkce f je dána svými hodnotami v bodech $x = 0, 1, 2$ a platí $f(0) = 1, f(1) = 2, f(2) = 2$. Nalezněte funkci $\varphi \in \mathcal{R}_{11}$, která interpoluje tato data.

[Návod: Kdyby bylo $b_0 = 0$, nebyla by φ definována pro $x = 0$. Funkci φ proto můžeme hledat ve tvaru $\varphi(x) = (a_0 + a_1x)/(1 + b_1x)$. Výsledek: Úloha není řešitelná. V \mathcal{R}_{11} taková funkce φ neexistuje.]

4.9.5

Popište algoritmus pro konstrukci přirozené interpolační spline-funkce v případě, že vstupní data jsou ekvidistantní. Přepište všechny vzorce, které se použijí během výpočtu. Jaká je nyní matice soustavy (4.6.12)?

[Návod: Položte $h_i = h$.]

4.9.6

Funkce f je dána svými hodnotami v bodech $x = 0, 1, 2, 3$ a je $f(0) = 0, f(1) = 1, f(2) = 0, f(3) = 0$.

a) Sestrojte interpolační polynom třetího stupně pro tato data.

[$N_3(x) = x - x(x-1) + \frac{1}{2}x(x-1)(x-2)$.]

b) Sestrojte přirozenou interpolační spline-funkci pro tato data.

[$M_1 = -3, 6; M_2 = 2, 4$.]

Porovnejte maxima obou interpolačních aproximací na intervalu $\langle 0, 1 \rangle$ a minima obou aproximací na intervalu $\langle 2, 3 \rangle$.

[Na $\langle 0, 1 \rangle$ je $\max N_3(x) \doteq 1,056; \max \varphi(x) \doteq 1,006$. Na $\langle 2, 3 \rangle$ je $\min N_3(x) \doteq -0,316; \min \varphi(x) \doteq -0,154$.]

4.9.7

Funkce f je na intervalu $\langle 0, \pi/2 \rangle$ dána tab. 12. Sestrojte k ní na $\langle 0, \pi/2 \rangle$ interpolační aproximaci, která je po částech polynom prvního stupně.

$[\varphi(x) \doteq 0,97449x$ ($0 \leq x \leq \pi/8$); $0,82614x + 0,05826$ ($\pi/8 \leq x \leq \pi/4$); $0,55201x + 0,27356$ ($\pi/4 \leq x \leq 3\pi/8$); $0,19384x + 0,69552$ ($3\pi/8 \leq x \leq \pi/2$).]

Tabulka 12

x	0	$\pi/8$	$\pi/4$	$3\pi/8$	$\pi/2$
$f(x)$	0,00000	0,382683	0,707107	0,923880	1,000000

4.9.8

Sestrojte přirozenou interpolační spline-funkci pro data z tab. 12.

$[M_1 \doteq -0,40571$; $M - 2 \doteq -0,64389$; $M_3 \doteq -1,20713$.]

4.9.9

Napište program, který realizuje algoritmus (4.6.13) včetně řešení soustavy (4.6.12) a umožňuje při dané tabulce hodnot funkce f počítat hodnoty příslušné přirozené interpolační spline-funkce. Položte $f(x) = (25x^2 + 1)^{-1}$ a pomocí tohoto programu sestrojte interpolační spline-funkci pro funkci f s ekvidistantními uzly interpolace $x_i = 1 + (2/n)i$, $i = 0, 1, \dots, n$. Volte $n = 5, 10, 15, 20$ a tabelujte hodnotu chyby aproximace v bodech $(x_i + x_{i+1})/2$, příp. znázorněte f i sestavenou spline-funkci graficky. Porovnejte s výsledky úlohy 3.33.13.

Tabulka 13

t	-1,000	-0,960	-0,860	-0,790	0,220	0,500	0,930
$f(t)$	-1,000	-0,151	0,894	0,986	0,895	0,500	-0,306

4.9.10

[9] Fiktivní fyzikální experiment dává jako výsledky data z tab. 13; chyba měření je zanedbatelná. Máte sestrojit interpolační aproximaci funkce f pro $t \in \langle -1, 1 \rangle$. Z fyzikálních úvah plyne, že $y = f(t)$ je velmi hladká křivka.

a) Znázorněte měření graficky a položte interpolační křivku podle své intuice od ruky.

b) Sestrojte a graficky znázorněte polynom $N_6(t)$ pro data z tab. 13.

c) Sestrojte k datům z tab. 13 přirozenou interpolační spline-funkci a graficky ji znázorněte. Porovnejte výsledky a), b), c).

5 Ortogonální systémy funkcí

5.1 Lineárně nezávislé systémy funkcí.

V článku 5 se vrátíme k obecné formulaci úlohy o aproximování dané funkce f aproximací φ lineárního typu (1.3.1), sestrojenou jako lineární kombinace základních funkcí $\varphi_0, \varphi_1, \dots, \varphi_n$. Budeme se zabývat systémy základních funkcí, které mají určité speciální vlastnosti, totiž lineárně nezávislémi a ortogonálními systémy. Protože pojmy lineární nezávislosti a ortogonalitě jsou zaváděny a podrobně studovány v teorii vektorových prostorů, algebře a teorii ortogonálních systémů [19], je celý čl. 5 pojat pouze jako přehled výsledků potřebných pro v ostatních částech tohoto sešitu.

Specifikou tohoto článku je skutečnost, že oba pojmy zavedeme ve dvou verzích, spojitě a diskrétní, stejně tak, jako jsme podle způsobu zadání funkce f v čl. 1 rozlišovali nejlepší aproximace ve spojitěm a diskrétním případě. Připomínáme proto, že *spojitý případ* je takový, kdy aproximovaná funkce je zadána jako funkce spojitá na celém intervalu $\langle a, b \rangle$ [pracujeme s hodnotami $f(x)$ ve všech bodech $x \in \langle a, b \rangle$], kdežto *diskrétní případ* se vztahuje na situaci, kdy je funkce f dána jako vektor

$$(5.1.1) \quad \mathbf{tab} f = (f(x_0), f(x_1), \dots, f(x_m))$$

funkčních hodnot v konečné množině tabulkových bodů $x_i \in \langle a, b \rangle$, $i = 0, 1, \dots, m$.

Položme si nejprve otázku, zda systém základních funkcí $\{\varphi_0, \varphi_1, \dots, \varphi_n\}$ neobsahuje zbytečně mnoho funkcí. To by nastalo v případě, že bychom mohli některou z funkcí $\varphi_0, \varphi_1, \dots, \varphi_n$ vynechat a třída \mathcal{V}_n aproximací lineárního typu sestrojených z daného systému základních funkcí existuje funkce φ_s , kterou je možno vyjádřit jako lineární kombinaci ostatních základních funkcí:

$$(5.1.2) \quad \varphi_s(x) = \sum_{\substack{j=0 \\ j \neq s}}^m \alpha_j \varphi_j(x), \quad x \in \langle a, b \rangle.$$

Systém funkcí, pro který na intervalu $\langle a, b \rangle$ platí (5.1.2), se nazývá *lineárně závislý*.

V souladu s touto definicí říkáme, že systém funkcí $\{\varphi_0, \varphi_1, \dots, \varphi_n\}$ je na intervalu $\langle a, b \rangle$ *spojitě lineárně nezávislý*, jestliže rovnost

$$(5.1.3a) \quad \sum_{j=0}^n \alpha_j \varphi_j(x) = 0$$

platí pro všechna $x \in \langle a, b \rangle$ právě jen při $\alpha_0 = \alpha_1 = \dots = \alpha_n = 0$. Podobně řekneme, že systém funkcí $\{\varphi_0, \varphi_1, \dots, \varphi_n\}$ je při daných tabulkových bodech x_i , $i = 0, 1, \dots, m$, *diskrétně lineárně nezávislý*, jestliže rovnost

$$(5.1.3b) \quad \sum_{j=0}^n \alpha_j \mathbf{tab} \varphi_j = \mathbf{0},$$

kde $\mathbf{tab} \varphi_j = (\varphi_j(x_0), \varphi_j(x_1), \dots, \varphi_j(x_m))$, platí právě jen pro $\alpha_0 = \alpha_1 = \dots = \alpha_n = 0$, tj. jsou-li vektory $\mathbf{tab} \varphi_0, \mathbf{tab} \varphi_1, \dots, \mathbf{tab} \varphi_n$ lineárně nezávislé ve smyslu lineární algebry. Z teorie

vektorových prostorů je známo, že při daném počtu $m+1$ tabulkových bodů nemůže mít diskrétně lineárně nezávislý systém funkcí více než $m+1$ prvků.

Z platnosti (5.1.3a) pro všechna $x \in \langle a, b \rangle$ plyne ihned platnost (5.1.3b) při libovolné volbě tabulkových bodů. Systém funkcí, který je lineárně nezávislý v některém diskrétním případě, je proto také spojitě lineárně nezávislý a nespĺňuje pro žádné s vztah (5.1.2). Ze spojitě lineární nezávislosti systému funkcí však ještě neplyne, že systém bude pro danou volbu tabulkových bodů diskrétně lineárně nezávislý.

5.2 Příklady

5.2.1

Systém funkcí $\{1, x, x^2\}$ je na intervalu $\langle -1, 1 \rangle$ spojitě lineárně nezávislý, protože je lineárně nezávislý např. v diskrétním případě daném volbou $x_0 = -1, x_1 = 0, x_2 = 1$. Skutečně, máme pak **tab** $\varphi_0 = (1, 1, 1)$, **tab** $\varphi_1 = (-1, 0, 1)$ a **tab** $\varphi_2 = (1, 0, 1)$, což jsou lineárně nezávislé vektory.

5.2.2

Systém funkcí $\{1, x, x^3\}$ je na intervalu $\langle -1, 1 \rangle$ spojitě lineárně nezávislý, protože je lineárně nezávislý např. v diskrétním případě daném volbou $x_0 = 0, x_1 = \frac{1}{2}, x_2 = 1$. Naproti tomu v diskrétním případě daném volbou $x_0 = -1, x_1 = 0, x_2 = 1$ není systém $\{1, x, x^3\}$ lineárně nezávislý, neboť je **tab** $\varphi_0 = (1, 1, 1)$, **tab** $\varphi_1 = (-1, 0, 1) = \mathbf{tab} \varphi_2$.

5.2.3

Dá se ukázat, že polynomiální systémy (1.3.2) a (1.3.3) jsou lineárně nezávislé na každém intervalu a pro každé n (s tím, že v diskrétním případě musí být $m \geq n$). Podobně se dá ukázat, že trigonometrické systémy (1.3.4) a (1.3.5) jsou lineárně nezávislé na intervalu $\langle 0, 2\pi \rangle$.

5.3 Ortogonální systémy funkcí.

Z analytické geometrie v prostoru je známo, že pojem lineární závislosti vektorů v trojrozměrném euklidovském prostoru znamená, že posuneme-li vektory tak, aby měly společný počáteční bod, budou ležet v rovině. Extrémním případem lineárně nezávislých vektorů jsou vektory vzájemně kolmé, ortogonální. Pojem kolmosti je přitom možno definovat pomocí skalárního součinu vektorů.

V tomto odstavci ukážeme, jak se definuje ortogonalita u obecných systémů funkcí $\{\varphi_0, \varphi_1, \dots, \varphi_n\}$, a podáme příklady ortogonálních systémů. Ortogonální systémy základních funkcí jsou v aproximačních metodách velmi užitečné. V mnoha situacích umožňují zjednodušení výpočtů a jejich použití zpravidla vede na algoritmy, které jsou numericky stabilní. Stejně jako v analytické geometrii začneme i zde definicí skalárního součinu dvou funkcí. Protože v dalším výkladu budeme výjimečně vyšetřovat i spojitě funkce, které na intervalu $\langle a, b \rangle$ nabývají komplexních hodnot, definujeme skalární součin již nyní obecně pro komplexní funkce reálné proměnné.

Skalární součin dvou spojitých funkcí f a g , které jsou definovány na intervalu $\langle a, b \rangle$ a mají tam obecně komplexní hodnoty, budeme označovat (f, g) a definujeme jej vztahem

$$(5.3.1a) \quad (f, g) = \int_1^b f(x) \overline{g(x)} \omega(x) dx \quad (\text{spojitý případ}),$$

kde pruhem označujeme číslo komplexně sdružené a ω je pevně daná váhová funkce, která je spojitá a kladná na intervalu $\langle a, b \rangle$ a integrovatelná na intervalu (a, b) , nebo vztahem

$$(5.3.1b) \quad (f, g) = \sum_{i=0}^m f(x_i) \overline{g(x_i)} \omega_i \quad (\text{diskrétní případ}),$$

kde váhy ω_i , $i = 0, 1, \dots, m$, jsou kladná čísla.

V diskrétním skalárním součinu vystupují pouze hodnoty funkcí f a g v tabulkových bodech a na chování funkcí mimo tyto body hodnoty součinu (f, g) nezávisí.

Skalární součin (5.3.1b) $\omega_i \equiv 1$ jsou-li hodnoty funkcí f a g reálné, je (f, g) totéž jako obvyklý skalární součin vektorů $\mathbf{tab} f$ a $\mathbf{tab} g$ v $(m+1)$ -dimenzionálním euklidovském prostoru \mathcal{R}^{m+1} . Váhy ve skalárních součinech zavádíme ze stejných důvodů jako v odst. 1.5 a 1.7.

Snadno se ukáže (provedte to), že pro takto definované skalární součiny platí (f, g, h jsou funkce, c_1 a c_2 jsou obecně komplexní čísla)

$$(5.3.2) \quad \begin{aligned} a) & (f, g) = \overline{(g, f)}, \\ b) & (c_1 f + c_2 g, h) = c_1 (f, h) + c_2 (g, h), \\ c) & (f, f) \geq 0, \\ d) & (f, f) = 0 \Leftrightarrow \begin{cases} f(x) \equiv 0, & x \in \langle a, b \rangle \quad (\text{spojitý případ}), \\ f(x_i) = 0, & i = 0, 1, \dots, m \quad (\text{diskrétní případ}). \end{cases} \end{aligned}$$

Z platnosti (5.3.2b) vyplývá (důkaz lze provést indukcí), že platí

$$(5.3.3) \quad \left(\sum_{j=0}^n c_j \varphi_j, \varphi_k \right) = \sum_{j=0}^n c_j (\varphi_j, \varphi_k).$$

Vlastnosti c) a d) skalárního součinu nás vedou k tomu, že definujeme pojem *normy funkce* f jako

$$\|f\| = (f, f)^{1/2}$$

Norma funkce má podobné vlastnosti jako absolutní hodnota čísla; z (5.3.2) plyne, že platí

$$(5.3.4) \quad \begin{aligned} a) & \|f\| \geq 0, \\ b) & \|f\| = 0 \Leftrightarrow \begin{cases} f(x) \equiv 0, & x \in \langle a, b \rangle \quad (\text{spojitý případ}), \\ f(x_i) = 0, & i = 0, 1, \dots, m \quad (\text{diskrétní případ}) \end{cases} \\ c) & \|cf\| = |c| \cdot \|f\|, \quad \text{kde } c \text{ je číslo,} \\ d) & \|f + g\| \leq \|f\| + \|g\|. \end{aligned}$$

Je-li v diskrétním případě $\omega_i \equiv 1$, je $\|f\|$ totéž jako $|\mathbf{tab} f|$, tj. délka vektoru $\mathbf{tab} f$.

Budeme říkat, že systém $\{\varphi_0, \varphi_1, \dots, \varphi_n\}$ je (spojitě nebo diskrétně) *ortogonální*, jestliže platí $(\varphi_j, \varphi_k) = 0$ pro $j \neq k$ a $\|\varphi_j\| \neq 0$ pro všechna $j = 0, 1, \dots, n$. Ortogonální systém je vždy lineárně nezávislý. Všimneme si ještě toho, že pojem ortogonalitě závisí na volbě váhové funkce ω či vah $\omega_i, i = 0, 1, \dots, m$ a v diskrétním případě opět na tabulkových bodech $x_i, i = 0, 1, \dots, m$.

Je-li dán nekonečný ortogonální systém funkcí $\{\varphi_0, \varphi_1, \dots\}$, můžeme pro každou funkci f , která je na intervalu $\langle a, b \rangle$ spojitá (příp. i pro obecnější funkce), formálně sestavit nekonečnou řadu

$$(5.3.5) \quad \sum_{j=0}^{\infty} c_j \varphi_j, \quad c_j = \frac{(f, \varphi_j)}{(\varphi_j, \varphi_j)}.$$

Taková řada se nazývá *ortogonální rozvoj* funkce f . Pro jisté ortogonální systémy řada (5.3.5) konverguje (za poměrně velmi mírných požadavků na funkci f) k funkci f a její částečné součty se dají použít jako dobré aproximace funkce f .

5.4 Příklady.

Uvedeme několik příkladů ortogonálních systémů funkcí pro různé váhy, intervaly a příp. i tabulkové body.

5.4.1

Spojitý případ. Funkce

$$\varphi_j(x) = \cos jx, \quad j = 0, 1, \dots,$$

tvoří pro $\langle a, b \rangle = \langle 0, \pi \rangle$ a $\omega(x) \equiv 1$ ortogonální systém. Platí přitom $\|\varphi_0\|^2 = \pi/2$ pro $j > 0$.

Skutečně, pro $j \neq k$ platí [srov. (5.3.1a)]

$$(\varphi_j, \varphi_k) = \int_0^{\pi} \cos jx \cos kx dx = \frac{1}{2} \int_0^{\pi} [\cos(j-k)x + \cos(j+k)x] dx = 0.$$

Výpočet $\|\varphi_j\|^2 = (\varphi_j, \varphi_j)$ přenecháváme čtenáři.

5.4.2

Diskrétní případ. Funkce

$$\varphi_j(x) = \cos jx, \quad j = 0, 1, \dots, n,$$

tvoří pro $\langle a, b \rangle = \langle 0, \pi \rangle$,

$$x_s = \frac{2s+1}{n+1} \frac{\pi}{2}, \quad s = 0, 1, \dots, n,$$

skalární součin tvaru (5.3.1b) s $m = n$ a $\omega_s \equiv 1, s = 0, 1, \dots, n$, ortogonální systém. Platí přitom $\|\varphi_0\|^2 = n+1, \|\varphi_j\|^2 = (n+1)/2$ pro $j > 0$.

Důkaz uvedených tvrzení spočívá v poměrně pracném výpočtu hodnoty skalárního součinu

$$(\varphi_j, \varphi_k) = \sum_{s=0}^n \cos jx_s \cos kx_s$$

(viz např. [5], věta 4.2.4).

5.4.3

Spojité případ, $\omega(x) \equiv 1$. Dá se ukázat (viz úlohu 5.7.3), že funkce a , $\cos x$, $\sin x$, $\cos 2x$, $\sin 2x$, ... tvoří na intervalu $\langle -\pi, \pi \rangle$ ortogonální systém funkcí.

5.4.4

Spojité případ, $\omega(x) \equiv 1$. Funkce

$$\varphi_j(x) = e^{ijx}, \quad i^2 = -1, \quad j = 0, \pm 1, \pm 2, \dots,$$

tvoří na intervalu $\langle -\pi, \pi \rangle$ ortogonální systém funkcí. Platí přitom $\|\varphi_j\|^2 = 2\pi$.

Skutečně, máme

$$(\varphi_j, \varphi_k) = \int_{-\pi}^{\pi} e^{i(j-k)x} dx.$$

Tento integrál se rovná 0 pro $j \neq k$, 2π pro $j = k$.

5.4.5

Diskrétní případ, $\omega_s \equiv 1$. Při volbě

$$x_s = 2\pi \frac{s}{m+1}, \quad s = 0, 1, \dots, m,$$

platí pro funkce $\varphi_j = e^{ijx}$, $j = 0, 1, 2, \dots$, vztahy

$$(\varphi_j, \varphi_k) = \begin{cases} m+1, & \text{jestliže } (j-k)/(m+1) \text{ je celé číslo,} \\ 0, & \text{v ostatních případech.} \end{cases}$$

Skutečně, podle definice skalárního součinu (5.3.1b) máme

$$(\varphi_j, \varphi_k) = \sum_{s=0}^m \exp \left[i(j-k)2\pi \frac{s}{m+1} \right],$$

což je částečný součet geometrické řady s kvocientem $q = \exp[i(j-k)2\pi/(m+1)]$. Je-li $(j-k)/(m+1)$ celé číslo, je $q = 1$ a součet se rovná $m+1$. V ostatních případech dostaneme podle vzorce pro součet členů geometrické posloupnosti, že $(\varphi_j, \varphi_k) = 0$.

5.4.6

Spojitéj případ, $\omega(x) = (1 - x^2)^{-1/2}$. Na intervalu $\langle -1, 1 \rangle$ tvoří funkce

$$\varphi_j(x) = \cos(j \arccos x), \quad j = 0, 1, \dots,$$

ortogonální systém funkcí. Platí přitom $\|\varphi_0\|^2 = \pi$, $\|\varphi_j\|^2 = \pi/2$ pro $j > 0$. Důkaz přenecháváme čtenáři (úloha 5.7.4).

5.5 Ortogonální polynomy.

S ortogonálními systémy funkcí budeme pracovat v čl. 6, 7 a v kap. II. Řadu příkladů ortogonálních systémů trigonometrických funkcí jsme uvedli v předcházejícím odstavci. V odstavcích 5.5 a 5.6 podáme přehled některých v numerické matematice používaných ortogonálních systémů polynomů a popíšeme jejich vlastnosti. Z teorie ortogonálních polynomů vychází celá řada numerických metod v různých oblastech matematiky (např. výpočet integrálu, algebraické úlohy na vlastní čísla). Podrobnější informace o teorii ortogonálních polynomů a jejich použití v numerické matematice a rovněž důkazy tvrzení uvedených v tomto odstavci nalezneme čtenář např. v [2], [5], [16].

Ortogonální systém polynomů můžeme sestavit pro každou váhovou funkci ω a každý interval $\langle a, b \rangle$. Ve spojitém případě dostáváme nekonečný systém polynomů $\{Q_j, j = 0, 1, \dots\}$ takový, že stupeň polynomu Q_j je roven j . Zadaným skalárním součinem je systém $\{Q_j, j = 0, 1, \dots\}$ určen jednoznačně až na to, že koeficienty $A_j, j = 0, 1, \dots$, u nejvyšších mocnin polynomů $Q_j, j = 0, 1, \dots$, můžeme předem zvolit jako libovolná nenulová čísla. V diskrétním případě není ortogonální systém polynomů nekonečný, protože pro tabulku, která má $(m + 1)$ bodů, má polynom Q_{m+1} kořeny ve všech tabulkových bodech $x_i, i = 0, 1, \dots, m$, a jeho diskrétní norma $\left[\sum_{i=0}^m Q_{m+1}^2(x_i) \right]^{1/2}$ je tudíž rovna nule. Ortogonální systém polynomů se tedy v diskrétním případě skládá z polynomů Q_0, Q_1, \dots, Q_m , kde stupeň polynomu Q_j je roven j .

V obou případech je možné ortogonální systémy polynomů postupně konstruovat podle tříčleného rekurentního vzorce

$$(5.5.1) \quad Q_{j+1}(x) = \alpha_j(x - \beta_j)Q_j(x) - \gamma_j Q_{j-1}(x), \quad j = 0, 1, \dots, \quad x \in \langle a, b \rangle,$$

kde $Q_0(x) \equiv A - 0 \neq 0$ a kde formálně klademe $\gamma_0 = 0, Q_{-1}(x) \equiv 0$. Koeficienty v (5.5.1) jsou dány vztahy

$$(5.5.2) \quad \begin{aligned} \alpha_j &= \frac{A_{j+1}}{A_j}, & j &= 0, 1, \dots, \\ \beta_j &= \frac{(xQ_j, Q_j)}{\|Q_j\|^2}, & j &= 0, 1, \dots, \\ \gamma_j &= \frac{\alpha_j \|Q_j\|^2}{\alpha_{j-1} \|Q_{j-1}\|^2}, & j &= 1, 2, \dots, \end{aligned}$$

kde čísla $A_j, j = 0, 1, \dots$, jsou volitelná nenulová čísla.

Vcelku snadno se dá ukázat, že ve spojitém případě je libovolný polynom $p_j \in \mathcal{P}_j$ lineární kombinací ortogonálních polynomů Q_0, Q_1, \dots, Q_j daného ortogonálního systému. Polynom Q_j je tudíž ortogonální nejen k polynomům Q_0, Q_1, \dots, Q_{j-1} nýbrž vůbec ke všem polynomům, jejichž stupeň je menší než j .

V numerické matematice hrají roli také kořeny ortogonálních polynomů. Dá se ukázat, že ortogonální polynom Q_j j -tého stupně sestrojený na intervalu $\langle a, b \rangle$ má j jednoduchých kořenů, které všechny leží v intervalu (a, b) . Dále platí pro $j = 1, 2, \dots$, že mezi každými dvěma polynomu Q_{j+1} leží alespoň jeden kořen polynomu Q_j .

5.6 Příklady

5.6.1

Čebyševovy polynomy (spojitý příklad). Položíme $\langle a, b \rangle = \langle -1, 1 \rangle$, $\omega(x) = (1 - x^2)^{-1/2}$. V příkladu 5.4.6 jsme pro tento případ uvedli ortogonální systém funkcí $\varphi_j(x) = \cos(j \arccos x)$, $j = 0, 1, \dots$. Ale z trigonometrických identit plyne, že funkce φ_j je polynom j -tého stupně v x (položíme $x = \cos t$ a vyjádříme $\cos jt$ jako polynom v $\cos t$). Tyto polynomy se nazývají *Čebyševovy polynomy*. Čebyševův polynom j -tého stupně budeme nadále značit T_j . Dá se ukázat (úloha 5.7.5), že rekurence (5.5.1) má v případě Čebyševových polynomů tvar (viz též [21], příkl. 1.5.5)

$$(5.6.1) \quad \begin{aligned} T_0(x) &\equiv 1, & T_1(x) &= x, \\ T_{j+1}(x) &= 2xT_j(x) - T_{j-1}(x), & j &\geq 1. \end{aligned}$$

Koeficient A_j u nejvyšší mocniny v polynomu T_j je tudíž $A_j = 2^{j-1}$ ($j > 0$). Normy Čebyševových polynomů jsou udány v příkl. 5.4.6.

Uvedme několik prvních Čebyševových polynomů:

$$\begin{aligned} T_0(x) &\equiv 0, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1. \end{aligned}$$

Čebyševovy polynomy jsou střídavě sudé a liché funkce; je

$$T_j(-x) = (-1)^j T_j(x).$$

Protože $T_j(x) = \cos(j \arccos x)$, jsou kořeny polynomu T_j dány vzorcem

$$(5.6.2) \quad x_s = \cos\left(\frac{2s+1}{j} \frac{\pi}{2}\right), \quad s = 0, 1, \dots, j-1.$$

Polynom T_j , $j > 0$ má na intervalu $\langle -1, 1 \rangle$ $j+1$ extrémů, kterých nabývá v bodech $x'_k = \cos(k\pi/j)$, $k = 0, 1, \dots, j$, a platí

$$T_j(x'_k) = (-1)^k, \quad k = 0, 1, \dots, j, \quad |T_j(x)| < 1 \text{ pro } x \neq x'_k.$$

Polynomy T_j mají také velmi významnou minimalizační vlastnost [2], [26]. Pro všechna $j \geq 0$ platí, že polynom $2^{1-j}T_j$ minimalizuje veličinu

$$\max_{x \in \langle -1, 1 \rangle} |p_j(x)|$$

ve třídě všech polynomů $p_j \in \mathcal{P}_j$ takových, že jejich koeficient u mocniny x_j je roven 1.

5.6.2

Čebyševovy polynomy (diskrétní případ). Položíme-li $\langle a, b \rangle = \langle -1, 1 \rangle$, $\omega_i \equiv 1$ a uvažujeme skalární součin (5.3.1b), kde za x_i , $i = 0, 1, \dots, m$, vezmeme kořeny Čebyševova polynomu T_{m+1} [viz (5.6.2)], dá se ukázat, že pro $0 \leq j \leq m$, $0 \leq k \leq m$ platí $(T_j, T_k) = 0$ a $\|T_0\|^2 = m + 1$, $\|T_j\|^2 = (m + 1)/2$ pro $j > 0$ (tyto výsledky plynou z výsledku příkl. 5.4.2, položíme-li $x = \cos t$). Systém $\{T_0, T_1, \dots, T_m\}$ je tudíž při daných tabulkových bodech diskrétně ortogonální.

5.6.3

Legendrové polynomy (spojitý případ). Jsou to ortogonální polynomy pro $\langle a, b \rangle = \langle -1, 1 \rangle$, $\omega(x) \equiv 1$ a jsou definovány vztahem

$$(5.6.3) \quad P_0(x) \equiv 1, \quad P_j(x) = \frac{1}{2^j j!} \frac{d^j}{dx^j} [(x^2 - 1)^j], \quad j = 1, 2, \dots$$

Rekurence (5.5.1) má v jejich případě tvar

$$(5.6.4) \quad \begin{aligned} P_0(x) &\equiv 1, & P_1(x) &= x, \\ P_{j+1}(x) &= \frac{2j+1}{j+1} x P_j(x) - \frac{j}{j+1} P_{j-1}(x). \end{aligned}$$

Uveďme několik Legendrových polynomů:

$$\begin{aligned} P_2(x) &= \frac{1}{2}(3x^2 - 1), \\ P_3(x) &= \frac{1}{2}(5x^3 - 3x), \\ P_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3). \end{aligned}$$

Platí

$$\|P_j\| = \frac{2}{2j+1}, \quad j = 0, 1, \dots,$$

a

$$P_j(-x) = (-1)^j P_j(x).$$

Legendrové polynomy hrají významnou roli při numerickém výpočtu integrálů (čl. 13).

5.6.4

Ortogonalní polynomy pro interval $\langle 0, +\infty \rangle$ a váhovou funkci e^{-x} se nazývají *Laguerrovy polynomy*. Ortogonalní polynomy pro interval $(-\infty, +\infty)$ a váhovou funkci $\omega(x) = \exp(-x^2)$ se nazývají *Hermitovy polynomy*. Jejich definice, rekurentní vztahy a vlastnosti jsou uvedeny např. v [2], [26]. tyto polynomy mají význam pro přibližný výpočet integrálů.

5.6.5

Gramovy polynomy. Je to soustavy diskřétně ortogonálních polynomů pro ekvidistantní tabulku a $\omega_i \equiv 1$, $i = 0, 1, \dots, m$. Gramovy polynomy G_j , $j = 0, 1, \dots, m$, pro tabulkové body $x_i = i - \frac{1}{2}m$, $i = 0, 1, \dots, m$, které jsou rozloženy symetricky kolem počátku s krokem $h = 1$, jsou dány rekurentním vztahem [26]

$$(5.6.5) \quad \begin{aligned} G_0(x) &\equiv 1, & G_1(x) &= x, \\ G_{j+1}(x) &= xG_j(x) - \frac{j^2[(m+1)^2 - j^2]}{4(4j^2 - 1)}G_{j-1}(x), & j &\geq 1. \end{aligned}$$

Podle (5.6.5) postupně dostaneme např.

$$(5.6.6) \quad \begin{aligned} G_2(x) &= x^2 - \frac{1}{12}[(m+1)^2 - 1], \\ G_3(x) &= x^3 - \frac{1}{20}[3(m+1)^2 - 7]x, \\ G_4(x) &= x^4 - \frac{1}{14}[3(m+1)^2 - 13]x^2 + \frac{3}{560}m(m+1)[(m+1)^2 - 9]. \end{aligned}$$

Při daném m lze podle rekurence (5.6.5) ovšem sestavit polynomy G_j i pro $j > m$. Ortogonalní soustavu však tvoří pouze G_0, G_1, \dots, G_m , neboť $G_{m+1}(x_i) = 0$, $i = 0, 1, \dots, m$ (srov. odst. 5.5), a ve smyslu užité diskřétní normy je tedy G_{m+1} nulový polynom. (Ověřte to např. pro G_2 a $m = 1$ nebo G_3 a $m = 2$.)

Gramovy polynomy budeme používat v metodě nejmenších čtverců (čl. 6). Tyto polynomy nemusí mít vždy koeficient u nejvyšší mocniny roven 1 (viz [26]). Gramovy polynomy pro jiné soustavy tabulkových bodů jsou uvedeny v [3], [5].

5.7 Úlohy

5.7.1

Dokažte, že jsou-li tabulkové body x_i , $i = 0, 1, \dots, m$, navzájem různé, jsou funkce $1, x, x^2, \dots, x^m$ diskřétně lineárně nezávislé.

[Návod: Kdyby byly lineárně závislé, existoval by nenulový polynom m -tého stupně, který by měl kořeny ve všech bodech x_i , $i = 0, 1, \dots, m$.]

5.7.2

Vyjádřete $(f, c_1g + c_2h)$ pomocí (f, g) a (g, h) .
[$\bar{c}_1(f, g) + \bar{c}_2(f, h)$.]

5.7.3

Ověřte tvrzení z příkl. 5.4.3.
[Návod: Postupujte podobně jako v příkl. 5.4.1.]

5.7.4

Ověřte tvrzení z příkl. 5.4.6.
[Návod: Položte $x = \cos t$ a užiňte výsledky příkl. 5.4.1.]

5.7.5

Dokažte, že platí (5.6.1).
[Návod: Platí $\cos(j+1)t + \cos(j-1)t = 2 \cos t \cos jt$, $t \geq 1$.]

5.7.6

Sestrojte $T_5(x)$ a $T_6(x)$.
[$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$.]

6 Metoda nejmenších čtverců

6.1 Nejlepší L_2 -aproximace.

Je-li naším úkolem aproximovat funkci danou tabulkou

$$\{(x_i, f(x_i)), \quad i = 0, 1, \dots, m\}$$

a jsou-li hodnoty $f(x_i)$ zatíženy většími chybami (jde např. o naměřené veličiny), není na místě provádět interpolaci, tj. použít aproximující funkci $\varphi \in \mathcal{V}_n$ (viz odst. 1.3), kde $n = m$ a $\varphi(x_i) = f(x_i)$, $i = 0, 1, \dots, m$. Místo toho volíme třídu aproximujících funkcí \mathcal{V}_n s $n < m$ a parametry c_0, c_1, \dots, c_n ve funkci $\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_n\varphi_n(x)$ [srov. (1.3.1)] se snažíme stanovit tak, abychom minimalizovali "souhrnnou odchylku" aproximující funkce φ od zadaných dat. Tak např. prokládáme tabulkou dvaceti hodnot přímkou $\varphi(x) = ax + b$ ($n = 1$). Předpokládáme přitom, že charakter měřené závislosti je možno s dostatečnou přesností vystihnout polynomem prvního stupně.

Definujeme-li "souhrnnou odchylku" aproximace φ od vstupních dat jako [srov. (1.7.4)]

$$(6.1.1) \quad \varrho_{2,\omega}^m(f, \varphi) = \left(\sum_{i=0}^m |f(x_i) - \varphi(x_i)|^2 \omega_i \right)^{1/2},$$

má příslušná nejlepší aproximace, která minimalizuje $\varrho_{2,\omega}^m$ ve třídě \mathcal{V}_n , $n < m$, určité vhodné statistické vlastnosti [18] a vyrovnává vliv náhodných chyb v zadaných (naměřených) hodnotách. Této nejlepší aproximace se říká *nejlepší diskrétní L_2 -aproximace* nebo *aproximace metodou nejmenších čtverců*. Váhy $\omega_i > 0$, $i = 0, 1, \dots, m$, umožňují zdůraznit vliv dat v některých bodech, jsou-li vstupní data udána s různou přesností. Volba $\omega_i = |f(x_i)|^{-1}$ znamená, že budeme posuzovat relativní odchylku aproximace φ od naměřených dat. Častá je ovšem volba $\omega_i \equiv 1$. Poznamenáváme, že stále předpokládáme, že tabulkové body jsou navzájem různé.

Ve spojitém případě se *nejlepší L_2 -aproximace* $\varphi \in \mathcal{V}_n$ hledá tak, aby minimalizovala [srov. (1.5.4)]

$$(6.1.2) \quad \varrho_{2,\omega}^m(f, \varphi) = \left(\int_a^b |f(x) - \varphi(x)|^2 \omega(x) dx \right)^{1/2}.$$

Pro numerické aplikace je však důležitější diskrétní případ nejlepší L_2 -aproximace. Spojitý případ nejlepší L_2 -aproximace má význam spíše pro teorii.

6.2 Normální rovnice.

Hledáme-li ve třídě \mathcal{V}_n nejlepší L_2 -aproximace k datům

$$\{(x_i, f(x_i)), \quad i = 0, 1, \dots, m\}, \quad m > n,$$

znamená to, že máme určit hodnoty parametrů c_0, c_1, \dots, c_n ve funkci φ tak, abychom minimalizovali veličinu $\varrho_{2,\omega}^m(f, \varphi)$. Je zřejmé, že tato úloha je ekvivalentní úloze o minimalizaci druhé mocniny $\varrho_{2,\omega}^m$, tj. veličiny

$$\sum_{i=0}^m |f(x_i) - \varphi(x_i)|^2 \omega_i.$$

Pracujeme-li s reálnými funkcemi, nemusíme zde psát absolutní hodnotu a dostáváme tak veličinu

$$R(f, \varphi) = \sum_{i=0}^m [f(x_i) - \varphi(x_i)]^2 \omega_i.$$

Z matematické analýzy je známo, že nutnou podmínkou pro to, aby veličina $R(f, \varphi)$ jako funkce parametrů c_0, c_1, \dots, c_n nabývala minima, je splnění rovnic

$$(6.2.1) \quad \frac{\partial R(f, \varphi)}{\partial c_k}, \quad k = 0, 1, \dots, n.$$

Dosadíme-li do (6.2.1) za $R(f, \varphi)$, položíme

$$\varphi(x) = \sum_{j=0}^n c_j \varphi_j(x)$$

a vypočítáme parciální derivace podle c_k , $k = 0, 1, \dots, n$, budou mít rovnice (6.2.1) tvar

$$\sum_{j=0}^m \omega_j \left[f(x_i) - \sum_{j=0}^m c_j \varphi_j(x_i) \right] \varphi_k(x_i) = 0, \quad k = 0, 1, \dots, n.$$

Po jednoduché úpravě tuto soustavu $n+1$ lineárních rovnic pro $n+1$ hledaných parametrů c_j , $j = 0, 1, \dots, n$, zapíšeme jako

$$(6.2.2) \quad \sum_{j=0}^m \left[\sum_{i=0}^m \omega_i \varphi_j(x_i) \varphi_k(x_i) \right] c_j = \sum_{i=0}^m \omega_i f(x_i) \varphi_k(x_i), \quad k = 0, 1, \dots, n.$$

Rovnice (6.2.2) se nazývají *normální rovnice*.

Mnohem přehlednější tvar budou tyto rovnice mít, zapíšeme-li je pomocí diskrétního skalárního součinu (5.3.1b). Oba součty přes i v (6.2.2) je možné psát jako skalární součiny a dostáváme tak soustavu lineárních rovnic

$$(6.2.3) \quad \sum_{j=0}^m (\varphi_j, \varphi_k) c_j = (f, \varphi_k), \quad k = 0, 1, \dots, n.$$

V označení odst. 5.3 jsou tyto rovnice nutnou podmínkou minima veličiny $R(f, \varphi) = (f - \varphi, f - \varphi) - \|f - \varphi\|^2$ a také veličiny $\|f - \varphi\|$. Rovnicím (6.2.3) se říká *normální rovnice* i ve spojitém případě a v případě funkcí nabývajících komplexních hodnot.

Nyní je třeba zjistit, zda existuje právě jedno řešení $c_0^*, c_1^*, \dots, c_n^*$ soustavy (6.2.3), a pokud ano, pak zda funkce φ^* daná vztahem

$$\varphi^* = c_0^* \varphi_0(x) + c_1^* \varphi_1(x) + \dots + c_n^* \varphi_n(x)$$

skutečně minimalizuje veličinu $\|f - \varphi\|$ ve třídě \mathcal{V}_n funkcí tvaru

$$\sum_{j=0}^m c_j \varphi_j.$$

Odpověď na tyto otázky podáme v odst. 6.4 a 6.5.

6.3 Příklady.

Popíšeme podrobněji tvar normálních rovnic ve dvou jednoduchých případech. V obou případech jde o diskrétní typ aproximace.

6.3.1

Uvažujme aproximace z třídy \mathcal{V}_0 ($n = 1$) a vezměme $\varphi_0(x) \equiv 1$. Půjde tedy aproximaci $\varphi(x) = c_0 \varphi_0(x) = c_0$, tj. o aproximaci konstantou. Normální rovnice se nám v tomto případě redukuje na rovnici jedinou,

$$(\varphi_0, \varphi_0) c_0 = (f, \varphi_0).$$

Odtud máme

$$c_0 = \frac{(f, \varphi_0)}{(\varphi_0, \varphi_0)},$$

což po rozepsání skalárních součinů podle (5.3.1b) a dosazení za φ_0 dává

$$f(x) \approx \varphi(x) = c_0 = \frac{\sum_{i=0}^m \omega_i f(x_i)}{\sum_{i=0}^m \omega_i}.$$

Říkáme, že c_0 je *vážený průměr* hodnot $f(x_0), f(x_1), \dots, f(x_m)$.

Je-li $\omega_i \equiv 1$, dostáváme

$$c_0 = \frac{\sum_{i=0}^m f(x_i)}{m+1}$$

Výpočet průměru je tedy speciálním případem metody nejmenších čtverců.

6.3.2

Nechť nyní $n = 1$ a uvažujme aproximace z \mathcal{V}_1 , kde $\varphi_0 = 1$, $\varphi_1(x) = x$. Bude tedy $\varphi(x) = c_0 + c_1(x)$. Volme pro jednoduchost $\omega_i \equiv 1$. Normální rovnice jsou v tomto případě

$$(6.3.1) \quad \begin{aligned} (\varphi_0, \varphi_0)c_0 + (\varphi_1, \varphi_0)c_1 &= (f, \varphi_0), \\ (\varphi_0, \varphi_1)c_0 + (\varphi_1, \varphi_1)c_1 &= (f, \varphi_1) \end{aligned}$$

neboli

$$(6.3.2) \quad \begin{aligned} (m+1)c_0 + \left(\sum_{i=0}^m x_i\right)c_1 &= \sum_{i=0}^m f(x_i), \\ \left(\sum_{i=0}^m x_i\right)c_0 + \left(\sum_{i=0}^m x_i^2\right)c_1 &= \sum_{i=0}^m x_i f(x_i). \end{aligned}$$

6.4 Existence nejlepší L_2 -aproximace.

Dosud jsme se nezabývali existencí a jednoznačností řešení soustavy (6.2.3) ani tím, zda takto získané řešení skutečně minimalizuje veličinu $R(f, \varphi) = \|f - \varphi\|$ [podmínky (6.2.1) jsou obecně pouze nutné podmínky pro extrém]. Dá se ukázat, že při lineárně nezávislém systému $\{\varphi_j, j = 0, 1, \dots, n\}$ (viz odst. 5.1) má úloha o nejlepší L_2 -aproximaci vždy právě jedno řešení, které je možno nalézt postupem uvedeným v odst. 6.2. Přesněji toto tvrzení formulujeme v následující větě, která ve spojitém i diskrétním případě. Důkaz věty 6.5 lze nalézt v [5]; podobná věta je dokázána také v [2].

6.5 Věta.

Je-li systém základních funkcí $\{\varphi_j, j = 0, 1, \dots, n\}$ lineárně nezávislý, pak existuje právě jedna aproximace $\varphi^* \in \mathcal{V}_n$,

$$\varphi^*(x) = \sum_{j=0}^n c_j^* \varphi_j(x),$$

kteřá minimalizuje ve třídě \mathcal{V}_n veličiny $\|f - \varphi\|$ a $\|f - \varphi\|^2$. Koeficienty c_j^* , $j = 0, 1, \dots, n$, se najdou jako jediné řešení soustavy normálních rovnic (6.2.3).

V diskrétním případě s tabulkovými body x_i , $i = 0, 1, \dots, m$, dostáváme při $n = m$ interpolační aproximaci, pro niž platí

$$(6.5.1) \quad \sum_{j=0}^m c_j^* \varphi_j(x_i) = f(x_i), \quad i = 0, 1, \dots, m.$$

Taková aproximace existuje ve \mathcal{V}_m právě jedna.

6.6 Příklady.

Sestrojíme ve spojitém i diskrétním případě podle věty 6.5 nejlepší L_2 -aproximaci funkce \sqrt{x} na intervalu $(0, 1)$ polynomem prvního stupně. Bude tedy $n = 1$, $\varphi(x) = c_0 + c_1x$; volíme $\omega(x) \equiv 1$, $\omega_i \equiv 1$.

6.6.1

Spojitý případ (viz příkl. 1.6). Je $\varphi_0(x) \equiv 1$, $\varphi_1(x) = x$ a soustava normálních rovnic (6.2.3) má tvar (6.3.1). Protože

$$(f, g) = \int_0^1 f(x)g(x)dx,$$

máme

$$(\varphi_0, \varphi_0) = 1, \quad (\varphi_1, \varphi_0) = (\varphi_0, \varphi_1) = \frac{1}{2}, \quad (\varphi_1, \varphi_1) = \frac{1}{3}.$$

Dále je

$$(f, \varphi_0) = \int_0^1 \sqrt{x}dx = \frac{2}{3}, \quad (f, \varphi_1) = \int_0^1 x\sqrt{x}dx = \frac{2}{5}.$$

Soustava normálních rovnic má tudíž tvar

$$\begin{aligned} c_0 + \frac{1}{2}c_1 &= \frac{2}{3}, \\ \frac{1}{2}c_0 + \frac{1}{3}c_1 &= \frac{2}{5}. \end{aligned}$$

Řešení je $c_0 = \frac{4}{15}$, $c_1 = \frac{4}{5}$ a hledaná nejlepší L_2 -aproximace polynomem prvního stupně je

$$\varphi(x) = \frac{4}{15} + \frac{4}{5}x$$

6.6.2

Diskrétní případ (viz příkl. 1.8). Zvolíme $m = 2$ a $x_0 = 0$, $x_1 = \frac{1}{2}$, $x_2 = 1$. Je potom

$$f(x_0) = 0, \quad f(x_1) = (\sqrt{2})/2, \quad f(x_2) = 1.$$

Normální rovnice pro aproximace z \mathcal{V}_1 a diskrétní případ jsme již rozepsali v příkl. 6.3.2. Dosadíme-li do (6.3.2) za m a potřebné součty, dostaneme pro koeficienty c_0 , c_1 soustavu rovnic

$$\begin{aligned} 3c_0 + \frac{3}{2}c_1 &= \frac{2 + \sqrt{2}}{2}, \\ \frac{3}{2}c_0 + \frac{5}{4}c_1 &= \frac{4 + \sqrt{2}}{4}. \end{aligned}$$

Řešení této soustavy je

$$c_0 = \frac{1}{6}(\sqrt{2} - 1), \quad c_1 = 1$$

a hledaná nejlepší diskrétní L_2 -aproximace polynomem prvního stupně je

$$\varphi(x) = \frac{1}{6}(\sqrt{2} - 1) + x.$$

Ukážeme ještě, že pro $m = n = 1$ dostaneme metodou nejmenších čtverců právě interpolační polynom prvního stupně. Volíme $x_0 = 0$, $x_1 = 1$, a je tudíž $f(x_0) = 0$, $f(x_1) = 1$. Soustava (6.3.2) má v tomto případě tvar $2c_0 + c_1 = 1$, $c_0 + c_1 = 1$ a řešení je $c_0 = 0$, $c_1 = 1$. Metoda nejmenších čtverců tedy dává $\varphi(x) = x$, což je interpolační polynom pro \sqrt{x} a uzly $x_0 = 0$, $x_1 = 1$.

6.7 Řešení normálních rovnic a volba základních funkcí.

Matice (a_{kj})

$$a_{kj} = (\varphi_j, \varphi_k), \quad j, k = 0, 1, \dots, n,$$

soustavy normálních rovnic (6.2.3) je hermitovská (platí $a_{kj} = \bar{a}_{jk}$), v případě reálných funkcí symetrická a pro lineárně nezávislý systém funkcí dokonce pozitivně definitní (viz např. [3]). Takové soustavy se obecně dají bez problémů řešit Gaussovou eliminační metodou bez výběru hlavního prvku. Speciální metodou pro pozitivně definitní symetrické matice, která je úspornější co do počtu operací i nároků na paměť počítače, je Choleského varianta LU-rozkladu matice [21].

Podmíněnost soustavy normálních rovnic, a tím i stabilita celého algoritmu metody nejmenších čtverců, však výrazně závisí na volbě základních funkcí φ_j , $j = 0, 1, \dots, n$. Omezíme se zde na nejlepší L_2 -aproximace algebraickými polynomy, kde se ukazuje že při klasické volbě $\varphi_j(x) = x^j$, $j = 0, 1, \dots, n$, jsou soustavy normálních rovnic při větším stupni aproximujícího polynomu (zhruba > 5) často špatně podmíněné. Řešení normálních rovnic je pak značně citlivé na vliv chyb ve

vstupních datech a chyb ze zaokrouhlování během výpočtu. Tato vlastnost se velmi silně projevuje zejména při ekvidistantních tabulkových bodech, kde již při $n \approx 9$ je (při výpočtu v jednoduché aritmetice) obtížné dostat výsledky, které by byly vůbec použitelné [26].

S volbou základních funkcí souvisí také volba jejich počtu n , tedy volba stupně polynomu, hledáme-li aproximaci polynomiálního tvaru. Zvyšování stupně aproximujícího polynomu nepřináší totiž vždy zvýšení přesnosti aproximace. Je zřejmé, že volbou $n = m$, tj. interpolací, můžeme dosáhnout toho, aby $\varrho_2^m(f, \varphi) = 0$. Ale v metodě nejmenších čtverců vycházíme z jiných předpokladů než při interpolaci. Naším základním předpokladem zde je, že funkce f může být z hlediska praxe považována za polynom N -tého stupně, kde $N \ll m$, přičemž hodnotu N předem neznáme, naši úlohou je ji rovněž určit. Zvolíme-li naproti tomu $n > N$, ztratíme dobré vlastnosti nejlepší L_2 -aproximace co do hladkosti a vyrovnávání náhodné rozložených chyb ve vstupních datech.

Nemáme-li předem informaci o vhodné volbě n , doporučují statistické úvahy [26] řešit normální rovnice postupně pro $n = 0, 1, 2, \dots$, počítat navíc hodnotu

$$\sigma_n^2 = \frac{[\varrho_2^m(f, \varphi)]^2}{m - n}$$

a pokračovat tak dlouho, dokud σ_n^2 s rostoucím n významně klesá. Hodnota n , po níž nenásleduje významný pokles σ_n^2 , je ze statistických důvodů vhodným stupněm N polynomu nejlepší L_2 -aproximace. Při volbě $\varphi_j(x) = x^j$ takový postup z početního hlediska znamená, že musíme stanovit řešení normálních rovnic pro každý stupeň n polynomu nejlepší aproximace znovu, protože hodnoty koeficientů polynomů q -tého stupně, $q < n$, se k výpočtu řešení normálních rovnic pro stupeň n použít nedají. To je další nevýhoda uvedené volby základních polynomů φ_j .

Oba nepříjemné důsledky volby $\varphi_j(x) = x^j$, $j = 0, 1, \dots, n$, odpadají, vezmeme-li za základní funkce φ_j ortogonální polynomy (třída aproximujících funkcí se tím nemění!). Je-li totiž systém základních funkcí ortogonální, tj. platí-li

$(\varphi_j, \varphi_k) = 0$ pro $j \neq k$, je matice soustavy normálních rovnic (6.2.3) diagonální a můžeme rovnou psát

$$(6.7.1) \quad c_j = \frac{(f, \varphi_j)}{(\varphi_j, \varphi_j)}, \quad j = 0, 1, \dots, n.$$

Koeficienty c_j v tomto případě nezávisí na n , a lze tedy postupně zvyšovat počet parametrů ve funkci φ (tj. stupeň polynomu, jsou-li základní funkce ortogonální polynomy) tak, že pokaždé pouze dopočítáme jeden potřebný koeficient. Všimneme si také toho, že částečné součty ortogonálních rozvoju funkce f , o kterých jsme se zmiňovali v odst. 5.3, jsou vlastně nejlepší L_2 -aproximace dané funkce [srov. (6.7.1) a (5.3.5)].

Nemáme-li v diskrétním případě k dispozici vhodný systém ortogonálních polynomů předem, bývá v praxi výhodné nesestrojovat tyto polynomy v uzavřeném tvaru, ale pomocí rekurentních vztahů (5.5.1) postupně počítat vektory $\mathbf{tab} Q_j$, $j = 0, 1, \dots, n$, hodnot těchto polynomů v tabulkových bodech. To je totiž to jediné, co v diskrétním případě potřebujeme pro výpočty skalárních součinů. Výpočet hodnoty aproximující funkce φ v bodě $x = \hat{x}$ můžeme při použití ortogonálních polynomů vždy založit rovněž na (5.5.1). Výsledkem je *Clenshawova rekurentní formule*: Použijeme označení z odst. 5.5, položíme $y_{n+2} = y_{n+1} = 0$ a pro $k = n, n-1, \dots, 1, 0$ postupně vypočítáme

$$(6.7.2) \quad y_k = \alpha_k(\hat{x} - \beta_k)y_{k+1} - \gamma_{k+1}y_{k+2} + c_k.$$

Dá se ukázat (úloha 6.10.2), že platí

$$A_0 y_0 = \varphi(\hat{x}) = \sum_{j=0}^n c_j Q_j(\hat{x}).$$

V odstavci 5.3 jsme popsali dva systémy ortogonálních polynomů vhodné pro užití v diskrétní verzi metody nejmenších čtverců. Jsou to především Čebyševovy polynomy (příkl. 5.6.2) pro tabulkové body rozložené jako kořeny těchto polynomů. Teoretické úvahy i praktické zkušenosti ukazují, že užití těchto ortogonálních polynomů je z mnoha hledisek optimální. Pro ekvidistantní tabulkové body jsme v příkl. 5.6.5 popsali Gramův systém ortogonálních polynomů. Zde je třeba mít na paměti, že teorie i praxe nedoporučují používat tyto polynomy pro $n > 2\sqrt{m}$. Gramovy polynomy vyšších stupňů mají mezi tabulkovými body velké oscilace.

I v metodě nejmenších čtverců se stejně jako při interpolaci může ukázat, že zadaná funkce ϵ se nehodí pro aproximaci polynomem. V takovém případě je třeba přejít k jiným systémům základních funkcí nebo hledat aproximaci nelineárního tapu. Osvědčuje se také užití spline-funkcí. Podrobnosti je možno nalézt ve speciální literatuře, např. v [17]. Poznamenáváme také, že pro konstrukci nejlepší L_2 -aproximace se stále častěji používají algoritmy založené na singulárním rozkladu matic [21]. Popis těchto poměrně komplikovaných, ale velmi stabilních algoritmů však přesahuje rámec tohoto sešitu (viz opět [17] a [9]).

6.8 Příklad.

Použijeme Gramovy polynomy a sestrojíme metodou nejmenších čtverců polynomy prvního až třetího stupně aproximující funkci f danou tab. 14.

Tabulka 14

x	-2	-1	0	1	2
$f(x)$	-1	-1	0	1	1

Protože o datech nemáme žádnou další informaci, volíme $\omega_i \equiv 1$. Gramovy polynomy potřebných stupňů jsou sestrojeny v příkl. 5.6.5. Je nyní $m = 4$ a podle (5.6.5) a (5.6.6) máme

$$G_0(x) \equiv 1, \quad G_1(x) = x, \quad G_2(x) = x^2 - 2, \quad G_3(x) = x^3 - \frac{17}{3}x.$$

Je

$$\begin{aligned} \mathbf{tab} G_0 &= (1, 1, 1, 1, 1), \\ \mathbf{tab} G_1 &= (-2, -1, 0, 1, 2), \\ \mathbf{tab} G_2 &= (2, -1, -2, -1, 2), \\ \mathbf{tab} G_3 &= \left(-\frac{6}{5}, \frac{12}{5}, 0, -\frac{12}{5}, \frac{6}{5}\right). \end{aligned}$$

Skalární součin (f, g) je nyní obvyklým skalárním součinem vektorů $\mathbf{tab} f$ a $\mathbf{tab} g$. Máme tudíž

$$\begin{aligned} c_0 &= \frac{(f, G_0)}{(G_0, G_0)} = 0, \\ c_1 &= \frac{(f, G_1)}{(G_1, G_1)} = \frac{6}{10} = \frac{3}{5}, \\ c_2 &= \frac{(f, G_2)}{(G_2, G_2)} = 0, \\ c_3 &= \frac{(f, G_3)}{(G_3, G_3)} = -\frac{1}{6}. \end{aligned}$$

Nejlepší diskrétní L_2 -aproximace polynomem prvního stupně je tedy

$$\varphi^*(x) = c_0\varphi_0(x) + c_1\varphi_1(x) = \frac{3}{5}x.$$

Protože $c_2 = 0$, je to i nejlepší aproximace polynomem druhého stupně. Nejlepší aproximace polynomem třetího stupně je

$$\varphi^*(x) + c_3\varphi_3(x) = \frac{1}{6}(7x - x^3);$$

lze se přesvědčit, že tato aproximace je již zároveň interpolačním polynomem pro data z tab. 14.

6.9 Algoritmus metody nejmenších čtverců.

Popíšeme obecný algoritmus metody nejmenších čtverců (diskrétní případ) vhodný zejména pro výpočty na počítači. V použitém systému ortogonálních polynomů (který postupně generujeme v průběhu výpočtu) volíme $A_j \equiv 1$. V rekurenci (5.5.1) pak máme $\alpha_j \equiv 1$. Stupeň aproximujícího polynomu φ zvyšujeme tak dlouho, dokud při zvýšení stupně poklesne hodnota σ_n^2 alespoň o Δ (vstupní parametr algoritmu). Algoritmus využívá toho, že k výpočtu skalárních součinů a norem ortogonálních polynomů potřebujeme pouze hodnoty $\mathbf{tab} Q_j$, $j = 0, 1, \dots$, a k výpočtu σ_n^2 pouze $\mathbf{tab} f$, $\mathbf{tab} \varphi$.

(6.9.1) Vstup: $m, x_0, x_1, \dots, x_m, \mathbf{tab} f, \omega_0, \omega_1, \dots, \omega_m, \Delta$.

tab $Q_0 = (1, 1, \dots, 1)$;

$v_0 = \|Q_0\|^2 = \sum_{i=0}^m \omega_i$;

$c_0 = (f, Q_0)/v_0; \quad \beta_0 = \left(\sum_{i=0}^m x_i \omega_i \right) / v_0$;

tab $Q_1 = (x_0 - \beta_0, x_1 - \beta_0, \dots, x_m - \beta_0)$;

$v_1 = \|Q_1\|^2 = \sum_{i=0}^m \omega_i (x_i - \beta_0)^2$;

$c_1 = (f, Q_1)/v_1$;

tab $\varphi = \mathbf{tab} (c_0 + c_1 Q_1); \quad \sigma^2 = \|f - \varphi\|^2 / (m + 1)$;

$s = 0$.

START: $s = s + 1$;

$\beta_s = (x Q_s, Q_s)/v_s; \quad \gamma_s = v_s/v_{s-1}$.

 Pro $i = 0, 1, \dots, m$:

$Q_{s+1}(x_i) = (x_i - \beta_s)Q_s(x_i) - \gamma_s Q_{s-1}(x_i)$,

$v_{s+1} = \|Q_{s+1}\|^2; \quad c_{s+1} = (f, Q_{s+1})/v_{s+1}$;

tab $\varphi = \mathbf{tab} \varphi + c_{s+1} \mathbf{tab} Q_{s+1}$.

 Jestliže $s + 1 = m$, pak se nepodařilo určit vhodný
 stupeň aproximujícího polynomu. Přeruš výpočet.

$\delta^2 > \sigma^2 - \Delta$, jdi na KONEC.

$\sigma^2 = \delta^2$; jdi na START.

KONEC: $n = s$.

 Výstup: $n, \beta_0, \beta_1, \dots, \beta_{n-1}, \gamma_1, \gamma_2, \dots, \gamma_{n-1}, c_0, c_1, \dots, c_n$.

Po provedení algoritmu (6.9.1) můžeme počítat hodnoty aproximujícího polynomu φ pomocí Clenshawovy rekurentní formule (6.7.2). Potřebné konstanty jsou výstupními parametry algoritmu (6.9.1).

6.10 Úlohy

6.10.1

Napište soustavu normálních rovnic pro obecné n , libovolně rozložené tabulkové body, $\omega_i \equiv 1$ a volbu $\varphi_j(x) = x^j, j = 0, 1, \dots, n$.

[Návod: Vyjděte z tvaru (6.2.2).]

6.10.2

Ukažte, že je-li y_0 hodnota získaná jako výsledek použití Clenshawovy rekurentní formule (6.7.2) pro $k = n, n - 1, \dots, 1, 0$, platí

$$A_0 y_0 = \varphi(\hat{x}) = \sum_{j=0}^n c_j Q_j(\hat{x}),$$

kde $\{Q_0, Q_1, \dots, Q_n\}$ je systém ortogonálních polynomů, pro něž platí rekurence (5.5.1) se stejnými koeficienty, jako vystupují v (6.7.2).

[Návod: Vyřešte (6.7.2) vzhledem k c_k a dosaďte tato hodnoty do výrazu pro $\varphi(\hat{x})$. Dále přepište tento výraz na tvar $\varphi(\hat{x}) = \sum_{k=0}^n y_k \{\dots\}$ a výrazy ve složených závorkách posuďte na základě vlastností ortogonálních polynomů.]

6.10.3

Řešte úlohu z příkl. 6.8 při volbě $\varphi_j(x) = x^j$. Porovnejte výsledky.
[Stejně výsledky. Pro $n = 1$ můžeme užít (6.3.2).]

6.10.4

Funkce f je dána svými hodnotami v bodech $x_i = i$, $i = 1, 2, \dots, 5$. Je přitom $f(1) = 0$, $f(2) = 2$, $f(3) = 2$, $f(4) = 5$, $f(5) = 4$. Volte $\omega_i \equiv 1$, $\varphi_j(x) = x^j$, $j = 0, 1$, a aproximujte funkci f metodou nejmenších čtverců polynomem prvního stupně.

$$[\varphi(x) = \frac{1}{10}(-7 + 11x).]$$

6.10.5

Metodou nejmenších čtverců najděte polynom prvního stupně, který aproximuje data u tab. 15. Volte $\omega_i \equiv 1$ a) Užijte (6.3.2). b) Proveďte vhodnou transformaci nezávislé proměnné a užijte Gramovy polynomy podobně jako v příkl. 6.8

$$[\varphi(x) \doteq 2,97 - 2x.]$$

Tabulka 15

x	0,0	0,1	0,2	0,3	0,4	0,5	0,6
$f(x)$	2,9	2,8	2,7	2,3	2,1	2,1	1,7

6.10.6

Funkce f je dána svými hodnotami v bodech $x = 1, 3, 4, 6, 7$. Je přitom $f(1) = -2, 1, f(3) = -0, 9, f(4) = -0, 6, f(6) = 0, 6, f(7) = 0, 9$. Volte $\omega_i \equiv 1, j = 0, 1$, a aproximujte funkci f metodou nejmenších čtverců polynomem prvního stupně.

[$\varphi \doteq 0, 5053x - 2, 542$.]

6.10.7

Metodou nejmenších čtverců najděte polynom druhého stupně, který aproximuje data z tab. 16. Volte $\omega_i \equiv 1$. a) Volte $\varphi_j(x) = x^j, j = 0, 1, 2$, a řešte normální rovnice (6.2.2). b) Volte za φ_j Gramovy polynomy z příkl. 5.6.5. [$\varphi(x) = \frac{1}{84}(56 - 117x - 11x^2)$.]

Tabulka 16

x	-3	-2	-1	0	1	1	3
$f(x)$	4	2	3	0	-1	-2	-5

6.10.8

Metodou nejmenších čtverců s $\varphi_j(x) = x^j$ aproximujte funkci f danou na intervalu $\langle 0, 1 \rangle$ hodnotami v bodech $x_i = i/100, i = 0, 1, \dots, 100$. Položte $f(x_i) = \exp(x_i)$. Volte $\omega_i \equiv 1$ a najděte aproximaci polynomem pátého, desátého a patnáctého stupně. Posuďte přesnost řešení soustavy normálních rovnic.

7 Aproximace trigonometrickými polynomy

7.1 Fourierovy řady.

Mnohé jevy v přírodě a technice mají periodický charakter. Základem matematického studia periodických jevů je důležitá věta pocházející od J. Fouriera, podle níž je možno za značně obecných předpokladů rozložit reálnou periodickou funkci f o periodě $2\pi/\omega$ v řadu

$$(7.1.1) \quad f(x) = \sum_{k=0}^{\infty} r_k \sin(k\omega x + v_k).$$

Tento rozvoj se často zapisuje v jiných, ekvivalentních tvarech. Položíme-li $a_k = r_k \sin v_k, b_k = r_k \cos v_k$ dostáváme

$$(7.1.2) \quad f(x) = \sum_{k=0}^{\infty} (a_k \cos k\omega x + b_k \sin k\omega x).$$

Koeficienty a_k, b_k jsou reálná čísla. Z hlediska snadnosti manipulace s rozvoji periodických funkcí je však výhodnější pracovat v oboru komplexních čísel. Použijeme-li známé vzorce

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}, \quad \sin x = \frac{e^{ix} - e^{-ix}}{2i},$$

kde $i^2 = -1$, a položíme-li

$$(7.1.3) \quad c_0 = a_0, \quad c_k = \frac{1}{2}(a_k - ib_k), \quad c_{-k} = \frac{1}{2}(a_k + ib_k), \quad k = 1, 2, \dots,$$

dostáváme

$$(7.1.4) \quad f(x) = \sum_{k=-\infty}^{+\infty} c_k e^{ik\omega x}.$$

S takovými řadami pak budeme pracovat i v případě funkcí f , které nabývají komplexních hodnot. Řadám (7.1.2) a (7.1.4) se říká *Fourierovy řady*. Jejich částečné součty jsou *trigonometrické polynomy*.

V tomto článku se budeme nadále zabývat pouze periodickými funkcemi s periodou 2π (tj. $\omega = 1$). Předpokládáme, že funkce mohou nabývat komplexních hodnot, a z tohoto důvodu poněkud porušíme dosud používaný způsob značení. V celém tomto článku znamená symbol i imaginární jednotku, tj. platí $i^2 = -1$. Jako index se zde i nebude vyskytovat. Výsledky teorie Fourierových řad nás vedou k myšlence, že pro aproximování periodických funkcí o periodě 2π bude vhodné volit jako základní funkce (viz odst. 1.3) systém $\{1, e^{ix}, e^{i2x}, \dots, e^{inx}\}$, příp. systém $\{1, \cos x, \sin x, \dots, \cos Lx, \sin Lx\}$.

V odstavci 5.4 jsme ukázali, že oba uvedené systémy jsou spojitě ortogonální na intervalu $\langle -\pi, \pi \rangle$ s vahou $\omega(x) \equiv 1$. Dále jsme ukázali, že pro tabulkové body $x_s = 2\pi s/(m+1)$, $s = 0, 1, \dots, m$, a váhy $\omega_2 \equiv 1$ je první z uvedených systémů také diskrétně ortogonální (pokud ovšem $n \leq m$). V dalším výkladu popíšeme metodu nejmenších čtverců používající uvedené systémy základních funkcí. V diskrétním případě se budeme zabývat také interpolací periodických funkcí. Důležitou výhodou obou popsaných systémů je skutečnost, že se v těchto případech koeficienty c_0, c_1, \dots, c_n hledané aproximace φ lineárního typu dají zapsat vzorci v uzavřeném tvaru.

7.2 Věta. Fourierova analýza – spojitý případ.

Pro každou spojitou periodickou funkci f o periodě 2π se dá sestavit Fourierova řada tvaru (7.1.2) nebo (7.1.4). Koeficienty těchto řad jsou dány vzorci

$$(7.2.1) \quad a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx, \quad b_0 = 0,$$

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx dx, \quad b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx dx, \quad k > 1,$$

a

$$(7.2.2) \quad c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx} dx.$$

Jsou-li funkce f a f' spojité pro všechna x , pak Fourierovy řady s těmito koeficienty konvergují pro všechna x a jejich součet je $f(x)$. Mají-li f a f' v každé periodě konečný počet skoků, konverguje Fourierova řada k $[f(x-) + f(x+)]/2$, kde $f(x+)$, resp. $f(x-)$ označuje limitu funkce f v bodě x zprava, resp. zleva.

Částečné součty Fourierových řad jsou přitom nejlepšími L_2 -aproximacemi funkce f trigonometrickými polynomy o příslušném počtu koeficientů.

Důkaz tvrzení o konvergenci lze nalézt (dokonce za slabších předpokladů) v učebnicích matematické analýzy (např. [12]). Výsledky týkající se metody nejmenších čtverců plynou z výsledků čl. 6. Pro ortogonální systém základních funkcí $\varphi_k(x) = e^{ikx}$ máme totiž podle (6.7.1) a výsledků příkl. 5.4.4

$$c_k = \frac{(f, \varphi_k)}{(\varphi_k, \varphi_k)} = (2\pi)^{-1} \int_{-\pi}^{\pi} f(x) e^{-ikx} dx,$$

což je právě (7.2.2). Vztahy (7.2.1) lze ověřit obdobně,

7.3 Přibližný výpočet Fourierových koeficientů.

Koeficienty a_k , b_k , c_k v (7.2.1) a (7.2.2) se nazývají *Fourierovy koeficienty*. Platí pro ně vztahy (7.1.3) a obrácené vztahy

$$(7.3.1) \quad \begin{aligned} a_0 &= c_0, & b_0 &= 0, \\ a_k &= c_k + c_{-k}, & b_k &= i(c_k - c_{-k}), \quad k > 1. \end{aligned}$$

Je-li f reálná funkce, jsou koeficienty a_k , b_k také reálné.

Při řešení úloh Fourierovy analýzy je našim úkolem nalézt hodnoty Fourierových koeficientů dané funkce (pro $k \leq K$). Numericky tuto úlohu řešíme jedním z těchto dvou způsobů:

a) Integrály (7.2.1), (7.2.2) počítáme některou z metod popsaných v kap. II. Celý integrand [např. $f(x) \cos kx$] přitom chápeme jako jedinou periodickou funkci. K výpočtu takových integrálů se velmi dobře hodí lichoběžníkové pravidlo (viz odst. 11.1.2 a 12.3). Při větších k integrand silně osciluje a dochází k potížím při výpočtu. Existují různé speciální vzorce pro numerický výpočet Fourierových koeficientů, které mají tyto obtíže překonávat.

b) Funkci f aproximujeme funkcí φ , jejíž Fourierovy koeficienty lze snadno spočítat, a jako přibližné hodnoty Fourierových koeficientů funkce f použijeme Fourierovy koeficienty funkce φ . S výhodou se tu používá aproximace trigonometrickými polynomy nebo periodickými kubickými spline-funkcemi. Výpočet Fourierových koeficientů pro větší k působí opět problémy (viz např. [2], [29]).

Je-li funkce f dána tabulkou funkčních hodnot v bodech $x_s = 2\pi s/(m+1)$, $s = 0, 1, \dots, m$, a aproximujeme-li ji interpolačním trigonometrickým polynomem, dají se koeficienty tohoto polynomu (které jsou zároveň jeho Fourierovými koeficienty) vypočítat podle věty 7.4.

7.4 Věta. Fourierova analýza – diskrétní případ.

Nechť jsou dány funkční hodnoty funkce f v bodech x_0, x_1, \dots, x_m , kde $x_s = 2\pi s/(m+1)$. Pak je možno k těmto datům sestavit interpolační trigonometrický polynom, který má tvar

$$(7.4.1) \quad \sum_{k=0}^m C_k e^{ikx}$$

nebo

$$(7.4.2) \quad \sum_{k=0}^L (A_k \cos kx + B_k \sin kx) + \frac{1}{2} \Theta A_{L+1} \cos(L+1)x;$$

je-li m sudé, je $\Theta = 0$, $L = m/2$ je-li m liché, zde $\Theta = 1$, $L = (m-1)/2$.

Pro koeficienty A_k, B_k, C_k přitom platí

$$(7.4.3) \quad \begin{aligned} A_0 &= (m+1)^{-1} \sum_{s=0}^m f(x_s), \\ A_k &= (m+1)^{-1} \sum_{s=0}^m f(x_s) \cos kx, \quad k = 1, 2, \dots, L+1, \\ B_0 &= 1, \\ B_k &= (m+1)^{-1} \sum_{s=0}^m f(x_s) \sin kx, \quad k = 1, 2, \dots, L, \\ C_k &= (m+1)^{-1} \sum_{s=0}^m f(x_s) e^{-ikx}, \quad k = 0, 1, \dots, m, \end{aligned}$$

Nevezmeme-li součet (7.4.1) celý, ale pouze do $k = M$, kde $M < m$, dostaneme trigonometrický polynom, který při daných tabulkových bodech x_s , $s = 0, 1, \dots, m$ nejlepší L_2 -aproximací funkce é ve třídě všech trigonometrických polynomů o stejném počtu koeficientů. Podobné tvrzení platí pro součet (7.4.2).

Důkaz pouze naznačíme. Uvažujme nejprve (7.4.1). Systém funkcí

$$\{e^{ikx}, \quad k = 0, 1, \dots, m\}$$

je při daných tabulkových bodech ortogonální, jak jsme ukázali v příkl. 5.4.5. Podle (6.7.1) a výsledků příkl. 5.4.5 pak máme

$$C_k = \frac{(f, \varphi_k)}{(\varphi_k, \varphi_k)} = (m+1)^{-1} \sum_{s=0}^m f(x_s) e^{ikx_s},$$

což je vzorec pro C_k ze (7.4.3). Pokud se počet základních funkcí rovná počtu tabulkových bodů, dostáváme tak podle věty 6.5 interpolační aproximaci.

V případě (7.4.2) by se postupovalo obdobným způsobem (dokázala by se nejprve ortogonalita užitého systému základních funkcí). To, že (7.4.2) je interpolační aproximace, plyne však ze vzájemného vztahu polynomů (7.4.2) a (7.4.1). Pro jejich koeficienty totiž platí

$$(7.4.4) \quad \begin{aligned} C_0 &= A_0, \\ C_k &= \frac{1}{2}(A_k - iB_k), \quad C_{m+1-k} = \frac{1}{2}(A_k + iB_k), \quad k = 1, 2, \dots, L; \\ A_k &= C_k - C_{m+1-k}, \quad B_k = i(C_k - C_{m+1-k}), \quad k = 1, 2, \dots, L. \end{aligned}$$

Dá se ukázat, že pro m liché je $C_{L+1} = \frac{1}{2}A_{L+1}$. Dosadíme-li odtud za C_k do (7.4.1) pro $x = x_s$, zjistíme, že hodnoty polynomů (7.4.1) a (7.4.2) v tabulkových bodech se sobě rovnají. Mimo tabulkové body se však mohou oba polynomy chovat zcela odlišně. Poznáváme ještě, že pro reálnou funkci f jsou koeficienty A_k, B_k opět reálná čísla.

7.5 Praktické důsledky věty 7.4.

Přístupujeme-li k přibližnému výpočtu Fourierových koeficientů způsobem b) z odst. 7.3, budeme podle věty 7.4 považovat koeficienty A_k, B_k a C_k ze (7.4.3) za přibližné hodnoty Fourierových koeficientů funkce f . po prostudování kap. II čtenář snadno zjistí, že vzorce (7.4.3) jsou ekvivalentní aplikaci lichoběžníkového pravidla pro přibližný výpočet integrálů na integrály (7.2.1) a (7.2.2) [pro periodickou funkci g a periodou 2π máme $\int_{-\pi}^{\pi} g(x)dx = \int_0^{2\pi} g(x)dx$]. Bylo by tudíž v souladu se způsobem a) odst. 7.3 prohlásit A_k, B_k, C_k za přibližné hodnoty koeficientů a_k, b_k, c_k pro všechna k . To ale není možné, takový přístup je pro k velká ve srovnání s m nepoužitelný.

Nasvědčuje tomu již ten důležitý fakt, že ve všech tabulkových bodech $x_s, s = 0, 1, \dots, m$ platí při $k_1 - k_2 = q(m+1)$, q libovolné celé číslo, identita

$$(7.5.1) \quad e^{ik_1x} = e^{ik_2x}$$

(ověřte dosazením za x_s). Platnost (7.5.1) v tabulkových bodech má za následek např. to, že je

$$(7.5.2) \quad C_{k+q(m+1)} = C_k$$

pro všechna celá q , zatímco z teorie Fourierových řad pro spojitou funkci f plyne $c_k \rightarrow 0$ pro $k \rightarrow \infty$. Je tedy třeba postupovat s rozvahou, ignorování popsané skutečnosti zcela mylných závěrů [2].

Platnost (7.5.1) umožňuje přepsat interpolační trigonometrický polynom (7.4.1) na tvar (L a Θ viz větu 7.4)

$$(7.5.3) \quad \sum_{k=-L}^{L+\Theta} C_k e^{ikx};$$

koeficienty C_k se zápornými indexy dodefinujeme podle (7.5.2). Snadno se lze přesvědčit o tom, že v tabulkových bodech nabývají polynomy (7.4.1) a (7.5.3) stejných hodnot. Tvar (7.5.3) je vhodnější pro aproximování funkce f mimo uzly interpolace [2].

Věta 7.4 má pro početní praxi značný význam. Výhodou použití vztahů (7.4.3) je skutečnost, že pro výpočet tohoto typu existuje velmi efektivní algoritmus, který popíšeme v odst. 7.6 až 7.8.

7.6 Příklad.

Na zjednodušeném příkladě ukážeme dva možné způsoby výpočtu součtů tvaru (7.4.3). Položme $N = m + 1$ a předpokládejme, že jsou dány čtyři tabulkové body x_s ; bude tedy $x_0 = 0$, $x_1 = \pi/2$, $x_2 = \pi$, $x_3 = 3\pi/2$ a $N = 4$. Naším úkolem je vypočítat hodnoty koeficientů C_0, C_1, C_2, C_3 , známe-li hodnoty $f(x_s)$, $s = 0, 1, 2, 3$. Pro kratší zápis použijeme označení $f_s = f(x_s)$.

Podle (7.4.3) máme vypočítat

$$(7.6.1) \quad C_k = \frac{1}{4}(f_0 + f_1 e^{-ik\pi/2} + f_2 e^{-ik\pi} + f_3 e^{-ik(3\pi/2)}), \quad k = 0, 1, 2, 3.$$

Označíme-li $w = \exp(-i\pi/2)$ a $F_s = \frac{1}{4}f_s$, přejde (7.6.1) ve

$$(7.6.2) \quad C_k = F_0 + F_1 w^k + F_2 w^{2k} + F_3 w^{3k}, \quad k = 0, 1, 2, 3.$$

Všimneme si toho, že platí $w^N = w^4 = 1$.

Jeden z možných způsobů výpočtu koeficientů C_k podle (7.6.2) je užití postupu podobného Hornerovu algoritmu pro výpočet hodnoty polynomu (viz [21]). Vztah (7.6.2) vhodně uzavorkujeme a dostaneme

$$(7.6.3) \quad C_k = [(F_3 w^{3k} + F_2)w^k + F_1]w^k + F_0, \quad k = 0, 1, 2, 3.$$

K výpočtu jednoho koeficientů C_k podle (7.6.3) potřebujeme provést 3 komplexní sčítání a 3 násobení. Celkově tedy k výpočtu C_k , $k = 0, 1, 2, 3$, potřebujeme $(N - 1)N = 3 \cdot 4 = 12$ sčítání a 12 násobení a hodnoty w^k , $k = 1, 2, 3$.

Skutečnost, že v našem příkladu platí $w^4 = 1$, nám umožňuje vypočítat součty (7.6.2) menším počtem operací. Vztahy (7.6.2) totiž můžeme přepsat na

$$(7.6.4) \quad \begin{aligned} C_0 &= (F_0 + F_2) + (F_1 + F_3), \\ C_1 &= (F_0 + F_2 w^2) + w(F_1 + F_3 w^2), \\ C_2 &= (F_0 + F_2) + w^2(F_1 + F_3), \\ C_3 &= (F_0 + F_2 w^2) + w^3(F_1 + F_3 w^2). \end{aligned}$$

Označíme-li nyní $R_0 = F_0 + F_2$, $R_1 = F_0 + F_2 w^2$, $S_0 = F_1 + F_3$, $S_1 = F_1 + F_3 w^2$, vidíme, že z těchto hodnot vypočteme všechny čtyři koeficienty C_k pomocí 4 sčítání a 3 násobení. Hodnoty R_0, R_1, S_0, S_1 vyžadují provedení 4 sčítání a 2 násobení. Celkově tedy k výpočtu stačí znalost hodnot w^k , $k = 0, 1, 2, 3$ a 5 komplexních násobení a 4 sčítání, což je značná úspora oproti (7.6.3).

Veličiny R_0, R_1, S_0, S_1 se počítají ze vztahu (7.6.2) pro C_k . Tato skutečnost umožňuje při $N = 2^M$, $M > 2$, celý postup rekurentně opakovat. Obecný algoritmus je popsán v odst. 7.7. Všimneme si na závěr ještě toho, že přechod od C_k , $k = 0, 1, 2, 3$, k R_{k_1}, S_{k_1} , $k_1 = 0, 1$, můžeme formálně zapsat následujícím způsobem, který použijeme také při odvození obecného algoritmu.

Přepíšeme (7.6.2) jako

$$C_k = [F_0 + F_2(w^2)^k] + w^k[F_1 + F_3(w^2)^k].$$

Nyní vydělíme k číslem $N/2 = 2$; dostaneme tak $k = q \cdot 2 + k_1$, $k_1 = 0, 1$. Protože platí $(w^4 = 1)$

$$(w^2)^k = (w^2)^{2q}(w^2)^{k_1} = (w^4)^q(w^2)^{k_1} = (w^2)^{k_1},$$

máme

$$C_k = R_{k_1} + w^k S_{k_1}, \quad k = 0, 1, 2, 3, \quad k = q \cdot 2 + k_1,$$

kde

$$R_{k_1} = F_0 + F_2(w^2)^{k_1}, \quad S_{k_1} = F_1 + F_3(w^2)^{k_1}, \quad k_1 = 0, 1.$$

7.7 Rychlá Fourierova analýza.

V tomto odstavci popíšeme rychlý algoritmus pro výpočet koeficientů C_k ze vztahu (7.4.3). Algoritmus je založen na úvahách provedených v příkl. 7.6 (viz též [2], [4]).

Pro snadnou formulaci pozměníme stejně jako v odst. 7.6 značení užívané v odst. 7.4. Předpokládáme, že jsou dány hodnoty funkce f v N bodech $x_s = 2\pi s/N$, $s = 0, 1, \dots, N-1$, a že naším úkolem je spočítat koeficienty C_k , $k = 0, 1, 2, \dots, N-1$. Podle vztahu (7.4.3) máme

$$(7.7.1) \quad C_k = \frac{1}{N} \sum_{s=0}^{N-1} f(x_s) \exp\left(i k \frac{2\pi s}{N}\right), \quad k = 0, 1, \dots, N-1.$$

Provedeme jednoduchou úpravu tohoto vzorce; položíme $F_s = f(x_s)/N$, $w = \exp(-2\pi i/N)$. Naši úlohu pak můžeme formulovat takto: Máme vypočítat

$$(7.7.2) \quad C_k = \sum_{s=0}^{N-1} F_s w^{ks}$$

pro $k = 0, 1, \dots, N-1$. Všimneme si již nyní, že platí $w^N = 1$; to později využijeme.

Algoritmus rychlé Fourierovy analýzy popíšeme pouze pro případ, kdy $N = 2^M$; toho lze většinou snadno docílit. Formulace algoritmu pro obecná N je složitá, ale proveditelná (viz např. [2], [29]), a i v obecném případě dostáváme účinné algoritmy.

Položíme pro s sudé $s_1 = s/2$ (neboli $s = 2s_1$) a pro s liché $s_1 = (s-1)/2$ (neboli $s = 2s_1 + 1$). Je $0 \leq s_1 \leq \frac{1}{2}N - 1$. Rozdělíme-li součet v (7.7.2) na sčítance se sudými a lichými indexy, můžeme psát ($k = 0, 1, \dots, N-1$)

$$(7.7.3) \quad C_k = \sum_{s_1=0}^{N/2-1} F_{2s_1} (w^2)^{ks_1} + w^k \sum_{s_1=0}^{N/2-1} F_{2s_1+1} (w^2)^{ks_1}.$$

Nyní ukážeme, jak výpočet $C - k$, $k = 0, 1, \dots, N-1$, rozložíme na výpočet dvou součtů tvaru (7.7.2) s $N/2 = 2^{M-1}$ členy podobně, jako jsme to pro $M = 2$ provedli v příkl. 7.6.

Vydělíme-li k číslem $N/2$, dostáváme $k = q(N/2) + k_1$ (q je podíl, k_1 zbytek). Protože $w^N = 1$, platí

$$(7.7.4) \quad (w^2)^{ks_1} = (w^2)^{q(N/2)s_1} (w^2)^{k_1 s_1} = (w^N)^{qs_1} (w^2)^{k_1 s_1} = (w^2)^{k_1 s_1}.$$

Označme

$$(7.7.5) \quad R_{k_1} = \sum_{s_1=0}^{N/2-1} F_{2s_1} (w^2)^{k_1 s_1}, \quad S_{k_1} = \sum_{s_1=0}^{N/2-1} F_{2s_1+1} (w^2)^{k_1 s_1},$$

$$k_1 = 0, 1, \dots, (N/2) - 1.$$

Zavedeme-li toto označení do (7.7.3) [bereme přitom v úvahu (7.7.4)], dostáváme

$$(7.7.6) \quad C_k = R_{k_1} + w^k S_{k_1}, \quad k = 0, 1, \dots, N - 1, \quad k = q(N/2) + k_1.$$

Výpočet R_{k_1} a S_{k_1} představuje 2 Fourierovy analýzy z $N/2 = 2^{M-1}$ hodnot funkce f . Rozumíme-li pod slovem "operace" jedno komplexní násobení a jedno komplexní sčítání, je vidět, že k výpočtu C_k z R_{k_1} a S_{k_1} podle (7.7.6) zapotřebí nejvýše $2 \cdot 2^M$ takových operací (počítaje v to i výpočet w^k , $k = 0, 1, \dots, N - 1$).

Vtip celého algoritmu spočívá v tom, že rovněž platí $(w^2)^{N/2} = 1$, takže uvedený postup můžeme aplikovat opakovaně dál. Dostaneme tak 4 Fourierovy analýzy z 2^{M-2} hodnot, ty dále rozdělíme a dostaneme 8 Fourierových analýz z 2^{M-3} hodnot atd. Označíme-li r_M celkový počet operací pro výpočet koeficientů při $N = 2^M$, platí pro počet operací při uvedeném postupu vztah

$$r_M \leq 2r_{M-1} + 2 \cdot 2^M, \quad M = 1, 2, 3, \dots$$

Ale $r_0 = 0$, a proto (indukcí)

$$r_M \leq 2M \cdot 2^M = 2N \log_2 N.$$

Je-li tedy N mocnina dvou, je k provedení algoritmu rychlé Fourierovy analýzy zapotřebí nejvýše $2N \log_2 N$ operací. Výpočet koeficientů ze vzorce (7.7.2) Hornerovým algoritmem je sice jednodušší, potřebuje ale řádově N^2 operací. Rozdíl je patrný již při poměrně malých N ; tak např. pro $N = 2^7 = 128$ potřebuje Hornerův algoritmus řádově $2^{14} = 16\,384$ operace, rychlá Fourierova analýza méně než $2 \cdot 2^7 \cdot 7 = 1\,792$ operace, a je tedy asi desetkrát rychlejší.

7.8 Algoritmus rychlé Fourierovy analýzy.

Algoritmus z odst. 7.7 popsali v roce 1965 Cooley a Tukey. Nazývá se také *rychlá Fourierova transformace*. Existují různé varianty tohoto algoritmu [2], [4], [29], které nacházejí použití v mnoha oblastech numerické matematiky, zejména při řešení parciálních diferenciálních rovnic.

Uvádíme příklad algoritmu v jazyce FORTRAN.³⁾ Algoritmus je určen pro $N = 2^M$. Vstupní data jsou hodnoty funkce f dělené číslem N v poli A a číslo M . Předpokládá se, že hodnota $f(x_s)/N$ je uložena na místě $A(s+1)$; připomínáme, že $x_s = 2\pi s/N$. Výstupní data jsou hodnoty C_k , $k = 0, 1, 2, \dots, N - 1$, opět uložené v poli A .

Algoritmus se skládá ze dvou částí. V první části, která končí příkazem s návěštím 7, se přerovnávají hodnoty $f(x_s)$ do pořadí, v němž budou do rychlé Fourierovy analýzy vstupovat (srov. příkl. 7.6). Druhá část je vlastní algoritmus popsany v odst. 7.7.

```

SUBROUTINE FFT(A,M)
  COMPLEX A(1), U, W, T
  N=2**M
  NV2=N/2
  NM1=N-1
  J=1

```

³⁾ Algoritmus je převzat z článku Cooley, Lewis, Welch, IEEE Transactions E-12, No. 1 (March 1965).

```

DO 7 I=1, MN1
IF(I.GE.J) GOTO 5
T=A(J)
A(J)=A(I)
A(I)=T
5 K=NV2
6 IF(K.GE.J) GOTO 7
J=J-K
K=K/2
GOTO 6
7 J=J+K
DO 20 L=1, M
LE=2**L
LE1=LE/2
U=CMPLX(1.,0.)
ANG=3,14159265358979/LE1
W=CMPLX(COS(ANG), -SIN(ANG))
DO 20 J=1, LE1
DO 10 I=J, N, LE
IP=I+LE1
T=A(IP)*U
A(IP)=A(I)-T
10 A(I)=A(I)+T
20 U=U*W
RETURN
END

```

7.9 Fourierova syntéza.

Všimneme si také toho, že výpočet koeficientů C_k podle (7.4.3) (Fourierova analýza) je ve své podstatě operací stejného typu, jako jsou výpočty prováděné při tabelování hodnot trigonometrického polynomu (7.4.1) pro $x_s = 2\pi s/(m+1)$, $s = 0, 1, \dots, m$ (*Fourierova syntéza*). Algoritmus popsáný v odst. 7.6 až 7.8 je tedy použitelný i pro Fourierovu syntézu.

7.10 Úlohy.

7.10.1

Funkce f je dána vztahy $f(x) = -1$ pro $x \in (-\pi, 0)$, $f(0) = 0$, $f(x) = 1$ pro $x \in (0, \pi)$. Vně intervalu $x \in (-\pi, \pi)$ je funkce definována periodickým prodloužením. Sestrojte Fourierův rozvoj funkce f (spojitý případ). Ověřte tvrzení věty 7.2 v bodech $x = n\pi$.

[$a_k = 0$ pro všechna k , $b_k = 0$ pro k sudá, $b_k = 4/(k\pi)$ pro k lichá.]

7.10.2

Proveďte Fourierovu analýzu (spojitý případ) pro

a) $f(x) = |x|$ na $\langle -\pi, \pi \rangle$.

[$a_0 = \pi/2$, $a_k = 0$ pro $k > 0$, k sudá; $a_k = -4/(k^2\pi)$ pro $k > 0$, k lichá; $b_k = 0$ pro všechna k .]

b) $f(x) = x^2$ na $\langle -\pi, \pi \rangle$,

[$a_0 = \pi^2/3$, $a_k = (-1)^k 4/k^2$, $b_k = 0$ pro všechna k .]

7.10.3

Označme $S_n(x)$ částečný součet řady (7.1.2) do $k = n$ včetně. Stanovte nejlepší L_2 -aproximace (spojitý případ) funkcí z úlohy 7.10.2 trigonometrickými polynomy $S_2(x)$, $S_4(x)$, $S_6(x)$ a vypočítejte hodnoty těchto aproximací v bodech $x = 0$ a $x = \pi$.

[a) $S_2(0) \doteq 0,297\,56$; $S_4(0) \doteq 0,156\,09$; $S_6(0) \doteq 0,105\,16$; $S_2(\pi) \doteq 2,844\,0$; $S_4(\pi) \doteq 0,985\,5$; $S_6(\pi) \doteq 3,036\,4$. b) $S_6(0) \doteq 0,046\,535$; $S_6(\pi) \doteq 9,255\,4$.]

7.10.4

Funkce f je dána vztahy $f(x) = x$ pro $x \in (0, 2\pi)$, $f(0) = \pi$ a je periodicky rozšířena pro ostatní reálná x . Sestrojte pro ni interpolační trigonometrický polynom tvaru (7.4.2) s uzly $x_0 = 0$, $x_1 = \pi/2$, $x_2 = \pi$, $x_3 = 3\pi/2$. Nakreslete graf funkce f i interpolačního polynomu s posuďte jejich shodu.

[Hledaný polynom je $\pi - \frac{1}{2}\pi \sin x$. Funkce f není v bodech $x = 2n\pi$ spojitá.]

7.10.5

Funkce f je dána tabulkou naměřených hodnot (tab. 17). Najděte metodou nejmenších čtverců nejlepší aproximace těchto hodnot trigonometrickými polynomy (7.4.2).

[Na 4D: $A_0 = 0,000\,2$, $A_1 = 0,000\,2$, $A_2 = 0,000\,0$, $A_3 = 1,000\,2$, $B_1 = 1,000\,2$, $B_2 = -0,000\,2$, $B_3 = 0,000\,1$.]

Tabulka 17

x	0	$\frac{2\pi}{7}$	$\frac{4\pi}{7}$	$\frac{6\pi}{7}$	$\frac{8\pi}{7}$	$\frac{10\pi}{7}$	$\frac{12\pi}{7}$
$f(x)$	1,000 4	-0,119 0	1,598 7	0,211 5	-0,656 7	-0,351 4	-1,682 4

7.10.6

Sestrojte tabulku hodnot trigonometrického polynomu $1 + e^{ix} + 2e^{-ix}$ v bodech $x_s = 2\pi s/4$, $s = 0, 1, 2, 3$. Sestrojte k těmto hodnotám podle věty 7.4 interpolační trigonometrický polynom

(7.4.1). Porovnejte s původním polynomem a vysvětlete rozdíly. Z polynomu (7.4.1) pak sestrojte interpolační polynom tvaru (7.5.3).

[Polynom (7.4.1) je $1 + e^{ix} + 2e^{3ix}$; v tabulkových bodech platí $e^{3ix} = e^{-ix}$. Vztah (7.5.3) dává původní polynom.

7.10.7

Ověřte funkci programu z odst. 7.8 na příkl. 7.6 (písemně).

[Návod: V příkl. $w^2 = -1$.]

7.10.8

Je známo, že data v tab. 18 jsou porušené funkční hodnoty jistého trigonometrického polynomu. Proveďte rychlou Fourierovu analýzu těchto dat a sestrojte tento polynom.

[$4 + \cos x + 2 \sin 2x$.]

Tabulka 18

x	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π	$\frac{5\pi}{4}$	$\frac{3\pi}{2}$	$\frac{7\pi}{4}$
$f(x)$	5,000 3	6,706 7	3,999 6	1,292 7	3,000 1	5,293 3	4,000 5	2,707 5

8 Čebyševova aproximace

8.1 Nejlepší stejnoměrná aproximace.

V tomto článku shrneme základní fakta o aproximacích, které minimalizují veličinu (1.5.1), tj. jejichž maximální chyba na intervalu $\langle a, b \rangle$

$$(8.1.1) \quad \varrho_{\max}(f, \varphi) = \max_{x \in \langle a, b \rangle} |f(x) - \varphi(x)|$$

je minimální. Jde tu o spojitý případ nejlepší aproximace. Jak jsme již uvedli v odst. 1.5, nazývají se takové aproximace *Čebyševovy* nebo také *nejlepší stejnoměrné aproximace*. Tyto aproximace hrají hlavní roli při konstrukci podprogramů pro výpočet standardních funkcí na počítači.

Pro jisté třídy aproximací je úloha o nalezení Čebyševovy aproximace k dané funkci f po teoretické stránce vyřešena. Uvedeme větu, která charakterizuje nejlepší stejnoměrnou aproximaci polynomem či racionální funkcí.

8.2 Věta.

Nechť je dána funkce f , která je spojitá na intervalu $\langle a, b \rangle$. Nechť \mathcal{F}_n znamená buď třídu \mathcal{P}_n polynomů nejvýše n -tého stupně, nebo třídu \mathcal{R}_{pq} racionálních funkcí s $n+1$ nezávislými volitelnými parametry.

Pak nejlepší stejnoměrná aproximace φ^* funkce f je ta funkce $\varphi^* \in \mathcal{F}_n$, pro kterou platí:

a) Existuje $n+2$ bodů $x_i \in \langle a, b \rangle$, $i = 0, 1, \dots, n+1$, takových, že v nich absolutní hodnota chyby aproximace nabývá maxima, tj.

$$(8.2.1) \quad |f(x_i) - \varphi^*(x_i)| = \max_{x \in \langle a, b \rangle} |f(x) - \varphi^*(x)| = \varrho_{\max}(f, \varphi^*);$$

b) Extrémy chyby aproximace $f - \varphi^*$ v bodech x_i mají střídavá znaménka a stejnou velikost, tj. pro $a \leq x_0 < x_1 < \dots < x_n < x_{n+1} \leq b$ platí

$$(8.2.2) \quad f(x_i) - \varphi^*(x_i) = -[f(x_{i+1}) - \varphi^*(x_{i+1})], \quad i = 0, 1, 2, \dots, n.$$

Důkaz z věty lze nalézt např. v [3], [26].

8.3 Příklad.

V příkladu 1.6 jsme jako Čebyševovu aproximaci funkce \sqrt{x} na intervalu $\langle 0, 1 \rangle$ polynomem prvního stupně uvedli polynom $x + \frac{1}{8}$. Ověřme podle věty 8.2, že je to skutečně nejlepší aproximace.

V tomto případě je $n = 1$ (polynom prvního stupně má dva koeficienty)ů hledejme extrémy funkce

$$e(x) \equiv f(x) - \varphi^*(x) = \sqrt{x} - x - \frac{1}{8}.$$

Je

$$e'(x) = (2\sqrt{x})^{-1} - 1$$

a z podmínky $e'(x) = 0$ dostáváme jako jediný extrém uvnitř intervalu $\langle 0, 1 \rangle$ bod $x_1 = \frac{1}{4}$ s hodnotou $e(x_1) = \frac{1}{8}$. Další extrémy jsou v krajních bodech $x_0 = 0$ a $x_2 = 1$. V těchto bodech je $e(x_0) = e(x_1) = -\frac{1}{8}$ (načrtněte si obrázek).

Extrémy jsou celkem tři, mají střídavá znaménka a stejnou absolutní hodnotu. V ostatních bodech intervalu $\langle 0, 1 \rangle$ je $|f(x) - \varphi^*(x)| < \frac{1}{8}$. Polynom $x + \frac{1}{8}$ tedy v \mathcal{P}_1 , tj. mezi polynomy tvaru $ax + b$, minimalizuje veličinu

$$\varrho_{\max}(\sqrt{x}, \varphi) = \max_{x \in \langle 0, 1 \rangle} |\sqrt{x} - (ax + b)|.$$

8.4 Algoritmus metody minimalizace maximální chyby.

Sestrojení Čebyševovy aproximace bývá v praktických případech obtížné. Postupuje se proto zpravidla tak, že se místo ní hledají aproximace splňující podmínky (8.2.1) a (8.2.2) co nejlépe. To znamená že v obou těchto vztazích se připouští pouze přibližná rovnost, počet extrému se střídavými znaménky však musí být v přesném souladu s počtem parametrů $n+1$ funkcí z \mathcal{F}_n .

Nesestrojíme tak ovšem nejlepší aproximaci φ^* , ale při vhodné volbě extrémálních bodů x_i , $i = 0, 1, \dots, n+1$, dostaneme aproximaci $\tilde{\varphi}$, která se kvalitativně od φ^* liší zpravidla bezvýznamně [$\varrho_{\max}(s, \tilde{\varphi}) \approx \varrho_{\max}(s, \varphi^*)$].

Při předem daných x_i se přitom vhodná aproximace $\tilde{\varphi}$ dá sestavit celkem snadno: Hodnoty parametrů se vypočítají ze soustavy rovnic (8.2.2). Hlavním problémem tedy je stanovení bodů x_i , $i = 0, 1, \dots, n+1$. Z důvodů, které zde pro nedostatek místa nemůžeme rozebírat⁴⁾, je při aproximaci polynomem nejvhodnější volit body x_i tak, aby byly rozloženy stejně jako extrémy Čebyševova polynomu T_{n+1} . Pracujeme-li na intervalu $\langle -1, 1 \rangle$, volíme tedy (srov. odst. 5.6.1)

$$x_{n+1-k} = \cos \frac{k\pi}{n+1}, \quad k = 0, 1, \dots, n+1,$$

neboli

$$(8.4.1) \quad x_i = \cos \left(\frac{n+1-i}{n+1} \pi \right), \quad i = 0, 1, \dots, n+1.$$

Pro obecný interval $\langle a, b \rangle$ přetransformujeme (8.4.1) lineární transformací

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

Poznamenejme ještě, že skutečně extrémy takto vzniklé funkce $f - \tilde{\varphi}$ nemusí padnout právě do bodů x_i a nemusí být všechny stejně velké.

Uvedený postup můžeme vzít za základ iteračního algoritmu pro minimalizaci maximální chyby. K $n+2$ bodům $x_i^{(0)} \in \langle a, b \rangle$ sestrojeným z bodů (8.4.1) vypočteme ze soustavy $n+1$ rovnic (8.2.2) hodnoty $n+1$ parametrů, které určují funkci $\varphi^{(0)} \in \mathcal{F}_n$. Najdeme extrémy funkce $f - \varphi^{(0)}$ a body, v nichž tato extrémy nastávají, vezmeme za nové extrémální body $x_i^{(1)}$, z nichž opět pomocí (8.2.2) vypočteme hodnoty $n+1$ parametrů. Tyto parametry určují novou aproximaci $\varphi^{(1)} \in \mathcal{F}_n$. Dále postupujeme podobně jako s aproximací $\varphi^{(0)}$ a konstruujeme posloupnost aproximací $\varphi^{(k)} \in \mathcal{F}_n$. Za určitých mírně omezujících předpokladů se dá dokázat [26], že $\varphi^{(k)}$ konvergují stejnoměrně na intervalu $\langle a, b \rangle$ k Čebyševově aproximaci $\varphi^* \in \mathcal{F}_n$ funkce f .

8.5 Příklad.

Použijeme algoritmus z odst. 8.4 k aproximování funkce \sqrt{x} na intervalu $\langle 0, 1 \rangle$ polynomem prvního stupně. Je $n = 1$ a podle (8.4.1) po transformaci na $\langle 0, 1 \rangle$ dostáváme $x_0^{(0)} = 0$, $x_1^{(0)} = \frac{1}{2}$, $x_2^{(0)} = 1$. Soustava (8.2.2) pro $f(x) = \sqrt{x}$, $\varphi(x) = ax + b$ má po úpravě tvar

$$(8.5.1) \quad a(x_{i+1}^{(k)} + x_i^{(k)}) + 2b = \sqrt{x_{i+1}^{(k)}} + \sqrt{x_i^{(k)}}, \quad i = 0, 1.$$

Pro $k = 0$ je tudíž

$$\begin{aligned} \frac{1}{2}a + 2b &= \frac{1}{2}\sqrt{2}, \\ \frac{3}{2}a + 2b &= 1 + \frac{1}{2}\sqrt{2} \end{aligned}$$

⁴⁾ Tato volba souvisí s minimalizační vlastností Čebyševových polynomů popsanou v odst. 5.6.

a jako řešení dostáváme $a = 1$, $b = (\sqrt{2} - 1)/4$. Máme

$$\varphi^{(0)}(x) = x + \frac{\sqrt{2} - 1}{4}$$

a

$$\varrho_{\max}(\sqrt{x}, \varphi^{(0)}) = \frac{2 - \sqrt{2}}{4} \doteq 0,146.$$

Této chyby se nabývá v bodě $x = \frac{1}{4}$, další extrémů funkce $f - \varphi^{(0)}$ jsou v bodech $x = 0$ a $x = 1$.

Položíme proto $x_0^{(1)} = 0$, $x_1^{(1)} = \frac{1}{4}$, $x_2^{(1)} = 1$. Ze soustavy (8.5.1) tentokrát obdržíme $a = 1$, $b = \frac{1}{8}$ a $\varphi^{(1)}$ je tudíž již nejlepší stejnoměrnou aproximací funkce \sqrt{x} na intervalu $\langle 0, 1 \rangle$ (viz např. 8.3). Pro srovnání připomínáme, že

$$\varrho_{\max}(\sqrt{x}, \varphi^{(1)}) = \frac{1}{8} = 0,125;$$

byla tedy již aproximace $\varphi^{(0)}$ relativně dobrou aproximací funkce \sqrt{x} .

8.6 Úlohy

8.6.1

Sestrojte Čebyševovu aproximaci funkce e^x na intervalu $\langle 0, 1 \rangle$ polynomem prvního stupně.

[$\varphi^{(0)} \doteq 1,71828x + 0,89479$; $\varrho_{\max}(e^x, \varphi^{(0)}) \doteq 0,1067$; $\varphi^{(1)}(x) \doteq \varphi^*(x) \doteq 1,71828x + 0,89407$; $\varrho_{\max}(e^x, \varphi^{(1)}) \doteq 0,1059$.]

8.6.2

Aproximujte funkci e^x na intervalu $\langle 0, 1 \rangle$ racionální funkcí tvaru $a/(1 + bx)$. Užijte postup z odst. 8.4 s body $x_0 = 0$, $x_1 = 0,7$, $x_2 = 1$ a sestrojte $\varphi^{(0)}$.

[$\varphi^{(0)}(x) \doteq 1,08466/(1 - 0,61303x)$; $\varrho_{\max} \doteq 0,0978$.]

9 Zvyšování přesnosti výpočtů extrapolací

9.1 Zvyšování přesnosti přibližných výpočtů.

V tomto článku popíšeme významnou metodu zvyšování přesnosti přibližných výpočtů, která najde uplatnění v následujících kapitolách při numerickém výpočtu integrálů a řešení diferenciálních rovnic.

Předpokládejme, že počítáme hodnotu veličiny F pomocí aproximace $T(h)$, která závisí na jistém parametru h , a že platí

$$(9.1.1) \quad F = \lim_{h \rightarrow 0} T(h).$$

V takové situaci jsme při použití celé řady numerických metod. Tak např. při studiu konvergence interpolačního procesu (odst. 3.32) hraje roli parametru h veličina $\max_i |x_{i+1}^{(s)} - x_i^{(s)}|$, u interpolačních spline-funkcí je to podobně $h = \max h_i$ (viz odst. 4.8). Jak ještě později uvidíme, je pro mnohé numerické metody typické, že pracnost výpočtu aproximace $T(h)$ při $h \rightarrow 0$ rychle roste a že navíc mohou zaokrouhlovací chyby znemožňovat užití dostatečně malých hodnot h , potřebných pro dosažení požadované přesnosti.

V numerické matematice však o chování chyby aproximace $F - T(h)$ při $h \rightarrow 0$ máme často informaci podrobnějšího charakteru než pouze (9.1.1). Ukážeme nyní, jak nám taková informace může pomoci k tomu, abychom při dané hodnotě parametru h dosáhli přesnosti vyšší, než je pro danou metodu charakteristické. Takovým postupem pak můžeme obejít nutnost pracovat s příliš malými hodnotami h .

Předpokládejme, že pro danou numerickou metodu platí při $h \rightarrow 0$ vztah

$$(9.1.2) \quad T(h) = F - ch^p + O(h^r),$$

kde c, p, r jsou konstanty a $x < p < r$. je tedy $F = T(0)$ a chyba aproximace je řádu h^p . Informace o chování chyby aproximace daná vztahem (9.1.2) nám umožňuje odhadnout F s chybou řádu h^r , $r > p$, na základě dvou hodnot $T(h)$, řekněme pro $h = H$ a $h = qH$ ($q \neq 1$). Pro celý postup přitom stačí znát hodnotu p ; koeficient c znát nemusíme. Podle (9.1.2) máme

$$(9.1.3) \quad \begin{aligned} T(H) &= F + cH^p + O(H^r), \\ T(qH) &= F + c(qH)^p + O(H^r), \end{aligned}$$

neboť při $H \rightarrow 0$ je $O((qH)^r) = O(H^r)$. Vyloučíme-li ze vztahů (9.1.3) výraz cH^p dostáváme

$$(9.1.4) \quad F = T(H) + \frac{T(H) - T(qH)}{q^p - 1} + O(H^r).$$

Ze vztahu (9.1.4) vidíme, že veličina

$$(9.1.5) \quad T_1(H) = T(H) + \frac{T(H) - T(qH)}{q^p - 1}$$

je obecně lepší aproximací hodnoty F než byly $T(H)$ a $T(qH)$, neboť řád její chyby je h^r a $r > p$. Vzhledem k tomu, že výchozí rovnost (9.1.2) má platnost pouze asymptotickou (pro $h \rightarrow 0$), je ovšem třeba mít na paměti, že použití popsaného postupu nemusí přinést zmenšení chyby při každém H .

9.2 Příklad.

Budiž f funkce spojitá na intervalu $\langle a, b \rangle$, která má na $\langle a, b \rangle$ spojitě první derivace a taková, že pro všechna $x \in \langle a, b \rangle$ existuje derivace $f^{(5)}(x)$. Podle věty 2.4 o aproximaci Taylorovým polynomem pak pro dostatečně malá h a $x \in \langle a, b \rangle$ platí

$$(9.2.1) \quad f(x \pm h) = f(x) \pm hf'(x) + \frac{1}{2}h^2 f''(x) \pm \frac{1}{6}h^3 f'''(x) + \frac{1}{24}h^4 f^{(4)}(x) + O(h^5).$$

Odtud ihned plyne vztah

$$(9.2.2) \quad \frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{1}{6}h^2 f'''(x) + O(h^4)$$

[neboť $O(h^5)/(2h) = O(h^4)$]. Vztah (9.2.2) dává numerickou metodu pro přibližný výpočet $f'(x)$ z hodnot funkce f . Klademe $f'(x) \approx [f(x+h) - f(x-h)]/(2h)$ s chybou aproximace řádu h^2 . Jak uvidíme v čl. 15, je tento postup pro velmi malá h nestabilní.

Položíme-li $T(h) = [f(x+h) - f(x-h)]/(2h)$, můžeme podle (9.1.4) s $q = 2$, $p = 2$, $r = 4$ ze dvou hodnot $T(h)$ pro $h = H$ a $h = 2H$ spočítat

$$(9.2.3) \quad f'(x) = T_1(H) + O(H^4), \quad T_1(H) = T(H) + \frac{1}{3}[T(H) - T(2H)]$$

Veličina $T_1(H)$ dává tedy přiblížení hodnoty $f'(x)$ s chybou $O(H^4)$ (při $H \rightarrow 0$).

9.3 Richardsonova extrapolace.

Postupu pro zvyšování přesnosti výsledků získaných numerickou metodou, který jsme popsali v odst. 9.1 a použili v příkl. 9.2, se říká *Richardsonova extrapolace* nebo *extrapolace na $h = 0$* .

Celý postup můžeme totiž chápat jako interpolační úlohu. První dva členy rozvoje (9.1.2) tvoří polynom $F + ch^p$ prvního stupně v h^p . Richardsonovou extrapolaci můžeme vykládat tak, že hodnotami $T(H)$, $T(qH)$ prokládáme polynom $L_1(h)$ prvního stupně v proměnné h^p , $L_1(h) = a + bh^p$, a hodnotu F aproximujeme hodnotou $a = L_1(0)$, [Protože bod $h = 0$ leží mimo interval (H, qH) , jde o extrapolaci.] Skutečně, použijeme-li vzorec (3.5.1), kde nyní bude $f(x_0) = T(H)$, $f(x_1) = T(qH)$, $x_0 = H^p$, $x_1 = (qH)^p$, $x = 0$, dostáváme po úpravě

$$L_1(0) = \frac{q^p T(H) - T(qH)}{q^p - 1},$$

což je totéž jako veličina $T_1(H)$ z (9.1.5).

9.4 Algoritmus opakované Richardsonovy extrapolace.

Pokud umíme vyjádřit $T(h)$ v mocninách h podrobněji [tj. rozepsat dále člen $O(h^r)$ v (9.1.2)], můžeme Richardsonovou extrapolaci opakovat způsobem, který nyní popíšeme, a prokládat hodnotami $T(h_i)$, $i = 1, 2, \dots$, polynomy vyšších stupňů. Algoritmus *opakované Richardsonovy extrapolace* je součástí některých velmi efektivních programů pro řešení úloh, kterými se budeme zabývat v následujících kapitolách.

Pro jednoduchost se v dalším výkladu omezíme na speciální případ, který je pro použití Richardsonovy extrapolace typický, a budeme předpokládat, že platí

$$(9.4.1) \quad T(h) = F + a_0 h^p + a_1 h^{2p} + a_2 h^{3p} + \dots,$$

kde a_0, a_1, a_2, \dots jsou konstanty nezávislé na h . Částečné součty tohoto rozvoje jsou tedy polynomy v h^p . Navíc budeme předpokládat, že parametry h_i , pro které počítáme $T(h_i)$, tvoří geometrickou posloupnost

$$(9.4.2) \quad H, q^{-1}H, q^{-2}H, \dots;$$

obvykle se tu klade $q = 2$. Algoritmus opakované Richardsonovy extrapolace bude nyní spočítat v tom, že hodnotami $T(H), T(q^{-1}H), \dots, T(q^{-s}H)$ proložíme interpolační polynom S -tého stupně v h^p a F aproximujeme hodnotou tohoto polynomu v bodě $h = 0$; k výpočtu této hodnoty použijeme Nevillův algoritmus (3.16.2) s $x_i = (q^{-i}H)^p$.

Výsledný algoritmus můžeme nyní zapsat takto:

$$(9.4.3) \quad \begin{aligned} &\text{Pro } s = 0, 1, \dots, S : \\ &\quad T_{s0} = T(q^{-s}H). \\ &\quad \text{Pro } k = 1, 2, \dots, s : \\ &\quad\quad T_{sk} = T_{s,k-1} + \frac{T_{s,k-1} - T_{s-1,k-1}}{q^{kp} - 1}. \end{aligned}$$

Dá se ukázat [29], že pro dostatečně malé H jsou hodnoty $T_{sk}, k = 1, 2, \dots$, stále lepšími přiblíženími veličiny F . Za našich předpokladů pro tato hodnoty totiž platí

$$(9.4.4) \quad T_{sk}(H) = F + a_k^{(s)} H^{(k+1)p} + a_{k+1}^{(s)} H^{(k+2)p} + \dots$$

Rozvoje typu (9.4.1) mají v praxi zpravidla $p = 1$ nebo $p = 2$; vidíme, že pro větší p bude účinek Richardsonovy extrapolace zřetelnější (chyba bude pro dostatečně malá H klesat rychleji).

Ruční výpočet podle algoritmu (9.4.3) se stejně jako u Nevillova algoritmu uspořádává do trojúhelníkového schématu hodnot $T_{sk}, k \leq s$. Výpočet je zde jednodušší než v obecném algoritmu (3.16.2). Hodnoty T_{sk} ve sloupci k se vypočtou tak, že se k hodnotě $T_{s,k-1}$ v sousedním levém sloupci schématu, přičte veličina $\Delta/(q^{kp} - 1)$, kde Δ je rozdíl dvou hodnot v sousedním levém sloupci, totiž $T_{s,k-1}$ a $T_{s-1,k-1}$. Pro typickou volbu $q = 2$ a $p = 1$ tedy přičítáme veličiny (postupně po sloupcích) $\Delta/1, \Delta/3, \Delta/7, \Delta/15, \dots$, pro $p = 2$ přičítáme $\Delta/3, \Delta/15, \Delta/63, \Delta/225, \dots$.

V praxi neznáme předem hodnotu S potřebnou k dosažení požadované přesnosti. Algoritmus (9.4.3) proto používáme pro $s = 0, 1, 2, \dots$ tak dlouho, až je $\varepsilon = |T_{sk} - T_{s-1,k}|$ menší než požadovaná přesnost, a jako přibližnou hodnotu F užijeme $T_{s,k+1}$. Absolutní hodnota chyby aproximace se pak dá odhadnout jako ε .

9.5 Příklad.

Pro dostatečně hladkou funkci f platí nejen (9.2.2), ale dokonce

$$(9.5.1) \quad \frac{f(x+h) - f(x-h)}{2h} = f'(x) + a_1 h^2 + a_2 h^4 + a_3 h^6 + \dots$$

Příkladem takové funkce je $\ln x$. Použijeme nyní algoritmus (9.4.3) v výpočtu $f'(3)$ pro $f(x) = \ln x$. Hodnoty funkce $\ln x$ vezmeme z šestimístných tabulek a položíme $q = 2, H = 0,8$. Podle (9.5.1) je $p = 2$. Bude tedy

$$h_s = (q^{-s})H = 2^{-s} \cdot 0,8$$

a

$$T_{s0} = \frac{\ln(3+h_s) - \ln(3-h_s)}{2h_s}.$$

Algoritmus opakované Richardsonovy extrapolace dá v tomto případě výsledky uspořádané do tab. 19. U hodnot $\Delta/3$ a $\Delta/15$ uvádíme - podobně jako v tabulkách diferencí - pouze významné

číslíce. Protože $T_{21} = T_{31}$ na 6D, předpokládáme, že $f'(3) \doteq T_{32} = 0,333\,330$ (na 6D). Přesná hodnota je $0,333\,333$ (na 6D) a skutečná chyba je $3 \cdot 10^{-6}$ (vliv zaokrouhlování).

V případě $p = 2, q = 2$ se dá obecně ukázat, že jsou-li hodnoty $T_{s0}, s = 0, 1, 2, \dots$, zatíženy chybami, jejichž absolutní hodnota nepřevyšuje δ , projeví se tyto chyby v dalších extrapoláčnících hodnotách T_{sk} chybami velikosti nejvýše 2δ . V našem příkladu je absolutní chyba T_{s0} vzniklá zaokrouhlováním maximálně $10^{-6}/(2h) \leq 5 \cdot 10^{-6}$, což dává pro chyby v T_{sk} jako odhad číslo 10^{-5} .

Tabulka 19

s	h_s	T_{s0}	$\Delta/3$	T_{s1}	$\Delta/15$	T_{s2}
0	0,8	0,341 590				
1	0,4	0,335 330	-2 087	0,333 243		
2	0,2	0,333 830	- 500	0,333 330	6	0,333 336
3	0,1	0,333 455	- 125	0,333 330	0	0,333 330

9.6 Varianty extrapoláčního algoritmu.

Myšlenka extrapolace na $h = 0$ je mnohem obecnější než postup uvedený v odst. 9.1 až 9.5. Není např. nutné, aby používané hodnoty h tvořily geometrickou posloupnost. Často se používá posloupnost

$$H, \frac{H}{2}, \frac{H}{3}, \frac{H}{4}, \frac{H}{6}, \frac{H}{8}, \frac{H}{12}, \dots,$$

která má u mnohých numerických metod za následek menší celkovou pracnost výpočtu.

Stejným způsobem jako polynom je možno hodnotami $T(h)$ prokládat interpolační racionální funkci a k výpočtu $T(0)$ použít algoritmus (4.1.3). Výsledný algoritmus, který se osvědčil při numerickém výpočtu integrálů a řešení diferenciálních rovnic, je ovšem složitější (viz např. [29]).

9.7 Úlohy

9.7.1

Předpokládejme, že hodnoty $T_{s0}, s = 0, 1, 2, \dots$, a algoritmu (9.4.3), kde $p = 2, q = 2$ jsou zatíženy chybami, jejichž absolutní hodnota je nejvýše δ . Odhadněte maximální možné nepřesnosti, způsobené těmito chybami v hodnotách $T_{sk}, k = 1, 2, 3$.

$$[k = 1 : 5\delta/3, k = 2 : 16\delta/9, k = 3 : 1\,040\delta/567.]$$

9.7.2

Je dána tabulka hodnot funkce $f(x) = (1 + \sin x)^{-1}$ (tab. 20). Použijte postup z příkl. 9.2 a vypočítejte přibližnou hodnotu $f'(0, 2)$.

$$[T_1(0, 1) \doteq -0,681\,8; \text{ přesná hodnota (na 4D) je } -0,682\,1.]$$

Tabulka 20

x	0,0	0,1	0,2	0,3	0,4
$f(x)$	1,000 0	0,909 2	0,834 3	0,771 9	0,719 7

9.7.3

Podobně jako v příkl. 9.5 vypočítejte s absolutní přesností $\frac{1}{2} \cdot 10^{-2}$ přibližnou hodnotu $f'(1)$, kde $f(x) = e^x$. Volte $H = 0,8$. Pracujte na 4D.

[$T_{22} \doteq 2,7183$, $T_{21} - T_{11} \doteq 2,2 \cdot 10^{-3}$, $f'(1) \doteq 2,7183$.]

9.7.4

[5] Předpokládá se, že jistá fyzikální veličina X závisí na tlaku P podle vztahu $X = c_0 + c_1 P^2 + c_2 P^3 + c_3 P^6$, kde c_0, c_1, c_2, c_3 jsou nenulové konstanty. Stanovte přibližnou hodnotu X ve vakuu ($P = 0$) z naměřených hodnot uvedených v tab. 21.

[Návod: Užijte Richardsonovou extrapolaci; $X(0) \doteq 491$.]

Tabulka 21

P	0,8	0,4	0,2	0,1	0,05
X	740	487	475	485	489

II. Numerický výpočet integrálu a derivace

10 Obecně o metodách pro výpočet integrálu a derivace

10.1 Úvod.

Metody aproximace funkcí popsané v předcházející kapitole jsou základem mnoha postupů a algoritmů numerické matematiky. V této kapitole se budeme zabývat převážně pozitivním některých aproximací popsaných v kap. I pro přibližný výpočet integrálu

$$(10.1.1) \quad I(f) = \int_a^b f(x) dx$$

(předpokládáme ovšem, že funkce f je na intervalu $\langle a, b \rangle$ integrovatelná). Poslední článek kapitoly věnujeme přibližnému výpočtu hodnoty derivace.

10.2 Princip metod pro přibližný výpočet integrálu a derivace.

Numerické metody výpočtu integrálu užíváme zejména tehdy, když integrál $I(f)$ není možno vypočítat analytickými metodami (velmi častý případ!). Numerické metody používáme ale i v těch případech, kdy je analytický výpočet příliš pracný. V případě, že je funkce f zadána pouze tabulkou či grafem, je použití některé přibližné metody k výpočtu integrálu či derivace nezbytností.

Přirozený princip numerických metod pro výpočet integrálu a derivace vychází z výsledků kap. I o aproximaci funkcí. Danou funkci f nahradíme její vhodnou aproximací φ . Pak za přibližnou hodnotu derivace $f'(x)$ prohlásíme $\varphi'(x)$; jako aproximaci integrálu $I(f)$ použijeme $I(\varphi)$, tj. integrál z funkce φ , který stanovíme analyticky. Třídou aproximací, z nichž vybíráme φ , volíme tak, aby bylo možné integrál či derivaci aproximující funkce snadno určit. V metodách této kapitoly budeme většinou f aproximovat Lagrangeovými interpolačními polynomy. Metody, které zde budeme studovat, budou pak k přibližnému výpočtu integrálu či derivace používat lineární kombinace hodnot $f(x)$ v některých bodech $x \in \langle a, b \rangle$.

Při numerickém výpočtu integrálu platí, že je-li integrovatelná funkce φ dobrou aproximací funkce f na celém intervalu $\langle a, b \rangle$, je integrál z φ dobrou aproximací integrálu z f , neboť

$$(10.2.1) \quad \left| \int_a^b f(x) dx - \int_a^b \varphi(x) dx \right| \leq \int_a^b |f(x) - \varphi(x)| dx \leq (b-a) \sup_{x \in \langle a, b \rangle} |f(x) - \varphi(x)|.$$

Při numerickém výpočtu derivace však musíme být obezřetní. I když φ je dobrou aproximací funkce f , nemusí být $\varphi'(x)$ stejně dobrou aproximací hodnoty $f'(x)$ ve všech bodech $x \in \langle a, b \rangle$. Maximum absolutní hodnoty funkce $f' - \varphi'$ na intervalu $\langle a, b \rangle$ může být mnohem větší než maximální absolutní hodnota $f - \varphi$. Uvedeme ilustrativní příklad.

10.3 Příklad.

Budiž $f(x) = e^x$, $\varphi(x) = 1 + x$. (Taylorův polynom prvního stupně pro e^x v okolí bodu $x = 0$) a za interval $\langle a, b \rangle$ vezměme interval $\langle 0; 0, 1 \rangle$. Pro chybu aproximace funkce f polynomem φ na intervalu $\langle 0; 0, 1 \rangle$ platí

$$\max |f(x) - \varphi(x)| = e^{0,1} - (1 + 0, 1) \doteq 0, 005 2$$

Všechny derivace funkce f se opět rovnají e^x . První dvě derivace aproximující funkce φ jsou $\varphi' \equiv 1$, $\varphi'' \equiv 0$. Platí tedy

$$\begin{aligned} \max |f'(x) - \varphi'(x)| &= e^{0,1} - 1 \doteq 0, 105 2, \\ \max |f''(x) - \varphi''(x)| &= e^{0,1}. \end{aligned}$$

Vidíme, že chyba v první derivaci je v bodě $x = 0, 1$, asi dvacetkrát větší než chyba aproximace $f''(0, 1)$, hodnota získaná tímto způsobem jako aproximace hodnoty φ je zcela nesmyslná.

Při výpočtu integrálu je situace jiná. Máme

$$\int_0^{0,1} e^x dx = e^{0,1} - 1 \doteq 0, 105 2.$$

Použitím aproximace φ dostáváme jako přibližnou hodnotu tohoto integrálu

$$\int_0^{0,1} (1 + x) dx = 0, 1 + \frac{1}{2} \cdot (0, 1)^2 = 0, 105$$

s chybou 0, 000 2.

10.4 Kvadrurní vzorce.

Pro numerickou aproximaci určitého integrálu se často používá termín *numerická kvadratura*, a to proto aby se tato úlohy odlišila od *numerické integrace*⁵⁾ obyčejných diferenciálních rovnic. Vzorce, které udávají aproximace určitých integrálů, se pak nazývají *kvadrurní vzorce*. Studium takových vzorců se nyní budeme zabývat.

10.5 Příklad.

Odvodíme kvadrurní vzorec pro přibližný výpočet integrálu (10.1.1) tak, že na intervalu $\langle a, b \rangle$ nahradíme funkci f interpolačním polynomem prvního stupně s uzly a, b . Položíme tedy (viz odst. 3.5)

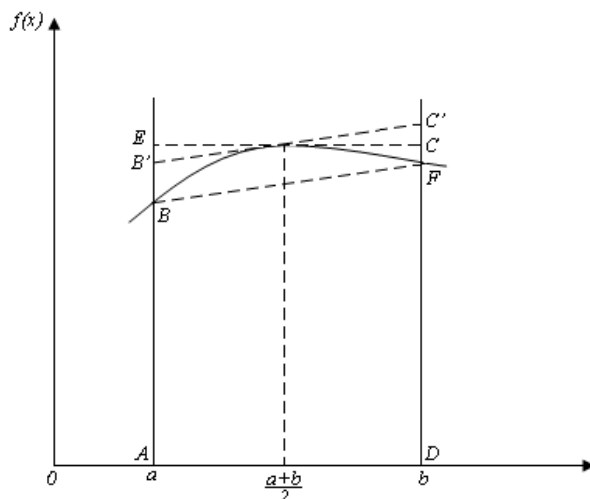
$$\varphi(x) = \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b)$$

⁵⁾ Oba termíny zavádíme pro potřeby tohoto sešitu, nejde tu o ustálenou terminologii.

a integrál $I(f)$ aproximujeme integrálem z funkce φ . Dostáváme tak

$$\begin{aligned}
 (10.5.1) \quad I(f) &= \int_a^b f(x) dx \approx \int_a^b \varphi(x) dx = \\
 &= f(a) \int_a^b \frac{x-b}{a-b} dx + f(b) \int_a^b \frac{x-a}{b-a} dx = \\
 &= \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b) \approx K(f).
 \end{aligned}$$

Vidíme, že získaný kvadraturní vzorec $K(f)$ má tvar lineární kombinace funkčních hodnot s koeficienty $(b-a)/2$, $(b-a)/2$. Celý postup je možno ilustrovat také geometricky. V případě funkce f , jejíž graf je znázorněn na obr. 4, představuje integrál $I(f)$ obsah křivočarého obrazce $ABCD$. Hodnota získaná kvadraturním vzorcem $K(f)$ je rovna obsahu $(b-a)[f(a) + f(b)]/2$ lichoběž-



Obrázek 4:

níka $ABCD$ [vzorec (10.5.1) se proto nazývá *lichoběžníkové pravidlo*]. Integrál $I(f)$ můžeme ale aproximovat také obsahem lichoběžníka $AB'C'D$, kde $B'C'$ je tečna ke křivce $y = f(x)$ v bodě $x = (a+b)/2$. Protože obsah lichoběžníka $AB'C'D$ je roven obsahu obdélníka $AEFD$, má příslušný kvadraturní vzorec tvar (*lichoběžníkové pravidlo*)

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

V tomto případě jsme f vlastně aproximovali konstantou $f((a+b)/2)$; všimněte si opět lineární závislosti hodnoty $K(f)$ na použité funkční hodnotě.

Obr. 4 naznačuje, že patrně můžeme očekávat, že absolutní hodnota *chyby kvadratury*, tj. rozdíl $I(f) - K(f)$, bude menší pro malé intervaly $\langle a, b \rangle$ a pro ty funkce f , které na $\langle a, b \rangle$ nevykazují

prudké změny. Dále můžeme čekat, že obdélníkové pravidlo bude obecně poněkud přesnější než lichoběžníkové. Všimneme si také toho, že bude-li f polynom prvního stupně (grafem takové funkce je přímka), bude jak lichoběžníkové, tak obdélníkové pravidlo dávat hodnotu integrálu přesně [bude $I(f) = K(f)$].

10.6 Interpolační kvadrurní vzorce.

V tomto odstavci popíšeme obecné kvadrurní vzorce pro výpočet $I(f)$ založené na aproximaci integrované funkce f interpolačním polynomem. Postup odvození je podobný jako v příkl. 10.5. Funkci f v integrálu $I(f)$ nahradíme interpolačním polynomem p_n n -tého stupně s uzly x_0, x_1, \dots, x_n navzájem různými a ležícími v intervalu $\langle a, b \rangle$. Polynom p_n zapíšeme v Lagrangeově tvaru (3.3.1)

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

a integrací od a do b dostáváme hledaný kvadrurní vzorec

$$(10.6.1) \quad K(f) = \int_a^b \sum_{i=0}^n f(x_i)l_i(x)dx = \sum_{i=0}^n w_i f(x_i),$$

kde

$$(10.6.2) \quad w_i = \int_a^b l_i(x)dx.$$

Vzorec (10.6.1) má opět tvar lineární kombinace funkčních hodnot v bodech x_i , $i = 0, 1, \dots, n$. Těmto bodům říkáme také *uzly kvadrurního vzorce*. Koefficienty w_i se nazývají *koefficienty kvadrurního vzorce* (10.6.1) (někdy také *váhy*). Všimneme si toho, že koefficienty w_i nezávisí na integrované funkci f a dají se při zadaných uzlech snadno vypočítat. Kvadrurní vzorec tvaru (10.6.1) se nazývá *interpolační kvadrurní vzorec*. Obdélníkové a lichoběžníkové pravidlo z příkl. 10.5 jsou tedy interpolační kvadrurní vzorce.

Interpolační kvadrurní vzorce s $n + 1$ uzly x_i , $i = 0, 1, \dots, n$, mají tu vlastnost, že integrují přesně všechny polynomy n -tého stupně (a stupňů menších než n -tých). Snadno se o tom přesvědčíme, uvážíme-li, že chyba aproximace E polynomu n -tého stupně (a stupňů menších než n -tých) interpolačním polynomem s $n + 1$ uzly je pro všechna $x \in \langle a, b \rangle$ rovna nule (viz odst. 3.12). Pro *chybu* $e(f) = I(f) - K(f)$ *interpolačního kvadrurního vzorce* totiž platí

$$(10.6.3) \quad e(f) = \int_a^b f(x)dx - \sum_{i=0}^n w_i f(x_i) = \int_a^b f(x)dx - \int_a^b p_n(x)dx = \int_a^b E(x)dx,$$

kde $E(x) = f(x) - p_n(x)$. Je-li tedy $E(x) \approx 0$ jako v případě integrace polynomů nejvýše n -tého stupně, je také $e(f) = 0$. Podmínka $E(x) \approx 0$ není ovšem nutnou podmínkou pro to, aby se integrál z funkce E , tj. $e(f)$, rovnal nule. Později uvidíme (viz čl. 13), že vhodnou volbou uzlů x_i ,

$i = 0, 1, \dots, n$, se dá docílit toho, že interpolační kvadrurní vzorec (10.6.1) bude dávat přesnou hodnotu integrálu pro polynomy až do $(2n + 1)$ -ního stupně.

V souvislosti s tím, co jsme právě řekli o integrování polynomů, a s Weierstrassovou větou 1.10 o aproximaci spojitě funkce polynomem se zavádí pojem *algebraického řádu kvadrurního vzorce* (10.6.1). Je to největší takové číslo N , že vzorec (10.6.1) dává hodnotu integrálu přesně pro polynomy x_i , $i = 0, 1, \dots, n$, [slovo "největší" zde znamená, že $e(x^{N+1}) \neq 0$]. Protože jak integrál, tak kvadrurní vzorec (10.6.1) závisejí na integrované funkci lineárně, je vzorec řádu N přesný pro všechny polynomy N -tého stupně (a stupňů menších). Algebraický řád interpolačních kvadrurních vzorců s $n + 1$ uzly x_i , $i = 0, 1, \dots, n$, je vždy nejméně n . Řád lichoběžníkového a obdélníkového pravidla je roven jedné (viz úlohu 10.8).

Pojem algebraického řádu kvadrurního vzorce dává jistou charakteristiku přesnosti aproximace získané tímto vzorcem. Užitím polynomů vyšších stupňů zřejmě můžeme dobře aproximovat širší třídu funkcí než polynomy nižšího stupně. Obecně můžeme proto očekávat, že pro mnohé dostatečně hladké funkce budou vzorce vyššího řádu dávat přibližnou hodnotu integrálu přesněji. Odhady chyb kvadrurních vzorců, které uvádíme v dalších článcích této kapitoly, jsou s touto intuitivní představou zcela v souladu.

Odnik ovšem neplatí, že vzorec vyššího řádu musí dát přesnější aproximaci integrálu $I(f)$ pro každou spojitou funkci f . To by čtenáře nemělo vůbec překvapit, vždyť jsme se již v odst. 3.32 zmínili o tom, že interpolační polynomy nemusí zejména při ekvidistantních uzlech $x_i^{(n)} = a + ih$, $i = 0, 1, \dots, n$, $h = (b - 1)/n$, konvergovat při $n \rightarrow \infty$ k aproximované spojitě funkci f . Podobně se při pevně zadaném systému uzlů $x_i^{(n)}$, $i = 0, 1, \dots, n$, a výpočtu integrálu interpolačními kvadrurními vzorci tvaru (10.6.1) nedá vždy dosáhnout toho, aby pro všechny spojitě funkce f platilo

$$(10.6.4) \quad \lim_{n \rightarrow \infty} \sum_{i=0}^n w_i^{(n)} f(x_i^{(n)}) = \int_a^b f(x) dx$$

(viz [3]). Takové systémy kvadrurních vzorců s koeficienty $w_i^{(n)}$, $i = 0, 1, \dots, n$, a uzly $x_i^{(n)}$, $i = 0, 1, \dots, n$, definovanými pro všechna $n = 0, 1, \dots$, které mají vlastnost (10.6.4) pro každou spojitou funkci f , budeme nazývat *konvergentní kvadrurní*.

10.7 Složené kvadrurní vzorec.

V předchozím odstavci jsme upozornili na to, že přes užitečnost kvadrurních vzorců vyšších řádů může být snaha o dosažení požadované přesnosti zvyšováním algebraického řádu vzorce neúspěšná. Navíc může být takový postup někdy nepohodlný, neboť uzly a koeficienty závisejí nejen na intervalu integrace $\langle a, b \rangle$, ale také na řádu kvadrurního vzorce. Změna řádu kvadrurního vzorce pak znamená nutnost nové uzly a koeficienty buď postupně vypočítat, nebo jejich dostatečně široký soubor uchovávat v paměti počítače.

Tyto skutečnosti vedou k tomu, že se při numerickém výpočtu integrálů mnohdy zvyšuje přesnost aproximace nikoli zvyšováním algebraického řádu použitého kvadrurního vzorce, ale postupným dělením intervalu $\langle a, b \rangle$ a užíváním stále stejného vzorce na jednotlivých dílčích intervalech (později uvidíme, že chyba aproximace při zmenšování $b - a$ obecně klesá).

Postupujme tedy formálně tak, že interval $\langle a, b \rangle$ rozdělíme na konečný počet intervalů $\langle a_j, b_j \rangle$, $j = 1, 2, \dots, m$ ($a_1 = a$, $b_m = b$, $b_j = a_{j+1}$), a položíme

$$(10.7.1) \quad I(f) = \int_a^b f(x)dx = \sum_{j=1}^m \int_{a_j}^{b_j} f(x)dx \approx \sum_{j=1}^m \int_{a_j}^{b_j} \varphi_j(x)dx$$

kde $\varphi_j(x)$ je aproximací funkce f na intervalu $\langle a_j, b_j \rangle$. Sečteme-li kvadraturní vzorce pro jednotlivé intervaly $\langle a_j, b_j \rangle$, $j = 1, 2, \dots, m$, dostáváme kvadraturní vzorec pro celý interval $\langle a, b \rangle$, který budeme nazývat *vzorcem složeným*; kvadraturním vzorcům pro dílčí intervaly $\langle a_j, b_j \rangle$ budeme říkat *vzorce základní*. Použití složeného kvadraturního vzorce je ekvivalentní náhradě funkce f aproximací φ takovou, že $\varphi(x) = \varphi_j(x)$ pro $x \in \langle a_j, b_j \rangle$, $j = 1, 2, \dots, m$.

Aproximujeme-li např. integrál přes každý z dílčích intervalů lichoběžníkovým pravidlem, dostáváme jako výsledek složené lichoběžníkové pravidlo. Použití složeného lichoběžníkového pravidla je pak ekvivalentní aproximaci funkce f funkcí φ , která je na každém intervalu $\langle a_j, b_j \rangle$ polynomem prvního stupně interpolujícím funkci f v uzlech a_j a b_j . Grafem funkce φ je lomená čára.

10.8 Úloha.

Ukažte, že algebraický řád lichoběžníkového pravidla a obdélníkového pravidla z příkl. 10.5 je roven 1.

[Návod: Vypočítejte těmito kvadraturními vzorci přibližnou hodnotu integrálu $I(f)$ pro $f(x) \approx 1$, $f(x) = x$, $f(x) = x^2$.]

11 Newtonovy-Cotesovy kvadraturní vzorce

11.1 Odvození Newtonových-Cotesových vzorců.

Základní tvar těchto kvadraturních vzorců vznikne tak, že na intervalu integrace nahradíme integrovanou funkci f jejím interpolačním polynomem s ekvidistantními uzly. Newtonovy-Cotesovy kvadraturní vzorce se většinou používají ve složeném tvaru z toho důvodu zaměříme náš popis a další označení tak, abychom mohli snadno zapsat jak vzorce základní, tak složené. Při konstrukci složených vzorců budeme pro jednoduchost interval $\langle a, b \rangle$ dělit na intervaly $\langle a_j, b_j \rangle$ rovněž ekvidistantním způsobem, tj. všechny tyto dílčí intervaly budou stejně dlouhé. Takto vzniklé složené vzorce budou mít velmi přehlednou strukturu a algoritmy výpočtu podle nich se dají snadno realizovat už pomocí kapesního kalkulátoru.

Pro pohodlný zápis nebudeme v čl. 11 odlišně označovat uzly základních kvadraturních vzorců a koncové body dílčích intervalů složených vzorců. Budeme vycházet z toho, že celý interval $\langle a, b \rangle$ ekvidistantně rozdělíme pomocí bodů

$$(11.1.1) \quad x_k = x_0 + kh, \quad k = 0, 1, \dots, N,$$

vzdálených od sebe h a takových, že $x_0 = a$, $x_N = b$. Bude tedy

$$(11.1.2) \quad h = \frac{b - a}{N}.$$

11.1.1 Obdélníkové pravidlo.

Základní kvadraturní vzorec zde odvodíme tak, že na intervalu $\langle x_k, x_{k+1} \rangle$ nahradíme funkci f konstantou rovnou hodnotě $f(x_k + \frac{1}{2}h)$ (viz příkl. 10.5). Dostáváme

$$(11.1.3) \quad \int_{x_k}^{x_{k+1}} f(x) dx \approx hf \left(x_k + \frac{1}{2}h \right) \equiv R_z(f; h).$$

Složený vzorec má tvar (interval $\langle a, b \rangle$ dělíme na N dílčích intervalů)

$$(11.1.4) \quad \int_a^b f(x) dx = \sum_{k=0}^{N-1} \int_{x_k}^{x_{k+1}} f(x) dx \approx h \sum_{k=0}^{N-1} f \left(x_k + \frac{1}{2}h \right) \equiv R_z(f; h)$$

a odpovídá tomu, že jsme nahradili funkci f na intervalu $\langle a, b \rangle$ po částech konstantní funkcí (srov. s definicí integrálu).

11.1.2 Lichoběžníkové pravidlo.

Základní vzorec odvodíme tak, že na intervalu $\langle x_k, x_{k+1} \rangle$ nahradíme funkci f lineárním interpolačním polynomem, který interpoluje f v uzlech x_k a x_{k+1} . Zcela stejně jako v příkl. 10.5 dostaneme

$$(11.1.5) \quad \int_{x_k}^{x_{k+1}} f(x) dx \approx \frac{h}{2} [f(x_k) + f(x_{k+1})] \equiv T_z(f; h).$$

Složený vzorec má tvar

$$(11.1.6) \quad \int_a^b f(x) dx \approx \frac{h}{2} [f(x_0) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{N-1}) + f(x_N)] = \\ = h \left[\frac{1}{2}f(x_0) + \sum_{k=1}^{N-1} f(x_k) + \frac{1}{2}f(x_N) \right] \equiv T(f; h)$$

a odpovídá nahrazení funkce f na intervalu $\langle a, b \rangle$ aproximací φ , která je po částech polynom prvního stupně.

11.1.3 Simpsonovo pravidlo.

Základní vzorec zde odvodíme tak, že na intervalu $\langle x_k, x_{k+2} \rangle$ nahradíme funkci f interpolačním polynomem φ_k druhého stupně (parabola), který interpoluje f v bodech x_k, x_{k+1}, x_{k+2} . Po

integraci funkce φ_k dostáváme (viz úlohu 11.10.1)

$$(11.1.7) \quad \int_{x_k}^{x_{k+2}} f(x) dx \approx \int_{x_k}^{x_{k+2}} \varphi_k(x) dx = \\ = \frac{h}{3} [f(x_k) + 4f(x_{k+1}) + f(x_{k+2})] \equiv S_z(f; h).$$

Složený vzorec má tvar

$$(11.1.8) \quad \int_a^b f(x) dx \approx \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots \\ \dots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)] \equiv S(f; h);$$

z jeho konstrukce vyplývá, že v případě složeného Simpsonova pravidla musí N být sudé číslo. (Interval $\langle a, b \rangle$ jsme tentokrát rozdělili na $N/2$ dílčích intervalů.)

11.2 Příklad.

Vypočítejme základními kvadraturními vzorci z předchozího odstavce přibližnou hodnotu integrálu

$$\int_1^{1,2} e^x dx.$$

Výpočet provedeme na 5D. Přesná hodnota integrálu (na 5D) je 0,60184.

Dostáváme

$$R_z(e^x; 0, 2) = 0,2e^{1,1} \doteq 0,60083$$

a dále

$$T_z(e^x; 0, 2) = \frac{0,2}{2}(e + e^{1,2}) \doteq 0,60384.$$

Chyba aproximace obdélníkovým pravidlem je tedy 0,00101 a chyba lichoběžníkového pravidla je $-0,00200$. Všimněme si toho, že obdélníkové pravidlo má vskutku o něco menší chybu než lichoběžníkové a že obě chyby mají opačná znaménka (srov. příkl. 10.5).

Simpsonovo pravidlo při $h = 0,1$, dá

$$S_z(e^x; 0, 1) = \frac{0,1}{3}(e + 4e^{1,1} + e^{1,2}) \doteq 0,60184.$$

což je na 5D přesná hodnota hledaného integrálu.

11.3 Algoritmus složeného Simpsonova pravidla.

Algoritmus výpočtu podle složeného obdélníkového a složeného lichoběžníkového pravidla se prakticky redukuje na přímočarý výpočet sumy tvaru $\sum f(x_j)$. Algoritmus výpočtu podle Simpsonova pravidla je jen nepatrně komplikovanější. Vstupním parametrem všech algoritmů pro výpočet

integrálu $I(f)$ je mimo jiné soubor hodnot integrované funkce ve všech uzlech použitého kvadrurního vzorce. Tento soubor hodnot může být např. zadán tabulkou. Při výpočtu na počítači je snad ještě častější ten způsob, že při numerickém výpočtu integrálu dodává potřebné funkční hodnoty podprogram, konstruovaný uživatelem daného kvadrurního algoritmu (např. na základě vzorce popisujícího integrovanou funkci). Skutečnost, že soubor hodnot funkce f je tím či oním způsobem k dispozici, znázorníme tak, že do vstupních parametrů algoritmu zahrneme symbol " $f(x)$ ".

Následuje jedna z možností, jak algoritmicky realizovat složené Simpsonovo pravidlo:

(11.3.1) Vstup: $a, b, f(x), N$.
 Konstanta, zda N je sudé, $N \leq 2$.
 $h = (b - a)/N$;
 $s = f(a) + 4f(a + h) + f(b)$; $x = a + 2h$.
 Je-li $N = 2$, jdi na POKR.
 Pro $k = 2, 4, 5, \dots, N - 1$:
 $s = s + 2f(x) + 4f(x + h)$;
 $x = x + 2h$.
 POKR: $s = hs/3$.
 Výstup: $S(f; (b - a)/N) = s$.

11.4 Příklad.

Ukážeme výsledky použití složeného obdélrníkového a lichoběžníkového pravidla k přibližnému výpočtu integrálu funkce zadané tabulkou hodnot (tab. 22). Počítáme přibližnou hodnotu

$$\int_1^{0,8} f(x)dx$$

pomocí kvadrurních vzorců $R(f; h)$ a $T(f; h)$. Výsledky (na 5D) jsou shrnuty v tab 23.

Tabulka 22 byla získaná z hodnot funkce $f(x) = (\sin x)/x$. Přesná hodnota počítaného integrálu na 4D je 0,7721. Ověřte, že v tomto příkladu jsou chyby $I(f) - R(f; h)$ zhruba úměrné $I(f) - T(f; h)$.

Tabulka 22

x	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
$f(x)$	1	0,998 33	0,993 34	0,985 07	0,973 55	0,958 85	0,941 07	0,920 31	0,896 70

Tabulka 23

h	$R(f; h)$	$T(f; h)$
0,8	0,778 84	0,758 68
0,4	0,773 76	0,768 76
0,2	0,772 51	0,771 26
0,1	-	0,771 89

11.5 Chyba obdélníkového, lichoběžníkového a Simpsonova pravidla.

Předpokládáme-li, že funkce f má dostatečný počet spojitých derivací, můžeme na základě vztahu (10.6.3), kam za $E(x)$ dosadíme konkrétní vyjádření pro chybu aproximace příslušným interpolačním polynomem, odvodit vzorce pro chybu základních kvadraturních vzorců R_z , T_z a S_z . Celý postup je technicky poměrně komplikovaný a zainteresovaného čtenáře odkazujeme na podrobné učebnice numerické matematiky, např. [26]. [Upozorňujeme např. na to, že nelze přímočaře integrovat vztah (3.10.1), neboť ξ je tam blíže neznámou funkcí proměnné x .]

Výsledkem takového odvození jsou vztahy

$$(11.5.1) \quad \int_{x_k}^{x_{k+1}} f(x) dx = R_z(f; h) + \frac{h^3}{24} f''(\xi_k^R),$$

$$(11.5.2) \quad \int_{x_k}^{x_{k+1}} f(x) dx = T_z(f; h) - \frac{h^3}{12} f''(\xi_k^T),$$

$$(11.5.3) \quad \int_{x_k}^{x_{k+2}} f(x) dx = S_z(f; h) - \frac{h^5}{90} f^{(4)}(\xi_k^S),$$

kde ξ_k^R a ξ_k^T jsou blíže neurčené body z intervalu (x_k, x_{k+1}) , zatímco ξ_k^S leží v intervalu (x_k, x_{k+2}) . U obdélníkového a lichoběžníkového pravidla zde předpokládáme spojitost druhé derivace integrované funkce, u Simpsonova pravidla spojitost čtvrté derivace na příslušném intervalu integrace. Opět vidíme, že obdélníkové a lichoběžníkové pravidlo integruje přesně polynomy prvního stupně. Simpsonovo pravidlo je přesné dokonce pro všechny polynomy třetího stupně.

Ze vztahů (11.5.1) až (11.5.3) se snadno odvodí analogické vztahy pro složené vzorce. Ukážeme to na příkladu lichoběžníkového pravidla. Sečtením vzorců (11.5.2) pro $k = 0, 1, \dots, N-1$ dostaneme

$$(11.5.4) \quad \int_a^b f(x) dx = S(f; h) - \frac{h^3}{12} \sum_{k=0}^{N-1} f''(\xi_k^T),$$

Předpokládáme-li, že f'' je funkce spojitá na celém intervalu $\langle a, b \rangle$, plyne ze základních vlastností spojitých funkcí, že suma na pravé straně (11.5.4) se rovná $Nf''(\xi)$, kde ξ je blíže neurčená bod intervalu $\langle a, b \rangle$.

Z (11.5.4) tak dostáváme konečný vztah pro chybu složeného lichoběžníkového pravidla

$$(11.5.5) \quad \int_a^b f(x)dx = T(f; h) - (b-a)\frac{h^2}{12}f''(\xi), \quad \xi \in \langle a, b \rangle,$$

(použili jsme ještě to, že $Nh = x_n - x_0 = b - a$.) Za stejného předpokladu o funkci f (tj. f'' je spojitá na $\langle a, b \rangle$) platí

$$(11.5.6) \quad \int_a^b f(x)dx = R(f; h) + (b-a)\frac{h^2}{24}f''(\xi), \quad \xi \in \langle a, b \rangle,$$

a je-li $f^{(4)}$ spojitá na $\langle a, b \rangle$, dostáváme

$$(11.5.7) \quad \int_a^b f(x)dx = S(f; h) - (b-a)\frac{h^4}{180}f^{(4)}(\xi), \quad \xi \in \langle a, b \rangle,$$

Z (11.5.5) až (11.5.7) vidíme, že za našich předpokladů o funkci f jsou při $h \rightarrow 0$ chyby složeného obdélníkového a lichoběžníkového pravidla velikosti $O(h^2)$, chyba složeného Simpsonova pravidla velikosti $O(h^4)$. Pro $h \rightarrow 0$ (a $N \rightarrow \infty$) tedy zejména přibližná hodnota získaná Simpsonovým vzorcem rychle konverguje k přesné hodnotě integrálu. Nebude-li předpoklad o patřičné hladkosti integrované funkce splněn, může být naproti tomu konvergence k integrálu z takové funkce velmi pomalá.

V příkladu 11.2 jsme obdélníkovým pravidlem získali z jedné funkční hodnoty řádově stejně přesný výsledek, jako dalo lichoběžníkové pravidlo ze dvou funkčních hodnot. Chyba obou vzorců ve složeném tvaru je $O(h^2)$, a i když základní obdélníkové pravidlo užívá pouze jedno hodnotu funkce f , kdežto lichoběžníkové pravidlo dvě hodnoty, tento rozdíl se u složených vzorců stírá. Při pevném $h = (b-a)/N$ je k výpočtu $R(f; h)$ zapotřebí N funkčních hodnot, k výpočtu $T(f; h)$ $N+1$ hodnot [viz (11.1.4), (11.1.6)]. Jsou tedy oba vzorce zhruba stejně efektivní.

Praktické použití vztahů (11.5.5) až (11.5.7) je limitováno tím, že neznáme čísla ξ , která v těchto vztazích vystupují; víme jen, že leží někde na intervalu $\langle a, b \rangle$. Uvedené vztahy tedy dávají spíše kvalitativní informaci o chování chyby při $h \rightarrow 0$. Stejně jako při odhadu chyby interpolace z nich můžeme odvodit odhady chyby popsanych slozenych kvadrturnich vzorců:

$$(11.5.8) \quad |I(f) - R(f; h)| \leq (b-a)\frac{h^2}{24}M_2,$$

$$(11.5.9) \quad |I(f) - T(f; h)| \leq (b-a)\frac{h^2}{12}M_2,$$

$$(11.5.10) \quad |I(f) - S(f; h)| \leq (b-a)\frac{h^4}{180}M_4,$$

kde $M_i = \max_{\langle a, b \rangle} |f^{(i)}(x)|$.

Tyto odhady jsou ale zpravidla pesimistické, v mnoha případech je skutečná chyby značně menší.

11.6 Příklad.

Určeme h tak, abychom měli zaručeno, že lichoběžníkovým pravidlem vypočítáme integrál

$$I = \int_2^3 (x-1)^{-1} dx$$

s chybou velikosti nejvýše 10^{-3} .

Podle (11.5.9) k tomu stačí, aby platila nerovnost

$$(b-a) \frac{h^2}{12} \max_{(2,3)} |f''(x)| \leq 10^{-3},$$

což po dosazení dává $\frac{1}{2}h^2 \cdot 2 \leq 10^{-3}$. Odtud dostáváme $h \leq (\sqrt{6}) \cdot 10^{-1,5}$ a dále $N = 6^{-1/2} \cdot 10 \sqrt{10} \doteq 12,9$. Stačí tedy vzít $N = 13$ a $h = \frac{1}{13}$. Provedeme-li výpočet s tímto N podle (11.1.6), dostaneme $T(f; \frac{1}{13}) \doteq 0,69352$ (na 5D). Přesná hodnota je $I = \ln 2 \doteq 0,69315$, a skutečná chyba je tedy asi $3,7 \cdot 10^{-4}$.

11.7 Obecné Newtonovy-Cotesovy vzorce.

Základní vzorce $T_z(f; h)$ a $S_z(f; h)$ jsou speciální případy kvadrturních vzorců tvaru

$$(11.7.1) \quad \int_{x_{k+n}}^{x_k} f(x) dx \approx \sum_{i=0}^n w_i f(x_{k+i}),$$

kteřé se nazývají *uzavřené* ⁶⁾ *Newtonovy-Cotesovy vzorce* a získají se integrací interpolačního polynomu n -tého stupně, který interpoluje funkci f v bodech $x_k, x_{k+1}, \dots, x_{k+n}$. Vzorec $T_z(f; h)$ odpovídá volbě $n = 1$, Simpsonovo pravidlo $S_z(f; h)$ volbě $n = 2$.

Uzavřené Newtonovy-Cotesovy vzorce se dají konstruovat pro každé n , ale z základním tvaru nepatří mezi konvergentní kvadratury (viz odst. 10.6). Algebraický řád těchto vzorců je n pro n liché, $n+1$ pro n sudé (srov. Simpsonovo pravidlo). Jako další příklad uzavřeného Newtonova-Cotesova vzorce uvádíme kvadrturní vzorec pro $n = 3$ (tzv. *pravidlo tří osmin*):

$$(11.7.2) \quad \int_{x_{k+3}}^{x_k} f(x) dx = \frac{3h}{8} [f(x_k) + 3f(x_{k+1}) + 3f(x_{k+2}) + f(x_{k+3})] - \frac{3h^5}{80} f^{(4)}(\xi_k).$$

Existují také *otevřené Newtonovy-Cotesovy vzorce*, které jsou tvaru

$$(11.7.3) \quad \int_{x_{k+n}}^{x_k} f(x) dx = \sum_{i=1}^{n-1} w_i f(x_{k+i});$$

⁶⁾ To proto, že mezi uzly patří i koncové body intervalu integrace.

tyto vzorce nepoužívají funkční hodnoty v koncových bodech intervalu $\langle x_k, x_{k+n} \rangle$. Jsou to opět interpolační kvadraturní vzorce s ekvidistantními uzly odvozené obvyklým způsobem. Nejjednodušším příkladem ($n = 2$) je obdélníkové pravidlo $R_z(f; 2h)$, pro které (11.5.1) platí

$$(11.7.4) \quad \int_{x_{k+2}}^{x_k} f(x) dx = 2hf(x_{k+i}) + \frac{h^3}{3} f''(\xi_k).$$

Ani otevřené Newtonovy-Cotesovy vzorce v základním tvaru nepatří mezi konvergentní kvadratury. Jejich algebraický řád je $n - 1$ pro n sudé, $n - 2$ pro n liché; používají ovšem o dva uzly méně než uzavřený vzorec pro týž interval integrace.

Další příklady otevřených Newtonových-Cotesových vzorců jsou

$$(11.7.5) \quad \int_{x_{k+3}}^{x_k} f(x) dx = \frac{3h}{2} [f(x_{k+1}) + f(x_{k+2})] + \frac{3h^3}{4} f''(\xi_k),$$

$$(11.7.6) \quad \int_{x_{k+4}}^{x_k} f(x) dx = \frac{4h}{3} [2f(x_{k+1}) - f(x_{k+2}) + 2f(x_{k+3})] + \frac{14h^5}{45} f^{(4)}(\xi_k),$$

Otevřené vzorce mají sotva nějakou výhodu oproti uzavřeným vzorcům se stejným počtem uzlů. Mají však určitý význam v metodách pro numerickou integraci obyčejných diferenciálních rovnic.

Podrobnější informace o obecných Newtonových-Cotesových kvadraturních vzorcích včetně důkazů tvrzení, která jsme uvedli, lze nalézt např. v [3], [26].

11.8 Jak dosáhnout požadované přesnosti.

V odstavci 11.5 jsme uvedli vzorce pro chybu nejčastěji používaných Newtonových-Cotesových vzorců a ukázali, jak odhadnout chybu kvadratury s daným h , příp. jak určit h takové, aby chyba kvadratury byla menší než předem zadaná tolerance. V celém postupu jsou však dva háčky: Především, výrazy pro chybu kvadratury obsahují derivace (často poměrně vysokého řádu), které se v praxi většinou odhadují jen pracně. Dále, výsledné odhady jsou většinou značně přehnané, pesimistické.

Při konkrétním výpočtu se proto zpravidla postupuje tak, že hledaný integrál se počítá opakovaně stále přesnějšími kvadraturními vzorci a ze shody výsledků se usuzuje na to, zda již bylo nebo nebylo požadované přesnosti dosaženo. Celý postup je do značné míry heuristický a vyžaduje určitou zkušenost. Z toho, co jsme řekli o Newtonových-Cotesových vzorcích, je zřejmé, že k postupnému zpřesňování aproximace hledaného integrálu nebude vhodné postupně zvyšovat řád použitého vzorce (proč?). Vhodnější je použít jeden základní kvadraturní vzorec (např. lichoběžníkové pravidlo) a zvyšovat počet uzlů složené formy tohoto vzorce. Pro účely posouzení přesnosti tímto způsobem se osvědčilo postupné zdvojnásobování počtu dílků, na něž dělíme interval $\langle a, b \rangle$ (tj. půlení h).

Runge navrhl poněkud preciznější způsob odhadu dosažené přesnosti, kterému se často říká *metoda polovičního kroku* [2]. Vyjdeme z toho, že výrazy pro chybu složených Newtonových-Cotesových vzorců mají vesměs tvar $e(f) = h^k M$, kde $h = (b - a)/N$, k je pevné číslo a M

je součin konstanty a k -té derivace integrované funkce v jistém bodě intervalu $\langle a, b \rangle$. Označíme-li I přesnou hodnotu integrálu a $K(h)$ přibližnou hodnotu získanou daným kvadraturním vzorcem, platí

$$(11.8.1) \quad I = K(h) + h^k M.$$

Vypočteme-li nyní integrál I stejným vzorcem, ale při rozdělení intervalu $\langle a, b \rangle$ na dvojnásobný počet intervalů (budeme tedy pracovat s $h/2$ na místě h), dostaneme

$$(11.8.2) \quad I = K\left(\frac{h}{2}\right) + \left(\frac{h}{2}\right)^k M_1.$$

Označíme-li chybu aproximace vzorcem $K(h/2)$ jako ε , bude

$$(11.8.3) \quad I = K\left(\frac{h}{2}\right) + \varepsilon, \quad I = K(h) + \frac{2^k \varepsilon M}{M_1}.$$

Předpokládejme nyní, že se derivace, která vystupuje ve vzorci pro $e(f)$, na intervalu $\langle a, b \rangle$ příliš nemění, a že tedy můžeme položit $M \approx M_1$. Za tohoto předpokladu z (11.8.3) plyne $K(h/2) + \varepsilon \approx K(h) + 2^k \varepsilon$ a vypočteme-li odtud ε , máme

$$(11.8.4) \quad \varepsilon = \frac{1}{2^k - 1} \left[\left(K\left(\frac{h}{2}\right) - K(h) \right) \right].$$

Známe-li tedy číslo k , můžeme podle (11.8.4) odhadnout chybu aproximace vzorcem $K(h/2)$. Celý postup je vlastně jedním krokem Richardsonovy extrapolace (viz čl. 9) a dává použitelné výsledky jen při splnění příslušných předpokladů o funkci f (kterých?).

11.9 Příklad.

Odhadněme Rungovou metodou chyby aproximaci $R(f; 0, 2)$ a $T(f; 0, 2)$ z příkl. 11.4. Podle (11.5.6) a (11.5.5) je v obou případech $k = 2$.

Pro obdélníkové pravidlo dostáváme

$$\varepsilon \approx \frac{1}{3} [R(f; 0, 2) - R(f; 0, 4)] \doteq -0,0004;$$

skutečná chyba na 4D je stejná. Pro lichoběžníkové pravidlo máme

$$\varepsilon \approx \frac{1}{3} [T(f; 0, 2) - T(f; 0, 4)] \doteq 0,0008,$$

což je opět na 4D rovno skutečné chybě. Tato mimořádně pěkná shoda je důsledkem příznivých vlastností integrované funkce.

11.10 Úlohy.

11.10.1

Odvoďte základní Simpsonovo pravidlo (11.1.7).

[Návod: Vyjádřete φ_k podle vzorce (3.5.2) a proveďte integraci.]

11.10.2

Odvoďte základní kvadrurní vzorce $R_z(f; h)$, $T_z(f; h)$ a $S_z(f; h)$ z odst. 11.1 metodou neurčitých koeficientů, tj. vyjděte ze vztahů (11.7.1) a (11.7.3) s vhodným n a požadujte, aby byly přesné pro polynomy příslušných stupňů.

[Návod: Použijte polynomy $1, x, x^2, x^3$ a sestavte soustavy lineárních algebraických rovnic pro koeficienty hledaných kvadrurních vzorců.]

11.10.3

Ukažte, že platí $T(f; h) = [T(f; 2h) + R(f; 2h)]/2$, $R(f; h) = 2T(f; h/2) - T(f; h)$.

11.10.4

Rychlost v rakety vypuštěné ze Země se měří (v m/s) každých 5 s. V tabulce 24 jsou naměřené hodnoty za prvních 50 s letu. Použijte složené Simpsonovo pravidlo a vypočítejte přibližnou výšku rakety po 50 s letu; předpokládáme, že raketa letí vertikálně.

[11 485 m.]

Tabulka 24

t	0	5	10	15	20	25	30	35	40	45	50
v	0	20,2	60,0	113,9	176,1	241,5	303,5	357,5	397,5	418,0	413,0

11.10.5

Částice se pohybuje po přímce tak, že její rychlost v v čase t je dána vztahem $v(t) = t(8-t^3)^{1/2}$. Použijte složené Simpsonovo pravidlo s $N = 8$ a určete přibližně, jak velkou dráhu s částice urazí od $t = 0$ do $t = 2$.

[Návod: $v(t) = s'(t)$. Výsledek: $s \doteq 4,109$.]

11.10.6

Vypočítejte $\int_0^{\pi/4} \cos x \, dx$ složeným lichoběžníkovým pravidlem. Počet uzlů volte tak, aby bylo zaručeno, že chyba aproximace bude menší než 0,01.

[$N = 3$, $T(f; h) \doteq 0,7031$.]

11.10.7

Vypočítejte $\int_0^1 \cos x \, dx$ složeným lichoběžníkovým pravidlem s $h = 0,2$ a $h = 0,1$. Odhadněte chybu výsledku s $h = 0,1$ a) podle (11.5.9); b) metodou polovičního kroku.

$[T(f; 0, 2) \doteq 0,744368, T(f; 0, 1) \doteq 0,746211$ Odhad chyby: a) $|f''| \leq 2, |I - T| \leq \frac{2}{2000} \doteq 0,0017$; b) $0,000614$. Skutečná chyba cca $6 \cdot 10^{-4}$.]

11.10.8

Vypočítejte $\int_1^3 (1/x) dx$ lichoběžníkovým pravidlem s $h = 1$ a $h = 0,5$. Odhadněte chybu výsledku s $h = 0,5$ stejně jako v úloze 11.10.7.

$[T(f; 0, 5) \doteq 1,1167$. Skutečná chyba je (na 4D) $-0,0181$; Odhad chyby a) $-0,0833$; b) $-0,0167$.]

12 Eulerův-Maclaurinův vzorec. Rombergova metoda

12.1 Eulerův-Maclaurinův vzorec.

V odstavci 11.5 jsme uvedli vzorec (11.5.2) pro chybu lichoběžníkového pravidla, platný za předpokladu, že integrovaná funkce f má na intervalu $\langle x_k, x_{k+1} \rangle$ spojitou druhou derivaci. Předpokládáme-li o funkci f více, totiž to, že má $2m + 2$ spojitě derivace, dá se ukázat [12], že platí

$$(12.1.1) \quad \int_{x_k}^{x_{k+1}} f(x) dx = T_z(f; h) - \sum_{s=1}^m \frac{B_{2s}}{(2s)!} h^{2s} [f^{(2s-1)}(x_{k+1}) - f^{(2s-1)}(x_k)] - \frac{B_{2m+2}}{(2m+2)!} h^{2m+3} f^{(2m+2)}(\xi_k),$$

kde $\xi_k = (x_k, x_{k+1})$ a koeficienty B_2, B_4, B_6, \dots jsou tzv. Bernoulliova čísla (viz [12]). Je např. $B_2 = \frac{1}{6}, B_4 = -\frac{1}{30}, B_6 = \frac{1}{42}, B_8 = \frac{1}{30}, \dots$. Vztahu (12.1.1) se říká *Eulerův-Maclaurinův vzorec*.

Má-li funkce f na intervalu $\langle a, b \rangle$ $2m + 2$ spojitě derivace, plyne z Eulerova-Maclaurinova vzorce pro složené lichoběžníkové pravidlo (podobně jako v odst. 11.5) vztah

$$(12.1.2) \quad \int_a^b f(x) dx = T(f; h) - \sum_{s=1}^m \frac{B_{2s}}{(2s)!} h^{2s} [f^{(2s-1)}(b) - f^{(2s-1)}(a)] - (b-a) \frac{B_{2m+2}}{(2m+2)!} h^{2m+2} f^{(2m+2)}(\xi_k),$$

kde $\xi = \langle a, b \rangle$. Tento vzorec je teoretickým základem Rombergovy metody pro výpočet určitého integrálu, kterou popíšeme v odst. 12.7 až 12.10. Eulerův-Maclaurinův vzorec má ale ještě řadu dalších aplikací; o některých z nich pojednávají odst. 12.2 až 12.6.

12.2 Kvadraturní vzorce s hodnotami derivací.

Známe-li hodnoty derivací funkce f v bodech $x = a$ a $x = b$, můžeme podle Eulerova-Maclaurinova vzorce pozměnit lichoběžníkové pravidlo tak, aby se tato informace zužítkovala. Algebraický řád takto vzniklých vzorců bude $2m + 1$ při použití prvních $2m - 1$ derivací a pro dostatečně hladké funkce bude chyba kvadratury řádu 2^{2m+2} (při $h \rightarrow 0$). Příkladem je vzorec

$$(12.2.1) \quad \int_a^b f(x)dx = T(f; h) - \frac{h^2}{12}[f'(a) - f'(b)] + (b - a) \frac{h^4}{720} f^{(4)}(\xi), \quad \xi \in \langle a, b \rangle,$$

přesný pro polynomy třetího stupně.

12.3 Výpočet integrálu z periodické funkce přes periodu.

Vzorec (12.1.2) ukazuje, že lichoběžníkové pravidlo je vysoce přesné pro ty funkce f , pro něž platí $f'(a) = f'(b)$, $f'''(a) = f'''(b)$, $f^{(5)}(a) = f^{(5)}(b)$, Příkladem takových funkcí jsou analytické periodické funkce, příp. periodické funkce s dostatečným počtem spojitých derivací, je-li interval integrace $\langle a, b \rangle$ celá jejich perioda. Podobné tvrzení platí i pro obdélníkové pravidlo (srov. úlohu 11.10.3).

Upozorňujeme však na to, že ani ro analytické periodické funkce v tomto případě obecně neplatí, že $T(f; h)$ je přesná hodnota integrálu $I(f)$, protože zbytek

$$(b - a)B_{2m+2}h^{2m+2}f^{(2m+2)}(\xi)/(2m + 2)!$$

na pravé straně vztahu (12.1.2) obecně nemusí pro $m \rightarrow \infty$ konvergovat k nule. Platí však, že chyba lichoběžníkového pravidla je velikosti $O(h^{2m+2})$ pro libovolné takové, že f má spojitou $(2m + 2)$ -hou derivaci. V tomto smyslu jde tedy chyba lichoběžníkového pravidla pro analytické periodické funkce přes periodu při $h \rightarrow 0$ rychleji než libovolná mocnina h . Nemá zde proto smysl odhadovat chybu Rungeovou metodou.

12.4 Příklad.

Vypočtěme

$$\int_0^{2\pi} (1 + \cos 4x) dx$$

složeným lichoběžníkovým pravidlem. Integrovaná funkce je analytická periodická funkce (trigonometrický polynom), interval integrace je roven její periodě. Přesná hodnota integrálu je 2π .

Volíme-li $h = \pi/2$, $N = 4$, dostáváme podle (11.1.6)

$$T(f; \pi/2) = \frac{1}{2}\pi(1 + 2 + 2 + 2 + 1) = 4\pi$$

Chyba je tedy 2π !

Vezmeme-li ale $h = \pi/4$, $N = 8$, dostaneme

$$T(f; \pi/4) = \frac{1}{4}\pi(1 + 0 + 2 + 0 + 2 + 0 + 2 + 0 + 1) = 2\pi$$

což je přesná hodnota hledaného integrálu. Dá se ukázat [13], že složené lichoběžníkové pravidlo s $k + 2$ uzly je přesné pro všechny trigonometrické polynomy k -tého stupně a stupňů menších (tj. obsahující nejvýše členy $\cos kx$, $\sin kx$), integrované přes periodu 2π .

12.5 Nekonečný interval integrace.

Podobná situace jako v odst. 12.3 může nastat při výpočtu integrálů typu

$$\int_{-\infty}^{+\infty} f(x) dx$$

lichoběžníkovým nebo obdélníkovým pravidlem. Aproximujeme-li uvedený integrál integrálem

$$\int_{-R_1}^{R_2} f(x) dx$$

za předpokladu, že funkce f a její nižší derivace jsou pro $x \leq -R_1$ a $x \geq R_2$ malé, dává nám výpočet pomocí $T(f; h)$ nebo $R(f; h)$ často překvapivě dobré výsledky. Poznamenejme ještě, že v tomto případě je situace obdobná jako v odst. 12.3, a že tedy nemá smysl provádět extrapolaci na $h = 0$ či užívat Rungovu metodu pro odhad chyby.

12.6 Příklad.

Vypočítejme

$$I = \int_{-\infty}^{+\infty} \exp(-x^2) dx.$$

Pro $x = \pm 4$ (tudíž i pro $|x| \geq 4$) je integrand menší než $0,5 \cdot 10^{-6}$. Počítáme-li tedy

$$I_1 = \int_{-4}^4 \exp(-x^2) dx.$$

lichoběžníkovým pravidlem (na 6D), dostáváme $T(f; 1) \doteq 1,772\,636$ a $T(f; 0,5) \doteq 1,772\,453$. Přesná hodnota integrálu I je $\sqrt{\pi} \doteq 1,772\,454$. Poznamenejme ještě, že $|I - I_1| < 10^{-7}$.

12.7 Rombergova integrační metoda.

Z Eulerova-Maclaurinova vzorce plyne, že má-li funkce f na intervalu $\langle a, b \rangle$ spojitou $(2m + 2)$ -hou derivaci, pak existují konstanty a_1, a_2, \dots, a_m takové, že platí

$$T(f; h) - I(f) = a_1 h^2 + a_2 h^4 + \dots + a_m h^{2m} + O(h^{2m+2}).$$

Ny výsledky získané lichoběžníkovým pravidlem tedy můžeme aplikovat opakovanou Richardsonovu extrapolaci, kterou jsme popsali v čl. 9. Výsledný algoritmus se nazývá *Rombergova kvadratura* nebo *Rombergova integrace*.

V tomto speciálním případě bereme v algoritmu (9.4.3) $q = 2$ a z rozvoje (12.7.1) vyplývá $p = 2$, takže dostáváme tento algoritmus:

$$(12.7.2) \quad \begin{aligned} &\text{Pro } s = 0, 1, \dots, S : \\ &\quad T_{s0} = T(f; 2^{-s}H). \\ &\quad \text{Pro } k = 1, 2, \dots, s : \\ &\quad\quad T_{sk} = T_{s,k-1} + \frac{T_{s,k-1} + T_{s-1,k-1}}{4^k - 1}. \end{aligned}$$

Richardsonovou extrapolaci bychom mohli aplikovat na obdélníkové pravidlo (srov. úlohu 11.10.3). Lichoběžníkové pravidlo má však tu významnou výhodu, že k výpočtu T_{s0} můžeme využít $T_{s-1,0}$ a ušetřit práci s výpočtem funkčních hodnot (která je hlavní částí celkové výpočetní práce v Rombergově algoritmu). Stačí si uvědomit, že při výpočtu T_{s0} používáme poloviční h než při výpočtu $T_{s-1,0}$, a použijeme tedy všechny funkční hodnoty, z nichž jsme stanovili $T_{s-1,0}$ (a ovšem další navíc).

Hodnotu S v algoritmu (12.7.2) není ani zde třeba volit předem. Výpočet provádíme pro postupně rostoucí s a sledujeme rozdíly $|T_{sk} - T_{s-1,k}|$. Je-li tento rozdíl menší než předepsaná chyba, považujeme $T_{s,k+1}$ za přibližnou hodnotu integrálu $I(f)$ s požadovanou přesností. Při výpočtu v ruce vystupují v algoritmu pomocné veličiny $\Delta/3, \Delta/15, \Delta/63, \Delta/255, \dots$ (srov. odst. 9.4).

12.8 Příklad.

Vypočtěme

$$\int_0^{0,8} x^{-1} \sin x dx$$

Rombergovou metodou (srov. příkl. 11.4). Položíme $H = 0,8$ a sestavíme schéma hodnot T_{sk} do tabulky podobně jako v příkl. 9.5 (tab. 25; pracujeme na 6D). Protože je $|T_{21} - T_{31}| = 10^{-6}$, klademe $I \approx T_{32} = 0,772095$. Přesná hodnota (na 6D) je $I \doteq 0,772095$.

Tabulka 25

s	h_s	$T_{s0} = T(f; h_s)$	$\Delta/3$	T_{s1}	$\Delta/15$	T_{s2}
0	0,8	0,758680				
1	0,4	0,768760	3 360	0,772120		
2	0,2	0,771262	834	0,772096	-2	0,772094
3	0,1	0,758887	208	0,772095	0	0,772095

12.9 Vlastnosti Rombergovy metody.

Rozbor vlastností Rombergovy metody provedli Bauer, Rutishauser a Stiefel v roce 1963. Jejich výsledky jsou značně hluboké a je mimo rámec této publikace uvádět je podrobněji; podáváme proto pouze celkový přehled. Některé informace lze najít např. v [2], [26].

Snadno se ukáže (viz úlohu 12.11.3), že druhý sloupec T_{s1} Rombergova schématu dává tytéž výsledky jako použití složeného Simpsonova pravidla, a je tedy přesný pro polynomy třetího stupně. Obecně platí, že sloupec T_{sk} dává výsledky, odpovídající výpočtu podle kvadraturního vzorce přesného pro polynomy $(2k + 1)$ -ního stupně; pro $k > 2$ ale tyto kvadraturní vzorce již nemají nic společného s vzorci Newtonovými-Cotesovými.

Rombergovo schéma v sobě skrývá celou řadu konvergentních kvadratur. Pro funkci f integrovatelnou (v Riemannově smyslu) platí $T_{sk} \rightarrow I(f)$ pro $s \rightarrow \infty$ a všechna $k = 0, 1, \dots$ a také $T_{kk} \rightarrow I(f)$ pro $k \rightarrow \infty$ (diagonála schématu). Dále se dá ukázat, že celá procedura je numerický stabilní (viz též příkl. 9.5 a úlohu 9.7.1).

Rombergova integrační metoda patří k velmi často používaným metodám pro přibližný výpočet určitého integrálu. Její význam je především v těch výpočtech, kde je požadována mála chyba aproximace. Existuje celá řada variant popsaného algoritmu (srov. odst. 9.6). I když je tato metoda velmi atraktivní, je třeba ji používat s rozmyslem. Metoda je neúčelná např. v případech integrálů studovaných v odst. 12.3 až 12.6. Dále, je to typická metoda pro hladké funkce a předpokladem pro její úspěšné použití je vždy existence dostatečného počtu derivací funkce f na celém intervalu $\langle a, b \rangle$ [a odtud plynoucí existence rozvoje (12.7.1)]. V opačném případě nepřináší Rombergova metoda žádné zrychlení konvergentního procesu oproti např. složenému lichoběžníkovému pravidlu.

12.10 Příklad.

Použijeme Rombergovu metodu k výpočtu integrálu

$$I = \int_0^{0,8} x^{1/2} dx.$$

Integrand nemá v bodě $x = 0$ konečnou derivaci. Výpočet provedeme na 5D, volíme $H = 0,8$. Výsledky zapisujeme již stručněji:

T_{s0}	T_{s1}	T_{s2}	T_{s3}	T_{s4}
0,35777				
0,43187	0,45657			
0,46030	0,46978	0,47066		
0,47092	0,47446	0,47477		
0,47482	0,47612	0,47623	0,47625	0,47626

Bylo by chybou usoudit ze shody dvou tučně vytištěných hodnot, že přibližná hodnota integrálu I je 0,47626. Je $|T_{43} - T_{33}| = 1,41 \cdot 10^{-3}$, můžeme tedy pouze říci, že $I \approx 0,476$. Přesná hodnota je $I \doteq 0,47703$. Rombergova metoda zde oproti lichoběžníkovému pravidlu nepřináší žádné zlepšení. Vidíme, že $T_{40} = T(f; 0,05) \doteq 0,475$ a za cenu práce navíc jsme v tomto případě nezískali podstatně přesnější výsledek.

12.11 Úlohy

12.11.1

Použijte kvadrurní vzorec (12.2.1) k výpočtu integrálu:

a) Z příkl. 11.2 (se stejným počtem uzlů; pracuje na 6D). Odhadněte chybu výsledku.

[0,601834; odhad chyby $1,5 \cdot 10^{-6}$.]

b) Z příkl. 11.6 (se stejným počtem uzlů).

[0,69315, tj. na 5D přesná hodnota.]

12.11.2

Vypočítejte lichoběžníkovým pravidlem s $h = \pi/2$ a $h = \pi/4$ integrál

$$(2\pi)^{-1} \int_0^{2\pi} \exp(2^{-1/2} \sin x) dx \doteq 1,128961.$$

Odhadněte chybu výsledku pro $h = \pi/4$ Rungovou metodou (odst. 11.8) a pokuste se vysvětlit, proč jste pro $h = \pi/2$ obdrželi mimořádně přesný výsledek.

[$T(f; \pi/2) \doteq 1,1276$; $T(f; \pi/4) \doteq 1,128961$. Odhad chyby: $4,5 \cdot 10^{-4}$. Návod k vysvětlení: viz odst. 12.3.]

12.11.3

Ukažte, že platí

$$S(f; h) = T(f; h) + \frac{T(f; h) - T(f; 2h)}{3}.$$

[Návod: Užijte (11.1.6) a (11.1.8).]

12.11.4

Vypočítejte integrál

$$\int_0^1 e^x dx$$

Rombergovou metodou s chybou zaručeně menší než 10^{-3} . Volte $H = 1$ a pracujte na 6D. Jaká je skutečná chyba získaného výsledku?

[$T_{22} \doteq 1,718283$; skutečná chyba je 10^{-6} .]

12.11.5

Vypočítejte integrál

$$\int_0^{\pi/2} e^{2x} \cos x dx = \frac{e^{\pi} - 2}{5} \doteq 4,228\,139$$

Rombergovou metodou s přesností 10^{-3} a 10^{-5} . Pracujte na 6D.

12.11.6

Vypočítejte integrál

$$\pi^{-1} \int_0^{\pi/2} \sin x dx$$

na 5 míst přesně Rombergovou metodou. Volte $H = \pi/2$ a pracujte na 6D.
[0,31831.]

12.11.7

[26] Vypočítejte Rombergovou metodou s $H = b - a$ aproximace T_{55} integrálů:

$$\begin{array}{ll} \text{a) } \int_{-4}^4 (1+x^2)^{-1} dx; & \text{b) } \int_0^1 e^{-10x} \sin x dx; \\ \text{c) } \int_0^5 x e^{-3x^2} dx; & \text{d) } \int_{-1}^1 (1-x^2)^{3/2} \cos x dx; \end{array}$$

Výsledky porovnejte s přesnými hodnotami integrálů.

[T_{55} : a) 2,65186; b) 0,0098970; c) 0,16659; d) 1,08296. Přesné hodnoty (zaokrouhleno): a) 2,65164; b) 0,0099010; c) 0,16667; d) 1,08291.]

12.11.8

Sestavte tabulku hodnot funkce

$$F(u) = \int_0^{\infty} (x^2 + 1)^{-1/2} e^{-ux} dx$$

pro $u = 0,5; 0,6; \dots 3,0$ s nepřesnostmi nejvýše 10^{-6} .

[Návod: Položte $ux = v$ Rombergovou metodou počítejte integrál v mezích od $v = 0$ do $v = 12$. Ukažte, že zbytek integrálu je zanedbatelný.]

13 Gaussovy kvadraturní vzorce

13.1 Vzorce s neekvidistantními uzly.

V odstavci 11.7 jsme uvedli, že Newtonovy-Cotesovy kvadraturní vzorce s $n + 1$ ekvidistantními (pevně zvolenými) uzly integrují přesně polynomy až do n -tého [příp. $(n + 1)$ -ního] stupně. Uvažujeme-li interpolační kvadraturní vzorec (10.6.1) s obecným rozložením uzlů $x_i, i = 0, 1, \dots, n$, na intervalu $\langle a, b \rangle$, dá se ukázat, že při vhodných x_i lze dosáhnout toho, aby algebraický řád vzorce (10.6.1) byl zhruba dvojnásobný a rovnal se $2n + 1$. Za uzly x_i je přitom třeba brát kořeny jistých ortogonálních polynomů. Tyto vzorce vyšší přesností se nazývají Gaussovy kvadraturní vzorce. Hodnoty jejich uzlů a vah je možno najít v tabulkách (např. [14]), příp. jsou již zabudovány do standardních programů, běžně zařazovaných do programového vybavení výpočetních středisek. Popis základních vlastností Gaussových kvadraturních vzorců zformulujeme jako větu.

13.2 Věta.

Budiž $\{Q_j, j = 0, 1, \dots\}$ soustava polynomů⁷⁾ ortogonálních na intervalu $\langle a, b \rangle$ s vahou $\omega(x) \equiv 1$ (spojitý případ). Zvolme pevně nezáporné celé číslo n a označme $x_i, i = 0, 1, \dots, n$, kořeny ortogonálního polynomu Q_{n+1} $(n + 1)$ -ního stupně. Sestrojme interpolační kvadraturní vzorec (10.6.1) s uzly x_i a koeficienty w_i danými vztahem (10.6.2) (Gaussův kvadraturní vzorec.)

Při této volbě uzlů $x_i, i = 0, 1, \dots, n$, pak pro každý polynom p_{2n+1} $(2n + 1)$ -ního stupně platí

$$(13.2.1) \quad \int_a^b p_{2n+1}(x) dx = \sum_{i=0}^n w_i p_{2n+1}(x_i).$$

Důkaz věty plyne téměř okamžitě z vlastností ortogonálních polynomů a lze jej nalézt např. v [2], [3]. Poznáváme pouze, že interpolační kvadraturní vzorec s uvedenými uzly existuje pro každé n , neboť všechny kořeny polynomů Q_{n+1} jsou reálné, různé a leží v intervalu $\langle a, b \rangle$ (viz odst. 5.5).

13.3 Vlastností Gaussových kvadraturních vzorců.

Věta 13.2 se dá dále upřesnit v tom smyslu, že algebraický řád Gaussova kvadraturního vzorce s $n + 1$ uzly je právě $2n + 1$. Je totiž dokázáno [13], že žádný interpolační kvadraturní vzorec s $n + 1$ uzly nemůže integrovat přesně všechny polynomy $(2n + 2)$ -tého stupně. Gaussovy vzorce mají tedy nejvyšší možný algebraický řád.

Snad ještě důležitější je ta vlastnost Gaussových vzorců, že při $n \rightarrow \infty$ tvoří konvergentní kvadraturu [3]. Konvergence k přesné hodnotě integrálu je přitom rychlejší, čím hladší je funkce f . Je dokázáno, že Gaussovy vzorce při $n \rightarrow \infty$ konvergují k přesné hodnotě integrálu dokonce i pro funkce, které jsou na intervalu $\langle a, b \rangle$ pouze po částech spojitě a mají tam konečný počet skoků. Pro takové funkce může být však konvergence velmi pomalá.

⁷⁾ Viz odst. 5.5 a 5.6.

V tabulkách a na počítačích bývají zpravidla uchovávány hodnoty koeficientů $w_i^{(0)}$ a uzlů $x_i^{(0)}$ Gaussových kvadraturních vzorců pro interval $\langle -1, 1 \rangle$. Koeficienty w_i a uzly x_i pro obecný interval $\langle a, b \rangle$ z nich získáme jednoduchou substitucí

$$(13.3.1) \quad w_i = \frac{b-a}{2} w_i^{(0)}, \quad x_i = \frac{b+a}{2} + \frac{b-a}{2} x_i^{(0)}, \quad i = 0, 1, \dots, n.$$

Z vlastností ortogonálních polynomů vyplývá [3], že uzly rozloženy symetricky vzhledem ke středu intervalu $\langle a, b \rangle$; pro váhy platí $w_i = w_{n-i} > 0$, $i = 0, 1, \dots, n$ ("symetrie").

13.4 Gaussovy kvadraturní vzorce pro interval $\langle -1, 1 \rangle$. Chyba aproximace

Soustava polynomů $\{Q_n\}$ ortogonálních na intervalu $\langle -1, 1 \rangle$ s vahou $\omega(x) \equiv 1$ jsou Legendrovy polynomy P_j , $j = 0, 1, \dots$, z odst. 5.6.3. Vezmeme-li za uzly kvadraturního vzorce postupně kořeny polynomů P_1 , P_2 a P_3 , dostáváme Gaussovy kvadraturní vzorce pro $n = 0, 1, 2$ a interval $\langle -1, 1 \rangle$:

$$(13.4.1) \quad \begin{aligned} \int_{-1}^1 f(x) dx &= 2f(0) + \frac{1}{3} f''(\xi), \\ \int_{-1}^1 f(x) dx &= f\left(-\sqrt{\frac{1}{3}}\right) + f\left(\sqrt{\frac{1}{3}}\right) + \frac{1}{135} f^{(4)}(\xi), \\ \int_{-1}^1 f(x) dx &= \frac{5}{9} f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{\frac{3}{5}}\right) + \frac{1}{15750} f^{(6)}(\xi), \end{aligned}$$

kde $\xi \in \langle -1, 1 \rangle$. Koeficienty a uzly vzorců vyšších řádů lze nalézt např. v [3], [4].

Je-li $f^{(2n+2)}$ spojitá na intervalu $\langle -1, 1 \rangle$, dokazuje se v učebnicích numerické analýzy [26], že chyba Gaussova vzorce (pro interval $\langle -1, 1 \rangle$) s $n+1$ uzly je rovna $d_{n+1} f^{(2n+2)}(\xi)$, kde $\xi \in \langle -1, 1 \rangle$ a

$$(13.4.2) \quad d_n = \frac{2^{n+1} (n!)^4}{(2n+1) [(2n)!]^3}.$$

Koeficient d_n s rostoucím n rychle klesá k nule a pro funkce s dostatečně velkým počtem spojitých derivací dávájí Gaussovy vzorce vyšších řádů velmi přesné výsledky.

Gaussovy vzorce se rovněž dají použít ve složených vzorcích tvaru (10.7.1). Rozdělíme interval $\langle a, b \rangle$ na intervaly $\langle a_j, b_j \rangle$, $j = 1, 2, \dots, m$, ($a_1 = a$, $b_m = b$, $b_j = a_{j+1}$) takové, že $b_j - a_j = h$, $j = 1, 2, \dots, m$ [všechny dílčí intervaly mají stejnou délku a $m = (b-a)/h$]. Použijeme-li na každém dílčím intervalu též Gaussův vzorec s $n+1$ uzly, pak podle (13.3.1) dostáváme složený kvadraturní vzorec

$$(13.4.3) \quad \int_a^b f(x) dx \approx \frac{h}{2} \sum_{j=1}^m \sum_{i=0}^n w_i^{(0)} f\left(a_j + \frac{h}{2} + \frac{hx_i^{(0)}}{2}\right),$$

kde $w_i^{(0)}, x_i^{(0)}, i = 0, 1, \dots, n$, jsou opět koeficienty a uzly pro interval $\langle -1, 1 \rangle$.

Nechť $f^{(2n+2)}$ je funkce spojitá na intervalu $\langle a, b \rangle$. Z toho, co jsme řekli o chybě Gaussova vzorce pro interval $\langle -1, 1 \rangle$, lze pomocí transformace proměnných [viz (13.3.1)] snadno stanovit chybu Gaussova vzorce s $n+1$ uzly při výpočtu integrálu přes interval $\langle a_j, a_j+h \rangle$ délky h . Sečtením těchto chyb a stejným postupem jako v odst. 11.5 odtud dostáváme, že chyba aproximace při použití vzorce (13.4.3) je

$$(13.4.4) \quad \frac{b-a}{2} \left(\frac{h}{2}\right)^{2n+2} d_{n+1} f^{(2n+2)}(\xi),$$

kde $\xi \in \langle a, b \rangle$. Pro větší n a dostatečně hladké funkce bude tedy při zmenšování h chyba aproximace složeným vzorcem velmi rychle klesat.

13.5 Příklady.

Vypočítáme prvními dvěma Gaussovými kvadraturními vzorci(13.4.1) hodnotu integrálu

$$\int_1^{1,2} e^x dx \doteq 0,60184$$

(viz příkl. 11.2).

Nejprve musíme uzly a koeficienty transformovat tak, abychom ze (13.4.1) dostali kvadraturní vzorec pro interval $\langle 1; 1, 2 \rangle$. Transformace (13.3.1) má v našem případě tvar

$$w_i = \frac{1}{2}(1,2-1)w_i^{(0)} = 0,1w_i^{(0)},$$

$$x_i = 1,1+0,1x_i^{(0)}, \quad i = 0, 1, \dots, n.$$

Odtud máme pro $n = 0$ vzorec

$$\int_1^{1,2} f(x) dx \approx 0,2f(1,1),$$

což v našem případě jako přibližnou hodnotu hledaného integrálu dává (na 5D) 0,60083 (pro $n = 0$ jde o stejný kvadraturní vzorec, jako je obdélníkové pravidlo).

Pro $n = 1$ dostáváme

$$\int_1^{1,2} f(x) dx \approx 0,1f(1,1-0,1 \cdot 3^{-1/2}) + 0,1f(1,1+0,1 \cdot 3^{-1/2}) \doteq$$

$$\doteq 0,1[f(1,042265) + f(1,157735)].$$

Dosadíme-li sem hodnoty e^x , dostáváme jako přibližnou hodnotu hledaného integrálu 0,60183 (na 5D). To je prakticky stejně dobrý výsledek jako jsme dostali v příkl. 11.2 Simpsonovým pravidlem při použití tří funkčních hodnot.

13.5.1

Výraz (13.4.4) poskytuje určitou kvalitativní informaci o chování chyby aproximace při $h \rightarrow 0$ (pro hladké integrandy). Tento výraz však může sloužit i jako základ pro odhady chyb, založené na odhadech $(2n + 2)$ -hé derivace (a tedy i zde zpravidla značně pesimistické).

Počítejme integrál (viz příkl. 11.6)

$$I = \int_2^3 (x - 1)^{-1} dx \doteq 0,69315$$

složeným vzorcem (13.4.3) založeným na základním vzorci se dvěma uzly podobně jako lichoběžníkové pravidlo). Odhadněme nejprve stejným postupem, jaký jsme použili v příkl. 11.6, velikost dílčích intervalů $b_j - a_j = h$ postačující k tomu abychom na základě (13.4.4) měli zaručeno, že chyba aproximace nepřevyší 10^{-3} .

Podle (13.4.4) k tomu stačí, aby (je $n = 1$)

$$\frac{b - a}{2} \left(\frac{h}{2}\right)^4 d_2 \max_{(2,3)} f^{(4)}(x) \leq 10^{-3},$$

což po dosazení dává

$$\frac{1}{2} \frac{h^4}{16} \frac{1}{135} \cdot 24 \leq 10^{-3},$$

Odtud dostáváme

$$h^4 \leq 18 \cdot 10^{-2}$$

a dále

$$m = \frac{b - a}{h} \geq (18 \cdot 10^{-2})^{-1/4} \doteq 1,54$$

Stačí tedy vzít $m = 2$ (a $h = \frac{1}{2}$). Protože na každém dílčím intervalu máme dva uzly, je celkový potřebný počet uzlů roven čtyřem. Ve srovnání s lichoběžníkovým pravidlem stačí tedy zhruba třetinový počet uzlů. Při odhadu chyby jsme zde však využili existenci čtvrté derivace, kdežto u lichoběžníkového pravidla jsme k odhadu vystačili s druhou derivací.

Provedeme-li výpočet podle (13.4.3) s $m = 2$, $n = 1$, $h = \frac{1}{2}$, dostaneme jako přibližnou hodnotu integrálu I číslo

$$K = 0,25[f(2,25 - 0,25 \cdot 3^{-1/2}) + f(2,25 + 0,25 \cdot 3^{-1/2}) + f(2,75 - 0,25 \cdot 3^{-1/2}) + f(2,75 + 0,25 \cdot 3^{-1/2})] \doteq 0,69308.$$

Skutečná chyba je tedy $7 \cdot 10^{-5}$ a je mnohem menší, než je odhadem podle (13.4.4) zaručeno.

13.6 Použití Gaussových vzorců.

Koeficienty a uzly Gaussových vzorců mají komplikovanější strukturu, než je tomu u jednoduchých vzorců Newtonových-Cotesových. Gaussovy vzorce jsou proto pro výpočty v ruce či na

kapesním kalkulátoru méně oblíbeny. Vybavení knihoven výpočetních středisek však umožňuje při výpočtu na počítači běžně používat Gaussovy vzorce řádů 10 až 20, někdy i více.

Porovnáme-li Gaussovy vzorce s Newtonovými-Cotesovými vzorci či s Rombergovou metodou, zjišťujeme, že pro dostatečně hladké funkce dají při stejné vynaložené práci Gaussovy vzorce nej přesnější výsledek. Pokud dokážeme předem odhadnout, jak vybrat řád (resp. počet uzlů) tak, abychom Gaussovým vzorcem dosáhli požadované přesnosti, je na místě dát tomuto vzorci přednost před jinými metodami. Taková situace se často vyskytuje tam, kde v nějakém cyklu (např. iterační proces, optimalizace) počítáme hodnoty posloupnosti integrálů, lišících se od sebe postupně se měnícími hodnotami některých parametrů

Nejí-li takový odhad prakticky proveditelný, postupuje se zpravidla tak, že se postupně zvyšuje algebraický řád použitého vzorce tak dlouho, až jsou dvě po sobě získané aproximace hledaného integrálu v mezích požadované přesnosti identické (využití konvergence!). Je také možno použít složený vzorec a postupně zjemňovat dělení intervalu integrace na dílčí intervaly. Protože ale funkční hodnoty použité ve vzorci s méně uzly nemohou být u Gaussových kvadraturních vzorců obecně využity ve vzorcích s více uzly, je takový výpočet často pracnější než užití Rombergovy metody. Pro výpočet hodnoty jednoho, samostatně stojícího integrálu z hladké (!) funkce je proto zpravidla účelnější použít Rombergovu metodu.

13.7 Příklady.

13.7.1

Porovnáme výsledky Rombergovy metody a Gaussova kvadraturního vzorce na výpočtu integrálu

$$\int_0^{0,8} f(x) dx,$$

kde

$$f(x) = \frac{\sin x}{x}$$

(srov. příkl. 12.8).

Použijeme základní Gaussův kvadraturní vzorec se třemi uzly. Podle (13.4.1) a (13.3.1), kde položíme $b = 0,8$ a $a = 0$, dostáváme koeficienty $\frac{20}{90}, \frac{32}{90}, \frac{20}{90}$ a uzly (na sedm cifer přesně) $0,09016133; 0,4; 0,7098387$. Vypočteme-li tři funkční hodnoty v těchto uzlech, dostáváme pro uvedenou funkci f jako přibližnou hodnotu integrálu

$$\int_0^{0,8} f(x) dx \approx \frac{20}{90} f(0,09016133) + \frac{32}{90} f(0,4) + \frac{20}{90} f(0,7098387) \doteq 0,772096.$$

To je stejně dobrý výsledek, jako jsme dostali v příkl. 12.8 Rombergovou metodou z devíti funkčních hodnot v devíti ekvidistantních uzlech.

13.7.2

Vypočítejme stejným Gaussovým kvadraturním vzorcem jako v příkl. 13.7.1 přibližnou hodnotu integrálu

$$\int_0^{0,8} x^{1/2} dx$$

(viz též příkl. 12.10).

Integrand nemá v bodě $x = 0$ konečnou derivaci. Zcela stejným postupem jako v příkl. 13.7.1 bychom ze tří funkčních hodnot jako výsledek dostali 0,47883 (na 5D) s chybou $-1,80 \cdot 10^{-3}$. To je lepší výsledek, než jsme dostali v příkl. 12.10 při použití devíti funkčních hodnot.

13.8 Přibližný výpočet integrálů s váhovou funkcí.

Existují a používají se také interpolační kvadraturní vzorce tvaru

$$(13.8.1) \quad \int_a^b \omega(x)f(x)dx \approx \sum_{i=0}^n w_i f(x_i),$$

kde ω je nějaká pevně daná váhová funkce, nezáporná a měřitelná na intervalu $\langle a, b \rangle$ [tento interval bývá v tomto případě mnohdy nekonečný; funkce f musí být ovšem taková, že integrál ve (13.8.1) existuje]. Při zadaných uzlech kvadratury přejde vzorec (10.6.2) pro stanovení koeficientů ve vzorec

$$(13.8.2) \quad w_i = \int_a^b \omega(x)l_i(x)dx$$

a pokud integrály (13.8.2) existují a jsou konečné, dá se kvadraturní vzorec (13.8.1) sestavit obvyklým způsobem. Závislost na váhové funkci je přitom skryta v koeficientech w_i .

Zcela stejným způsobem jako při odvození klasických Gaussových kvadraturních vzorců [kde je $\omega(x) \equiv 1$] se dá dosáhnout toho, že algebraický řád vzorce (13.8.1) bude $2n + 1$. Je k tomu třeba za uzly x_i , $i = 0, 1, \dots, n$, vzít kořeny polynomu $(n + 1)$ -ního stupně z posloupnosti polynomů, které jsou na intervalu $\langle a, b \rangle$ ortogonální s váhovou funkcí ω (viz odst. 5.5 a 5.6). Výsledným vzorcům typu (13.8.1) se pak opět říká *Gaussovy kvadraturní vzorce*. Gaussovy vzorce pro integrály s váhovou funkcí jsou odvozeny např. v případech uvedených v tab. 26. Hodnoty koeficientů a uzlů najdeme opět v tabulkách či knihovných programech (např. [14]).

Tabulka 26

$\langle a, b \rangle$	$\omega(x)$	ortogonální polynomy
$\langle -1, 1 \rangle$	$(1 - x^2)^{-1/2}$	Čebyševovy polynomy
$\langle 0, +\infty \rangle$	e^{-x}	Laguerrovy polynomy
$\langle -\infty, +\infty \rangle$	$\exp(-x^2)$	Hermitovy polynomy

Všimneme si také toho, že vzorce s váhovou funkcí můžeme použít k výpočtu integrálu

$$\int_a^b F(x) dx$$

tak, že položíme

$$F(x) = \omega(x) \left(\frac{F(x)}{\omega(x)} \right)$$

v vzorec (13.8.1) aplikujeme na funkci $f(x) = F(x)/\omega(x)$.

13.9 Úlohy

13.9.1

Volbou $f(x) = 1, x, x^2, \dots$ prověřte, jaký je algebraický řád Gaussových kvadraturních vzorců (13.4.1). Výsledek porovnejte s výrazem pro chybu ve (13.4.1).

[Řád je 1,3 a 5].

13.9.2

Pomocí transformace (13.3.1) odvoďte z (13.4.1) kvadraturní vzorce pro výpočet

$$\int_0^1 f(x) dx.$$

[Návod: Bude $w_i = \frac{1}{2}w_i^{(0)}$, $x_i = \frac{1}{2} + \frac{1}{2}x_i^{(0)}$.]

13.9.3

Podle (13.4.4) vyjádřete chybu kvadraturních vzorců pro výpočet

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

získaných z Gaussových vzorců (13.4.1). Speciálně stanovte hodnoty konstant u derivací v tomto vyjádření pro $a = 0$, $b = 1$.

[Návod: Položte $h = b - a$. Uvažte pak, že původní konstanty ve (13.4.1) jsou právě d_{n+1} , $n = 0, 1, 2$.]

13.9.4

Vypočítejte pomocí Gaussova kvadraturního vzorce se třemi uzly přibližnou hodnotu

$$\int_0^1 \frac{\sin x}{1+x} dx.$$

[0,284249 (na 6D).]

13.9.5

Vypočítejte přibližnou hodnotu

$$\int_0^1 (1+x)^{-1} dx \doteq 0,69315 :$$

a) Pomocí Gaussova kvadraturního vzorce se dvěma uzly; odhadněte chybu aproximace a porovnejte se skutečnou chybou.

[0,69231 (na 5D); odhad chyby podle (13.4.4) a (13.4.2) dává $5,56 \cdot 10^{-3}$, skutečná chyba je $0,84 \cdot 10^{-3}$.]

b) Pomocí Gaussova kvadraturního vzorce se třemi uzly

[0,69312 (na 5D).]

c) Složeným vzorcem (13.4.3), který vychází ze základního vzorce se třemi uzly a kde položíme $h = \frac{1}{2}$

[0,69315 (na 5D).]

13.9.6

Vypočítejte složeným Gaussovým kvadraturním vzorcem, který vychází ze základního vzorce se dvěma uzly, přibližnou hodnotu integrálu

$$\int_0^{0,8} x^{-1} \sin x dx.$$

Položte $h = 0,4$. Počítejte na 6 platných cifer, hodnoty uzlů na 7 cifer přesně.

[0,772095]

13.9.7

Sestavte program, kterým vypočítáte hodnoty integrálů z úlohy 12.11.7 na 5 cifer přesně. Požadované přesnosti dosáhnete tak, že: a) Budete postupně zvyšovat algebraický řád základního vzorce. b) Použijete složený vzorec založený na některém pevně zvoleném základním vzorci a postupně budete zjemňovat dělení intervalu integrace. Získané výsledky porovnejte s přesnými hodnotami a stanovte skutečnou chybu, které jste dosáhli.

14 Několik poznámek k numerické kvadratuře

14.1 Adaptivní programy pro výpočet integrálů.

Některé programy pro přibližný výpočet určitého integrálu používají složené kvadraturní vzorce založené na jednom či dvou základních vzorcích a velikosti dílčích intervalů v (10.7.1) určují automaticky tak, aby získaný výsledek splňoval zadané požadavky na přesnost. V různých částech intervalu $\langle a, b \rangle$ se příp. používá dělení jemnější nebo hrubší podle toho, zda integrand je v dané části $\langle a, b \rangle$ hladký a mění se pomalu či zda je na dané části intervalu přibližný výpočet integrálu obtížný. Programy, které přizpůsobují hustotu uzlů v té či oné části intervalu $\langle a, b \rangle$ charakteru integrandu, se nazývají *adaptivní programy*. Odkazy na některé existující adaptivní programy pro výpočet jednorozměrných integrálů jsou uvedeny v kap. IV.

Uživatel adaptivního programu zadá jako vstupní data interval $\langle a, b \rangle$, podprogram pro výpočet $f(x)$ při $x \in \langle a, b \rangle$ a požadovanou hranici ε pro chybu aproximace. Program se snaží stanovit Q tak, aby platilo

$$\left| \int_a^b f(x) dx - Q \right| \leq \varepsilon$$

(příp. aby bylo splněno nějaké jiné kritérium pro posouzení velikosti chyby, např. relativní). Výstupem z adaptivního programu může být také hlášení, že požadované přesnosti nelze dosáhnout.

Způsoby odhadu chyby aproximace používané programy pro adaptivní kvadraturu jsou založeny jednak na jistých heuristických úvahách (srov. odst. 11.8) o chování vyšších derivací integrované funkce. Je proto vždy možné vymyslet takový integrand f , pro nějž příslušný adaptivní program dá chybný výsledek, tj. nevypočte hledaný integrál s požadovanou přesností a nepodá hlášení o tom, že této přesnosti nebylo dosaženo. Pro dobré dnes používané programy je však třída těchto integrandů velmi malá. Opatrný uživatel může vždy jako vstupní parametr ε zadat větší přesnost, než ve skutečnosti požaduje, a zvýšit tak spolehlivost výpočtu. Doba výpočtu se tím ovšem prodlužuje.

14.2 Použití adaptivních programů.

Značná část výpočetní práce se u adaptivních programů spotřebuje na určení vhodného dělení intervalu $\langle a, b \rangle$ na dílčí intervaly. Odhaduje se, že adaptivní programy používají zhruba dvakrát až třikrát více funkčních hodnot integrandu, než by stačilo k dosažení požadované přesnosti vhodně vybraným Gaussovým kvadraturním vzorcem. Z tohoto důvodu adaptivní programy nejsou vhodné pro výpočty velkých souborů integrálů lišících se od sebe malými změnami některých parametrů (např. při různých iteračních či minimalizačních procesech).

Hlavní oblast použití adaptivních programů je tam, kde počítáme jeden nebo několik integrálů bez nějaké vnitřní souvislosti, příp. tam, kde je uživatel programu připraven zaplatit jistou cenu navíc za to, že se při výpočtu uspoří lidský čas potřebný k sestavení efektivnějšího programu pro ten který konkrétní integrál. Přes určité varování před bezmyšlenkovitým používáním programů pro automatický výpočet integrálu je oblast použití adaptivních automatických programů poměrně široká.

14.2.1 Stabilita kvadraturních algoritmů.

Funkční hodnoty vystupující v kvadraturním vzorci (10.6.1) budou v praxi mnohdy zatíženy chybou měření nebo chybou aproximace. Vyšetříme proto nejdříve šíření chyb funkčních hodnot v procesu numerického výpočtu integrálu, tj. podmíněnost úlohy (viz [21]).

Snadno ukážeme, že výpočet podle kvadraturních vzorců s kladnými koeficienty w_i , $i = 0, 1, \dots, n$, je dobře podmíněná úloha. K důkazu tohoto tvrzení využijeme ještě to, že budeme předpokládat, že vzorec je přesný alespoň pro konstantní funkci. Z tohoto předpokladu ihned plyne, že

$$\sum_{i=0}^n w_i = b - a$$

(ověřte!).

Předpokládejme tedy, že místo přesných hodnot $f(x_i)$, $i = 0, 1, \dots, n$, použijeme čísla $f^*(x_i) = f(x_i) + \varepsilon_i$, $i = 0, 1, \dots, n$. Místo (10.6.1) pak počítáme

$$\sum_{i=0}^n w_i f^*(x_i) = \sum_{i=0}^n w_i f(x_i) + \sum_{i=0}^n w_i \varepsilon_i,$$

a platí tedy

$$(14.3.1) \quad \left| \sum_{i=0}^n w_i f^*(x_i) - \sum_{i=0}^n w_i f(x_i) \right| = \left| \sum_{i=0}^n w_i \varepsilon_i \right| \leq \\ \leq \sum_{i=0}^n w_i |\varepsilon_i| \leq (b - a) \varepsilon, \quad \varepsilon = \max_i \varepsilon_i,$$

kde jsme k odhadu absolutní hodnoty součtu použili trojúhelníkovou nerovnost známou z algebry.

Srovnáme-li tento výsledek s (10.2.1), visíme, že za našich předpokladů je citlivost matematického problému $[I(f)]$ a jeho numerické aproximace $[K(f)]$ na poruchy v datech stejná. Předpoklad $w_i > 0$, $i = 0, 1, \dots$, splňují všechny kvadraturní vzorce popsané v této kapitole s výjimkou Newtonových-Cotesových vzorců vyšších řádů.

Vyšetřujeme-li za předpokladů uvedených na začátku tohoto odstavce šíření relativní chyby, dostáváme podle [21], že číslo podmíněnosti při výpočtu podle vzorce (10.6.1) je

$$(14.3.2) \quad C_2 = (b - 1) \frac{\max_i |f(x_i)|}{|K(f)|}.$$

Zcela analogický vztah platí opět pro integrál $I(f)$. Relativní chyby se tedy mohou zesilovat tehdy, počítáme-li integrál, jehož absolutní hodnota je malá, z velkých funkčních hodnot. V takovém případě totiž také zřejmě bude docházet k odčítání sobě blízkých velkých čísel, které - jak známo [21] - má na šíření relativní chyby a stabilitu algoritmů vůbec velmi nepříznivý vliv.

Pokud jde o zaokrouhlovací chyby při výpočtu součtu

$$\sum_{i=0}^n w_i f(x_i),$$

dá se vcelku snadnou algebraickou manipulací ukázat (viz např. [28]), že na celkové chybě ze zaokrouhlování se podílejí převážnou měrou jen chyby operace sčítání a že tato celková zaokrouhlovací chyba roste lineárně s počtem sčítanců. Takové procesy se považují za numericky stabilní; ostatně je možné použitím tzv. částečné dvojnásobné aritmetiky [28] obecně docílit toho, že vliv zaokrouhlovacích chyb na kvadraturní proces se sníží na zanedbatelnou úroveň

15 Numerické derivování

15.1 Derivování interpolačního polynomu.

O numerickém výpočtu derivace jsme se zmínili na několika místech již dříve. V odstavci 4.8 jsme např. uvedli, jak se pomocí interpolační spline-funkce dají aproximovat derivace nízkých řádů. V celém čl. 15 se omezíme na ty způsoby aproximace derivací dané funkce, které vycházejí z toho, že funkci samu aproximujeme interpolačním polynomem. Přibližné hodnoty derivací funkce pak počítáme jako hodnoty derivací aproximujícího interpolačního polynomu. Stupeň použitého interpolačního polynomu nesmí být přirozeně menší, než je řád počítané derivace (proč?).

Pro účely tohoto článku bude výhodné interpolační polynom zapisovat v Newtonově tvaru. Vycházíme-li při přibližném výpočtu k -té derivace funkce f z $n + 1$ jejích hodnot, $n \geq k$ a aproximujeme-li tedy f Newtonovým interpolačním polynomem (3.23.2) n -tého stupně, dostáváme pro přibližný výpočet derivací vztah (viz úlohu 15.9.1)

$$(15.1.1) \quad f^{(k)}(x) \approx N_n^{(k)}(x) = \\ = f[x_0, x_1, \dots, x_k] \frac{d^k}{dx^k} \{(x - x_0)(x - x_1) \dots (x - x_{k-1})\} + \dots + \\ + f[x_0, x_1, \dots, x_k] \frac{d^k}{dx^k} \{(x - x_0)(x - x_1) \dots (x - x_{n-1})\}.$$

Jsou-li uzly interpolace ekvidistantními a zapíšeme-li interpolační polynom jako Newtonův polynom s dopřednými diferencemi (3.29.3), dostaneme místo vzorce (15.1.1) vztah

$$(15.1.2) \quad f^{(k)}(x) \approx \frac{d^k}{dx^k} N_n^+(t) = h^{-k} \frac{d^k}{dt^k} N_n^+(t) = \\ = h^{-k} \left[\Delta^k f_0 \frac{d^k}{dt^k} \binom{t}{k} + \dots + \Delta^n f_0 \frac{d^k}{dt^k} \binom{t}{n} \right],$$

neboť podle (3.29.1) máme $t = (x - x_0)/h$ a $dt/dx = 1/h$. Vzorce (15.1.1) a (15.1.2) používáme pro x z intervalu ohraničeného nejmenším a největším uzlem interpolace.

15.2 Příklad.

V odstavci 9.2 jsme pro přibližný výpočet derivace funkce f v bodě \hat{x} odvodili vzorec

$$f'(\hat{x}) \approx \frac{f(\hat{x} + h) - f(\hat{x} - h)}{2h}.$$

Ukažme, že je to speciální případ vzorců (15.1.1) a (15.1.2).

Vezměme za uzly interpolace body $x_0 = \hat{x} - h$, $x_1 = \hat{x}$, $x_2 = \hat{x} + h$ a interpolujme funkci f polynomem $N_2^+(t)$. Podle (15.1.2), kde položíme $k = 1$, $n = 2$, dostáváme

$$(15.2.1) \quad f'(x) \approx h^{-1} \left[\Delta f_0 + \Delta^2 f_0 \frac{t + (t-1)}{2} \right]$$

Položíme-li zde nyní $x = x_1 = \hat{x}$, máme podle (3.29.1) $t = 1$ a dostáváme

$$f'(\hat{x}) \approx \left(\Delta f_0 + \frac{1}{2} \Delta^2 f_0 \right) = \frac{f_2 - f_0}{2h} = \frac{f(\hat{x} + h) - f(\hat{x} - h)}{2h}.$$

Užití vztahu (15.1.1) by dalo zcela stejný výsledek (ověřte to).

15.3 Chyba aproximace.

Stejně jako u numerických metod pro výpočet integrálu se dá při odhadu chyby aproximace pro vzorce (15.1.1) a (15.1.2) vyjít ze vzorce pro chybu interpolace (3.10.1), podle kterého za předpokladů věty 3.10 platí

$$(15.3.1) \quad f(x) = N_n(x) + e_n(x) \frac{f^{(n+1)}(\xi)}{(n+1)!},$$

kde $e_n(x) = (x - x_0)(x - x_1) \dots (x - x_n)$. Vzhledem k tomu, že ξ v (15.3.1) je neznámou funkcí proměnné x , jsou odhady chyby založené na derivování vztahu (15.3.1) poměrně komplikované a jejich detailní znalost přináší pro praktické počítání zpravidla malý užitek. Uvedeme proto jen přehled základních výsledků. Další podrobnosti lze nalézt v [2], [26].

Pro případ, kdy (15.1.1) použijeme k přibližnému výpočtu hodnoty první derivace funkce f v některém tabulkovém bodě $x = x_s$, dostaneme derivováním vztahu (15.3.1) snadno jednoduchý výsledek

$$(15.3.2) \quad f'(x_s) = N'_n(x_s) + e'_n(x_s) \frac{f^{(n+1)}(\xi)}{(n+1)!},$$

neboť člen obsahující $e_n(x_s)$ [a derivaci funkce $f^{(n+1)}(\xi)$] se rovná nule. Pro ekvidistantní body $x_i = x_0 + ih$ přejde (15.3.2) v

$$(15.3.3) \quad f'(x_s) = N'_n(x_s) + (-1)^{n-s} h^n s!(n-s)! \frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

Je tedy chyba aproximace první derivace v uzlu interpolace při $h \rightarrow 0$ velikosti $O(h^n)$. Obecně se pro dostatečně hladké funkce dá ukázat [2], že při ekvidistantních uzlech interpolace a aproximaci k -té derivace v některém z uzlů interpolace je chyba aproximace velikosti O^{n+1-k} , kde n je stupeň interpolačního polynomu použitého v (15.1.1) nebo (15.1.2). Pro některé speciální situace přitom může být chyba i velikosti O^{n+2-k} , tedy řádově menší (při $h \rightarrow 0$).

15.4 Některé často užívané vzorce.

V tomto odstavci podáme přehled některých užitečných vzorců pro výpočet první a druhé derivace. Jde tu vesměs o speciální případy vzorce (15.1.2); předpokládáme tedy, že uzly interpolace jsou ekvidistantní s krokem h . Dále předpokládáme, že hodnotu derivace počítáme v některém z uzlů. Abychom docílili jednotného zápisu, označujeme v tomto odstavci uzel x_s , v němž počítáme přibližnou hodnotu derivace, vždy jako x . Ostatní uzly rovněž nečíslujeme, ale vyjadřujeme je pomocí x jako $x + h$, $x - h$ apod. Ve vzorcích uvádíme zároveň (zčásti pouze řádově) chybu aproximace. Příslušný údaj je platný pro funkce, které mají dostatečný počet spojitých derivací. Bod ξ leží vždy mezi nejmenším uzlem použitým v daném vzorci.

Pro výpočet první derivace ($k = 1$) dostáváme vztahy

$$(15.4.1) \quad n = 1 :$$

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{1}{2}hf''(\xi),$$

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \frac{1}{2}hf''(\xi);$$

$$(15.4.2) \quad n = 2 :$$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{1}{6}h^2f'''(\xi),$$

$$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + \frac{1}{3}h^2f'''(\xi),$$

$$f'(x) = \frac{3f(x) - 4f(x-h) + f(x-2h)}{2h} + \frac{1}{3}h^2f'''(\xi).$$

Druhou derivaci ($k = 2$) aproximujeme pomocí vztahů

$$(15.4.3) \quad n = 2 :$$

$$f''(x) = \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} - hf'''(\xi),$$

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{h^2f^{(4)}(\xi)}{12},$$

$$f''(x) = \frac{f(x) - 2f(x-h) + f(x-2h)}{h^2} + hf'''(\xi);$$

$$(15.4.4) \quad n = 3 :$$

$$f''(x) = \frac{12f(x) - 30f(x \pm h) + 24f(x \pm 2h) - 6f(x \pm 3h)}{6h^2} + O(h^2);$$

$$(15.4.5) \quad n = 4 :$$

$$f''(x) = \frac{-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)}{12h^2} + O(h^4).$$

Četné další vzorce lze nalézt např. v [3].

15.5 Příklad.

Vypočítáme podle vzorců (15.4.1) a (15.4.2) přibližnou hodnotu první derivace funkce $f(x) = x^{-1} \sin x$ v bodě $x = 0,5$. Funkční hodnoty vezmeme v tab. 22, odst. 11.4, a použijeme $h = 0,1$. Přesná hodnota derivace je $f'(0,5) \doteq -0,16254$.

Výpočet podle vzorců (15.4.1) dává jako přibližnou hodnotu derivace po řadě

$$f'(0,5) \approx \frac{0,94107 - 0,95885}{0,1} = -0,1778$$

a

$$f'(0,5) \approx \frac{0,95885 - 0,97355}{0,1} = -0,1470.$$

Vzorci (15.4.2) dávají po řadě výsledky

$$f'(0,5) \approx \frac{0,94107 - 0,97355}{0,2} = -0,1624$$

a dále

$$f'(0,5) \approx -0,1629, \quad f'(0,5) \approx -0,1629.$$

Všimneme si zejména přesnosti prvního ze vzorců (15.4.2) ve srovnání se vzorcem (15.4.1).

15.6 Podmíněnost numerického výpočtu derivace.

Chyby v hodnotách funkce f , ať už jde o chyby měření nebo chyby zaokrouhlovací, mají při numerickém derivování mnohem větší význam než při interpolaci nebo numerickém výpočtu integrálu. Vliv nepřesnosti ve vstupních datech se totiž během numerického výpočtu derivace zesiluje. Ukážeme to na výpočtu první derivace podle prvního ze vzorců (15.4.1).

Označme člen vyjadřující chybu metody v prvním vzorcem (15.4.1) jako $r_1 = -\frac{1}{2}hf''(\xi)$. Necht $|f''(\xi)| \leq M_2$; pak je $|r_1| \leq M_2h/2$. Tím jsme odhadli chybu aproximace pomocí odhadu druhé derivace funkce f . Předpokládejme nyní, že místo přesných hodnot $f(x+h)$, $f(x)$, které vystupují v (15.4.1), použijeme hodnoty $f^*(x+h)$, $f^*(x)$, takové, že $|f(x+h) - f^*(x+h)| \leq \varepsilon$, $|f(x) - f^*(x)| \leq \varepsilon$. Tím k chybě aproximace r_1 přidáváme další nepřesnost.

$$r_2 = \frac{f(x+h) - f(x)}{h} - \frac{f^*(x+h) - f^*(x)}{h}.$$

Tuto nepřesnost můžeme odhadnout jako $|r_2| \leq 2\varepsilon/h$. Díky činiteli $1/h$ se při zmenšování h budou nepřesnosti ve vstupních datech $f(x+h)$ a $f(x)$ stále silněji projevat v celkovém výsledku. Pro malá h jde tedy o typickou špatně podmíněnou úlohu.

Zanedbáme-li pro jednoduchost zcela zaokrouhlovací chyby vzniklé při výpočtu podle (15.4.1), dostáváme pro celkovou chybu r vypočítané přibližné hodnoty derivace $f'(x)$ odhad

$$|r| \leq |r_1| + |r_2| \leq \frac{M_2h}{2} + \frac{2\varepsilon}{h} \equiv g(h).$$

Aby byla celková chyba r malá, musí být bezesporu malá chyba aproximace r_1 , a k dosažení požadované přesnosti tedy potřebujeme, aby bylo h dostatečně malé. Jak ale ukazuje průběh

funkce g , poroste při zmenšování h obecně vliv nepřesností ve vstupních datech a pro $h \rightarrow 0$ bude $g(h) \rightarrow \infty$. Vyšetřením průběhu funkce g zjistíme, že je-li např. $\varepsilon = \frac{1}{2} \cdot 10^{-t}$, tj. jsou-li funkční hodnoty udány přesně na t desetinných míst, a je-li M_2 řádově velikosti jedné, není možno spočítat podle (15.4.1) hodnotu první derivace přesněji než na $t/2$ desetinných míst. Pro velké hodnoty M_2 se může stát, že určení derivace s rozumným stupněm přesnosti bude zcela nemožné.

Podobná situace je i u ostatních vzorců a při výpočtu derivací vyšších řádů (viz [2], [26]). Máme-li co činit s empirickými hodnotami malé přesnosti, je tudíž numerický výpočet derivací díky své špatné podmíněnosti hazardní operace. Z důvodů uvedených v tomto odstavci je mnohdy rozumnější použít při výpočtu derivací Richardsonovou extrapolaci tak, jak jsme to dělali v odst. 9.2 a 9.5 při výpočtu první derivace. Takový postup umožní vyhnout se příliš malým krokům h .

15.7 Příklad.

Počítejme podle prvního ze vzorců (15.4.2) přibližné hodnoty první derivace funkce $f(x) = e^x$ v bodě $x = 1$.

Použijeme-li v (15.4.2) hodnoty $f(x+h)$ a $f(x-h)$ zaokrouhlené na 4D a volíme-li postupně $h = 0,8; 0,4; 0,2; \dots$ dostáváme výsledky shrnuté v tab. 27 [přesná hodnota je $f'(1) \doteq 2,7183$]. Vidíme, že zřejmě nemá smysl volit h menší než 0,02.

Tabulka 27

h	0,8	0,4	0,2	0,1	0,05	0,02	0,01	0,005
$\frac{f(x+h)-f(x-h)}{2h}$	3,0176	2,7914	2,7365	2,7230	2,7200	2,7175	2,7200	2,7200

15.8 Aproximace derivací diferencemi.

Položíme-li ve vzorci (15.1.1) $n = k$, tj. aproximujeme-li k -tou derivací pomocí interpolačního polynomu k -tého stupně, dostáváme

$$(15.8.1) \quad f^{(k)}(x) \approx k!f[x_0, x_1, \dots, x_k],$$

neboť $(d^k/dx^k)\{(x-x_0)(x-x_1)\dots(x-x_{k-1})\} = k!$, jak se dá snadno dokázat indukcí. Vzorec (15.1.2) přejde v tomto případě na

$$(15.8.2) \quad f^{(k)}(x) \approx \frac{\Delta^k f_0}{h^k}, \quad x \in \langle x_0, x_k \rangle.$$

Se vztahy (15.8.1) a (15.8.2) jsme se již vlastně setkali v odst. 3.21 a 3.27 [viz (3.21.2), (3.27.2)]. Speciálními případy vzorců (15.8.1) a (15.8.2) jsou vzorce (15.4.1) a (15.4.3). Podobně lze použít i zpětné diference.

Chyba aproximace vzorce (15.8.2) je obecně velikosti (viz odst. 15.3) $O(h)$, ve speciálních případech $O(h^2)$. Tyto speciální případy vlastně odpovídají aproximování derivací centrálními diferencemi $\delta^k f_i$, o kterých jsme se zmínili v odst. 3.31.

Aproximace derivací diferencemi tvoří základ metody sítí pro řešení okrajových úloh pro diferenciální rovnice. Dále lze aproximaci derivací diferencemi použít k přibližnému určování chyb numerických metod. V mnoha výrazech pro chybu metody vystupují totiž (jak jsme viděli v kap. I i v kap. II) derivace vyšších řádů, které se většinou špatně odhadují. Snazší postup je ten, že příslušnou derivaci ve výrazu pro chybu aproximujeme podle (15.8.1) nebo (15.8.2) pomocí diference stejného řádu. Výsledný odhad je sice velmi hrubý, zejména pro větší k , jsou-li však derivace řádu vyššího než k , a tedy i diference vyšších řádů, rozumné, může dát popsany postup dobrou představu o velikosti chyby metody.

15.9 Úlohy

15.9.1

Ověřte vztahy (15.1.1) a (15.1.2).

[Návod: Uvažte, že k -tá derivace polynomu M -tého stupně je pro $k > M$ nulová.]

15.9.2

Vypočítejte přibližnou hodnotu prvních čtyř derivací funkce f zadané tabulkou v příkl. 3.28 v bodě $x = 0,8$. Použijte vzorec (15.1.2) s $n = 4$. Výsledky porovnejte se skutečnými hodnotami pro $f(x) = \sin x$.

[Návod: Nejprve spočítejte analyticky derivace polynomů $\binom{t}{k}$ v (15.1.2) pro $k = 1, 2, 3, 4$.

Výsledky (na 5D): 0,69682; -0,71728; -0,69450; 0,69000. Skutečné hodnoty (na 5D): 0,69671; -0,71736; -0,69471; 0,69036.]

15.9.3

Vypočítejte přibližnou hodnotu první derivace funkce, jejíž hodnoty jsou dány v tab. 15 (úloha 6.10.5) v bodě $x = 0,3$.

a) Užijte vzorce (15.4.1) s $h = 0,1$. [$f'(0,3) \approx -2$, $f'(0,3) \approx -4$.]

b) Derivaci aproximujte derivací polynomu prvního stupně, který byl metodou nejmenších čtverců sestaven v úloze 6.10.5. Co se dá říci o přesnosti získaných výsledků?

[$f'(0,3) \approx -2$. Výpočet podle (15.4.1) dá pro data zatížená náhodnými chybami měření velikosti ε (zřejmě $\geq 0,05$) sotva výsledky s chybou menší než 20ε viz odst. 15.6.]

15.9.4

Vypočítejte přibližnou hodnotu první derivace funkce $f(x) = \ln x$ v bodě $x = 3$ pomocí vzorců (15.4.2) s $h = 0,1$. Výsledky porovnejte s výsledkem příkl. 9.5. Použijte funkční hodnoty přesné na 6D.

Pokuste se volbou menšího h dosáhnout stejně přesné aproximace, jako se dosáhlo extrapolací v příkl. 9.5 (stále používáme funkční hodnoty přesné na 6D). Je to proveditelné?

[Výsledky: 0,333455; 0,333105; 0,333055. Maximální dosažitelná je cca na 4D.]

15.9.5

Proveďte pro první ze vzorců (15.4.2) podobnou analýzu celkové chyby, jako jsme provedli v odst. 15.6 pro vzorec (15.4.1). Vyšetřete podrobně průběh příslušné funkce g a ze získaného výsledku usudte na maximální dosažitelnou přesnost v úloze 15.9.4.

[Pro celkovou chybu r vyjde $|r| \leq M_3 h^2/6 + \varepsilon/h \equiv g(h)$. Funkce g nabývá minima pro $h = h_{\text{opt}} = (2\varepsilon/M_3)^{1/3}$ a je $g(h_{\text{opt}}) = \frac{1}{2}M_3^{1/3}(3\varepsilon)^{2/3}$.]

15.9.6

Odvoďte pomocí věty 2.4 o Taylorově polynomu tvar výrazu pro chybu aproximace v prostředním ze vzorců (15.4.3). Za jakých předpokladů má chyba aproximace uvedený tvar?

[Návod: Vyjádřete podle věty 2.4 $f(x+h)$ a $f(x-h)$ polynomem čtvrtého stupně v h .]

15.9.7

Vypočítejte přibližnou hodnotu druhé derivace funkce f zadané tabulkou v příkl. 3.28 v bodě $x = 0,8$. Použijte vzorec (15.4.3). Výsledky porovnejte s výsledky úlohy 15.9.2.

[Na 4D: $-0,7827$; $-0,7167$; $-0,6438$.]