

Technická univerzita v Liberci
**FAKULTA PŘÍRODOVĚDNĚ-HUMANITNÍ
A PEDAGOGICKÁ**

Katedra: KMD
Studijní program: 2. stupeň
Kombinace: matematika – informatika

Numerická kvadratura a geometrická aplikace určitého
integrálů s využitím matematického software

Numerical Quadrature and Geometrical Applications of Definite
Integral Using Mathematical Software

Diplomová práce: 09–FP–KMD–004

Autor:
Martin Gottfried

Podpis:

Adresa:
Svébořická 146
471 24, Mimoň

Vedoucí práce: Gerta Plačková, prom. mat.

Počet:

Stran	Slov	Obrázků	Tabulek	Pramenů	Příloh
78	7827	12	12	10	6

V Liberci dne 20.5.2009

zadání

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

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

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci dne 20.5.2009

Martin Gottfried

Poděkování

Rád bych touto cestou poděkoval vedoucí mé diplomové práce paní Gertě Plačkové, prom. mat. z Katedry matematiky a didaktiky matematiky za odborný dohled a cenné rady pro vypracování diplomové práce.

Dále děkuji své rodině a blízkým za veškerou podporu, bez kterých bych svou diplomovou práci nikdy nedokončil.

Numerická kvadratura a geometrická aplikace určitého integrálu s využitím matematického software

GOTTFRIED Martin DP - 2009 Vedoucí DP: Gerta Plačková, prom. mat.

Abstrakt

V diplomové práci jsou uvedeny a odvozeny základní numerické metody pro výpočet určitého integrálu. Praktické využití těchto metod je demonstrováno na ukázkových příkladech. Dále je diplomová práce zaměřena na využití matematického softwaru MATLAB. Je vysvětleno základní užívání programu a výpočet určitých integrálů v tomto prostředí. Uvedený matematický software je použit na vytvoření vlastních modulů. Všechny uvedené metody jsou aplikovány na dalších příkladech.

Klíčová slova: určitý integrál, obdélníková metoda, lichoběžníková metoda, Simpsonova metoda, Newton-Cotesovy kvadrurní vzorce, Gaussův kvadrurní vzorec, matematický software, MATLAB.

Abstract

This diploma thesis introduces and infers basic numerical methods for the definite integral calculus. Practical applications of these methods are demonstrated on sample exercises. This diploma thesis also focuses on the use of mathematical software MATLAB. Basic use of this software and calculation of definite integral in the software interface are explained. The mathematical software is used for creating customized modules. All methods mentioned are applied on more examples.

Key words: definite integral, rectangle method, trapezoid method, Simpson's method, Newton–Cotes quadrature formulas, Gaussian quadrature formula, mathematical software, MATLAB.

Das Resümee

In der Diplomarbeit sind elementaren numerischen Verfahren für die Rechnung des bestimmten Integral enthalten und abgeleitet. Die praktische Nutzung dieser Methoden ist an Vorzeigebeispiele präsentieren. Die Diplomarbeit orientiert sich an die Verwendung der Software Matlab auf. Die elementare Anwendung des Programms und die Rechnung des bestimmten Integrals in dieser Umgebung erklärt sind. Die genannte Software ist verwendet für Schaffung von Eigenmodulen. Alle genannten Methoden sind an weiteren Beispielen appliziert.

Schlüsselwörter: der bestimmte Integral, die Rechteckmethode, die Trapezmethode, die Simpsonmethode, die Newton-Cotes Quadraturformel, die Gauss Quadraturformel, die mathematische Software, Matlab.

Obsah

1 Úvod.....	8
2 Numerická kvadratura.....	9
2.1 Riemannův určitý integrál.....	9
2.2 Jednoduché metody pro přibližný výpočet určitého integrálu..	12
2.2.1 Obdélníková metoda.....	12
2.2.2 Lichoběžníková metoda.....	13
2.2.3 Simpsonova metoda.....	14
2.3 Složené metody pro přibližný výpočet určitého integrálu.....	16
2.3.1 Složená obdélníková metoda.....	16
2.3.2 Složená lichoběžníková metoda.....	19
2.3.3 Složená Simpsonova metoda.....	22
2.4 Metoda polovičního kroku.....	26
2.5 Obecný problém numerické kvadratury.....	28
2.6 Newton-Cotesovy kvadraturní vzorce.....	29
2.6.1 Interpolace.....	29
2.6.2 Ekvidistantní uzly.....	31
2.6.3 Obecná formule pro ekvidistantní uzly.....	32
2.6.4 Newton-Cotesovy metody uzavřeného typu.....	33
2.6.5 Newton-Cotesovy metody otevřeného typu.....	36
2.7 Gaussova kvadratura.....	39
2.7.1 Hermitova interpolace.....	39
2.7.2 Legendrovy polynomy.....	42
2.7.3 Odvození Gaussova kvadraturního vzorce.....	43
3 MATLAB.....	47
3.1 Úvod.....	47
3.2 Pracovní prostředí.....	47
3.3 Vytvoření a zápis proměnných.....	48
3.4 Matice a vektory	50
3.4.1 Plnění matic a vektorů.....	50
3.4.2 Speciální znaky.....	51
3.5 Funkce.....	51
3.6 Funkce feval.....	53
3.7 Funkce pro numerickou kvadraturu.....	54
3.8 Gaussova kvadratura v MATLABu.....	56
3.9 Porovnání metod na příkladech.....	59
3.10 Další řešené příklady.....	64
4 Závěr.....	68
5 Použitá literatura.....	69
6 Seznam příloh.....	70

1 Úvod

V matematice se velmi často pracuje s integrálním počtem. Ať už se jedná o zjišťování plochy pod křivkou nebo objemu těles, které vzniknou rotací např. kolem některé z os. Z fyzikálních aplikací můžeme jmenovat určení těžiště tělesa či momentu strvačnosti. Analytické způsoby výpočtů jsou však v mnoha případech zdoluhavé, pracné a dokonce v některých případech i nemožné. V těchto případech nastupují numerické metody řešení, kterými se zabývá má diplomová práce.

Cílem diplomové práce je seznámit se s různými numerickými metodami výpočtu určitého integrálu a tyto metody aplikovat na praktických případech. K řešení příkladů je využít matematický software MATLAB, který umožňuje jak numerickou integraci pomocí vestavěných funkcí, tak rozšíření o vlastní funkce. Dalším cílem mé diplomové práce je tedy vytvořit vlastní funkci v programu MATLAB, která bude vracet co nejpřesnější hodnotu určitého integrálu. Všechny výsledné hodnoty budou mezi sebou porovnány.

V diplomové práci nejsou uvedeny všechny způsoby numerické integrace. Zaměřil jsem se hlavně na metody obecně známé. Čtenář se však seznámí i s metodami složitějšími, mezi které bych zařadil Gaussovu kvadraturu. U každé metody je uvedeno teoretické odvození vzorce, který je následně využíván k výpočtu ukázkového příkladu.

2 Numerická kvadratura

Integrovaní je běžným úkolem aplikované matematiky. Řešením určitého integrálu je nějaké konkrétní číslo. Avšak dosáhnout tohoto řešení není mnohdy snadné, protože velké množství integrálů nelze vyjádřit pomocí elementárních funkcí, např. integrace transcendentních funkcí¹. V takovýchto případech využíváme ke zjištění hodnoty integrálů přibližných metod, mezi které řadíme i numerickou integraci.

Při numerickém integrování se snažíme integrál nahradit jiným druhem výpočtu, avšak musíme zajistit, aby se získané řešení co nejméně lišilo od hodnoty původního integrálu.

Tyto numerické metody, které využíváme ke zjištění hodnoty určitého integrálu, jsou velmi vhodné pro použití ve výpočetní technice, protože jejich algoritmizace je relativně lehká.

2.1 Riemannův určitý integrál

Tento integrál je považován za jeden z nejstarších integrálů. Již staří Řekové znali jeho podstatu a dokázali díky němu počítat obsahy a objemy některých geometrických útvarů. Nese jméno po německém matematikovi Bernhardovi Riemannovi (1826-1866), který vyslovil jeho přesnou definici. Tato definice však umožňuje použití integrálu pouze na reálné ose.

Definice vychází z myšlenky spočtení obsahu plochy pod grafem funkce. Při praktickém zjišťování obsahu bychom plochu vyplnili geometrickými útvary, u kterých se obsah snadno počítá. Tyto útvary vkládáme tak, aby se nepřekrývaly a nepřesáhly hranici plochy. Výsledná hodnota je určitě menší než požadovaný obsah. Toto číslo nazveme *dolní odhad*. Obdobným způsobem, avšak pokrytím celé měřené plochy, dostáváme tzv. *horní odhad*. Hodnota, kterou hledáme, leží

¹ Transcendentní funkce lze rozdělit na nižší, do této skupiny řadíme např. funkce exponenciální, logaritmické, goniometrické, a vyšší, které nelze vyjádřit pomocí elementárních funkcí v konečném tvaru.

právě mezi těmito odhady. Vykládáním plochy stále menšími útvary budeme výsledek zpřesňovat.

Definice 1.: (Dělení intervalu) Necht' je $\langle a, b \rangle$ uzavřený interval na \mathbb{R} . Dělením intervalu $\langle a, b \rangle$ rozumíme konečnou posloupnost

$D = \{x_0, x_1, \dots, x_n\}$ bodů z intervalu $\langle a, b \rangle$ takových, že $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$.

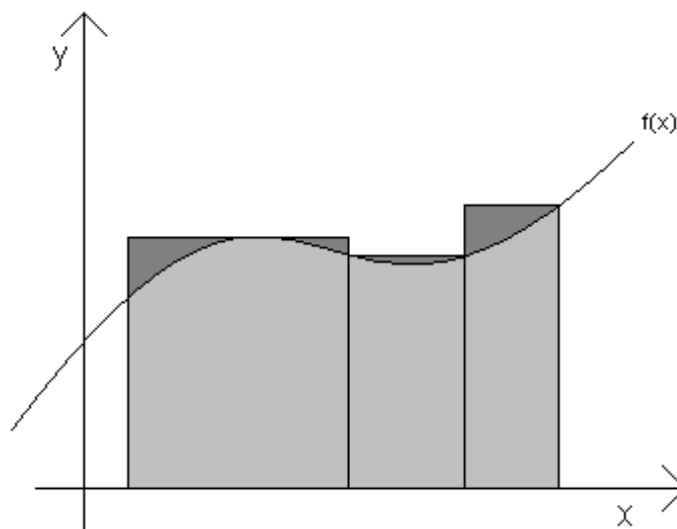
Čísla x_i nazýváme dělicími body.

Definice 2.: (Horní součet) Necht' f je omezená funkce definovaná na intervalu $\langle a, b \rangle$. Je-li dáno dělení D intervalu $\langle a, b \rangle$, pro $k = 1, 2, \dots, n$ označíme

$$M_k = \sup \{ f(t) ; x_{k-1} \leq x_k \} ,$$

pak definujeme horní součet příslušný dělení D vztahem

$$S(f, D) = \sum_{k=1}^n M_k (x_k - x_{k-1}) .$$



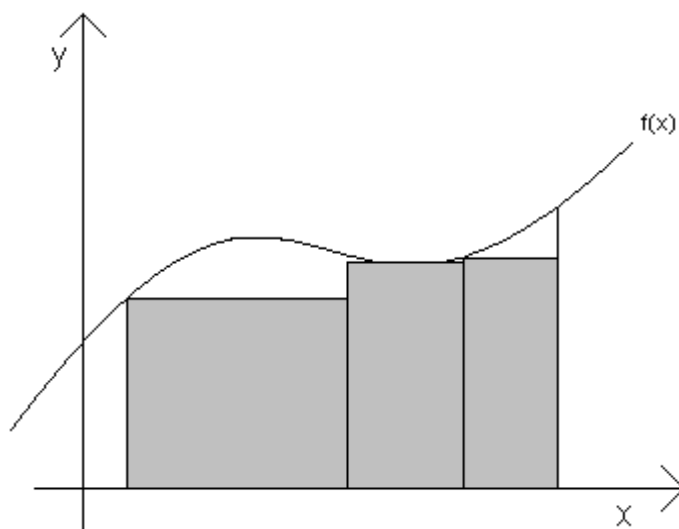
Obr. 1: Horní součet

Definice 3.: (Dolní součet) Necht' f je omezená funkce definovaná na intervalu $\langle a, b \rangle$. Je-li dáno dělení D intervalu $\langle a, b \rangle$, pro $k = 1, 2, \dots, n$ označíme

$$m_k = \inf \{ f(t); x_{k-1} \leq t \leq x_k \} .$$

Poté definujeme dolní součet příslušný dělení D vzorcem

$$s(f, D) = \sum_{k=1}^n m_k (x_k - x_{k-1}) .$$



Obr. 2: Dolní součet

Definice 4.: (Riemannův integrál) Necht' f je omezená funkce definovaná na uzavřenem intervalu $\langle a, b \rangle$. Řekneme, že f je integrovatelná na $\langle a, b \rangle$ v Riemannově smyslu, jestliže infimum horních součtů přes všechna dělení intervalu $\langle a, b \rangle$ je rovno supremu dolních součtů přes všechna dělení intervalu $\langle a, b \rangle$.

Pak definujeme Riemannův určitý integrál funkce f od a do b jako

$$\int_a^b f(x) dx = \inf \{ S(f, D) \} = \sup \{ s(f, D) \} .$$

2.2 Jednoduché metody pro přibližný výpočet určitého integrálu

Tyto základní metody pro přibližný výpočet určitého integrálu vycházejí z aproximace konstantou, lineární funkcí nebo aproximace kvadratickou funkcí. Necht' funkce f je integrovatelná na $\langle a, b \rangle$.

2.2.1 Obdélníková metoda

Tato metoda vychází z aproximace konstantou, tedy

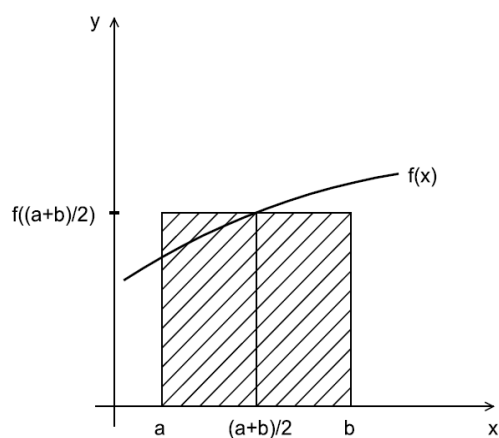
$$f(x) \approx q_0 \text{ pro } x \in \langle a, b \rangle .$$

Jestliže $q_0 = f\left(\frac{a+b}{2}\right)$, pak

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

Geometrický význam této aproximace je nahrazení plochy pod funkcí $f(x)$

plochou obdélníka, který má výšku $f\left(\frac{a+b}{2}\right)$.



Obr. 3: Obdélníková metoda

Příklad 2.2.1

Zadání: Určete hodnotu integrálu $\int_0^1 x^2 \cos(x^2) dx$ pomocí jednoduché obdélníkové metody.

Řešení: Dosadíme do vzorce (viz výše) a dostáváme

$$\int_0^1 x^2 \cos(x^2) dx = (1-0) f\left(\frac{1+0}{2}\right) = \left(\frac{1}{2}\right)^2 \cos\left(\frac{1}{4}\right) = 0,242228105 \text{ .}$$

2.2.2 Lichoběžníková metoda

V případě této metody je funkce $f(x)$ na intervalu $\langle a, b \rangle$ nahrazena lineární funkcí,

$$f(x) \approx q_1 x + q_0 \text{ ,}$$

koeficienty q_1, q_2 jednoduše vyjádříme z podmínek

$$aq_1 + q_0 = f(a) \text{ , } bq_1 + q_0 = f(b) \text{ .}$$

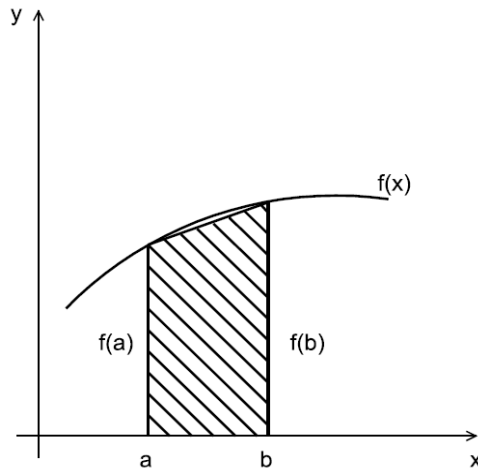
Řešením této soustavy dostáváme

$$q_1 = \frac{f(b) - f(a)}{b - a} \text{ a } q_0 = \frac{bf(a) - af(b)}{b - a} \text{ .}$$

Integrál tedy aproximujeme výrazem

$$\int_a^b f(x) dx \approx \int_a^b (q_1 x + q_0) dx = (b - a) \frac{f(a) + f(b)}{2} \text{ .}$$

Geometricky vyjadřuje aproximace nahrazení plochy pod funkcí $f(x)$ lichoběžníkem o výšce $b - a$ a základnách $f(a), f(b)$.



Obr. 4: Lichoběžníková metoda

Příklad 2.2.2

Zadání: Určete hodnotu integrálu $\int_0^1 x^2 \cos(x^2) dx$ pomocí jednoduché lichoběžníkové metody.

Řešení: Dosazením do vzorce pro jednoduchou lichoběžníkovou metodu dostáváme

$$\int_0^1 x^2 \cos(x^2) dx = (1-0) \frac{f(0)+f(1)}{2} = \frac{0,540302306}{2} = 0,270151153 \text{ .}$$

2.2.3 Simpsonova metoda

Při výpočtu určitého integrálu pomocí jednoduché Simpsonovy metody je funkce $f(x)$ aproximována kvadratickou funkcí

$$p(x) = q_2 x^2 + q_1 x + q_0 \text{ .}$$

Interval $\langle a, b \rangle$ je rozdělen uzly

$$x_0 = a, \quad x_1 = \frac{a+b}{2}, \quad x_2 = b$$

na dva podintervaly o velikosti $h = \frac{b-a}{2}$.

Protože je aproximující funkce polynomem druhého stupně, tak koeficienty q_0, q_1, q_2 jsou určeny soustavou tří rovnic²

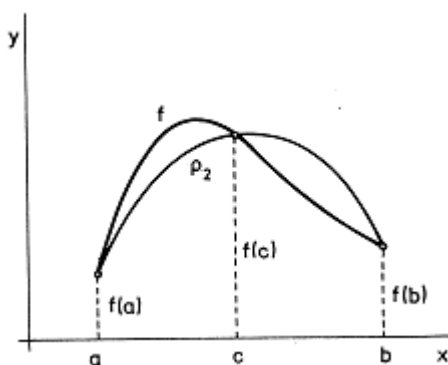
$$x_0^2 q_2 + x_0 q_1 + q_0 = f(x_0),$$

$$x_1^2 q_2 + x_1 q_1 + q_0 = f(x_1),$$

$$x_2^2 q_2 + x_2 q_1 + q_0 = f(x_2).$$

Po vyřešení této soustavy rovnic a po dosazení již můžeme aproximovat a integrál funkce $f(x)$ vyjadřuje vzorec

$$\int_a^b f(x) dx \approx \int_a^b p(x) dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)].$$



Obr. 5: Simpsonova metoda

² Obecně: Funkce $p(x) = q_n x^n + q_{n-1} x^{n-1} + \dots + q_1 x + q_0$ obsahuje $n+1$ koeficientů $q_n, q_{n-1}, \dots, q_1, q_0$. Tyto koeficienty jsou určeny soustavou rovnic $p(x_k) = f(x_k)$, $k = 0, 1, \dots, n$. V našem případě je $n=2$.

Příklad 2.2.3

Zadání: Určete hodnotu integrálu $\int_0^1 x^2 \cos(x^2) dx$ pomocí jednoduché

Simpsonovy metody.

Řešení: Využijeme odvozený vzorec a dostáváme

$$\int_0^1 x^2 \cos(x^2) dx = \frac{(1-0)}{6} [f(0) + 4f(\frac{1}{2}) + f(1)] = \frac{1}{6} (1,509214728) = 0,251535788.$$

2.3 Složené metody pro přibližný výpočet určitého integrálu

2.3.1 Složená obdélníková metoda

Při užití složené obdélníkové metody je nutné rozdělit interval $\langle a, b \rangle$ body, resp. uzly x_i , kde $a = x_0 < x_1 < x_2 < \dots < x_n = b$. Necht' uzly daný interval rozdělí na n podintervalů, které mají stejnou velikost. Tuto velikost lze vyjádřit vztahem

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

Hodnoty funkce $f(x)$ v k -tém podintervalu nahradíme hodnotami $f(x_{k-1})$.

Integrál funkce $f(x)$ na intervalu $\langle a, b \rangle$ lze zapsat jako

$$\begin{aligned} \int_a^b f(x) dx &= \int_a^{x_1} f(x) dx + \int_{x_1}^{x_2} f(x) dx + \dots + \int_{x_{n-2}}^{x_{n-1}} f(x) dx + \int_{x_{n-1}}^b f(x) dx \approx \\ &\approx hf(x_0) + hf(x_1) + \dots + hf(x_{n-2}) + hf(x_{n-1}). \end{aligned}$$

Tento integrál zapisujeme následujícím způsobem

$$I = \int_a^b f(x) dx \approx \frac{b-a}{n} \sum_{i=1}^n f(x_{i-1}) = I_n .$$

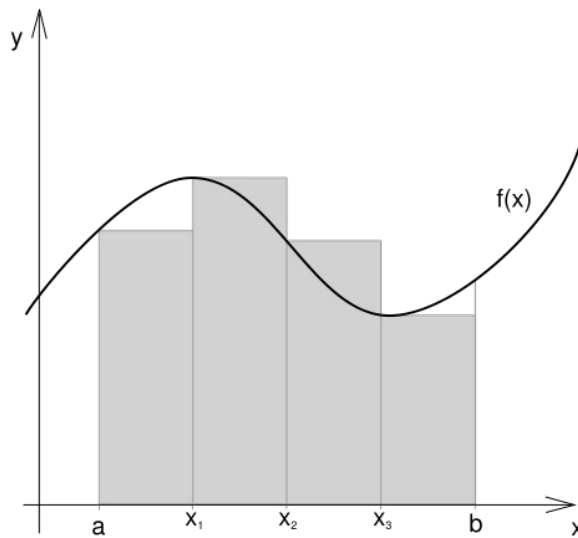
V tomto zápisu I značí hodnotu integrálu a I_n přibližnou hodnotu při rozdělení intervalu $\langle a, b \rangle$ na n podintervalů.

Chybu výpočtu vyjadřuje vzorec

$$R_n = |I - I_n| \leq \frac{D(b-a)^2}{n} \leq \varepsilon ,$$

kde $D = \max |f'(x)|$, $x \in \langle a, b \rangle$ a ε je povolená chyba, na jejímž základě se interval $\langle a, b \rangle$ rozděluje na n podintervalů, tedy

$$n \geq \frac{D(b-a)^2}{\varepsilon} .$$



Obr. 6: Složená obdélníková metoda

Příklad 2.3.1

Zadání: Určete hodnotu integrálu $\int_0^1 x^2 \cos(x^2) dx$ pomocí složené obdélníkové metody.

Řešení: Budeme se držet vzorce

$$\int_a^b f(x) dx \approx \frac{b-a}{n} \sum_{i=1}^n f(x_{i-1}) .$$

Rozdělme interval $\langle a, b \rangle = \langle 0, 1 \rangle$ na $n=10$ podintervalů, které mají velikost

$$h = \frac{b-a}{n} = \frac{1}{10} , \text{ a pro které platí}$$

$$0 = x_0 < x_1 < \dots < x_{10} = 1 .$$

Po dosazení zjišťujeme, že velikost podintervalů je $0,1$. Rozepsáním vzorce dostáváme výraz

$$\int_0^1 x^2 \cos(x^2) \approx \frac{1}{10} (f(0) + f(0,1) + f(0,2) + f(0,3) + \dots + f(0,8) + f(0,9)) .$$

Pro jednotlivá x_{i-1} sestavíme následující tabulku.

x_{i-1}	$f(x_{i-1})$
0,0	0,0000000
0,1	0,0099995
0,2	0,0399680
0,3	0,0896357
0,4	0,1579564
0,5	0,2422281
0,6	0,3369229
0,7	0,4323431
0,8	0,5133413
0,9	0,5584937

Tabulka 1

Sečtením funkčních hodnot $f(x_{i-1})$ dostáváme hodnotu 2,3808887 .
Pro konečný výsledek je ještě potřeba toto číslo vynásobit 0,1 .

Závěr: Přibližná hodnota integrálu vypočtená obdélníkovou metodou je
0,238088869 .

2.3.2 Složená lichoběžníková metoda

Lichoběžníková metoda používá stejné dělení intervalu $\langle a, b \rangle$, tedy dělení uzly x_i , kde $a = x_0 < x_1 < x_2 < \dots < x_n = b$. Těchto n podintervalů má opět stejnou velikost, kterou lze vyjádřit vztahem

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

Hodnotu funkce $f(x)$ pro všechna x v k -tém podintervalu tentokrát nahradíme hodnotou $\frac{f(x_{k-1})+f(x_k)}{2}$. Integrál funkce $f(x)$ na intervalu $\langle a, b \rangle$ má poté tvar

$$\int_a^b f(x) dx = \int_a^{x_1} f(x) dx + \int_{x_1}^{x_2} f(x) dx + \dots + \int_{x_{n-2}}^{x_{n-1}} f(x) dx + \int_{x_{n-1}}^b f(x) dx \approx$$

$$\approx h \frac{f(x_0) + f(x_1)}{2} + h \frac{f(x_1) + f(x_2)}{2} + \dots + h \frac{f(x_{n-2}) + f(x_{n-1})}{2} + h \frac{f(x_{n-1}) + f(x_n)}{2}$$

Úpravami dosáhneme tvaru

$$I = \int_a^b f(x) dx \approx \frac{b-a}{2n} (f(a) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(b)) = I_n ,$$

kde I označuje skutečnou hodnotu integrálu a I_n hodnotu přibližnou při rozdělení intervalu $\langle a, b \rangle$ na n intervalů.

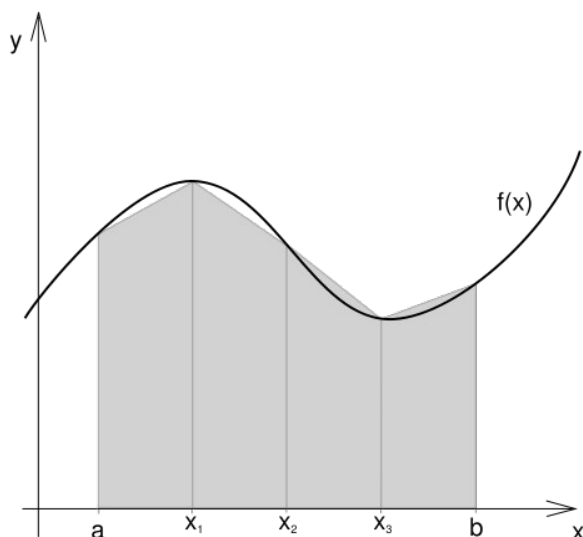
Chyba metody je dána vztahem

$$R_n = |I - I_n| < \frac{D(b-a)^3}{12n^2} < \varepsilon ,$$

kde je v tomto případě $D = \max |f''(x)|$, $x \in \langle a, b \rangle$ a ε je opět povolená chyba, pomocí které určíme rozdělení intervalu $\langle a, b \rangle$ na n podintervalů.

Platí tedy

$$n > \sqrt{\frac{D(b-a)^3}{12\varepsilon}} .$$



Obr. 7: Složená lichoběžníková metoda

Příklad 2.3.2

Zadání: Určete hodnotu integrálu $\int_0^1 x^2 \cos(x^2) dx$ pomocí složené lichoběžníkové metody.

Řešení: V tomto případě se budeme držet obecného vzorce

$$\int_a^b f(x) dx \approx \frac{b-a}{2n} (f(a) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(b)) .$$

Interval $\langle a, b \rangle = \langle 0, 1 \rangle$ rozdělíme uzly x_i , kde $0 = x_0 < x_1 < \dots < x_n = 1$,

$n = 10$. Tyto vzniklé podintervaly mají velikost $h = \frac{b-a}{n} = \frac{1}{10}$.

Dosadíme do vzorce a dostáváme

$$\int_0^1 x^2 \cos(x^2) \approx \frac{1}{20} (f(0) + 2f(\frac{1}{10}) + 2f(\frac{2}{10}) + \dots + 2f(\frac{9}{10}) + f(1)).$$

Pro přehlednost vytvoříme tabulku funkčních hodnot $f(x_i)$.

x_i	$f(x_i)$
0,0	0,0000000
0,1	0,0099995
0,2	0,0399680
0,3	0,0896357
0,4	0,1579564
0,5	0,2422281
0,6	0,3369229
0,7	0,4323431
0,8	0,5133413
0,9	0,5584937
1,0	0,5403023

Tabulka 2

Zpracováním těchto funkčních hodnot do vzorce dostáváme číslo 5,3020797 ,

které pro konečný výsledek musíme vynásobit $\frac{b-a}{2n} = \frac{1}{20}$.

Závěr: Přibližná hodnota integrálu vypočtená lichoběžníkovou metodou je 0,265103985 .

2.3.3 Složená Simpsonova metoda

Při volbě Simpsonovy metody opět interval $\langle a, b \rangle$ rozdělíme uzly x_i , kde $a = x_1 < x_2 < \dots < x_{2n-1} < x_{2n} = b$. Tímto dělením vznikne $2n$ stejně velkých podintervalů o velikosti

$$h = \frac{b-a}{2n} .$$

Hodnoty funkce $f(x)$ v podintervalu $\langle x_{2k-1}, x_{2k+1} \rangle$ jsou nahrazeny parabolou, která prochází body $f(x_{2k-1})$, $f(x_{2k})$ a $f(x_{2k+1})$. Tuto podmínku lze vyjádřit vztahy

$$f(x_{2k-1}) = Ah^2 - Bh + C,$$

$$f(x_{2k}) = C,$$

$$f(x_{2k+1}) = Ah^2 + Bh + C, \quad \text{kde } A, B, C \text{ jsou parametry.}$$

Integrál funkce $f(x)$ na intervalu $\langle x_{2k-1}, x_{2k+1} \rangle$ pak lze zapsat jako

$$\int_{x_{2k-1}}^{x_{2k+1}} f(x) dx \approx \int_{-h}^h (Ax^2 + Bx + C) dx = \frac{h}{3} (2Ah^2 + 6C).$$

Po dosazení do předchozích vztahů dostaneme výraz

$$\frac{h}{3} (2Ah^2 + 6C) = \frac{h}{3} (f(x_{2k-1}) + f(x_{2k+1}) + 4f(x_{2k})).$$

Po změně indexů můžeme výraz přepsat na již konečný tvar

$$\frac{h}{3} (2Ah^2 + 6C) = \frac{h}{3} (f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k}))$$

a integrál funkce $f(x)$ na intervalu $\langle a, b \rangle$ lze již snadno aproximovat

$$\begin{aligned} \int_a^b f(x) dx &\approx \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] + \frac{h}{3} [f(x_2) + 4f(x_3) + f(x_4)] + \\ &+ \dots + \frac{h}{3} [f(x_{2n-2}) + 4f(x_{2n-1}) + f(x_{2n})] = \\ &= \frac{b-a}{6n} \{ f(x_0) + 4[f(x_1) + f(x_3) + \dots + f(x_{2n-1})] + \\ &+ 2[f(x_2) + f(x_4) + \dots + f(x_{2n-2})] + f(x_{2n}) \}. \end{aligned}$$

Chybu Simpsonovy metody vyjadřuje vztah

$$R_{2n} < \frac{D(b-a)^5}{180(2n)^4} < \varepsilon \quad ,$$

kde v tomto způsobu integrace je $D = \max |f^{(4)}(x)|$, $x \in \langle a, b \rangle$ a ε značí požadovanou přesnost (povolenou chybu), podle které se určí počet podintervalů n , na které je interval $\langle a, b \rangle$ rozdělen, a platí

$$2n > \sqrt[4]{\frac{D(b-a)^5}{180\varepsilon}} \quad .$$

Příklad 2.3.3

Zadání: Určete hodnotu integrálu $\int_0^1 x^2 \cos(x^2) dx$ pomocí složené Simpsonovy metody.

Řešení: Při této metodě výpočtu budeme využívat vzorec

$$\int_a^b f(x) dx = \frac{b-a}{6n} \{ f(x_0) + 4[f(x_1) + f(x_3) + \dots + f(x_{2n-1})] + \\ + 2[f(x_2) + f(x_4) + \dots + f(x_{2n-2})] + f(x_{2n}) \} \quad .$$

Při volbě Simpsonova pravidla rozdělíme interval $\langle a, b \rangle = \langle 0, 1 \rangle$ na $2n = 20$ subintervalů, platí tedy $0 = x_0 < x_1 < \dots < x_{19} < x_{20} = 1$. Velikost těchto intervalů

vyjádříme vztahem $\frac{b-a}{2n} = \frac{1}{20}$.

Dosazením do vzorce dostáváme vztah

$$\int_0^1 x^2 \cos(x^2) \approx \frac{1}{20} \{ f(0) + 4[f(\frac{1}{20}) + f(\frac{3}{20}) + \dots + f(\frac{19}{20})] + \\ + 2[f(\frac{2}{20}) + f(\frac{4}{20}) + \dots + f(\frac{18}{20})] + f(1) \} \quad .$$

Pro jednoduchost a přehlednost sestavíme tabulku funkčních hodnot $f(x_i)$.

x_i	$f(x_i)$
0,00	0,0000000
0,05	0,0025000
0,10	0,0099995
0,15	0,0224943
0,20	0,0399680
0,25	0,0623780
0,30	0,0896357
0,35	0,1215820
0,40	0,1579564
0,45	0,1983623
0,50	0,2422281
0,55	0,2887649
0,60	0,3369229
0,65	0,3853482
0,70	0,4323431
0,75	0,4758325
0,80	0,5133413
0,85	0,5419869
0,90	0,5584937
0,95	0,5592339
1,00	0,5403023

Tabulka 3

Dosazením těchto funkčních hodnot do vzorce a následným výpočtem dostáváme hodnotu integrálu.

Závěr: Přibližná hodnota integrálu vypočtená Simpsonovou metodou je 0,265600193 .

2.4 Metoda polovičního kroku

Tato metoda slouží ke zpřesnění výsledku numerické integrace, tedy k minimalizaci chyby. V některé literatuře můžeme tuto kapitolu nalézt pod názvem Richardsonova extrapolace nebo Rungeova metoda.

Předpokládejme, že pro chybu metody platí

$$\hat{A} = A(h) + ch^r + O(h^s), \quad (2.4-1)$$

kde $s > r > 0$. \hat{A} v tomto zápisu označuje požadovanou přesnou hodnotu numerické metody. $A(h)$ vyjadřuje aproximaci \hat{A} , která závisí na parametru h , c je konstanta. Zřejmě platí $\lim_{h \rightarrow 0} A(h) = \hat{A}$. Součet $ch^r + O(h^s)$ je zde ve významu chyby metody. Člen ch^r udává hlavní část chyby a $O(h^s)$ členy chyby vyššího řádu ($s > r$). Jestliže je $p \neq 0$ reálné číslo, pak platí také vztah

$$\hat{A} = A(ph) + c(ph)^r + O(h^s). \quad (2.4-2)$$

Vztah (2.4-1) vynásobme $(-p^r)$ a sečteme s výrazem (2.4-2). Jestliže z výsledné rovnice vyjádříme \hat{A} , dostáváme tvar

$$\hat{A} = \frac{A(ph) - p^r A(h)}{1 - p^r} + O(h^s).$$

Tento vztah umožňuje aproximovat \hat{A} s chybou $s > r$. K tomuto vyjádření jsou využity dvě aproximace, které jsou zastoupeny hodnotami parametrů h a

ph . Jestliže zvolíme $p = \frac{1}{2}$, pak mluvíme o *metodě polovičního kroku*.

Jestliže budeme metodu polovičního kroku aplikovat na lichoběžníkové pravidlo, dostáváme předpis

$$I = I_{\frac{h}{2}} + \frac{(I_{\frac{h}{2}} - I_h) \frac{2}{3}}{3} ,$$

kde I značí přesnou hodnotu integrálu, I_h aproximovanou hodnotu integrálu vypočtenou lichoběžníkovou metodou s krokem h a $I_{\frac{h}{2}}$ aproximovanou hodnotu integrálu získanou lichoběžníkovým pravidlem s krokem $\frac{h}{2}$.

Pro chybu metody platí vztah

$$R_{\frac{h}{2}} = \frac{I_{\frac{h}{2}} - I_h}{3} , \text{ kde } R_{\frac{h}{2}} = A \left(\frac{h}{2} \right)^2 .$$

Nyní zpřesníme řešení v Simpsonově metodě. Pro určení přesné hodnoty integrálu platí vztah

$$I = I_{\frac{h}{2}} + \frac{(I_{\frac{h}{2}} - I_h) \frac{2}{15}}{15} .$$

Chyba metody je určena vzorcem

$$R_{\frac{h}{2}} = \frac{I_{\frac{h}{2}} - I_h}{15} .$$

Označení je stejné jako u zpřesnění lichoběžníkové metody.

2.5 Obecný problém numerické kvadratury

Obecný operátor numerické integrace můžeme vyjádřit ve tvaru

$$L[f(x)] = f(b) - f(a) + \sum_{j=1}^n \sum_{i=1}^m A_{ij} f^{(i)}(a_{ij}) .$$

Jestliže dosadíme do tohoto vztahu za $f(x)$ funkci $\int_{-\infty}^x g(x) dx$, dostaneme

$$L[f(x)] = L\left[\int_{-\infty}^x g(x) dx\right] = \int_a^b g(x) dx + \sum_{j=1}^n \sum_{i=1}^m A_{ij} g^{(i-1)}(a_{ij}) .$$

Integrační rovnice má poté tvar

$$\int_a^b g(x) dx + \sum_{j=1}^n \sum_{i=1}^m A_{ij} g^{(i-1)}(a_{ij}) = E \quad (3)$$

Jestliže bude $E=0$ a pokud budeme předchozí rovnici řešit vzhledem

k $\int_a^b g(x) dx$, dojdeme k tomu, že aproximace určitého integrálu funkce

$g(x)$ je lineární kombinace hodnot funkce $g(x)$ a jejich derivací.

Za problém numerické integrace považujeme určení koeficientů A_{ij} a bodů a_{ij} tak, aby aproximace dosahovala požadované přesnosti.

Jestliže je funkce $g(x)$ polynom dostatečně nízkého stupně, využijeme aproximaci pomocí polynomů, tzn. budeme volit koeficienty A_{ij} a body a_{ij} tak, aby bylo $E=0$. Soustředíme se na případ, kdy je $m=1$, což znamená, že integrál vyjádříme jako lineární kombinaci funkčních hodnot (viz lichoběžníkové pravidlo). Z pohledu teoretického i praktického se jedná o nejdůležitější případ. Po úpravách a změnách v označení předchozí rovnice, dostáváme tvar

$$\int_a^b f(x) dx = \sum_{j=1}^n H_j f(a_j) + E \quad (2.5-1)$$

3 E označuje chybu numerické kvadratury

Otázkou zůstává, jakého musí být polynom stupně, aby bylo $E=0$, neklademe-li žádná omezení na body a_j ani na koeficienty H_j . Tento stupeň polynomu nazveme *řád přesnosti* vzorce. Máme k dispozici $2n$ konstant (n bodů a_j , n koeficientů H_j), proto můžeme tvrdit, že tento nejvyšší stupeň je $2n-1$.

2.6 Newton-Cotesovy kvadraturní vzorce

2.6.1 Interpolace

V praktických výpočtech je velmi vhodné a důležité nahradit danou funkci $f(x)$ funkcí $g(x)$, se kterou se lépe pracuje. Takový požadavek, aby funkce $g(x)$ měla v daných bodech stejné hodnoty jako funkce $f(x)$, se nazývá *interpolace*.

Nejčastějším způsobem nahrazování funkce je sestavení polynomu, tzv. interpolačního polynomu.

Věta 1.: (Existence a jednoznačnost interpolačního polynomu) Necht' v intervalu I je dáno $n+1$ různých bodů (uzlů) x_0, x_1, \dots, x_n . Dále necht' jsou dány funkční hodnoty $f_k = f(x_k)$, $k = 0, 1, \dots, n$. Pak existuje právě jeden polynom P (interpolační polynom funkce f s uzly x_0, \dots, x_n) stupně nejvýše n takový, že

$$P(x_k) = f_k, \quad k = 0, 1, \dots, n.$$

Příklad 2.6.1.a)

Zadání: Sestrojíme interpolační polynom pro funkci $f(x)$, která je dána tabulkou.

x_k	-1	0	2	3
f_k	4	4	-14	-8

Tabulka 4

Řešení: Dosadíme-li do vztahu

$$L_k(x) \equiv \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i}$$

a poté do vzorce

$$P(x) = \sum_{k=0}^n f_k L_k(x),$$

dostáváme

$$\begin{aligned} P(x) &= f_0 L_0(x) + f_1 L_1(x) + f_2 L_2(x) + f_3 L_3(x) = \\ &= 4 \frac{x(x-2)(x-3)}{-1(-1-2)(-1-3)} + 4 \frac{(x+1)(x-2)(x-3)}{1(-2)(-3)} - \\ &- 14 \frac{(x+1)x(x-3)}{(2+1)2(2-3)} - 8 \frac{(x+1)x(x-2)}{(3+1)3(3-2)} = 2x^3 - 5x^2 - 7x + 4. \end{aligned}$$

Příklad 2.6.1.b)

Pro $n=1$, tzn. pro dva uzly, dostáváme lineární interpolaci.

$$\begin{aligned} P(x) &= \sum_{k=0}^1 f_k \prod_{i=0, i \neq k}^1 \frac{x - x_i}{x_k - x_i} = f_0 \frac{x - x_1}{x_0 - x_1} + f_1 \frac{x - x_0}{x_1 - x_0} = \frac{f_0(x - x_1) - f_1(x - x_0)}{x_0 - x_1} = \\ &= f_0 + \frac{f_1 - f_0}{x_1 - x_0} (x - x_0). \end{aligned}$$

4 Tzv. Lagrangeovy koeficienty interpolačního polynomu. Platí $L_k(x_k) = 1$ pro $k = 0, 1, \dots, n$, ale pro $j \neq k$ je $L_k(x_j) = 0$

Zápis ve tvaru

$$P(x) = \sum_{k=0}^n f_k \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i}$$

nazveme *Lagrangeovým tvarem interpolačního polynomu*. Tento vzorec má dobré využití ve výpočetní technice, protože je dobře programovatelný.

Při nahrazení funkce f interpolačním polynomem v intervalu I i mimo uzly x_k je nutné zjistit chybu metody, tj. odhad rozdílu $R(x) \equiv f(x) - P(x)$.

Věta 2.: Necht' interpolovaná funkce f má $(n+1)$ -ní derivaci spojitou na intervalu I . Necht' interpolační uzly x_0, x_1, \dots, x_n leží v I . Pak ke každému $x \in I$ existuje bod $\zeta = \zeta(x) \in \langle \min(x, x_0, \dots, x_n), \max(x, x_0, \dots, x_n) \rangle$ takový,

$$\text{že } R(x) \equiv f(x) - P(x) = L(x) \frac{f^{(n+1)}(\zeta)}{(n+1)!},$$

$$\text{kde } L(x) = (x - x_0)(x - x_1) \dots (x - x_n).$$

2.6.2 Ekvidistantní uzly

Nejčastější způsob dělení intervalu $\langle a, b \rangle$ uzly x_0, x_1, \dots, x_n je ten, že jsou tyto uzly rovnoměrně rozloženy a vzestupně uspořádány. Označíme-li h jako krok, tzn. vzdálenost dvou sousedních uzlů, můžeme uzel x_k vyjádřit jako

$$x_k = x_0 + kh,$$

kde $k = 0, 1, \dots, n$. Uzly, které splňují tuto podmínku, nazýváme ekvidistantní.

Zavedeme proměnnou t , která vyjadřuje polohu bodu x vzhledem k uzlu x_0 a platí pro ní vztah

$$t = \frac{(x - x_0)}{h}.$$

Substituce $x = x_0 + th$ převede uzly $x_0, x_0 + h, \dots, x_0 + nh$ na celočíselné uzly $0, 1, \dots, n$.

2.6.3 Obecná formule pro ekvidistantní uzly

Položíme $x_k = x_0 + kh$, $k=0, \pm 1, \pm 2, \dots$. Necht' integrační meze a, b , $a < b$ splývají s nějakými uzly x_k . Např. $a = x_p$, $b = x_q$, kde $p < q$. Funkci $f(x)$ nahradíme interpolačním polynomem $P(x)$, který má uzly x_0, x_1, \dots, x_n .

Pozn.: Vzájemná poloha integračního intervalu $\langle a, b \rangle = \langle x_p, x_q \rangle$ a interpolačního intervalu $\langle x_0, x_n \rangle$ je zcela libovolná.

Budeme integrovat rovnost

$$f(x) = P(x) + R(x),$$

ve které R značí chybu interpolace. Integrál má tedy tvar

$$\int_{x_p}^{x_q} f(x) dx = \int_{x_p}^{x_q} P(x) dx + \int_{x_p}^{x_q} R(x) dx = \int_{x_p}^{x_q} P(x) dx + r,$$

kde r vyjadřuje chybu integrační metody.

Substitucí $x = x_0 + th$, přičemž $dx = h dt$ a při dosazení do vztahu

$$p(t) = \sum_{k=0}^n f(k) \frac{(-1)^{n-k}}{k!(n-k)!} \prod_{i=0, i \neq k}^n (t-i)^5$$

dostaneme výraz

$$\int_{x_p}^{x_q} P(x) dx = h \sum_{k=0}^n f(k) \frac{(-1)^{n-k}}{k!(n-k)!} \int_p^q \prod_{i=0, i \neq k}^n (t-i) dt.$$

Obecný tvar pro metody, které využívají interpolaci při ekvidistantních uzlech, lze tedy vyjádřit jako

$$\int_{x_p}^{x_q} f(x) dx = h \sum_{k=0}^n A_k f(x_k) + r,$$

kde je označeno

5 Langrangeův tvar interpolačního polynomu pro ekvidistantní uzly.

$$A_k = \frac{(-1)^{n-k}}{k!(n-k)!} \int_p^q \prod_{i=0, i \neq k}^q (t-i) dt .$$

Koeficienty A_k jsou funkcemi čtyř proměnných (celočíslných) p, q, n, k , kde $p < q, n \geq 0, 0 \leq k \leq n$. Avšak nezávisejí na funkci f , ani na kroku h . Integrační metody se rozlišují způsobem volby vzájemné polohy integračního a interpolačního intervalu, resp. volbou hodnot dvojice p, q .

2.6.4 Newton-Cotesovy metody uzavřeného typu

Jedná se o speciální tvar kvadraturního vzorce

$$\int_{x_p}^{x_q} f(x) dx = h \sum_{k=0}^n A_k f(x_k) + r ,$$

kde volíme $p=0, q=n$. Jedná se tedy o případ totožnosti integračního a interpolačního intervalu, což má za důsledek splynutí integračních mezí a, b s krajními uzly x_0, x_n interpolačního intervalu. Rozdělení intervalu $\langle a, b \rangle$ je rovnoměrné. Pro uzly interpolace tedy platí

$$x_k = a + kh, \quad k = 0, 1, \dots, n, \quad h = \frac{(b-a)}{n} .$$

Ve výsledném vzorci bychom požadovali, aby obsahoval délku intervalu $b-a$. Z tohoto důvodu vynásobíme výraz

$$A_k = \frac{(-1)^{n-k}}{k!(n-k)!} \int_0^n \prod_{i=0, i \neq k}^n (t-i) dt$$

činitelem n a výsledný tvar kvadraturního vzorce je

$$\int_a^b f(x) dx = (b-a) \sum_{k=0}^n K_{n,k} f(x_k) + r ,$$

kde r značí chybu integrační metody a $K_{n,k}$ jsou konstanty, pro které platí

$$K_{n,k} = \frac{(-1)^{n-k}}{nk!(n-k)!} \int_0^n \prod_{i=0, i \neq k}^n (t-i) dt .$$

Tyto konstanty jsou nezávislé, jak vzorec napovídá, na f i na h .

Věta 1.: Pro Newton-Cotesovy koeficienty $K_{n,k}$ platí

$$1. \quad K_{n,k} = K_{n,n-k} \quad \text{pro } k=0, 1, \dots, n.$$

Vlastnost nazveme *symetrií koeficientů*.

$$2. \quad \sum_{k=0}^n K_{n,k} = 1.$$

Tuto vlastnost označme jako *kontrolní součtový vztah*.

Uvedme některé Newton-Cotesovy koeficienty. V následující tabulce jsou vypočteny pro $n=1, 2, 3, 4, 5, 6$. Z důvodu symetričnosti je uvedena pouze polovina pro každé n . Z výše uvedené věty lze snadno druhou polovinu doplnit. Koeficienty jsou vyjádřeny ve tvaru zlomku, společný jmenovatel těchto zlomků je uveden ve zvláštním sloupci.

n \ k	0	1	2	3	Jmenovatel	Chyba r
1	1				2	$-\frac{1}{12} f^{(2)}(\xi) h^3$
2	1	4			6	$-\frac{1}{90} f^{(4)}(\xi) h^5$
3	1	3			8	$-\frac{3}{80} f^{(4)}(\xi) h^5$
4	17	32	12		90	$-\frac{8}{945} f^{(6)}(\xi) h^7$
5	19	75	50		288	$-\frac{275}{12096} f^{(6)}(\xi) h^7$
6	41	216	27	272	840	$-\frac{9}{1400} f^{(8)}(\xi) h^9$

Tabulka 5

Důkladnějším zjišťováním chyby dospějeme k tvrzení:

Věta 2.: Necht' funkce f má $(n+2)$ -tou spojitou derivaci v intervalu $\langle a, b \rangle$.

Pak lze chybu

$$r_n(f) = \int_a^b f(x) dx - (b-a) \sum_{k=0}^n K_{n,k} f(x_k)$$

Newton-Cotesovy metody uzavřeného typu s $(n+1)$ rovnoměrně rozloženými

uzly $x_k = a + kh$, $k = 0, 1, \dots, n$, $h = \frac{(b-a)}{n}$, vyjádřit pro n sudé

$(n=2m)$ ve tvaru

$$r_n = h^{n+3} \frac{f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t \prod_{i=0}^n (t-i) dt$$

a pro n liché $(n=2m+1)$ ve tvaru

$$r_n = h^{n+2} \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_0^n \prod_{i=0}^n (t-i) dt,$$

kde $a < \xi < b$. Integrály vyjdou záporně.

Při praktických výpočtech se ve většině případů upřednostňují vzorce s lichým počtem uzlů, protože pro ně je n sudé, např. $n=2m$. Na rozdíl od vzorců se sudým počtem uzlů, pro ně je $n=2m+1$. Výhodou užití vzorce s lichým počtem uzlů je, že se počítá o jednu funkční hodnotu integrované funkce méně, avšak chyba je řádově stejná u obou vzorců.

Speciálním případem Newton-Cotesovy metody uzavřeného typu, kdy zvolíme $n=1$, je *lichoběžníková metoda*. V tomto případě použijeme lineární interpolaci. Jestliže položíme $x_0=a$, $x_1=b$, $h=b-a$. Po dosazení do vzorce pro Newton-Cotesovu metodu uzavřeného typu

$$\int_a^b f(x) dx = (b-a) \sum_{k=0}^n K_{n,k} f(x_k) + r$$

a využitím tabulky 5 dostáváme známý vzorec pro výpočet určitého integrálu lichoběžníkovou metodou

$$\int_a^b f(x) dx = \frac{(b-a)}{2} (f(a) + f(b)) - \frac{h^3}{12} f^{(2)}(\xi) ,$$

kde $\xi \in (a, b)$.

Další využívanou metodou, která vychází z Newton-Cotesovy metody uzavřeného typu, je *Simpsonova metoda*. V tomto případě zvolíme $n=2$. Jedná

se tedy o kvadratickou interpolaci. Položme tentokrát $x_0=a$, $x_1=\frac{(a+b)}{2}$,

$x_2=b$, $h=\frac{(b-a)}{2}$. Využijeme opět vztah pro Newton-Cotesovu metodu

uzavřeného typu a tabulku 5. Po dosazení dostáváme vzorec

$$\int_a^b f(x) dx = \frac{b-a}{6} [f(a) + 4f(\frac{a+b}{2}) + f(b)] - \frac{h^5}{90} f^{(4)}(\xi) ,$$

kde $\xi \in (a, b)$.

2.6.5 Newton-Cotesovy metody otevřeného typu

Zde mluvíme o speciálním případě interpolačního kvadrurního vzorce

$$\int_{x_p}^{x_q} f(x) dx = h \sum_{k=0}^n A_k f(x_k) + r ,$$

kde položíme $p=-1$, $q=n+1$. Což má za následek využití pouze vnitřních

uzlů $x_k = a + (k+1)h$, $k=0, 1, \dots, n$, $h = \frac{(b-a)}{(n+2)}$ v integračním intervalu

$\langle a, b \rangle \equiv \langle x_{-1}, x_{n+1} \rangle$ interpolačním polynomem P .

Analogickým postupem jako při metodách uzavřeného typu (vynásobení činitelem n , apod.) zapíšeme vzorec ve tvaru

$$\int_a^b f(x) dx = (b-a) \sum_{k=0}^n A_{n,k} f(x_k) + r,$$

kde r opět značí chybu metody. Pro konstanty $A_{n,k}$ platí

$$A_{n,k} = \frac{(-1)^{n-k}}{(n+2)k!(n-k)!} \int_{-1}^{n+1} \prod_{i=0, i \neq k}^{n+1} (t-i) dt.$$

U Newton-Cotesovy metody otevřeného typu můžeme vyslovit analogickou větu 2. Tentokrát však koeficienty vycházejí kladně.

Při volbě $n=0$ dostáváme speciální případ Newton-Cotesovy metody otevřeného typu, tzv. *obdélníkovou metodu*. Pro výpočet určitého integrálu touto metodou platí vztah

$$\int_a^b f(x) dx = (b-a) f\left(\frac{a+b}{2}\right) + \frac{h^3}{3} f^{(2)}(\zeta),$$

kde $\zeta \in (a, b)$, $h = \frac{(b-a)}{2}$.

Další dobře známou metodou, která je odvozena od Newton-Cotesovy metody otevřeného typu je *Simpsonova otevřená metoda*, která se hojně využívá při numerickém řešení diferenciálních rovnic metodou predikce-korekce.

V následující tabulce 6 jsou uvedeny koeficienty $A_{n,k}$ a chyby pro Newton-Cotesovy metody otevřeného typu. Systém vyplňování tabulky 6 je stejný jako u tabulky 5, tedy je uvedena jen polovina koeficientů. Tyto koeficienty jsou uvedeny opět ve tvaru zlomků. Společný jmenovatel je uveden ve zvláštním sloupci.

n \ k	0	1	2	Jmenovatel	Chyba r
0	1			1	$\frac{1}{3} f^{(2)}(\xi) h^3$
1	1			2	$\frac{3}{4} f^{(2)}(\xi) h^3$
2	2	-1		3	$\frac{14}{45} f^{(4)}(\xi) h^5$
3	11	1		24	$\frac{95}{144} f^{(4)}(\xi) h^5$
4	11	-14	26	20	$\frac{41}{140} f^{(6)}(\xi) h^7$
5	611	-453	562	1440	$\frac{5257}{8640} f^{(6)}(\xi) h^7$

Tabulka 6

2.7 Gaussova kvadratura

2.7.1 Hermitova interpolace

Při hledání Hermitovy interpolace budeme vycházet z obecného interpolačního operátoru, který má tvar

$$L[f(x)] = f(x) + \sum_{j=1}^n \sum_{i=0}^m A_{ij}(x) f^{(i)}(a_{ij}),$$

kde a_{ij} jsou konstanty a $A_{ij}(x)$ polynomy v x . Zvolme $m=1$. Také budeme předpokládat, že známe v r bodech, $r < n$ kromě funkčních hodnot i hodnoty prvních derivací. Z těchto předpokladů lze vyjádřit $f(x)$ vztahem

$$f(x) = \sum_{j=1}^n h_j(x) f(a_j) + \sum_{j=1}^r \bar{h}_j(x) f'(a_j) + E(x) = y(x) + E(x),$$

kde $h_j(x)$ i $\bar{h}_j(x)$ jsou polynomy. Podle kritéria pro interpolační aproximaci požadujeme, aby pro chybu $E(x)$ platilo

$$\begin{aligned} E(a_j) &= 0, \quad j = 1, 2, \dots, n, \\ E'(a_j) &= 0, \quad j = 1, 2, \dots, r. \end{aligned}$$

Podobně jako u Lagrangeovy interpolace chceme, aby pro polynomy $h_j(x)$, $\bar{h}_j(x)$ platily podmínky

$$\begin{aligned} h_j(a_k) &= \delta_{jk}, \quad j, k = 1, 2, \dots, n, \\ \bar{h}_j(a_k) &= 0, \quad j = 1, 2, \dots, r, \quad k = 1, 2, \dots, n, \\ h'_j(a_k) &= 0, \quad j = 1, 2, \dots, n, \quad k = 1, 2, \dots, r, \\ \bar{h}'_j(a_k) &= \delta_{jk}, \quad j, k = 1, 2, \dots, r. \end{aligned}$$

⁶ δ je tzv. Kroneckerovo delta, pro které platí $\delta_{jk} = 0$ pro $j \neq k$ a $\delta_{jk} = 1$ pro $j = k$

Pro odvození tvaru funkcí $h_j(x)$, $\bar{h}_j(x)$ budeme používat označení

$$p_n(x) = (x - a_1) \dots (x - a_n),$$

$$p_r(x) = (x - a_1) \dots (x - a_r),$$

$$l_{jn}(x) = \frac{p_n(x)}{(x - a_j) p_n'(a_j)}, \quad j = 1, 2, \dots, n,$$

$$l_{jr}(x) = \frac{p_r(x)}{(x - a_j) p_r'(a_j)}, \quad j = 1, 2, \dots, r.$$

Z důvodu splnění podmínek položíme

$$h_j(x) = t_j(x) l_{jn}(x) l_{jr}(x), \quad j = 1, 2, \dots, r,$$

$$h_j(x) = l_{jn}(x) \frac{p_r(x)}{p_r(a_j)}, \quad j = r+1, \dots, n,$$

kde $t_j(x)$ značí lineární polynom. Polynom $h_j(x)$ splňuje všechny výše uvedené podmínky až na $h_j(a_j) = 1$, $j = 1, 2, \dots, r$ a $\bar{h}_j(a_j) = 0$, $j = 1, 2, \dots, r$. Pro splnění musíme zvolit

$$t_j(a_j) = 1, \quad j = 1, 2, \dots, r,$$

$$t_j'(a_j) + l_{jn}'(a_j) + l_{jr}'(a_j) = 0, \quad j = 1, 2, \dots, r.$$

Podobným způsobem vyjádříme i polynom $\bar{h}_j(x)$. Položíme

$$\bar{h}_j(x) = s_j(x) l_{jr}(x) l_{jn}(x), \quad j = 1, 2, \dots, r,$$

kde $s_j(x)$ je opět lineární polynom. Pro splnění podmínek musíme položit

$$s_j(a_j) = 0, \quad j = 1, 2, \dots, r,$$

$$s_j'(a_j) = 1, \quad j = 1, 2, \dots, r.$$

Lineární funkce, které splňují všechny výše uvedené podmínky, jsou

$$t_j(x) = 1 - (x - a_1)(l_{jn}'(a_j) + l_{jr}'(a_j)),$$

$$s_j(x) = x - a_j.$$

V další části nalezneme chybu $E(x)$. Necht' tedy

$$F(z) = f(z) - y(z) - (f(x) - y(x)) \frac{p_n(z) p_r(z)}{p_n(x) p_r(x)},$$

kde x není tabulkový bod⁷. Funkce $F(z)$ má $n+r+1$ kořenů⁸. Podle zobecněné Rollovy věty existuje bod ζ , který je v intervalu určeném kořeny tak, že

$$0 = F^{(n+r)}(\zeta) = f^{(n+r)}(\zeta) - (f(x) - y(x)) \frac{(n+r)!}{p_n(x) p_r(x)} .$$

Pro chybu $E(x)$ tedy platí

$$E(x) = \frac{p_n(x) p_r(x)}{(n+r)!} f^{(n+r)}(\zeta) .$$

Tento předpis platí i pro taková x , která jsou tabulková.

Interpolační vzorec je tedy ve tvaru

$$f(x) = \sum_{j=1}^n h_j(x) f(a_j) + \sum_{j=1}^r \bar{h}_j(x) f'(a_j) + \frac{p_n(x) p_r(x)}{(n+r)!} f^{(n+r)}(\zeta) , \quad (2.7.1-1)$$

kde

$$h_j(x) = [1 - (x - a_j)(l'_{jn}(a_j) + l'_{jr}(a_j))] l_{jn}(x) l_{jr}(x), \quad j = 1, \dots, r ,$$

$$h_j(x) = l_{jn}(x) \frac{p_r(x)}{p_r(a_j)}, \quad j = r+1, \dots, n ,$$

$$\bar{h}_j(x) = (x - a_j) l_{jr}(x) l_{jn}(x), \quad j = 1, \dots, r .$$

Vzorec (2.7.1-1) nazveme *modifikovaný Hermitův interpolační vzorec*.

Jestliže zvolíme $r=n$, pak má vzorec tvar

$$f(x) = \sum_{j=1}^n h_j(x) f(a_j) + \sum_{j=1}^n \bar{h}_j(x) f'(a_j) + \frac{p_n^2(x)}{(2n)!} f^{(2n)}(\zeta) , \quad (2.7.1-2)$$

ve kterém je

$$h_j(x) = [1 - 2(x - a_j)l'_j(a_j)]l_j^2(x),$$

$$\bar{h}_j(x) = (x - a_j)l_j^2(x), \quad j = 1, \dots, n.$$

⁷ Znamé hodnoty funkce $f(x)$ (popř. i hodnoty některých derivací) v jisté množině bodů.

⁸ dvojnásobné a_1, \dots, a_r , jednoduché a_{r+1}, \dots, a_n a x .

Ve vzorcích pro $h_j(x)$ a $\bar{h}_j(x)$ je voleno $l_{jn}(x)=l_j(x)$. Výše uvedený vzorec (2.7.1-2) se nazývá *Hermitův interpolační vzorec*.

2.7.2 Legendrovy polynomy

Legendrovým polynomem n -tého stupně nazveme polynom

$$P_n(x) = \frac{1 \cdot 2 \cdot 3 \cdot \dots \cdot (2n-1)}{n!} \left[x^n - \frac{n(n-1)}{2(2n-1)} x^{n-2} + \frac{n(n-1)(n-2)(n-3)}{24(2n-1)(2n-3)} x^{n-4} - \dots \right].$$

Funkce $y=P_n(x)$ vyhovuje Legendrově diferenciální rovnici

$$(1-x^2)y'' - 2xy' + n(n+1)y = 0 .$$

Mezi základní vlastnosti Legendrových polynomů patří

$$P_n(-x) = (-1)^n P_n(x),$$

$$P_n(1) = 1 .$$

Pro $|x| \leq 1$ platí

$$|P_n(x)| \leq 1, \frac{1}{n^2} |P_n'(x)| \leq 1, \frac{1}{n^4} |P_n''(x)| \leq 1, \dots, \frac{1}{n^{2k}} |P_n^{(k)}(x)| \leq 1 .$$

Kořeny rovnice $P_n(x)=0$ pro $n=1, 2, \dots$ leží v intervalu $(-1,1)$.

Pro $n \geq 2$ platí vztah

$$P_n(x) = \frac{(2n-1)}{n} x P_{n-1}(x) - \frac{n-1}{n} P_{n-2}(x) .$$

Legendrův polynom řadíme mezi ortogonální polynomy⁹ (stejně i Hermitův polynom), které lze vyjádřit vztahem

$$f_n(x) = d_n \sum_{m=0}^N c_m g_m(x) ,$$

kde při vhodné volbě $d_n, c_m, g_m(x), N$ dostáváme požadovaný polynom.

2.7.3 Odvození Gaussova kvadraturního vzorce

Nechť jsou meze a, b ve vzorci (2.5-1) konečné. Tento vzorec je přesný pro polynomy stupně nejvýše $2n-1$. Jestliže do vzorce dosadíme za $f(x) = x^k$, $k = 0, 1, \dots, 2n-1$ a položíme $E=0$, dostaneme soustavu $2n$ rovnic o $2n$ neznámých konstant, tedy

$$\alpha_k = \sum_{j=1}^n H_j a_j^k, \quad k = 0, 1, \dots, 2n-1,$$

kde

$$\alpha_k = \int_a^b x^k dx = \frac{b^{k+1} - a^{k+1}}{k+1} .$$

Řešením těchto nelineárních rovnic jsou uzly a koeficienty, které hledáme. V následujícím textu se budeme snažit objasnit tři problémy:

1. bez počítání zjistit, zda jsou koeficienty a uzly reálné;
2. určit E v případě, že $f(x)$ není polynom stupně nejvýše $2n-1$;
3. ukázat, že uzly jsou v mnoha případech kořeny obecně známých polynomů.

⁹ Necht' $w(x)$ je funkce integrovatelná na $\langle a, b \rangle$ a $w(x) > 0$ skoro všude na $\langle a, b \rangle$. Tato funkce se nazývá váhová funkce.

Skalární součin funkcí f, g je definován takto $(f, g) = \int_a^b w(x) f(x) g(x) dx$.

Jestliže $\langle f, g \rangle = 0$, pak říkáme, že f, g jsou ortogonální na $\langle a, b \rangle$ s váhou w .

Budeme vycházet z Hermitova interpolačního vzorce

$$f(x) = \sum_{j=1}^n h_j(x) f(a_j) + \sum_{j=1}^n \bar{h}_j(x) f'(a_j) + \frac{p_n^2(x)}{(2n)!} f^{(2n)}(\zeta) .$$

Tento vzorec je přesný pro polynomy stupně nejvýše $2n-1$. Jestliže budeme vzorec integrovat od a do b , dostaneme

$$\int_a^b f(x) dx = \sum_{j=1}^n H_j f(a_j) + \sum_{j=1}^n \bar{H}_j f'(a_j) + E ,$$

kde

$$H_j = \int_a^b h_j(x) dx, \quad \bar{H}_j = \int_a^b \bar{h}_j(x) dx$$

a

$$E = \int_a^b \frac{p_n^2(x)}{(2n)!} f^{(2n)}(\zeta) dx .$$

Nutnou a postačující podmínkou, abychom dosáhli požadované přesnosti $2n-1$, je splnění $\bar{H}_j = 0, j = 1, \dots, n$. Ve výše uvedeném vzorci nejsou kladena žádná omezení na uzly, proto neexistuje žádný vzorec typu (2.5-1) s řádem přesnosti $2n-1$, jestliže nelze nalézt takové uzly, pro které $\bar{H}_j = 0, j = 1, \dots, n$.

Platí

$$\bar{H}_j = \int_a^b (x-a_j) l_j^2(x) dx = \int_a^b p_n(x) \frac{l_j(x)}{p_n'(a_j)} dx \quad ^{10} .$$

Polynom $p_n(x)$ je n -tého stupně a polynom $l_j(x)$ stupně $n-1$. Pro splnění $\bar{H}_j = 0, j = 1, \dots, n$ stačí, aby $p_n(x)$ byl ortogonální v intervalu $\langle a, b \rangle$ ke všem polynomům stupně nejvýše $n-1$. Tuto podmínku budeme považovat také za nutnou. Jestliže budeme předpokládat, že interval

¹⁰ $\bar{h}_j(x) = (x-a_j) l_j^2(x), j = 1, \dots, n.$

$\langle a, b \rangle = \langle -1, 1 \rangle$ ¹¹, pak je ortogonální polynom $p_n(x)$ násobkem Legendrova polynomu $P_n(x)$. Jestliže má polynom $p_n(x)$ koeficient u nejvyšší mocniny roven 1, pak podle definice Legendrových polynomů platí

$$p_n(x) = \frac{2^n (n!)^2}{(2n)!} P_n(x) .$$

Protože kořeny Legendrových polynomů leží v intervalu $(-1, 1)$, je problém reálných uzlů a koeficientů vyřešen. Kořeny těchto polynomů jsou tabelovány pro všechny prakticky významné hodnoty n . Některé tyto hodnoty jsou uvedeny v následující tabulce 7.

n	Uzly a_j	Koeficienty H_j
2	$\pm 0,577350 = \pm \frac{1}{\sqrt{3}}$	1
3	0	$\frac{8}{9}$
	$\pm 0,774597$	$\frac{5}{9}$
4	$\pm 0,339981$	0,652145
	$\pm 0,861136$	0,347855
5	0	0,568889
	$\pm 0,538469$	0,478629
	$\pm 0,906180$	0,236927

Tabulka 7: Kořeny Legendrových polynomů

K nalezení příslušných koeficientů využijeme rovnici $h_j(x) = [1 - 2(x - a_j)l'_j(a_j)]l_j^2(x)$, $j = 1, \dots, n$. Dostáváme tedy

$$H_j = \int_{-1}^1 h_j(x) dx = \int_{-1}^1 [1 - 2l'_j(a_j)(x - a_j)]l_j^2(x) dx =$$

¹¹ změnou proměnné $y = \left(\frac{1}{b-a}\right)(2x - a - b)$ přechází interval $\langle a, b \rangle$ v x na interval $\langle -1, 1 \rangle$ v y .

$$\begin{aligned}
&= \int_{-1}^1 l_j^2(x) dx - 2l_j'(a_j) \int_{-1}^1 (x-a_j)l_j^2(x) dx = \\
&= \int_{-1}^1 l_j^2(x) dx .
\end{aligned}$$

Pozn.: Druhý integrál je roven \bar{H}_j , tedy roven nule.

Z výše uvedeného vzorce je zřejmé, že koeficienty jsou kladné. Jednodušší vztah pro výpočet koeficientů získáme, jestliže dosadíme za $f(x)=l_k(x)$ ve vzorci (2.5-1). Protože je $l_k(x)$ polynom stupně $n-1$, je $E=0$ a dostáváme

$$\int_{-1}^1 l_k(x) dx = \sum_{j=1}^n H_j l_k(a_j) = H_k , \quad (2.7.3-1)$$

protože $l_k(a_j)=\delta_{kj}$, vzorce dohromady dávají

$$\int_{-1}^1 l_j^2(x) dx = \int_{-1}^1 l_j(x) dx .$$

Chybu, kterou vyjadřuje vzorec

$$E = \int_a^b \frac{p_n^2(x)}{(2n)!} f^{(2n)}(\zeta) dx ,$$

můžeme zjednodušit díky větě o střední hodnotě pro integrály ($p_n^2(x)$ je všude kladné). Necht' interval $\langle a, b \rangle = \langle -1, 1 \rangle$, pak platí

$$E = \frac{f^{(2n)}(\eta)}{(2n)!} \int_{-1}^1 p_n^2(x) dx ,$$

kde $\eta \in (-1, 1)$.

Kvadrurní vzorec, na jehož uzly a koeficienty nejsou kladena žádná omezení a zároveň jsou tyto uzly určeny tak, abychom dosáhli co největšího stupně přesnosti, nazýváme *Gaussův kvadrurní vzorec*. Speciálně vzorec (2.5-1) s uzly danými kořeny Legendrových polynomů stupně n a koeficienty, které nám dává vzorec (2.7.3-1) nazýváme *Legendrův-Gaussův kvadrurní vzorec*.

3 MATLAB

3.1 Úvod

MATLAB je programovací jazyk, který slouží k technickým výpočtům. V jednoduchém prostředí je možné provádět jak výpočty tak vizualizaci a samozřejmě vytvářet vlastní programy (aplikace). Problémy, ale i řešení, jsou vyjádřeny matematickým zápisem, proto je psaní programů intuitivní.

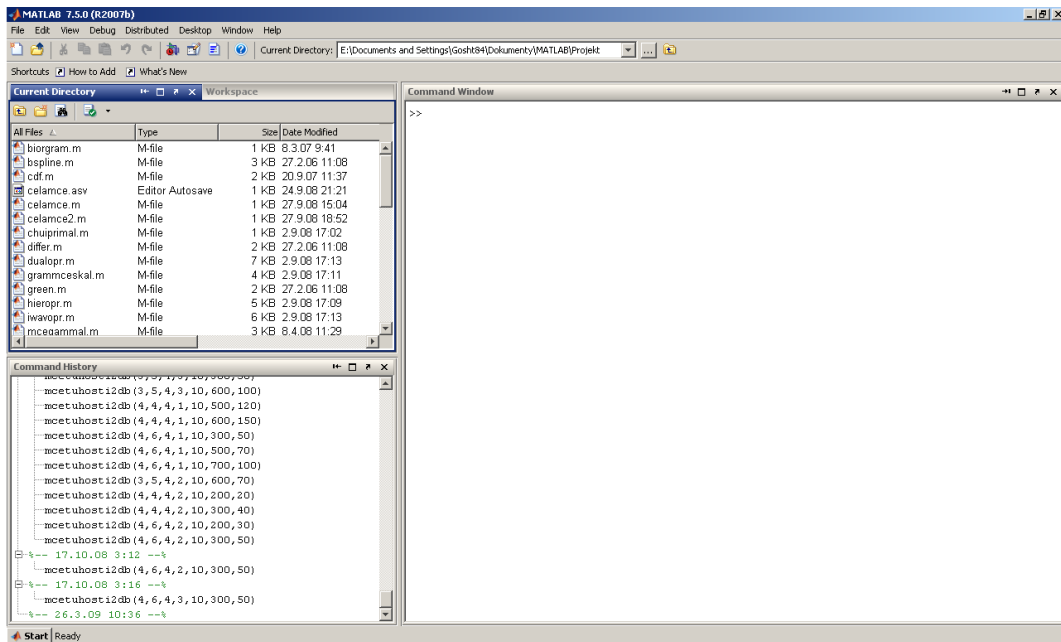
Jedná se o interaktivní systém. Základním datovým typem je matice, u které není nutné deklarovat její rozměr. Díky velkému množství vestavěných funkcí je možné řešit obtížné technické problémy, zejména ty, které lze zapsat maticovou formulací. Vzhledem k této vlastnosti je výpočet snadnější ve smyslu časové náročnosti, než kdyby uživatel pracoval v klasických programovacích jazycích, jako je např. C. Jako hlavní výhodu oproti ostatním matematickým programům bych označil možnost jeho rozšíření o tzv. M-soubory, což jsou funkce (programy), které vytváří sám uživatel.

Oblasti, kde se MATLAB využívá, jsou rozmanité. Ať už se jedná o inženýrské výpočty, modelování, simulaci fyzikálních jevů, analýzu dat, zpracování signálu nebo o vývoj aplikací včetně grafického uživatelského rozhraní. Tato diplomová práce využívá verzi 7.5.

3.2 Pracovní prostředí

Po spuštění MATLABu nás uvítá pracovní plocha systému, která je ve většině případů rozdělena do tří podoken. Pro uživatele je nejdůležitější okno, které nese název *Command Window*. Toto okno nám umožňuje pracovat se systémem jako takovým (psaní příkazů, ukládání proměnných, apod.). Okno v levé horní části pracuje se dvěma záložkami a to *Workspace* (zobrazení seznamu vytvořených proměnných) a *Current Directory* (zobrazení seznamu souborů v aktuálním pracovním adresáři). Třetí okno se nazývá *Command History*,

ve kterém jsou uvedeny všechny příkazy, které byly během naší práce se systémem spuštěny (použity).



Obr. 8: Pracovní prostředí MATLABu

3.3 Vytvoření a zápis proměnných

Každá proměnná bez ohledu na programovací jazyk má svůj název, typ a určitou hodnotu. V MATLABu je situace ulehčena tím, že každá proměnná je matice. Název proměnné musí začínat písmenem a může mít délku až 63^{32} znaků. Uživatel musí dbát na to, že MATLAB rozlišuje velká a malá písmena (A5 není to samé jako a5). Proměnná tedy může obsahovat znaky A-Z, a-z, 1-9 a znak podtržítka (_).

Vytvoření proměnné je v prostředí MATLABu vcelku jednoduchou záležitostí. Při vytváření napíšeme do *Command Window* přiřazovací příkaz

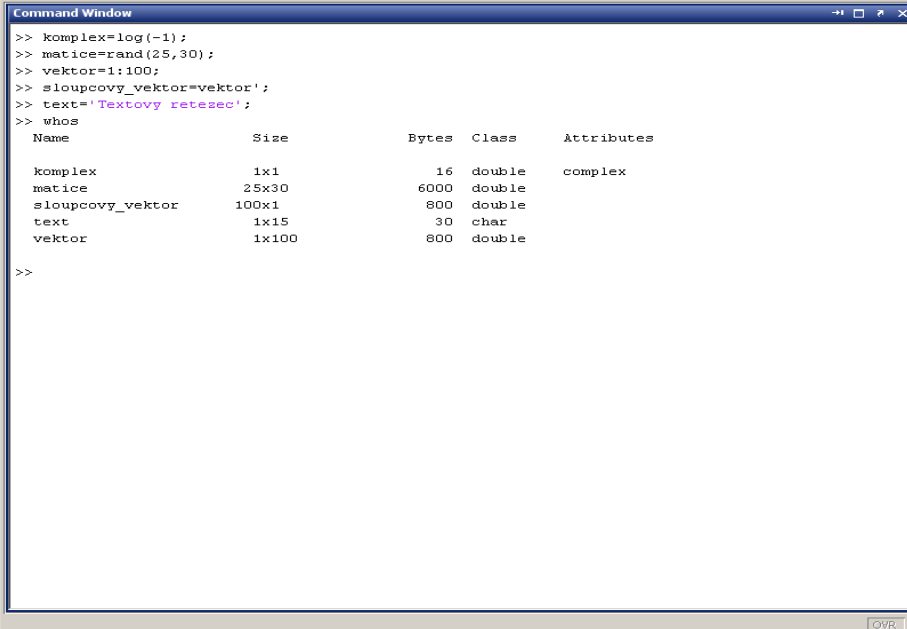
```
>> promenna=vyraz.
```

Určitá jména není vhodné použít pro vlastní proměnnou, protože mají předdefinovanou hodnotu. Jejich užitím na levé straně přiřazovacího příkazu se původní hodnota změní, např. příkazem `pi=9.5` předdefinujeme hodnotu

Ludolfova čísla na hodnotu 9,5. Některé často užívané proměnné uvedeme v následujícím výčtu:

<i>ans</i>	proměnná používaná MATLABem, jestliže jsme předem neuvedli vlastní název proměnné
<i>pi</i>	Ludolfovo číslo
<i>i,j</i>	imaginární jednotka komplexního čísla
<i>Inf</i>	nekonečno
<i>NaN</i>	neplatná numerická hodnota

Proměnná má platnost do ukončení programu. Jestliže chceme danou proměnnou zrušit, použijeme příkaz `clear jméno_proměnné`. Tento příkaz bez názvu proměnných smaže veškeré vytvořené proměnné. Během práce můžeme vytvořit velké množství proměnných s různou velikostí. Hodnoty proměnných jsou v paměti počítače uloženy v binárním formátu pohyblivé řádové čárky (každé číslo zabírá 8 byte). Pomocí příkazu `whos` vypíšeme seznam veškerých proměnných s jejich velikostí.



```
Command Window
>> komplex=log(-1);
>> matice=rand(25,30);
>> vektor=1:100;
>> sloupcovy_vektor=vektor';
>> text='Textovy resetec';
>> whos
Name                Size          Bytes  Class      Attributes
-----
komplex              1x1            16  double    complex
matice               25x30          6000  double
sloupcovy_vektor    100x1           800  double
text                 1x15            30   char
vektor               1x100           800  double
>>
```

Obr. 9: příkaz `whos`

3.4 Matice a vektory

Jak již bylo řečeno výše, základním datovým typem MATLABu je matice. Jestliže chceme do proměnné uložit skalár, pracujeme vlastně s maticí, která má rozměr 1×1 . Podobné je to i s vektorem, v tomto případě je však jeden z rozměrů různý od jedné. Samozřejmostí je rozlišování vektorů řádkových a sloupcových. V této práci se budeme držet standardního označení, tedy že první rozměr označuje počet řádků a druhý počet sloupců. Pro potřebu vyjádřit opačný vektor se v matematice užívá transpozice, kterou v MATLABu značí apostrof ('). Matici můžeme chápat jako vektor vektorů, z tohoto důvodu můžeme matici zapisovat i pomocí předem uložených vektorů.

3.4.1 Plnění matic a vektorů

Uložení hodnot do proměnných názorně vysvětlí následující tabulka.

	Skalár	Řádkový vektor	Sloupcový vektor	Matice
Příkaz	<code>>> x=3</code>	<code>>> v=[2,4,6]</code>	<code>>> u=[1;3;5]</code>	<code>>> A=[1 2;2 1]</code>
Výpis	x = 3	v = 2 4 6	u = 1 3 5	A = 1 2 2 1

Tabulka 8

MATLAB umožňuje pracovat s jednotlivými prvky matice nebo vektoru, popř. s nějakou částí matice (vektoru). Jestliže máme vytvořen vektor *a* a chceme zjistit hodnotu např. druhého prvku, stačí zadat příkaz

`>>a(2)` .

Jestliže potřebujeme zjistit hodnotu prvku v matici A, který je např. v třetím řádku a druhém sloupci, zadáme příkaz

```
>>A(3,2) .
```

3.4.2 Speciální znaky

V následujícím výčtu popíšeme nejpoužívanější speciální znaky. Je nutné si zapamatovat význam jednotlivých znaků, nelze např. zaměňovat čárku (,) a středník (;). Některé znaky jsme již užili, viz tabulka 8.

;	<i>středník</i>	ukončení příkazu bez výpisu, oddělovač sloupců
,	<i>čárka</i>	obecný oddělovač, oddělovač v řádku
	<i>mezera</i>	obecný oddělovač, oddělovač v řádku
:	<i>dvojtečka</i>	oddělovač ve výčtu (určení rozsahu)
'	<i>apostrof</i>	označení transpozice (umístěn za proměnnou)
()	<i>kulaté závorky</i>	přednost v matematickém výrazu, seznam parametrů funkce, index matice
[]	<i>hranaté závorky</i>	určují, že se jedná o vektor či matici
{}	<i>složené závorky</i>	určují, že jde o strukturu

3.5 Funkce

Funkce je posloupnost příkazů, které vstupní parametry (konstanty, proměnné) zpracovávají do parametrů výstupních podle určitého algoritmu. Rozlišujeme mezi funkcemi, které jsou přímo součástí MATLABu, tzv. built-in function, a funkcemi, které jsou ve formě předpisu (kombinují matematické a logické operace, built-in funkce, apod.). Tento předpis je uložen v samostatném souboru s příponou „.m“. Tyto funkce nazýváme m-funkce.

Z pohledu uživatele není mezi built-in funkcemi a m-funcemi rozdíl. Vždy je nutné vědět, jak se námi požadovaná funkce jmenuje, jaké má vstupní parametry a v jaké formě vrací výsledek. Obecně vypadá syntaxe volání funkce následovně.

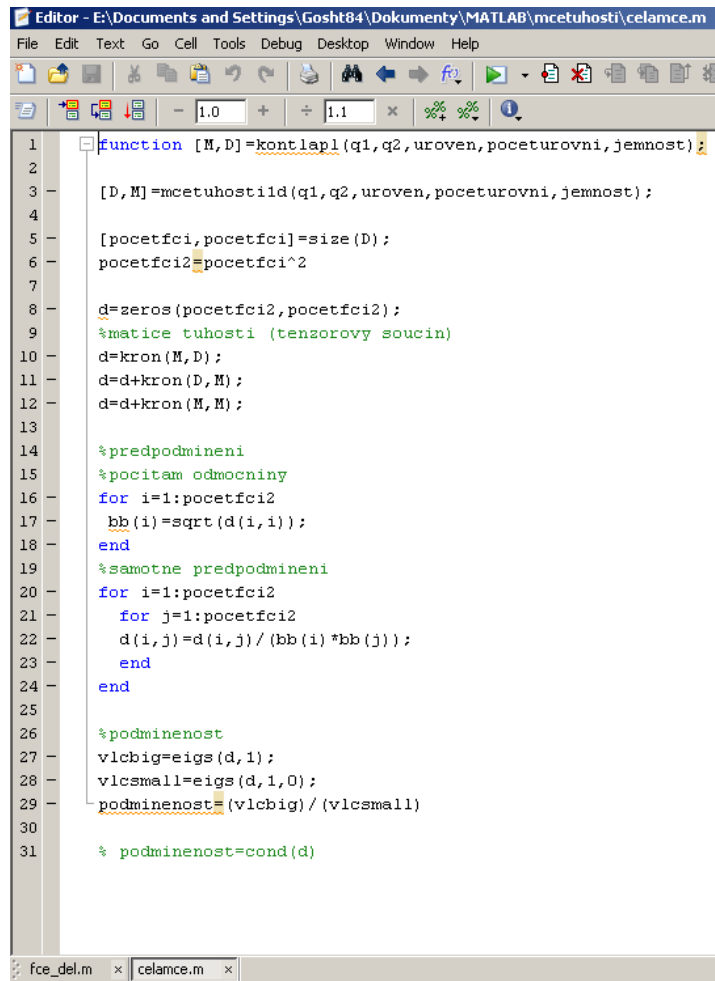
```
[prom1,prom2,...]=navez_funkce[par1,par2,...] .
```

Každý uživatel má možnost vytvořit si vlastní funkci a tu využívat. Každá funkce musí splňovat určité náležitosti. První řádek souboru, který tvoří funkci, obsahuje tzv. hlavičku. Tato hlavička má formální tvar

```
function [out1,out2,...]=jmeno_funkce(inp1,inp2,...).
```

Hlavička musí začínat klíčovým slovem **function**. Poté následuje seznam výstupních parametrů. Jestliže je těchto parametrů více než jeden, je nutné je dát do hranatých závorek a oddělit je čárkami. Poté následuje rovnítko a jméno funkce, které by se mělo shodovat s názvem m-funkce. Za jménem funkce jsou uvedeny výstupní parametry (v kulatých závorkách). Na dalších řádcích je obvykle popis (komentář) funkce. Tyto řádky musí začínat znakem %. Tento text se vypíše po vyvolání nápovědy k této funkci. Po komentáři již následuje tělo funkce (vlastní příkazy). V příkazech se mohou užívat jména vstupních parametrů – při výpočtu se použijí hodnoty, které byly na místě vstupního parametru při volání funkce zapsány. Dále je možné používat lokální proměnné libovolného jména. Jestliže uložíme hodnotu do výstupního parametru, bude tato hodnota přenesena do proměnné, která byla použita na místě výstupního parametru při volání funkce. Funkce může obsahovat také built-in funkce nebo m-funkce dříve vytvořené. Příkazy a funkce, které je možno využít k tvorbě funkcí, jsou v adresáři matlab\lang a jejich seznam lze vypsát pomocí příkazu `help lang`.

Ukázka, jak může vypadat uživatelsky vytvořená funkce, je na následujícím obrázku.



```
1 function [M,D]=kontlap1(q1,q2,uroven,poceturovni,jemnost);
2
3 [D,M]=mcetuhosti1d(q1,q2,uroven,poceturovni,jemnost);
4
5 [pocetfci,pocetfci]=size(D);
6 pocetfci2=pocetfci^2;
7
8 d=zeros(pocetfci2,pocetfci2);
9 %matice tuhosti (tenzorovy soucin)
10 d=kron(M,D);
11 d=d+kron(D,M);
12 d=d+kron(M,M);
13
14 %predpodmineni
15 %pocitam odmocniny
16 for i=1:pocetfci2
17     bb(i)=sqrt(d(i,i));
18 end
19 %samotne predpodmineni
20 for i=1:pocetfci2
21     for j=1:pocetfci2
22         d(i,j)=d(i,j)/(bb(i)*bb(j));
23     end
24 end
25
26 %podminenost
27 vlcbig=eigs(d,1);
28 vlcsmall=eigs(d,1,0);
29 podminenost=(vlcbig)/(vlcsmall);
30
31 % podminenost=cond(d)
```

Obr. 10: Ukázka vytvořené funkce

3.6 Funkce feval

Považuji za nutné se na tomto místě zmínit o funkci `feval`, poněvadž bude využita v algoritmu pro výpočet určitého integrálu Gaussovou metodou. Tato funkce se používá v případě, kdy je nutné zavolat existující funkci, ale není dopředu známé, jak se volaná funkce bude jmenovat. Nejčastějším využitím je vytváření univerzálního nástroje, např. pro numerickou integraci, pro výpočet kořenů funkcí (polynomů), pro řešení diferenciálních rovnic s nenulovou pravou stranou.

Základní syntaxe je následující:

```
>>[o1,o2,...,on]=feval(f,i,i2,...,in),
```

kde f je funkce popsána ve vlastním m-souboru, i_1, \dots, i_n jsou vstupní hodnoty a o_1, \dots, o_n hodnoty výstupní.

3.7 Funkce pro numerickou kvadraturu

MATLAB podporuje numerickou integraci funkce jedné proměnné, tedy

výpočet určitého integrálu $I = \int_a^b f(x) dx$ dvěma built-in funkcemi `quad` a

`quadl`¹². Funkce `quad` používá adaptivní Simpsonovo pravidlo a funkce `quadl` pracuje na základě Lobattova pravidla¹³. Základní syntaxe obou funkcí je následující

`hodnota_integ=quad('navez_funkce',dolni_mez,horni_mez),`

kde `navez_funkce` je jméno m-funkce, ve které je definována integrovaná funkce. Zadává se jméno m-funkce v apostrofech nebo jméno proměnné typu řetězec obsahující název m-funkce. Další dva parametry určují integrační meze.

Praktické využití ukážeme na následujícím příkladu.

Příklad 3.7

Zadání: Spočtete délku křivky zadané parametricky

$$x=t \sin(t) \quad y=t \cos(t) \quad z=t \quad t \in \langle 0, 10\pi \rangle .$$

Řešení: Pro výpočet parametricky zadané křivky využijeme vzorec

$$l = \int_{t_1}^{t_2} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt .$$

¹² funkce `quadl` nahradila používanou funkci `quad8`, tato funkce je stále podporována. Funkce `quad8` používá adaptivní Newton-Cotesovo pravidlo.

¹³ Kvadrurní vzorec: $\int_{-1}^1 f(x) dx = \frac{2}{n(n-1)}(f(-1)+f(1)) + \sum_{j=1}^{n-2} H_j f(a_j) + E$.

Odvození viz [1], str. 128.

Pro naši konkrétní křivku tedy dostáváme

$$l = \int_0^{10\pi} \sqrt{(\sin(t) + t \cos(t))^2 + (\cos(t) - t \sin(t))^2 + 1} dt .$$

K vyřešení je nutné zapsat funkci, jejíž integrál hledáme, formou m-funkce např. s názvem *fce_del*. Tento soubor může vypadat následovně:

```
function y=fce_del(x)
% Funkce pro ukazkovy priklad
% y=sqrt((sin(x)+x*cos(x))^2+(cos(x)-
% -x*sin(x))^2+1)
y=sqrt((sin(x)+x.*cos(x)).^2+(cos(x)-
-x.*sin(x)).^2+1);.
```

Výpočet integrálu pomocí funkce *quad* vypadá takto

```
>> del_1=quad('fce_del',0,100*pi)
del_1 = 4.935461848180390e+004.
```

Jestliže použijeme funkci *quadl*, dostáváme

```
>> del_2=quadl('fce_del',0,100*pi)
del_2 = 4.935461848164194e+004.
```

3.8 Gaussova kvadratura v MATLABu

Tato kvadrurní metoda není v programu MATLAB implementována, proto jsem se rozhodl, že vytvořím funkci, která bude vracet hodnotu určitého integrálu na základě Gaussova kvadrurního vzorce s uzly danými kořeny Legendrových polynomů. Tato funkce je uložena v souboru, který jsem pojmenoval *Gauss.m*. Podívejme se na vlastní strukturu této funkce.

Funkce začíná hlavičkou, která definuje volání funkce *Gauss*. Jsou zde uvedeny vstupní parametry, které požaduje funkce ke správnému běhu a nechybí implicitní tvar výstupu.

```
function I=Gauss(f,a,b,n).
```

Parametr *f* je ve významu funkce, která je definována ve vlastním m-souboru. Tento parametr je zadáván jako jméno m-souboru v apostrofech. Parametr *a* označuje dolní mez integračního intervalu, *b* horní mez a *n* určuje stupeň Legendrova polynomu.

V další části souboru je uveden komentář, který slouží jako nápověda. Tento komentář lze vyvolat příkazem

```
>> help Gauss.
```

Po komentáři následuje vlastní zdrojový kód funkce. Jako první jsou uvedeny matice *a_j* (kořeny Legendrových polynomů stupně *n*) a *H_j* (příslušné koeficienty). Stupeň *n* je v rozmezí 2-5.


```

aj=[-0.5773502692 -0.7745966692 -0.8611363116 -0.9061798459;
    0.5773502692 0.0000000000 -0.3399810436 -0.5384693101;
    0.0000000000 0.7745966692 0.3399810436 0.0000000000;
    0.0000000000 0.0000000000 0.8611363116 -0.5384693101;
    0.0000000000 0.0000000000 0.0000000000 -0.9061798459]

```

```

Hj=[1.0000000000 0.5555555556 0.3478548451 0.2369268850;
    1.0000000000 0.8888888889 0.6521451549 0.4786286705;
    0.0000000000 0.5555555556 0.6521451549 0.5688888889;
    0.0000000000 0.0000000000 0.3478548451 0.4786286705;
    0.0000000000 0.0000000000 0.0000000000 0.2369268850]

```

Dále jsou v kódu uvedeny podmínky pro vstupní parametr n :

```

if n>5
    disp('n musi byt v rozsahu 2-5');
    return
end

if n<2
    disp('n musi byt v rozsahu 2-5')
    return
end

```

V další části programu již následuje samotný výpočet hodnoty integrálu. Nejdříve je nutné užít substituci

$$x = \frac{(b-a)t + a + b}{2} ,$$

která převede interval $\langle a, b \rangle$ na $\langle -1, 1 \rangle$. Po zderivování dostáváme předpis

$$dx = \frac{(b-a)}{2} dt .$$

Hodnota t je určena z matice aj podle parametru n .

Substituce vypadá následovně:

```
t(1:n)=0.5*((b-a).*aj(1:n,n-1)+b+a);
```

Dalším krokem je výpočet funkčních hodnot pomocí funkce `feval`, výsledné hodnoty jsou uloženy do pomocné proměnné `y`, tedy

```
y=feval(f,t).
```

Poté následuje vyhledání příslušných koeficientů a transpozice kvůli možnosti násobení s `y`,

```
kon(1:n)=Hj(1:n,n-1)
```

```
kon_ok=kon';.
```

Nyní vynásobíme vektor funkčních hodnot `y` s vektorem koeficientů `kon_ok`.

```
int=y*kon_ok;
```

Poslední částí výpočtu je sečtení hodnot ve vektoru `int` a poté vynásobení výrazem

$\frac{(b-a)}{2}$, který jsme získali substitucí a nakonec přiřazení výsledné hodnoty

výstupnímu parametru `I`.

```
I=(sum(int))*(b-a)/2.
```

Nyní ještě vykreslíme graf funkce na intervalu $\langle a, b \rangle$ a zvýrazníme funkční hodnoty dané kořeny Legendrových polynomů.

```
g=a:0.1:b;
```

```
y2=feval(f,g);
```

```
figure(1);
```

```
plot(t,y,'*')
```

```
hold on
```

```
plot(g,y2,'r')
```

Kompletní zdrojový kód funkce Gauss je uveden v příloze č. 2.

3.9 Porovnání metod na příkladech

V této kapitole porovnáme hodnoty integrálů. Na příkladech porovnáme výsledné hodnoty všech výše uvedených metod, tedy využijeme metodu obdélníkovou, lichoběžníkovou. Pro tyto metody jsou vytvořeny vlastní funkce (*obde.m*, *lich.m*)¹⁴ Následně bude integrál počítán built-in funkcemi `quad` (Simpsonova metoda) a `quadl` (Lobattovo pravidlo) a na závěr funkcí `Gauss`. Každou funkci $f(x)$ je nutné zapsat do samostatného m-souboru, který slouží jako jeden ze vstupů u obou metod výpočtu. Jméno m-souboru je totožné s pořadím příkladu (*priklad1.m*, *priklad2.m*, apod.)¹⁵.

Příklad 3.9.1

Zadání: Určete plošný obsah obrazce ohraničený křivkami

$$y = \frac{1}{x}, \quad y = 0, \quad x = 1, \quad x = 3.$$

Řešení: Jedná se o výpočet určitého integrálu $\int_1^3 \frac{dx}{x}$. Analytickým výpočtem dojdeme k hodnotě $\ln 3$, což je přibližně 1,098612.

Nejprve spočteme integrál obdélníkovou metodou. Zvolme $n = 50$. Příkaz pro výpočet je ve tvaru

```
pr1_obd=obdel(1,3,50,'priklad1'),
```

a dostáváme výslednou hodnotu

```
pr1_obd =1.125397.
```

Nyní spočteme hodnotu lichoběžníkovým pravidlem:

```
pr1_lich=lich(1,3,50,'priklad1').
```

¹⁴ viz příloha č. 3, č. 4

¹⁵ Definice jednotlivých funkcí viz příloha č. 5

Dostáváme výsledek

```
pr1_lich=1.098730.
```

Built-in funkce `quad`, tedy příkaz

```
pr1_quad=quad('prikklad1',1,3),
```

nám vrací hodnotu

```
pr1_quad=1.098612.
```

Druhá built-in funkce, příkaz

```
pr1_quadl=quadl('prikklad1',1,3),
```

dává výsledek

```
pr1_quadl=1.098613.
```

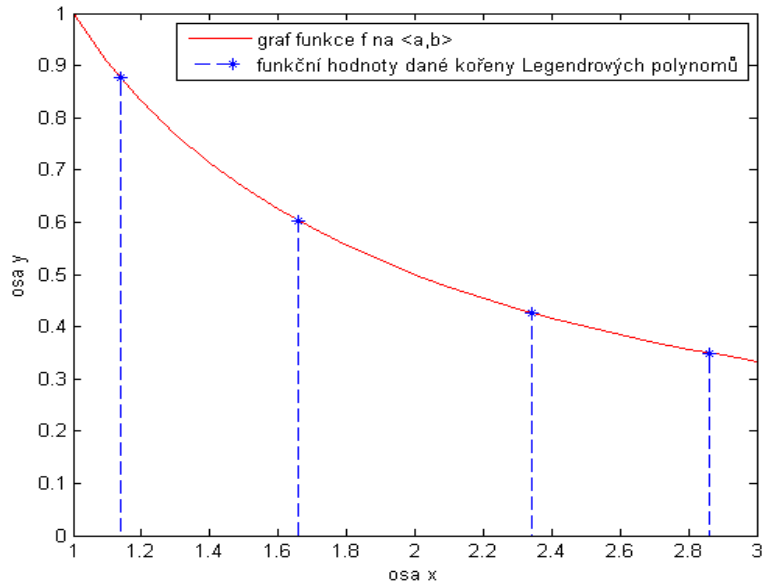
Jako poslední spočteme hodnotu daného integrálu pomocí funkce `Gauss`.

Vstupní hodnotu n zvolme např. 4. Příkaz v *Command Window* vypadá následovně:

```
pr1_Ga=Gauss('prikklad1',1,3,4).
```

Výstupní proměnná má hodnotu

```
pr1_Ga=1.098039.
```



Obr. 11: Graf funkce $f(x) = \frac{1}{x}$, $x \in \langle a, b \rangle$ a funkční hodnoty dané kořeny Legendrových polynomů

Závěr: Nejpřesnější hodnotu nám dává funkce `quad`, tedy Simpsonova metoda. Hodnota, která se nejvíce liší od hodnoty přesné, byla vypočtena metodou obdélníkovou.

Příklad 3.9.2

Zadání: Vypočtete plochu ohraničenou křivkami

$$y = e^{x^2}, \quad y = 0, \quad x = 1, \quad x = 2.$$

Řešení: Výsledkem je hodnota určitého integrálu $\int_1^2 e^{x^2}$. Tento integrál

nelze spočítat jednoduchými analytickými metodami, a proto jej musíme počítat metodami numerickými.

Nejprve opět využijeme metodu obdélníkovou, zvolme $n = 60$.

$$\text{pr2_obd} = \text{obdel}(1, 2, 60, \text{'prikklad2'}).$$

Výsledná hodnota je ve tvaru

```
pr2_obd=15.472542.
```

Druhou metodou je metoda lichoběžníková, tedy

```
pr2_lich=lich(1,2,60,'priklad2').
```

Dostáváme výsledek

```
pr2_lich=14.994905
```

Jako třetí řešení využijeme built-in funkci quad

```
pr2_quad=quad('priklad2',1,2),
```

a dostáváme hodnotu

```
pr2_quad=14.989976.
```

Dalším způsobem řešení je funkce quadl, tedy příkaz

```
pr2_quadl=quadl('priklad2',1,2),
```

v tomto případě dostáváme

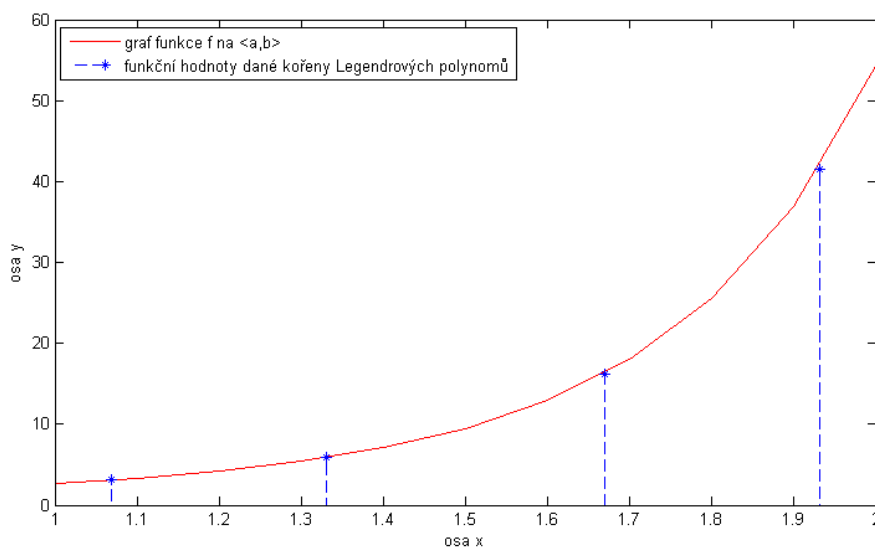
```
pr2_quadl=14.989976.
```

Poslední metodou výpočtu je funkce Gauss, zvolme $n=5$,

```
pr2_Ga=Gauss('priklad2',1,2,4).
```

Výsledná hodnota je

```
pr2_Ga=14.988996.
```



Obr 12: Graf funkce $f(x)=e^{x^2}$ na $\langle a,b \rangle$ a funkční hodnoty dané kořeny Legendrových polynomů

Závěr: Největší rozdíl je patrný u obdélníkové metody. U built-in funkcí můžeme pozorovat rozdíl až na osmém desetinném místě. Nejpřesnější řešení tedy získáme užitím těchto funkcí. Výsledná hodnota funkce Gauss se liší již na třetím desetinném místě, což považuji za poměrně značnou nepřesnost.

Příklad 3.9.3

Zadání: Ověřte řád přesnosti Gaussova kvadrurního vzorce pro $n=2$.

Řešení: Jak bylo uvedeno, Gaussův vzorec pro n uzlů je přesný pro polynomy stupně nejvýše $2n-1$. Jestliže bude polynom stupně vyššího, vzniká nepřesnost. Zvolme $n=2$. Z tohoto předpokladu vyplývá, že polynom může být maximálně stupně 3. Libovolný polynom je např. ve tvaru

$$f(x)=x^3+5.$$

Tuto funkci budeme integrovat na intervalu $\langle 1,2 \rangle$ pomocí Gaussova kvadrurního vzorce, tedy

```
>>pol=Gauss('poly',1,2,3),
```

kde `poly` je vlastní m-soubor, ve kterém je funkce definována (viz příloha č. 6).

Výsledná hodnota je

```
pol=8.75.
```

Což je i hodnota, kterou získáme analytickým výpočtem. Nyní zvýšíme stupeň polynomu na $2n$. Výsledná hodnota by se již měla lišit od hodnoty získané analytickými metodami. Polynom má nyní např. tvar

$$f(x)=x^4+5.$$

Integrační interval zachováme. Příkaz zadaný do MATLABu tedy vypadá

```
>>poly2=Gauss('poly2',1,2,2),
```

`poly2` je opět samostatný m-soubor (viz příloha č. 6). Výsledná hodnota je

```
poly2=11.19.
```

Analytickým výpočtem dospějeme k hodnotě 11,2. Chyba Gaussova kvadrurního vzorce pro polynom $f(x)=x^4+5$ je tedy 0,01.

3.10 Další řešené příklady

V následujících tabulkách je vyřešeno několik dalších příkladů. Jsou vypočteny přibližné hodnoty určitých integrálů pomocí metody obdélníkové a lichoběžníkové. U těchto metod je nejprve voleno $n=4$ (tabulka 9) a poté $n=60$ (tabulka 10). Dále je integrál počítán built-in funkcí `quad` a `quadl` a pomocí funkce `Gauss`. V tomto případě je zvoleno nejprve $n=2$ a poté $n=4$.

$\int_a^b f(x) dx$	obdélníková metoda	lichoběžníková metoda	built-in quad	built-in quadl	Gaussova metoda	přesná hodnota
$\int_0^3 \sqrt{9-x^2} dx$	7,86535	6,74035	7,06858	7,06858	7,16502	7,06858
$\int_1^{1,2} e^x dx$	0,75292	0,60196	0,60184	0,60184	0,60183	0,60184
$\int_4^9 \frac{\sqrt{x}}{\sqrt{x-1}} dx$	10,60045	8,41295	8,38629	8,38630	8,37693	8,38629
$\int_0^{\frac{\pi}{2}} x^2 \sin(x) dx$	1,66695	1,18248	1,14159	1,14159	1,16779	1,14159
$\int_0^1 3^{2x-7} dx$	0,00228	0,00171	0,00167	0,00167	0,00166	0,00166

Tabulka 9

$\int_a^b f(x) dx$	obdélníková metoda	lichoběžníková metoda	built-in quad	built-in quadl	Gaussova metoda	přesná hodnota
$\int_0^3 \sqrt{9-x^2} dx$	7,13789	7,06289	7,06858	7,06858	7,08352	7,06858
$\int_1^{1,2} e^x dx$	0,61189	0,60184	0,60184	0,60184	0,60184	0,60184
$\int_4^9 \frac{\sqrt{x}}{\sqrt{x-1}} dx$	8,53225	8,38642	8,38629	8,38630	8,38626	8,38629
$\int_0^{\frac{\pi}{2}} x^2 \sin(x) dx$	1,17407	1,14177	1,14159	1,14159	1,14159	1,14159
$\int_0^1 3^{2x-7} dx$	0,00170	0,00167	0,00167	0,00167	0,00166	0,00166

Tabulka 10

Podle získaných hodnot lze usoudit, že nejpřesnější hodnoty nám dávají funkce quad a quadl. Výsledky z quad a quadl se ve většině případů navzájem liší až na osmém či devátém desetinném místě. Tyto metody vracejí hodnoty, které se minimálně liší od hodnot přesných (vypočtených analytickým způsobem).

V následujících tabulkách jsou uvedeny relativní chyby pro dosažené výsledky. Nejdříve jsou počítány chyby pro hodnoty v tabulce 9 a poté pro hodnoty v tabulce 10. Pro relativní chybu platí vztah

$$r_x = \left| \frac{e_x}{x} \right|,$$

kde x značí přibližnou hodnotu integrálu a $e_x = \hat{x} - x$, \hat{x} značí přesnou hodnotu integrálu. Rozdíl $e_x = \hat{x} - x$ nazýváme absolutní chybou. Chyba je uvedena v procentech.

$\int_a^b f(x) dx$	chyba obdélníkové metody	chyba lichoběžníkové metody	chyba built-in quad	chyba built-in quadl	chyba Gaussovy metody
$\int_0^3 \sqrt{9-x^2} dx$	10,13003	4,86976	0,00005	0,00005	1,34594
$\int_1^{1,2} e^x dx$	20,06588	0,01993	0,00083	0,00083	0,00166
$\int_4^9 \frac{\sqrt{x}}{\sqrt{x}-1} dx$	20,88742	0,31689	0,00005	0,00006	0,11174
$\int_0^{\frac{\pi}{2}} x^2 \sin(x) dx$	31,51608	3,45776	0,00003	0,00003	2,24333
$\int_0^1 3^{2x-7} dx$	26,98167	2,64222	0,01093	0,01093	0,29024

Tabulka 11

$\int_a^b f(x) dx$	chyba obdélníkové metody	chyba lichoběžníkové metody	chyba built-in quad	chyba built-in quadl	chyba Gaussovy metody
$\int_0^3 \sqrt{9-x^2} dx$	0,97097	0,08061	0,00005	0,00005	0,21086
$\int_1^{1,2} e^x dx$	1,64245	0,00083	0,00083	0,00083	0,00083
$\int_4^9 \frac{\sqrt{x}}{\sqrt{x}-1} dx$	1,71066	0,00149	0,00005	0,00006	0,00033
$\int_0^{\frac{\pi}{2}} x^2 \sin(x) dx$	2,76622	0,01571	0,00003	0,00003	0,00012
$\int_0^1 3^{2x-7} dx$	2,24205	0,01093	0,01093	0,01093	0,04916

Tabulka 12

Výsledky potvrzují, že velmi záleží na volbě parametru n . Jestliže zvolíme u obdélníkové či lichoběžníkové metody malé n , obdržíme značně nepřesný výsledek. Naopak, zvolíme-li velké n , výsledek bude přesnější. Volba parametru n u funkce Gauss, tedy stupně Legendrova polynomu, také ovlivňuje výslednou hodnotu integrálu. Budeme-li opět pracovat s vyššími n , bude chyba kvadraturní metody menší.

4 Závěr

V teoretické části své diplomové práce se zabývám teorií určitého integrálu a způsobu jeho výpočtu. Jsou zde uvedeny jak metody obecně známé, tak metody, které vyžadují hlubší znalosti. Mezi základní způsoby, které se vyučují v prvních ročnících vysoké školy, bych zařadil metodu obdélníkovou, lichoběžníkovou a metodu Simpsonovu. Volba kvadraturního vzorce není snadnou záležitostí a nejsou stanovena přesná pravidla. Ve většině případů však vystačí lichoběžníkové pravidlo nebo Simpsonův vzorec. Ke správné volbě je nutné mít již nějakou předchozí zkušenost.

Praktická část diplomové práce nejdříve seznamuje s matematickým softwarem MATLAB a jeho základním užíváním. Poté je na příkladu ukázáno, jakým způsobem lze v MATLABu určit hodnotu určitého integrálu. Software obsahuje dvě vestavěné funkce pro výpočet určitého integrálu a to funkce `quad`, která pracuje na základě Simpsonova vzorce, a `quadl`, která využívá Lobattovu metodu. Autoři doporučují využívat druhou jmenovanou funkci z důvodu větší přesnosti výsledků. Díky tomu, že MATLAB umožňuje vytvořit vlastní funkce, mohl jsem naprogramovat Gaussovu metodu, která je v textu popsána. V další části jsou na praktických příkladech srovnány všechny uvedené metody a výsledky, které s jejich pomocí získáme. Jako nejvhodnější a nejpřesnější se ukázaly vestavěné funkce, tedy určování hodnoty určitého integrálu Simpsonovým a Lobattovým pravidlem. Využití softwaru MATLAB při výuce nebo při samostudiu této problematiky bych doporučil, jeho použití umožní snadnější pochopení látky.

5 Použitá literatura

- [1] RALSTON, A.: *Základy numerické matematiky*. 2. vydání. Academia. Praha 1978. ISBN 104 – 21 – 852
- [2] REKTORYS, K. a kol.: *Přehled užití matematiky*. 1. vydání. SNTL. Praha 1963. ISBN 04 – 018 – 63
- [3] NEKVINDA, M. – ŠRUBAŘ, J. – VILD, J.: *Numerická matematika*. 1. vydání. VŠST v Liberci. Liberec 1976. ISBN neuvedeno
- [4] NEKVINDA, M. – ŠRUBAŘ, J. – VILD, J.: *Úvod do numerické matematiky*. 1. vydání. SNTL. Praha 1976. ISBN neuvedeno
- [5] JARNÍK, V.: *Integrální počet 1*. 6. vydání. Academia. Praha 1984. ISBN 101 – 21 – 852
- [6] BITTNEROVÁ, D. - PLAČKOVÁ, G.: *Louškáček 2*. 1. vydání. TUL. Liberec 2008. ISBN 978 – 80 – 7372 – 332 – 3
- [7] DUŠEK, F.: *Matlab a Simulink, úvod do používání*. 1. vydání. Univerzita Pardubice. Pardubice 2000. ISBN 80 – 7194 – 273 – 1
- [8] DUŠEK, F. - HONC, D.: *Matlab a Simulink, úvod do používání*. 1. vydání. Univerzita Pardubice. Pardubice 2005. ISBN 80 – 7194 – 776 – 8
- [9] HANSELMAN, D. - LITTLEFIELD, B.: *Mastering MATLAB 6 (A Comprehensive Tutorial and Reference)*. Prentice Hall. New Jersey 2001, anglicky
- [10] The MathWorks - MATLAB and Simulink for Technical Computing [online]. 1994 [cit. 2009-04-20]. Dostupný z WWW: <http://www.mathworks.com/>, anglicky

6 Seznam příloh

1. Příloha č. 1 – soubor fce_del.m
2. Příloha č. 2 – soubor Gauss.m
3. Příloha č. 3 – soubor obdel.m
4. Příloha č. 4 – soubor lichom.m
5. Příloha č. 5 – soubory prik1.m, prik2.m, prik3.m, prik4.m, prik5.m, prik6.m, prik7.m
6. Příloha č. 6 – soubory poly.m, poly2.m

Příloha č. 1 – soubor fce_del.m

```
function y=fce_del(x)
% Funkce pro ukazkovy priklad
% y=sqrt((sin(x)+x*cos(x))^2+(cos(x)-x*sin(x))^2+1)
y=sqrt((sin(x)+x.*cos(x)).^2+(cos(x)-x.*sin(x)).^2+1);
```

Příloha č. 2 – soubor Gauss.m

```
function I=Gauss(f,a,b,n)
%Funkce Gauss pocita hodnotu urciteho integralu na
%intervalu <a,b> pomoci Gaussovy kvadraturni metody
%s uzly danymi koreny Legendrovych polynomu.
%
%**Ukazka zadani**
%
% I=Gauss(f,a,b,n)
% f... funkce ulozena ve vlastnim m-file
% a... dolni mez intervalu
% b... horni mez intervalu
% n... kladne cele cislo urcujici stupen Legendrova
% polynomu (n=2,3,4,5)
%*****

%           n=2           n=3           n=4           n=5
aj=[-0.5773502692  -0.7745966692  -0.8611363116  -0.9061798459;
    0.5773502692   0.0000000000  -0.3399810436  -0.5384693101;
    0.0000000000   0.7745966692   0.3399810436   0.0000000000;
    0.0000000000   0.0000000000   0.8611363116  -0.5384693101;
    0.0000000000   0.0000000000   0.0000000000  -0.9061798459]

Hj=[1.0000000000  0.5555555556   0.3478548451   0.2369268850;
    1.0000000000  0.8888888889   0.6521451549   0.4786286705;
    0.0000000000  0.5555555556   0.6521451549   0.5688888889;
    0.0000000000  0.0000000000   0.3478548451   0.4786286705;
    0.0000000000  0.0000000000   0.0000000000   0.2369268850]

if n>5
    disp('n musi byt v rozsahu 2-5')
    return
end
if n<2
    disp('n musi byt v rozsahu 2-5')
    return
end
```



```

%substituce prevadejici interval <a,b> na <-1,1> ...
%... t={ (b-a)x+b+a}/2
t(1:n)=0.5*((b-a).*aj(1:n,n-1)+b+a);

%urceni funkcnich hodnot
y=feval(f,t)

%vyber koeficientu podle daneho n
kon(1:n)=Hj(1:n,n-1);
kon_ok=kon';

%soucín dane funkční hodnoty a příslušného koeficientu
int=y*kon_ok;
format long

%dokonceni vypoctu; {(b-a)/2}*dt=dx
I=(sum(int))*(b-a)/2

%graf funkce f na intervalu <a,b>
g=a:0.1:b;
y2=feval(f,g);
figure(1);
plot(t,y,'*') %f(t) dane koreny Legendrovych polynomu
hold on
plot(g,y2,'r')

```

Příloha č. 3 – soubor obdel.m

```
function I=obdel(a,b,n,f)
% obdel - funkce spocita hodnotu urciteho integralu
% pomoci obdelnikoveho pravidla
%
%** I=obdel(a,b,n,f)**
% a... dolni mez
% b... horni mez
% n... pocet podintervalu, na ktere je <a,b> rozdeleno
% f... nazev existujici funkce (tj. je ulozena
% v m-souboru)
% I... priblizna hodnota integralu
% priklad: I=obdel(0,pi,100,'sin')
%*****

if a>=b
    disp('Chybne zadane meze!')
end
if n<1
    disp('n musi byt kladny!')
end

%krok h
h = (b-a)/n;

%samotny vypocet
I = 0;
for i=a:h:b
    y = feval(f,i);
    I = I + h*y;
end
```

Příloha č. 4 – soubor lichom.m

```
function I=lichom(a,b,n,f)
%lichom - funkce spočítá hodnotu určitého integrálu
pomocí lichobeznickového pravidla
%
%**I=lichom(a,b,n,f)**
% a... dolní mez
% b... horní mez
% n... počet podintervalů, na které je <a,b> rozděleno
% f... název existující funkce (tj. je uložena
% v m-souboru)
% I... přibližná hodnota integrálu
% příklad: I=lichom(0,pi,100,'sin')
%*****

if a>=b
    disp('Chybně zadane meze!')
end
if n<1
    disp('n musí být kladné!')
end

%krok h
h=(b-a)/n;
h2=(b-a)/(2*n);

%funkční hodnota v uzlech a,b
Qa=feval(f,a);
Qb=feval(f,b);

%funkční hodnoty v uzlech x_i
for i=1:n-1
    x(i)=a+h*i;
    fce_hod(i)=feval(f,x(i));
    soucet=sum(fce_hod);
end

%dokonceni vypočtu
I=h2*(Qa+2*soucet+Qb)
```

Příloha č. 5

příklad1.m

```
function y=příklad1(x)
%Prakticky příklad číslo 1
%y=x^(-1)
y=x.^(-1);
```

příklad2.m

```
function y=příklad2(x)
%Prakticky příklad číslo 2
%y=e^{x^2}
y=exp(x.^2);
```

příklad3.m

```
function y=příklad3(x)
%Prakticky příklad číslo 3
%y=sqrt(9-x^2)
y=sqrt(9-x.^2);
```

příklad4.m

```
function y=příklad4(x)
%Prakticky příklad číslo 4
%y=e^x
y=exp(x);
```

příklad5.m

```
function y=příklad5(x)
%Prakticky příklad číslo 5
%y=sqrt(x)/{sqrt(x)-1}
y=sqrt(x).*(sqrt(x)-1).^(-1);
```

příklad6.m

```
function y=příklad6(x)
%Prakticky příklad číslo 6
%y=x^2*sin(x)
y=(x.^2).*sin(x);
```

příklad7.m

```
function y=příklad7(x)
%Prakticky příklad číslo 7
%y=3^(2x-7)
y=3.^(2.*x-7);
```

Příloha č. 6

poly.m

```
function y=poly(x)
%Ukazkovy priklad (rad kvadrurniho vzorce)
%y=x^3+5
y=x.^3+5;
```

poly2.m

```
function y=poly2(x)
%Ukazkovy priklad (rad kvadrurniho vzorce)
%y=x^4+5
y=x.^4+5;
```