

Vysoká škola báňská – Technická univerzita Ostrava
Západočeská univerzita v Plzni

NUMERICKÉ METODY I

(obrazovková verze)

Lukáš Pospíšil, Vít Vondrák



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Obsah

1. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Lukáš Pospíšil, Vít Vondrák
Numerické metody 1

© Lukáš Pospíšil, Vít Vondrák, 2012
ISBN

Obsah

2. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Předmluva

Vážení čtenáři,

dostává se Vám do rukou učební text *Numerické metody 1*. Základ tohoto textu vznikl již před několika lety pro podporu výuky předmětu Numerické metody na Fakultě elektrotechniky a informatiky Vysoké školy báňské-Technické Univerzity Ostrava. Předmět byl zejména určen pro obor Výpočetní matematika. Tento předmět si stejně jako i učební text klade za cíl seznámit studenty se základními numerickými metodami, které se běžně používají v inženýrské praxi. A nejen tam. Se základními numerickými metodami se totiž můžete setkat i ve fyzice, chemii, elektrotechnice, informatice, environmentálních vědách a vůbec všude, kde je zapotřebí řešit základní matematické úlohy, jejichž analytické řešení buď není známo, nebo je jen velmi těžko řešitelné.

Celkový obsah učebního textu je možné rozdělit do tří základních oblastí. Úvod učebního textu je věnován problematice chyb v inženýrských výpočtech, jejich zdrojům a analýze. Druhou oblastí je oblast numerických metod pro řešení základních úloh lineární algebry. Zde se seznámíte s metodami řešení soustav lineárních rovnic, nebo s problémem hledání vlastních čísel. Do této oblasti byla taktéž zařazena problematika řešení nelineárních rovnic a jejich soustav, byť by mohla být zařazena i do oblasti matematické analýzy. Třetí oblastí je oblast numerických metod používaných pro řešení úloh vyskytujících se v matematické analýze. Mezi tyto úlohy se řadí interpolace a aproximace funkcí, numerická derivace a integrování či řešení počátečních úloh pro obyčejné diferenciální rovnice.

Pro pochopení učebního textu je proto potřeba základních znalostí z výše uvedené problematiky. Na Fakultě elektrotechniky a informatiky jsou tyto prerekvizity zajištěny následujícími předměty: Lineární algebra, Matematická analýza 1 a 2. Důležitá je taktéž znalost základů algoritmizace a programování, neboť popis jednotlivých numerických metod je orientován na jejich praktické použití na počítači. Odvození a popis metod je však doplněn i konvergenční teorií a odhady chyb. Tyto poznatky jsou totiž nezbytné pro správnou volbu numerické metody při řešení konkrétních úloh.

V Ostravě 20. 4. 2012

Vít Vondrák a Lukáš Pospíšil

Obsah

4. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Obsah

Předmluva	3
1 Úvod do numerických metod	10
1.1 Matematické řešení inženýrských problémů	11
1.2 Korektnost úloh a stabilita výpočetních postupů	16
1.3 Typy chyb	23
1.3.1 Aproximace čísla	24
1.3.2 Šíření chyb při aritmetických operacích	26
1.3.3 Obecný vzorec chyb	29
1.3.4 Obrácená úloha	31
1.4 Aproximace čísel na počítači	33
1.5 Test	38
2 Numerické řešení nelineárních rovnic	46
2.1 Separace kořenů	47
2.2 Metoda půlení intervalu	54
2.2.1 Konvergence	57
2.3 Metoda prostých iterací	64
2.4 Newtonova metoda	73



Obsah

5. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2.4.1	První modifikace Newtonovy metody	77
2.4.2	Druhá modifikace Newtonovy metody	78
2.4.3	Newtonova metoda pro soustavy	79
2.5	Test	84
3	Iterační řešení soustav lineárních rovnic	92
3.1	Normy vektorů a matic	94
3.2	Lineární metody	102
3.2.1	Jacobiova metoda	106
3.2.2	Gaussova-Seidelova metoda	106
3.2.3	Successive over-relaxation method (SOR)	107
3.3	Gradientní metody	116
3.3.1	Metoda největšího spádu	119
3.3.2	Metoda sdružených gradientů	128
3.3.2.1	Volba sdružených směrů	128
3.3.2.2	Gramův-Schmidtův A -ortogonalizační proces	130
3.3.2.3	Algoritmus CG a jeho vlastnosti	133
3.3.3	Předpodmínění	136
3.3.3.1	Jacobiho diagonální předpodmínění	140
3.3.3.2	Symmetric successive over-relaxation method (SSOR)	140
3.3.3.3	Neúplná Choleského faktorizace	141
3.4	Test	146
4	Výpočet vlastních čísel a vektorů matic	157
4.1	Výpočet charakteristického polynomu	158
4.1.1	Geršgorinova věta	160

4.1.2	Krylovova metoda	163
4.1.3	Výpočet charakteristického polynomu třídiagonální matice	168
4.2	Výpočet dominantního vlastního čísla	171
4.2.1	Mocninná metoda	172
4.2.2	Posun a redukce spektra	179
4.3	Metody založené na podobnosti matic	182
4.3.1	Algoritmus QR	184
4.3.2	Givensova metoda	194
4.3.3	Householderova metoda	203
4.3.4	Lanczosova metoda	207
4.4	Test	212
5	Interpolace	221
5.1	Interpolace polynomem	224
5.2	Lagrangeův interpolační polynom	231
5.3	Newtonův interpolační polynom	239
5.3.1	Ekvidistantní uzly interpolace	246
5.4	Trigonometrická interpolace	252
5.5	Interpolace lineárními spline funkcemi	260
5.6	Interpolace kubickými spline funkcemi	266
5.6.1	Konstrukce spline funkce	268
5.7	Test	278
6	Aproximace	293
6.1	Metoda nejmenších čtverců	300
6.2	Ortogonální systémy funkcí	311

6.3	Test	315
7	Numerická derivace a integrace	322
7.1	Numerická derivace	324
7.2	Newtonovy-Cotesovy vzorce	334
7.3	Gaussovy vzorce	343
7.4	Výpočet integrálů pomocí extrapolace	346
7.5	Výpočet nevlastních integrálů	347
7.6	Test	348
8	Řešení počátečních úloh pro obyčejné diferenciální rovnice	357
8.1	Jednokrokové metody	363
8.1.1	Eulerova metoda	364
8.1.2	Metody Runge-Kutta	371
8.1.2.1	Metody Runge-Kutta 2. řádu	372
8.1.2.2	Metoda Runge-Kutta 4. řádu	374
8.1.2.3	Implicitní metody	378
8.2	Vícekové metody	382
8.2.1	Metody založené na numerické integraci	385
8.2.1.1	Adams-Bashforthovy metody	387
8.2.1.2	Adams-Moultonovy metody	389
8.3	Soustavy diferenciálních rovnic a diferenciální rovnice vyšších řádů	391
8.3.1	Soustavy obyčejných diferenciálních rovnic	391
8.3.2	Diferenciální rovnice vyšších řádů	392
8.4	Test	397



Literatura

406

Rejstřík

409



Obsah

9. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Kapitola 1

Úvod do numerických metod

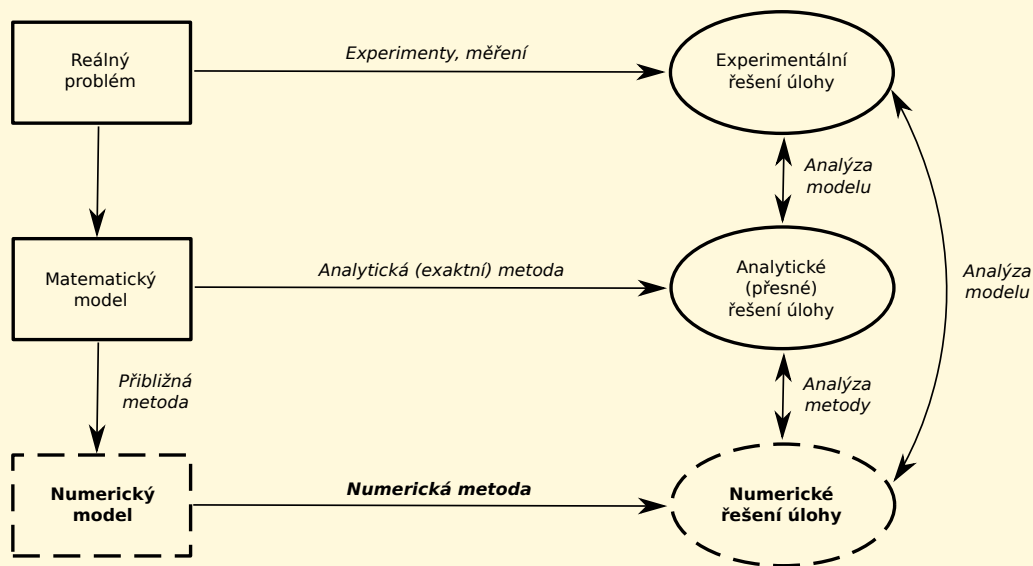
Průvodce studiem

V této kapitole si představíme základní principy při řešení inženýrských úloh. Důraz bude kladen na jejich numerické řešení, které připouští jistou míru nepřesnosti. Následovat proto bude obecná teorie chyb. Budou zavedeny základní pojmy jako jsou stabilita, podmíněnost matematických a numerických úloh, diskutovány budou typy a zdroje chyb. Speciální pozornost bude věnována šíření chyb v numerických výpočtech. Na závěr bude představena reprezentace reálných čísel na počítačích.

[Obsah](#)[10. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka/Okno](#)

1.1. Matematické řešení inženýrských problémů

Na obrázku 1.1 je zobrazeno standardní schéma řešení inženýrských úloh.



Obr. 1.1: Postup při řešení inženýrské úlohy

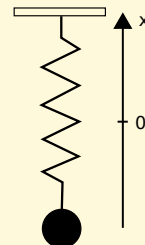
Příklad 1.1. Na následujícím vzorovém příkladě budeme demonstrovat postup řešení modelové inženýrské úlohy. Za tento modelový příklad jsme vybrali úlohu kmitání tělesa na pružině.

Reálný problém

Hmotné těleso je zavěšeno na pružině. Uvolnění tělesa v nerovnovážném stavu způsobí jeho kmitání ve svislém směru. Cílem je zjistit polohu tělesa v libovolně určeném časovém okamžiku. Pro zjišťování této polohy si zvolíme souřadný systém ve svislém směru jak je zobrazeno na obrázku. Počátek souřadného systému $x = 0$ reprezentuje rovnovážný stav.

Experimentální řešení problému

Sestrojení fyzikálního modelu a provedení experimentu. Měření je možné provádět například pomocí kamerového systému a pravítka.



Obsah

12. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Matematický model

Pomocí matematického aparátu je možné při zanedbání veličin, které nemají podstatný vliv na kmitání tělesa (odpor vzduchu, hmotnost pružiny atd) popsat úlohu pomocí lineární diferenciální rovnice druhého řádu s neznámou x popisující polohu tělesa v čase t . V této rovnici veličina m popisuje hmotnost tělesa na pružině, b reprezentuje buzení a k představuje tuhost pružiny. Pro jednoznačnost řešení úlohy musíme zvolit počáteční polohu x_0 a počáteční rychlost x_1 .

$$\begin{aligned} mx''(t) &= -bx'(t) - kx(t) \\ x(t_0) &= x_0, \quad x'(t_0) = x_1 \end{aligned}$$

$$m = 1\text{kg}, \quad b = 1\text{N}, \quad k = 2\text{N.m}^{-1}, \quad t_0 = 0\text{s}, \quad x_0 = -5\text{m}, \quad x_1 = 0\text{m.s}^{-1}$$

Analytické (přesné) řešení úlohy

Čtenáři, kteří jsou již seznámeni s řešením lineárních diferenciálních úloh, snadno odvodí, že funkce

$$x(t) = -5e^{-t} - 5te^{-t}$$

je řešením výše uvedeného matematického modelu. Ti, kteří se s problematikou řešení diferenciálních rovnic ještě neseznámili, mohou tak učinit ve skriptech Obyčejné diferenciální rovnice [3], případně stačí když do diferenciální rovnice funkci $x(t)$ dosadí a ověří tak její správnost.

[Obsah](#)[13. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka / Okno](#)

Přibližná metoda a numerický model

Pro sestavení numerického modelu, který budeme chtít řešit na kalkulačce, počítači či logaritmickém pravítku budeme muset nejdříve vyřešit problém spojitě závislosti polohy na čase. Ani jedno ze zmíněných zařízení nám totiž neumožní počítat s nekonečně mnoha údaji, které popisují polohy tělesa v libovolném časovém intervalu. Z tohoto důvodu se proto omezíme jen na několik vybraných časových okamžiků v nichž budeme hledat řešení dané úlohy. Této technice se říká *metoda časové diskretizace*. Pro naši modelovou úlohu si tedy zvolme 6 časových okamžiků

$$t_0 = 0\text{s}, t_1 = 0.2\text{s}, t_2 = 0.4\text{s}, t_3 = 0.6\text{s}, t_4 = 0.8\text{s}, t_5 = 1\text{s}$$

Numerická metoda

Snad nejznámější a nejjednodušší numerická metoda pro řešení diferenciálních rovnic se nazývá Eulerova metoda. S odvozením této metody se blíže seznámíte v kapitole 8.1.1. Pokud si přibližné řešení $x(t_n)$ v čase t_n označíme x_n , pak řešení v následujícím časovém okamžiku t_{n+1} obdržíme ze vzorce

$$x_{n+1} = x_n + h \cdot f(t_n, x_n) ,$$

kde $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ je funkce popisující diferenciální rovnici a h je diskretizační krok.

[Obsah](#)[14. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka/Okno](#)

Analýza metody

V následující tabulce můžeme srovnat přesnost numerického řešení s řešením analytickým.

Čas $t[\text{s}]$	Numerické řešení $x[\text{m}]$	Analytické řešení $x[\text{m}]$
0	-5	-5
0.2	-5	-4.9124
0.4	-4.8	-4.6992
0.6	-4.48	-4.3905
0.8	-4.036	-4.0440
1	-3.6864	-3.6788

[Obsah](#)[15. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka/Okno](#)

1.2. Korektnost úloh a stabilita výpočetních postupů

Při studiu numerického řešení matematických modelů se budeme zabývat pouze takovými úlohami, které jsou jednoznačně řešitelné a jejichž řešení spojitě závisí na vstupních datech. Takové úlohy budeme nazývat *korektní* a jejich přesnou definici bychom mohli formulovat následovně:

Definice 1.2. Buď dána množina vstupních dat B_1 a množina výstupních dat B_2 .

Úloha $y = U(x), x \in B_1, y \in B_2$ se nazývá *korektní* na dvojici (B_1, B_2) jestliže

1. ke každému $x \in B_1$ existuje jediné řešení $y \in B_2$
2. řešení spojitě závisí na vstupních datech,
tzn. $x_n \rightarrow x, U(x_n) = y_n \Rightarrow U(x_n) \rightarrow U(x) = y$



Obsah

16. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Při řešení praktických úloh se bohužel velmi často stává, že i korektní úloha vykazuje při zanedbatelné změně vstupních dat obrovské chyby ve výsledcích. Takové úlohy nazýváme *špatně podmíněné* a jsou velmi problematicky řešitelné pokud jsou jejich vstupní data zatíženy chybami jako jsou chyby měření, měřících zařízení, zaokrouhlovacími chybami apod. Níže uvedená definice nám vymezuje pojmy *dobře* a *špatně podmíněné* úlohy.

Definice 1.3. Budeme říkat, že korektní úloha je *dobře podmíněná* jestliže malá změna ve vstupních datech vyvolá malou změnu řešení.

Je-li $x + \Delta x$ resp. x změna řešení resp. řešení a $y + \Delta y$ resp. y změna vstupních dat resp. vstupní data, pak číslo

$$C_p = \frac{\frac{\|\Delta x\|}{\|x\|}}{\frac{\|\Delta y\|}{\|y\|}} = \frac{\text{relativní chyba na výstupu}}{\text{relativní chyba na vstupu}}$$

nazýváme *číslem podmíněnosti úlohy* $y = U(x)$.

Poznámka 1.4. Za dobře podmíněné úlohy považujeme úlohy s číslem podmíněnosti $1 < C_p < 100$. Pokud je $C_p > 100$ považujeme úlohy za hůře podmíněné a $C_p \gg 10000$ považujeme za špatně podmíněné.



Obsah

17. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 1.5. Následující příklad demonstruje případ špatně podmíněné úlohy.

Uvažujme soustavu lineárních rovnic

$$\begin{aligned}x_1 + 0.99x_2 &= 1.99 \\ 0.99x_1 + 0.98x_2 &= 1.97\end{aligned}$$

Snadno nalezneme řešení této *původní* soustavy $x = (x_1, x_2) = (1, 1)$.

Dále uvažujme stejnou soustavu s mírně pozměněnou pravou stranou

$$\begin{aligned}\tilde{x}_1 + 0.99\tilde{x}_2 &= 1.989903 \\ 0.99\tilde{x}_1 + 0.98\tilde{x}_2 &= 1.970106\end{aligned}$$

Řešení této *pozměněné* soustavy je $\tilde{x} = (\tilde{x}_1, \tilde{x}_2) = (3, -1.021)$.

Velikost změny řešení soustavy při změně pravé strany spočítáme snadno

$$\|\Delta x\| = \|\tilde{x} - x\| = \|(2, -2.021)\| \doteq 2.843315.$$

Pod změnou vstupních dat budeme v tomto případě rozumět velikost změny pravé strany (původní pravé strany b a změněné pravé strany $\tilde{b}$)

$$\|\Delta b\| = \|\tilde{b} - b\| = \|(1.989903, 1.970106) - (1.99, 1.97)\| \doteq 1.436836 \cdot 10^{-4}.$$

Pak číslo podmíněnosti úlohy spočítáme dle Definice 1.3 dosazením

$$C_p = \frac{\frac{\|\Delta \Delta x\|}{\|x\|}}{\frac{\|\Delta b\|}{\|b\|}} = \frac{\frac{2.843315}{1.414213}}{\frac{1.436836 \cdot 10^{-4}}{2.800178}} \doteq 39182.$$



Obsah

18. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Katastrofické důsledky na výslednou přesnost numerického řešení může mít i nevhodně zvolená numerická metoda. Šíření numerických chyb při takto nevhodně zvolené metodě může vést ke zcela chybnému výsledku. Následující příklad ukazuje případ takovéto metody.

Příklad 1.6. Pro libovolné $n \in \mathbb{N}$ určete hodnotu integrálu

$$J_n = \frac{1}{e} \int_0^1 x^n e^x dx .$$

Řešení. Při použití základních znalostí z matematické analýzy bychom mohli udělat následující odhady

$$|J_n| \leq \frac{1}{n+1}, \quad J_n \geq 0 .$$

Zřejmě můžeme snadno analyticky spočítat hodnotu integrálu pro volbu $n = 1$

$$J_0 = 1 - e^{-1} .$$

S využitím pravidla „per-partes“ můžeme získat následující předpis pro výpočet integrálu

$$J_n = \frac{1}{e} ([x^n e^x]_0^1 - n \int_0^1 x^{n-1} e^x dx) = 1 - n J_{n-1} . \quad (1.1)$$

Pokud použijeme tento rekursivní předpis pro výpočet, získáme hodnoty uvedené v následující tabulce.



Obsah

19. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

V přesné aritmetice:	Na 2 platné cifry:
$J_0 = 1 - \frac{1}{e} \doteq 0.6321$	$J_0 = 0.63$
$J_1 = \frac{1}{e} \doteq 0.3679$	$J_1 = 1 - 1 \cdot 0.63 \doteq 0.37$
$J_2 = \frac{e-2}{e} \doteq 0.2642$	$J_2 = 1 - 2 \cdot 0.37 \doteq 0.26$
...	
$J_6 = \frac{265e-720}{e} \doteq 0.1268$	$J_6 = -1.4$
...	
$J_{10} = \frac{24-9e}{e} \doteq 0.0835$	$J_{10} = -7800$
...	
$J_{20} \doteq 0$	$J_{20} = -5 \cdot 10^{15}$

Při výpočtu na 2 platné cifry tedy zaokrouhlovací chyby způsobily, že již J_6 je záporné (rozpor s odhadem $J_n \geq 0$) a J_{20} je výrazně mimo rozsah $|J_n| \leq 1/(n+1)$.

Očividně tato numerická metoda není *stabilní* (viz následující definice). ▲



Obsah

20. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Definice 1.7. Numerická metoda je *numericky stabilní*, jestliže malé odchylky v průběhu výpočtu (zaokrouhlení) nevedou k velkým změnám v řešení.

Příklad 1.8. Uvažujme stejnou úlohu jako v předchozím Příkladě 1.6:
Pro libovolné $n \in \mathbb{N}$ určete hodnotu integrálu

$$J_n = \frac{1}{e} \int_0^1 x^n e^x dx .$$

Pokusme se nalézt předpis stabilní metody.

Řešení. Upravme předpis pro rekursivní výpočet integrálu (1.1)

$$\begin{aligned} J_n &= 1 - nJ_{n-1} \\ nJ_{n-1} &= 1 - J_n \\ J_{n-1} &= \frac{1}{n}(1 - J_n) . \end{aligned}$$

Pro užití této formule odhadneme

$$J_{20} \doteq 0 \leq \frac{1}{1+20} .$$

Pak užitím nové formule při výpočtu na dvě platné cifry získáme tyto odhady.

$$J_{20} = 0, J_{10} = 0.084, J_6 = 0.13, J_2 = 0.26, J_1 = 0.37, J_0 = 0.63 .$$



Obsah

21. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 1.9. Numerická stabilita se posuzuje podobně jako podmíněnost úlohy pomocí čísla podmíněnosti algoritmu (stejná definice jako číslo podmíněnosti úlohy - viz Definice 1.3). Určuje se pomocí tzv. experimentálních perturbací, tzn. že mírně pozměňujeme vstupní data a sledujeme výsledky na výstupu.

Zatímco špatná podmíněnost úlohy je dána úlohou samou a v podstatě ji nemůžeme ovlivnit, stabilitu výpočetního postupu ovlivňujeme sami volbou metody.

[Obsah](#)[22. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka / Okno](#)

1.3. Typy chyb

Při řešení reálných úloh se setkáváme s celou řadou možných chyb, které ovlivňují přesnost výsledného řešení. Podle zdroje chyb bychom mohli chyby rozdělit do několika následujících skupin:

a) **Chyba matematického modelu**

Zjednodušováním původní úlohy jakož i zanedbáváním veličin, které nemají velký vliv na matematický model vznikají nepřesnosti v řešení matematického modelu. Tuto chybu lze kvantifikovat rozdílem mezi řešením reálného problému a řešením matematického modelu. Tato chyba vypovídá o přesnosti matematického modelu.

Př. Zanedbání odporu vzduch v Příkladě 1.1 vede k nepřesnosti v řešení matematického modelu.

b) **Chyba numerické metody**

Použití nepřesné metody pro řešení matematického modelu vede k chybě numerického řešení.

Př. Limitu posloupnosti aproximujeme členem s dostatečně velkým indexem,

$$\lim_{n \rightarrow \infty} a_n \approx a_{5000}.$$

c) **Chyba aproximace**

Aplikací přibližné metody na matematický model vzniká chyba aproximace. Pak rozdíl mezi řešením matematické úlohy a řešením numerické úlohy je chyba aproximace.

Př. Nahrazení spojitých vstupních dat diskrétními, viz. diskretizace času v Příkladě 1.1.



Obsah

23. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

d) **Chyby ve vstupních datech**

Vznikají chybami měření a reprezentací čísel v počítači.

e) **Chyby zaokrouhlovací**

Nepřesnosti způsobené realizací algoritmu v počítači a nepřesné provádění aritmetických operací.

1.3.1. Aproximace čísla

Při realizaci výpočtu nahrazujeme reálnou (přesnou) hodnotu x jeho aproximací (přiblížením) $\tilde{x}$. Značíme $\tilde{x} \approx x$.

Definice 1.10. Rozdíl $\Delta x = x - \tilde{x}$ nazýváme *absolutní chybou* aproximace $\tilde{x}$. Hodnotu $\varepsilon \geq 0$ takovou, že $|x - \tilde{x}| = |\Delta x| \leq \varepsilon(\tilde{x})$ nazýváme *odhad absolutní chyby*.

Definice 1.11. Číslo $\frac{\Delta x}{x} = \frac{x - \tilde{x}}{x}$, $x \neq 0$ nazýváme *relativní chybou* aproximace $\tilde{x}$. Číslo $\delta(x) \geq 0$ takové, že $|\frac{x - \tilde{x}}{x}| \leq \frac{\varepsilon(\tilde{x})}{|x|} = \delta(\tilde{x})$ nazýváme *odhad relativní chyby*.

Poznámka 1.12. Jelikož hodnota x není obvykle známa, pro odhad relativní chyby používáme vzorec $\frac{\Delta x}{\tilde{x}}$.

Poznámka 1.13. Z definice plyne, že $x = \tilde{x} \cdot (1 \pm \delta)$, kde $\delta = \frac{|\Delta x|}{|\tilde{x}|}$.



Obsah

24. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 1.14. Určete odhady absolutní a relativní chyby čísla $\tilde{x} = 3.141$, které nahrazuje číslo π .

Řešení. Platí (odhadněme)

$$3.141 < \pi < 3.142 \quad \Rightarrow \quad 0 < \pi - 3.141 < 0.001 .$$

Pak pro absolutní chybu lze psát

$$|\Delta x| = |\pi - \tilde{x}| = |\pi - 3.141| < 0.001 .$$

Tedy odhadem absolutní chyby je $\epsilon(\tilde{x}) = 0.001$.

Pro odhad relativní chyby dosadíme do vzorce z Definice 1.11

$$\delta(\tilde{x}) = \frac{\epsilon(\tilde{x})}{|\tilde{x}|} = \frac{0.001}{3.141} = 0.0003 ,$$

tedy lze psát $\pi = 3.141 \cdot (1 \pm 0.0003)$.



Obsah

25. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

1.3.2. Šíření chyb při aritmetických operacích

Aplikací výpočetního postupu na řešení numerických úloh provádíme celou řadu různých početních operací. Pokud do těchto operací vstupují veličiny, které jsou zatíženy nějakou aproximační chybou, zřejmě dochází k další propagaci těchto chyb do dílčích výsledků. Abychom lépe pochopily jak se tyto chyby v rámci složitějších výpočtů šíří či eliminují, je nutné si ukázat jaký vliv na jejich šíření mají základní aritmetické operace.

V dalším textu uvažujeme přesnou hodnotu x_i a její aprioximaci $\tilde{x}_i$. Dále ϵ_i je odhad její absolutní chyby a δ_i odhad relativní chyby. Označme pomocí u přesný výsledek aritmetické operace zatímco $\tilde{u}$ výsledek operace s nepřesnými operandy. Dále označme ϵ odhad absolutní chyby u a δ odhad relativní chyby u .

Dále uvedeme ϵ a δ pro jednotlivé elementární algebraické operace.

a) Chyba násobku veličiny

$$\begin{aligned} \text{Operace :} \quad & u = k \cdot x_1 \\ \text{Absolutní chyba :} \quad & |\Delta u| \leq |k| \cdot \epsilon_1 =: \epsilon \\ \text{Relativní chyba :} \quad & \left| \frac{\Delta u}{u} \right| \leq \delta_1 =: \delta \end{aligned}$$

b) Chyba algebraického součtu

$$\begin{aligned} \text{Operace :} \quad & u = x_1 + x_2 \\ \text{Absolutní chyba :} \quad & |\Delta u| \leq \epsilon_1 + \epsilon_2 =: \epsilon \\ \text{Relativní chyba :} \quad & \left| \frac{\Delta u}{u} \right| \leq \frac{1}{|\tilde{x}_1 + \tilde{x}_2|} (|\tilde{x}_1| \delta_1 + |\tilde{x}_2| \delta_2) =: \delta \end{aligned}$$



Obsah

26. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



c) Chyba rozdílu

$$\begin{aligned}
 \text{Operace :} \quad u &= x_1 - x_2 \\
 \text{Absolutní chyba :} \quad |\Delta u| &\leq \epsilon_1 + \epsilon_2 =: \epsilon \\
 \text{Relativní chyba :} \quad \left| \frac{\Delta u}{u} \right| &\leq \frac{1}{|\tilde{x}_1 - \tilde{x}_2|} (|\tilde{x}_1| \delta_1 + |\tilde{x}_2| \delta_2) =: \delta
 \end{aligned}$$

d) Chyba součinu

$$\begin{aligned}
 \text{Operace :} \quad u &= x_1 \cdot x_2 \\
 \text{Absolutní chyba :} \quad |\Delta u| &\leq \tilde{x}_1 \epsilon_2 + |\tilde{x}_2| \epsilon_1 =: \epsilon \\
 \text{Relativní chyba :} \quad \left| \frac{\Delta u}{u} \right| &\leq \delta_1 + \delta_2 =: \delta
 \end{aligned}$$

e) Chyba podílu

$$\begin{aligned}
 \text{Operace :} \quad u &= \frac{x_1}{x_2} \\
 \text{Absolutní chyba :} \quad |\Delta u| &\leq \frac{|\tilde{x}_2| \epsilon_1 + |\tilde{x}_1| \epsilon_2}{|\tilde{x}_2|^2} =: \epsilon \\
 \text{Relativní chyba :} \quad \left| \frac{\Delta u}{u} \right| &\leq \delta_1 + \delta_2 =: \delta
 \end{aligned}$$

f) Chyba mocniny

$$\begin{aligned}
 \text{Operace :} \quad u &= x_1^n \\
 \text{Absolutní chyba :} \quad |\Delta u| &\leq n |\tilde{x}_1|^{n-1} \epsilon_1 =: \epsilon \\
 \text{Relativní chyba :} \quad \left| \frac{\Delta u}{u} \right| &\leq n \cdot \delta_1 =: \delta
 \end{aligned}$$

Obsah

27. strana ze 415



Zavřít dokument

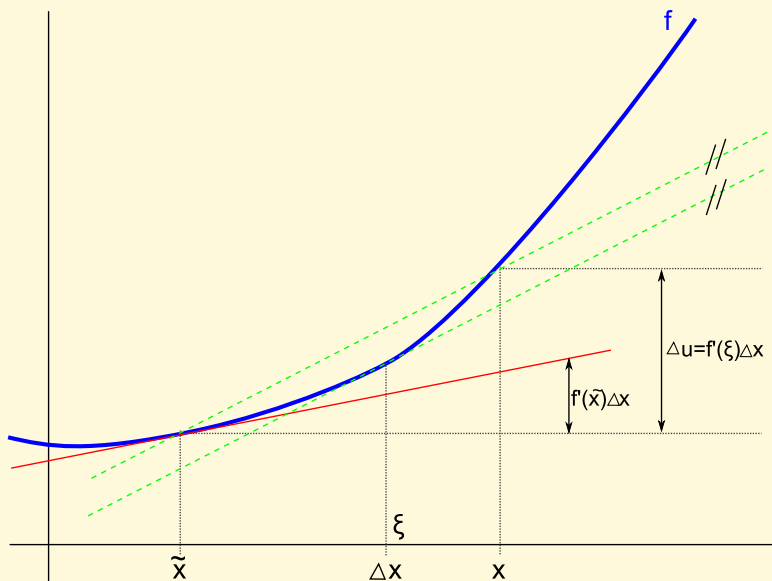
Celá obrazovka / Okno

g) Chyba funkční hodnoty

Operace : $u = f(x_1)$

Absolutní chyba : $|\Delta u| \leq M\xi =: \epsilon$

Relativní chyba : $|\frac{\Delta u}{u}| \leq \frac{|\tilde{x}_1| M \delta_1}{|f'(\tilde{x}_1)|} =: \delta$



Obr. 1.2: Chyba funkční hodnoty



Obsah

28. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 1.15. Z výše uvedených odhadů je patrné, že nekritičtější operací z hlediska šíření chyb je rozdíl dvou téměř shodných veličin, kdy odhad relativní chyby díky dělení tilde u výrazně narůstá a dělení číslem blízkým nule, které má za následek enormní nárůst odhadu absolutní chyby. Ve složitých výpočetních postupech je proto nutné se těchto operací vyvarovat !!!

1.3.3. Obecný vzorec chyb

Zobecněním chyby funkční hodnoty (viz případ *g*) v předešlé kapitole) na funkce více proměnných můžeme získat *obecný odhad chyby*. Jednotlivé aritmetické operace totiž můžeme reprezentovat jako funkce dvou proměnných (např. pro součet $f(x_1, x_2) = x_1 + x_2$) Navíc do funkce f můžeme zahrnout celou řadu operací s více veličinami $x_1, \dots, x_n$.

$$\begin{aligned} \text{Operace :} \quad & u = f(x_1, x_2, \dots, x_n) \\ \text{Absolutní chyba :} \quad & |\Delta u| \leq \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \epsilon_i =: \epsilon \\ \text{Relativní chyba :} \quad & \left| \frac{\Delta u}{u} \right| \leq \sum_{i=1}^n |\tilde{x}_i| \left| \frac{\partial \ln u}{\partial x_i} \right| \delta_i =: \delta \end{aligned}$$



Obsah

29. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 1.16. Určete odhad absolutní a relativní chyby objemu koule $V = \frac{1}{6}\pi d^3$ je-li

$$d = 3.7\text{cm} \pm 0.05\text{cm}, \quad \pi = 3.14 .$$

Řešení. Pro určení odhadu absolutní chyby použijeme výše odvozený obecný vzorec

$$\epsilon = \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \epsilon_i .$$

Očividně funkce objemu V je závislá na přesnosti proměnných π a d . Nejdříve napočítejme potřebné parciální derivace

$$\begin{aligned} \frac{\partial V(\pi, d)}{\partial \pi} &= \frac{1}{6}d^3, & \frac{\partial V(3.14, 3.7)}{\partial \pi} &= \frac{1}{6} \cdot 3.7^3 \doteq 8.44, \\ \frac{\partial V(\pi, d)}{\partial d} &= \frac{1}{2}\pi d^2, & \frac{\partial V(3.14, 3.7)}{\partial d} &= \frac{1}{2} \cdot 3.14 \cdot 3.7^2 \doteq 21.5. \end{aligned}$$

a dosazením

$$\epsilon = \left| \frac{\partial V}{\partial \pi} \right| |\Delta \pi| + \left| \frac{\partial V}{\partial d} \right| |\Delta d| = 8.44 \cdot 0.0016 + 21.5 \cdot 0.05 \doteq 1.1$$

$$\text{Tedy } V = \frac{1}{6}\pi d^3 \approx 27.4\text{cm}^3 \pm 1.1\text{cm}^3.$$

Pro určení odhadu relativní chyby použijeme snažší vyjádření

$$\delta = \left| \frac{\Delta V}{V} \right| \approx \frac{1.088}{27.4} \doteq 3.97 .$$



Obsah

30. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

1.3.4. Obrácená úloha

Mnohdy je nutné odhadnout chybu nezávislé proměnné tak, aby přitom byla zaručena přesnost výsledné veličiny.

Platí

$$|\Delta u| \leq \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \epsilon_i .$$

Jsou-li chyby n nezávislých proměnných x_i řádově stejné, pak můžeme jejich příspěvek do celkové chyby rozdělit stejným dílem tj.

$$\left| \frac{\partial f}{\partial x_i} \right| \epsilon_i \doteq \frac{\Delta u}{n} \text{ pro } \forall i = 1, 2, \dots, n ,$$

pak

$$\epsilon_i \doteq \frac{|\Delta u|}{n \left| \frac{\partial f}{\partial x_i} \right|} . \quad (1.2)$$



Obsah

31. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 1.17. Necht je dán válec o poloměru $r \approx 2\text{m}$ a výšce $h \approx 3\text{m}$. Jaké musí být absolutní chyby Δr a Δh aby objem válce byl určen s přesností na 0.1m^3 ?

Řešení. Objem válce lze spočítat pomocí vzorce

$$V = \pi r^2 h ,$$

přičemž ze zadání víme, že

$$\begin{aligned} r &= 2\text{m} , & h &= 3\text{m} , \\ \pi &= 3.14 , & \Delta V &= 0.1\text{m}^3 . \end{aligned}$$

Potřebné odhady budeme počítat užitím vzorce (1.2), je proto nutno nejdříve napočítat potřebné parciální derivace

$$\begin{aligned} \frac{\partial V(r,h,\pi)}{\partial r} &= 2\pi r h , & \frac{\partial V(2,3,3.14)}{\partial r} &= 2 \cdot 3.14 \cdot 2 \cdot 3 \doteq 37.7 , \\ \frac{\partial V(r,h,\pi)}{\partial h} &= \pi r^2 , & \frac{\partial V(2,3,3.14)}{\partial h} &= 3.14 \cdot 2^2 \doteq 12.6 , \\ \frac{\partial V(r,h,\pi)}{\partial \pi} &= r^2 h , & \frac{\partial V(2,3,3.14)}{\partial \pi} &= 2^2 \cdot 3 = 12 . \end{aligned}$$

Při dosazování do vzorce je nutno si uvědomit, že přesnost výsledného objemu je závislá na třech vstupních parametrech, tedy do vzorců budeme dosazovat $n = 3$.

$$\begin{aligned} |\Delta r| &\leq \epsilon_r = \frac{|\Delta V|}{n \left| \frac{\partial V}{\partial r} \right|} \doteq \frac{0.1}{3 \cdot 37.7} < 0.001 \\ |\Delta h| &\leq \epsilon_h = \frac{|\Delta V|}{n \left| \frac{\partial V}{\partial h} \right|} \doteq \frac{0.1}{3 \cdot 12.6} < 0.003 \\ |\Delta \pi| &\leq \epsilon_\pi = \frac{|\Delta V|}{n \left| \frac{\partial V}{\partial \pi} \right|} \doteq \frac{0.1}{3 \cdot 12} < 0.003 \end{aligned}$$

Proto pro dosažení požadované přesnosti výpočtu objemu válce je nutno, aby $\Delta r < 0.001\text{m}$ a $\Delta h < 0.003\text{m}$. ▲



Obsah

32. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

1.4. Aproximace čísel na počítači

Zásadím problémem použití počítačů pro výpočty s požadavkem na vysokou přesnost je nemožnost reprezentace reálných, přesněji iracionálních čísel v počítači. Je zřejmé, že iracionální čísla, která mají nekonečný počet desetinných míst nelze nikterak uložit v paměti počítače. Abychom tato čísla byli schopni v počítači reprezentovat, musíme je aproximovat čísly s konečným počtem desetinných míst.

Pro tyto účely se nejvíce hodí tzv. *semilogaritmický tvar s normalizovanou mantisou*.

Definice 1.18. Pod pojmem *semilogaritmický tvar s normalizovanou mantisou* rozumíme zápis libovolného racionálního čísla a ve tvaru

$$a = \operatorname{sgn} a \cdot (a_1 q^{-1} + a_2 q^{-2} + \cdots + a_t q^{-t}) q^b,$$

kde $q \in \mathbb{N} \setminus \{1\}$ je *základ číselné soustavy*, $a_i \in \{0, 1, 2, \dots, q-1\}$ jsou *číslíčky mantisy* a $b \in \mathbb{Z}$ je *exponent*, který je omezen maximální a minimální hodnotou $m_1, m_2 \in \mathbb{Z}$, tj.

$$m_1 \leq b \leq m_2.$$



Obsah

33. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 1.19. Dříve než budeme pokračovat ve výkladu, zopakujme si na konkrétním příkladě alespoň převody mezi dekadickou ($q = 10$) a binární ($q = 2$) soustavou.

Převedte čísla $[10011]_2, [10.011]_2$ do dekadické soustavy¹.

Převedte čísla $[1234]_{10}, [12.34]_{10}$ do binární soustavy.

Řešení. Dle standartní teorie číselných soustav známé ze střední školy lze psát

$$\begin{aligned} [10011]_2 &= [1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0]_{10} \\ &= [16 + 0 + 0 + 2 + 1]_{10} = [19]_{10} \end{aligned}$$

$$\begin{aligned} [10.011]_2 &= [1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}]_{10} \\ &= [2 + 0 + 0 + 0.25 + 0.125]_{10} = [2.375]_{10} \end{aligned}$$

Převod z dekadické do binární soustavy lze provést obdobně - hledáním vhodných koeficientů v součtu jednotlivých mocnin základu $q = 2$ (tj. cifer v binárním zápisu). Nicméně opět ze střední školy je znám rychlejší postup. Uvedeme jej bez zbytečného komentáře.

$$\left. \begin{array}{ll} 1234 : 2 = 617, & \text{zbytek } 0 \\ 617 : 2 = 308, & \text{zbytek } 1 \\ 308 : 2 = 154, & \text{zbytek } 0 \\ 154 : 2 = 77, & \text{zbytek } 0 \\ 77 : 2 = 38, & \text{zbytek } 1 \\ 38 : 2 = 19, & \text{zbytek } 0 \\ 19 : 2 = 9, & \text{zbytek } 1 \\ 9 : 2 = 4, & \text{zbytek } 1 \\ 4 : 2 = 2, & \text{zbytek } 0 \\ 2 : 2 = 1, & \text{zbytek } 0 \\ 1 : 2 = 0, & \text{zbytek } 1 \end{array} \right\} \Rightarrow [1234]_{10} = [10001010]_2$$

¹Zápis $[x]_q$ znamená, že číslo x je uvedeno v číselné soustavě se základem q .



Obsah

34. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Neméně zajímavá je i modifikace postupu pro převod desetinných čísel.

12 : 2 = 6,	zbytek 0	0.34 · 2 = 0.68,	celá část 0
6 : 2 = 3,	zbytek 0	0.68 · 2 = 1.36,	celá část 1
3 : 2 = 1,	zbytek 1	0.36 · 2 = 0.72,	celá část 0
1 : 2 = 0,	zbytek 1	0.72 · 2 = 1.44,	celá část 1
		0.44 · 2 = 0.88,	celá část 0
		0.88 · 2 = 1.76,	celá část 1
		0.76 · 2 = 1.52,	celá část 1
		0.52 · 2 = 1.04,	celá část 1
		...	

$$\Rightarrow [12.34]_{10} = [1100.01010111 \dots]_2$$



Obsah

35. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Definice 1.20. Množinu všech čísel zapsaných pomocí semilogaritmického tvaru s normalizovanou mantisou (viz Definice 1.18) nazýváme *systémem s pohyblivou řádovou čárkou*^a a značíme

$$F(q, t, m_1, m_2) .$$

^aangl. *floating point system*

Poznámka 1.21. V případě tzv. dvojnásobné aritmetiky nahrazujeme t hodnotou $2t$ a zvětšujeme m_1 a m_2 .

V případě reálných čísel je tedy možné každé číslo $x \in \mathbb{R}$ reprezentovat následujícím způsobem

$$x = a.q^b = \operatorname{sgn} x \left(\sum_{n=1}^{\infty} x_n q^{-n} \right) q^b$$

Jeho aproximace v semilogaritmickém tvaru s normalizovanou mantisou pak může mít tvar

$$\tilde{x} \in F(q, t, m_1, m_2) \Rightarrow \tilde{x} = \tilde{a}q^b = \operatorname{sgn} \tilde{x} \left(\sum_{n=1}^t \tilde{x}_n q^{-n} \right) q^b$$



Obsah

36. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Reprezentaci reálných čísel na počítači tedy definujeme pomocí zobrazení

$$fl : \mathbb{R} \rightarrow F(q, t, m_1, m_2)$$

Zobrazení provádíme

a) řezáním - $\tilde{x}_k = x_k, k = 1, 2, \dots, t$

b) zaokrouhlováním -

$$\begin{aligned} \tilde{x}_k &= x_k, \text{ pro } k = 1, 2, \dots, t-1 \\ x_t &= \begin{cases} x_t + 1 & \text{pro } x_{t+1} \geq \frac{q}{2} \\ x_t & \text{pro } x_{t+1} < \frac{q}{2} \end{cases} \end{aligned}$$

Poznámka 1.22. $\left| \frac{x - fl(x)}{x} \right| = \kappa q^{1-t}$, kde $\kappa = 1$ pro řezání a $\kappa = \frac{1}{2}$ pro zaokrouhlování.
 $|x - fl(x)| = \kappa q^{b-t}$

Nejmenší možné kladné číslo, které jsme schopni uložit je určeno délkou mantisy. Čísla lišící se o toto číslo se jeví jako totožná.

Definice 1.23. Nejmenší číslo $x \in F(q, t, m_1, m_2)$ takové, že $1 + x > 1, x > 0$ nazýváme počítačovým ϵ (počítačová jednotka).



Obsah

37. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

1.5. Test

Následující test zahájíte kliknutím na tlačítko „Zacátek testu“. U otázek, kde lze volit jen jednu odpověď (test nedovolí zaškrtnout více odpovědí), je správná odpověď bodována počtem bodů uvedených v závorce u zadání a špatná odpověď je bodována 0 body. U otázek, kde lze volit více správných odpovědí, je součet bodů správných odpovědí uveden v závorce u zadání a za každou špatnou odpověď bude odečten jeden bod.

1. (4b.) V standardním schématu řešení inženýrské úlohy nenajdeme

reálný problém,

analytické řešení numerického modelu,

numerické řešení matematického modelu,

numerické řešení numerického modelu,

analýza experimentálně naměřených dat a použitých měřících přístrojů,

porovnání analytického a numerického řešení s experimentálním řešením úlohy, přímé numerické řešení reálného problému.



Obsah

38. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2. (4b.) Matematický model se vytváří z reálného problému

matematickým řešením reálného problému,

experimentálním měřením,

analytickým řešením reálného problému,

vytvořením vhodného numerického modelu úlohy.

zanedbáním některých fyzikálních veličin,

3. (4b.) Experimentální řešení inženýrské úlohy se provádí například

měřením fyzikálních vlastností, laboratorními testy,

numerickými experimenty,

porovnáním analytického řešení s naměřenými daty,

porovnáním numerického řešení s naměřenými daty,

vytvořením vhodného matematického modelu úlohy,

vytvořením vhodného numerického modelu úlohy.



Obsah

39. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4. (4b.) Dokončete pravdivé tvrzení: Korektní úloha ...

je jednoznačně řešitelná.

je taková úloha, jejíž řešení nemusí být jednoznačné.

je taková úloha, u které ke každé množině parametrů úlohy existuje právě jedno řešení této úlohy.

je taková úloha, že její řešení spojitě závisí na vstupních datech,

je dobře podmíněná.

je numericky stabilní.

má malé číslo podmíněnosti úlohy.

5. (4b.) Dokončete pravdivé tvrzení: Dobře podmíněná úloha ...

je libovolná korektní úloha.

je úloha, jejíž řešení je jednoznačné.

je taková úloha, při které malá změna vstupních dat vyvolá malou změnu řešení.

má malé číslo podmíněnosti.

je numericky stabilní.



Obsah

40. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

6. (4b.) Dokončete pravdivé tvrzení: Numericky stabilní numerická metoda ...

je metoda, která řeší korektní úlohu.

je metoda, která řeší dobře podmíněnou úlohu.

je metoda, při které malé odchylky v průběhu výpočtu nevedou k velkým změnám v řešení.

je metoda, u které zanedbáváme chyby zaokrouhlením.

má vždy malé číslo podmíněnosti úlohy.

7. (4b.) Při řešení reálných úloh se můžeme například setkat s těmito typy chyb ...

chyba matematického modelu,

chyba aproximace,

chyba použité numerické metody.

zaokrouhlovací chyby.

chyba analytického řešení numerického modelu.



Obsah

41. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



8. (4b.) Aproximace čísla ...
je nahrazení přesné hodnoty jeho přibližnou hodnotou, je malá perturbace ve vstupních datech,
je podíl přesné a přibližné hodnoty čísla, je absolutní hodnota rozdílu přesné a přibližné hodnoty.
9. (4b.) Absolutní chyba aproximace čísla je ...
je malá perturbace ve vstupních datech, je rozdíl přesné a přibližné hodnoty podělený přesnou hodnotou,
je rozdíl přesné a přibližné hodnoty. je nahrazení přesné hodnoty jeho přibližnou hodnotou,
je rovna číslu podmíněnosti numerické metody. je rovna číslu podmíněnosti korektní úlohy.
10. (4b.) Absolutní chyba algebraického součtu dvou hodnot, které jsou zatíženy chybou, je rovna ...
součtu absolutních chyb jednotlivých hodnot, rozdíl absolutních chyb jednotlivých hodnot,
maximu absolutních chyb jednotlivých hodnot, minimu absolutních chyb jednotlivých hodnot,
rozdílu absolutních chyb jednotlivých hodnot, součinu absolutních chyb jednotlivých hodnot.

Obsah

42. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

11. (4b.) Relativní chyba algebraického součinu dvou hodnot, které jsou zatížené chybou, je rovna ...

součtu absolutních chyb jednotlivých hodnot,

součtu relativních chyb jednotlivých hodnot,

maximu absolutních chyb jednotlivých hodnot,

maximu relativních chyb jednotlivých hodnot,

součinu absolutních chyb jednotlivých hodnot,

součinu relativních chyb jednotlivých hodnot.

12. (2b.) Určete odhad absolutní chyby obsahu čtverce se stranou $a = 10\text{cm} \pm 1\text{cm}$.
 cm^2

13. (2b.) Určete odhad absolutní chyby obvodu čtverce se stranou $a = 10\text{cm} \pm 1\text{cm}$.
 cm

14. (4b.) Nekritičtější operací z hlediska šíření chyb je ...

součet dvou téměř shodných veličin,

rozdíl dvou téměř shodných veličin,

součin dvou téměř shodných veličin,

podíl dvou téměř shodných veličin.



Obsah

43. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



15. (4b.) Problém aproximace čísel na počítači je způsoben

rychlostí výpočtu zvolenou numerickou metodou,

reprezentací iracionálních čísel v počítači,

reprezentací celých čísel v počítači,

nekorektností úloh,

chybou algebraických operací.

16. (4b.) Počítačová jednotka (počítačové ϵ) je ...

nejmenší reálné číslo, které je možno reprezentovat v počítači,

největší reálné číslo, které je možno reprezentovat v počítači,

nejmenší číslo x z množinu všech čísel zapsaných pomocí semilogaritmického tvaru s normalizovanou mantisou, pro které $1 + x > 1$,

je nejmenší část počítače,

je algebraická operace s nejmenší absolutní chybou,

je algebraická operace s nejmenší relativní chybou.

17. (2b.) Výsledkem součtu čísel $[11101]_2$ a $[1011]_2$ v dvojkové soustavě je (výsledek vyjádřete také v dvojkové soustavě)

18. (2b.) Od čísla $[1011]_2$ odečtete číslo $[110]_2$. Výsledek vyjádřete v dvojkové soustavě.

Obsah

44. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Správně zodpovězené otázky:

Získané body:

Procento úspěšnosti:

Pro ukončení testu je třeba kliknout na tlačítko „Konec testu“. Opravení testu se provede kliknutím na tlačítko „Výsledky“, správné odpovědi budou označeny zeleně a červeně budou označeny odpovědi chybné. Při použití otázky s tvořenou odpovědí se správná odpověď zobrazí v rámečku umístěném vpravo dole v navigačním panelu po kliknutí na tlačítko „Odpověď“.



Obsah

45. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Kapitola 2

Numerické řešení nelineárních rovnic

Průvodce studiem

Cílem této kapitoly je představit základní numerické metody pro řešení nelineárních rovnic a jejich soustav. Hledání kořenů obecně nelineárních rovnic je snad nejběžnější matematickou úlohou, se kterou se můžeme setkat nejen v inženýrské praxi, ale i v každodenním životě, například při řešení jednoduchých geometrických úloh. Analytické řešení těchto rovnic je však velmi často netriviální a nezbývá než nasadit na jejich řešení numerické metody. V této kapitole se nejprve čtenář seznámí s metodami separace kořenů těchto rovnic. Následně budou odvozeny a analyzovány základní numerické metody, které je na řešení nelineárních rovnic a jejich soustav možné použít.

[Obsah](#)

46. strana ze 415

[Zavřít dokument](#)[Celá obrazovka/Okno](#)

2.1. Separace kořenů

Prvním krokem, který předchází numerickému řešení nelineárních rovnic je lokalizace jejich kořenů. Pro obecné nelineární rovnice může být tento úkol velmi problematicky řešitelný a nelze jej řešit jinak než zobrazením grafu funkce. V případě soustav mnoha nelineárních rovnic je však i tento postup prakticky neřešitelný. Naopak pro některé specifické typy rovnic jako jsou algebraické rovnice, existuje již poměrně rozsáhlý aparát. Procesu určování počtu kořenů a jejich lokalizace říkáme separace kořenů a základní principy si ukážeme v následujícím textu.

Nejprve si definujme co budeme rozumět pod pojmem rovnice a jí kořen.

Definice 2.1. Nechť funkce $f : \mathbb{R} \rightarrow \mathbb{R}$ je definována na intervalu $\langle a, b \rangle \subset \mathbb{R}$. Číslo $\bar{x} \in \langle a, b \rangle$, pro které $f(\bar{x}) = 0$, nazýváme *kořenem rovnice* $f(x) = 0$.

Pro určení intervalu, ve kterém zcela jistě nalezneme kořen rovnice, můžeme použít následujícího triviálního tvrzení.



Obsah

47. strana ze 415

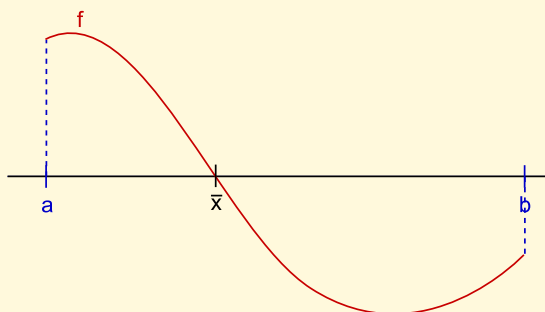


Zavřít dokument

Celá obrazovka / Okno

Lemma 2.2. *Nechť f je spojitá na $\langle a, b \rangle \subset \mathbb{R}$. Pak je-li $f(a) \cdot f(b) < 0$ má rovnice $f(x) = 0$ alespoň jeden kořen v intervalu $\langle a, b \rangle$.*

Platnost tohoto lemmatu lze demonstrovat na Obrázku 2.1.



Obr. 2.1: Příklad funkce spojité na $\langle a, b \rangle$ s vlastností $f(a) \cdot f(b) < 0$

Uvedené Lemma nám sice zaručuje existenci kořenu v daném intervalu, avšak nezaručuje, že kořen v daném intervalu bude právě jeden. Pro nasazení numerických metod je však mnohdy velmi důležité aby kořen byl v daném intervalu pouze jeden.

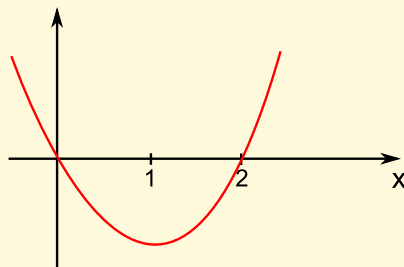
Navíc se jedná pouze o podmínku postačující neboť v případě, že v daném intervalu budou kořeny dva nebo obecněji sudý počet, nebude podmínka splněna.

Pro obecné funkce existuje několik základních způsobů separace kořenů:

1. Grafická separace

a) polohu a počet kořenů určíme z grafu funkce $f(x)$.

Například z grafu funkce $f(x) = (x-1)^2 - 1$ (viz Obrázek 2.2) lze určit přibližnou



Obr. 2.2: Průběh funkce $f(x)$

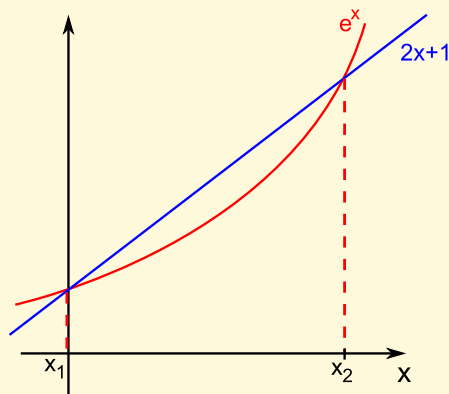
polohu koeficientů $\bar{x}_1 \in (-\frac{1}{2}, \frac{1}{2})$, $\bar{x}_2 \in (1, 5)$.

b) polohu a počet kořenů určíme vhodným rozložením funkce $f(x)$ na více funkcí a z grafů těchto funkcí.

$$(f(x) = f_1(x) - f_2(x) = 0, f_1(x) = f_2(x))$$

Například funkci $f(x) = e^x - 2x - 1$ lze rozložit na rozdíl $f(x) = e^x - (2x + 1) \stackrel{\text{ozn}}{=} f_1(x) - f_2(x)$ a pak graficky hledat řešení rovnice $f_1(x) = f_2(x)$ (viz Obrázek 2.3).

Z obrázku pak lze odhadnout polohu kořenů $\bar{x}_1 \in (-\frac{1}{2}, \frac{1}{2})$, $\bar{x}_2 \in (1, 2)$.



Obr. 2.3: Průběh funkcí $f_1(x)$ a $f_2(x)$

2. Tabelací funkce

Například pro funkci $f(x) \stackrel{\text{def}}{=} e^x - 2x - 1$ lze spočítat funkční hodnotu v několika bodech a ze změny znaménka následně odhadnout polohu kořenů.

x_i	$f(x_i)$	$\text{sgn } f(x_i)$
-1	1.3679	+
-0.5	0.6065	+
0	0	0
0.5	-0.3513	-
1	-0.4817	-
1.5	0.4817	+
2	2.3691	+

Ve zvoleném příkladu jsme měli štěstí a podařilo se nám přímo nalézt jeden kořen $\bar{x}_1 = 0$. Očividně pro druhý kořen platí $\bar{x}_2 \in (1, 1.5)$.

Poznámka 2.3. Pro polynomické rovnice ($p(x) = 0, p \in P_n$) existují speciální metody pro separaci kořenů (viz např. [7]).



Obsah

51. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Podobně jako pro funkce jedné reálné proměnné definujeme nelineární rovnici a její kořen, tak pro vektorové funkce více proměnných můžeme definovat soustavu rovnic a vektor kořenů soustavy.

Definice 2.4. Necht funkce $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ je definována na souvislé oblasti (ohraničená, otevřená podmnožina $\mathbb{R}^n$) $\Omega \subseteq \mathbb{R}^n$. Vektor $\underline{x} \in \Omega$, pro který $F(\underline{x}) = \underline{0}$ nazýváme *vektorem kořenů soustavy nelineárních rovnic* $F(\underline{x}) = \underline{0}$.

Poznámka 2.5. Soustavu $F(\underline{x}) = \underline{0}$ obvykle zapisujeme ve tvaru:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned}$$



Obsah

52. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pro srovnání kvality jednotlivých numerických metod co do rychlosti konvergence, zavádíme pojem *řád iterační metody*.

Definice 2.6. Necht $\{x_n\}$ je posloupnost aproximací čísla x , $x_n \rightarrow \bar{x}$ pro $n \rightarrow \infty$. Říkáme, že iterační metoda je *řádu p* , jestliže

$$\exists C = \text{konst} > 0, \text{ takové, že } |\bar{x} - x_{n+1}| \leq C|\bar{x} - x_n|^p, \text{ pro } n = 0, 1, 2, \dots$$

Prakticky tedy znamená, že metoda řádu p má chybu nové aproximace až na konstantu menší než p -tou mocninu chyby předchozí aproximace. Čím větší řád metody, tím rychleji tedy metoda konverguje.

[Obsah](#)[53. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka/Okno](#)

2.2. Metoda půlení intervalu

Snad nejznámější a nejjednodušší numerickou metodou je *metoda půlení intervalu*. Tato metoda je založena na Větě 2.2.

Nechť $a_0 = a, b_0 = b$ jsou takové, že $f(a_0) \cdot f(b_0) < 0$ a nechť f je spojitá na $\langle a, b \rangle$. Z Věty 2.2 vyplývá, že v intervalu $\langle a, b \rangle$ leží alespoň jeden kořen. Předpokládejme dále, že zde leží právě jeden kořen.

Metoda půlení intervalu je založena na konstrukci posloupnosti do sebe vnořených podintervalů $\{a_n\}, \{b_n\}$

$$\langle a_0, b_0 \rangle \supset \langle a_1, b_1 \rangle \supset \cdots \supset \langle a_n, b_n \rangle \supset \langle a_{n+1}, b_{n+1} \rangle \supset \cdots$$

takových, že

$$f(a_k) \cdot f(b_k) \leq 0, \quad \forall k = 0, 1, 2, \dots$$

Je-li $\langle a_k, b_k \rangle$ interval pro něhož je $f(a_k) \cdot f(b_k) \leq 0$, pak $\langle a_{k+1}, b_{k+1} \rangle$ dostaneme následujícím způsobem:

1. vypočteme $s_{k+1} = \frac{1}{2}(a_k + b_k)$ (střed intervalu),
2. je-li $f(a_k) \cdot f(s_{k+1}) < 0$, pak $a_{k+1} = a_k, b_{k+1} = s_{k+1}$,
3. je-li $f(s_{k+1}) \cdot f(b_k) < 0$, pak $a_{k+1} = s_{k+1}, b_{k+1} = b_k$.



Obsah

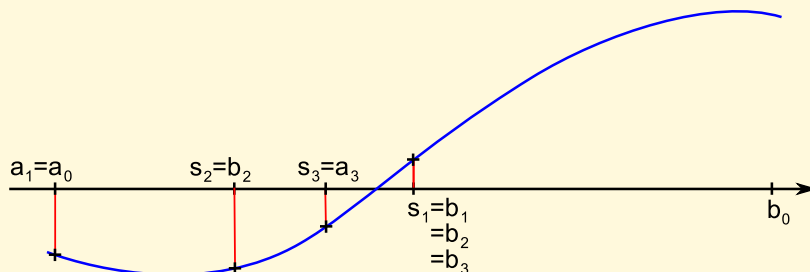
54. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Princip metody můžeme snadno demonstrovat na Obrázku 2.4. Z obrázku je taktéž patrné odkud pochází název metody, neboť v každém kroce dochází k rozpůlení intervalu a zmenšení intervalu, ve kterém se nachází kořen, na polovinu.



Obr. 2.4: Metoda půlení intervalu

Algoritmus 1**METODA PŮLENÍ INTERVALU****Input:** a_0, b_0 , přesnost $\epsilon > 0$

```
1:  $k := 0$ 
2: while  $\|b_k - a_k\| \geq \epsilon$  do
3:    $s_{k+1} := (a_k + b_k)/2$ 
4:   if  $f(a_k) \cdot f(s_{k+1}) < 0$  then
5:      $a_{k+1} := a_k$ 
6:      $b_{k+1} := s_{k+1}$ 
7:   else
8:     if  $f(s_{k+1}) \cdot f(b_k) < 0$  then
9:        $a_{k+1} := s_{k+1}$ 
10:       $b_{k+1} := b_k$ 
11:    else
12:       $a_{k+1} := s_{k+1}$ 
13:       $b_{k+1} := s_{k+1}$ 
14:    end if
15:  end if
16:   $k := k + 1$ 
17: end while
```

Output: s_k, a_k, b_k 

Obsah

56. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

2.2.1. Konvergence

Konvergenci metody půlení intervalu zaručuje následující lemma.

Lemma 2.7. *Jsou-li splněny předpoklady věty 2.2, pak je metoda půlení intervalu konvergentní. Řád konvergence je 1.*



Obsah

57. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Příklad 2.8. Nalezněte kořeny polynomu

$$f(x) := x^3 - x^2 - 2x + 2$$

s přesností $\epsilon = 10^{-4}$.

Řešení. Nejdříve provedeme jednoduchou analýzu dané funkce pro nalezení intervalových odhadů jednotlivých kořenů.

Z první a druhé derivace zadané funkce

$$f'(x) = 3x^2 - 2x - 2, \quad f''(x) = 6x - 2$$

lze sestavit tabulky monotonie a konvexnosti/konkávnosti dané funkce.

x	$f'(x)$	f
$\left(-\infty, \frac{1-\sqrt{7}}{3}\right)$	> 0	rostoucí
$\left(\frac{1-\sqrt{7}}{3}, \frac{1+\sqrt{7}}{3}\right)$	< 0	klesající
$\left(\frac{1+\sqrt{7}}{3}, \infty\right)$	> 0	rostoucí

x	$f'(x)$	f
$\left(-\infty, \frac{1}{3}\right)$	< 0	konkávní
$\left(\frac{1}{3}, \infty\right)$	> 0	konvexní



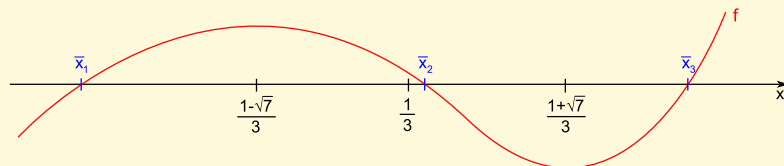
Obsah

58. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Obr. 2.5: Kvalitativní vlastnosti funkce f .

Následně lze dopočítat potřebné funkční hodnoty a načrtnout průběh funkce f (viz Obrázek 2.5).

Budeme pátrat po hodnotách $\bar{x}_1, \bar{x}_2, \bar{x}_3$. Očividně

$$\bar{x}_2 \in \left(\frac{1}{3}, \frac{1+\sqrt{7}}{3} \right). \quad (2.1)$$

Zbývá určit intervalové odhady kořenů $\bar{x}_1, \bar{x}_2$ a to například pomocí tabelace funkce. Volme tabulační parametr $h = 0,2$.

x	f	x	f
$\frac{1-\sqrt{7}}{3}$	2,6311	$\frac{1+\sqrt{7}}{3}$	-0,1126
$\frac{1-\sqrt{7}}{3} - h$	2,5173	$\frac{1+\sqrt{7}}{3} + h$	0,0012
$\frac{1-\sqrt{7}}{3} - 2h$	2,1438		
$\frac{1-\sqrt{7}}{3} - 3h$	1,4627		
$\frac{1-\sqrt{7}}{3} - 4h$	0,4258		
$\frac{1-\sqrt{7}}{3} - 5h$	-1,0146		

Z předchozí tabelace jsou zřejmé odhady kořenů

$$\begin{aligned}\bar{x}_1 &\in \left(\frac{1-\sqrt{7}}{3} - 1, \quad \frac{1-\sqrt{7}}{3} - 0,8 \right) \\ \bar{x}_3 &\in \left(\frac{1+\sqrt{7}}{3}, \quad \frac{1+\sqrt{7}}{3} + 0,2 \right).\end{aligned}\tag{2.2}$$



Obsah

60. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Pak metodou půlení intervalu získáme v jednotlivých iteracích následující odhady. Pro hledání kořene $\bar{x}_1$ získáme následující tabulku.

i	a	s	b
0.	-1.5486	-1.4486	-1.3486
1.	-1.4486	-1.3986	-1.3486
2.	-1.4486	-1.4236	-1.3986
3.	-1.4236	-1.4111	-1.3986
4.	-1.4236	-1.4173	-1.4111
5.	-1.4173	-1.4142	-1.4111
6.	-1.4173	-1.4158	-1.4142
7.	-1.4158	-1.4150	-1.4142
8.	-1.4150	-1.4146	-1.4142
9.	-1.4146	-1.4144	-1.4142
10.	-1.4144	-1.4143	-1.4142
11.	-1.4143	-1.4143	-1.4142

[Obsah](#)[61. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka/Okno](#)

Pro hledání kořene $\bar{x}_2$ získáme následující tabulku.

i	a	s	b
0.	0.3333	0.3333	1.2153
1.	0.7743	0.7743	1.2153
2.	0.9948	0.9948	1.2153
3.	0.9948	1.1050	1.2153
4.	0.9948	1.0499	1.1050
5.	0.9948	1.0223	1.0499
6.	0.9948	1.0086	1.0223
7.	0.9948	1.0017	1.0086
8.	0.9982	0.9982	1.0017
9.	0.9999	0.9999	1.0017
10.	0.9999	1.0008	1.0017
11.	0.9999	1.0004	1.0008
12.	0.9999	1.0002	1.0004
13.	0.9999	1.0000	1.0002
14.	1.0000	1.0000	1.0000



Obsah

62. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Pro hledání kořene $\bar{x}_3$ získáme následující tabulku.

i	a	s	b
0.	1.2153	1.3153	1.4153
1.	1.3153	1.3653	1.4153
2.	1.3653	1.3903	1.4153
3.	1.3903	1.4028	1.4153
4.	1.4028	1.4090	1.4153
5.	1.4090	1.4121	1.4153
6.	1.4121	1.4137	1.4153
7.	1.4137	1.4145	1.4153
8.	1.4137	1.4141	1.4145
9.	1.4141	1.4143	1.4145
10.	1.4141	1.4142	1.4143
11.	1.4142	1.4142	1.4143

Hledané kořeny mají hodnoty $\bar{x}_1 = -1.4143$, $\bar{x}_2 = 1$, $\bar{x}_3 = 1.4142$ s přesností $\epsilon = 10^{-4}$. ▲



Obsah

63. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2.3. Metoda prostých iterací

Metoda prostých iterací je založena na konstrukci posloupnosti aproximací

$$\{x_k\}, \quad k = 0, 1, 2, \dots$$

rekurzivním předpisem

$$x_{k+1} = G(x_k) .$$

Pokud má zobrazení $G : \mathbb{R} \rightarrow \mathbb{R}$ určité níže uvedené vlastnosti, tak je metoda konvergentní a konverguje k hledanému řešení.

Definice 2.9. Zobrazení $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ se nazývá *kontraktivní*, jestliže existuje $0 < \rho < 1$ takové, že

$$\forall x, y \in \mathbb{R}^n : \|G(x) - G(y)\| \leq \rho \|x - y\|$$

Definice 2.10. Nechť je dáno zobrazení $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Bod $\tilde{x} \in \mathbb{R}^n$ se nazývá *pevný bod* zobrazení G jestliže $G(\tilde{x}) = \tilde{x}$.

Otázku existence pevného bodu zobrazení G řeší následující věta.



Obsah

64. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

**Věta 2.11.** (o pevném bodě)*Nechť*

- je dána množina $\Omega \subset \mathbb{R}^n$, $\Omega \neq \emptyset$, souvislá, ohraničená a uzavřená,
- je dáno zobrazení $G : \mathbb{R} \rightarrow \mathbb{R}^n$ kontraktivní v Ω ,
- postupné aproximace $x_k = G(x_{k-1})$, $k = 1, 2, \dots$ leží v Ω .

Pak

1. $\{x_k\}$ konverguje k $\bar{x}$ nezávisle na volbě počáteční aproximace x_0
t.j. $x_k \rightarrow \bar{x}$, $k \rightarrow \infty$,
2. $\bar{x} = G(\bar{x})$ v Ω ($\bar{x}$ je pevným bodem),
3. $\|\bar{x} - x_k\| \leq \frac{\rho^k}{1-\rho} \|x_1 - x_0\|$.

Pro zájemce

Výše uvedená věta je aplikací Banachovy věty o pevném bodě na reálný vektorový prostor. Banachovu větu o pevném bodě pro obecné metrické prostory lze nalézt ve skriptu [4].

Pro důkaz existence pevného bodu zobrazení G je tedy zásadní rozhodnout, zda-li je zobrazení G kontraktivní. To však je velmi netriviální úloha. Z tohoto důvodu vystačíme s postačující podmínkou, která nám taktéž zajistí existenci pevného bodu a všechna tvrzení Věty 2.11.

Obsah

65. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Lemma 2.12. (postačující podmínka)

Nechť $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\Omega \subset \mathbb{R}^n$ z věty 2.11. Nechť G je v Ω spojitá a nechť existuje G' v Ω a je spojitá. Nechť navíc

$$\|G'(x)\| \leq M < 1 \quad \text{na } \Omega$$

kde $M \in \mathbb{R}$ je konstanta.

Jestliže postupně aproximace $x_k = G(x_{k-1})$ leží v Ω , pak platí tvrzení Věty 2.11.

Hledání pevného bodu zobrazení $x = G(x)$ je totéž jako hledání kořene rovnice

$$F(x) = x - G(x) = 0.$$

Vždy můžeme rovnici $F(x) = 0$ přepsat na tvar $x = G(x)$, kde zobrazení G splňuje dané předpoklady a pro výpočet kořene rovnice použít postupné aproximace hledání pevného bodu. Otázka existence pevného bodu G , tedy i kořene rovnice $F(x) = 0$, je pak dána Větou 2.11 a 2.12.

Je zřejmé, že neexistuje pouze jedinný předpis a najít ten, který bude splňovat podmínky konvergence nemusí být zcela triviální.

Algoritmus 2**METODA PROSTÝCH ITERACÍ**

Input: x_0 , přesnost $\epsilon > 0$

```

1:  $x_1 := G(x_0)$ 
2:  $k := 1$ 
3: while  $\|x_k - x_{k-1}\| \geq \epsilon$  do
4:    $x_{k+1} := G(x_k)$ 
5:    $k := k + 1$ 
6: end while
```

Output: x_k



Obsah

66. strana ze 415



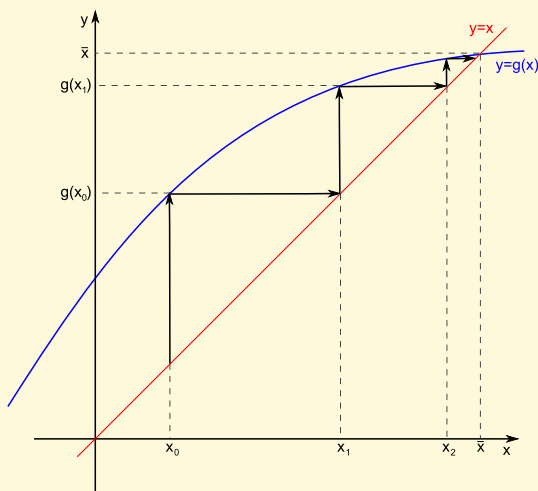
Zavřít dokument

Celá obrazovka / Okno

Poznámka 2.13. $f(x) = 0 \Leftrightarrow x = g(x)$, $f : \mathbb{R} \rightarrow \mathbb{R}, g : \mathbb{R} \rightarrow \mathbb{R}$

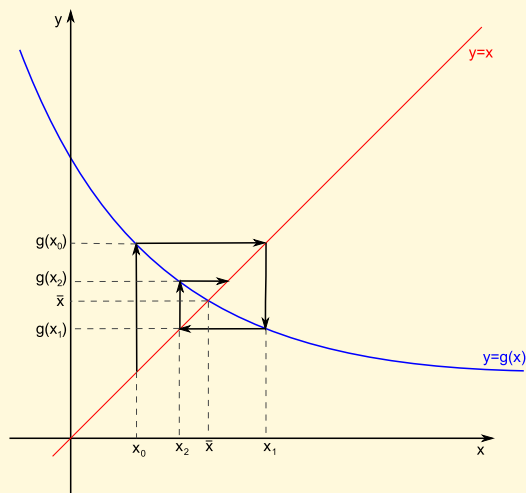
Postačující podmínka (viz věta 2.12) pak obsahuje odhad $|g'(x)| < M < 1$ na $\langle a, b \rangle \subset \mathbb{R}$

- $1 > g'(x) > 0$



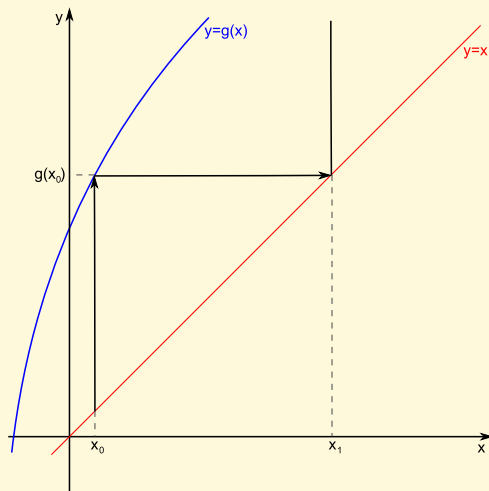
Obr. 2.6: Monotonní konvergence

- $-1 < g'(x) < 0$



Obr. 2.7: Oscilační konvergence

- $|g'(x)| > 1$



Obr. 2.8: Nekonverguje



Obsah

69. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Příklad 2.14. Vyčíslete hodnotu $\sqrt{a}$, $a \in \mathbb{R}^+$ s libovolnou přesností.

Řešení. Sestavme nejdříve funkci f popisující rovnici, jejíž řešení hledáme.

$$\begin{aligned} x &= \sqrt{a} \\ x^2 &= a \\ x^2 - a &= 0, & f(x) &= x^2 - a \\ f(x) &= 0 \end{aligned}$$

Hledejme vhodný předpis pro pevný bod ekvivalentní s původní rovnicí.

$$\begin{aligned} x^2 &= a \\ x &= \frac{a}{x} \\ 2x &= x + \frac{a}{x} \\ x &= \frac{x + \frac{a}{x}}{2}, & g(x) &= \frac{x + \frac{a}{x}}{2} \\ x &= g(x) \end{aligned}$$

Ověřme splnění podmínek konvergence - předpokladů Věty 2.12

$$\begin{aligned} g'(x) &= \frac{1 + \frac{-a}{x^2}}{2} = \frac{1}{2} \frac{x^2 - a}{x^2}, & x &> 0 \\ g'(x) &> 0 & \text{pro } x &> \sqrt{a} \\ g'(x) &< 0 & \text{pro } x &< \sqrt{a} \end{aligned}$$

Extrémy $g'(x)$:

$$\begin{aligned} \varphi(x) &:= g'(x), & \varphi'(x) &= \frac{1}{2} \frac{2a}{x^3} = \frac{a}{x^3} \\ \varphi'(x) &> 0 & \text{je – li } a &> 0 \\ \varphi'(x) &< 0 & \text{je – li } a &< 0 \end{aligned}$$

Obsah

70. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Funkce g' je tedy pro $a > 0$ rostoucí.

$$\begin{aligned}\min_{x \in \langle \alpha, \beta \rangle} \{g'(x)\} &= g'(\alpha) \\ \max_{x \in \langle \alpha, \beta \rangle} \{g'(x)\} &= g'(\beta) \\ |g'(x)| &< \max\{|g'(\alpha)|, |g'(\beta)|\}, x \in \langle \alpha, \beta \rangle\end{aligned}$$

Například pro konkrétní hodnotu úlohy $a = 2$ dostáváme

$$x \in \langle 1, 2 \rangle, \quad g'(2) = \frac{1}{4}, \quad g'(1) = -\frac{1}{2}, \quad |g'(x)| = \left| \frac{1}{2} \frac{x^2 - 2}{x^2} \right| < \frac{1}{2}.$$

Předpis

$$x_{k+1} = g(x_k) = \frac{1}{2} \left(x_k + \frac{2}{x_k} \right)$$

bude tedy konvergovat pro libovolné $x_0 \in \langle 1, 2 \rangle$.

Volme proto počáteční aproximaci například $x_0 = 2$.

Při požadované přesnosti $\epsilon = 10^{-5}$ je řešení nalezeno po pěti iteracích, viz následující tabulka.

k	x	$\ x_k - x_{k-1}\ $
0	2	
1	1.5	0.5
3	1.4167	0.0833
4	1.4142	0.0025
5	1.4142	$2.1239 \cdot 10^{-6}$



Obsah

71. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 2.15. Označme chybu

$$e_k = x_k - \bar{x}, \quad \bar{x} = g(\bar{x}).$$

Pak lze upravit

$$\begin{aligned} e_{k+1} &= x_{k+1} - \bar{x} = g(x_k) - g(\bar{x}) \\ &= (x_k - \bar{x})g'(\bar{x}) + (x_k - \bar{x})^2 \frac{g''(\bar{x})}{2!} + (x_k - \bar{x})^3 \frac{g'''(\bar{x})}{3!} + \dots \\ &= e_k g'(\bar{x}) + e_k^2 \frac{g''(\bar{x})}{2!} + e_k^3 \frac{g'''(\bar{x})}{3!} + \dots \end{aligned}$$

Řád konvergence tedy závisí na derivaci g v bodě $\bar{x}$.

Je-li

$$\begin{aligned} g^{(k)}(\bar{x}) &= 0, \quad \forall k = 1, 2, \dots, p-1, \\ g^{(p)}(\bar{x}) &\neq 0, \end{aligned}$$

tak je metoda prostých iterací řádu p .

Poznámka 2.16. V případě, že $x = G(x)$, $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$, G je diferencovatelná v oblasti $\Omega \subset \mathbb{R}^n$ je

$$G'(x) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial g_n}{\partial x_1}(x) & \dots & \frac{\partial g_n}{\partial x_n}(x) \end{pmatrix}, \quad G(x) = \begin{pmatrix} g_1(x) \\ \vdots \\ g_n(x) \end{pmatrix}$$

Odhad $\|G'(x)\| < M$ je pak odhad normy matice.

Vypočíst normu takovéto matice nemusí být vůbec snadná úloha!



Obsah

72. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2.4. Newtonova metoda

Původní řešenou rovnici

$$f(x) = 0, \quad f: \mathbb{R} \rightarrow \mathbb{R} \quad (2.3)$$

lze upravit na tvar

$$x = x + \lambda(x) \cdot f(x), \quad (2.4)$$

kde $\lambda: \mathbb{R} \rightarrow \mathbb{R}$ je libovolná nenulová funkce.

Z rovnice (2.4) lze odvodit předpis kontraktivního zobrazení pro metodu prostých iterací s pevným bodem řešení původní rovnice (2.3) (tj. $G(x) = x$)

$$G(x) = x + \lambda(x) \cdot f(x). \quad (2.5)$$

Budeme-li požadovat, aby byla metoda řádu 2, pak

$$\begin{aligned} G'(\bar{x}) &= 0, \\ G'(x) &= 1 + \lambda'(x)f(x) + \lambda(x)f'(x), \\ G'(\bar{x}) &= 1 + \lambda'(\bar{x})f(\bar{x}) + \lambda(\bar{x})f'(\bar{x}) = 1 + \lambda(\bar{x})f'(\bar{x}). \end{aligned}$$

Odtud $\lambda = -\frac{1}{f'(\bar{x})}$ pro $f'(\bar{x}) \neq 0$.

Předpokládejme tedy, že $f'(x)$ je nenulová, pak můžeme označit

$$G(x) = x - \frac{f(x)}{f'(x)}.$$

Metoda, která hledá pevný bod takovéto funkce se pak nazývá *Newtonova metoda* pro řešení rovnice (2.3) a má předpis

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \quad (2.6)$$



Obsah

73. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Algoritmus 3

NEWTONOVA METODA

Input: x_0 , přesnost $\epsilon > 0$

```
1:  $x_1 := x_0 - f(x_0)/f'(x_0)$ 
2:  $k := 1$ 
3: while  $\|x_k - x_{k-1}\| \geq \epsilon$  do
4:    $x_{k+1} := x_k - f(x_k)/f'(x_k)$ 
5:    $k := k + 1$ 
6: end while
```

Output: x_k

Poznámka 2.17. Předpis Newtonovy metody lze snadno získat i z aproximace řešení pomocí Taylorova polynomu. Jelikož

$$f(\bar{x}) \approx f(x_k) + (\bar{x} - x_k)f'(x_k) \quad (\text{plyne z Taylorovy věty})$$

a jelikož požadujeme splnění řešení rovnice $f(\bar{x}) = 0$, pak

$$f(x_k) + (\bar{x} - x_k)f'(x_k) = 0 \Rightarrow \bar{x} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Iteračně pak získáme předpis (2.6).



Obsah

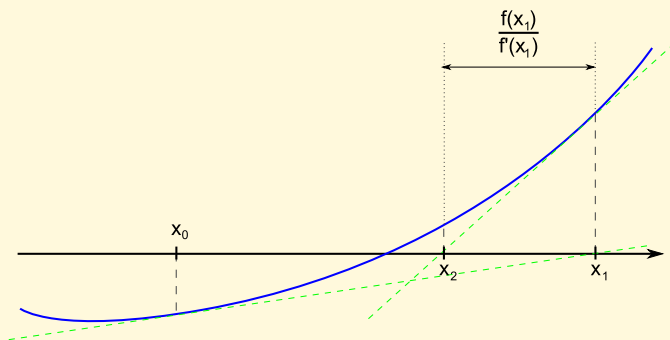
74. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Newtonově metodě se taktéž říká metoda tečen. Volba tohoto názvu je patrná z geometrické interpretace, kterou naleznete na obrázku.



Obr. 2.9: Newtonova metoda

Podmínky konvergence Newtonovy metody zaručuje následující lemma.

Lemma 2.18. *Nechť funkce $f : \mathbb{R} \rightarrow \mathbb{R}$ vyhovuje následujícím podmínkám na $\langle a, b \rangle$:*

1. $f(a) \cdot f(b) < 0$
2. $f'(x) \neq 0, \forall x \in \langle a, b \rangle$
3. f'' nemá znaménkové změny na $\langle a, b \rangle$
4. $|f(a)/f'(a)| < b - a$
 $|f(b)/f'(b)| < b - a$

Pak pro libovolné $x_0 \in \langle a, b \rangle$ konverguje Newtonova metoda pro řešení rovnice $f(x) = 0$ na $\langle a, b \rangle$.



Obsah

76. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2.4.1. První modifikace Newtonovy metody

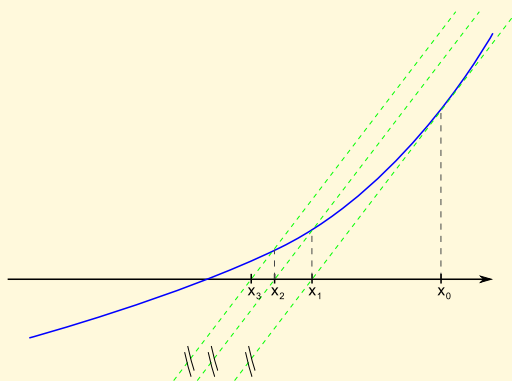
V každém kroce Newtonovy metody musíme kromě funkční hodnoty $f(x_k)$ vyhodnocovat i funkční hodnotu její první derivace $f'(x)$. To však může být výpočetně velmi náročné. Z tohoto důvodu se nám může vyplatit vyhodnotit tuto derivaci jen jednou za čas a počítat několik kroků se stejnou, byť nepřesnou, hodnotou.

Předpis má potom tvar

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_l)},$$

kde $f'(x_l) = f'(x_k)$, je-li k dělitelné m .

Geometrickou interpretaci této modifikace lze nalézt na obrázku.



Obr. 2.10: První modifikace Newtonovy metody



Obsah

77. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2.4.2. Druhá modifikace Newtonovy metody

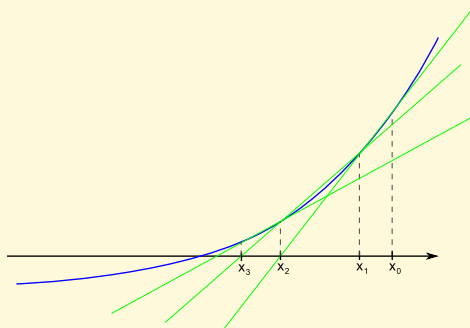
Vyhodnocení první derivace lze rovněž obejít její aproximací předpisem

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}},$$

pak předpis metody má tvar

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}.$$

Tato metoda se nazývá *metoda sečen*.



Obr. 2.11: Druhá modifikace Newtonovy metody

2.4.3. Newtonova metoda pro soustavy

Uvažujme nyní soustavu

$$F(x) = 0, x \in \Omega,$$

tj.

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0, \\ f_2(x_1, \dots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, \dots, x_n) &= 0. \end{aligned}$$

Pak v předpisu (2.5) bude $\lambda(x) = -[F'(x)]^{-1}$, kde

$$F'(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(x) & \dots & \frac{\partial f_n}{\partial x_n}(x) \end{bmatrix}.$$

Pak předpis pro Newtonovu formuli bude vypadat následovně

$$x_{k+1} = x_k - [F'(x_k)]^{-1} \cdot F(x_k), \quad x_{k+1} = (x_{k+1}^1, x_{k+1}^2, \dots, x_{k+1}^n)$$

Uvedený předpis však znamená v každé iteraci sestavit matici $F'(x_k)$.



Obsah

79. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pokud označíme přírutek řešení v k -tém kroce $\Delta x_k = x_{k+1} - x_k$ a vynásobíme-li předpis maticí $F'(x_k)$, dostaneme nový předpis, ve kterém řešíme soustavu

$$F'(x_k)\Delta x_k = -F(x_k), x_{k+1} = x_k + \Delta x_k.$$

V uvedeném předpise je jedinou neznámou přírutek Δx_k . Jeho výpočet je řešením soustavy lineárních rovnic s maticí $F'(x_k)$ a pravou stranou $-F(x_k)$. Této metodě se taktéž říká *Newtonova-Raphsonova metoda*. V této metodě se jako velmi výhodná ukazuje její první modifikace. Při jejím použití se totiž, řeší několik soustav lineárních rovnic se stejnou maticí. K efektivnímu řešení je pak možné použít LU rozklad, který provedeme pouze jednou za čas a soustavu pro různé pravé strany řešíme pouze dopřednou a zpětnou substitucí.

Algoritmus 4**NEWTONOVA METODA PRO SOUSTAVY**

Input: x_0 , přesnost $\epsilon > 0$

- 1: nalezni Δx_0 , které řeší soustavu $F'(x_0)\Delta x_0 = -F(x_0)$
- 2: $x_1 := x_0 + \Delta x_0$
- 3: $k := 1$
- 4: **while** $\|x_k - x_{k-1}\| \geq \epsilon$ **do**
- 5: nalezni Δx_k , které řeší soustavu $F'(x_k)\Delta x_k = -F(x_k)$
- 6: $x_{k+1} := x_k + \Delta x_k$
- 7: $k := k + 1$
- 8: **end while**

Output: x_k



Obsah

80. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 2.19. Pomocí Newtonovy metody nalezněte průsečík křivky

$$\sigma : x^2 + y = 0$$

a elipsy

$$\rho : x^2 + 4y^2 = 1 .$$

Řešení. Úkolem je tedy řešit soustavu

$$F(x) = o ,$$

kde

$$F(x) \stackrel{\text{def}}{=} \begin{bmatrix} f_1(x, y) \\ f_2(x, y) \end{bmatrix} = \begin{bmatrix} x^2 + y \\ x^2 + 4y^2 - 1 \end{bmatrix} .$$

Pro předpis Newtonovy metody bude nutno spočítat kvadratickou formu

$$F'(x) = \begin{bmatrix} \frac{\partial f_1(x, y)}{\partial x} & \frac{\partial f_1(x, y)}{\partial y} \\ \frac{\partial f_2(x, y)}{\partial x} & \frac{\partial f_2(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x & 1 \\ 2x & 8y \end{bmatrix} .$$

Následně inverzi

$$\begin{aligned} \left[\begin{array}{cc|cc} 2x & 1 & 1 & 0 \\ 2x & 8y & 0 & 1 \end{array} \right] &\sim \left[\begin{array}{cc|cc} 2x & 1 & 1 & 0 \\ 0 & 1-8y & 1 & -1 \end{array} \right] \sim \left[\begin{array}{cc|cc} 2x & 1 & 1 & 0 \\ 0 & 1 & \frac{1}{1-8y} & -\frac{1}{1-8y} \end{array} \right] \sim \\ &\sim \left[\begin{array}{cc|cc} 2x & 0 & 1 - \frac{1}{1-8y} & \frac{1}{1-8y} \\ 0 & 1 & \frac{1}{1-8y} & -\frac{1}{1-8y} \end{array} \right] \sim \left[\begin{array}{cc|cc} 1 & 0 & \frac{8y}{2x(1-8y)} & \frac{1}{2x(1-8y)} \\ 0 & 1 & \frac{1}{1-8y} & -\frac{1}{1-8y} \end{array} \right] , \end{aligned}$$



Obsah

81. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

tedy

$$-[F'(x)]^{-1} = \frac{1}{2x(8y-1)} \begin{bmatrix} 8y & 1 \\ 2x & -2x \end{bmatrix}.$$

Pak předpis pro Newtonovu metodu má tvar

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \frac{1}{2x_k(8y_k-1)} \begin{bmatrix} 8y_k & 1 \\ 2x_k & -2x_k \end{bmatrix} \cdot \begin{bmatrix} x_k^2 + y_k \\ x_k^2 + 4y_k^2 - 1 \end{bmatrix}.$$

Při přesnosti $\epsilon = 10^{-5}$ a volbě počáteční aproximace $[x_0, y_0]^T = [-2, -1]^T$ je algoritmus je ukončen po 4 iteracích, viz Obrázek 2.12.



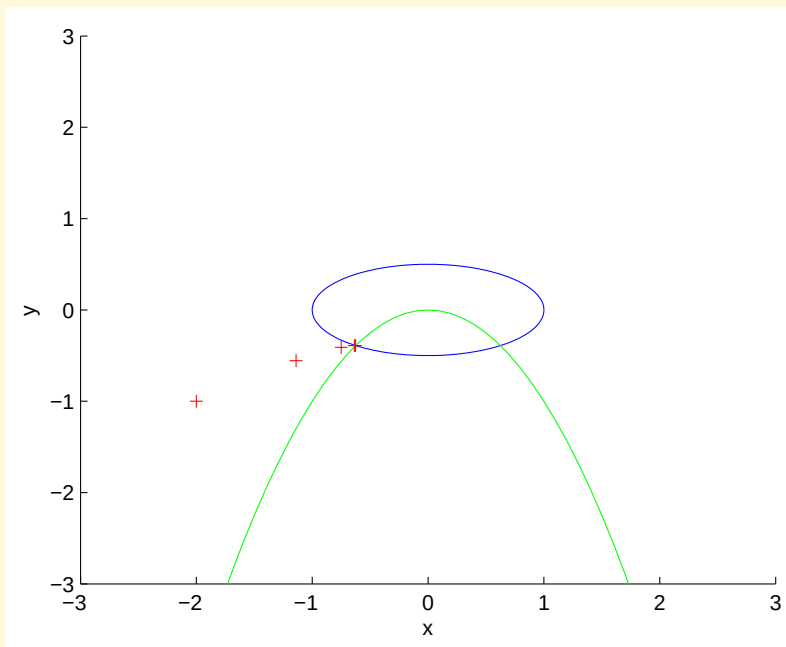
Obsah

82. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 2.12: Konvergence Newtonovy metody v Příkladě 2.19



Obsah

83. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

2.5. Test

Následující test zahájíte kliknutím na tlačítko „Začátek testu“. U otázek, kde lze volit jen jednu odpověď (test nedovolí zaškrtnout více odpovědí), je správná odpověď bodována počtem bodů uvedených v závorce u zadání a špatná odpověď je bodována 0 body. U otázek, kde lze volit více správných odpovědí, je součet bodů správných odpovědí uveden v závorce u zadání a za každou špatnou odpověď bude odečten jeden bod.

1. (4b.) Necht je dána rovnice $f(x) = 0$, kde $f : \mathbb{R} \rightarrow \mathbb{R}$. Prvek $\bar{x} \in \mathbb{R}$ takový, že $f(\bar{x}) = 0$ nazýváme

kořen rovnice

stacionární bod rovnice

inflexní bod rovnice

pevný bod rovnice

minimum funkce f

2. (4b.) Necht funkce f je spojitá na intervalu $\langle a, b \rangle \subset \mathbb{R}$. Pak jestliže $f(a) < 0 \wedge f(b) > 0$, pak rovnice $f(x) = 0$

nemá žádný kořen

má právě jeden kořen

má alespoň jeden kořen

má právě dva kořeny

o kořenech dané rovnice nelze nic tvrdit



Obsah

84. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3. (4b.) Necht $\{x_n\}$ je posloupnost aproximací čísla x , $x_n \rightarrow \bar{x}$, $n \rightarrow \infty$. Říkáme, že iterační metoda je řádu Γ , jestliže

pro libovolné aproximace x_k a x_{k+1} (kde $k \in \mathbb{N}_0$) a řešení $\bar{x}$ platí

$$|\bar{x} - x_{n+1}| \leq \Gamma |\bar{x} - x_n|$$

pro libovolné $C > 0$, libovolné aproximace x_k a x_{k+1} (kde $k \in \mathbb{N}_0$) a řešení $\bar{x}$ platí

$$|\bar{x} - x_{n+1}| \leq C |\bar{x} - x_n|^\Gamma$$

existuje konstanta $C > 0$ taková, že pro libovolné aproximace x_k a x_{k+1} (kde $k \in \mathbb{N}_0$) a řešení $\bar{x}$ platí

$$|\bar{x} - x_{n+1}| \leq C |\bar{x} - x_n|^\Gamma$$

existuje právě jedna konstanta $C > 0$ taková, že pro libovolné aproximace x_k a x_{k+1} (kde $k \in \mathbb{N}_0$) a řešení $\bar{x}$ platí

$$|\bar{x} - x_{n+1}| \leq C |\bar{x} - x_n|^\Gamma$$



Obsah

85. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4. (4b.) Necht je daná rovnice $f(x) = 0$. Z procesu separace kořenů víme, že v intervalu $\langle 1, 2 \rangle$ leží právě jeden kořen. Pro nalezení tohoto kořenu použijeme metodu půlení intervalu. Maximálně po kolika iteracích bude metoda ukončena, pokud hledáme řešení s přesností $\epsilon = 10^4$?

po 10-ti iteracích,

po 13-ti iteracích,

po 14-ti iteracích,

po 15-ti iteracích,

po 16-ti iteracích,

po 20-ti iteracích,

nelze určit.

5. (4b.) Metoda půlení intervalu je
nultého řádu,

prvního řádu,

druhého řádu,

nelze dokázat řád konvergence.



Obsah

86. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



6. (4b.) Zobrazení $H : \mathbb{R}^n \rightarrow \mathbb{R}^n$ se nazývá *kontraktivní*, jestliže

pro libovolné $0 < \rho < 1$ platí

existuje $0 < \rho < 1$ takové, že

$$\forall x, y \in \mathbb{R}^n : \|H(x) - H(y)\| \leq \rho \|x - y\| \quad \forall x, y \in \mathbb{R}^n : \|H(x) - H(y)\| \leq \rho \|x - y\|$$

pro libovolné $0 < \rho < 1$ platí

existuje $0 < \rho < 1$ takové, že

$$\forall x, y \in \mathbb{R}^n : \|H(x) - H(y)\| \leq \|x - y\|^\rho \quad \forall x, y \in \mathbb{R}^n : \|H(x) - H(y)\| \leq \|x - y\|^\rho$$

7. (4b.) Nechť $h : \mathbb{R} \rightarrow \mathbb{R}$ je spojitá funkce na intervalu $\langle a, b \rangle$ a má na intervalu $\langle a, b \rangle$ spojitou první derivaci. Metoda prostých iterací pro hledání kořene na intervalu $\langle a, b \rangle$ je konvergentní, pokud navíc

existuje konstanta $M \in \mathbb{R}$ taková, že

všechny aproximace definované

předpisem $x_k = h(x_{k-1})$ leží v $\langle a, b \rangle$,

$$\forall x \in \langle a, b \rangle : |h'(x)| \leq M < 1$$

a všechny aproximace definované

předpisem $x_k = h(x_{k-1})$ leží v $\langle a, b \rangle$,

zobrazení $f(x) = x - h(x)$ je

kontraktivní na intervalu $\langle a, b \rangle$,

zobrazení $h(x)$ je kontraktivní na

intervalu $\langle a, b \rangle$.

Obsah

87. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8. (4b.) Newtonova metoda je

nultého řádu,

prvního řádu,

druhého řádu,

nelze dokázat řád konvergence.

9. (4b.) Nechť je dána funkce $f : \mathbb{R} \rightarrow \mathbb{R}$. Vyberte vhodnou kombinaci postačujících podmínek na tuto funkci tak, aby pro libovolnou volbu počáteční aproximaci $x_0 \in \langle a, b \rangle$ konvergovala Newtonova metoda pro řešení rovnice $f(x) = 0$ na $\langle a, b \rangle$.

$$f'(x) \neq 0, \forall x \in \langle a, b \rangle,$$

$$\exists \tilde{x} \in \langle a, b \rangle : f'(\tilde{x}) = 0,$$

f'' nemá znaménkové změny na $\langle a, b \rangle$,

$$|f(a)/f'(a)| < b - a$$

$$|f(b)/f'(b)| < b - a,$$

$$f(a).f(b) > 0,$$

f' nemá znaménkové změny na $\langle a, b \rangle$,

$$f(a).f(b) < 0,$$

$$|f(a)/f'(b)| < b - a$$

$$|f(b)/f'(a)| < b - a,$$



Obsah

88. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

10. (4b.) Při použití Newtonovy metody pro řešení rovnice $f(x) = 0$ lze derivaci funkce f aproximovat podílem

$$f'(x_k) \approx \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

$$f'(x_k) \approx \frac{f(x_k) - x_{k-1}}{x_k - f(x_{k-1})}$$

$$f'(x_k) \approx \frac{f(x_{k-1}) - x_k}{x_{k-1} - f(x_k)}$$

11. (4b.) Mezi nevýhody metody prostých iterací patří

pomalá konvergence,

je třeba dokázat konvergenci složitým způsobem,

nelze nalézt předpis pro řád konvergence vyšší než 1,

nelze nalézt předpis pro řád konvergence vyšší než 2.

12. (4b.) Mezi nevýhody metody půlení intervalu patří

řád konvergence je 2,

je třeba dokázat konvergenci složitým způsobem,

silné podmínky pro konvergenci,

konvergence je pouze lineární.



Obsah

89. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

13. (4b.) Newtonova metoda se také nazývá metoda tečen. Proč?

Všechny tečny sestrojené v jednotlivých iteracích jsou navzájem rovnoběžné.

Vždy po sobě jsou iterace splňují tečnicovou rovnici.

Z geometrické interpretace. Průsečík osy x a tečny k funkci f sestrojené v bodě aktuální iterace určuje následující iteraci.

Všechny tečny sestrojené v jednotlivých iteracích jsou navzájem kolmé.

Z geometrické interpretace. Průsečík osy x a přímkou rovnoběžnou s tečnou k funkci f sestrojené v bodě aktuální iterace určuje následující iteraci.

14. (3b.) Necht je dána rovnice $x^2 = \sin x$. Nalézt kořeny této rovnice s danou přesností lze pomocí

metody tabelace funkce,

metody půlení intervalu,

metody prostých iterací,

metody grafické separace kořenů,

metody odhadu kořenů,

newtonovy metody.

[Obsah](#)[90. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka / Okno](#)

15. (3b.) Úkolem je vyčíslit hodnotu $\sqrt{5}$ pomocí metody prostých iterací. Jaký je předpis zobrazení, pro které bude metoda konvergovat k hledané hodnotě?

$$x_{k+1} = \sqrt{5x_k},$$

$$x_{k+1} = x_k^2 + 5,$$

$$x_{k+1} = \frac{5}{x_k^2},$$

$$x_{k+1} = (x_k + \frac{2}{x})/2,$$

$$x_{k+1} = (x_k - \frac{5}{x})/5,$$

$$x_{k+1} = (x_k + \frac{5}{x})/5,$$

$$x_{k+1} = (x_k + \frac{5}{x})/2,$$

$$x_{k+1} = (x_k + \frac{2}{x})/5.$$

Správně zodpovězené otázky:

Získané body:

Procento úspěšnosti:

Pro ukončení testu je třeba kliknout na tlačítko „Konec testu“. Opravení testu se provede kliknutím na tlačítko „Výsledky“, správné odpovědi budou označeny zeleně a červeně budou označeny odpovědi chybné. Při použití otázky s tvořenou odpovědí se správná odpověď zobrazí v rámečku umístěném vpravo dole v navigačním panelu po kliknutí na tlačítko „Odpověď“.



Obsah

91. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Kapitola 3

Iterační řešení soustav lineárních rovnic

Průvodce studiem

V této kapitole si představíme několik základních metod numerického řešení soustav lineárních rovnic.

Řada praktických úloh popisující lineární závislost výstupních dat na datech vstupních vede na řešení soustav lineárních rovnic. Prakticky všechny fyzikální jevy popisované parciálními diferenciálními rovnicemi vedou na tuto úlohu. Na konci předchozí kapitoly jsme si dokonce ukázali, že i numerické metody pro řešení soustav nelineárních rovnic se rovněž převádí na posloupnost řešení soustav lineárních rovnic. Zvládnutí efektivního řešení tohoto problému je tedy klíčové.

S analytickými metodami řešení soustav lineárních rovnic jako je Gaussova eliminační metoda či LU rozklad se čtenář může seznámit ve skriptech [2]. Tyto metody však v současnosti ustupují

Obsah

92. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

do pozadí, neboť pro řešení velmi rozsáhlých úloh s velkým počtem neznámých jsou prakticky nepoužitelné díky své velké výpočetní a paměťové náročnosti. Z tohoto důvodu jsou nahrazovány metodami iteračními, které mají výrazně menší paměťové nároky a v určitých případech jsou i výrazně rychlejší. Hlavní výhodou iteračních metod je však jejich snadná paralelizace pro nasazení na velmi výkonné výpočetní prostředky.

V této kapitole se proto seznámíme se základními iteračními metodami a postupy, kterak zefektivnit rychlost jejich konvergence.

[Obsah](#)

93. strana ze 415

[Zavřít dokument](#)[Celá obrazovka/Okno](#)

3.1. Normy vektorů a matic

Úvodem této kapitoly si připomeňme několik základních pojmů Lineární algebry (viz [2]), které při popisu a analýze iteračních metod řešení soustav lineárních rovnic budeme často používat.

Definice 3.1. Necht $\mathcal{V}$ je vektorový prostor. Zobrazení $\|\cdot\| : \mathcal{V} \rightarrow \mathbb{R}_0^+$ nazýváme *normou* jestliže platí

1. $\|x\| \geq 0, \|x\| = 0 \Leftrightarrow x = 0,$
2. $\|\alpha x\| = |\alpha| \cdot \|x\|,$
3. $\|x + y\| \leq \|x\| + \|y\|.$

Necht $\mathcal{V} = \mathbb{R}^n$. Pak prvky vektorového prostoru jsou n -dimenzionální aritmetické vektory a hovoříme o *vektorových normách*. Uvedme si několik příkladů vektorových norem:

- $\|x\|_\infty = \max\{|x_i|, i = 1, 2, \dots, n\}$
- $\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}$
- $\|x\|_1 = \sum_{i=1}^n |x_i|$



Obsah

94. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 3.2. Načrtněte v $\mathbb{R}^2$ množiny

$$\begin{aligned}\Omega_1 &:= \{x \in \mathbb{R}^2 : \|x\|_1 = 1\} , \\ \Omega_2 &:= \{x \in \mathbb{R}^2 : \|x\|_2 = 1\} , \\ \Omega_\infty &:= \{x \in \mathbb{R}^2 : \|x\|_\infty = 1\} .\end{aligned}$$



Obsah

95. strana ze 415



Zavřít dokument

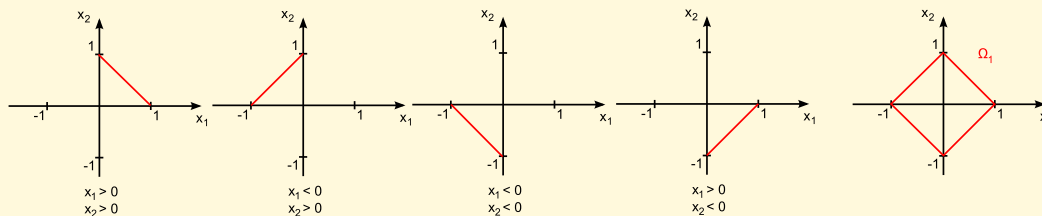
Celá obrazovka/Okno

Řešení. 1. Ω_1 - dle definice $\|x\|_1 = 1 \Rightarrow |x_1| + |x_2| = 1$.

Rozdělme naše úvahy na jednotlivé kvadranty:

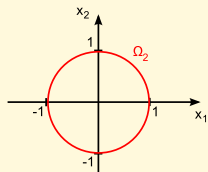
- pro $x_1 > 0 \wedge x_2 > 0$ - přímka s restrikcí na první kvadrant $x_1 + x_2 = 1$,
- pro $x_1 < 0 \wedge x_2 > 0$ - přímka s restrikcí na druhý kvadrant $-x_1 + x_2 = 1$,
- pro $x_1 < 0 \wedge x_2 < 0$ - přímka s restrikcí na třetí kvadrant $-x_1 - x_2 = 1$,
- pro $x_1 > 0 \wedge x_2 < 0$ - přímka s restrikcí na čtvrtý kvadrant $x_1 - x_2 = 1$.

Spojením řešení v jednotlivých kvadrantech získáme celkové řešení (viz Obrázek 3.1).



Obr. 3.1: Konstrukce množiny Ω_1 .

2. Ω_2 - dle definice $\|x\|_2 = 1 \Rightarrow x_1^2 + x_2^2 = 1$, což je kružnice s poloměrem 1 (viz Obrázek 3.2).



Obr. 3.2: Konstrukce množiny Ω_2 .



Obsah

97. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

3. Ω_∞ - Dle definice $\|x\|_\infty = 1 \Rightarrow \max\{|x_1|, |x_2|\} = 1$.

Uvažujme dva případy (ve kterých je maximum jeden z prvků x_1, x_2):

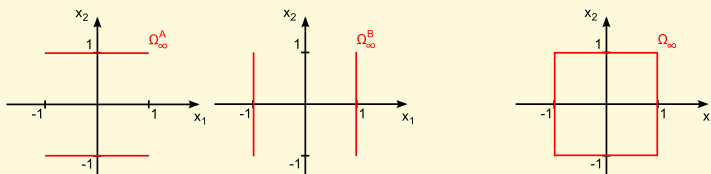
- pokud $|x_1| \leq |x_2| = 1$, pak se jedná o množinu bodů

$$\{[x_1, x_2] \in \mathbb{R}^2 : x_1 \in \langle -1, 1 \rangle \wedge x_2 \in \{-1, 1\}\} \stackrel{\text{ozn}}{=} \Omega_\infty^A$$

- pokud $|x_2| \leq |x_1| = 1$, pak se jedná o množinu bodů

$$\{[x_1, x_2] \in \mathbb{R}^2 : x_1 \in \{-1, 1\} \wedge x_2 \in \langle -1, 1 \rangle\} \stackrel{\text{ozn}}{=} \Omega_\infty^B$$

Spojením těchto dvou případů získáme celkové řešení (viz Obrázek 3.3).



Obr. 3.3: Konstrukce množiny Ω_∞ .

Poznámka 3.3. Pro $\mathcal{V} = \mathbb{R}^{m,n}$ nazýváme normu *normou maticovou*.

Příklad 3.4. Zobrazení $\|\cdot\| : \mathbb{R}^{m,n} \rightarrow \mathbb{R}$ definované předpisem

$$\|A\|_F := \left(\sum_{i,j} |a_{ij}|^2 \right)^{\frac{1}{2}}$$

nazýváme *Frobeniova norma*.



Obsah

99. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Definice 3.5. Necht' je dána vektorová norma $\|\cdot\|_V$ na prostoru $\mathbb{R}^n$. Maticovou normu odpovídající vektorové normě $\|\cdot\|_M$ definujeme následujícím předpisem

$$\|A\|_M := \max_{x \neq 0} \left\{ \frac{\|Ax\|_V}{\|x\|_V} \right\}.$$

Uvedme maticové normy odpovídající dříve uvedeným vektorovým normám

- $\|A\|_\infty := \max\left\{\sum_{k=1}^n |a_{jk}|, j = 1, \dots, n\right\}$ odpovídá $\|\cdot\|_\infty$ a nazývá se *řádková norma*,
- $\|A\|_1 := \max\left\{\sum_{j=1}^n |a_{jk}|, k = 1, \dots, n\right\}$ odpovídá $\|\cdot\|_1$ a nazývá se *sloupcová norma*,
- $\|A\|_2 := \sqrt{\rho(A^T A)} = \sqrt{\rho(A A^T)} = \rho(A)$, kde A je normální matice¹², odpovídá $\|\cdot\|_2$ a nazývá se *spektrální norma*.

Lemma 3.6. Necht' $\|\cdot\|_M$ značí maticovou a $\|\cdot\|_V$ odpovídající vektorovou normu. Pak platí

$$\begin{aligned} \forall A \in \mathbb{R}^{n \times n} \quad \forall x \in \mathbb{R}^n : \quad & \|Ax\|_V \leq \|A\|_M \cdot \|x\|_V, \\ \forall A, B \in \mathbb{R}^{n \times n} : \quad & \|AB\|_M \leq \|A\|_M \cdot \|B\|_M. \end{aligned}$$

¹Matice A se nazývá *normální*, pokud $AA^T = A^T A$

² $\rho(A)$ je spektrální poloměr matice A

Obsah

100. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Uvedme si ještě jednu důležitou charakteristiku matic, kterou je tzv. *číslo podmíněnosti matice*. V dalším textu ukážeme, že rychlost konvergence některých metod řešení soustav lineárních rovnic souvisí s tímto číslem.

Definice 3.7. Nechť $A \in \mathbb{R}^{n,n}$ je regulární. Pak *číslem podmíněnosti matice* rozumíme číslo

$$\text{cond}(A) \stackrel{\text{def}}{=} \|A\| \cdot \|A^{-1}\| .$$

Spektrálním číslem podmíněnosti matice rozumíme číslo

$$\kappa(A) \stackrel{\text{def}}{=} \rho(A) \rho(A^{-1}) .$$

Poznámka 3.8. Je-li navíc matice A symetrická a pozitivně definitní, pak

$$\kappa(A) = \text{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}} ,$$

kde $\lambda_{\max}$, $\lambda_{\min}$ jsou největší a nejmenší vlastní čísla matice A . Očividně

$$0 < \lambda_{\min} \leq \lambda_{\max} \quad \Rightarrow \quad \kappa(A) \geq 1 .$$



Obsah

101. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3.2. Lineární metody

Uvažujme soustavu lineárních rovnic

$$Ax = b, \quad (3.1)$$

kde

- $A \in \mathbb{R}^{n,n}$ je matice soustavy,
- $b \in \mathbb{R}^n$ je vektor pravých stran,
- $x \in \mathbb{R}^n$ je vektor neznámých.

Je-li matice soustavy regulární, pak existuje jediné řešení $\bar{x}$ takové, že $A\bar{x} = b$. Jinými slovy, je-li matice A regulární, existuje *právě jedno řešení* $\bar{x} = A^{-1}b$.

Zvolme počáteční aproximaci řešení soustavy $x_0 \neq \bar{x}$. Naším úkolem je sestavit posloupnost aproximací

$$x_0, x_1, \dots, x_k, x_{k+1}, \dots$$

konvergující k řešení $\bar{x}$, tedy

$$\lim_{k \rightarrow \infty} x_k = \bar{x}.$$

V následujícím výkladu budeme pátrat po předpisech zobrazení, které každé předešlé aproximaci přiřadí novou aproximaci řešení soustavy $\bar{x}$. Toto zobrazení budeme nazývat *iterační metoda*.



Obsah

102. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Definice 3.9. *Iterační metoda* je zobrazení $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$x_{k+1} = \Phi(x_k), \quad k \geq 0.$$

kde x_k je k -tá iterace.

Pokud navrhujeme nějakou iterační metodu, bude nutné zajistit, aby metoda konvergovala i k řešení soustavy lineárních rovnic pro kterou byla navržena. O takových metodách pak říkáme, že jsou *konzistentní* s danou soustavou lineárních rovnic.

Definice 3.10. Iterační metoda Φ se nazývá *konzistentní* se soustavou lineárních rovnic $Ax = b$, jestliže pro každou pravou stranu $b \in \mathbb{R}^n$ všechna řešení $\bar{x}$ jsou *pevným bodem* Φ , tj.

$$\bar{x} = \Phi(\bar{x}).$$

Také je nutno, aby se metoda postupnými iteracemi vždy blížila k řešení, a to nezávisle na volbě počáteční aproximace x_0 .

Definice 3.11. Iterační metoda Φ se nazývá *konvergentní*, jestliže pro $\forall b \in \mathbb{R}^n$ je limita

$$\lim_{k \rightarrow \infty} x_k = \bar{x}$$

nezávislá na počáteční aproximaci x_0 .

Obsah

103. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

V následujícím textu se omezíme na nejjednodušší třídu iteračních metod, které nazýváme *lineární iterační metody*. Na těchto metodách si ukážeme podmínky konzistence a konvergence těchto metod a uvedeme si příklady nejběžněji používaných metod.

Definice 3.12. Iterační metoda se nazývá *lineární*, jestliže

$$\Phi(x) = Mx + Nb$$

kde $M, N \in \mathbb{R}^{n \times n}$ jsou regulární matice.

Lineární iterační metoda má pak předpis

$$x_{k+1} = Mx_k + Nb, \quad k \geq 0.$$

Věta 3.13. *Lineární iterační metoda je konzistentní se soustavou lineárních rovnic $Ax = b$, právě když*

$$M = I - NA.$$

Poznámka 3.14. Konzistentní lineární iterační metody mají tvary

- 1) $x_{k+1} = x_k - N(Ax_k - b)$,
- 2) $W(x_k - x_{k+1}) = Ax_k - b, N = W^{-1}$.

Lemma 3.15. Chyba lineární iterační metody v k -té iteraci, tj.

$$e_k := \bar{x} - x_k$$

je rovna

$$e_k = M^k e_0.$$



Obsah

104. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Věta 3.16. *Lineární metoda je konvergentní, pokud*

$$\rho(M) < 1 ,$$

kde $\rho(M)$ je spektrální poloměr matice M , tj.

$$\rho(M) \stackrel{\text{def}}{=} \max\{|\lambda_1|, \dots, |\lambda_n|\}$$

a λ_i jsou vlastní čísla matice M .

Poznámka 3.17. Číslo $-\log(\rho(M))$ nazýváme *řád konvergence*.

Určení spektrálního poloměru je výpočetně náročný proces, který je složitější než samo řešení soustavy lineárních rovnic. Abychom tedy vůbec rozhodli o konvergenci lineární iterační metody, museli bychom řešit ještě náročnější problém, což by nebylo příliš efektivní. Z tohoto důvodu můžeme k rozhodnutí o konvergenci lineární iterační metody využít následující postačující podmínku.

Lemma 3.18. *Nechť $\|\cdot\|_M$ je maticová norma a*

$$\|M\|_M < 1 .$$

Pak konzistentní lineární iterační metoda je konvergentní.

Obsah

105. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Uvedme si několik základních lineárních iteračních metod. Nejprve matici A rozložme

$$A = L + D + U, \quad (3.2)$$

kde

- D je diagonální matice,
- L je ostře dolní trojúhelníková matice,
- U je ostře horní trojúhelníková matice.

3.2.1. Jacobiova metoda

$$M = -D^{-1}(L + U), \quad N = D^{-1} \text{ pokud } D_i \neq 0.$$

Předpis iterační metody má tvar

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^k \right).$$

3.2.2. Gaussova-Seidelova metoda

$$M = -(D + L)^{-1}U, \quad N = (D + L)^{-1}.$$

Předpis iterační metody má tvar

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^n a_{ij} x_j^k \right).$$



Obsah

106. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3.2.3. Successive over-relaxation method (SOR)

$$M = -(D + \omega L)^{-1}(\omega U + (1 - \omega)D), \quad N = \omega(D + \omega L)^{-1}.$$

Předpis iterační metody má tvar

$$x_i^{k+1} = (1 - \omega)x_i^k + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^n a_{ij}x_j^k \right),$$

kde $0 < \omega < 2$ je reálná konstanta.



Obsah

107. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Poznámka 3.19. (Odvození lineárních iteračních metod)

Uvažujme soustavu

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

kde a_{ij}, b_i jsou koeficienty soustavy a x_i jsou neznámé.

Soustavu lze upravit na tvar

$$\begin{aligned} a_{11}x_1 &= b_1 - a_{12}x_2 - a_{13}x_3 \\ a_{22}x_2 &= b_2 - a_{21}x_1 - a_{23}x_3 \\ a_{33}x_3 &= b_3 - a_{31}x_1 - a_{32}x_2 \end{aligned}$$

a předpokládáme-li nenulové prvky na diagonále, lze například pro x_1 psát (vydělením prvního řádku upravené soustavy koeficientem a_{11})

$$x_1 = \frac{1}{a_{11}} (b_1 - a_{12}x_2 - a_{13}x_3) .$$

Uvažujeme-li navíc iterační metodu, pak pro první složku následující iterace spočtené ze složek předchozí iterace bude platit

$$[x_{k+1}]_1 = \frac{1}{a_{11}} (b_1 - a_{12}[x_k]_2 - a_{13}[x_k]_3) . \quad (3.3)$$

Pokud budeme uvažovat postupně všechny složky x , získáme předpis Jacobiho metody. Z něj již snadno určíme matice M a N .



Obsah

108. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Jelikož v předpisu (3.3) lze při výpočtu $[x_{k+1}]_i$ místo x_k použít již vypočtené složky x_{k+1} . Získáme tak předpis pro Gaussovu-Seidelovu metodu.

Princip SOR metody pak spočívá v tom, že novou aproximaci x_{k+1} určíme jako kombinaci předchozí aproximace x_k a aproximace určenou Gaussovou-Seidelovou metodou. Váhu této kombinace pak určuje parametr ω . Pro $\omega = 1$ se jedná o Gaussovu-Seidelovu metodu.



Obsah

109. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Algoritmus 5

LINEÁRNÍ ITERAČNÍ METODA

Input: x_0, M, N, b , přesnost $\epsilon > 0$

```
1:  $k := 0$ 
2: while  $\|g_k\| \geq \epsilon\|b\|$  do
3:    $x_{k+1} := Mx_k - Nb$ 
4:    $k := k + 1$ 
5: end while
```

Output: x_k

Poznámka 3.20. Jako ukončovací kritérium byla v metodě použita podmínka na relativní chybu rezidua soustavy $g_k := Ax_k - b$, tj. $\|g_k\|/\|b\| < \epsilon$.

Abychom se v praxi vyhnuli podílu dvou malých čísel, které je co do šíření chyby velmi kritické, používáme raději podmínku

$$\|g_k\| < \epsilon\|b\| .$$



Obsah

110. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Příklad 3.21. Řešte soustavu

$$\begin{array}{rrcr} 5x_1 & +2x_2 & & = 3 \\ -x_1 & +4x_2 & +x_3 & = 0 \\ 2x_1 & -x_2 & +6x_3 & = 1 \end{array}$$

pomocí Jacobiho metody, Gaussovy-Seidelovy metody a SOR s vhodnou volbou parametru $\omega \in (0, 2)$.

Řešení. Nejprve matici soustavy rozložíme dle jednoduchého rozkladu (3.2)

$$A = \begin{bmatrix} 5 & 2 & 0 \\ -1 & 4 & 1 \\ 2 & -1 & 6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 2 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 5 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} =: L + D + U .$$

Dle předpisů jednotlivých metod získáme matice M, N a pomocí algoritmu obecné lineární metody spočteme řešení.

Při požadované přesnosti $\epsilon = 10^{-6}$ je Jacobiho metoda ukončena po 17-ti iteracích, Gaussova-Seidelova metoda po 9-ti iteracích. Na počet iterací algoritmu SOR vplývá volba parametru ω , pro jednotlivé volby je počet iterací znázorněn na Obrázku 3.4 (přičemž pro lepší názornost byl maximální počet iterací omezen na 1000).

Při volbě optimálního $\omega \approx 0.9$ je počet iterací algoritmu SOR 8.



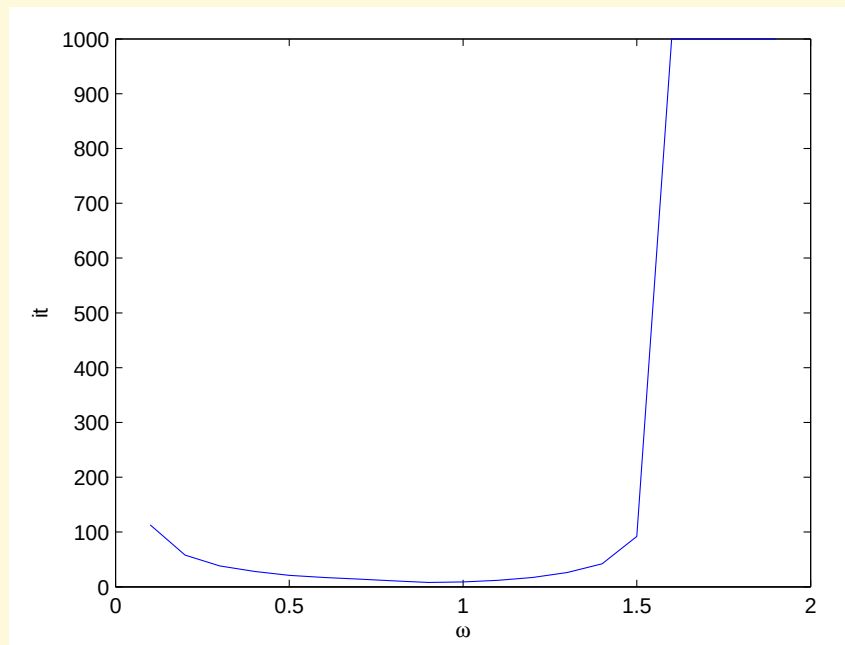
Obsah

111. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 3.4: Vliv volby ω na počet iterací SOR - Příklad 3.21



Obsah

112. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Příklad 3.22. Otestujte rychlost výše uvedených algoritmů na testovací úloze

$$\begin{aligned} A &:= \text{fivediag}(-1, -1, 4, -1, -1) \in \mathbb{R}^{n,n} \\ b &:= [1, \dots, 1]^T \in \mathbb{R}^n \\ x_0 &:= [0, \dots, 0]^T \in \mathbb{R}^n \\ \epsilon &:= 10^{-6} \end{aligned}$$

přičemž volte různou dimenzi úlohy $n = 4, 5, \dots, 70$.

Řešení. Nejdříve nalezneme vhodný parametr ω pro algoritmus SOR. Volme menší dimenzi úlohy, například $n = 10$. Vliv volby parametru ω na počet iterací SOR je vykreslen na Obrázku 3.5 (přičemž pro lepší názornost byl maximální počet iterací omezen na 5000).

Ukazuje se, že pro algoritmus SOR je vhodné použít optimální hodnotu $\omega \approx 1.44$.

Počet iterací pro jednotlivé metody je pak vykreslen na Obrázku 3.6.



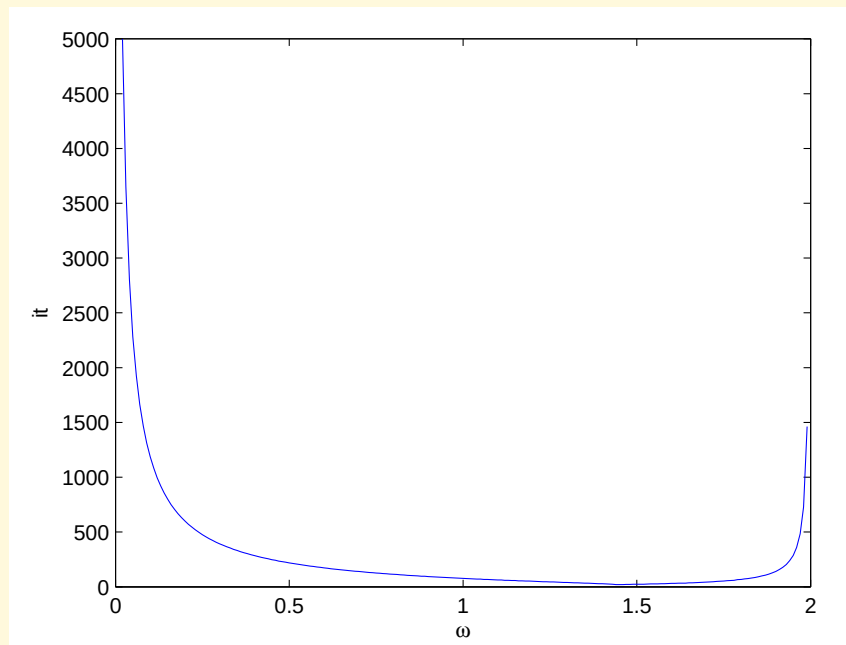
Obsah

113. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 3.5: Vliv volby ω na počet iterací SOR - Příklad 3.22



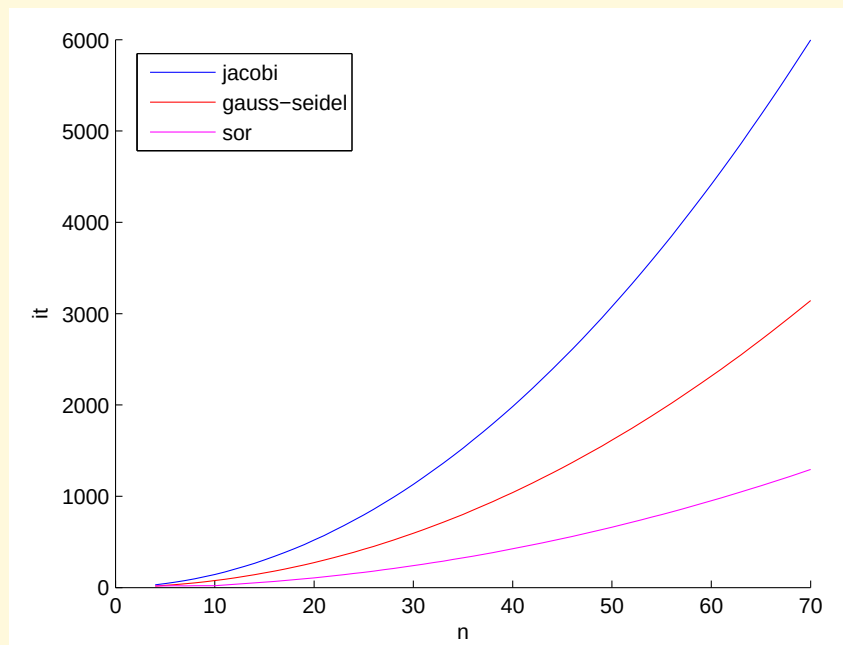
Obsah

114. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Obr. 3.6: Počet iterací algoritmů lineárních iteračních metod v závislosti na dimenzi úlohy
- Příklad 3.22



Obsah

115. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3.3. Gradientní metody

Uvažujme soustavu $Ax = b$, kde matice A je symetrická pozitivně definitní, tj.

$$A \in \mathbb{R}^{n,n} : \quad A = A^T \quad \wedge \quad \forall v \in \mathbb{R}^n, v \neq 0 : v^T A v > 0 .$$

Jiným přístupem k iteračnímu řešení soustav lineárních rovnic je řešení minimalizační úlohy, jejíž výsledek je totožný s řešením soustavy lineárních rovnic. Za tímto účelem tedy uvažujme úlohu

$$\min_{x \in \mathbb{R}^n} f(x) , \quad (3.4)$$

kde

$$f(x) := \frac{1}{2}(Ax, x) - (b, x), \quad f : \mathbb{R}^n \rightarrow \mathbb{R} . \quad (3.5)$$

V předpisu tzv. *kvadratické funkce* (3.5) symbol $(.,.)$ představuje Eukleidovský skalární součin definovaný předpisem

$$\forall u, v \in \mathbb{R}^n : \quad (u, v) := u^T v = \sum_{i=1}^n u_i v_i . \quad (3.6)$$



Obsah

116. strana ze 415



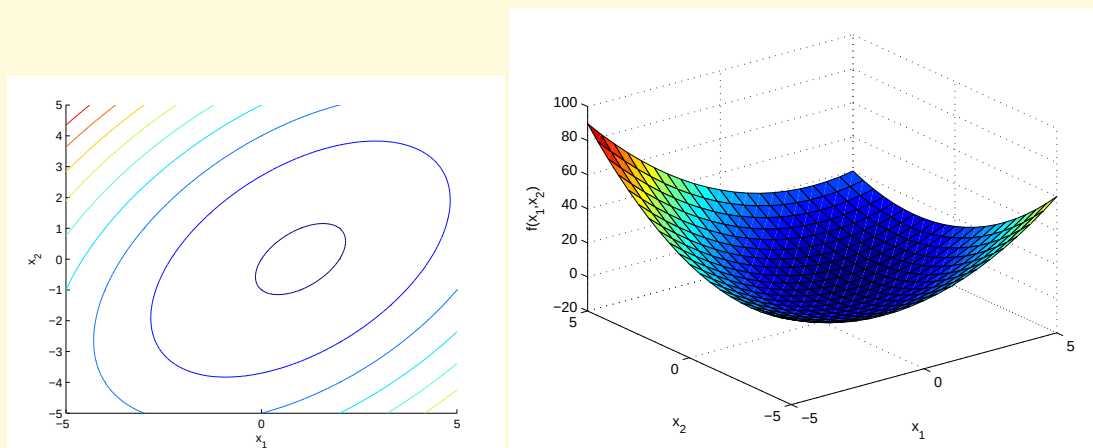
Zavřít dokument

Celá obrazovka / Okno

Příklad 3.23. Nechť je dána kvadratická funkce předpisem (3.5) s hodnotami

$$n := 2, \quad A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$$

Grafickou reprezentací této kvadratické funkce je paraboloid znázorněný na Obrázku 3.7.



Obr. 3.7: Grafická reprezentace dvoudimenzionální kvadratické funkce.

Spojitost mezi úlohou minimalizace ryze konvexní kvadratické funkce a řešením soustavy se symetrickou pozitivně definitní maticí popisuje následující lemma.

Lemma 3.24. *Nechť matice A je symetrická pozitivně definitní. Řešení soustavy $Ax = b$ je ekvivalentní s minimalizační úlohou (3.4).*



Obsah

118. strana ze 415



Zavřít dokument

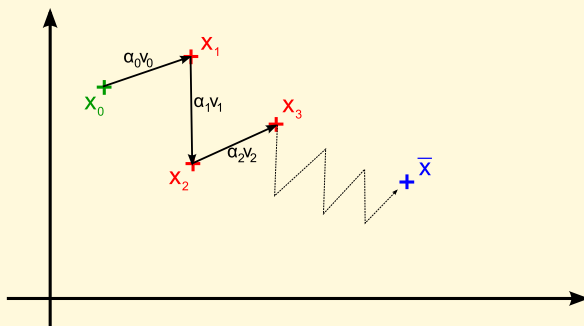
Celá obrazovka/Okno

3.3.1. Metoda největšího spádu

Metoda největšího spádu patří do skupiny tzv. *line-search* minimalizačních metod. Následující iteraci získáme tak, že funkci v každém iteračním kroku minimalizujeme ve směru vektoru v s délkou iteračního kroku α .

Pro nalezení nové iterace používáme předpis

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k v_k . \\ x_k &\in \mathbb{R}^n, v_k \in \mathbb{R}^n, \alpha_k \in \mathbb{R} . \end{aligned} \quad (3.7)$$



Obr. 3.8: Princip line-search metody - předpis (3.10)

- Při návrhu vhodného směru minimalizace funkce se jako logické jeví vydat se ve směru největšího spádu, tj. ve opačném směru než je gradient.
 v_k volíme tak, aby $\frac{df(x_k)}{dv_k}$ (derivace funkce f ve směru v_k , viz [13]) byla co nejmenší

$$\frac{df(x_k)}{dv_k} = (\nabla f(x_k), v_k) = \cos \varphi \cdot \|g_k\| \cdot \|v_k\| .$$

Tato hodnota bude nejmenší při volbě $\varphi = \pi \Leftrightarrow \cos \varphi = -1$, tedy úhel mezi vektory g_k a v_k je roven π , tedy $v_k = -g_k$ (opačný gradient je skutečně vektorem největšího spádu, tj. směru ve kterém funkce nejvíc klesá).



Obsah

120. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

- Předpokládejme tedy, že směr minimalizace v_k je vhodně zvolen. Otázka zůstává, jak daleko je možné se v tomto směru pohybovat, než-li hodnota funkce začne opět stoupat. Za tím účelem musíme zvolit optimální délku kroku α_k .
Za tímto účelem si tedy zvolme funkci $\varphi(\alpha) := f(x_k + \alpha v_k)$ jedné reálné proměnné, kterou je délka kroku α . Vhodnými úpravami dostáváme

$$\begin{aligned}
 \varphi(\alpha) &= \frac{1}{2}(A(x_k + \alpha v_k), x_k + \alpha v_k) - (b, x_k + \alpha v_k) \\
 &= \frac{1}{2}(Ax_k + \alpha Av_k, x_k + \alpha v_k) - (b, x_k) - (b, \alpha v_k) \\
 &= \frac{1}{2}(Ax_k, x_k) + \frac{1}{2}(Ax_k, \alpha v_k) + \frac{1}{2}(\alpha Av_k, x_k) \\
 &\quad + \frac{1}{2}(\alpha Av_k, \alpha v_k) - (b, x_k) - (b, \alpha v_k) \\
 &= \frac{1}{2}(Ax_k, x_k) - b, x_k + \frac{1}{2}(A\alpha v_k, \alpha v_k) + (\alpha Ax_k, v_k) - (b, \alpha v_k) \\
 &= f(x_k) + \frac{1}{2}(A\alpha v_k, \alpha v_k) + \alpha((Ax_k, v_k) - (b, v_k)) \\
 &= f(x_k) + \frac{1}{2}(A\alpha v_k, \alpha v_k) + \alpha(Ax_k - b, v_k) \\
 &= f(x_k) + \frac{1}{2}\alpha^2(Av_k, v_k) + \alpha(g_k, v_k) .
 \end{aligned} \tag{3.8}$$

Hledáme co nejvhodnější α , tj. takové, při kterém $\varphi(\alpha)$ je minimální. Za tímto účelem musíme minimalizovat funkci $\varphi(\alpha)$ jako funkci o jediné proměnné α

$$\varphi'(\alpha) = \alpha(Av_k, v_k) + (g_k, v_k) = 0 .$$

V našem případě je stacionárním bodem

$$\alpha = -\frac{(g_k, v_k)}{(Av_k, v_k)} . \tag{3.9}$$

V průběhu funkce $\varphi(\alpha)$ nedochází k žádným inflexím, funkce je konvexní na celém $\mathbb{R}$, protože

$$\varphi''(\alpha) = (Av_k, v_k) > 0 ,$$

což plyne z pozitivní definitnosti matice A , $\alpha_k = -\frac{(g_k, v_k)}{(Av_k, v_k)}$ je tedy minimum funkce φ .



Obsah

121. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Po dosazení do předpisu (3.10) dostáváme

$$x_{k+1} = x_k + \alpha_k v_k = x_k - \frac{(g_k, v_k)}{(Av_k, v_k)} g_k = x_k - \frac{(g_k, g_k)}{(Ag_k, g_k)} g_k . \quad (3.10)$$



Obsah

122. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Algoritmus 6

METODA NEJVĚTŠÍHO SPÁDU

Input: x_0, A, b , přesnost $\epsilon > 0$

```

1:  $g_0 := Ax_0 - b$ 
2:  $k := 0$ 
3: while  $\|g_k\| \geq \epsilon\|b\|$  do
4:    $\alpha_k := -\|g_k\|^2 / (Ag_k, g_k)$ 
5:    $x_{k+1} := x_k + \alpha_k g_k$ 
6:    $g_{k+1} := Ax_{k+1} - b$ 
7:    $k := k + 1$ 
8: end while
    
```

Output: x_k

Poznámka 3.25.

$$g_{k+1} = Ax_{k+1} - b = A(x_k + \alpha_k g_k) - b = Ax_k + A\alpha_k g_k - b = g_k + \alpha_k Ag_k$$

Tento způsob rezidua je výpočetně výhodnější, neboť jeho využitím nemusíme v algoritmu násobit Ax_k ale k výpočtu gradientu můžeme využít Ag_k (které jsme již spočetli při výpočtu α_k).

Obsah

123. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Lemma 3.26. *Metoda největšího spádu konverguje pro libovolnou volbu $x_0 \in \mathbb{R}^n$ k řešení rovnice $Ax = b$. Navíc pro chyby dvou po sobě jdoucích iterací metody největšího spádu platí*

$$(Ae_{k+1}, e_{k+1}) \leq \left(1 - \frac{1}{\kappa(A)}\right) (Ae_k, e_k) . \quad (3.11)$$

Poznámka 3.27. Jelikož matice A je symetrická a pozitivně definitní, lze skalární součin v odhadu chyby metody největšího spádu (3.11) značit jednoduše

$$(Ae_k, e_k) =: (e_k, e_k)_A = \|e_k\|_A^2 .$$

Příslušnou normu *indukovanou* tímto skalárním součinem označujeme pojmem *energetická*.

Pro zájemce

V algoritmu metody největšího spádu lze volit

$$\alpha_k = \frac{1}{\|A\|} .$$

Tato volba má obecně pomalejší konvergenci, nicméně konstantní volba koeficientů má svůj smysl - není nutno každý zvlášť počítat. Metoda se nazývá *Richardsonova metoda* a souvisí s odhadem posloupnosti koeficientů pomocí vlastních čísel, jak je vidět v podkapitole 4.2.1. Tyto koeficienty jsou totiž převrácené hodnoty Rayleighových podílů.



Obsah

124. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 3.28. Nalezněte průsečík přímek

$$\begin{aligned} p_1 : y &= 2x - 1 \\ p_2 : y &= \frac{1}{3}x - \frac{1}{3} \end{aligned}$$

pomocí metody největšího spádu.

Řešení. Jistě není problém tuto úlohu vyřešit analyticky, nicméně na tomto příkladě bychom rádi ukázali vlastnosti největšího spádu. Jelikož se jedná pouze o dvě dimenze, lze postup iterací vykreslit.

Soustavu lze maticově zapsat ve tvaru $Ax = b$, kde

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Pokud volíme například počáteční aproximaci $x_0 = (0, 0)^T$ a požadovanou přesnost $\epsilon = 10^{-4}$, algoritmus je ukončen po sedmi iteracích. Na Obrázku 3.9 je vykreslen postup iterací algoritmu vzhledem k vrstevnicím odpovídající kvadratické funkce,

na Obrázku 3.3.1 pak pokles normy chyby v jednotlivých iteracích a pokles normy residua.



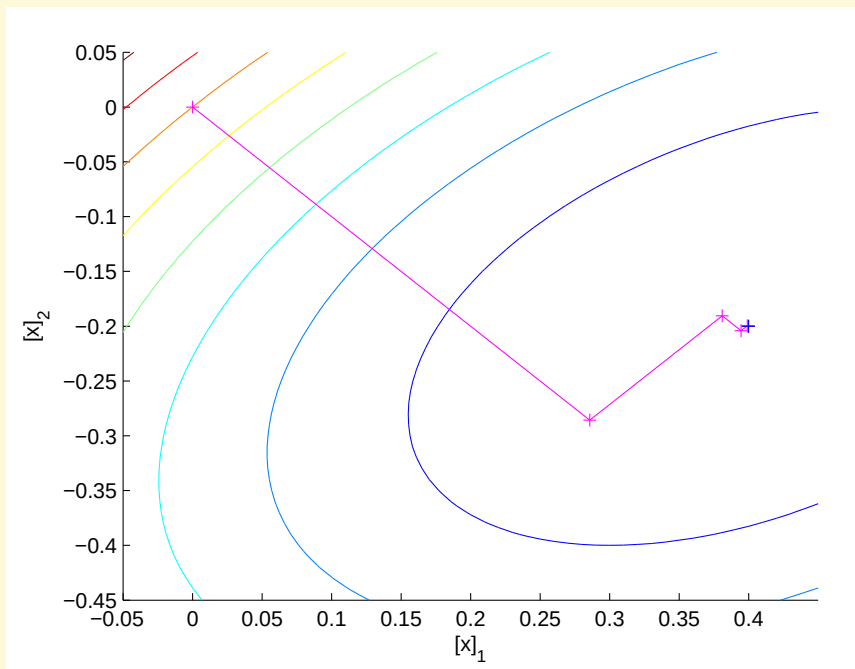
Obsah

125. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 3.9: Postup iterací metody největšího spádu - Příklad 3.28

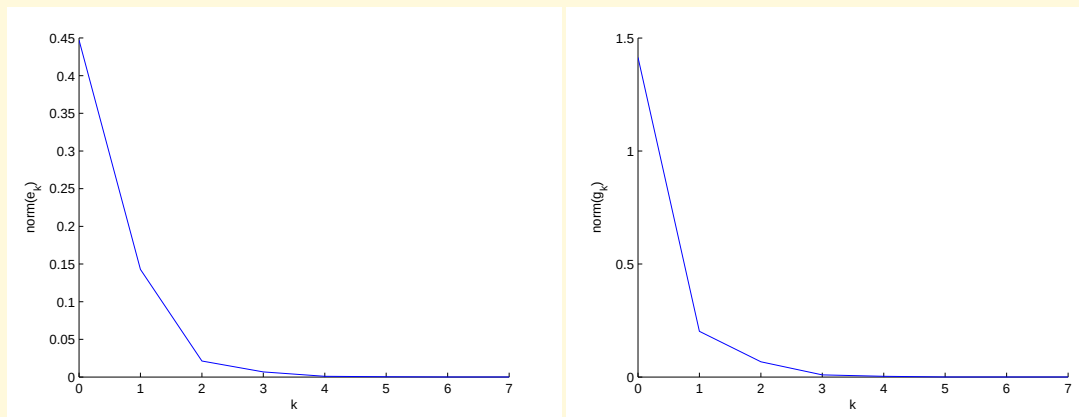
Obsah

126. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Obr. 3.10: Pokles normy chyby a normy residua při použití algoritmu největšího spádu - Příklad 3.28

Obsah

127. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3.3.2. Metoda sdružených gradientů

Ačkoliv se může zdát, že nejrychlejší minimalizaci lze dosáhnout použitím vektorů největšího spádu, není tomu tak. Existuje i mnohem efektivnější volba využívající tzv. *sdružených směrů*. Právě těchto směrů využívá metoda *sdružených gradientů*. Nejprve si tedy zavedeme pojem *sdružené směry* a poté si ukážeme jak je můžeme využít pro vhodnou volbu minimalizačního směru.

3.3.2.1. Volba sdružených směrů

Definice 3.29. O neprázdné množině nenulových vektorů $\{p_0, p_1, \dots, p_m\}$ řekneme, že je množinou *sdružených* (*A-ortogonálních*) *vektorů*, pokud

$$\forall i, j = 0, 1, \dots, m : i \neq j \Rightarrow p_i^T A p_j = 0 ,$$

kde matice A je symetrická pozitivně definitní.

Poznámka 3.30. Množinu sdružených vektorů lze ekvivalentně definovat pomocí skalárního součinu

$$\forall i, j = 0, 1, \dots, m : i \neq j \Rightarrow (A p_i, p_j) = 0 .$$



Obsah

128. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Lemma 3.31. *Prvky množiny sdružených vektorů jsou lineárně nezávislé.*

Předpokládejme, že v k -tém kroce máme k dispozici množinu sdružených vektorů $\mathcal{K} := \{p_0, p_1, \dots, p_k\}$ a následující iteraci získáme předpisem

$$x_{k+1} = x_k + \alpha_k p_k . \quad (3.12)$$

Koeficient α_k volíme tak, abychom minimalizovali funkci f v daném směru co nejvíce. Tedy podobně jako v případě metody největšího spádu zkonstruujeme funkci popisující funkci f v závislosti na volbě parametru α a následně ji minimalizujeme (viz (3.8)). Obdobně jako v případě metody největšího spádu získáme koeficient

$$\alpha_k = -\frac{(p_k, g_k)}{(Ap_k, p_k)} . \quad (3.13)$$

Otázkou však zůstává, proč volit směr p_k a nikoliv směr největšího poklesu $-g_k$ jako u metody největšího spádu (viz podkapitola 3.3.1).

Lemma 3.32. *Pro libovolnou počáteční aproximaci $x_0 \in \mathbb{R}^n$ posloupnost $\{x_k\}$ generována předpisem (3.12) dosáhne řešení $\bar{x} \in \mathbb{R}^n$ soustavy lineárních rovnic (3.1) po maximálně n krocích.*

Poznámka 3.33. Všechna tvrzení, která jsme dosud použili pro důkaz konvergence metody sdružených algoritmů jsou založeny na přesné reprezentaci reálných čísel. Při počítačové realizaci je však nutno počítat s nepřesnou aritmetikou. Pak navržená metoda algoritmus nebude ukončena po n krocích a výpočetní postup bude muset být zastaven vhodnou ukončovací podmínkou.



Obsah

129. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3.3.2.2. Gramův-Schmidtův A -ortogonalizační proces

Smysl využití sdružených vektorů pro generování nových aproximací je nám znám. Otázkou zůstává, jak tyto vektory generovat.

Překvapivě se dá k této úloze využít postup dobře známý z Lineární algebry, tzv. *Gramův-Schmidtův ortogonalizační proces* (viz [6]).

Jako první A -ortogonální vektor p_0 budeme volit gradient v bodě x_0 . O tomto gradientu víme, že je nenulový (pokud by byl nulový, pak $g_0 = Ax_0 - b = 0$ a to znamená, že x_0 je řešení a algoritmus je ukončen).

Uvažujme k -tou iteraci. Necht máme k dispozici množinu A -ortogonálních vektorů $\{p_0, p_1, \dots, p_k\}$ a úkolem je nalézt p_{k+1} splňující podmínku sdruženosti (A -ortogonalita) ke stávajícím sdruženým vektorům, tj.

$$\forall i = 0, 1, \dots, k : p_{k+1}^T A p_i = 0 \quad (\text{ekvivalentně } (A p_{k+1}, p_i) = 0) .$$

Pak lze vektor p_{k+1} vyjádřit jako lineární kombinaci

$$p_{k+1} = g_{k+1} - \beta_{k+1,0} p_0 - \beta_{k+1,1} p_1 - \dots - \beta_{k+1,k} p_k = g_{k+1} - \left(\sum_{j=0}^k \beta_{k+1,j} p_j \right) . \quad (3.14)$$

Dosazením do podmínky A -ortogonalita

$$0 = (A p_{k+1}, p_i) = \left(A \left(g_{k+1} - \sum_{j=0}^k \beta_{k+1,j} p_j \right), p_i \right) = (A g_{k+1}, p_i) - \sum_{j=0}^k \beta_{k+1,j} (A p_j, p_i) .$$

Využitím A -ortogonalita stávajících vektorů $p_i, i = 0, \dots, k$, se rovnice zjednoduší na tvar

$$0 = (A g_{k+1}, p_i) - \beta_{k+1,i} (A p_i, p_i) .$$



Obsah

130. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Odkud lze snadno získat předpis pro i -tý koeficient lineární kombinace

$$\beta_{k+1,i} = \frac{(Ag_{k+1}, p_i)}{(Ap_i, p_i)} = \frac{(Ap_i, g_{k+1})}{(Ap_i, p_i)} . \quad (3.15)$$

Poslední úpravu jsme mohli provést díky symetrii skalárního součinu (tj. $\forall u, v \in \mathbb{R}^n : (u, v) = (v, u)$). Toto vyjádření je snažší z důvodu pozdější implementace, protože budeme násobit maticí A pouze jednou.

Zpětným dosazením do lineární kombinace (3.14) získáme předpis pro výpočet p_{k+1}

$$p_{k+1} = g_{k+1} - \sum_{j=0}^k \frac{(Ap_j, g_{k+1})}{(Ap_j, p_j)} p_j . \quad (3.16)$$

Tento předpis je však výpočetně i paměťově náročný, neboť je nutno uchovávat v paměti všechny předchozí směry a pro výpočet směru nového je nutno v sumaci procházet tyto směry. Díky vlastnostem sdruženosti, je však možné situaci značně zjednodušit. Nejprve si však uvedme následující pomocná tvrzení.



Obsah

131. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Lemma 3.34. (o ortogonalitě gradientů metody sdružených gradientů)

Nechť $\{g_0, \dots, g_{n-1}\}$ je množina gradientů v příslušných bodech množiny aproximací $\{x_0, \dots, x_{n-1}\}$, které jsou zkonstruovány předpisem (3.12).

Tato množina je ortogonální.

Lemma 3.35. Nechť je dána množina gradientů $\{g_0, g_1, \dots, g_{k+1}\}$ a množina sdružených vektorů $\{p_0, p_1, \dots, p_k\}$ zkonstruovaná z množiny gradientů A -ortogonalizačním procesem (3.16).

Pak platí

$$(Ap_j, g_{k+1}) = \begin{cases} = 0 & \text{pro } j < k, \\ \neq 0 & \text{pro } j = k. \end{cases}$$

Poznámka 3.36. Přímým důsledkem předchozí věty 3.35 je

$$\beta_{k+1,i} = \frac{(Ap_i, g_{k+1})}{(Ap_i, p_i)} = \begin{cases} = 0 & \text{pro } i < k, \\ \neq 0 & \text{pro } i = k. \end{cases}$$

Z toho vyplývá, že ze všech koeficientů $\beta_{k+1,i}$ je nenulový jediný koeficient

$$\beta_{k+1} := \beta_{k+1,k} = \frac{(Ap_k, g_{k+1})}{(Ap_k, p_k)}. \quad (3.17)$$

Využitím předchozí věty lze předpis (3.16) zjednodušit na tvar

$$p_{k+1} = g_{k+1} - \beta_{k+1} p_k = g_{k+1} - \frac{(Ap_k, g_{k+1})}{(Ap_k, p_k)} p_k, \quad (3.18)$$

což znamená, že při výpočtu nového -sdruženého směru vystačíme pouze s jediným sdruženým směrem z předchozí iterace.



Obsah

132. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3.3.2.3. Algoritmus CG a jeho vlastnosti

Zhrneme-li všechna předchozí pozorování, můžeme sestavit algoritmus metody sdružených gradientů.

Algoritmus 7

METODA SDRUŽENÝCH GRADIENTŮ

Input: x_0, A, b , přesnost $\epsilon > 0$

```

1:  $g_0 := Ax_0 - b$ 
2:  $p_0 := g_0$ 
3:  $k := 0$ 
4: while  $\|g_k\| \geq \epsilon\|b\|$  do
5:    $\alpha_k := -(p_k, g_k)/(Ap_k, p_k)$ 
6:    $x_{k+1} := x_k + \alpha_k p_k$ 
7:    $g_{k+1} := Ax_{k+1} - b$ 
8:    $\beta_{k+1} := (Ap_k, g_{k+1})/(Ap_k, p_k)$ 
9:    $p_{k+1} := g_{k+1} - \beta_{k+1} p_k$ 
10:   $k := k + 1$ 
11: end while
    
```

Output: x_k

Lemma 3.37. *V algoritmu metody sdružených gradientů můžeme nahradit*

$$\alpha_k = \frac{\|g_k\|^2}{p_k^T A p_k}, \quad \beta_{k+1} = -\frac{\|g_{k+1}\|^2}{\|g_k\|^2}.$$



Obsah

133. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 3.38. 1. Pro řešení $\bar{x}$ soustavy $Ax = b$ platí

$$\begin{aligned}\bar{x} \in \mathcal{K}_n &= \langle p_0, \dots, p_{n-1} \rangle, \\ &= \langle g_0, \dots, g_{n-1} \rangle, \\ &= \langle g_0, Ag_0, \dots, A^{n-1}g_0 \rangle,\end{aligned}$$

kde $\langle v_0, \dots, v_{n-1} \rangle$ značí lineární obal množiny vektorů $\{v_0, \dots, v_{n-1}\}$, viz [2].

Prostory $\mathcal{K}_n$ nazýváme *Krylovovy prostory*.

2. Metoda sdružených gradientů musí v přesné aritmetice nalézt řešení po n krocích. Může však nastat situace, že řešení

$$\bar{x} \in \langle g_0, Ag_0, \dots, A^{m-1}g_0 \rangle.$$

Pak bude $g_{m-1} = 0$ a řešení je nalezeno po m krocích.

3. Na metodu sdružených gradientů můžeme taktéž pohlížet jako na iterační metodu. Hledání řešení ukončíme po splnění určité přesnosti (viz algoritmus).



Obsah

134. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Lemma 3.39. *Pro chyby dvou po sobě jdoucích iterací metody sdružených gradientů platí*

$$(Ae_{k+1}, e_{k+1}) \leq \left(\frac{\kappa(A) - 1}{\kappa(A) + 1} \right)^2 (Ae_k, e_k) . \quad (3.19)$$

Poznámka 3.40. Často se nerovnost (3.19) udává s energetickou normou, viz Poznámka 3.27.



Obsah

135. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3.3.3. Předpodmínění

Z předchozích kapitol jasně vyplývá, že konvergence gradientních metod závisí na čísle podmíněnosti matice soustavy. Čím menší je číslo podmíněnosti, tím rychleji metoda konverguje k předem zadané chybě. Pokud chceme zajistit rychlou konvergenci gradientních metod, musíme se pokusit nahradit původní soustavu lineárních rovnic $Ax = b$ jinou ekvivalentní soustavou, jejíž číslo podmíněnosti bude nižší.

Proces nahrazení soustavy $Ax = b$ ekvivalentní soustavou $\tilde{A}x = \tilde{b}$, kde číslo podmíněnosti nové matice soustavy $\tilde{A}$ je menší než číslo podmíněnosti matice A , nazýváme *předpodmínění*.



Obsah

136. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Zvolíme-li regulární matici C , která je „blízká“ matici A v tom smyslu, že $C^{-1}A$ má číslo podmíněnosti ideálně blízké 1, pak tzv. *předpodmíněnou soustavu* získáme přenásobením obou stran původní soustavy maticí C .

$$\begin{aligned} Ax &= b \\ C^{-1}Ax &= C^{-1}b \\ \tilde{A}x &= \tilde{b}, \quad \tilde{A} \stackrel{\text{def}}{=} C^{-1}A, \tilde{b} \stackrel{\text{def}}{=} C^{-1}b. \end{aligned}$$

Výsledná matice předpodmíněné soustavy $\tilde{A} := C^{-1}A$ však nemusí být symetrická a ani pozitivně definitní. V takovém případě by něšlo na předpodmíněnou soustavu použít gradientních metod.

Abychom se této situaci vyhlí, volíme za matici předpodmínění matici ve tvaru $C := C^{1/2}C^{T/2}$, která je symetrická a pozitivně definitní.

Pak původní soustavu $Ax = b$ přenásobíme zleva $C^{-1/2}$ a provedeme transformaci vektoru řešení $\tilde{x} = C^{T/2}x$.

Tímto dostáváme novou předpodmíněnou soustavu

$$\tilde{A}\tilde{x} = \tilde{b},$$

kde

$$\tilde{A} := C^{-1/2}AC^{-T/2}, \quad \tilde{x} := C^{T/2}x, \quad \tilde{b} := C^{-1/2}b.$$



Obsah

137. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pro metodu sdružených gradientů pak platí

$$\begin{aligned}\tilde{g}_k &= C^{-1/2}g_k \\ s_k &= C^{-1/2}\tilde{g}_k \\ v_k &= C^{-1/2}\tilde{v}_k \\ \tilde{\beta}_{k-1} &= -\frac{(As_k, v_{k-1})}{(Av_{k-1}, v_{k-1})} \\ v_k &= s_k + \beta_{k-1} \\ \tilde{\alpha}_k &= -\frac{(\tilde{g}_k, \tilde{v}_k)}{(A\tilde{v}_k, \tilde{v}_k)} = -\frac{(C^{-1/2}g_k, C^{\frac{1}{2}}v_k)}{(C^{-\frac{1}{2}}AC^{-\frac{1}{2}}C^{\frac{1}{2}}v_k, C^{\frac{1}{2}}v_k)} = -\frac{(g_k, v_k)}{(Av_k, v_k)} \\ x_{k+1} &= x_k + \tilde{\alpha}_k v_k\end{aligned}$$

Je tedy vidět, že se algoritmus liší pouze ve výpočtu β_{k-1} a konstrukce v_k .

Je nutné vypočítat vektor s_k ze soustavy $Cs_k = r_k$. Proto je taktéž nutné, aby tato soustava byla snadno řešitelná.



Obsah

138. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Algoritmus 8

METODA SDRUŽENÝCH GRADIENTŮ S PŘEDPODMÍNĚNÍM

Input: x_0, A, b, C , přesnost $\epsilon > 0$

```

1:  $g_0 := Ax_0 - b$ 
2:  $s_0 := C^{-1}g_0$ 
3:  $p_0 := s_0$ 
4: while  $\|g_k\| \geq \epsilon\|b\|$  do
5:    $\alpha_k := -(s_k, g_k)/(Ap_k, p_k)$ 
6:    $x_{k+1} := x_k + \alpha_k p_k$ 
7:    $g_{k+1} := g + \alpha_k Ap_k$ 
8:    $s_{k+1} := C^{-1}g_{k+1}$ 
9:    $\beta_{k+1} := (s_{k+1}, g_{k+1})/(s_k, g_k)$ 
10:   $p_{k+1} := s_{k+1} + \beta_{k+1}p_k$ 
11:   $k := k + 1$ 
12: end while
    
```

Output: x_k



Obsah

139. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



3.3.3.1. Jacobiho diagonální předpodmínění

Jednou z nejjednodušších metod předpodmínění je použití diagonály matice A . Matice předpodmínění pak má tvar

$$\begin{aligned} C &= \text{diag}(A) \\ C^{1/2} &= C^{T/2} = \text{diag}(A)^{1/2} = \text{diag}(\sqrt{a_{ii}}). \end{aligned}$$

Tento typ předpodmínění není příliš robustní a největšího efektu dosáhne u soustav, kde v matici soustavy A výrazně dominují diagonální prvky, tj. jsou výrazně větší než prvky mimodiagonální.

3.3.3.2. Symmetric successive over-relaxation method (SSOR)

Významnějšího efektu předpodmínění u obecnějších typů matic je možné dosáhnout využitím tzv. SSOR metody.

Matice předpodmínění se konstruuje následujícím způsobem

$$\begin{aligned} A &= D + L + L^T \\ C_\omega &= \frac{1}{2-\omega} \left(\frac{1}{\omega} D + L \right) \left(\frac{1}{\omega} D \right)^{-1} \left(\frac{1}{\omega} D + L \right)^T, \quad 0 < \omega < 2 \\ C_\omega^{\frac{1}{2}} &= \sqrt{\frac{1}{2-\omega}} \left(\frac{1}{\omega} D + L \right) \left(\frac{1}{\omega} D \right)^{-1/2}. \end{aligned}$$

Velmi efektivní implementace této metody je využití tzv. *Eisenstatova triku*. Výpočetní náročnost metody předpodmíněných gradientů se pak prakticky nezvyšuje oproti použití standartní metody sdružených gradientů.

Obsah

140. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

3.3.3.3. Neúplná Choleského faktorizace

Nejrobustnějším typem předpodmínění, se kterým se seznámíme je tzv. *neúplná Choleského faktorizace*. Jak je patrné z jejího názvu, vychází z *Choleského rozkladu*, se kterým jste se seznámili v předmětu Lineární algebra [2].

Nejprve si tedy zopakujeme co se rozumí pod pojmem Choleského rozklad.

Věta 3.41. (*Choleského rozklad*)

Nechť $A \in \mathbb{R}^{n,n}$ je symetrická pozitivně definitní matice. Pak existuje jednoznačně daná dolní trojúhelníková matice $L \in \mathbb{R}^{n,n}$ s kladnými diagonálními prvky taková, že

$$A = LL^T. \quad (3.20)$$

Myšlenka neúplné Choleského faktorizace spočívá v nalezení matice $\tilde{L}$, která je *blízká* matici L . Pak matice předpodmínění pomocí neúplné Choleského faktorizace má tvar

$$C = \tilde{L}\tilde{L}^T, \quad C^{1/2} = \tilde{L}.$$

Očividně čím více se bude matice $\tilde{L}$ blížit matici L z plného Choleského rozkladu, tím blíže bude matice $\tilde{A}$ k jednotkové matici, která má číslo podmíněnosti 1.

Speciálně pokud bychom vzali $\tilde{L} = L$, dostáváme ideální předpodmínění. Je však jasné, že použitím metody předpodmíněných sdružených gradientů bychom pak nemohli překonat co do výpočetní náročnosti Choleského rozklad. Takový typ předpodmínění by proto nedával smysl.



Obsah

141. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Vraťme se tedy ke konstrukci matice $\tilde{L}$.

V principu existují 2 základní přístupy k sestavení této matice. V obou případech je konstrukce založena na sestavování matice L jako v případě standardního Choleského rozkladu (viz [2] a [1]), s tím, že některé prvky matice L nebudeme do výsledného faktoru zapisovat. Podívejme se tedy na tyto dva přístupy.

1. Neúplný rozklad podle vzorku

Pokud je některý prvek $[A]_{i,j}$ nulový, položíme odpovídající $[\tilde{L}]_{i,j}$ automaticky roven nule. Počet operací nutných k výpočtu je pak mnohem menší. Tento postup je obzvláště efektivní, pokud původní matice soustavy A je řídká, pak zřejmě i matice předpodmínění $C = \tilde{L}\tilde{L}^T$ bude řídká. Při tomto zjednodušení však není zaručeno, že matice C bude pozitivně definitní. Toto lze však napravit přenásobením diagonálních prvků matice A vhodnou konstantou $\omega \gg 1$.

2. Neúplný rozklad podle hodnoty

Vypočtený nový prvek $[\tilde{L}]_{i,j}$ zanedbáváme, tj. položíme roven nule, pokud je relativně menší než diagonální prvek matice A , tj. $|\tilde{L}_{i,j}| \leq \epsilon |A_{i,i}|$. Tento způsob zaručuje pozitivní definitnost výsledné matice předpodmínění, nicméně tato matice může mít více nenulových prvků než původní matice. Je zřejmé, že pro volbu $\epsilon = \infty$ dostáváme úplnou Choleského faktorizaci. Volbou většího ϵ můžeme dosáhnout lepšího efektu předpodmínění a tím i menšího počtu iterací metodou sdružených gradientů. Toto je však vyváženou větší výpočetní náročností sestavení matice předpodmínění a následného výpočtu předpodmíněného gradientu s_k v každém kroce metody PCG. Výsledná efektivita tak nemusí být zřejmá a liší se případ od případu.



Obsah

142. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Algoritmus 9

NEÚPLNÁ CHOLESKÉHO FAKTORIZACE

Input: A , přesnost $\epsilon > 0$

```

1:  $n :=$  velikost  $A$ 
2: for  $j = 1, \dots, n$  do
3:    $v = s_j^A - \sum_{k=1}^{j-1} [L]_{j,k} s_k^L$ 
4:    $s = \frac{1}{\sqrt{[v]_j}} v$ 
5:   for  $i = j, \dots, n$  do
6:     if  $|s_i| < \epsilon |A_{j,j}|$  then
7:        $\tilde{L}_{j,i} = 0$ 
8:     else
9:        $\tilde{L}_{j,i} = s_i$ 
10:    end if
11:  end for
12: end for
    
```

Output: $\tilde{L}$

Více o předpokladiovacích technikách lze nalézt například v [18], případně [22].



Obsah

143. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Příklad 3.42. Porovnejte rychlosti numerických řešičů soustav lineárních rovnic uvedených v této kapitole na testovací soustavě

$$Ax = b,$$

kde

$$A = \text{fivediag}(-1, -1, 4, -1, -1) \in \mathbb{R}^{n,n}, \quad b = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n$$

a volte různou dimenzi úlohy n .

Řešení. Pro požadovanou přesnost $\epsilon = 10^{-4}$ získáme počty iterací pro jednotlivé metody vykreslené na Obrázku 3.11. ▲



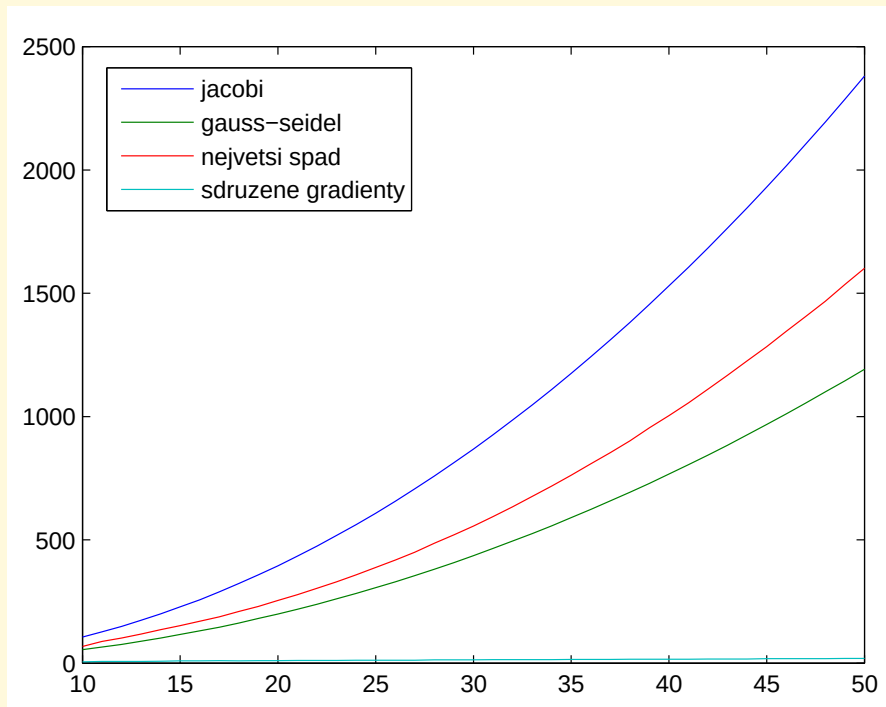
Obsah

144. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Obr. 3.11: Počet iterací algoritmů pro řešení soustavy lineárních rovnic pro různou dimenzi úlohy

3.4. Test

Následující test zahájíte kliknutím na tlačítko „Začátek testu“. U otázek, kde lze volit jen jednu odpověď (test nedovolí zaškrtnout více odpovědí), je správná odpověď bodována počtem bodů uvedených v závorce u zadání a špatná odpověď je bodována 0 body. U otázek, kde lze volit více správných odpovědí, je součet bodů správných odpovědí uveden v závorce u zadání a za každou špatnou odpověď bude odečten jeden bod.

1. (4b.) Které z následujících tvrzení platí pro normu?

$$\|x - y\| \leq \|x\| - \|y\|$$

$$\|x\| + \|y\| \geq \|x + y\|$$

$$\|x\| - \|y\| \geq 0$$

$$\|1/\alpha x\| = |1/\alpha| \cdot \|x\|$$

$$\|x\| = 0 \Leftrightarrow x = 0$$

$$\|\alpha x\| + \|\beta y\| = |\alpha\beta| \cdot \|x + y\|$$



Obsah

146. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2. (4b.) Vyberte z následujících ta zobrazení, které jsou normy (vektorové nebo maticové).

$$\sqrt{\sum_{i=1}^n x_i^2}$$

$$\sqrt{\sum_{i=1}^n x_i}$$

$$\max\left\{\sum_{k=1}^n |a_{jk}|, j = 1, \dots, n\right\}$$

$$\sum_{i=1}^n |x_i|$$

$$\sum_{i=1}^n x_i$$

$$\sqrt{\sum_{i,j} a_{ij}}$$

$$\sqrt{\sum_{i,j} |a_{ij}|^2}$$

$$\prod_{i=1}^n x_i$$

3. (3b.) Necht $\|\cdot\|_M$ značí maticovou a $\|\cdot\|_V$ odpovídající vektorovou normu. Která z následujících tvrzení platí?

$$\forall A \in \mathbb{R}^{n \times n} \forall x \in \mathbb{R}^n : \|A\|_M \cdot \|x\|_V \geq \|Ax\|_V$$

$$\forall A \in \mathbb{R}^{n \times n} \forall x \in \mathbb{R}^n : \|A\|_M \cdot \|x\|_V \leq \|Ax\|_V$$

$$\forall A, B \in \mathbb{R}^{n \times n} \forall x \in \mathbb{R}^n : \|AB\|_M \geq \|A\|_M \cdot \|B\|_M$$

$$\forall A, B \in \mathbb{R}^{n \times n} : \|ABx\|_M \leq \|A\|_V \cdot \|B\|_M \cdot \|x\|_V$$



Obsah

147. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

4. (4b.) Frobeniova norma

je rovna odmocnině z největšího vlastního čísla,

je rovna součtu kvadrátů vlastních čísel,

je rovna odmocnině součtu kvadrátů všech prvků matice,

je rovna největšímu prvku matice v absolutní hodnotě.

5. (4b.) Iterační metoda Φ se nazývá *konzistentní* se soustavou lineárních rovnic $Ax = b$, jestliže

pro každou aproximaci $x_k \in \mathbb{R}^n$ všechna řešení $\bar{x}$ jsou pevným bodem Φ ,

pro každou pravou stranu $b \in \mathbb{R}^n$ všechna řešení $\bar{x}$ jsou pevným bodem Φ ,

pro posloupnost aproximací platí $\lim_{k \rightarrow \infty} x_k = \bar{x}$,

pro každou pravou stranu $\Phi(x) = Mx + Nb$, kde $M, N \in \mathbb{R}^{n \times n}$.



Obsah

148. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

6. (4b.) Lineární iterační metoda $x_{k+1} = Mx_k + Nb, k \geq 0$ je konzistentní, právě když

$$N = M - A,$$

$$N = I - AM,$$

$$M = N - A,$$

$$M = I - NA.$$

7. (4b.) Lineární iterační metoda $x_{k+1} = Mx_k + Nb, k \geq 0$ je konvergentní, pokud

$$\|M\|_M < 1, \quad \rho(M) > 1,$$

$$\|M\|_M > 1, \quad \rho(M) < 1.$$



Obsah

149. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

8. (3b.) Lineární iterační metoda má předpis $x_{k+1} = Mx_k + Nb, k \geq 0$. Dále rozložme $A = L + D + U$, kde D je diagonální matice, L je ostře dolní trojúhelníková matice a U je ostře horní trojúhelníková matice. Která z následujících tvrzení jsou platná?

Jacobiova metoda je metoda s volbou $M = D(L + U), N = D$,

Jacobiova metoda je metoda s volbou $M = -D^{-1}(L + U), N = D^{-1}$,

Gaussova-Seidelova metoda je metoda s volbou $M = -(D + L)^{-1}U, N = (D + L)^{-1}$,

Gaussova-Seidelova metoda je metoda s volbou $M = -(D + L)U^{-1}, N = (D + L)^{-1}$,

Successive over-relaxation method (SOR) je metoda s volbou $M = -(D + \omega L)^{-1}(\omega U + (1 - \omega)D), N = \omega(D + \omega L)^{-1}$,

Successive over-relaxation method (SOR) je metoda s volbou $M = -(D + \omega L)(\omega U + (1 - \omega)D)^{-1}, N = \omega(D + \omega L)$,



Obsah

150. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

9. (4b.) Gradientní spádové metody jsou založeny na ekvivalenci řešení soustavy lineárních rovnic se symetrickou pozitivně definitní maticí a minimalizací

ryze konvexní kvadratické funkce,

lineární funkce,

ryze konkávní kvadratické funkce,

polynomiální interpolační funkce.

10. (4b.) Necht $\bar{x} \in \mathbb{R}^n$ je řešení soustavy lineárních rovnic $Ax = b$, kde $b \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, $A = A^T$, $\forall x \in \mathbb{R}^n \setminus \{0\} : x^T Ax > 0$. Pak platí

$\bar{x}$ je minimum funkce $1/2x^T Ax - b^T x + c$, kde $c \in \mathbb{R}$ je libovolná konstanta,

$\bar{x}$ je maximum funkce $1/2x^T Ax - b^T x$,

residuum v bodě $\bar{x}$ je rovno nule,

$\bar{x}$ je jediný stacionární bod funkce $1/2x^T Ax - b^T x$.



Obsah

151. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

11. (4b.) Metoda největšího spádu pro minimalizaci funkce f volí délku kroku tak, aby
- následující iterace byla minimem funkce f ve směru opačného gradientu v bodě aktuální iterace,
 - následující iterace byla minimem funkce f ve směru gradientu v bodě aktuální iterace,
 - následující iterace byla minimem funkce f ,
 - gradient v bodě následující iterace byl kolmý na gradient v bodě aktuální iterace.
12. (4b.) Počet iterací metody největšího spádu je závislý na
- volbě počáteční aproximace,
 - čísle podmíněnosti matice soustavy,
 - velikosti pravé strany,
 - na požadované přesnosti.



Obsah

152. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

13. (4b.) Metoda sdružených gradientů jako směr posunu k následující iteraci volí
 gradient v bodě aktuální iterace,
 minus gradient v bodě aktuální iterace,
 rozdíl aktuální a předchozí iterace,
 prvky množiny A -sdružených vektorů.

14. (4b.) Prvky Krylovova prostoru $\mathcal{K}_n$
 jsou ortogonální,
 jsou ortonormální,
 jsou lineárně nezávislé,
 se dají vyjádřit jako lineární kombinace vektorů $\{g_0, Ag_0, \dots, A^{m-1}g_0\}$.



Obsah

153. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

15. (4b.) Máme za úkol nalézt řešení soustavy $Bx = c$, kde $B \in \mathbb{R}^{n,n}$ je symetrická pozitivně definitní matice a $c \in \mathbb{R}^n$ je vektor pravých stran. Která z následujících metod bude (obecně, bez předpokládání) pro tento problém nejrychleji konvergovat?

metoda největšího spádu,

metoda sdružených gradientů,

Gaussova eliminační metoda,

Gauss-Seidelova metoda,

Jacobiho metoda.

SOR metoda s ideálním parametrem ω .



Obsah

154. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

16. (4b.) Máme za úkol nalézt řešení soustavy $Bx = c$, kde $B \in \mathbb{R}^{n,n}$ je matice se spektrálním poloměrem menším než jedna a $c \in \mathbb{R}^n$ je vektor pravých stran. Která z následujících metod bude (obecně, bez předpokládání) pro tento problém nejrychleji konvergovat?

metoda největšího spádu,

metoda sdružených gradientů,

Gaussova eliminační metoda,

Gauss-Seidelova metoda,

Jacobiho metoda.

SOR metoda s ideálním parametrem ω .



Obsah

155. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

17. (3b.) Nechť je dána soustava lineárních rovnic $Ax = b$, kde $A \in \mathbb{R}^{n,n}$ je reálná symetrická pozitivně definitní matice a $b \in \mathbb{R}^n$ je vektor pravých stran. Pokud neuvažujeme chyby způsobené aproximací čísel na počítači, je metoda sdružených gradientů při řešení této soustavy

ukončena po maximálně n krocích,

ukončena po maximálně $n - 1$ krocích,

je konvergentní,

má rychlejší konvergenci než metoda největšího spádu.

Správně zodpovězené otázky:

Získané body:

Procento úspěšnosti:

Pro ukončení testu je třeba kliknout na tlačítko „Konec testu“. Opravení testu se provede kliknutím na tlačítko „Výsledky“, správné odpovědi budou označeny zeleně a červeně budou označeny odpovědi chybné. Při použití otázky s tvořenou odpovědí se správná odpověď zobrazí v rámečku umístěném vpravo dole v navigačním panelu po kliknutí na tlačítko „Odpověď“.



Obsah

156. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Kapitola 4

Výpočet vlastních čísel a vektorů matic

Průvodce studiem

Další velmi významnou skupinou úloh lineární algebry je problém hledání vlastních čísel a vlastních vektorů. Jejich fyzikální interpretace spočívá v popisu vlnění či vibrací, nacházejí však uplatnění i více teoretické při dělení grafů apod. Největší a nejmenší vlastní číslo jsme potřebovali i v předchozí kapitole pro výpočet čísla podmíněnosti.

V této kapitole se proto seznámíme se základními metodami řešení těchto úloh. Nejprve se budeme zabývat tzv. částečným problémem vlastních čísel, kdy budeme hledat jen několik vlastních čísel a na závěr si ukážeme i řešení úplného problému, kdy výstupem numerických metod bude celé spektrum matice.

Obsah

157. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

4.1. Výpočet charakteristického polynomu

Nejprve si zopakujme pojmy vlastních čísel a vlastních vektorů.

Definice 4.1. Nechť $A \in \mathbb{R}^{n,n}$ je matice.

Jestliže existuje nenulový vektor $e \in \mathbb{C}^n$ a skalár $\lambda \in \mathbb{C}$ tak, že

$$Ae = \lambda e,$$

pak se λ nazývá *vlastní číslo* matice A , e se nazývá *vlastní vektor* příslušný k λ a (λ, e) se nazývá *vlastní dvojice* matice A .

Definice 4.2. Pod pojmem *charakteristický polynom matice A* rozumíme polynom

$$p(\lambda) := \det(A - \lambda I) \quad (4.1)$$

a pojmem *charakteristická rovnice matice A* rozumíme rovnici

$$p(\lambda) = 0. \quad (4.2)$$

Analytický postup pro nalezení vlastních čísel matice spočívá v sestavení charakteristického polynomu (4.2) a následně nalezení jeho kořenů, tj. řešení charakteristické rovnice (4.2) (viz např. [6]).



Obsah

158. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 4.3. Nalezněte vlastní čísla matice

$$A := \begin{pmatrix} 4 & -2 \\ 1 & 2 \end{pmatrix}.$$

Řešení. Nejprve sestavíme charakteristický polynom dosazením do předpisu (4.2)

$$p(\lambda) = \det(A - \lambda I) = \begin{vmatrix} 4 - \lambda & -2 \\ 1 & 2 - \lambda \end{vmatrix} = (4 - \lambda)(2 - \lambda) - (-2) \cdot 1 = \lambda^2 - 6\lambda + 10$$

a vlastní čísla jsou kořeny tohoto charakteristického polynomu, tj. jsou řešením rovnice

$$\lambda^2 - 6\lambda + 10 = 0.$$

Využitím např. diskriminantu získáme hodnoty $\lambda_1 = 3 + i$, $\lambda_2 = 3 - i$. ▲

Čtenář si jistě všiml základního nedostatku výše uvedeného postupu. Problém nečiní hledání kořenů charakteristického polynomu (lze použít například některý z algoritmů z kapitoly 2), ale sestavení samotného charakteristického polynomu pomocí symbolického výpočtu determinantu. Pro matici řádu n je nutno provést $(n - 1)n!$ součinů, což pro matice vyšších řádů je prakticky nemožné.



Obsah

159. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.1.1. Geršgorinova věta

Nejjednodušší postup, jak lokalizovat spektrum matice se skrývá v následující větě.

Věta 4.4. (Geršgorinova)

Všechny vlastní čísla matice $A \in \mathbb{R}^n$ leží ve sjednocení kruhů

$$S := S_1 \cup S_2 \cup \dots \cup S_n ,$$

kde

$$S_i := \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}|\}, \quad i = 1 \dots n .$$



Obsah

160. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 4.5. Nechť je dána matice

$$A = \begin{pmatrix} 10 & 1 & 1 & 2 \\ 1 & 5 & 1 & 0 \\ 1 & 1 & -5 & 0 \\ 2 & 0 & 0 & -10 \end{pmatrix}.$$

Lokalizujte spektrum této matice pomocí Geršgorinovy věty.

Řešení. Prostým dosazením do předpisu ve větě (4.4) získáme předpisy pro jednotlivé kruhy

$$S_1 := \{z \in \mathbb{C} : |z - 10| \leq 1 + 1 + 2 = 4\}$$

$$S_2 := \{z \in \mathbb{C} : |z - 5| \leq 1 + 1 + 0 = 2\}$$

$$S_3 := \{z \in \mathbb{C} : |z + 5| \leq 1 + 1 + 0 = 2\}$$

$$S_4 := \{z \in \mathbb{C} : |z + 10| \leq 2 + 0 + 0 = 2\}$$

a výsledek lze zakreslit do Gaussovy roviny, viz Obrázek 4.1.

Spektrum zadané matice leží ve sjednocení těchto kruhů. ▲



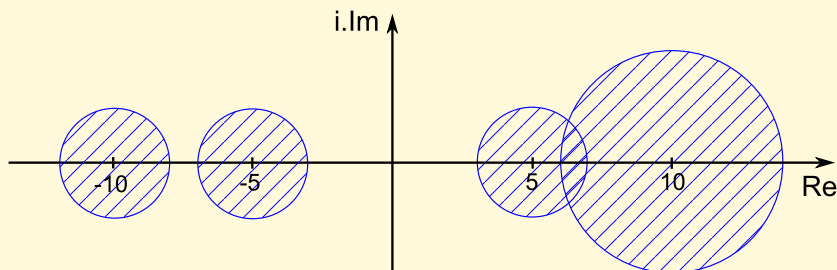
Obsah

161. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 4.1: Příklad 4.5, lokalizace spektra pomocí Geršgorinovy věty

Poznámka 4.6. O vlastních číslech reálné symetrické matice víme, že jsou reálná (viz [2]). V takovém případě se Geršgorinovy kruhy redukují na intervaly reálných čísel a spektrum matice pak leží ve sjednocení těchto intervalů.



Obsah

162. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

4.1.2. Krylovova metoda

V úvodu kapitoly jsme se zmínili, že jedním z možných postupů výpočtu vlastních čísel matice je řešení charakteristické rovnice. Výpočet charakteristického polynomu pomocí determinantu je však netriviální úloha. Ukažme si nyní, jak charakteristický polynom nalézt mnohem efektivněji.

Za tímto účelem uvažujme *charakteristický polynom matice* A ve tvaru

$$p(\lambda) := \lambda^n + \sum_{i=0}^{n-1} b_i \lambda^i, \quad (4.3)$$

kde λ je libovolné vlastní číslo matice A .

Obecně samozřejmě nemusí být zaručeno, že charakteristický polynom má koeficient u nejvyšší mocniny roven 1, nicméně na řešení charakteristické rovnice toto zjednodušení nemá vliv, neboť celou rovnici můžeme vydělit nenulovým koeficientem u nejvyšší mocniny. Metoda výpočtu charakteristického polynomu je založena na následující větě.

Věta 4.7. (*Cayley-Hamilton*)

Nechť $p(\lambda)$ je charakteristický polynom matice A .

Pak

$$p(A) = O.$$



Obsah

163. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Dosaďme do předpisu charakteristického polynomu (4.3) místo vlastního čísla matice matici A , tj.

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

Pak dle Věty (4.7) platí $p(A) = O$.

Přenásobíme-li tuto soustavu libovolným $y \in \mathbb{R}^n$ zprava, získáme

$$A^n y + \sum_{i=0}^{n-1} b_i A^i y = o$$

a po drobné úpravě

$$\sum_{i=0}^{n-1} b_i A^i y = -A^n y .$$

Předchozí rovnici lze zapsat v maticové formě

$$\tilde{A}b = \tilde{p} , \tag{4.4}$$

kde

$$\tilde{A} := (A^0 y, A^1 y, A^2 y, \dots, A^{n-1} y) , \quad \tilde{p} := -A^n y ,$$

$$b := (b_0, b_1, \dots, b_{n-1})^T .$$



Obsah

164. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Podařilo se nám nalézt postup, kterým lze získat koeficienty charakteristického polynomu bez nutnosti symbolického výpočtu determinantu jako řešení soustavy lineárních rovnic (4.4).

Algoritmus 10

KRYLOVOVA METODA

Input: A

- 1: $n :=$ velikost A
- 2: $y :=$ libovolný jednotkový vektor dimenze n
- 3: **for** $j = 1, \dots, n$ **do**
- 4: $s_j^{\tilde{A}} = A^{j-1}y$
- 5: **end for**
- 6: $\tilde{b} = -A^n y$
- 7: b je řešením soustavy $\tilde{A}b = \tilde{b}$

Output: $b_0, \dots, b_{n-1}$

Poznámka 4.8. Pro velké matice je však sestavení matice soustavy poměrně náročné, neboť obsahuje n násobení matice a vektoru.

[Obsah](#)

165. strana ze 415

[Zavřít dokument](#)[Celá obrazovka/Okno](#)

Příklad 4.9. Sestavte charakteristický polynom matice

$$A = \begin{pmatrix} 10 & 1 & 1 & 2 \\ 1 & 5 & 1 & 0 \\ 1 & 1 & -5 & 0 \\ 2 & 0 & 0 & -10 \end{pmatrix}.$$

pomocí Krylovovy metody.

Řešení. Nejdříve zvolíme libovolný vektor $y \in \mathbb{R}^n$. Jistě každého napadne, že nulový vektor není právě vhodnou volbou, jelikož soustava (4.4) by měla nulovou matici soustavy a nulový vektor pravých stran. Takové rovnici pak vyhovuje libovolné $b \in \mathbb{R}^n$.

Volme tedy

$$y = [1, 0, 0, 0]^T.$$

Sestavme soustavu (4.4). Pro jednotlivé sloupce matice $\tilde{A}$ a vektor $\tilde{p}$ platí

$$\begin{aligned} s_1^{\tilde{A}} &= A^0 y = y = (1, 0, 0, 0)^T \\ s_2^{\tilde{A}} &= A^1 y = A s_1^{\tilde{A}} = (10, 1, 1, 2)^T \\ s_3^{\tilde{A}} &= A^2 y = A s_2^{\tilde{A}} = (106, 16, 6, 0)^T \\ s_4^{\tilde{A}} &= A^3 y = A s_3^{\tilde{A}} = (1082, 192, 92, 212)^T \\ \tilde{p} &= -A^4 y = -A s_4^{\tilde{A}} = (-11528, -2134, -814, -44)^T. \end{aligned}$$

Hledaná soustava má tvar

$$\begin{pmatrix} 1 & 10 & 106 & 1028 \\ 0 & 1 & 16 & 192 \\ 0 & 1 & 6 & 92 \\ 0 & 2 & 0 & 212 \end{pmatrix} b = \begin{pmatrix} -11528 \\ -2134 \\ -814 \\ -44 \end{pmatrix}$$



Obsah

166. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

a jejím řešením (analyticky nebo lépe využitím libovolné metody z Kapitoly 3) je

$$b = (2684, -22, -132, 0)^T .$$

Dosazením hodnot do předpisu (4.3) získáme charakteristický polynom v tvaru

$$p(\lambda) = \lambda^4 - 132\lambda^2 - 22\lambda + 2684 .$$



Obsah

167. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.1.3. Výpočet charakteristického polynomu třídiagonální matice

V kapitolách (4.3.2) a (4.3.3) si představíme numerické metody pro převod reálných symetrických matic na třídiagonální matice. Je tedy důležité si ukázat, jak je možné efektivně sestavit charakteristický polynom třídiagonálních matic.

Definice 4.10. Pojmem *třídiagonální matice* rozumíme matici v tvaru

$$T_n := \begin{pmatrix} a_1 & b_1 & 0 & 0 & 0 & \dots & 0 \\ c_1 & a_2 & b_2 & 0 & 0 & \dots & 0 \\ 0 & c_2 & a_3 & b_3 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & c_{n-3} & a_{n-2} & b_{n-2} & 0 \\ 0 & \dots & 0 & 0 & c_{n-2} & a_{n-1} & b_{n-1} \\ 0 & \dots & 0 & 0 & 0 & c_{n-1} & a_n \end{pmatrix},$$

kde $a_i, b_j, c_j \in \mathbb{R}$, $i = 1, \dots, m$, $j = 1, \dots, n-1$.

Lemma 4.11. *Nechť je dána třídiagonální matice T_n .*

Pak pro charakteristický polynom matice T_n platí

$$p_n(\lambda) = (a_n - \lambda)p_{n-1}(\lambda) - b_{n-1}c_{n-1}p_{n-2}(\lambda),$$

kde p_{n-1} a p_{n-2} jsou charakteristické polynomy matic T_{n-1} a T_{n-2} a

$$p_0(\lambda) = 1, \quad p_1(\lambda) = a_1 - \lambda.$$



Obsah

168. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 4.12. Určete charakteristický polynom matice

$$A = \begin{pmatrix} 1 & -3 & 0 & 0 \\ 4 & 2 & -2 & 0 \\ 0 & 5 & 3 & -1 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

Řešení. Postupně určíme dle Věty 4.11 potřebné polynomy

$$p_0(\lambda) = 1$$

$$p_1(\lambda) = a_1 - 1 = 1 - \lambda$$

$$p_2(\lambda) = (a_2 - \lambda)p_1(\lambda) - b_1c_1p_0(\lambda) = (2 - \lambda)(1 - \lambda) + 12 = \lambda^2 - 3\lambda + 14$$

$$p_3(\lambda) = (a_3 - \lambda)p_2(\lambda) - b_1c_1p_1(\lambda) = (3 - \lambda)(\lambda^2 - 3\lambda + 14) + 10 \cdot (1 - \lambda) \\ = -\lambda^3 + 6\lambda^2 - 33\lambda + 52$$

$$p_4(\lambda) = (a_4 - \lambda)p_3(\lambda) - b_1c_1p_2(\lambda) \\ = (3 - \lambda)(-\lambda^3 + 6\lambda^2 - 33\lambda + 52) + 0 \cdot (\lambda^2 - 3\lambda + 14) \\ = \lambda^4 - 8\lambda^3 + 45\lambda^2 - 118\lambda + 104.$$

Tedy charakteristický polynom dané matice má tvar

$$p(\lambda) = \lambda^4 - 8\lambda^3 + 45\lambda^2 - 118\lambda + 104.$$



Obsah

169. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 4.13. Numerický výpočet koeficientů $p_n(\lambda)$ však bohužel neumožňuje symbolickou manipulaci s proměnnou λ . Nicméně polynomy lze nahradit aritmetickými vektory, jejichž složky reprezentují koeficienty u mocnin λ .

Označme koeficienty v polynomu $p_n(\lambda)$ jako vektor $p_n \in \mathbb{R}^{n+1}$ (daný polynom je n -tého stupně, proto bude mít $n + 1$ koeficientů). Tedy polynom ve tvaru

$$p_n(\lambda) = \gamma_0 \lambda^0 + \gamma_1 \lambda^1 + \cdots + \gamma_n \lambda^n$$

bude mít příslušný vektor koeficientů

$$p_n = (\gamma_0, \gamma_1, \dots, \gamma_n) .$$

Zbývá odvodit rekurentní předpis ve Větě 4.11 pro vektory koeficientů.

Nechť

$$\begin{aligned} p_{n-2} &= (\alpha_0, \alpha_1, \dots, \alpha_{n-2}) \\ p_{n-1} &= (\beta_0, \beta_1, \dots, \beta_{n-1}) , \end{aligned}$$

pak

$$p_n = (\gamma_0, \gamma_1, \dots, \gamma_n) ,$$

kde

$$\begin{aligned} \gamma_0 &= \alpha_n \beta_0 - b_{n-1} c_{n-1} \alpha_0 \\ \gamma_i &= \alpha_n \beta_i - b_{n-1} c_{n-1} \alpha_i, \quad i = 1, \dots, n-2 \\ \gamma_{n-1} &= \alpha_n \beta_{n-1} + \beta_{n-2} \\ \gamma_n &= -\beta_{n-1} . \end{aligned}$$



Obsah

170. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.2. Výpočet dominantního vlastního čísla

V následujícím textu se zaměříme na metody řešící částečný problém vlastních čísel. Zejména si představíme metody, které nám umožní vypočítat tzv. *dominantní* vlastní číslo. Případné další vlastní čísla budeme schopni dopočítat pomocí maticové redukce či posunem spektra. Nejprve si však zavedeme pojem dominantního vlastního čísla.

Definice 4.14. Nechť je dána matice A a vlastní čísla splňující nerovnost

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$$

a nechť $x_1, x_2, x_3, \dots, x_n \in \mathbb{R}^n$ jsou odpovídající lineárně nezávislé vlastní vektory.

Pak *dominantním vlastním číslem* rozumíme vlastní číslo λ_1 .

V následujícím výkladu se zaměříme na mocninnou metodu, která dokáže nalézt dominantní vlastní číslo.



Obsah

171. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.2.1. Mocninná metoda

Předpokládejme, že

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$$

jsou vlastní čísla matice A řádu n , a necht $U = \{x_1, \dots, x_n\}$ je množina jim odpovídajících lineárně nezávislých vektorů. Pak množina U tvoří bázi vektorového prostoru $\mathbb{R}^n$ a libovolný vektor $v_0 \in \mathbb{R}^n$ lze vyjádřit jako lineární kombinaci prvků množiny U , tj. existují reálné konstanty $\alpha_1, \dots, \alpha_n$ takové, že

$$v_0 = \alpha_1 x_1 + \dots + \alpha_n x_n . \quad (4.5)$$

Definujme posloupnost $\{v_k\}, v_k \in \mathbb{R}^n$ takovou, že

- v_0 je libovolné takové, že $(v_0, x_1) \neq 0$
- v_k vypočteme předpisem

$$v_k = A v_{k-1} . \quad (4.6)$$

Dosazením lineární kombinace (4.5) do předpisu (4.6) získáme

$$v_k = A v_{k-1} = A^k v_0 = A_k \sum_{i=1}^n \alpha_i x_i = \sum_{i=1}^n \alpha_i A^k x_i = \sum_{i=1}^n \alpha_i \lambda_i^k x_i .$$

Tento předpis se nám hodí v důkazu Věty 4.16.



Obsah

172. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Definice 4.15. (Schwarzova konstanta)

Nechť $y \in \mathbb{R}^n$ takové, že $(y, x_1) \neq 0$, kde x_1 je vlastní vektor odpovídající dominantnímu vlastnímu číslu.

Číslo

$$\sigma_k := (y, v_k) = (y, A^k v_0)$$

nazýváme *Schwarzovou konstantou*.

Lemma 4.16. *Podíl Schwarzových konstant*

$$\frac{\sigma_{k+1}}{\sigma_k} = \frac{(y, A^{k+1} v_0)}{(y, A^k v_0)}$$

konverguje k dominantnímu vlastnímu číslu λ_1 , tj.

$$\lim_{k \rightarrow \infty} \frac{\sigma_{k+1}}{\sigma_k} = \lambda_1 .$$



Obsah

173. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Algoritmus 11

MOCNINNÁ METODA

Input: v_0, y , přesnost $\epsilon > 0$

```

1:  $v_1 := Av_0$ 
2:  $\lambda_{1,1} := (y, v_1)/(y, v_0)$ 
3:  $k := 1$ 
4: while  $\|Av_k - \lambda_{1,k}v_k\| \geq \epsilon$  do
5:    $v_{k+1} := Av_k$ 
6:    $\lambda_{1,k+1} := (y, v_{k+1})/(y, v_k)$ 
7:    $k := k + 1$ 
8: end while
    
```

Output: $\lambda_{1,k}, v_k$

Poznámka 4.17. V průběhu algoritmu může nastat, že složky vlastního vektoru budou velmi rychle růst. Toto je důsledek toho, že $v_{k+1} = A^k v_0$ a s vysokými mocninami se budou složky velmi zvětšovat. To má ovšem za následek velké chyby v počítačové aritmetice. Proto je nutné použít normalizaci vlastních vektorů.



Obsah

174. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Algoritmus 12

NORMOVANÁ MOCNINNÁ METODA

Input: v_0 , přesnost $\epsilon > 0$

```
1:  $y_0 := v_0 / \|v_0\|$ 
2:  $v_1 := Ay_0$ 
3:  $y_1 := v_1 / \|v_1\|$ 
4:  $\lambda_{1,1} := \|v_1\|$ 
5:  $k := 1$ 
6: while  $\|Ay_k - \lambda_{1,k}y_k\| \geq \epsilon$  do
7:    $v_{k+1} := Ay_k$ 
8:    $y_{k+1} := v_{k+1} / \|v_{k+1}\|$ 
9:    $\lambda_{1,k+1} := \|v_{k+1}\|$ 
10:   $k := k + 1$ 
11: end while
```

Output: $\lambda_{1,k}, y_k$ 

Obsah

175. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Příklad 4.18. Otestujte algoritmus normované mocinné metody na matici

$$A \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 2 & 1 & -1 \\ -2 & 0 & 3 & 5 \\ 4 & -1 & 0 & 2 \end{pmatrix}.$$

Nalezněte touto metodou dominantní vlastní číslo s přesností $\epsilon = 10^{-4}$.

Řešení. Jako počáteční vektor volme $v_0 = (1, 0, 0, 0)^T$. Pak velikost dominantního vlastního čísla λ_1 v průběhu iterací je znázorněn na Obrázku 4.2. Algoritmus je ukončen po 12-ti iteracích, dominantním vlastním číslem s požadovanou přesností je

$$\lambda_1 = 6.5445.$$



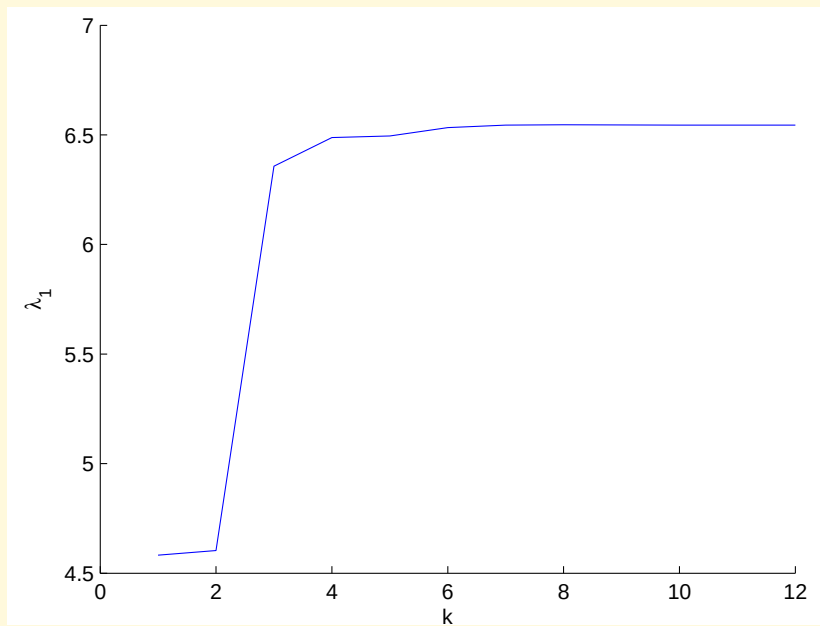
Obsah

176. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Obr. 4.2: Příklad 4.18 - numerický test algoritmu normované mocninné metody



Obsah

177. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pro symetrické matice počítáme místo podílů Schwarzových konstant tzv. *Rayleighovy podíly*:

$$r(v_k) \stackrel{\text{def}}{=} \frac{(v_k, v_{k+1})}{(v_k, v_k)}, \quad \lim_{k \rightarrow \infty} \frac{(v_k, v_{k+1})}{(v_k, v_k)} = \lambda_1 .$$

Pokud však uvažujeme pouze symetrické pozitivně definitní matice A , pak jsou všechna vlastní čísla kladná reálná, tedy je lze uspořádat

$$\lambda_1 > \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n . \quad (4.7)$$

Kromě konvergence k největšímu vlastnímu číslu mají Rayleighovy podíly mnohem silnější vlastnost.

Lemma 4.19. *Nechť $A \in \mathbb{R}^{n,n}$ je symetrická pozitivně definitní reálná matice s vlastními čísly (4.7). Pak pro libovolné $v \in \mathbb{R}^n$ platí*

$$\lambda_n v^T v \leq v^T A v \leq \lambda_1 v^T v . \quad (4.8)$$

Poznámka 4.20. Uvažujeme-li $v \neq 0$, pak rovnici (4.8) lze upravit

$$\lambda_n \leq \frac{v^T A v}{v^T v} \leq \lambda_1$$

a získat tak odhad pro Rayleighovy podíly.



Obsah

178. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.2.2. Posun a redukce spektra

Mocninná metoda slouží pouze k výpočtu dominantního vlastního čísla. Pokud bychom chtěli vypočítat i jiná vlastní čísla, museli bychom využít některých maticových operací, které upravují spektrum požadovaným způsobem.

Podívejme se na některé z nich.



Obsah

179. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Lemma 4.21. (*Posun spektra*)

Nechť λ je vlastní číslo matice A a v odpovídající vlastní vektor. Pak matice

$$A - qI, \quad q \in \mathbb{R}$$

má vlastní číslo $\lambda - q$ a stejný vlastní vektor v .

Poznámka 4.22. Výše uvedené věty se dá využít například na zjištění nejmenšího vlastního čísla mocninnou metodou. Je-li

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_{n-1}| > |\lambda_n|$$

Pak $A - \lambda_1 I$ má vlastní čísla

$$0, \lambda_2 - \lambda_1, \dots, \lambda_{n-1} - \lambda_1, \lambda_n - \lambda_1$$

přičemž bude platit

$$|\lambda_n - \lambda_1| > |\lambda_{n-1} - \lambda_1| > \dots > |\lambda_2 - \lambda_1| > 0$$

Použijeme-li tedy mocninnou metodu na takto upravenou matici, dostaneme dominantní vlastní číslo $\tilde{\lambda} := \lambda_n - \lambda_1$ a odtud

$$\lambda_n = \tilde{\lambda} + \lambda_1.$$



Obsah

180. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Lemma 4.23. (Maticová redukce)

Nechť λ, v jsou vlastní číslo a vlastní vektor matice A . Nechť w je levý vlastní vektor λ , tj. $A^T w = \lambda w$, a nechť $(v, w) = 1$.

Pak matice

$$W \stackrel{\text{def}}{=} A - \lambda v w^T$$

má místo vlastního čísla λ nulu a všechny ostatní vlastní čísla stejná.

Poznámka 4.24. Maticové redukce se dá využít k výpočtu druhého dominantního čísla. Je-li

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|,$$

můžeme vypočíst pomocí mocninné metody λ_1 a jemu odpovídající vlastní vektor v_1 i levý vlastní vektor $\tilde{w}_1$ aplikací mocninné metody na matici A^T . Normováním

$$w_1 = \frac{\tilde{w}_1}{(v_1, \tilde{w}_1)},$$

pak splníme podmínku $(v_1, w_1) = 1$. Použitím maticové redukce $W = A - \lambda_1 v_1 w_1^T$ můžeme mocninou metodou spočítat vlastní číslo λ_1 a jemu odpovídající vlastní vektor, neboť původní dominantní vlastní číslo λ_1 se maticovou redukcí vynulovalo.

Pochopitelně lze celý proces opakovat a získat tak λ_3 , atd.

Výpočet všech vlastních čísel je však tímto postupem výpočetně velmi náročný a nepoužívá se. Vhodnější je využití některé z metod pro úplný problém vlastních čísel.



Obsah

181. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.3. Metody založené na podobnosti matic

Na rozdíl od předchozí Kapitoly 4.2, kdy jsme se zabývali hledáním dominantního vlastního čísla, v dalším textu představíme metody pro hledání všech vlastních čísel a vlastních vektorů.

Úlohu, která řeší všechny vlastní čísla a vektory matice nazýváme *úplný problém vlastních čísel*. Všechny metody, které si v následujícím textu představíme budou založeny na základní vlastnosti, které říkáme *podobnost matic*.

Definice 4.25. O maticích $A, B \in \mathbb{R}^{n,n}$ řekneme, že jsou *podobné*, existuje-li regulární matice T taková, že

$$B = TAT^{-1} . \quad (4.9)$$

Následující lemma naznačuje důvod zavedení pojmu *podobné matice* v tomto textu.

Lemma 4.26. *Podobné matice mají stejná vlastní čísla.*



Obsah

182. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

V následujícím výkladu se zaměříme na metody založené na Větě 4.26. Nebudeme hledat vlastní čísla zadané matice A , nýbrž se budeme snažit nalézt vhodnou regulární matici T , která původní matici A převede na „jednodušší matici“ B , která bude buď diagonální nebo alespoň třídiagonální.

Z důvodu lepší stability výpočtů se jeví jako vhodnější použití místo regulární matice T tzv. matice *ortogonální*. Pokud navíc však zvolíme ortogonální matici T , výpočet inverze nebude nutný.

Definice 4.27. (Ortogonální matice)

Ortogonální matice je symetrická regulární matice pro kterou platí

$$Q^{-1} = Q^T .$$

Pak pro každou ortogonální matici platí $Q^{-1}Q = Q^TQ = I$.

Lemma 4.28. *Matice $A \in \mathbb{R}^{n,n}$ je ortogonální, právě když její řádky (případně sloupce) tvoří ortonormální množinu vektorů.*

Pokud Q je ortogonální matice, lze předpis pro podobné matice (4.9) zjednodušit na tvar

$$B = QAQ^T . \quad (4.10)$$



Obsah

183. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.3.1. Algoritmus QR

Věta 4.29. *Nechť $A \in \mathbb{R}^{n,n}$ je regulární matice. Pak existují horní trojúhelníková matice $R \in \mathbb{R}^{n,n}$ a ortogonální matice $Q \in \mathbb{R}^{n,n}$ takové, že*

$$A = QR . \quad (4.11)$$

Jelikož Q v rozkladu (4.11) je ortogonální, lze ji využít jako matici podobnostní transformace. Nechť $A_k = Q_k R_k$, pak předpis pro novou aproximaci bude mít tvar

$$A_{k+1} = Q_k^T A_k Q_k = Q_k^T (Q_k R_k) Q_k = R_k Q_k, \quad A_0 = A . \quad (4.12)$$

Věta 4.30. *Nechť matice A je reálná a symetrická a nechť $A_0, A_1, A_2, \dots, A_k, \dots$ je posloupnost generovaná rekursivním předpisem*

$$A_0 = A, \quad A_{k+1} = R_k Q_k, \quad k > 0 ,$$

kde matice Q_k, R_k jsou QR rozkladem matice A_k . Pak

$$\lim_{k \rightarrow \infty} A_k = D ,$$

kde D je diagonální matice s vlastními čísly matice A na diagonále.



Obsah

184. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 4.31. Věta 4.30 garantuje konvergenci matice A_k k diagonální matici, avšak obecně po nekonečně mnoha podobnostních transformacích

$$Q_k^T A_k Q_k .$$

Při praktickém výpočtu se spokojíme s maticí $\tilde{D}$, která má mimodiagonální prvky pouze blízké nule.

Tato situace nastane po m krocích. Tedy výsledkem bude $\tilde{D} = \tilde{Q}^T A \tilde{Q}$, kde

$$\tilde{Q} := Q_1 Q_2 \dots Q_m$$

je ortogonální matice jejíž sloupce aproximují ortonormální vlastní vektory odpovídající vlastním číslům v matici D .

Diagonální prvky matice $\tilde{D}$ jsou aproximacemi vlastních čísel v matici D .



Obsah

185. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

QR algoritmus pro výpočet vlastních čísel a vlastních vektorů matice A , můžeme shrnout do následujících kroků.

Algoritmus 13

ALGORITMUS QR

Input: A , přesnost $\epsilon > 0$

```
1:  $A_0 := A$ 
2:  $n :=$  velikost  $A$ 
3:  $k := 0$ 
4: while  $\exists [A_k]_{i,j} > \epsilon, \quad i, j = 1, \dots, n : i \neq j$  do
5:     najdi  $Q_k, R_k$  jako QR rozklad matice  $A_k$ 
6:      $A_{k+1} := R_k Q_k$ 
7:      $k := k + 1$ 
8: end while
```

Output: A_k

Zbývá otázka, jak určit QR rozklad dané regulární matice.



Obsah

186. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Matici A lze převést na ortogonální matici pomocí Gramova-Schmidtova ortonormalizačního procesu, který z množiny lineárně nezávislých vektorů $\{s_1^A, \dots, s_n^A\}$ (ze sloupců regulární matice A) vytvoří množinu ortonormálních vektorů $\{s_1^Q, \dots, s_n^Q\}$ (ortonormální sloupce ortogonální matice Q). Grammův-Schmitův ortonormalizační algoritmus lze nalézt například v [2].

V následujícím výkladu budeme pomocí $s_1^A, s_2^A, \dots, s_n^A$ značit sloupce matice A , pomocí $\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_n$ ortogonální sloupcové vektory vytvořené Gramovým-Schmitovým ortogonalizačním procesem a výsledné ortonormální sloupce matice Q pomocí $s_1^Q, s_2^Q, \dots, s_n^Q$.

Pak Grammův-Schmitův ortonormalizační proces lze vyjádřit pomocí následujících rovnic

$$\begin{aligned} \tilde{s}_1 &= s_1^A & s_1^Q &= \frac{1}{\|\tilde{s}_1\|} \tilde{s}_1 \\ \tilde{s}_2 &= s_2^A + \alpha_{1,2} s_1^Q & s_2^Q &= \frac{1}{\|\tilde{s}_2\|} \tilde{s}_2 \\ \tilde{s}_3 &= s_3^A + \alpha_{1,3} s_1^Q + \alpha_{2,3} s_2^Q & s_3^Q &= \frac{1}{\|\tilde{s}_3\|} \tilde{s}_3 \\ &\vdots & &\vdots \\ \tilde{s}_n &= s_n^A + \alpha_{1,n} s_1^Q + \alpha_{2,n} s_2^Q + \dots + \alpha_{n-1,n} s_{n-1}^Q & s_n^Q &= \frac{1}{\|\tilde{s}_n\|} \tilde{s}_n \end{aligned}$$

kde α jsou příslušné ortogonalizační konstanty.



Obsah

187. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Jednoduchou úpravou předchozích rovnic lze získat

$$\begin{aligned} s_1^A &= \|\tilde{s}_1\| s_1^Q \\ s_2^A &= -\alpha_{1,2} s_1^Q + \|\tilde{s}_2\| s_2^Q \\ s_3^A &= -\alpha_{1,3} s_1^Q - \alpha_{2,3} s_2^Q + \|\tilde{s}_3\| s_3^Q \\ &\vdots \\ s_n^A &= -\alpha_{1,n} s_1^Q - \alpha_{2,n} s_2^Q - \dots - \alpha_{n-1,n} s_{n-1}^Q + \|\tilde{s}_n\| s_n^Q \end{aligned}$$

nebo maticově

$$A = Q \begin{pmatrix} \|\tilde{s}_1\| & -\alpha_{1,2} & -\alpha_{1,3} & \dots & -\alpha_{1,n} \\ 0 & \|\tilde{s}_2\| & -\alpha_{2,3} & \dots & -\alpha_{2,n} \\ 0 & 0 & \|\tilde{s}_3\| & \dots & -\alpha_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \|\tilde{s}_n\| \end{pmatrix} = QR .$$



Obsah

188. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Předchozí pozorování lze šikovně zakomponovat do výsledného algoritmu.

Algoritmus 14
KONSTRUKCE QR ROZKLADU

Input: A

```

1:  $n :=$  velikost  $A$ 
2:  $R :=$  nulová matice typu  $n \times n$ 
3: for  $j := 1, \dots, n$  do
4:    $v := s_j^A$ 
5:   for  $i := 1, \dots, j-1$  do
6:      $r_{i,j} := (s_i^Q)^T s_j^A$ 
7:      $v := v - r_{i,j} s_i^Q$ 
8:   end for
9:    $r_{j,j} := \|v\|$ 
10:   $s_j^Q := v / \|v\|$ 
11: end for

```

Output: Q, R



Obsah

189. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 4.32. Nalezněte QR rozklad matice

$$A := \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Řešení. Nejprve pomocí Gramova-Schmidtova ortonormalizačního procesu zkonstruujeme sloupce matice Q .

$$\begin{aligned} s_1^Q &= \frac{s_1^A}{\|s_1^A\|} = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right] \\ s_2^Q &= \frac{s_2^A - (s_2^A, s_1^Q) s_1^Q}{\|s_2^A - (s_2^A, s_1^Q) s_1^Q\|} = \left[\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}} \right] \\ s_3^Q &= \frac{s_3^A - (s_3^A, s_2^Q) s_2^Q - (s_3^A, s_1^Q) s_1^Q}{\|s_3^A - (s_3^A, s_2^Q) s_2^Q - (s_3^A, s_1^Q) s_1^Q\|} = \left[-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right], \end{aligned}$$

pak

$$Q = [s_1^Q, s_2^Q, s_3^Q] = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{pmatrix}.$$

Pro konstrukci matice R využijeme předpis (4.11), získáme

$$R = Q^T A = \begin{pmatrix} (s_1^Q, s_1^A) & (s_1^Q, s_2^A) & (s_1^Q, s_3^A) \\ (s_2^Q, s_1^A) & (s_2^Q, s_2^A) & (s_2^Q, s_3^A) \\ (s_3^Q, s_1^A) & (s_3^Q, s_2^A) & (s_3^Q, s_3^A) \end{pmatrix} = \begin{pmatrix} \frac{2}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{3}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{pmatrix}.$$



Obsah

190. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 4.33. Otestujte rychlost konvergence algoritmu QR na různě-diagonálních testovacích maticích

$$\begin{aligned}A_1 &= \text{diag } (5) \\A_3 &= 3\text{-diag } (-1, 5, -1) \\A_5 &= 5\text{-diag } (-1, -1, 5, -1, -1) \\A_7 &= 7\text{-diag } (-1, -1, -1, 5, -1, -1, -1) \\A_9 &= 9\text{-diag } (-1, -1, -1, -1, 5, -1, -1, -1, -1) .\end{aligned}$$

Volte různé dimenze testovacích matic.

Řešení. Pro různou volbu dimenze matic n jsou počty iterací zobrazeny na Obrázku 4.3.



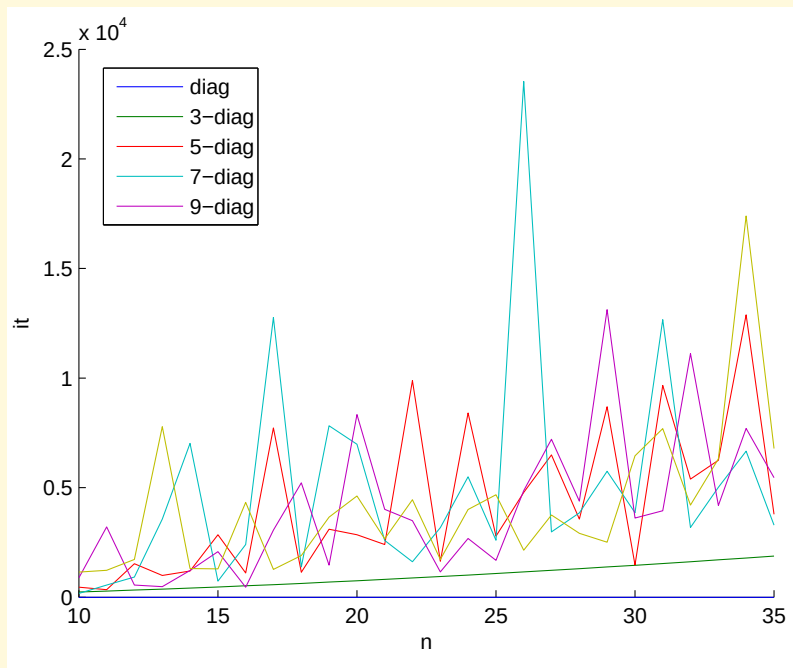
Obsah

191. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 4.3: Rychlost konvergence QR a různě diagonální matice



Obsah

192. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 4.34. Výpočet QR rozkladu pomocí Gramova-Schmidtova procesu je pro velké dimenze matic numericky nestabilní. Proto se v praxi k výpočtu QR rozkladu používají ortogonální matice, jako je Givensova matice rovinné rotace nebo Householderova matice rovinného zrcadlení, viz [1].

Tyto matice využijeme i v dalším textu při řešení problému vlastních čísel.



Obsah

193. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.3.2. Givensova metoda

V předchozí kapitole jsme si popsali QR algoritmus, který postupně nuluje mimodiagonální prvky. Tento postup je však poměrně zdlouhavý.

Proto se nabízí postup, který by v konečném počtu kroků převedl matici A pomocí podobnostních transformací na matici třídiagonální. Na tuto matici pak lze aplikovat zmiňovaný QR algoritmus, nebo lze přímo vyčíslit charakteristický polynom. K podobnostním transformacím využijeme ortogonálních matic rovinné rotace.

Definice 4.35. Maticí rovinné rotace rozumíme matici s předpisem

$$R_{i,j}^{\varphi} := \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & \cos \varphi & \dots & \sin \varphi \\ & & & \vdots & \ddots & \vdots \\ & & & -\sin \varphi & \dots & \cos \varphi \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix}, \quad (4.13)$$

ve které se goniometrické funkce nachází v i -tém a j -tém řádku a sloupci.

Tato matice otáčí bod v rovině (i, j) o úhel φ kolem počátku.

Poznámka 4.36. Matice rovinné rotace je ortogonální.



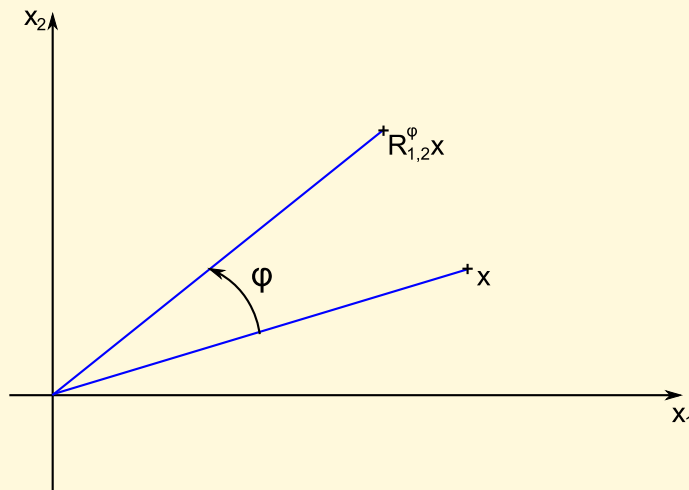
Obsah

194. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 4.4: Givensovy rotace

Cílem Givensovy metody je nalézt ortogonální matici R takovou, že

$$T := R^T A R \quad (4.14)$$

je třídiagonální matice. Přitom matice A a T jsou podobné a tudíž mají stejná vlastní čísla.

Hledejme matici rovinné rotace $R_{p,q,r}$ v rovině (p, q) , která pomocí transformace

$$\tilde{A} = R_{p,q,r}^T A \quad (4.15)$$



Obsah

195. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

nuluje prvek $[A]_{q,r}$ a mění jen prvky v řádcích p, q .

Prvek $[A]_{q,r}$ získáme jako skalární součin q -tého řádku matice $R_{p,q,r}^T$ a r -tého sloupce matice A , tj.

$$\tilde{A}_{q,r} = (r_q^{R^T}, s_r^A) = \cos \varphi A_{p,r} - \sin \varphi A_{q,r} = 0 .$$

Pokud nyní přenásobíme rovnici koeficientem

$$\alpha := \frac{1}{\sqrt{A_{p,r}^2 + A_{q,r}^2}} ,$$

pak získáme rovnici

$$\cos \varphi (\alpha A_{p,r}) - \sin \varphi (\alpha A_{q,r}) = 0 ,$$

s triviálním řešením

$$\begin{aligned} \cos \varphi &= \alpha A_{q,r} \\ \sin \varphi &= \alpha A_{p,r} , \end{aligned} \quad (4.16)$$

které splňuje podmínky oboru hodnot goniometrických funkcí $\cos \varphi, \sin \varphi \in \langle -1, 1 \rangle$. Toto řešení lze přímo využít pro sestavení dané matice rotace, není potřeba vyčíslovat goniometrické nebo cyklometrické funkce.

Pokud provedeme na matici A podobnostní transformaci

$$\tilde{A} = R_{p,q,r}^T A R_{p,q,r} , \quad (4.17)$$

mění se prvky v řádcích p, q a sloupcích p, q . Pokud transformace (4.17) nuluje prvek $[A]_{r,q}$, je také prvek na pozici $[A]_{q,r}$ vynulován. Navíc matice A a $\tilde{A}$ jsou podobné.



Obsah

196. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 4.37. Necht' je dána matice

$$A = \begin{bmatrix} 4 & -1 & -1 & 2 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ 2 & -1 & -1 & 4 \end{bmatrix}.$$

Nalezněte matici B , která má na pozici $[3, 1]$ nulový prvek, tj. $[B]_{3,1} = 0$, a je s maticí A podobná. Využijte vhodnou matici Givensových rotací.

Řešení. Nejdříve dosadíme do předpisů (4.16) dané hodnoty $r = 3, q = 1$ (dle pozice nulovaného prvku $[A]_{r,q}$) a libovolné $p \in \{2, 4\}$.

Při volbě $p = 2$ získáme

$$\begin{aligned} \sin \varphi &= -\frac{[A]_{3,1}}{\sqrt{[A]_{3,2}^2 + [A]_{3,1}^2}} = \frac{1}{\sqrt{2}} \\ \cos \varphi &= -\frac{[A]_{3,2}}{\sqrt{[A]_{3,2}^2 + [A]_{3,1}^2}} = -\frac{1}{\sqrt{2}} \end{aligned}$$

a dosadíme do předpisu matice Givensových rotací (4.3.2) na pozice p, q

$$R_{2,1}^\varphi = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 & 0 \\ \sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_{2,1,3} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$



Obsah

197. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Následně provedeme transformaci podobnosti (4.14)

$$B = R_{2,1,3}^T A R_{2,1,3} = \begin{bmatrix} 5 & 0 & 0 & -\frac{3}{\sqrt{2}} \\ 0 & 3 & \frac{2}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{2}{\sqrt{2}} & 4 & -1 \\ -\frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -1 & 4 \end{bmatrix}.$$

Při volbě $p = 4$ dopadne situace obdobně

$$R_{4,1,3} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad B = R_{4,1,3}^T A R_{4,1,3} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 4 & -1 & \frac{2}{\sqrt{2}} \\ 0 & -1 & 4 & \frac{2}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{2}} & 6 \end{bmatrix}.$$



Obsah

198. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Algoritmus Givensových rotací je pak založen na opakování podobnostních transformací

$$A_k = R_k^T A_{k-1} R_k, \quad k = 1, \dots, m, \quad A_0 = A,$$

kde matice R_k postupně nulují prvky mimo třídiagonály.

Hledaná matice podobnostní transformace má tedy tvar

$$R = R_1 R_2 R_3 \dots R_m.$$

Indexy p, q, r v předpisu (4.17) je však nutno volit tak, aby algoritmus nezaplňoval prvky, které byly předchozích krocích již vynulované. K nulování je proto nutno přistupovat systematicky.

Na následujícím schématu je pořadovými čísly naznačen postup nulování na matici $\mathbb{R}^{6,6}$.

$$\begin{bmatrix} x & x & 1 & 2 & 3 & 4 \\ x & x & x & 5 & 6 & 7 \\ 1 & x & x & x & 8 & 9 \\ 2 & 5 & x & x & x & 10 \\ 3 & 6 & 8 & x & x & x \\ 4 & 7 & 9 & 10 & x & x \end{bmatrix}$$



Obsah

199. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pak algoritmus převodu symetrické matice na třídiagonální pomocí Givensových rotací má tvar

Algoritmus 15GIVENSOVA METODA

Input: A

```
1:  $T := A$ 
2:  $n :=$  velikost  $A$ 
3: for  $k := 2, \dots, n$  do
4:   for  $i := k + 1, \dots, n$  do
5:     sestav  $R_{k,i,k-1}$ 
6:      $T := R_{k,i,k-1}^T T R_{k,i,k-1}$ 
7:   end for
8: end for
```

Output: T



Obsah

200. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 4.38. Transformace s maticí $R_{k,i,k-1}$ znamená, že je nulován prvek $[A]_{k-1,i}$ a přitom se mění pouze řádky a sloupce s indexy k a i . Navíc ty řádky, které jsou již vynulovány, zůstávají nulové.

$$\begin{aligned}
 A_1 &= R_{2,3,1}^T A_0 R_{2,3,1} \\
 A_2 &= R_{2,4,1}^T A_1 R_{2,4,1} \\
 A_3 &= R_{3,4,2}^T A_2 R_{3,4,2}
 \end{aligned}$$

V případě symetrické matice je výsledkem třídiagonální matice podobná původní matici. Hledat vlastní čísla pak můžeme iteračně algoritmem QR (viz Kapitola 4.3.1) nebo přes charakteristický polynom třídiagonální matice (viz Kapitola 4.1.3).



Obsah

201. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 4.39. Algoritmus můžeme použít i na nesymetrické matice. Pak obdržíme matici v tzv. dolním *Hessenbergově tvaru*, t.j

$$\begin{bmatrix} x & x & 0 & 0 & \dots & 0 \\ x & x & x & 0 & \dots & 0 \\ x & x & x & x & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ x & x & x & x & \dots & x \\ x & x & x & x & \dots & x \end{bmatrix}$$

V tomto případě je také možné, podobně jako u třídiagonální matice, nalézt rekursivní vztah pro nalezení charakteristického polynomu.

[Obsah](#)

202. strana ze 415

[Zavřít dokument](#)[Celá obrazovka/Okno](#)

4.3.3. Householderova metoda

Obdobně jako matice rovinné rotace v případě Givensovy transformace, můžeme k výpočtu QR rozkladu použít tzv. *matice rovinného zrcadlení*. Aplikaci matic rovinného zrcadlení taktéž říkáme *Householderovy transformace*.

Definice 4.40. *Maticí rovinného zrcadlení* rozumíme matici s předpisem

$$P := I - 2ww^T = \begin{pmatrix} 1 - 2w_1^2 & -2w_1w_2 & \dots & -2w_1w_n \\ -2w_2w_1 & 1 - 2w_2^2 & \dots & -2w_2w_n \\ \vdots & \vdots & \ddots & \vdots \\ -2w_nw_1 & -2w_nw_2 & \dots & 1 - 2w_n^2 \end{pmatrix} \quad (4.18)$$

kde $w^T w = 1$.

Poznámka 4.41. Matice rovinného zrcadlení je ortogonální a symetrická.



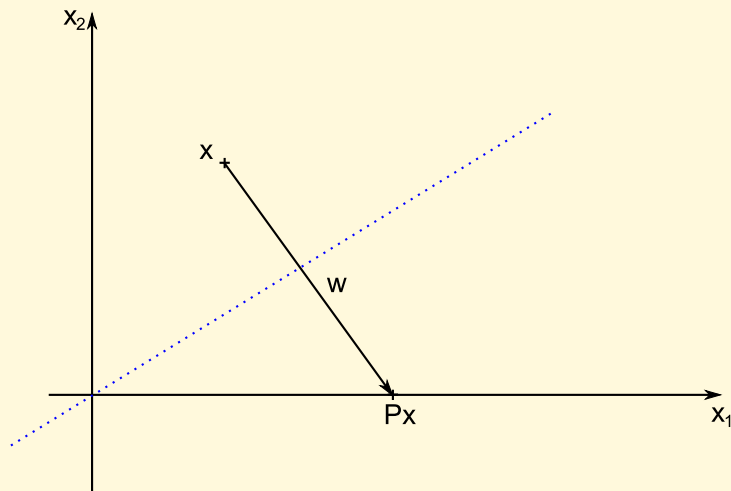
Obsah

203. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 4.5: Householderovo zrcadlení

Ještě efektivněji než-li v případě Givensovy metody můžeme pro transformaci matice na tridiagonální tvar využít Housholderových transformací pomocí matic rovinného zrcadlení.

Vektory w_j pro konstrukci matice $P_j = I - 2w_j w_j^T$ můžeme brát ve tvaru

$$w_j := [0, \dots, 0, w_{j+1}^{(j)}, \dots, w_n^{(j)}]^T.$$

Vhodnou volbou koeficientů $w^{(j)}$ lze docílit toho, že matice $A_j = P_j A_{j-1} P_j$ má všechny



Obsah

204. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

prvky

$$[A]_{i,k} = 0, \quad \text{pro } i + 1 > k, k = 1, \dots, n$$

tzn. že v každém kroku se nulují prvky

$$[A]_{i,j}, i = j + 2, \dots, n .$$

Touto volbou je vektor se složkami

$$w_{j+1}^{(j)} = \frac{s_j + |[A_{j-1}]_{j+1,j}|}{2s_j}$$

$$w_k^{(j)} = \operatorname{sgn} ([A_{j-1}]_{j+1,j}) \frac{[A_{j-1}]_{k,j}}{2w_{j+1}^{(j)} s_j}, \quad k = j + 2, \dots, n$$

kde

$$s_j \stackrel{\text{def}}{=} \sqrt{\sum_{k=j+1}^n ([A_{j-1}]_{k,j})^2}$$

a $[A_{j-1}]_{k,j}$ jsou prvky matice A_{j-1} .



Obsah

205. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Algoritmus 16

HOUSEHOLDEROVA METODA

Input: A

```

1:  $T := A$ 
2:  $n := \text{velikost } A$ 
3: for  $j := 1, \dots, n - 2$  do
4:    $s := \sqrt{\sum_{k=j+1}^n [T]_{k,j}^2}$ 
5:    $[w]_{j+1} := (s + |[T]_{j+1,j}|) / (2s)$ 
6:   for  $k := j + 2, \dots, n$  do
7:      $[w]_k := \text{sgn}([T]_{j+1,j}) \cdot [T]_{k,j} / (2[w]_{j+1}s)$ 
8:   end for
9:    $P := I - 2ww^T$ 
10:   $T := P^T T P$ 
11: end for
    
```

Output: T

Poznámka 4.42. V případě nesymetrické matice dostáváme matici v Hessenbergově tvaru. Postup dalšího řešení je pak stejný jako u Givensovy metody.



Obsah

206. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4.3.4. Lanczosova metoda

Stejně jako u předchozí Givensovy a Householderovy metody budeme hledat takovou matici ortogonální matici Q , která v podobnostní transformaci $T = Q^T A Q$ převede symetrickou matici A na třídiagonální matici.



Obsah

207. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Předpokládejme nejdříve, že matici Q máme k dispozici a označme

$$T := \begin{pmatrix} a_1 & b_1 & 0 & 0 & \dots & 0 \\ b_1 & a_2 & b_2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & \dots & 0 & 0 & b_{n-1} & a_n \end{pmatrix}.$$

Jelikož

$$T = Q^T A Q \quad \Rightarrow \quad Q T = A Q, \quad (4.19)$$

tak dosazením

$$[s_1^Q, \dots, s_n^Q] \begin{pmatrix} a_1 & b_1 & 0 & 0 & \dots & 0 \\ b_1 & a_2 & b_2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & \dots & 0 & 0 & b_{n-1} & a_n \end{pmatrix} = A[s_1^Q, \dots, s_n^Q]$$

a porovnáním k -tých řádků získáme rovnice

$$b_{k-1}s_{k-1}^Q + a_k s_k^Q + b_k s_{k+1}^Q = A s_k^Q, \quad k = 1, \dots, n-1, \quad (4.20)$$

přičemž $b_0 = 0, s_0^Q = 0$.



Obsah

208. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Koeficienty a_k, b_k můžeme určit následujícím způsobem.

- Přenásobíme rovnici (4.20) skalárně vektorem s_k^Q , získáme

$$b_{k-1}(s_{k-1}^Q, s_k^Q) + a_k(s_k^Q, s_k^Q) + b_k(s_{k+1}^Q, s_k^Q) = (As_k^Q, s_k^Q)$$

a díky ortogonalitě matice Q (ortonormalitě vektorů $s_1^Q, \dots, s_n^Q$) se předpis zjednoduší na tvar

$$a_k = (As_k^Q, s_k^Q). \quad (4.21)$$

- Přenásobíme rovnici (4.20) skalárně vektorem s_{k+1}^Q , získáme

$$b_{k-1}(s_{k-1}^Q, s_{k+1}^Q) + a_k(s_k^Q, s_{k+1}^Q) + b_k(s_{k+1}^Q, s_{k+1}^Q) = (As_{k+1}^Q, s_{k+1}^Q)$$

a díky ortonormalitě vektorů $s_1^Q, \dots, s_n^Q$ se předpis zjednoduší na tvar

$$b_k = (As_{k+1}^Q, s_{k+1}^Q). \quad (4.22)$$

Otázkou zůstává, jak co nejjednodušeji vytvořit ortogonální matici Q .

Uvažujme rovnici (4.20) a upravme ji na tvar

$$s_{k+1}^Q = \frac{1}{b_k}((A - a_k I)s_k^Q - b_{k-1}s_{k-1}^Q). \quad (4.23)$$

Tento předpis lze považovat za jakýsi ortonormalizační proces, zvláště pokud označíme

$$r_k := (A - a_k I)s_k^Q - b_{k-1}s_{k-1}^Q \quad (4.24)$$

a volíme

$$b_k = \|r_k\|. \quad (4.25)$$



Obsah

209. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pak dosazením do rovnice (4.23) získáme

$$s_{k+1}^Q = \frac{r_k}{\|r_k\|}. \quad (4.26)$$

Navíc lze dokázat, že volba (4.25) vyhovuje rovnici (4.22) a zkonstruovaná matice Q je ortogonální.



Obsah

210. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Celý postup můžeme shrnout do následujícího algoritmu.

Algoritmus 17**LANCZOSOVA METODA**

Input: A

```
1:  $n :=$  velikost  $A$ 
2:  $T :=$  nulová matice typu  $n \times n$ 
3:  $s_0^Q :=$  nulový vektor dimenze  $n$ 
4:  $r_1 :=$  libovolný jednotkový vektor dimenze  $n$ 
5:  $b_1 := 1$ 
6: for  $j := 1, \dots, n$  do
7:    $s_j^Q := r_j / b_j$ 
8:    $a_j := (s_j^Q)^T A s_j^Q$ 
9:    $r_{j+1} := (A - a_j I) s_j^Q - b_j s_{j-1}^Q$ 
10:   $b_{j+1} := \|r_{j+1}\|$ 
11:   $[T]_{j,j} := a_j$ 
12:  if  $j < n$  then
13:     $[T]_{j+1,j} := b_j$ 
14:     $[T]_{j,j+1} := b_j$ 
15:  end if
16: end for
```

Output: T

[Obsah](#)[211. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka / Okno](#)

4.4. Test

Následující test zahájíte kliknutím na tlačítko „Začátek testu“. U otázek, kde lze volit jen jednu odpověď (test nedovolí zaškrtnout více odpovědí), je správná odpověď bodována počtem bodů uvedených v závorce u zadání a špatná odpověď je bodována 0 body. U otázek, kde lze volit více správných odpovědí, je součet bodů správných odpovědí uveden v závorce u zadání a za každou špatnou odpověď bude odečten jeden bod.

1. (4b.) Vlastní čísla reálné čtvercové symetrické matice jsou
reálná, reálná kladná,
reálná nezáporná, komplexně sdružená,
komplexní.
2. (4b.) Krylovova metoda
hledá vlastní čísla matice,
slouží k určení koeficientů charakteristického polynomu matice,
hledá dominantní vlastní číslo,
hledá největší vlastní číslo.

[Obsah](#)

212. strana ze 415

[Zavřít dokument](#)[Celá obrazovka / Okno](#)

3. (4b.) Dominantní vlastní číslo reálné symetrické pozitivně definitní matice je rovno odmocnině ze spektrálního poloměru matice,
- maximové normě matice,
- libovolnému kořenu charakteristického polynomu,
- nule,
- nejmenšímu vlastnímu číslu.
- největšímu vlastnímu číslu.

4. (4b.) Charakteristický polynom třídiagonální matice

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & -1 \\ 0 & 0 & 3 \end{bmatrix}$$

má tvar

$$\lambda^3 - 5\lambda^2 + 7\lambda - 3,$$

$$\lambda^3 - 5\lambda^2 + 3\lambda + 9,$$

$$\lambda^3 + 5\lambda^2 - 7\lambda + 3,$$

$$\lambda^3 + 5\lambda^2 - 3\lambda - 9.$$



Obsah

213. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

5. (4b.) Mocninná metoda

dokáže v jednom cyklu nalézt všechna vlastní čísla matice,

slouží k určení koeficientů charakteristického polynomu matice,

slouží k určení dominantního vlastního čísla,

vždy určí největší vlastní číslo.

6. (4b.) Nechť $A \in \mathbb{R}^{n \times n}$ je libovolná reálná čtvercová symetrická pozitivně definitní matice a $v \in \mathbb{R}^n$ je libovolný nenulový vektor. Pak podíl

$$R = \frac{v^T A v}{v^T v}$$

se nazývá Rayleighův a platí pro něj

$$0 < R < \kappa(A),$$

$$1/\lambda_{\max} \leq R \leq 1/\lambda_{\min},$$

$$\lambda_{\min} \leq R \leq \lambda_{\max},$$

$$0 < R < 1/\kappa(A).$$



Obsah

214. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

7. (4b.) Pro ortogonální matici $Q \in \mathbb{R}^{n \times n}$ platí

$$Q^T = Q^{-1},$$

řádky matice Q tvoří ortonormální množinu vektorů,

sloupce matice Q tvoří ortonormální množinu vektorů,

řádky matice Q tvoří množinu lineárně nezávislých vektorů.

8. (4b.) QR algoritmus

slouží na určení všech vlastních čísel symetrické matice,

slouží k určení koeficientů charakteristického polynomu matice,

slouží k určení dominantního vlastního čísla,

slouží k určení největšího vlastního čísla symetrické pozitivně definitní matice.



Obsah

215. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

9. (4b.) Metody Givsových rotací a Householderova zrcadlení

hledají vlastní čísla matice,

počítají koeficienty charakteristického polynomu matice,

převádí symetrické matice na třídiagonální,

hledají pouze dominantní vlastní číslo.

10. (4b.) Nechť je dána matice

$$A := \begin{pmatrix} 1 & 2 & 0 \\ 2 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Pomocí Geršgorinovy věty určete nejmenší horní odhad spektra matice A , tj. nejmenší takové číslo δ , že pro všechna vlastní čísla λ_i matice A lze díky této větě tvrdit

$$\lambda_i \leq \delta, \quad i = 1, 2, 3 .$$

Vaše odpověď:



Obsah

216. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

11. (4b.) Podobné matice (vyberte všechna pravdivá tvrzení)

- mají stejná vlastní čísla,
- mají stejnou velikost,
- mají stejné vlastní vektory,
- mají stejný součet diagonálních prvků,
- mají stejný součin diagonálních prvků.

12. (4b.) QR rozklad matice lze určit pomocí

- posunu a redukce spektra,
- Gram-Schmitova ortonormalizačního procesu,
- Householderových zrcadlení,
- Geršgorinovy věty.
- algoritmu QR,
- Givensových rotací,
- podobnostních transformací,



Obsah

217. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

13. (4b.) Lanczosova metoda

hledá vlastní čísla matice,
počítá koeficienty charakteristického polynomu matice,
převádí symetrickou matici na třídiagonální,
hledá pouze dominantní vlastní číslo matice.

14. (3b.) Které z následujících matic jsou ortogonální?

matice Householderova zrcadlení,
matice Givensových rotací,
matice R z QR rozkladu regulární matice,
matice Q z QR rozkladu regulární matice,
libovolná symetrická třídiagonální matice.



Obsah

218. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

15. (4b.) Charakteristický polynom čtvercové reálné matice

je vždy nulový,

je vždy roven nule,

je stejného stupně jako je velikost matice,

má kořeny rovny vlastním číslům matice,

je spojitá funkce na celém $\mathbb{R}$,

je nezáporná funkce,

má počet kořenů roven počtu vlastních čísel,

je funkce definovaná pouze na množině vlastních čísel matice.



Obsah

219. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

16. (4b.) Nechť λ je vlastní číslo matice A a v odpovídající vlastní vektor.

Pak matice

$$A - qI, \quad q \in \mathbb{R}$$

má vlastní číslo $\lambda + q$ a stejný vlastní vektor v ,

má vlastní číslo $\lambda - q$ a stejný vlastní vektor v ,

má vlastní číslo q a stejný vlastní vektor v ,

má vlastní číslo $q - \lambda$ a stejný vlastní vektor v .

Správně zodpovězené otázky:

Získané body:

Procento úspěšnosti:

Pro ukončení testu je třeba kliknout na tlačítko „Konec testu“. Opravení testu se provede kliknutím na tlačítko „Výsledky“, správné odpovědi budou označeny zeleně a červeně budou označeny odpovědi chybné. Při použití otázky s tvořenou odpovědí se správná odpověď zobrazí v rámečku umístěném vpravo dole v navigačním panelu po kliknutí na tlačítko „Odpověď“.



Obsah

220. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Kapitola 5

Interpolace

Průvodce studiem

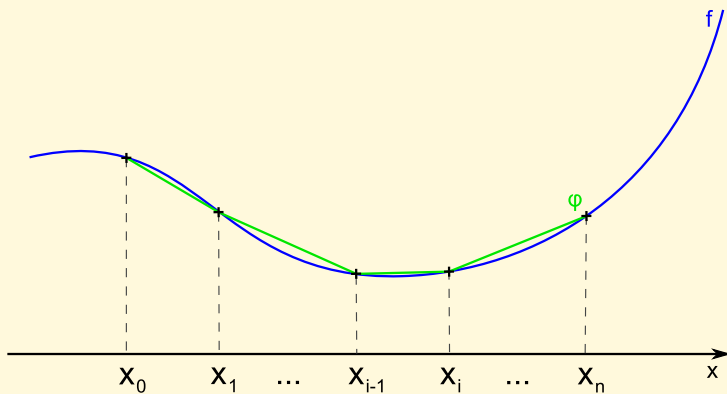
Zřejmě nejznámější a nejčastěji používanou úlohou matematické analýzy je interpolace. V podstatě každý z nás se setkal s úlohou, kdy jsme museli odhadnout hodnotu funkce v bodě a přitom jsme znali pouze její hodnoty v krajních bodech. Tuto úlohu běžně řešíme při pohledu na ručičku ukazatele množství paliva v nádrži a snažíme se odhadnout kolik kilometrů ještě ujedeme. V této kapitole se tedy seznámíme se základními interpolačními metodami a ukážeme si, jak odhadnout chybu, kterou se při interpolaci dopouštíme.

[Obsah](#)[221. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka / Okno](#)

Definice 5.1. Necht je dána funkce $f : \mathbb{R} \rightarrow \mathbb{R}$. Funkce $\varphi : \mathbb{R} \rightarrow \mathbb{R}$, která v bodech $x_0, \dots, x_n \in D_f$ splňuje

$$\forall i = 0, \dots, n : \varphi(x_i) = f(x_i) \quad (5.1)$$

se nazývá interpolační funkce v uzlech $x_0, \dots, x_n$ a podmínky nazýváme *interpolační podmínky*.



Obr. 5.1: Úloha interpolace

Důvod, proč nahrazujeme funkci f funkcí φ může být různý. Většinou požadujeme od hledané funkce φ lepší vlastnosti, než má původní funkce f - spojitost, diferencovatelnost, snazší vyčíslitelnost apod.

Jednou z nejžádanějších vlastností je, aby interpolační funkce byla prvkem vektorového prostoru konečné dimenze a tudíž, aby bylo možné ji vyjadřovat pomocí souřadnic. Takovou třídou funkcí je prostor všech polynomů stupně nejvýše n .

[Obsah](#)[223.](#) strana ze 415[Zavřít dokument](#)[Celá obrazovka / Okno](#)

5.1. Interpolace polynomem

Polynomem n -tého stupně p_n budeme rozumět funkci $p_n : \mathbb{R} \rightarrow \mathbb{R}$ s předpisem

$$p_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = \sum_{i=0}^n a_ix^i$$

kde $a_0, a_1, \dots, a_n \in \mathbb{R}$ jsou konstantní koeficienty.

Za interpolační funkci budeme v dalším uvažovat právě polynom stupně n . Uvažujme hledanou interpolační funkci jako polynom stupně n .

Budeme dále uvažovat sít $n + 1$ bodů

$$x_0 < x_1 < \cdots < x_n ,$$

ve kterých je zadána funkční hodnota

$$f(x_0), f(x_1), \dots, f(x_n) .$$

Po interpolačním polynomu p_n požadujeme splnění interpolačních podmínek

$$\forall i = 0, \dots, n : p_n(x_i) = f(x_i) . \quad (5.2)$$

Jelikož předchozí interpolační podmínky reprezentují $n + 1$ algebraických rovnic a v polynomu je právě $n + 1$ neznámých koeficientů a_i , lze k nalezení těchto koeficientů přistupovat jako k řešení soustavy $n + 1$ lineárních rovnic o $n + 1$ neznámých.



Obsah

224. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Konkrétně

$$\begin{array}{rclcl}
 p_n(x_1) & = & f(x_1) & \Rightarrow & a_0 + a_1x_1 + a_2x_1^2 + \dots a_nx_1^n & = & f(x_1) \\
 p_n(x_2) & = & f(x_2) & \Rightarrow & a_0 + a_1x_2 + a_2x_2^2 + \dots a_nx_2^n & = & f(x_2) \\
 \vdots & & & & & & \vdots \\
 p_n(x_n) & = & f(x_n) & \Rightarrow & a_0 + a_1x_n + a_2x_n^2 + \dots a_nx_n^n & = & f(x_n)
 \end{array}$$

maticově pak

$$Aa = b, \quad \text{kde } A_{i,j} = x_i^j, b_i = f(x_i), \\
 i = 0, \dots, n, \quad j = 0, \dots, n.$$



Obsah

225. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Lemma 5.2. *Nechť jsou zadány interpolační uzly*

$$\begin{aligned} x_0, x_1, \dots, x_n &\in \langle a, b \rangle \\ x_0 < x_1 < \dots < x_n \end{aligned}$$

a příslušné funkční hodnoty $f(x_0), f(x_1), \dots, f(x_n) \in \mathbb{R}$.

Pak existuje právě jeden interpolační polynom P_n stupně n takový, že

$$\forall i = 0, \dots, n : P_n(x_i) = f(x_i) .$$

Algoritmus 18

 INTERPOLACE POLYNOMEM

Input: $x_0, \dots, x_n, f(x_0), \dots, f(x_n)$

```

1: for  $i := 0, \dots, n$  do
2:   for  $j := 0, \dots, n$  do
3:      $[A]_{i,j} = x_i^j$ 
4:   end for
5:    $[b]_i := f(x_i)$ 
6: end for
7:  $[a_0, \dots, a_n]^T := a$  je řešení soustavy  $Aa = b$ 

```

Output: $a_0, \dots, a_n$



Obsah

226. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 5.3. Sestavte interpolační polynom 4-tého stupně, pokud je zadáno

$$\begin{aligned}x_0 &= 0, & f(x_0) &= 1, \\x_1 &= 2, & f(x_1) &= 2, \\x_2 &= 3, & f(x_2) &= -1, \\x_3 &= 4, & f(x_3) &= 3, \\x_4 &= 5, & f(x_4) &= 2.\end{aligned}$$

Řešení. Budeme hledat polynom ve tvaru

$$P_4(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$$

a uvažujme 5 požadovaných interpolačních vlastností

$$\begin{aligned}P_4(x_0) &= f(x_0) &\Leftrightarrow & a_0 + a_1 \cdot 0 + a_2 \cdot 0^2 + a_3 \cdot 0^3 + a_4 \cdot 0^4 = 1 \\P_4(x_1) &= f(x_1) &\Leftrightarrow & a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + a_3 \cdot 2^3 + a_4 \cdot 2^4 = 2 \\P_4(x_2) &= f(x_2) &\Leftrightarrow & a_0 + a_1 \cdot 3 + a_2 \cdot 3^2 + a_3 \cdot 3^3 + a_4 \cdot 3^4 = -1 \\P_4(x_3) &= f(x_3) &\Leftrightarrow & a_0 + a_1 \cdot 4 + a_2 \cdot 4^2 + a_3 \cdot 4^3 + a_4 \cdot 4^4 = 3 \\P_4(x_4) &= f(x_4) &\Leftrightarrow & a_0 + a_1 \cdot 5 + a_2 \cdot 5^2 + a_3 \cdot 5^3 + a_4 \cdot 5^4 = 2\end{aligned}$$

tedy pro nalezení koeficientů budeme řešit soustavu lineárních rovnic

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \\ 1 & 5 & 25 & 125 & 625 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \\ 3 \\ 2 \end{bmatrix}.$$



Obsah

227. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Řešením této soustavy je

$$[a_0, a_1, a_2, a_3, a_4]^T = \left[-\frac{19}{30}, \frac{103}{15}, -\frac{352}{15}, \frac{751}{30}, 1\right]^T,$$

tedy algebraický interpolační polynom má předpis

$$P_4(x) = -\frac{19}{30}x^4 + \frac{103}{15}x^3 - \frac{352}{15}x^2 + \frac{751}{30}x + 1.$$



Obsah

228. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 5.4. (O nešikovné volbě báze)

Matice v předchozím příkladě se nazývá *Vandermondova matice*, která má obecný tvar

$$V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_m & \alpha_m^2 & \dots & \alpha_m^{n-1} \end{bmatrix}, \quad \text{nebo po složkách } [V]_{i,j} = \alpha_i^{j-1},$$

kde $\alpha_1, \dots, \alpha_m$ je posloupnost reálných konstant. O této matici je známo, že je velmi špatně podmíněná (viz [23]).

Volme například posloupnost $\alpha_i = i, i = 1, \dots, n$ (tj. interpolace na uzlech $x_i = \alpha_i$) a podívejme se na číslo podmíněnosti čtvercové Vandermondova matice ($m = n$) pro jednotlivá n .

n	cond(V)
1	1
2	6,85
3	$7,09 \cdot 10^1$
4	$1,17 \cdot 10^3$
5	$2,62 \cdot 10^4$
6	$7,31 \cdot 10^5$
7	$2,45 \cdot 10^7$
8	$9,52 \cdot 10^8$
9	$4,23 \cdot 10^{10}$
10	$2,11 \cdot 10^{12}$



Obsah

229. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Jednodušeji řečeno - pokud bychom počítali interpolaci na uzlech

$$x_1 = 1, x_2 = 2, \dots, x_{10} = 10,$$

pak největší prvek v matici soustavy pro výpočet požadovaných koeficientů by byl $[A]_{10,10} = 10^9$. Při výpočtu s tak velkým číslem nelze zaručit požadovanou přesnost. Navíc v praxi se vyskytují úlohy s interpolací na mnohem vyšším počtu uzlů.

Existuje mnohem rafinovanější pohled na daný problém: úloha hledání interpolačního polynomu je úlohou hledání prvku vektorového prostoru všech polynomů stupně nejvýše n , který je jednoznačně určen požadovanými interpolačními vlastnostmi (viz Věta 5.2 o jednoznačnosti). Jelikož množina polynomů $\{1, x, x^2, \dots, x^n\}$ tvoří bázi vektorového prostoru, ze kterého pochází hledaný prvek, pak způsob hledání koeficientů $a_0, a_1, \dots, a_n$ je de-facto stejný jako hledání souřadnic požadovaného polynomu vzhledem k této bázi.

Volba báze, kterou jsme použili v předchozím příkladě, však není právě ideální, neboť vede na soustavu s Vandermondovou maticí.

Očividně však existují mnohem šikovnější volby báze. Pak soustava pro hledání koeficientů je snadněji řešitelná, jak se ukáže v dalším výkladu u Lagrangeova a Newtonova interpolačního polynomu.



Obsah

230. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

5.2. Lagrangeův interpolační polynom

Opět uvažujme hledanou interpolační funkci jako polynom stupně n a značme ji $L_n(x)$. Dále uvažujme síť $n+1$ interpolačních uzlů $x_0, \dots, x_n$ a množinu funkčních hodnot v těchto uzlech $f(x_0), \dots, f(x_n)$, přičemž

$$x_0 < x_1 < \dots < x_n .$$

Funkci L_n můžeme vyjádřit ve tvaru

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

kde funkce l_i jsou polynomy stupně n , které tvoří bázi prostoru všech polynomů stupně n . Tuto bázi budeme šikovně volit tak, aby každý bázev polynom byl roven 1 pouze v jednom uzlu interpolace a v ostatních by byl roven 0. Takovou volbu báze lze snadno nalézt a popisuje ji následující definice.



Obsah

231. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Definice 5.5. Polynom $L_n(x) = \sum_{i=0}^n f(x_i)l_i(x)$ nazýváme *Lagrangeův interpolační polynom* jestliže l_i jsou polynomy n -tého stupně, kde

$$\begin{aligned}l_i(x_k) &= 0 \quad \text{pro } k \neq i \\l_i(x_i) &= 1\end{aligned}$$

a jsou zadány předpisem

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



Obsah

232. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Příklad 5.6. Určete hodnotu Lagrangeova interpolačního polynomu v bodě $L_n(1)$ pokud jsou zadány hodnoty

$$\begin{aligned} L_n(0) &= 1, & L_n(4) &= 3, \\ L_n(2) &= 2, & L_n(5) &= 2. \\ L_n(3) &= -1, \end{aligned}$$

Řešení. Nejdříve označme

$$\begin{aligned} x_0 &= 0, & x_3 &= 4, \\ x_1 &= 2, & x_4 &= 5. \\ x_2 &= 3, \end{aligned}$$

a jelikož hledaná funkce L_n je zadána v pěti bodech, budeme uvažovat polynom čtvrtého stupně ($n = 4$) s předpisem

$$L_4(x) = \sum_{i=0}^4 f(x_i)l_i(x) = 1.l_0(x) + 2.l_1(x) + (-1).l_2(x) + 3.l_3(x) + 2.l_4(x)$$

kde jednotlivé funkce $l_i(x)$ mají předpis

$$\begin{aligned} l_0(x) &= \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)} = \frac{(x-2)(x-3)(x-4)(x-5)}{(0-2)(0-3)(0-4)(0-5)} \\ &= \frac{1}{120}x^4 - \frac{7}{60}x^3 + \frac{71}{120}x^2 - \frac{77}{60}x + 1 \\ l_1(x) &= \frac{(x-x_0)(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)(x_1-x_4)} = \frac{(x-0)(x-3)(x-4)(x-5)}{(2-0)(2-3)(2-4)(2-5)} \\ &= -\frac{1}{12}x^4 + x^3 - \frac{47}{12}x^2 + 5x \end{aligned}$$



Obsah

233. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

$$\begin{aligned}
 l_2(x) &= \frac{(x-x_0)(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)(x_2-x_4)} = \frac{(x-0)(x-2)(x-4)(x-5)}{(3-0)(3-2)(3-4)(3-5)} \\
 &= \frac{1}{6}x^4 - \frac{11}{6}x^3 + \frac{19}{3}x^2 - \frac{20}{3}x
 \end{aligned}$$

$$\begin{aligned}
 l_3(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)(x_3-x_4)} = \frac{(x-0)(x-2)(x-3)(x-5)}{(4-0)(4-2)(4-3)(4-5)} \\
 &= -\frac{1}{8}x^4 + \frac{5}{4}x^3 - \frac{31}{8}x^2 + \frac{15}{4}x
 \end{aligned}$$

$$\begin{aligned}
 l_4(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_0)(x_4-x_1)(x_4-x_2)(x_4-x_3)} = \frac{(x-0)(x-2)(x-3)(x-4)}{(5-0)(5-2)(5-3)(5-4)} \\
 &= \frac{1}{30}x^4 - \frac{3}{10}x^3 + \frac{13}{15}x^2 - \frac{4}{5}x .
 \end{aligned}$$

Dosazením a úpravou získáme polynom čtvrtého stupně

$$L_4(x) = -\frac{19}{30}x^4 + \frac{103}{15}x^3 - \frac{352}{15}x^2 + \frac{751}{30}x + 1 .$$

Tento polynom je vykreslen na Obrázku 5.2.



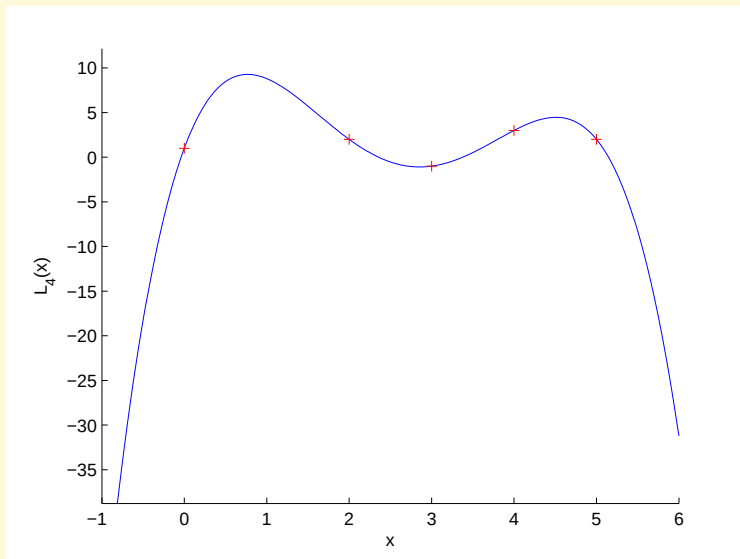
Obsah

234. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 5.2: Příklad interpolace Lagrangeovým polynomem

Zbývá určit hodnotu interpolačního polynomu v bodě 1

$$L_4(1) = -\frac{19}{30} + \frac{103}{15} - \frac{352}{15} + \frac{751}{30} + 1 = \frac{44}{5} .$$



Poznámka 5.7. Množina vektorů $\{l_0(x), l_1(x), \dots, l_n(x)\}$ tvoří bázi vektorového prostoru všech polynomů nejvýše n -tého stupně a vektor $[f(x_0), \dots, f(x_n)]$ je vektor souřadnic hledaného polynomu vzhledem k této bázi. Narozdíl od algebraického interpolačního polynomu tedy není nutno řešit soustavu lineárních rovnic, stačí *pouze* zkonstruovat potřebnou bázi (tj. $l_i(x), i = 0, \dots, n$) a dosadit do předpisu dle Definice 5.5.

Lemma 5.8. *Nechť $\langle a, b \rangle$ je libovolný interval, který obsahuje body $x_0, \dots, x_n$. Nechť $f, f', \dots, f^{(n)} \in C(\langle a, b \rangle)$ a $\forall x \in (a, b)$ existuje $f^{(n+1)}(x)$. Pak pro chybu Lagrangeova interpolačního polynomu L_n platí*

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

Algoritmus 19

LAGRANGEŮV INTERPOLAČNÍ POLYNOM

Input: $x_0, \dots, x_n, f(x_0), \dots, f(x_n), x$

```

1:  $L_n(x) := 0$ 
2: for  $i := 1, \dots, n$  do
3:    $l_i := \prod_{j=0, j \neq i}^n (x - x_j) / \prod_{j=0, j \neq i}^n (x_i - x_j)$ 
4:    $L_n(x) := L_n(x) + l_i f(x_i)$ 
5: end for
```

Output: $L_n(x)$



Obsah

236. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Nyní předpokládejme, že máme možnost zvolit si uzly interpolace. Zvolme je tak, aby chyba interpolace byla co nejmenší.

Existuje-li konstanta $M \in \mathbb{R}^+$ taková, že $\forall x \in (a, b) : |f^{(n+1)}(x)| \leq M$, pak pro chybu interpolačního polynomu platí

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

kde jsme použili označení

$$e_n(x) := (x - x_0) \dots (x - x_n) .$$

To znamená, že vhodnou volbou sítě $x_0, \dots, x_n$ lze zlepšit interpolační funkci. Nejvhodnější volbou jsou kořeny tzv. *Čebyševových polynomů*.



Obsah

237. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Lemma 5.9. *Nechť jsou dány uzly interpolace $x_0, x_1, \dots, x_n \in \langle -1, 1 \rangle$ jako kořeny Čebyševova polynomu, tj.*

$$T_n(x_i) := \cos((n+1) \arccos x_i) = 0, \quad \text{pro } i = 0, \dots, n.$$

Pak pro odhad chyby interpolace polynomem n -tého stupně P_n platí

$$\forall \bar{x} \in \langle -1, 1 \rangle : |f(\bar{x}) - P_n(\bar{x})| \leq \frac{M_{n+1}}{(n+1)!2^n},$$

kde $M \in \mathbb{R}^+$ je konstanta taková, že

$$\forall x \in (-1, 1) : |f^{(n+1)}(x)| \leq M.$$

Poznámka 5.10. Při interpolaci pomocí Čebyševových polynomů na obecném intervalu $\langle a, b \rangle$ uijeme lineární transformaci

$$x_k = \frac{1}{2}((b-a)z_k + b+a), \quad k = 0, 1, \dots, n,$$

kde $z_k \in \langle -1, 1 \rangle$ jsou kořeny Čebyševova polynomu $(n+1)$ -ního stupně. Pro odhad chyby interpolace pak bude platit

$$\forall \bar{x} \in \langle a, b \rangle : |f(\bar{x}) - P_n(\bar{x})| \leq \frac{M_{n+1}}{(n+1)!} \frac{(b-a)^{n+1}}{2^{2n+1}}.$$



Obsah

238. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

5.3. Newtonův interpolační polynom

Volba Lagrangeova tvaru interpolačního polynomu se jeví jako velmi výhodná z hlediska výpočetní stability. V některých případech však není ideální. Typicky se jedná o případy, kdy uzly interpolace jsou časové okamžiky, které v čase přibývají. V takovém případě je nutné pokaždé, když přibude nový časový údaj znovu přepočítat všechny báze funkce. Tento nedostatek lze odstranit využitím tzv. *Newtonova tvaru* interpolačního polynomu.

Uvažujme nyní interpolační polynom tvaru

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

Opět budeme vyžadovat splnění interpolačních podmínek

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

kde $x_0, x_1, \dots, x_n \in \langle a, b \rangle$ jsou uzly interpolace a $f(x_0), f(x_1), \dots, f(x_n)$ jsou zadané funkční hodnoty v těchto uzlech.



Obsah

239. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Neznámé koeficienty $a_0, \dots, a_n$ získáme řešením následující soustavy

$$\begin{aligned} N_n(x_0) &= a_0 & &= f(x_0) \\ N_n(x_1) &= a_0 + a_1(x_1 - x_0) & &= f(x_1) \\ &\vdots & & \\ N_n(x_n) &= a_0 + a_1(x_n - x_0) + \dots + a_n(x_n - x_0) \dots (x_n - x_{n-1}) & &= f(x_n) \end{aligned}$$

Množina vektorů $\{(x - x_0)(x - x_1) \dots (x - x_i), i = 0, \dots, n - 1\}$ tvoří bázi vektorového prostoru všech polynomů nejvýše n -tého stupně. Z předešlých úvah je vidět, že pro nalezení souřadnic hledaného polynomu v této bázi bude nutno vyřešit dolní trojúhelníkovou matici. Tuto soustavu však není nutné řešit pokaždé, když rozšíříme uzly interpolace o nové, které jsou větší než ty dosavadní.

V takovém případě stačí stávající soustavu rozšířit o nové rovnice a dopřednou substitucí dopočítat nové koeficienty. Původní rovnice nadále zůstávají v platnosti.

Historicky se však problém řešil i bez sestavování soustavy pomocí tzv. *poměrných diferencí*.



Obsah

240. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Definice 5.11. Poměrnými diferencemi 0. řádu nazýváme (v bodě x_i) hodnoty

$$f[x_i] = f(x_i), \quad i = 0, \dots, n$$

Poměrnými diferencemi 1. řádu nazýváme

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}, \quad i = 0, \dots, n-1$$

Poměrnými diferencemi k . řádu nazýváme

$$f[x_i, x_{i+1}, \dots, x_{i+k+1}] = \frac{f[x_{i+1}, \dots, x_{i+k+1}] - f[x_i, \dots, x_{i+k}]}{x_{i+k+1} - x_i}, \quad i = 0, \dots, n-k-1$$

Poznámka 5.12. Poměrné difference lze zapsat do tabulky

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

Tato tabulka dává návod, jak jednoduše rekurentně konstruovat poměrné difference bez nutnosti odvozování komplikovaných vzorců. Vypočtených poměrných diferencí lze pak výhodně využít pro sestavení *Newtonova interpolačního polynomu*.

Obsah

241. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Definice 5.13. Polynom

$$N_n(x) := f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, \dots, x_n](x - x_0) \cdots (x - x_{n-1}), \quad (5.3)$$

kde

$$x_0 < x_1 < \cdots < x_n \quad (5.4)$$

jsou uzly interpolace, nazýváme *Newtonův interpolační polynom s poměrnými diferencemi*.

Věta 5.14. *Newtonův interpolační polynom s poměrnými diferencemi splňuje podmínky interpolace (5.2).*



Obsah

242. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Algoritmus 20

NEWTONŮV INTERPOLAČNÍ POLYNOM

Input: $x_0, \dots, x_n, f(x_0), \dots, f(x_n), x$

```
1: for  $i := 0, \dots, n$  do
2:    $sum_1 := 0$ 
3:    $sum_2 := 1$ 
4:   for  $j := 1, \dots, i - 1$  do
5:      $sum_1 := sum_1 + f[x_0, \dots, x_j].sum_2$ 
6:      $sum_2 := sum_2 \cdot (x_i - x_j)$ 
7:   end for
8:    $f[x_0, \dots, x_i] := (f(x_i) - sum_1) / sum_2$ 
9: end for

10:  $sum_1 := 0$ 
11:  $sum_2 := 1$ 
12: for  $j := 1, \dots, n$  do
13:    $sum_1 := sum_1 + f[x_0, \dots, x_j].sum_2$ 
14:    $sum_2 := sum_2 \cdot (x - x_j)$ 
15: end for
16:  $N_n(x) := sum_1$ 
```

Output: $N_n(x)$ 

Obsah

243. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 5.15. Necht jsou dány uzly interpolace

$$\begin{aligned}x_0 &= 0, & f(x_0) &= 1, \\x_1 &= 2, & f(x_1) &= 2, \\x_2 &= 3, & f(x_2) &= -1, \\x_3 &= 4, & f(x_3) &= 3, \\x_4 &= 5, & f(x_4) &= 2.\end{aligned}$$

Sestavte Newtonův interpolační polynom s poměrnými diferencemi a určete hodnotu tohoto polynomu v bodě $\bar{x} = 1$.

Řešení. Nejdříve sestavíme tabulku poměrných diferencí pro polynom 5-tého stupně

$$\begin{array}{c|c|c|c|c|c|c}x_0 & f[x_0] & & & & & \\x_1 & f[x_1] & f[x_0, x_1] & & & & \\x_2 & f[x_2] & f[x_1, x_2] & f[x_0, x_1, x_2] & & & \\x_3 & f[x_3] & f[x_2, x_3] & f[x_1, x_2, x_3] & f[x_0, x_1, x_2, x_3] & & \\x_4 & f[x_4] & f[x_3, x_4] & f[x_2, x_3, x_4] & f[x_1, x_2, x_3, x_4] & f[x_0, x_1, x_2, x_3, x_4]\end{array}$$

kde jednotlivé poměrné difference vypočítáme užitím tabulky pro výpočet poměrných diferencí.

$$\begin{array}{c|c|c|c|c|c|c}x_0 = 0 & 1 & & & & & \\x_1 = 2 & 2 & 1/2 & & & & \\x_2 = 3 & -1 & -3 & -7/6 & & & \\x_3 = 4 & 3 & 4 & 7/2 & 7/6 & & \\x_4 = 5 & 2 & -1 & -5/2 & -2 & -19/30 & \end{array}$$



Obsah

244. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pak dosazením do předpisu Newtonova interpolačního polynomu s poměrnými diferencemi a po úpravě získáme polynom

$$N_4(x) = -\frac{19}{30}x^4 + \frac{103}{15}x^3 - \frac{352}{15}x^2 + \frac{751}{30}x + 1 .$$

Zbývá určit hodnotu interpolačního polynomu v bodě 1

$$N_4(1) = -\frac{19}{30} + \frac{103}{15} - \frac{352}{15} + \frac{751}{30} + 1 = \frac{44}{5} .$$



Obsah

245. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

5.3.1. Ekvidistantní uzly interpolace

Uvažujme nyní ekvidistantní síť interpolačních uzlů, tj. existuje konstanta $h \in \mathbb{R}^+$ taková, že

$$\forall i = 0, \dots, n-1 : x_{i+1} - x_i = h, \quad x_i = x_0 + hi.$$

Tato volba sítě nám umožňuje výpočet poměrných diferencí významně zjednodušit. Místo poměrných diferencí totiž můžeme k sestavení Newtonova interpolačního polynomu využít tzv. *dopředných diferencí*.



Obsah

246. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Definice 5.16. Obyčejnou dopřednou diferencí k. řádu rozumíme

$$\Delta^k f = \sum_{j=0}^n (-1)^{n-j} \binom{n}{j} f(x_j)$$

Dopřednou diferencí 0. řádu rozumíme

$$\Delta^0 f_i = f(x_i)$$

Dopřednou diferencí 1. řádu rozumíme

$$\Delta^1 f_i = \Delta^0 f_{i+1} - \Delta^0 f_i = f(x_{i+1}) - f(x_i)$$

Dopřednou diferencí k. řádu rozumíme

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i$$

Poznámka 5.17. Očividně

$$\Delta^k f_i = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} f(x_{i+j}) .$$



Obsah

247. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Poznámka 5.18. Tabulka dopředných diferencí

$$\begin{array}{l|l}
 x_0 & f_0 = \Delta^0 f_0 \quad \rightarrow \quad \Delta^1 f_0 \quad \rightarrow \quad \Delta^2 f_0 \quad \cdots \rightarrow \quad \Delta^n f_0 \\
 x_1 & f_1 = \Delta^0 f_1 \quad \nearrow \quad \quad \quad \nearrow \\
 \vdots & \vdots \\
 x_{n-1} & f_{n-1} = \Delta^0 f_{n-1} \quad \rightarrow \quad \Delta^1 f_{n-1} \quad \nearrow \\
 x_n & f_n = \Delta^0 f_n \quad \nearrow
 \end{array}$$

Lemma 5.19. Na ekvidistantní síti uzlů $x_0, x_1, \dots, x_n$ platí

$$f[x_0, \dots, x_n] = \frac{\Delta^n f_0}{n!h^n}$$

Věta 5.20. Necht $x_i = x_0 + hi, i = 0, \dots, n$

$x = x_0 + ht, t \in \mathbb{R}_0^+$ a necht

$$\binom{t}{j} := \frac{t!}{j!(t-j)!} = \frac{t(t-1)\dots(t-j+1)}{j!}, \binom{t}{0} = 1 \quad \text{pro } t, j \in \mathbb{R}_0^+, t \geq j.$$

Pak polynom

$$N_n^+(t) = f_0 + \binom{t}{1} \Delta f_0 + \cdots + \binom{t}{n} \Delta^n f_0 \quad (5.5)$$

je Newtonův interpolační polynom s dopřednými diferencemi.



Obsah

248. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 5.21. Interpolujte funkci f pomocí Newtonova interpolačního polynomu s dopřednými diferencemi na zadané ekvidistanční síti uzlů a příslušných funkčních hodnotách

$$\begin{aligned}x_0 &= 0, & f(x_0) &= 1 \\x_1 &= 1, & f(x_1) &= 3 \\x_2 &= 2, & f(x_2) &= 2 \\x_3 &= 3, & f(x_3) &= 3 \\x_4 &= 4, & f(x_4) &= 4.\end{aligned}$$

Řešení. Nejdříve vyplníme tabulku dopředných diferencí dle předpisů v Definici 5.16.

$x_i = i$	$\Delta^0 f_i$	$\Delta^1 f_i$	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0	$f_0 = 1$	$2 - 1 = 1$	$-3 - 1 = -4$	$7 - (-4) = 11$	$-12 - 11 = -23$
1	$f_1 = 2$	$-1 - 2 = -3$	$4 - (-3) = 7$	$-5 - 7 = -12$	
2	$f_2 = -1$	$3 - (-1) = 4$	$-1 - 4 = -5$		
3	$f_3 = 3$	$2 - 3 = -1$			
4	$f_4 = 2$				

Následně stačí dosadit do předpisu Newtonova interpolačního polynomu s dopřednými diferencemi (5.5).

$$\begin{aligned}N_4^+(x) &= f_0 + \binom{x}{1} \Delta^1 f_0 + \binom{x}{2} \Delta^2 f_0 + \binom{x}{3} \Delta^3 f_0 + \binom{x}{4} \Delta^4 f_0 \\&= 1 + \binom{x}{1} \cdot 1 + \binom{x}{2} \cdot (-4) + \binom{x}{3} \cdot 11 + \binom{x}{4} \cdot (-24) \\&= 1 + \frac{x}{1!} - \frac{4x(x-1)}{2!} + \frac{11x(x-1)(x-2)}{3!} - \frac{24x(x-1)(x-2)(x-3)}{4!} \\&\quad \dots \\&= -\frac{23}{24}t^4 + \frac{91}{12}t^3 - \frac{433}{24}t^2 + \frac{149}{12}t + 1\end{aligned}$$



Obsah

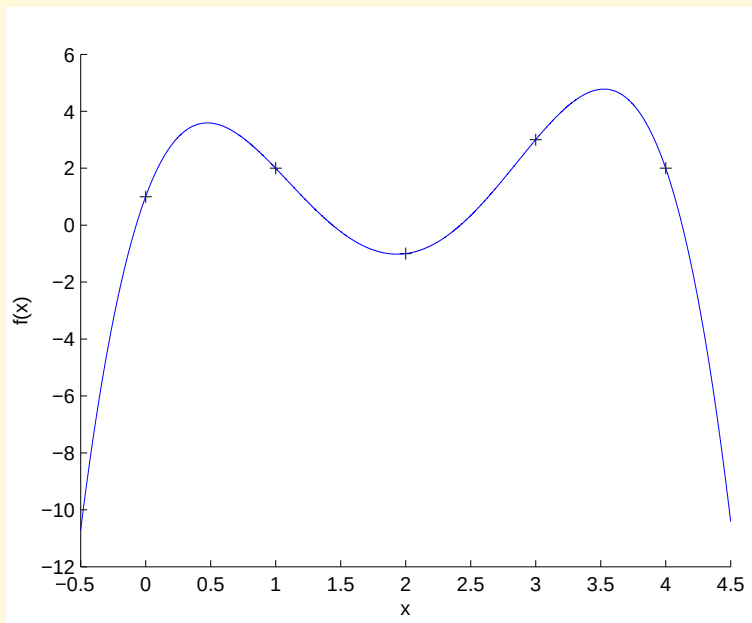
249. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Výsledný interpolační polynom je vykreslen na obrázku 5.3. ▲



Obr. 5.3: Příklad 5.21 - Newtonův interpolační polynom s dopřednými diferencemi

Pro zájemce

Obdobně lze také definovat *zpětné* a *centrální* difference.

- *Zpětné difference*

$$\begin{aligned}\nabla^0 f_i &= f_i \\ \nabla^1 f_i &= \nabla^0 f_i - \nabla^0 f_{i-1} = f_i - f_{i-1} \\ \nabla^k f_i &= \nabla^{k-1} f_i - \nabla^{k-1} f_{i-1}\end{aligned}$$

V takovém případě má Newtonův interpolační polynom tvar

$$N_n^-(t) = f_n - \binom{-t}{1} \Delta f_n + \cdots + (-1)^n \binom{-t}{n} \Delta^n f_n.$$

- *Centrální difference*

$$\begin{aligned}\delta^0 f_i &= f_i \\ \delta^1 f_i &= \delta^0 f_{i+\frac{1}{2}} - \delta^0 f_{i-\frac{1}{2}} = f(x_0 + (i + \frac{1}{2})h) - f(x_0 + (i - \frac{1}{2})h) \\ \delta^k f_i &= \delta^{k-1} f_{i+\frac{1}{2}} - \delta^{k-1} f_{i-\frac{1}{2}} = \delta^{k-1} f(x_0 + (i + \frac{1}{2})h) - \delta^{k-1} f(x_0 + (i - \frac{1}{2})h)\end{aligned}$$

Z těchto diferencí je taktéž možné sestavit Newtonův interpolační polynom.



Obsah

251. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

5.4. Trigonometrická interpolace

Uvažujme interpolační funkci T_n ve tvaru

$$T_n(x) \stackrel{\text{def}}{=} \frac{1}{2}A_0 + \sum_{i=1}^n (A_i \cos ix + B_i \sin ix), \quad i = 0, \dots, 2n.$$

Opět budeme předpokládat splnění interpolačních podmínek

$$T_n(x_i) = f(x_i)$$

na ekvidistanční síti interpolačních uzlů

$$x_i = \frac{2\pi i}{2n+1}, \quad i = 0, \dots, 2n \quad (5.6)$$

Koeficienty interpolační funkce T_n lze získat na základě následující věty.



Obsah

252. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Věta 5.22. *Polynom*

$$T_n(x) \stackrel{\text{def}}{=} \frac{1}{2}A_0 + \sum_{i=1}^n (A_i \cos ix + B_i \sin ix) \quad (5.7)$$

kde

$$A_i := \frac{2}{2n+1} \sum_{j=0}^{2n} f(x_j) \cos ix_j, i = 0, \dots, n \quad (5.8)$$

$$B_i := \frac{2}{2n+1} \sum_{j=0}^{2n} f(x_j) \sin ix_j, i = 1, \dots, n \quad (5.9)$$

je interpolační polynom na síti interpolačních uzlů

$$x_0, x_1, \dots, x_n$$

$$x_{j+1} = x_j + h, j = 0, \dots, n-1, h \in \mathbb{R}^+ \text{ const.}$$



Obsah

253. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Algoritmus 21

TRIGONOMETRICKÁ INTERPOLACE

Input: f, n, x

```
1: for  $i := 0, \dots, n$  do  
2:    $x_i = (2\pi i)/(2n + 1)$   
3:   if  $i > 0$  then  
4:      $A_i := \frac{2}{2n+1} \sum_{j=0}^{2n} f(x_j) \cos ix_j$   
5:      $B_i := \frac{2}{2n+1} \sum_{j=0}^{2n} f(x_j) \sin ix_j$   
6:   else  
7:      $A_0 := \frac{2}{2n+1} \sum_{j=0}^{2n} f(x_j)$   
8:   end if  
9: end for  
10:  $T_n(x) := \frac{1}{2}A_0 + \sum_{i=1}^n (A_i \cos ix + B_i \sin ix)$ 
```

Output: $T_n(x)$ 

Obsah

254. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Příklad 5.23. Uvažujme funkci

$$f(x) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{pro } x \in \langle 0, \pi/2 \rangle \\ 2 & \text{pro } x \in \langle \pi/2, \pi \rangle \\ 3 & \text{pro } x \in \langle \pi, 3/2\pi \rangle \\ 4 & \text{pro } x \in \langle 3/2\pi, 2\pi \rangle \end{cases}$$

a její periodické rozšíření $f^P(x)$.

Interpolujte tuto nespojitou a 2π -periodickou funkci na síti bodů (5.6). Volte různý počet uzlů a porovnejte interpolační funkci s původní funkcí.

Řešení. Volme například počet uzlů $2n + 1 \stackrel{\text{def}}{=} 5$, pak $n = 2$. Budeme interpolovat na síti bodů (viz (5.6))

$$\begin{array}{ll} x_0 = 0, & f(x_0) = 1 \\ x_1 = \frac{2}{5}\pi, & f(x_1) = 1 \\ x_2 = \frac{4}{5}\pi, & f(x_2) = 2 \\ x_3 = \frac{6}{5}\pi, & f(x_3) = 3 \\ x_4 = \frac{8}{5}\pi, & f(x_4) = 4. \end{array}$$



Obsah

255. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Nejdříve spočteme potřebné koeficienty dle (5.8) a (5.9).

$$A_0 = \frac{2}{5} \sum_{j=0}^4 f(x_j) = \frac{22}{5}$$

$$A_1 = \frac{2}{5} \sum_{j=0}^4 f(x_j) \cos x_j = \frac{2}{5} \left(1 + 1 \cdot \cos \frac{2\pi}{5} + 2 \cdot \cos \frac{4\pi}{5} + 3 \cdot \cos \frac{6\pi}{5} + 4 \cdot \cos \frac{8\pi}{5} \right)$$

$$A_2 = \frac{2}{5} \sum_{j=0}^4 f(x_j) \cos 2x_j = \frac{2}{5} \left(1 + 1 \cdot \cos \frac{4\pi}{5} + 2 \cdot \cos \frac{8\pi}{5} + 3 \cdot \cos \frac{12\pi}{5} + 4 \cdot \cos \frac{16\pi}{5} \right)$$

$$B_1 = \frac{2}{5} \sum_{j=0}^4 f(x_j) \sin x_j = \frac{2}{5} \left(1 \cdot \sin \frac{2\pi}{5} + 2 \cdot \sin \frac{4\pi}{5} + 3 \cdot \sin \frac{6\pi}{5} + 4 \cdot \sin \frac{8\pi}{5} \right)$$

$$B_2 = \frac{2}{5} \sum_{j=0}^4 f(x_j) \sin 2x_j = \frac{2}{5} \left(1 \cdot \sin \frac{4\pi}{5} + 2 \cdot \sin \frac{8\pi}{5} + 3 \cdot \sin \frac{12\pi}{5} + 4 \cdot \sin \frac{16\pi}{5} \right) ,$$

které následně dosadíme do předpisu interpolačního polynomu (5.7)

$$T_2(x) = \frac{11}{5} + A_1 \cos x + B_1 \sin x + A_2 \cos 2x + B_2 \sin 2x .$$

Jedna perioda výsledné funkce je vykreslena na Obrázku 5.4.

Při použití většího počtu interpolačních uzlů získáme přesnější řešení, viz Obrázek 5.5 a 5.6.



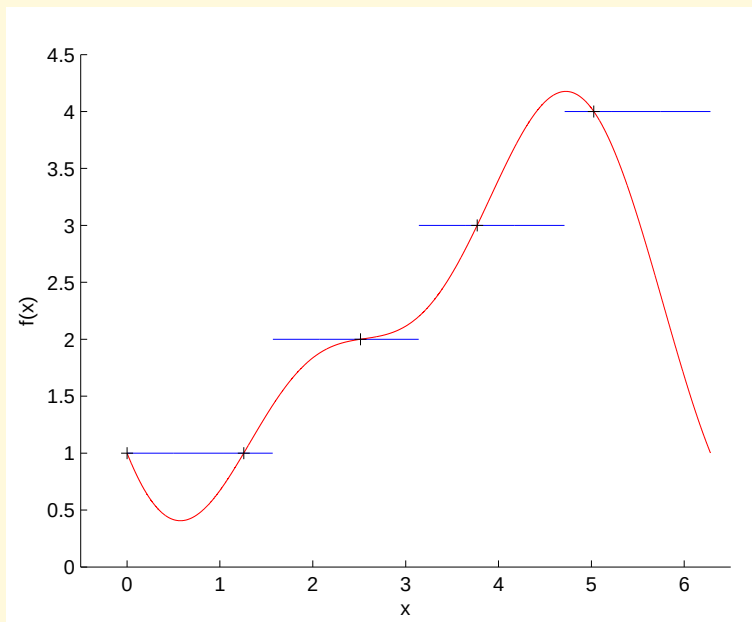
Obsah

256. strana ze 415

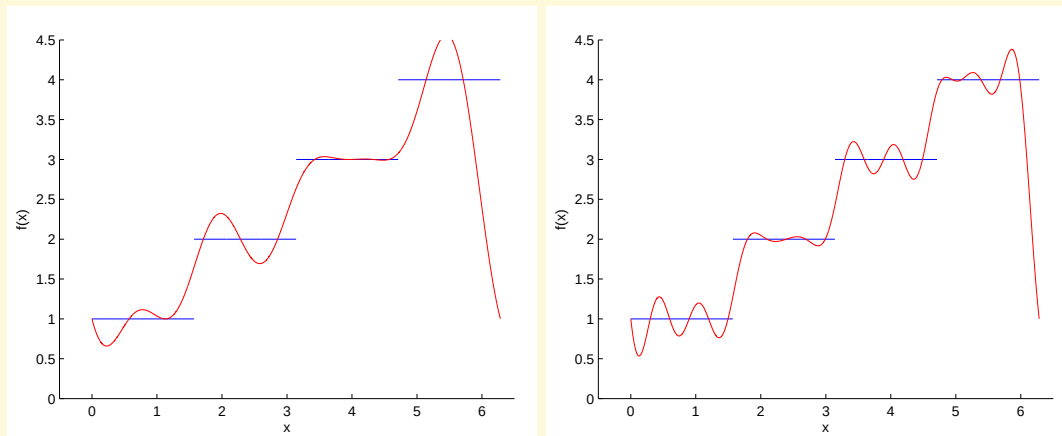


Zavřít dokument

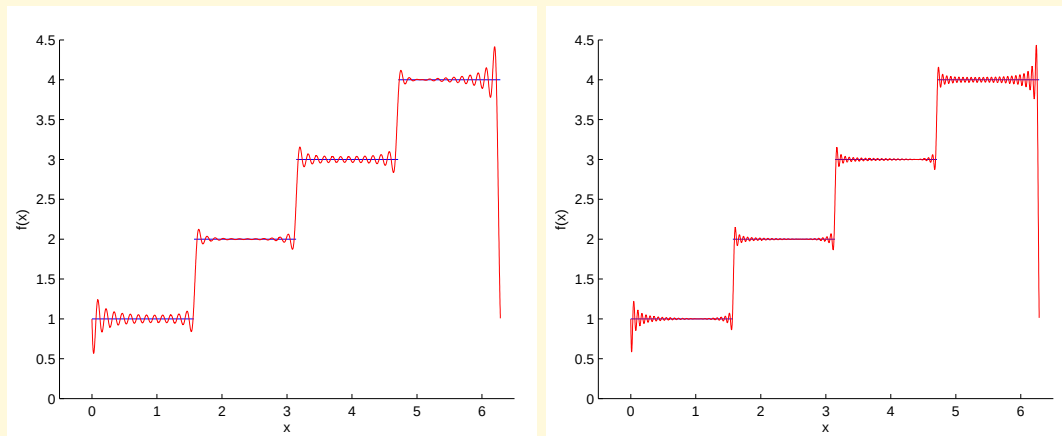
Celá obrazovka / Okno



Obr. 5.4: Příklad 5.23 - trigonometrická interpolace s volbou $n = 2$



Obr. 5.5: Příklad 5.23 - trigonometrická interpolace s volbou $n = 5, 10$

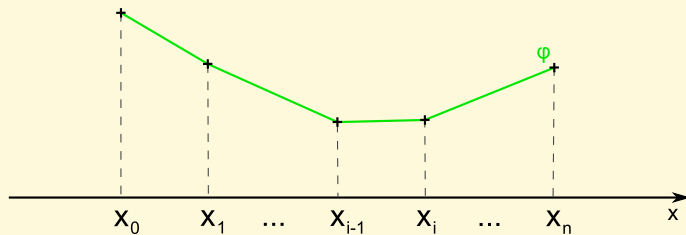


Obr. 5.6: Příklad 5.23 - trigonometrická interpolace s volbou $n = 50, 100$

5.5. Interpolace lineárními spline funkcemi

Použití algebraických či trigonometrických interpolačních polynomů je velice omezené. V případě velkého počtu interpolačních uzlů totiž začne velmi narůstat řád interpolačního polynomu a polynom samotný se velmi „rozkmitá“.

V takovém případě se jako vhodnější ukazuje použití tzv. *spline funkcí*, které jsou postupně definovány jako polynomy nižších stupňů na intervalech ohraničených uzly interpolace. Nejjednodušším případem spline funkce je tzv. *lineární spline*, kterou můžeme charakterizovat jako spojitou po částech lineární funkci.



Obr. 5.7: Interpolace lineární spline funkcí.



Obsah

260. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Definice 5.24. Funkce $\varphi \in C$ se nazývá *lineární spline funkce*, jestliže na síti interpolačních uzlů $x_0, x_1, \dots, x_n$ splňuje interpolační vlastnost a na jednotlivých disjunktních intervalech

$$\langle x_j, x_{j+1} \rangle, \quad j = 0, \dots, n-1$$

je lineární.

Uvažujme tedy posloupnost interpolačních uzlů

$$a = x_0 < x_1 < \dots < x_n = b,$$

která generuje rozklad intervalu $\langle a, b \rangle$ na $n-1$ podintervalů

$$\langle a, b \rangle = \langle x_0, x_1 \rangle \cup \langle x_1, x_2 \rangle \cup \dots \cup \langle x_{i-1}, x_i \rangle \cup \langle x_i, x_{i+1} \rangle \cup \dots \cup \langle x_{n-1}, x_n \rangle.$$

Označme podinterval $I_i = \langle x_i, x_{i+1} \rangle, i = 0, \dots, n-1$ a definujme na něm lineární funkci

$$\varphi_i(x) = a_i + b_i x, x \in \langle x_i, x_{i+1} \rangle.$$

Neznámé koeficienty $a_i, b_i \in \mathbb{R}$ lze snadno určit z požadované interpolační vlastnosti v krajních intervalech, tj. koeficienty jsou řešením soustavy

$$\begin{aligned} f(x_i) &= a_i + b_i x_i \\ f(x_{i+1}) &= a_i + b_i x_{i+1}. \end{aligned}$$

Po substituci $h_i = x_{i+1} - x_i$ (velikost uvažovaného intervalu) získáme

$$a_i = \frac{1}{h_i} \begin{vmatrix} f(x_i) & x_i \\ f(x_{i+1}) & x_{i+1} \end{vmatrix} \quad b_i = \frac{1}{h_i} \begin{vmatrix} 1 & f(x_i) \\ 1 & f(x_{i+1}) \end{vmatrix}.$$



Obsah

261. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pak část interpolační funkce φ_i na intervalu I_i má tvar

$$\varphi_i(x) = \begin{cases} f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{h_i}(x - x_i) & \text{na } \langle x_i, x_{i+1} \rangle \\ 0 & \text{mimo } \langle x_i, x_{i+1} \rangle . \end{cases} \quad (5.10)$$

Pak interpolační funkci φ lze vyjádřit jako součet

$$\varphi(x) = \sum_{i=0}^{n-1} \varphi_i(x), x \in \langle a, b \rangle .$$

Algoritmus 22

 INTERPOLACE LINEÁRNÍMI SPLINE FUNKCEMI

Input: $x_0, \dots, x_n, f(x_0), \dots, f(x_n)$

```

1: for  $i := 0, \dots, n - 1$  do
2:    $a_i := f(x_i)$ 
3:    $h_i := x_{i+1} - x_i$ 
4:    $b_i := (f(x_{i+1}) - f(x_i))/h_i$ 
5: end for
```

Output: $a_0, \dots, a_n, b_0, \dots, b_n$



Obsah

262. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 5.25. Nechť je dána funkce f body a funkčními hodnotami

$$\begin{array}{ll} x_0 = 1 & f(x_0) = 1 \\ x_1 = 3 & f(x_1) = 4 \\ x_2 = 4 & f(x_2) = 2 \\ x_3 = 5 & f(x_3) = 0 \\ x_4 = 8 & f(x_4) = 3 \\ x_5 = 9 & f(x_5) = 3. \end{array}$$

Na této síti interpolujte funkci f pomocí lineární spline funkce φ .

Řešení. Nejprve spočteme jednotlivé difference

$$\begin{array}{ll} h_0 = x_1 - x_0 = 2 \\ h_1 = x_2 - x_1 = 1 \\ h_2 = x_3 - x_2 = 1 \\ h_3 = x_4 - x_3 = 3 \\ h_4 = x_5 - x_4 = 1 \end{array}$$

a jednotlivé části hledané interpolační funkce φ zkonstruujeme pomocí předpisu (5.10)

$$\begin{array}{ll} \varphi_0 = f(x_0) + \frac{f(x_1) - f(x_0)}{h_0}(x - x_0) = \frac{3}{2}x - \frac{1}{2} \\ \varphi_1 = f(x_1) + \frac{f(x_2) - f(x_1)}{h_1}(x - x_1) = -2x + 10 \\ \varphi_2 = f(x_2) + \frac{f(x_3) - f(x_2)}{h_2}(x - x_2) = -2x + 10 \\ \varphi_3 = f(x_3) + \frac{f(x_4) - f(x_3)}{h_3}(x - x_3) = x - 5 \\ \varphi_4 = f(x_4) + \frac{f(x_5) - f(x_4)}{h_4}(x - x_4) = 3. \end{array}$$



Obsah

263. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pak výsledná interpolační funkce má tvar

$$\varphi(x) = \begin{cases} \frac{3}{2}x - \frac{1}{2} & \text{pro } x \in \langle 1, 3 \rangle \\ -2x + 10 & \text{pro } x \in \langle 3, 5 \rangle \\ x - 5 & \text{pro } x \in \langle 5, 8 \rangle \\ 3 & \text{pro } x \in \langle 8, 9 \rangle . \end{cases}$$

Výsledná funkce je vykreslena na Obrázku 5.8.



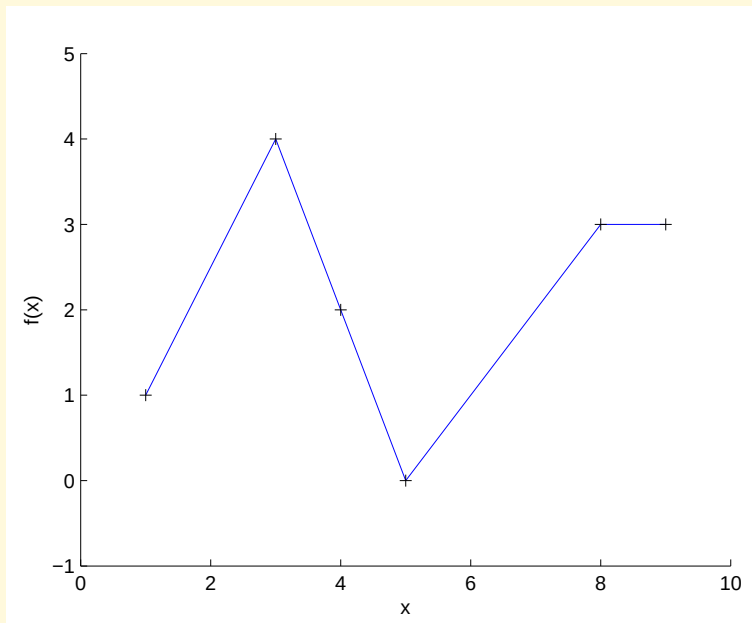
Obsah

264. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 5.8: Příklad 5.25 - interpolace lineární spline funkcí.



5.6. Interpolace kubickými spline funkcemi

Lineární spline představený v předchozí kapitole sice řeší problém zvyšujícího se stupně polynomu se zvyšujícím se počtem uzlů interpolace, avšak výsledná interpolační funkce díky nediferencovatelnosti v uzlech interpolace není hladká, ačkoliv bychom to v řadě případů očekávali. V takovém případě je možné využít spline funkcí vyšších řádů, tj. takových, které jsou počástech kvadratické, kubické, kvintické atp. V takovém případě se dá zajistit i určitá hladkost výsledné interpolační funkce.

Jedním s nejčastěji používaných splinů je *kubický spline*. V následujícím textu si tento spline odvodíme.

Definice 5.26. Necht jsou dány dvojice $(x_i, f_i), i = 0, \dots, n$ a necht $f(x_i) = f_i$. Funkci $\varphi(x)$ nazýváme *kubickou spline funkcí* jestliže

1. $\varphi \in C^2(\langle a, b \rangle)$
2. φ je kubický polynom na $\langle x_i, x_{i+1} \rangle, \forall i = 0, \dots, n-1$
3. $\varphi(x_i) = f_i, i = 0, \dots, n$



Obsah

266. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 5.27. Z definice je patrné, že kubická spline funkce φ není určena jednoznačně. Abychom jednoznačně určili interpolační funkci a její části na jednotlivých intervalech

$$\begin{aligned}\varphi(x) &:= \sum_{i=0}^{n-1} \varphi_i(x) \\ \varphi_i(x) &:= a_i + b_i x + c_i x^2 + d_i x^3 \quad \text{na } \langle x_i, x_{i+1} \rangle\end{aligned}$$

je nezbytné pro jednotlivé intervaly určit $4n$ neznámých a_i, b_i, c_i, d_i , $i = 0, 1, \dots, n-1$. Z interpolačních podmínek

$$\varphi(x_i) = f_i, i = 0, \dots, n$$

dostáváme $n+1$ rovnic. Ze spojitosti $\varphi, \varphi', \varphi''$ v bodech interpolace $x_i, i = 1, \dots, n-1$ dostáváme dalších $3(n-1) = 3n-3$ rovnic.

Celkem tedy máme $4n$ neznámých a $4n-2$ rovnic.

Dvě podmínky tedy musíme dodat.

Například

1. $\varphi'(a) = f'_0, \varphi'(b) = f'_n$
2. $\varphi''(a) = f''_0, \varphi''(b) = f''_n$
3. $\varphi''(a) = 0, \varphi''(b) = 0$ (tzv. přirozený kubický spline)

Tyto podmínky nazýváme *okrajové podmínky*.



Obsah

267. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



5.6.1. Konstrukce spline funkce

Na každém intervalu interpolace budeme hledat kubický spline ve tvaru

$$\varphi_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, i = 0, \dots, n - 1 \quad (5.11)$$

φ_i'' je tedy polynom prvního stupně a tudíž φ'' je funkce po částech lineární a spojitá neboť $\varphi \in C^2(\langle a, b \rangle)$.

Označme $\varphi''(x_i) := M_i, i = 0, \dots, n$ jako *i-tý moment spline funkce*.

Můžeme sestavit interpolaci po částech lineárními funkcemi (viz 5.5)

$$\varphi_i''(x) = \varphi_i''(x_i) \frac{x_{i+1} - x}{h_i} + \varphi_i''(x_{i+1}) \frac{x - x_i}{h_i}, x \in \langle x_i, x_{i+1} \rangle, i = 0, \dots, n - 1 .$$

$$\varphi_i''(x) = M_i \frac{x_{i+1} - x}{h_i} + M_{i+1} \frac{x - x_i}{h_i}$$

Integrováním φ_i'' dostáváme

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

a následně

$$\varphi_i(x) = \int \varphi_i'(x) dx = M_i \frac{(x_{i+1} - x)^3}{6h_i} + M_{i+1} \frac{(x - x_i)^3}{6h_i} + A_i(x - x_i) + B_i .$$

Z podmínek interpolace plyne

$$\varphi_i(x_i) = f(x_i), \varphi_i(x_{i+1}) = f(x_{i+1}), i = 0, \dots, n - 1 ,$$

Obsah

268. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

tedy

$$\varphi_i(x_i) = M_i \frac{h_i^2}{6} + B_i = f(x_i)$$

a

$$\varphi_i(x_{i+1}) = M_{i+1} \frac{h_i^2}{6} + A_i h_i + B_i = f(x_{i+1}) .$$

Řešením této soustavy 2 rovnic o 2 neznámých A_i , B_i dostáváme

$$B_i = f(x_i) - M_i \frac{h_i^2}{6}$$

$$A_i = \frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{h_i}{6} (M_{i+1} - M_i) .$$



Obsah

269. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 5.28. Pomocí M_i lze jednoznačně spočítat všechny funkce φ_i , tzn. můžeme určit a_i, b_i, c_i, d_i následujícími předpisy

$$\begin{aligned} a_i &= \varphi_i(x_i) = f(x_i) \\ b_i &= \varphi'_i(x_i) = -M_i \frac{h_i}{2} + A_i = \frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{2M_i + M_{i+1}}{6} h_i \\ c_i &= \frac{1}{2} \varphi''_i(x_i) = \frac{1}{2} M_i \\ d_i &= \frac{1}{6} \varphi'''_i(x_i^+) = \frac{M_{i+1} - M_i}{6h_i} \end{aligned} \quad (5.12)$$

kde $\varphi'''_i(x_i^+)$ je derivace φ''_i zprava v x_i .

Nyní stačí pouze určit momenty $\varphi''(x_i), i = 0, \dots, n$. K tomu využijeme spojitost φ' v $x_i, i = 1, \dots, n-1$, tzn.

$$\varphi'_{i-1}(x_i^-) = \varphi'_i(x_i^+), i = 1, \dots, n-1.$$

Vyjádříme si příslušné jednostrané derivace

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

Dosazením do podmínek spojitosti φ' dostaneme rovnice

$$\frac{h_{i-1}}{6} M_{i-1} + \frac{h_{i-1} + h_i}{3} M_i + \frac{h_i}{6} M_{i+1} = \frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{f(x_i) - f(x_{i-1})}{h_{i-1}}, i = 1, \dots, n-1$$



Obsah

270. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Čtenář si určitě všimnul, že se jedná o $n - 1$ rovnic pro $n + 1$ neznámých. Pro jednoznačnost řešení je proto doplníme libovolnou okrajovou podmínkou.

Označíme-li dále

$$\lambda_i = \frac{h_{i-1}}{h_{i-1} + h_i}, \quad \frac{h_i}{h_{i-1} + h_i} = 1 - \lambda_i = \mu_i, \quad (5.13)$$

$$g_i = \frac{6}{h_{i-1} + h_i} \left(\frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{f(x_i) - f(x_{i-1}))}{h_{i-1}} \right). \quad (5.14)$$

Dostáváme třídiagonální soustavu lineárních rovnic, jejíž řešením jsou hledané momenty kubického splinu.

$$\begin{array}{rcl} 2M_1 & +\mu_1 M_2 & = g_1 - \lambda_1 f_0'' \\ \lambda_2 M_1 & +2M_2 & +\mu_2 M_3 = g_2 \\ & & \vdots \\ \lambda_{n-2} M_{n-3} & +2M_{n-2} & +\mu_{n-1} M_{n-1} = g_{n-2} \\ & \lambda_{n-1} M_{n-2} & +2M_{n-1} = g_{n-1} - \mu_{n-1} f_n'' \end{array} \quad (5.15)$$



Obsah

271. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Lemma 5.29. *Nechť $f \in C^4(\langle a, b \rangle)$, K je konstanta taková, že při dělení $\langle a, b \rangle$ na $h_i, i = 0, 1, \dots, n-1$ platí*

$$\frac{\max h_i}{\min h_i} \leq K .$$

Nechť φ je kubická spline funkce pro funkci f a nechť φ splňuje okrajové podmínky. Pak pro $\forall x \in \langle a, b \rangle$ a $j = 0, 1, 2, 3$ platí

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

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



Obsah

272. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obsah

273. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Algoritmus 23

INTERPOLACE KUBICKÝMI SPLINE FUNKCEMI

Input: $x_0, \dots, x_n, f(x_0), \dots, f(x_n), f''(x_0), f''(x_n)$

```

1: for  $i := 0, \dots, n-1$  do
2:    $h_i := x_{i+1} - x_i$ 
3: end for
4: for  $i := 0, \dots, n-1$  do
5:    $\lambda_i := h_i / (h_i + h_{i+1})$ 
6:    $\mu_i := 1 - \lambda_i$ 
7:    $g_i := 6 / (h_{i-1} + h_i) ((f(x_{i+1}) - f(x_i)) / h_i - (f(x_i) - f(x_{i-1})) / h_{i-1})$ 
8: end for

9:  $\tilde{M}$  je nulová matice typu  $(n-1) \times (n-1)$ 
10:  $\tilde{m}$  je nulový vektor dimenze  $n-1$ 
11:  $[\tilde{M}]_{1,1} := 2, [\tilde{M}]_{1,2} := \mu_1$ 
12:  $[\tilde{m}]_1 := g_1 - \lambda_1 f''(x_0)$ 
13: for  $i := 2, \dots, n-2$  do
14:    $[\tilde{M}]_{i,i-1} := \lambda_i$ 
15:    $[\tilde{M}]_{i,i} := 2$ 
16:    $[\tilde{M}]_{i,i+1} := \mu_i$ 
17: end for
18:  $[\tilde{M}]_{n-1,n-1} := 2, [\tilde{M}]_{n-1,n-2} := \lambda_{n-1}$ 
19:  $[\tilde{m}]_{n-1} := g_{n-1} - \mu_{n-1} f''(x_n)$ 
20:  $[M_1, \dots, M_{n-1}]^T := m$  je řešení soustavy  $\tilde{M}m = \tilde{m}$ 

21: for  $i := 0, \dots, n-1$  do
22:    $a_i := f(x_i)$ 
23:    $b_i := (f(x_{i+1}) - f(x_i)) / h_i - h_i(2M_i + M_{i+1}) / 6$ 
24:    $c_i := (1/2)M_i$ 
25:    $d_i := (M_{i+1} - M_i) / (6h_i)$ 
26: end for

```

Output: $a_0, \dots, a_n, b_0, \dots, b_n, c_0, \dots, c_n, d_0, \dots, d_n$

Příklad 5.30. Interpolujte funkci f pomocí přirozené interpolace kubickými spline funkcemi na zadaných bodech a příslušných funkčních hodnotách

$$\begin{aligned}x_0 &= 1, & f(x_0) &= 1 \\x_1 &= 2, & f(x_1) &= 3 \\x_2 &= 3, & f(x_2) &= 2 \\x_3 &= 4, & f(x_3) &= 3 \\x_4 &= 5, & f(x_4) &= 4.\end{aligned}$$

Řešení. Nejprve je potřeba si všimnout, že zadané uzly jsou ekvidistatní, tj.

$$h_0 = h_1 = h_2 = h_3 = 1.$$

To nám vcelku zjednoduší výpočet. Nejdříve spočteme $\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3$ dle předpisu (5.13)

$$\begin{aligned}\lambda_1 &= \frac{h_0}{h_0+h_1} = \frac{1}{2}, & \mu_1 &= 1 - \lambda_1 = \frac{1}{2} \\ \lambda_2 &= \frac{h_1}{h_1+h_2} = \frac{1}{2}, & \mu_2 &= 1 - \lambda_1 = \frac{1}{2} \\ \lambda_3 &= \frac{h_2}{h_2+h_3} = \frac{1}{2}, & \mu_3 &= 1 - \lambda_1 = \frac{1}{2}\end{aligned}$$

a dosadíme do předpisu pomocných proměnných (5.14)

$$\begin{aligned}g_1 &= \frac{6}{h_0+h_1} \left(\frac{f(x_2)-f(x_1)}{h_1} - \frac{f(x_1)-f(x_0)}{h_0} \right) = -9 \\ g_2 &= \frac{6}{h_1+h_2} \left(\frac{f(x_3)-f(x_2)}{h_2} - \frac{f(x_2)-f(x_1)}{h_1} \right) = 6 \\ g_3 &= \frac{6}{h_2+h_3} \left(\frac{f(x_4)-f(x_3)}{h_3} - \frac{f(x_3)-f(x_2)}{h_2} \right) = 0.\end{aligned}$$



Obsah

274. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Pak lze sestavit soustavu (5.15) (jelikož se jedná o přirozený spline, tak $f_0'' = f_4'' = M_0 = M_4 = 0$)

$$\begin{array}{rclcl} 2M_1 & +\mu_1 M_2 & & = g_1 - \lambda_1 f_0'' & \\ \lambda_2 M_1 & +2M_2 & +\mu_2 M_3 & = g_2 & \\ & \lambda_3 M_2 & +2M_3 & = g_3 - \mu_3 f_4'' & \end{array} \Rightarrow \begin{array}{rclcl} 2M_1 & +\frac{1}{2}M_2 & & = -9 & \\ \frac{1}{2}M_1 & +2M_2 & +\frac{1}{2}M_3 & = 6 & \\ & \frac{1}{2}M_2 & +2M_3 & = 0 & \end{array} .$$

Řešením soustavy jsou hledané momenty

$$M_0 = 0, \quad M_1 = -\frac{159}{28}, \quad M_2 = \frac{33}{7}, \quad M_3 = -\frac{33}{28}, \quad M_4 = 0 ,$$

které lze následně dosadit do předpisů koeficientů pro kubické polynomy (5.12) na jednotlivých intervalech

$$\begin{array}{ll} a_0 = f(x_0) = 1 & a_2 = f(x_2) = 2 \\ b_0 = \frac{f(x_1)-f(x_0)}{h_0} - \frac{2M_0+M_1}{6}h_0 = \frac{165}{56} & b_2 = \frac{f(x_3)-f(x_2)}{h_2} - \frac{2M_2+M_3}{6}h_2 = -\frac{3}{8} \\ c_0 = \frac{1}{2}M_0 = 0 & c_2 = \frac{1}{2}M_2 = \frac{33}{14} \\ d_0 = \frac{M_1-M_0}{6h_0} = -\frac{53}{56} & d_2 = \frac{M_3-M_2}{6h_2} = -\frac{55}{56} \\ \\ a_1 = f(x_1) = 3 & a_3 = f(x_3) = 3 \\ b_1 = \frac{f(x_2)-f(x_1)}{h_1} - \frac{2M_1+M_2}{6}h_1 = \frac{3}{28} & b_3 = \frac{f(x_4)-f(x_3)}{h_3} - \frac{2M_3+M_4}{6}h_3 = \frac{39}{28} \\ c_1 = \frac{1}{2}M_1 = -\frac{159}{56} & c_3 = \frac{1}{2}M_3 = -\frac{33}{56} \\ d_1 = \frac{M_2-M_1}{6h_1} = \frac{97}{56} & d_3 = \frac{M_4-M_3}{6h_3} = \frac{11}{56} . \end{array}$$



Obsah

275. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Tedy výsledná interpolační funkce (5.11) má předpis

$$\varphi(x) \begin{cases} = 1 + \frac{165}{56}(x-1) - \frac{53}{56}(x-1)^3 & \text{pro } x \in \langle 1, 2 \rangle \\ = 3 + \frac{3}{28}(x-2) - \frac{159}{56}(x-2)^2 + \frac{97}{56}(x-2)^3 & \text{pro } x \in (2, 3) \\ = 2 - \frac{3}{8}(x-3) - \frac{33}{14}(x-3)^2 - \frac{55}{56}(x-3)^3 & \text{pro } x \in (3, 4) \\ = 3 + \frac{39}{28}(x-4) - \frac{33}{56}(x-4)^2 + \frac{11}{56}(x-4)^3 & \text{pro } x \in (4, 5) \end{cases}.$$

Tato funkce je vykreslena na Obrázku 5.9.



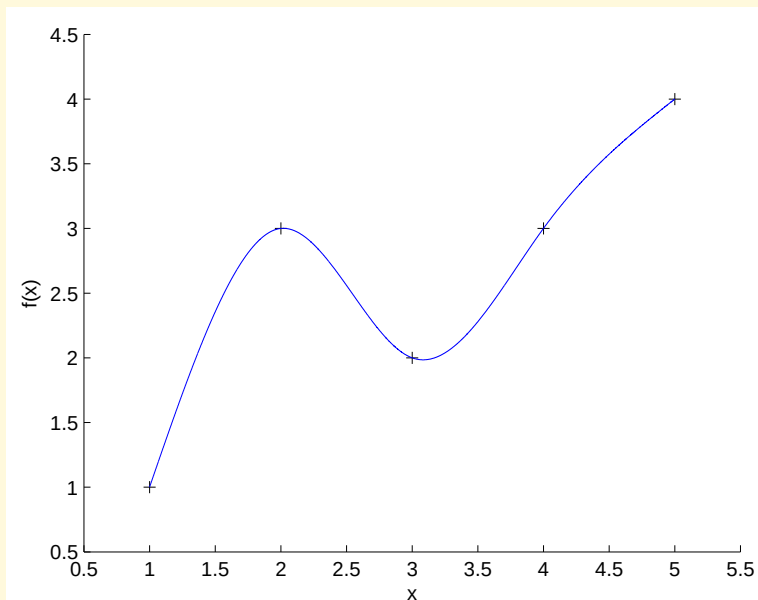
Obsah

276. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 5.9: Příklad 5.30 - interpolace kubickými spline funkcemi.



5.7. Test

Následující test zahájíte kliknutím na tlačítko „Zacátek testu“. U otázek, kde lze volit jen jednu odpověď (test nedovolí zaškrtnout více odpovědí), je správná odpověď bodována počtem bodů uvedených v závorce u zadání a špatná odpověď je bodována 0 body. U otázek, kde lze volit více správných odpovědí, je součet bodů správných odpovědí uveden v závorce u zadání a za každou špatnou odpověď bude odečten jeden bod.

1. (4b.) Pod pojmem *interpolace polynomem* rozumíme

nalezení polynomu, který splňuje interpolační podmínky,

nalezení polynomu nejnižšího stupně, který splňuje alespoň jednu interpolační podmínku,

nalezení polynomu, který nejlépe aproximuje zadanou funkci,

určení okrajových podmínek splajnu,

funkci, která se v každém bodě uvažovaného intervalu rovná interpolované funkci,

nalezení polynomu, jehož integrál na uvažovaném intervalu je nejnižší možný.



Obsah

278. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2. (4b.) Interpolační polynom n -tého stupně je na síti $n + 1$ různých bodů s příslušnými funkčními hodnotami v těchto bodech

je určen nejednoznačně, proto musíme přidat okrajové podmínky,

je určen jednoznačně,

neexistuje,

existuje pouze při splnění určitých podmínek.

3. (4b.) Kubickou spline funkci lze zkonstruovat na (vyberte nejobecnější podmínku)

libovolné síti interpolačních uzlů,

speciální ekvidistantní síti interpolačních uzlů,

síti nezáporných uzlů,

síti uzlů z kořenů Čebyševových polynomů.



Obsah

279. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4. (3b.) Necht jsou zadány dva různé body x_1 a x_2 . Necht jsou zadány také funkční hodnoty interpolované funkce v těchto bodech $f(x_1)$ a $f(x_2)$.

Pak pro úlohu interpolace funkce f na síti bodů x_1, x_2 (vyberte všechna pravdivá tvrzení)

neexistuje interpolační polynom prvního stupně,

existuje právě jeden interpolační polynom prvního stupně,

existuje nekonečně mnoho interpolačních polynomů druhého stupně,

neexistuje interpolační polynom třetího stupně,

existuje právě jeden interpolační polynom třetího stupně.

existuje nekonečně mnoho interpolačních polynomů prvního stupně,

neexistuje interpolační polynom druhého stupně,

existuje právě jeden interpolační polynom druhého stupně,

existuje nekonečně mnoho interpolačních polynomů třetího stupně,



Obsah

280. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

5. (4b.) Pokud z interpolačních podmínek sestavíme za účelem hledání koeficientů interpolačního polynomu soustavu, pak matice této soustavy je špatně podmíněná a nazýváme ji

matice interpolační funkce,

třídiagonální matice,

Grammova matice,

Givensova matice,

Krylovova matice,

Householderova matice,

trigonometrická matice,

Vandermondova matice,

ortogonální matice.



Obsah

281. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

6. (5b.) Lagrangeův interpolační polynom je (vyberte všechny možnosti dokončení tvrzení tak, aby tvrzení bylo pravdivé)

spojitá funkce,

speciální případ lineárního splajnu,

speciální případ kubického splajnu,

spojitě diferencovatelná funkce,

polynom, který splňuje interpolační podmínky,

je polynom nejnižšího stupně ze všech polynomů, které splňují interpolační podmínky,

je roven Newtonovu interpolačnímu polynomu,

je polynom, který ve všech bodech definičního oboru má stejnou funkční hodnotu jako interpolovaná funkce.

je složen s Čebyševových polynomů.



Obsah

282. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

7. (5b.) Newtonův interpolační polynom je (vyberte všechny možnosti dokončení tvrzení tak, aby tvrzení bylo pravdivé)

spojitá funkce,

speciální případ lineárního splajnu,

speciální případ kubického splajnu,

spojitě diferencovatelná funkce,

polynom, který splňuje interpolační podmínky,

je polynom nejnižšího stupně ze všech polynomů, které splňují interpolační podmínky,

je roven Lagrangeovu interpolačnímu polynomu,

je polynom, který ve všech bodech definičního oboru má stejnou funkční hodnotu jako interpolovaná funkce.

je složen s Čebyševových polynomů.



Obsah

283. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8. (4b.) Newtonův interpolační polynom s poměrnými diferencemi konstruujeme
- z tabulky poměrných diferencí,
 - z Čebyševových polynomů,
 - z Krylovových prostorů,
 - speciální volbou báze,
 - speciální volbou interpolačních uzlů,
 - řešením speciální soustavy lineárních rovnic,
 - kubickou spline funkcí,
9. (4b.) Newtonův interpolační polynom s dopřednými diferencemi lze zkonstruovat na (vyberte nejobecnější podmínku)
- libovolné síti interpolačních uzlů,
 - ekvidistantní síti interpolačních uzlů,
 - síti nezáporných uzlů,
 - síti uzlů z kořenů Čebyševových polynomů.



Obsah

284. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

10. (4b.) Trigonometrickou interpolaci lze zkonstruovat na (vyberte nejobecnější podmínku)
- libovolné síti interpolačních uzlů,
 - speciální ekvidistantní síti interpolačních uzlů,
 - síti nezáporných uzlů,
 - síti uzlů z kořenů Čebyševových polynomů.
11. (4b.) U interpolace spline funkcemi s narůstajícím počtem interpolačních uzlů narůstá také
- stupeň interpolačního polynomu,
 - dimenze soustavy pro určení koeficientů interpolačního polynomu,
 - počet použitých poměrných, dopředných či zpětných diferencí,
 - nic z předchozího.



Obsah

285. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

12. (4b.) Necht' je dána síť interpolačních uzlů s funkčními hodnotami

$$\begin{array}{rcl} x_1 & = & 0 \quad f(x_1) = 2 \\ x_2 & = & 1 \quad f(x_2) = 0 \end{array}$$

Pak lineární spline funkce má v tomto případě předpis

$$\varphi(x) = -x^2 + 1,$$

$$\varphi(x) = x + 2,$$

$$\varphi(x) = \frac{1}{2}x + 1,$$

$$\varphi(x) = -\frac{1}{2}x + 2,$$

$$\varphi(x) = -2x + 2,$$

$$\varphi(x) = -x + 1,$$

$$\varphi(x) = -x + 2,$$

neexistuje.



Obsah

286. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

13. (4b.) Necht je dána síť interpolačních uzlů s funkčními hodnotami a okrajové podmínky

$$\begin{array}{ll} x_1 = 0 & f(x_1) = 2 \\ x_2 = 1 & f(x_2) = 0 \\ \varphi'(x_1) = 1 & \varphi'(x_2) = 1 \end{array}$$

Pak kubická spline funkce má v tomto případě předpis

$$\varphi(x) = 0,$$

$$\varphi(x) = x^2 - x,$$

$$\varphi(x) = x^2 + x - 2,$$

$$\varphi(x) = 2x^3 - 3x^2 + x,$$

$$\varphi(x) = 2x^3 - 3x^2 + x + 1,$$

$$\varphi(x) = -2x^3 + 3x^2 + x - 1,$$

$$\varphi(x) = -2x^3 + 3x^2 - x,$$

neexistuje.



Obsah

287. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

14. (4b.) Konstrukce kubické spline funkce na ekvidistantní síti uzlů

je jednoznačná,

není jednoznačná, je nutno přidat hodnotu první derivace v počátečním bodě,

není jednoznačná, je nutno přidat hodnotu druhé derivace v koncovém bodě,

není jednoznačná, je nutno přidat hodnotu první derivace v počátečním bodě,

není jednoznačná, je nutno přidat hodnotu druhé derivace v koncovém bodě,

není jednoznačná, je nutno přidat hodnotu první derivace v libovolném krajním bodě,

není jednoznačná, je nutno přidat hodnotu druhé derivace v libovolném krajním bodě,

není jednoznačná, je nutno přidat libovolnou okrajovou podmínku,

není jednoznačná, je nutno přidat dvě vhodné okrajové podmínky,

neexistuje.



Obsah

288. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

15. (4b.) Lineární spline funkci lze zkonstruovat na (vyberte nejobecnější podmínku)
- libovolné síti interpolačních uzlů,
 - speciální ekvidistantní síti interpolačních uzlů,
 - síti nezáporných uzlů,
 - síti uzlů z kořenů Čebyševových polynomů.

[Obsah](#)[289. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka/Okno](#)

16. (4b.) Poměrné difference k . řádu konstruujeme předpisem

$$f[x_i, x_{i+1}, \dots, x_{i+k+1}] = \frac{f[x_{i+1}, \dots, x_{i+k+1}] - f[x_i, \dots, x_{i+k}]}{x_{i+k+1} - x_i}, i = 0, \dots, n - k - 1$$

$$f[x_0, x_1, \dots, x_{i+k+1}] = \frac{f[x_{i+1}, \dots, x_{i+k+1}] - f[x_i, \dots, x_{i+k}]}{x_{i+k+1} - x_i}, i = 0, \dots, n - k - 1$$

$$f[x_i, x_{i+1}, \dots, x_{i+k+1}] = \frac{f[x_i, \dots, x_{i+k+1}] - f[x_i, \dots, x_{i+k}]}{x_{i+k+1} - x_i}, i = 0, \dots, n - k - 1$$

$$f[x_i, x_{i+1}, \dots, x_{i+k+1}] = \frac{f[x_{i+1}] - f[x_{i+k}]}{x_{i+k} - x_i}, i = 0, \dots, n - k - 1$$



Obsah

290. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

17. (4b.) Obyčejnou dopřednou diferencí k . řádu konstruujeme předpisem

$$\Delta^k f = \sum_{j=0}^n \binom{n}{j} f(x_j)$$

$$\Delta^k f = \Delta^0 f_{k+1} - \Delta^0 f_k$$

$$\Delta^k f = \Delta^0 f_k - \Delta^0 f_{k+1}$$

$$\Delta^k f = \sum_{j=0}^n (-1)^{n-j} \binom{n}{j} f(x_j)$$



Obsah

291. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Správně zodpovězené otázky:

Získané body:

Procento úspěšnosti:

Pro ukončení testu je třeba kliknout na tlačítko „Konec testu“. Opravení testu se provede kliknutím na tlačítko „Výsledky“, správné odpovědi budou označeny zeleně a červeně budou označeny odpovědi chybné. Při použití otázky s tvořenou odpovědí se správná odpověď zobrazí v rámečku umístěném vpravo dole v navigačním panelu po kliknutí na tlačítko „Odpověď“.



Obsah

292. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Kapitola 6

Aproximace

Průvodce studiem

Obdobně jako v případě interpolace se problematika aproximace funkcí zabývá metodami nalezení jednodušších funkcí, které jsou dostatečným přiblížením (aproximací) funkce jiné. V případě aproximačních metod však nebudeme vyžadovat ani splnění interpolačních podmínek, tzn. že výsledná aproximační funkce nemusí dokonce procházet známými funkčními hodnotami funkce aproximované.

V této kapitole se seznámíme s principy teorie aproximací funkcí a se ze základními aproximačními metodami.

Obsah

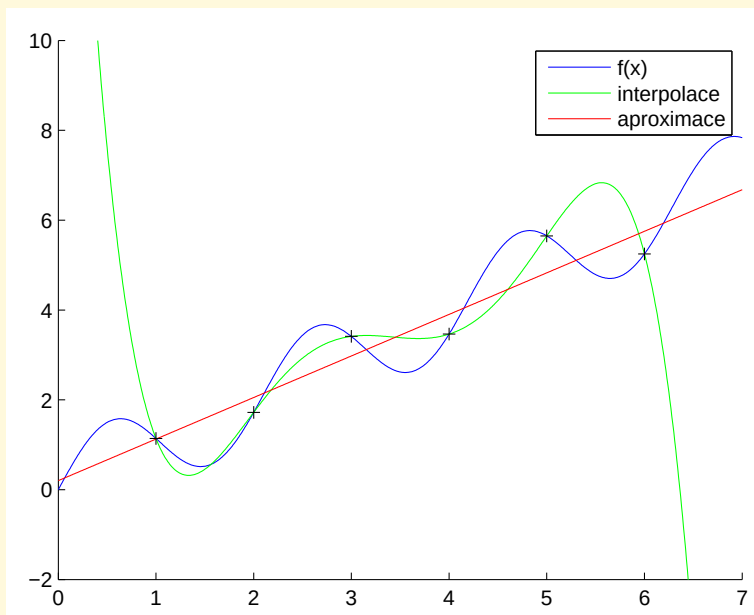
293. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Aproximace si klade za cíl nahradit danou funkcí f z prostoru funkcí $\mathcal{F}$ nějakou jednodušší aproximační funkcí φ z prostoru aproximačních funkcí Φ . Na rozdíl od interpolační funkce však nevyžadujeme, aby se aproximační funkce φ shodovala v předem daných bodech $x_i, i = 0, \dots, m$ s původní funkcí f (viz Obrázek 6.1).



Obr. 6.1: Porovnání interpolačního polynomu s lineární aproximační funkcí.



Obsah

294. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Za podprostor Φ si obvykle vybíráme konečnědimenzionální podprostory funkcí, neboť se výsledné aproximační funkce dají dobře vyjádřit jako lineární kombinace vektorů báze. Typicky jsou to pak prostory polynomů, konečněrozměrné prostory trigonometrických funkcí apod. Je zřejmé, že chyba aproximace bude závislá především na volbě tohoto prostoru.

Důležitou otázkou v aproximační úloze je tedy otázka „míry vzdálenosti“ aproximační funkce φ od původní funkce f . Čím je tato „vzdálenost“ menší, tím zřejmě lepší aproximační vlastnosti získáváme.

Za tímto účelem nám dobře poslouží prostory funkcí se skalárním součinem (f, g) a normou $\|f\| = \sqrt{(f, f)}$ indukovanou tímto skalárním součinem. „Míru vzdálenosti“ f od φ pak můžeme měřit metrikou

$$\rho(f, \varphi) = \|f - \varphi\| = \sqrt{(f - \varphi, f - \varphi)}$$

kde $f \in \mathcal{F}$ a $\varphi \in \Phi$.

Pro hledanou aproximační funkci $\varphi^* \in \Phi$ pak požadujeme, aby splňovala vlastnost

$$\rho(f, \varphi^*) \leq \rho(f, \varphi) \quad \text{pro } \forall \varphi \in \Phi,$$

tj.

$$\|f - \varphi^*\| \leq \|f - \varphi\| \quad \text{pro } \forall \varphi \in \Phi.$$



Obsah

295. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Velmi často se v aproximačních úlohách volí za prostor funkcí $\mathcal{F}$ prostor $L_2(a, b)$ (tj. $\int_a^b f^2(x)dx < +\infty$) a za prostor aproximačních funkcí Φ můžeme například brát prostor všech polynomů stupně nejvýše n .

Na prostoru $L_2(a, b)$ zavádíme skalární součin

$$(f, g)_2 = \int_a^b f(x)g(x)dx$$

a normu

$$\|f\|_2 = \left(\int_a^b f^2(x)dx \right)^{\frac{1}{2}}.$$

Definujeme-li metriku pro vzdálenost funkcí

$$\rho(f, \varphi) \equiv \rho_2(f, \varphi) = \|f - \varphi\|_2,$$

pak aproximační úlohu můžeme předepsat jako minilalizační úlohu

$$\begin{cases} \text{Najdi } \varphi^* \in \Phi \text{ takové, že} \\ \|f - \varphi^*\|_2^2 \leq \|f - \varphi\|_2^2 \quad \text{pro } \forall \varphi \in \Phi. \end{cases} \quad (6.1)$$



Obsah

296. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Všimněme si, že místo normy $\|f - \phi\|_2$ jsme použili v minimalizační úloze (6.1) její kvadrát $\|f - \phi\|_2^2$. Tento tvar nemá žádný vliv na výsledek minimalizační úlohy a přitom je mnohem vhodnější pro úpravy minimalizované funkce neboť zatímco

$$\|f - \phi\|_2 = \sqrt{(f - \phi, f - \phi)_2} ,$$

tak

$$\|f - \phi\|_2^2 = (f - \phi, f - \phi) .$$

Pokud je tedy funkce f zadána pouze na diskrétní síti $\{x_i, i = 0, \dots, m\}$, dostáváme funkci f ve tvaru $m + 1$ rozměrného aritmetického vektoru funkční hodnot

$$(f_0, \dots, f_m) \in \mathbb{R}^m ,$$

kde $f_i \equiv f(x_i), i = 0, \dots, m$.

Množinu funkcí definovaných na konečných množinách bodů pak můžeme ztotožnit s vektorovým prostorem aritmetických vektorů $\mathbb{R}^{m+1}$, na kterém zavedeme skalární součin a normu ve tvaru

$$(f, g)_{2,m} := \sum_{i=0}^m f_i g_i, \quad \|f\|_{2,m} := \left(\sum_{i=0}^m f_i^2 \right)^{\frac{1}{2}} .$$

Původní aproximační úloha má pak tvar

$$\begin{cases} \text{Najdi } \varphi^* \in \mathbb{R}^{m+1} \text{ tak, že} \\ \|f - \varphi^*\|_{2,m}^2 \leq \|f - \varphi\|_{2,m}^2 \quad \text{pro } \forall \varphi \in \mathbb{R}^{m+1} . \end{cases} \quad (6.2)$$

V obou těchto případech (6.1) a (6.2) se metoda aproximace nazývá *metoda nejmenších čtverců*. Její název vychází z geometrické interpretace této metody v diskrétním případě.



Obsah

297. strana ze 415

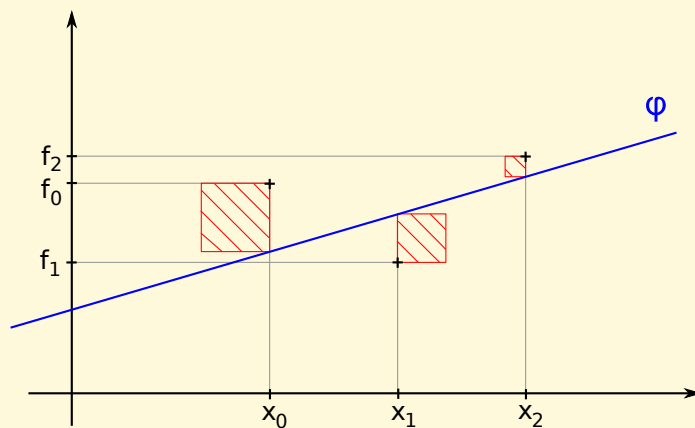


Zavřít dokument

Celá obrazovka / Okno

Za aproximační polynom se vybírá ten polynom, jehož součet čtverců odchylek od hodnot f_i je nejmenší.

Nejlépe lze princip metody pochopit z Obrázku 6.2.



Obr. 6.2: Metoda nejmenších čtverců minimalizuje obsah čtverců

Z výše uvedeného je patrné, že výsledný aproximační polynom závisí na volbě metriky pro vzdálenost funkcí. Změnou normy resp. skalárního součinu tak můžeme dostat jiné aproximační funkce, jejichž aproximační vlastnosti se budou lišit v závislosti na metrice.

Kromě metody nejmenších čtverců se pro aproximace taktéž často používá tzv. *stejněměrná aproximace*, kdy za normu f volíme

$$\|f\|_{\infty} := \max_{x \in \langle a, b \rangle} |f(x)|, \quad \text{kde } f \in C(\langle a, b \rangle).$$

Pokud si za třídu aproximačních funkcí vezmeme opět prostor všech polynomů stupně nejvýše n , pak aproximační úloha má tvar

$$\begin{cases} \text{Najdi } \varphi^* \in \Phi \text{ takové, že} \\ \min_{\phi} \max_{x \in \langle a, b \rangle} |f(x) - \phi(x)|. \end{cases}$$

Pokud funkce f je opět známa pouze na konečné množině bodů, můžeme volit normu

$$\|f\|_{\infty, m} := \max_{i=0, \dots, m} |f_i|$$

a aproximační úloha má tvar

$$\begin{cases} \text{Najdi } \varphi^* \in \Phi \text{ takové, že} \\ \min_{\phi} \max_{i=0, \dots, m} |f(x_i) - \phi(x_i)|. \end{cases}$$

Pokud se zamyslíme nad rozdílem mezi metodou nejmenších čtverců a stejnoměrnou aproximací, tak výsledná „vzdálenost“ aproximační funkce nepřesáhne určitou mez, danou právě normou $\|f - \phi^*\|_{\infty}$, zatímco v případě metody nejmenších čtverců takováto mez není garantována, ale za to funkce ϕ^* se přibližuje k funkci v průměru po celé délce intervalu, na které aproximuje.



Obsah

299. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

6.1. Metoda nejmenších čtverců

V předchozím textu jsme již definovali metodu nejmenších čtverců jako minimalizaci kvadrátu Eukleidovské normy chyby aproximace.

Řekněme, že prostor Φ má uspořádanou bázi $(\varphi_0, \dots, \varphi_n)$.

Pak $\forall \varphi \in \Phi$ platí

$$\varphi = c_0\varphi_0 + \dots + c_n\varphi_n,$$

kde $c_0, \dots, c_n \in \mathbb{R}$ jsou souřadnice funkce φ v dané bázi.

Budeme se tedy snažit nalézt takové $c_0^*, \dots, c_n^* \in \mathbb{R}$, že

$$\|f - c_0^*\varphi_0 - \dots - c_n^*\varphi_n\|_2^2 \leq \|f - c_0\varphi_0 - \dots - c_n\varphi_n\|_2^2$$

pro $\forall c_0, \dots, c_n \in \mathbb{R}$.

Dostáváme tedy minimalizační úlohu

$$\min_{c_0, \dots, c_n} \rho_2(f, c_0, \dots, c_n)^2$$

s $\rho_2(f, c_0, \dots, c_n)^2 = (f - c_0\varphi_0 - \dots - c_n\varphi_n, f - c_0\varphi_0 - \dots - c_n\varphi_n)_2$.

Pro tuto úlohu si můžeme předsat nutnou podmínku existence minima ve tvaru

$$\frac{\partial \rho_2^2(c_0, \dots, c_n)}{\partial c_k} = 0, k = 0, \dots, n$$



Obsah

300. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Tedy

$$\rho_2^2(c_0, \dots, c_n) = (f, f)_2 - 2 \sum_{i=0}^n c_i (f, \varphi_i)_2 + \sum_{i=0}^n \sum_{j=0}^n c_i c_j (\varphi_i, \varphi_j)_2$$

a

$$\frac{\partial \rho_2^2(c_0, \dots, c_n)}{\partial c_k} = -2(f, \varphi_k)_2 + 2 \sum_{i=0}^n c_i (\varphi_i, \varphi_k)_2 = 0.$$

Nutná podmínka má tedy tvar

$$\sum_{i=0}^n c_i (\varphi_i, \varphi_k)_2 = (f, \varphi_k)_2, \quad k = 0, \dots, n$$

což představuje soustavu lineárních rovnic s neznámými $c_0, \dots, c_n$.



Obsah

301. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Tyto rovnice se nazývají *normální rovnice* a jejich maticový zápis vypadá následovně

$$Ac = b$$

$$A = \begin{bmatrix} (\varphi_0, \varphi_0)_2 & (\varphi_0, \varphi_1)_2 & \cdots & (\varphi_0, \varphi_n)_2 \\ (\varphi_1, \varphi_0)_2 & (\varphi_1, \varphi_1)_2 & \cdots & (\varphi_1, \varphi_n)_2 \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_n, \varphi_0)_2 & (\varphi_n, \varphi_1)_2 & \cdots & (\varphi_n, \varphi_n)_2 \end{bmatrix} = [(\varphi_i, \varphi_j)], \quad b = \begin{bmatrix} (f, \varphi_0)_2 \\ (f, \varphi_1)_2 \\ \vdots \\ (f, \varphi_n)_2 \end{bmatrix}.$$

Postačující podmínkou existence minima je pak pozitivní definitnost Hessiánu

$$D^2 \rho_2^2(c_0, \dots, c_n),$$

což je právě matice A .

Tato matice se nazývá *Grammova* a je o ní známo, že je symetrická a pozitivně definitní pokud $\varphi_0, \dots, \varphi_n$ jsou nezávislé, což v našem případě platí (neboť tvoří bázi prostoru Φ).

Aproximační funkce nalezená touto metodou tedy má požadované minimalizační vlastnosti. V diskrétním případě získáváme stejné normální rovnice, ve kterých se pouze vyskytuje jiný skalární součin.

Konkrétně

$$A = [(\varphi_i, \varphi_j)_{2,m}], \quad b = [(f, \varphi_i)_{2,m}],$$

kde

$$(f, g)_{2,m} = \sum_{i=0}^m f_i g_i.$$



Obsah

302. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 6.1. Je nutné poznamenat, že pro velká n je matice A zkonstruovaná výše uvedeným předpisem velmi špatně podmíněná pro běžné polynomiální báze jako je báze $\varphi_i(x) = x^i$.

To nás přivádí k využití ortogonálních systémů v aproximaci nejmenších čtverců (viz další podkapitola 6.2).

Algoritmus 24

METODA NEJMENŠÍCH ČTVERCŮ

Input: $f, (\varphi_0, \dots, \varphi_n)$

```
1: for  $i := 0, \dots, n$  do  
2:   for  $j := 0, \dots, n$  do  
3:      $[A]_{i,j} := (\varphi_i, \varphi_j)$   
4:   end for  
5:    $[b]_i := (f, \varphi_i)$   
6: end for  
7:  $[c_0, \dots, c_n]^T := c$  je řešení soustavy  $Ac = b$ 
```

Output: $c_0, \dots, c_n$



Obsah

303. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 6.2. Necht funkce $f : \mathbb{R} \rightarrow \mathbb{R}$ je zadána v bodech

$$\begin{aligned} f(0) &= 1 \\ f(2) &= 2 \\ f(3) &= -1 \end{aligned}$$

Aproximujte tuto funkci polynomem prvního stupně $p \in \mathcal{P}_1$ metodou nejmenších čtverců.

Řešení. Budeme tedy hledat polynom p ve tvaru

$$p(x) = c_0 + c_1 x.$$

V ideálním případě by hledaná funkce interpolovala dané body, tedy body by ležely na přímce a „míra vzdálenosti“ by byla nulová. To by ovšem znamenalo, že hledané koeficienty řeší soustavu sestavenou pro hledání interpolačního polynomu (viz Příklad 5.3)

$$Ac = b,$$

kde

$$A = \begin{bmatrix} x_0^0 & x_0^1 \\ x_1^0 & x_1^1 \\ x_2^0 & x_2^1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}, \quad c = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}.$$

Samozřejmě, tato soustava nemá řešení (protože body na přímce neleží, řádky nejsou lineárně závislé).

Můžeme však nalézt „co nejlepší“ řešení, tj. takové, aby chyba řešení

$$\text{err}(c) := Ac - b$$



Obsah

304. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

byla co nejmenší.

Ve smyslu nejmenších čtverců budeme uvažovat co nejmenší chybu s kvadrátem, tj.

$$c = \arg \min \|Ac - b\|^2.$$

Nejdříve upravme

$$\|Ac - b\|^2 = (Ac - b, Ac - b) = c^T A^T Ac - 2c^T A^T b + b^T b.$$

Jelikož minimum je nezávislé na konstantě $b^T b$, lze minimalizační úlohu přepsat do tvaru

$$\min \frac{1}{2} c^T A^T Ac - c^T A^T b.$$

Nutná podmínka minima (jedná se o kvadratickou funkci s symetrickou pozitivně definitní maticí, viz Věta 3.24) má tvar

$$A^T Ac = A^T b.$$

Tedy místo původní interpolační soustavy budeme řešit tuto soustavu.

$$A^T A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 5 & 13 \end{bmatrix}, \quad A^T b = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Konkrétním řešením je $c_0 = 1.5$, $c_1 = -0.5$. Porovnání zadaných bodů a aproximační funkce je vykresleno na Obrázku 6.3.



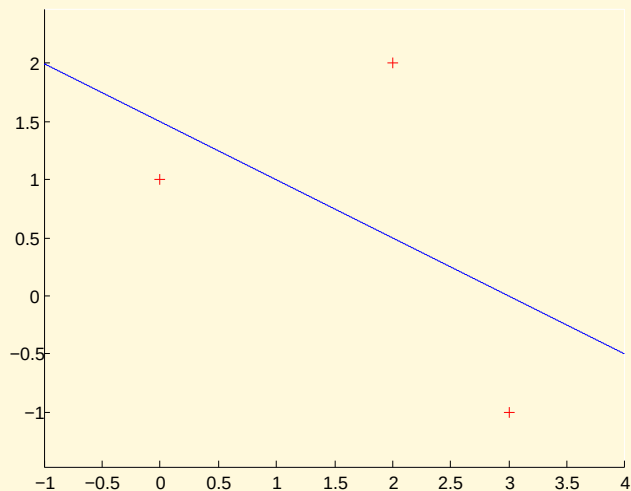
Obsah

305. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 6.3: Příklad 6.2 - řešení, aproximace lineární funkcí.



Obsah

306. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Tento způsob řešení odpovídá výše uvedené teorii. Zvolíme-li standardní bázi vektorového prostoru $\mathcal{P}_1$, tj. (φ_0, φ_1) , kde

$$\begin{aligned}\varphi_0 &= x^0 \\ \varphi_1 &= x^1,\end{aligned}$$

pak odpovídající Grammova matice má tvar

$$\begin{aligned}\begin{bmatrix} (\varphi_0, \varphi_0)_2 & (\varphi_0, \varphi_1)_2 \\ (\varphi_1, \varphi_0)_2 & (\varphi_1, \varphi_1)_2 \end{bmatrix} &= \begin{bmatrix} \sum_{i=0}^2 \varphi_0(x_i) \cdot \varphi_0(x_i) & \sum_{i=0}^2 \varphi_0(x_i) \cdot \varphi_1(x_i) \\ \sum_{i=0}^2 \varphi_1(x_i) \cdot \varphi_0(x_i) & \sum_{i=0}^2 \varphi_1(x_i) \cdot \varphi_1(x_i) \end{bmatrix} = \\ &= \begin{bmatrix} 0^0 \cdot 0^0 + 2^0 \cdot 2^0 + 3^0 \cdot 3^0 & 0^0 \cdot 0^1 + 2^0 \cdot 2^1 + 3^0 \cdot 3^1 \\ 0^1 \cdot 0^0 + 2^1 \cdot 2^0 + 3^1 \cdot 3^0 & 0^1 \cdot 0^1 + 2^1 \cdot 2^1 + 3^1 \cdot 3^1 \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 5 & 13 \end{bmatrix}\end{aligned}$$

a pravá strana normální rovnice

$$\begin{bmatrix} (f, \varphi_0)_2 \\ (f, \varphi_1)_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^2 f(x_i) \cdot \varphi_0(x_i) \\ \sum_{i=0}^2 f(x_i) \cdot \varphi_1(x_i) \end{bmatrix} = \begin{bmatrix} 1 \cdot 0^0 + 2 \cdot 2^0 + (-1) \cdot 3^0 \\ 1 \cdot 0^1 + 2 \cdot 2^1 + (-1) \cdot 3^1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$



Obsah

307. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 6.3. Pro funkci $f(x) := \cos(x)$ najděte na intervalu $\langle 0, \pi \rangle$ aproximaci polynomem třetího stupně $\varphi(x)$ pomocí metody nejmenších čtverců.

Řešení. Hledaná funkce je polynom třetího stupně

$$\varphi(x) := c_0 + c_1x + c_2x^2 + c_3x^3,$$

tedy úkolem je nalézt konstanty $c_0, c_1, c_2, c_3 \in \mathbb{R}$.

Uvažujme standardní bázi vektorového prostoru polynomů nejvýše třetího stupně

$$\begin{aligned} \varphi_0(x) &:= 1 & \varphi_2(x) &:= x^2 \\ \varphi_1(x) &:= x & \varphi_3(x) &:= x^3 \end{aligned}$$

a sestavme normální rovnice. Za tímto účelem bude nutno spočítat dané skalární součiny.

$$\begin{aligned} (\varphi_0, \varphi_0)_2 &= \int_0^\pi 1 \cdot 1 \, dx = [x]_0^\pi = \pi \\ (\varphi_1, \varphi_1)_2 &= \int_0^\pi x \cdot x \, dx = \left[\frac{x^3}{3} \right]_0^\pi = \frac{\pi^3}{3} \\ (\varphi_2, \varphi_2)_2 &= \int_0^\pi x^2 \cdot x^2 \, dx = \left[\frac{x^5}{5} \right]_0^\pi = \frac{\pi^5}{5} \\ (\varphi_3, \varphi_3)_2 &= \int_0^\pi x^3 \cdot x^3 \, dx = \left[\frac{x^7}{7} \right]_0^\pi = \frac{\pi^7}{7} \\ (\varphi_0, \varphi_1)_2 &= \int_0^\pi 1 \cdot x \, dx = \left[\frac{x^2}{2} \right]_0^\pi = \frac{\pi^2}{2} \\ (\varphi_0, \varphi_2)_2 &= \int_0^\pi 1 \cdot x^2 \, dx = \left[\frac{x^3}{3} \right]_0^\pi = \frac{\pi^3}{3} \\ (\varphi_0, \varphi_3)_2 &= \int_0^\pi 1 \cdot x^3 \, dx = \left[\frac{x^4}{4} \right]_0^\pi = \frac{\pi^4}{4} \end{aligned}$$



Obsah

308. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obsah

309. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

$$(\varphi_1, \varphi_2)_2 = \int_0^\pi x \cdot x^2 \, dx = \left[\frac{x^4}{4} \right]_0^\pi = \frac{\pi^4}{4}$$

$$(\varphi_1, \varphi_3)_2 = \int_0^\pi x \cdot x^3 \, dx = \left[\frac{x^5}{5} \right]_0^\pi = \frac{\pi^5}{5}$$

$$(\varphi_2, \varphi_3)_2 = \int_0^\pi x^2 \cdot x^3 \, dx = \left[\frac{x^6}{6} \right]_0^\pi = \frac{\pi^6}{6}$$

$$(\varphi_0, f)_2 = \int_0^\pi 1 \cdot \cos(x) \, dx = [\sin(x)]_0^\pi = 0$$

$$(\varphi_1, f)_2 = \int_0^\pi x \cdot \cos(x) \, dx = [x \sin(x) + \cos(x)]_0^\pi = \pi$$

$$(\varphi_2, f)_2 = \int_0^\pi x^2 \cdot \cos(x) \, dx = [x^2 \sin(x) + 2x \cos(x) - 2 \sin(x)]_0^\pi = -2\pi$$

$$\begin{aligned} (\varphi_3, f)_2 &= \int_0^\pi x^3 \cdot \cos(x) \, dx = [x^3 \sin(x) + 3x^2 \cos(x) - 6x \sin(x) - 6 \cos(x)]_0^\pi \\ &= 12 - 3\pi^2 \end{aligned}$$

Pak soustava normálních rovnic má tvar

$$\begin{bmatrix} \frac{\pi^2}{2} & \frac{\pi^3}{3} & \frac{\pi^4}{4} & \frac{\pi^5}{5} \\ \frac{\pi^3}{3} & \frac{\pi^4}{4} & \frac{\pi^5}{5} & \frac{\pi^6}{6} \\ \frac{\pi^4}{4} & \frac{\pi^5}{5} & \frac{\pi^6}{6} & \frac{\pi^7}{7} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \\ -2\pi \\ 12 - 3\pi^2 \end{bmatrix}$$

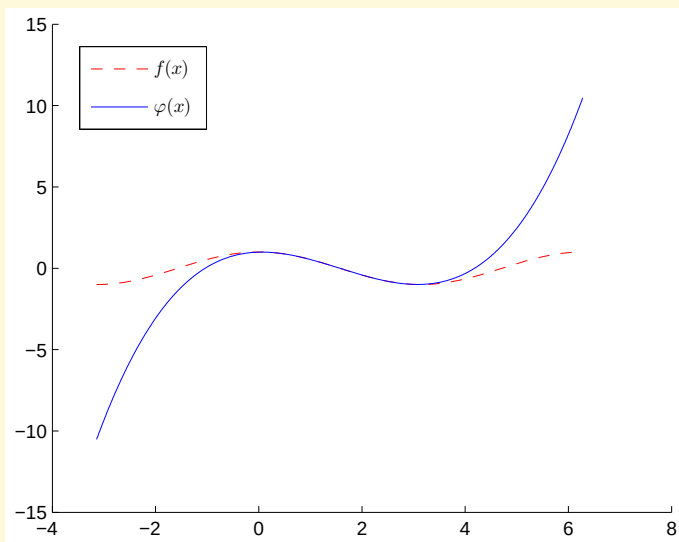
Tuto soustavu lze řešit numericky (viz kapitola 3)

$$c_0 \doteq 0.9910, \quad c_1 \doteq 0.0850, \quad c_2 \doteq -0.6836, \quad c_3 \doteq 0.1451.$$

Zbývá dosadit řešení do předpisu hledaného polynomu a získáme aproximační funkci

$$\varphi(x) = 0.9910 + 0.0850x - 0.6836x^2 + 0.1451x^3.$$

Porovnání zadané funkce f a aproximační funkce φ je vykresleno na Obrázku 6.4.



Obr. 6.4: Příklad 6.3 - řešení, aproximace metodou nejmenších čtverců.



Obsah

310. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

6.2. Ortogonální systémy funkcí

V poznámce 6.1 jsme upozornili na skutečnost, že volba standardních polynomických bází jako je $\phi_i = x^i$ vede na velmi špatně podmíněné soustavy normálních rovnic.

Této situaci se dá elegantně vyhnout pokud za bázi zvolíme ortogonální systém funkcí. Nejprve si tedy připomeňme, co rozumíme pod pojmem *ortogonální systém funkcí*.

Definice 6.4. Nechť je dán vektorový prostor Φ se skalárním součinem $(f, g), f, g \in \Phi$. Říkáme, že množina $\{\varphi_0, \dots, \varphi_b\} \subset \Phi$ je *ortogonální*, jestliže

$$(\varphi_i, \varphi_j) \begin{cases} = 0 & \text{pro } i \neq j \\ \neq 0 & \text{pro } i = j \end{cases}$$

Zobecněme skalární součin $(f, g)_2$ na prostoru $L_2(a, b)$ následovně

$$(f, g)_{2, \omega} = \int_a^b f(x)g(x)\omega(x)dx ,$$

kde $\omega(x)$ je tzv. *váhová funkce*, $\forall x \in (a, b) : \omega(x) > 0$.

Pro $f, g \in \mathbb{R}^{m+1}$ pak zobecňujeme Eukleidův skalární součin následovně

$$(f, g)_{2, \omega, m} = \sum_{i=0}^m f_i g_i \omega_i ,$$

kde $\omega_i > 0, i = 0, \dots, m$.



Obsah

311. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Uvedme příklady některých ortogonálních systémů:

Příklad 6.5.

a) spojitý případ

Funkce $\varphi_j(x) = \cos jx, j = 0, 1, \dots$ tvoří na $\langle 0, \pi \rangle$ ortogonální systém s $\omega(x) \equiv 1$

b) diskrétní případ

Funkce $\varphi_j(x) = \cos jx, j = 0, 1, \dots, m$ tvoří na síti bodů $x_i = \frac{2i+1}{m+1} \frac{\pi}{2}, i = 0, \dots, m$ s $\omega(x) \equiv 1$ ortogonální systém

Příklad 6.6. Čebyševovy polynomy

a) Polynomy $T_j(x) = \cos(j \arccos x), j = 0, 1, \dots$ tvoří na intervalu $\langle -1, 1 \rangle$ ortogonální systém s $\omega(x) = (1 - x^2)^{-\frac{1}{2}}$. Tyto polynomy můžeme taktéž získat rekurentním vzorcem

$$T_0(x) \equiv 1, T_1(x) = x, T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x)$$

b) Polynomy $T_j(x) = \cos(j \arccos x), j = 0, 1, \dots, m$ jsou ortogonální na síti

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

s $\omega_i \equiv 1$.



Obsah

312. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Příklad 6.7. Legendrovy polynomy

Polynomy

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

tvoří na $\langle -1, 1 \rangle$ ortogonální systém s $\omega(x) \equiv 1$.

Tyto polynomy je také možno získat rekurentním vzorcem

$$P_0(x) \equiv 1, P_1(x) = x, P_{j+1}(x) = \frac{2j+1}{j+1} x P_j(x) - \frac{j}{j+1} P_{j-1}(x).$$

Poznámka 6.8. Polynomy v Příkladech 6.6 a 6.7 lze rozšířit z intervalu $\langle -1, 1 \rangle$ na interval $\langle a, b \rangle$ prostřednictvím transformace

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



Obsah

313. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Lemma 6.9. *Nechť funkce $\varphi_0, \dots, \varphi_n$ jsou ortogonální. Pak řešení normálních rovnic má tvar*

$$c_i = \frac{(f, \varphi_i)}{(\varphi_i, \varphi_i)}, i = 0, \dots, n .$$

Z uvedeného Lematu 6.9 tedy plyne, že při použití ortogonálního bázevého systému funkcí není potřeba řešit soustavu normálních rovnic, neboť její matice je diagonální a řešení tudíž můžeme vypočítat přímo dosazením do rovnic

$$c_i = \frac{(f, \varphi_i)}{(\varphi_i, \varphi_i)} .$$

Tento postup pochopitelně zcela eliminuje jakýkoliv vliv numerické nestability na hledání aproximační funkce.



Obsah

314. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

6.3. Test

Následující test zahájíte kliknutím na tlačítko „Zacátek testu“. U otázek, kde lze volit jen jednu odpověď (test nedovolí zaškrtnout více odpovědí), je správná odpověď bodována počtem bodů uvedených v závorce u zadání a špatná odpověď je bodována 0 body. U otázek, kde lze volit více správných odpovědí, je součet bodů správných odpovědí uveden v závorce u zadání a za každou špatnou odpověď bude odečten jeden bod.

1. (4b.) Pod pojmem *aproximace funkce* rozumíme

nalezení funkce, která splňuje interpolační podmínky,

nalezení funkce, která nemusí splňovat interpolační podmínky, ale má nejmenší „vzdálenost“ od této funkce,

nalezení polynomu nejnižšího stupně, který splňuje alespoň jednu interpolační podmínku,

nalezení polynomu, který nejlépe interpoluje zadanou funkci,

určení okrajových podmínek splajnu,

funkci, která se v každém bodě uvažovaného intervalu rovná původní funkci,

nalezení polynomu, jehož integrál na uvažovaném intervalu je nejnižší možný.



Obsah

315. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2. (4b.) Za prostor aproximačních funkcí volíme
vždy prostor polynomů nejvýše pátého stupně,
Krylovovy prostory,
jednorozměrný reálný prostor,
jednodušší prostory funkcí, než ze kterého pochází původní funkce.
3. (4b.) Hodnotu „vzdálenosti“ původní aproximované funkce a aproximační funkce lze měřit
pomocí pravítka nebo posuvného měřidla,
pomocí příslušné metriky,
pomocí Krylovových prostorů,
pomocí Čebyševových polynomů,
vždy pomocí integrální normy.
vždy pomocí Eukleidovské normy.



Obsah

316. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4. (4b.) Metoda nejmenších čtverců pro aproximaci funkce zadané na diskrétní síti uzlů vybírá za aproximační polynom ten, jehož součet čtverců odchylek od funkčních hodnot v uzlech sítě je nejmenší,
- minimalizuje integrální normu,
- hledá aproximační polynom pomocí Čebyševových polynomů,
- vybírá vhodné prvky Krylovova prostoru,
- interpoluje zadanou funkci na dané síti bodů,
- vybírá za aproximační polynom ten, jehož Eukleidovská norma je nejmenší.
5. (4b.) Výsledný aproximační polynom závisí na
- konstrukci vhodné spline funkce,
- volbě báze daného prostoru polynomů,
- přesnosti použitého interpolačního polynomu,
- volbě použité metriky,
- matematickém modelu úlohy.



Obsah

317. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

6. (4b.) Pro určení aproximačního polynomu pomocí metody nejmenších čtverců je nutno vyřešit soustavu s Vandermondovou maticí,
interpolovat zadanou funkci na síti bodů,
zkonstruovat lineární spline na zadané síti bodů,
sestavit a vyřešit soustavu normálních rovnic,
ověřit pozitivní definitnost Grammovy matice.

7. (4b.) Necht je dána síť bodů a příslušné funkční hodnoty

$$\begin{array}{rcl} x_1 & = & 0 \quad f(x_1) = 1 \\ x_2 & = & 1 \quad f(x_2) = 2 \end{array}$$

Pak polynom nultého stupně, který nejlépe aproximuje funkci f ve smyslu nejmenších čtverců má předpis

$$\varphi(x) = x + 1,$$

$$\varphi(x) = x + 2,$$

$$\varphi(x) = 0,$$

$$\varphi(x) = 1,$$

$$\varphi(x) = 2,$$

$$\varphi(x) = 3,$$

$$\varphi(x) = 1/2,$$

$$\varphi(x) = 3/2,$$

neexistuje.



Obsah

318. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8. (4b.) Necht je dána funkce $f(x) = \sin(x)$. Pak polynom nultého stupně, který nejlépe aproximuje funkci f na intervalu $\langle 0, \pi/2 \rangle$ ve smyslu nejmenších čtverců má předpis

$$\varphi(x) = \pi,$$

$$\varphi(x) = 0,$$

$$\varphi(x) = \pi/2,$$

$$\varphi(x) = 2/\pi,$$

$$\varphi(x) = 1,$$

$$\varphi(x) = -1,$$

$$\varphi(x) = -\pi/2,$$

neexistuje.

9. (4b.) Mezi ortogonální systémy funkcí patří například (vyberte všechny možné správné možnosti)

Legendrovy polynomy,

standartní báze prostoru polynomů,

Čebyševovy polynomy,

prvky krylovova prostoru,

Newtonovy interpolační polynomy,

polynomy generované metodou
nejmenších čtverců,

Lagrangeovy interpolační polynomy,

kubické spline funkce.



Obsah

319. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

10. (4b.) Systémy ortogonálních funkcí v metodách aproximací zavádíme, protože (vyberte nejsilnější tvrzení)

příslušná Grammova matice je regulární,

příslušná Grammova matice je dobře podmíněná, tedy snadno řešitelná gradientní metodou,

příslušná soustava normálních rovnic má řešení,

příslušná matice soustavy normálních rovnic je diagonální.

11. (4b.) Ortogonální systémy funkcí

lze použít pouze pro interpolaci na intervalu $\langle -1, 1 \rangle$,

lze použít pouze pro interpolaci na intervalu $\langle 0, \pi \rangle$,

lze použít na libovolné intervalu pomocí jednoduché lineární transformace,

nelze použít při řešení úlohy aproximace.



Obsah

320. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

12. (4b.) Necht f je dána sít $n + 1$ aproximačních bodů a příslušných $n + 1$ funkcích hodnot funkce f . Pak pokud funkci f aproximujeme polynomem n -tého stupně, pak
- tento polynom je současně interpolačním polynomem,
 - globální chyba aproximace je nulová,
 - tento polynom je současně lineárním splinem,
 - tento polynom je současně kubickým splinem.

Správně zodpovězené otázky:

Získané body:

Procento úspěšnosti:

Pro ukončení testu je třeba kliknout na tlačítko „Konec testu“. Opravení testu se provede kliknutím na tlačítko „Výsledky“, správné odpovědi budou označeny zeleně a červeně budou označeny odpovědi chybné. Při použití otázky s tvořenou odpovědí se správná odpověď zobrazí v rámečku umístěném vpravo dole v navigačním panelu po kliknutí na tlačítko „Odpověď“.



Obsah

321. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Kapitola 7

Numerická derivace a integrace

Průvodce studiem

Výpočet derivace a integrálu je další velmi hojně používanou matematickou úlohou. Sami víme, že se však v případě velmi složitých funkcí může jednat o velice náročnou úlohu. Proto mnohdy nezbývá, nežli využít numerických metod k jejich vyčíslení.

V následující kapitole se proto seznámíme se základními numerickými metodami, které umožní najít jejich aproximace.

[Obsah](#)[322. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka/Okno](#)

Numerická derivace i integrace je založena na prosté myšlence nahradit původní derivovanou funkci či integrovanou funkci interpolačním polynomem a poté derivovat resp. integrovat tento polynom. Pochopitelně čím přesnější bude interpolační polynom, tím přesnější aproximaci derivace či integrálu dostaneme.

Jak je známo z teorie interpolace, její přesnost můžeme řídit vhodnou volbou rozmístění interpolačních uzlů. Toho se pak dá využít i při přesnějším vyhodnocení numerické derivace či integrálu.

[Obsah](#)[323.](#) strana ze 415[Zavřít dokument](#)[Celá obrazovka/Okno](#)

7.1. Numerická derivace

Jak již bylo zmíněno, numerická derivace spočívá v derivování interpolačního polynomu nahrazujícího derivovanou funkcí f . K tomuto účelu tedy použijeme Newtonův interpolační polynom

$$f(x) \approx N_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1})$$

První derivace pak může být numericky vypočítána následovně

$$f'(x) \approx N'_n = f[x_0, x_1] + f[x_0, x_1, x_2] \frac{\partial}{\partial x} [(x - x_0)(x - x_1)] + \dots \\ \dots + f[x_0, \dots, x_n] \frac{\partial}{\partial x} [(x - x_0) \dots (x - x_{n-1})]$$

Označíme-li chybu interpolace

$$E_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} e(x)$$

kde

$$\begin{aligned} e(x) &= (x - x_0) \dots (x - x_n) \\ \xi &\in \langle x_0, x_n \rangle. \end{aligned}$$

Pak chyba první derivace je

$$E'_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \frac{d}{dx} e(x).$$



Obsah

324. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Předpis pro k -tou numerickou derivaci funkce f pak má tvar

$$N_n^{(k)}(x) = f[x_0, \dots, x_k] \frac{d^k}{dx^k} [(x - x_0) \dots (x - x_{k-1})] + \dots \\ \dots + f[x_0, \dots, x_n] \frac{d^k}{dx^k} [(x - x_0) \dots (x - x_{n-1})]$$

a její chyba má tvar

$$E_{n+1}^{(k)}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \frac{d^k}{dx^k} e(x) .$$

Pro výpočet numerické derivace pak obvykle volíme ekvidistantní síť uzlů, tj.

$$x_{i+1} - x_i = h, \quad i = 0, \dots, n-1.$$

Pak například pro volbu $x_1 = x$ získáváme interpolační body

$$\begin{aligned} x_0 &= x - h \\ x_1 &= x \\ x_2 &= x + h \\ &\dots \end{aligned}$$



Obsah

325. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Uvedme tedy několik základních formulí pro numerickou derivaci včetně jejich chyby:
Vzorce pro první derivaci

1. Stupeň interpolačního polynomu $n = 1$

Pro volbu

$$\begin{aligned}x_0 &= x \\x_1 &= x + h\end{aligned}$$

dostáváme následující formuli pro numerickou derivaci

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{1}{2}hf''(\xi) \quad \xi \in \langle x, x+h \rangle, \quad h > 0$$

Volbu x_0 a x_1 můžeme však provést i následovně

$$\begin{aligned}x_0 &= x - h \\x_1 &= x.\end{aligned}$$

Pak dostáváme následující vzorec pro první derivaci

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \frac{1}{2}hf''(\xi) \quad \xi \in \langle x-h, x \rangle, \quad h > 0$$



Obsah

326. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2. Stupeň interpolačního polynomu $n = 2$

Nejpřirozenější volbou pro uzly interpolace se v případě kvadratického interpolačního polynomu jeví

$$\begin{aligned}x_0 &= x - h \\x_1 &= x \\x_2 &= x + h.\end{aligned}$$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{1}{6}hf'''(\xi) \quad \xi \in \langle x-h, x+h \rangle, \quad h > 0$$

Velmi často se stává, že je nutné počítat i jednostranné derivace. V takovém případě můžeme volit

$$\begin{array}{ll}x_0 = x & \text{resp. } x_0 = x - 2h \\x_1 = x + h & x_1 = x - h \\x_2 = x + 2h & x_2 = x.\end{array}$$

Derivováním příslušných interpolačních polynomů pak dostáváme následující vzorce pro derivaci

$$\begin{aligned}f'(x) &= \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + \frac{1}{3}h^2 f'''(\xi) & \xi \in \langle x, x+2h \rangle, \quad h > 0 \\f'(x) &= \frac{3f(x) - 4f(x-h) + f(x-2h)}{2h} + \frac{1}{3}h^2 f'''(\xi) & \xi \in \langle x-2h, x \rangle, \quad h > 0\end{aligned}$$



Obsah

327. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Vzorce pro druhé derivace

1. Stupeň interpolačního polynomu $n = 2$

$$\begin{aligned} f''(x) &= \frac{f(x+h)-2f(x)+f(x-h)}{h^2} - \frac{1}{12}h^2 f^{IV}(\xi) & \xi \in \langle x-h, x+h \rangle \\ f''(x) &= \frac{f(x+2h)-2f(x+h)+f(x)}{h^2} - hf'''(\xi) & \xi \in \langle x, x+2h \rangle \\ f''(x) &= \frac{f(x)-2f(x-h)+f(x-2h)}{h^2} - hf'''(\xi) & \xi \in \langle x-2h, x \rangle \end{aligned}$$

2. Stupeň interpolačního polynomu $n = 3$

$$f''(x) = \frac{12f(x) - 30f(x \pm h) + 24f(x \pm 2h) - 6f(x \pm 3h)}{6h^2} + O(h^3)$$

Poznámka 7.1. Je-li stupeň interpolačního polynomu shodný s řádem derivace (tj. $n = k$) pak

$$\begin{aligned} f^{(k)}(x) &\approx k! f[x_0, \dots, x_n] \\ f^{(k)}(x) &\approx \frac{\Delta^k f_0}{h^k} \end{aligned}$$



Obsah

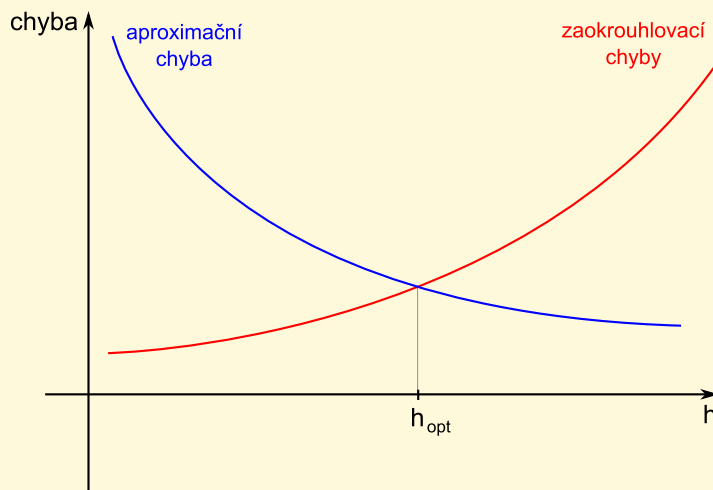
328. strana ze 415



Zavřít dokument

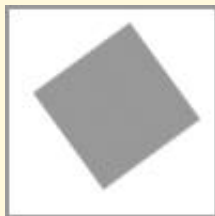
Celá obrazovka / Okno

Poznámka 7.2. Ze vzorců je patrné, že čím menší je h tím je vzorec přesnější. Na druhé straně však vždy dělíme h a pokud je h příliš malé, roste zaokrouhlovací chyba. Optimální volba h_{opt} je zobrazena na Obrázku 7.1.



Obr. 7.1: Volba optimální diskretizace

Příklad 7.3. Navrhněte jednoduchý gradientní algoritmus pro detekci hran na černobílém Obrázku 7.2.



Obr. 7.2: Příklad - detekce hran.

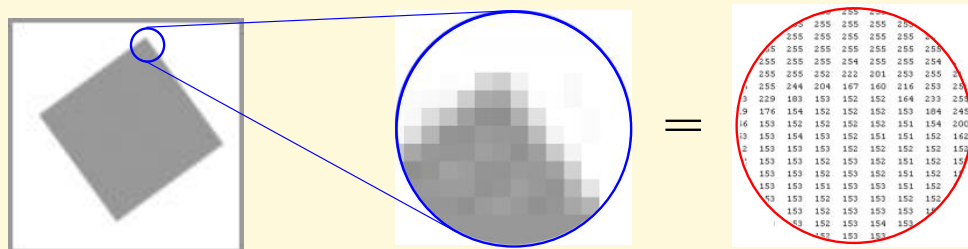
Řešení. Obrázek bude v počítači reprezentován jako dvourozměrné pole jasových hodnot v jednotlivých bodech (viz Obrázek 7.3).

Náš algoritmus nejprve spočítá změny jasové funkce ve směrech x, y v jednotlivých obrazových bodech a pokud bude v těchto bodech změna dostatečná, pak se jedná o hranu. Jelikož změna funkce je definována gradientem, bude nutno spočítat numerický gradient. Využijeme-li předpis pro numerickou derivaci jednorozměrné reálné funkce

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

a aplikujeme-li jej v obou směrech naší obrazové funkce, získáme předpis

$$\begin{aligned} f_x(x, y) &\approx f(x+1, y) - f(x, y) \\ f_y(x, y) &\approx f(x, y+1) - f(x, y) . \end{aligned}$$



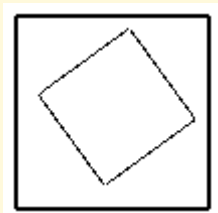
Obr. 7.3: Reprezentace obrazu v počítači.

Pak velikost změny je dána velikostí gradientu, tj.

$$e(x, y) = \sqrt{(f_x(x, y))^2 + (f_y(x, y))^2}.$$

Následně *prahováním* určíme pouze relevantní body, t.j. budeme uvažovat pouze body s dostatečnou velikostí změny jasové funkce (jinak řečeno body $[x, y]$ pro které $e(x, y) > \sigma$, kde $\sigma \in \mathbb{R}^+$ je pevně vhodně zvolená konstanta nebo-li *práh*).

Výsledek lze pozorovat na Obrázku 7.4.



Obr. 7.4: Řešení - detekce hran.

Ačkoliv tento jednoduchý algoritmus vypadá na první pohled účinně, při komplikovanějších obrázcích je těžké určit vhodný práh (viz Obrázek 7.5). Od algoritmu využívajícího informace o jasové funkci z dvou sousedících bodů však ani nelze očekávat uspokojivější výsledky. Více informací o zpracování a analýze obrazu lze nalézt například v [17].



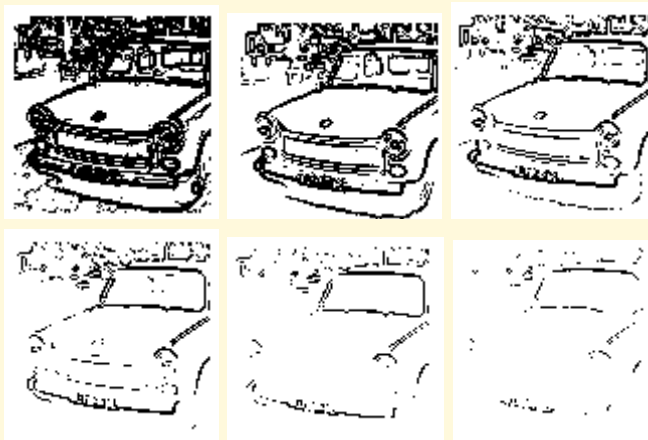
Obsah

332. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Obr. 7.5: Příklad detekce hran na složitějším obrázku.

Obsah

333. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

7.2. Newtonovy-Cotesovy vzorce

Nejjednodušší numerická metoda pro numerickou integraci vychází z integrace Lagrangeova interpolačního polynomu na ekvidistantní síti bodů.

Pokud nahradíme

$$f(x) \approx \varphi_n(x) = \sum_{i=0}^n f(x_i) l_i(x)$$

získáme vzorec pro numerickou integraci

$$I(f) = \int_a^b \sum_{i=0}^n f(x_i) l_i(x) dx = \sum_{i=0}^n w_i f(x_i) , \quad (7.1)$$

kde

$$w_i := \int_a^b l_i(x) dx .$$

Věta 7.4. Pro chybu integračního vzorce (7.1) platí

$$E(f) = \begin{cases} = \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_a^b w_{n+1}(x) dx & \text{pro liché } n , \\ = \frac{f^{(n+2)}(\xi)}{(n+2)!} \int_a^b x w_{n+1}(x) dx & \text{pro sudé } n . \end{cases}$$

Nyní si odvodíme základní Newtonovy-Cotesovy vzorce.



Obsah

334. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

- *Obdélníkové pravidlo*

$$\begin{aligned}
 f &\approx \varphi_0, n = 0 \\
 f(x) &\approx f\left(\frac{a+b}{2}\right) \\
 I(f) &= \int_a^b f(x) dx \approx \int_a^b f\left(\frac{a+b}{2}\right) dx = (b-a)f\left(\frac{a+b}{2}\right) \\
 I(f) &= (b-a)f\left(\frac{a+b}{2}\right) + E(f)
 \end{aligned}$$

Pro chybu obdélníkového pravidla $E(f)$ platí

$$E(f) = \frac{1}{24}(b-a)^3 f''(\xi) .$$

Z Obrázku 7.6 je patrná i geometrická interpretace obdélníkového pravidla i odkud pochází název této numerické metody.



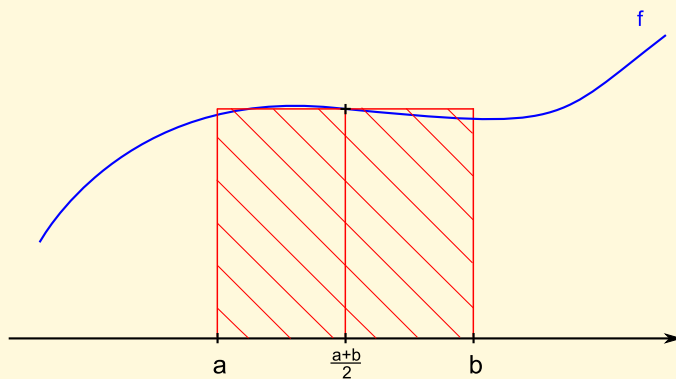
Obsah

335. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 7.6: Obdélníkové pravidlo

Obsah

336. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

- *Lichoběžníkové pravidlo*

$$\begin{aligned}f &\approx \varphi_1, n = 1 \\f(x) &\approx f(a) + \frac{f(b)-f(a)}{b-a}(x-a) \\I(f) &= \int_a^b f(x)dx \approx \frac{(b-a)}{2} (f(a) + f(b)) \\I(f) &= \frac{(b-a)}{2} (f(a) + f(b)) + E(f),\end{aligned}$$

kde

$$E(f) = -\frac{1}{12}(b-a)^3 f''(\xi)$$

reprezentuje chybu vzorce.

Z Obrázku 7.7 je patrná i geometrická interpretace lichoběžníkového pravidla z níž je zřejmé i pojmenování integračního vzorce.



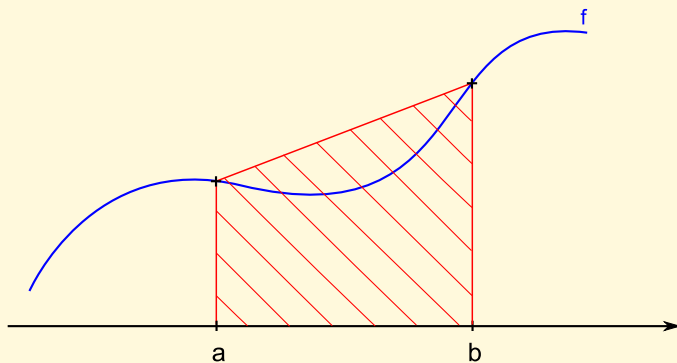
Obsah

337. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Obr. 7.7: Lichoběžníkové pravidlo

Obsah

338. strana ze 415

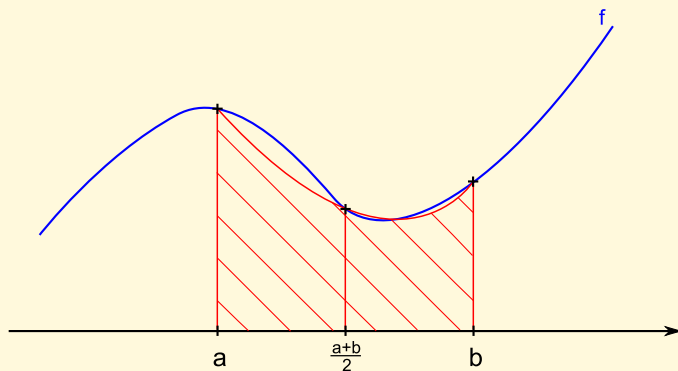


Zavřít dokument

Celá obrazovka / Okno

- *Simpsonovo pravidlo*

$$\begin{aligned}
 f &\approx \varphi_2, n=2 \\
 f(x) &\approx \frac{(x-\frac{a+b}{2})(x-b)}{(a-\frac{a+b}{2})(a-b)} f(a) + \frac{(x-a)(x-\frac{a+b}{2})}{(\frac{a+b}{2}-a)(\frac{a+b}{2}-b)} f(\frac{a+b}{2}) + \frac{(x-a)(x-b)}{(b-a)(b-\frac{a+b}{2})} f(b) \\
 I(f) &= \frac{(b-a)}{6} (f(a) + 4f(\frac{a+b}{2}) + f(b)) + E(f) \\
 E(f) &= -\frac{1}{90} \left(\frac{b-a}{2}\right)^5 f^{IV}(\xi)
 \end{aligned}$$



Obr. 7.8: Simpsonovo pravidlo



Obsah

339. strana ze 415



Zavřít dokument

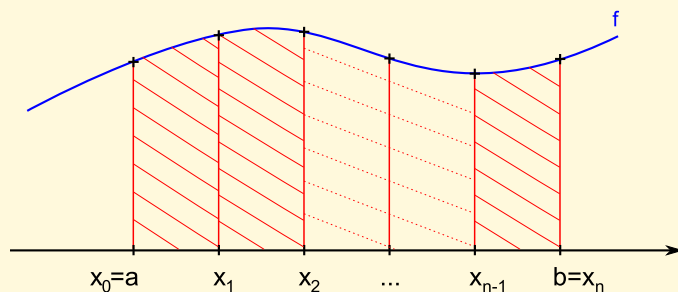
Celá obrazovka/Okno

Z předpisů chyb jednotlivých vorců je zřejmé, že pro velký integrační interval jsou vzorce díky velké chybě prakticky nepoužitelné. Jako vhodné řešení se nabízí rozdělení původního intervalu $\langle a, b \rangle$ na n podintervalů délky $h = \frac{b-a}{n}$.

Na jednotlivé podintervaly můžeme použít Newtonovy-Cotesovy vzorce a výsledný integrál je pak roven součtu integrálů přes všechny podintervaly.

$$I(f) = \int_{x_0}^{x_1} f(x)dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)dx$$

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx, \quad x_i = x_0 + ih, i = 0, \dots, n$$



Obr. 7.9: Složené pravidlo.



Obsah

340. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Výsledné integrační vzorce pak nazýváme *složenými* integračními vzorci a pro jednotlivé Newton-Cotesovy vzorce mají následující tvary:

- *Složené obdélníkové pravidlo*

$$\int_a^b f(x)dx = h \sum_{i=0}^{n-1} f\left(\frac{x_{i+1} + x_i}{2}\right) + \frac{1}{24}(b-a)h^2 f''(\xi)$$

- *Složené lichoběžníkové pravidlo*

$$\int_a^b f(x)dx = \frac{h}{2} \left(f(x_0) + f(x_n) + 2 \sum_{i=1}^{n-1} f(x_i) \right) - \frac{1}{12}(b-a)h^2 f''(\xi)$$

- *Složené Simpsonovo pravidlo*

$$\begin{aligned} \int_a^b f(x)dx &= \frac{h}{3} \left(f(x_0) + f(x_n) + 4 \sum_{i=1}^{m-1} f(x_{2i-1}) + 2 \sum_{i=1}^{m-1} f(x_{2i}) \right) \\ &\quad - \frac{1}{180}(b-a)h^4 f^{IV}(\xi), \quad n = 2m \end{aligned}$$



Obsah

341. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Algoritmus 25

SLOŽENÉ OBDÉLNÍKOVÉ PRAVIDLO

Input: f, a, b, n

```
1:  $h := (b - a)/n$ 
2:  $I := 0$ 
3: for  $i := 0, \dots, n - 1$  do
4:    $S_i := h \cdot f(ih + h/2)$ 
5:    $I := I + S_i$ 
6: end for
```

Output: I 

Obsah

342. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

7.3. Gaussovy vzorce

Z teorie interpolace vyplývá, že chybu interpolačního polynomu můžeme minimalizovat vhodným rozložením uzlů interpolace. Těmito body pak jsou obvykle kořeny ortogonálních polynomů. Stejně tak, můžeme zpřesnit výpočet integrálů, pokud nebudeme integrovat interpolační polynom na ekvidistantní síti bodů, nýbrž na uzlech, které jsou kořeny ortogonálních polynomů.

Výsledné integrační vzorce nazýváme *Gaussovými kvadraturními vzorci* a o jejich přesnosti vypovídá následující věta.



Obsah

343. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Věta 7.5. *Nechť $\{Q_j, j = 0, 1, \dots\}$ je soustava ortogonálních polynomů na intervalu $\langle a, b \rangle$ s volbou $\omega(x) \equiv 1$. Nechť $x_i, i = 0, \dots, n$ jsou kořeny polynomu Q_{n+1} .*

Sestrojme Gaussův kvadraturní vzorec

$$\int_a^b f(x)dx \approx \int_a^b \varphi(x)dx = \sum_{i=0}^n w_i f(x_i), \quad w_i = \int_a^b l_i(x)dx$$

Pak pro každý polynom $p_{2n+1} \in P_{2n+1}$ stupně $2n + 1$ platí

$$\int_a^b p_{2n+1}(x)dx = \sum_{i=0}^n w_i p_{2n+1}(x_i) .$$

Poznámka 7.6. Z výše uvedené věty vyplývá, že pokud za uzly integrace bereme kořeny ortogonálních polynomů, jsou kvadraturní formule přesné pro polynomy stupně $2n + 1$ na rozdíl od Newtonových-Cotesových formulí, které jsou přesné pouze pro polynomy stupně n .



Obsah

344. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 7.7. *Gaussovy-Legendrovy kvadraturní vzorce*

Za systém ortogonálních polynomů bereme Legendrovy polynomy definované v kapitole 6.2.

1. $n = 0$

$$\int_{-1}^1 f(x) dx = 2f(0) + \frac{1}{3}f''(\xi)$$

2. $n = 2$

$$\int_{-1}^1 f(x) dx = f\left(-\sqrt{\frac{1}{3}}\right) + f\left(\sqrt{\frac{1}{3}}\right) + \frac{1}{135}f^{IV}(\xi)$$

3. $n = 3$

$$\int_{-1}^1 f(x) dx = \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}f(0) + \frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right) + \frac{1}{15750}f^{VI}(\xi)$$

Poznámka 7.8. Integraci na intervalu $\langle a, b \rangle$ lze provést transformací

$$\bar{w}_i = \frac{b-a}{2}w_i, \quad \bar{x}_i = \frac{b+a}{2} + \frac{b-a}{2}x_i, \quad i = 0, \dots, n$$

kde w_i a x_i jsou váhy, resp. kořeny, z intervalu $\langle -1, 1 \rangle$.

Poznámka 7.9. Rozdělením intervalu integrace na n podintervalů můžeme zpřesnit výpočet použitím složených formulí obdobně jak v případě složených Newtonových-Cotesových vzorců.



Obsah

345. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

7.4. Výpočet integrálů pomocí extrapolace

Zatímco interpolací zjišťujeme hodnoty uvnitř intervalu interpolace, extrapolací nazýváme výpočet hodnot vně tohoto intervalu. Dá se totiž očekávat, že interpolační polynom bude dobře aproximovat i v blízkém okolí krajních bodů interpolace. Tohoto lze využít např. ke zpřesňování výpočtu integrálů.

Označme si $I(f)$ počítaný integrál a $I(f, h)$ složenou kvadraturní formuli s délkou podintervalů h . Provedeme-li výpočet integrálu pro dvě různé hodnoty h , např. h_1, h_2 , dostáváme dvě aproximace $I(f, h_1)$ a $I(f, h_2)$. Novou aproximaci pak získáme vzorcem

$$I^e(f, h) = \frac{1}{h_1^k - h_2^k} \left(h_1^k I(f, h_2) - h_2^k I(f, h_1) \right),$$

kde k je řád kvadraturní formule.

Speciální volbou $h_1 = 2h_2 =: h$ dostáváme vztah

$$I^e(f, h) = \frac{1}{2^k - 1} \left(2^k I(f, h) - I(f, 2h) \right)$$

a odhad chyby

$$I(f) - I^e(f, h) = \sum_{j=k+1}^{\infty} \frac{2^k - 2^j}{2^k - 1} a_j h^j$$

Z odhadu chyby je tedy patrné, že řád chyby nové aproximace je vyšší, konkrétně $k + 1$.



Obsah

346. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

7.5. Výpočet nevlastních integrálů

$$\int_{-\infty}^{\infty} f(x) dx \approx \int_{-R_1}^{R_2} f(x) dx$$

Hodnoty R_1 a R_2 přitom volíme tak, aby

$$\forall x \in \mathbb{R} \setminus \langle -R_1, R_2 \rangle : f(x) < \epsilon$$

kde ϵ je dostatečně malá konstanta.



Obsah

347. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

7.6. Test

Následující test zahájíte kliknutím na tlačítko „Zacátek testu“. U otázek, kde lze volit jen jednu odpověď (test nedovolí zaškrtnout více odpovědí), je správná odpověď bodována počtem bodů uvedených v závorce u zadání a špatná odpověď je bodována 0 body. U otázek, kde lze volit více správných odpovědí, je součet bodů správných odpovědí uveden v závorce u zadání a za každou špatnou odpověď bude odečten jeden bod.

1. (4b.) Numerická derivace je založena na

nahrazení funkce aproximačním
polynomem a poté derivováním,

nahrazení funkce interpolačním
polynomem a poté derivováním,

řešení speciálních soustav
diferenciálních rovnic.

nahrazení funkce aproximačním
polynomem a poté integrováním,

nahrazení funkce splinem a poté
derivováním,



Obsah

348. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

2. (4b.) Při použití interpolace Newtonovým interpolačním polynomem prvního stupně lze snadno odvodit předpis pro numerickou derivaci

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

$$f'(x) \approx \frac{x - f(x)}{h},$$

$$f'(x) \approx \frac{x + f(x)}{h},$$

$$f'(x) \approx \frac{h}{f(x+h) - f(x)},$$

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

3. (4b.) Při použití interpolace Newtonovým interpolačním polynomem druhého stupně lze snadno odvodit předpis pro numerickou derivaci

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{h},$$

$$f'(x) \approx \frac{f(x+h) + f(x-h)}{h},$$

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h},$$

$$f'(x) \approx \frac{2f(x+h)}{h},$$

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

[Obsah](#)

349. strana ze 415

[Zavřít dokument](#)[Celá obrazovka / Okno](#)

4. (4b.) Newtonovy-Cotesovy vzorce lze odvodit z

integrace lineární spline funkcí,

integrace Newtonova interpolačního polynomu na ekvidistantní síti uzlů,

integrace Lagrangeova interpolačního polynomu na ekvidistantní síti uzlů,

integrace aproximačního polynomu získaného metodou nejmenších čtverců na ekvidistantní síti uzlů.



Obsah

350. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

5. (4b.) Při použití obdélníkového pravidla pro numerický výpočet integrálu

$$\int_0^4 x^2 dx$$

získáme hodnotu

6. (4b.) Při použití lichoběžníkového pravidla pro numerický výpočet integrálu

$$\int_0^4 x^2 dx$$

získáme hodnotu

7. (4b.) Při použití složeného obdélníkového pravidla s ekvidistantním dělením na dva podintervaly pro numerický výpočet integrálu

$$\int_0^4 x^2 dx$$

získáme hodnotu



Obsah

351. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

8. (4b.) Gaussovy vzorce získáme

volbou ekvidistantní sítě interpolačních uzlů,

vhodnou volbou aproximačního polynomu,

vhodným rozložením interpolačních uzlů,

vhodnou volbou metriky.

9. (4b.) Newtonovy-Cotesovy formule odvozené z interpolačního polynomu stupně n jsou přesné pro integraci

polynomů stupně n ,

polynomů stupně $n - 1$,

polynomů stupně $n + 1$,

polynomů stupně $2n$,

polynomů stupně $2n - 1$,

polynomů stupně $2n + 1$.



Obsah

352. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

10. (4b.) Při použití ortogonálních polynomů v Gaussových-Legendrových vzorcích na síti uzlů z n kořenů těchto ortogonálních polynomů získáme integrační vzorce, které jsou přesné pro integraci

polynomů stupně n ,

polynomů stupně $n - 1$,

polynomů stupně $n + 1$,

polynomů stupně $2n$,

polynomů stupně $2n - 1$,

polynomů stupně $2n + 1$.

11. (4b.) Gaussovy-Legendrovy vzorce využívají pro integraci

Čebyševovy polynomy,

Legendrovy polynomy,

Newtonovy-Cotesovy vzorce,

Krylovovy prostory.



Obsah

353. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

12. (4b.) Pokud máme za úkol numericky určit hodnotu neurčitého integrálu

$$\int_{-\infty}^{\infty} f(x) dx$$

můžeme to udělat tak, že

neurčitý integrál nahradíme určitým integrálem. Integrační interval $\langle -R1, R2 \rangle$ volíme tak, aby

$$\forall x \in \mathbb{R} : \int_{-R1}^{R2} f(x) dx < \epsilon ,$$

kde ϵ je dostatečně malá konstanta.

neurčitý integrál nahradíme určitým integrálem. Integrační interval $\langle -R1, R2 \rangle$ volíme tak, aby

$$\forall x \in \mathbb{R} : f(x) < \epsilon ,$$

kde ϵ je dostatečně malá konstanta.

neurčitý integrál nahradíme určitým integrálem. Integrační interval $\langle -R1, R2 \rangle$ volíme tak, aby

$$\forall x \in \mathbb{R} : \int_{-\infty}^{-R1} f(x) dx + \int_{R2}^{\infty} f(x) dx < \epsilon ,$$

kde ϵ je dostatečně malá konstanta.



Obsah

354. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

neurčitý integrál nahradíme určitým integrálem. Integrační interval $\langle -R1, R2 \rangle$ volíme tak, aby

$$\forall x \in \mathbb{R} : \int_{-R1}^{R2} f(x) dx > \epsilon ,$$

kde ϵ je dostatečně malá konstanta.



Obsah

355. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Správně zodpovězené otázky:

Získané body:

Procento úspěšnosti:

Pro ukončení testu je třeba kliknout na tlačítko „Konec testu“. Opravení testu se provede kliknutím na tlačítko „Výsledky“, správné odpovědi budou označeny zeleně a červeně budou označeny odpovědi chybné. Při použití otázky s tvořenou odpovědí se správná odpověď zobrazí v rámečku umístěném vpravo dole v navigačním panelu po kliknutí na tlačítko „Odpověď“.



Obsah

356. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Kapitola 8

Řešení počátečních úloh pro obyčejné diferenciální rovnice

Průvodce studiem

Celá řada fyzikálních jevů, které jsou závislé na čase nebo na jedné prostorové proměnné se popisují pomocí obyčejných diferenciálních rovnic. Analytické řešení těchto diferenciálních rovnic je však mnohdy velmi složité. Z tohoto důvodu se pro řešení těchto úloh používají metody numerické.

V následující kapitole seznámíme se základními metodami řešení obyčejných diferenciálních rovnic.

[Obsah](#)[357. strana ze 415](#)[Zavřít dokument](#)[Celá obrazovka/Okno](#)

Definice 8.1. Obyčejnou diferenciální rovnici prvního řádu budeme rozumět rovnici, kterou lze napsat ve tvaru

$$y'(x) = f(x, y(x)), \quad (8.1)$$

kde

- $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ je zadaná funkce definující diferenciální rovnici,
- $y : \mathbb{R} \rightarrow \mathbb{R}$ je neznámá funkce proměnné x ,
- $x \in \mathbb{R}$ je reálná proměnná.

Poznámka 8.2. Samozřejmě, existují mnohem obecnější obyčejné diferenciální rovnice vyšších řádů nebo celé soustavy diferenciálních rovnic. Na jejich řešení se však obvykle dají velmi jednoduše aplikovat numerické metody, které řeší tento nejjednodušší typ diferenciální rovnice z Definice 8.1.



Obsah

358. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 8.3. Jednoduchým příkladem úlohy, která vede na obyčejnou diferenciální rovnici prvního řádu, je jednoduchý model množení králíků.

Označme-li y jako funkci popisující počet králíků v určitém čase $t \geq 0$, pak růst počtu králíků lze popsat rovnicí

$$y'(t) = ay(t),$$

kde $a \in \mathbb{R}$ je konstanta určující závislost mezi počtem králíků a rychlostí jejich množení (kopulační parametr).

Naším úkolem je nalézt funkci určující počet králíků v daném čase, tj. nalézt předpis funkce $y(t)$ (pro $t \geq 0$).

Analytické řešení¹ předchozí obyčejné diferenciální rovnice je

$$y(t) = c \cdot e^{at},$$

kde $c \in \mathbb{R}$ je libovolná konstanta.

Očividně předchozí úloha je nejednoznačná. Čím více králíků budeme mít na počátku (tj. v čase $t = 0$), tím strmější bude graf funkce y . Tato nejednoznačnost se v analytickém řešení projeví volbou parametru c .

Nechť $a = 0.3$, pak pro různé počáteční stavy $y(0)$ získáme různá řešení, viz Obrázek 8.1.

¹Analytickým řešením obyčejných diferenciálních rovnic se zabývá skriptum [3].



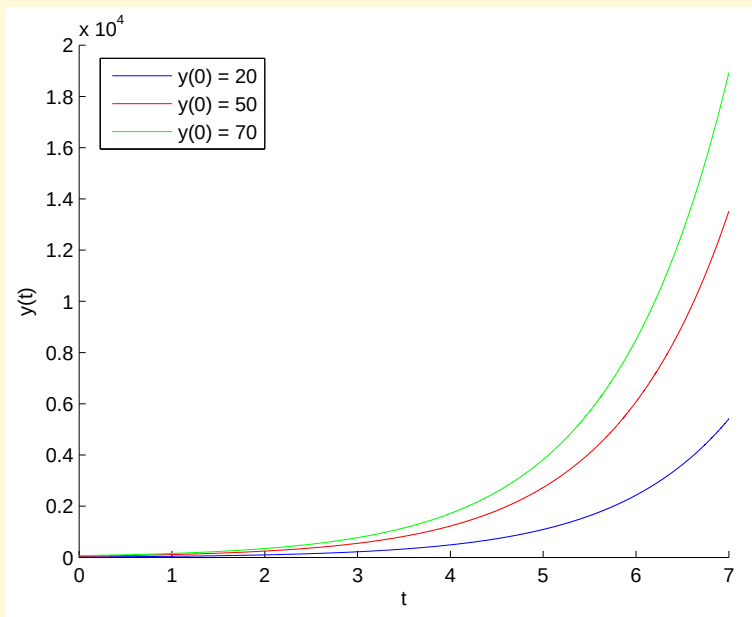
Obsah

359. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 8.1: Funkce popisující počet králíků v daném čase pro různé počáteční stavy.



Obsah

360. strana ze 415

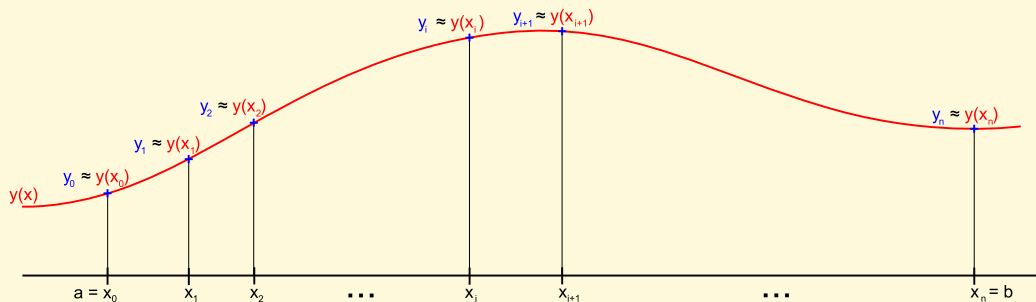


Zavřít dokument

Celá obrazovka / Okno

Upřesňující informace o hledaném řešení v počátečním bodě uvažovaného intervalu se nazývá *počáteční podmínka*.

Princip numerických metod řešení počátečních úloh pro obyčejné diferenciální rovnice prvního řádu spočívá v tom, že se nebude hledat analytický předpis řešení, ale pouze přibližné hodnoty řešení v bodech diskretizace uvažovaného intervalu, viz Obrázek 8.2.



Obr. 8.2: Algoritmus hledá řešení v pouze bodech diskretizace.



Obsah

361. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Cílem numerické metody pro řešení počátečních úloh pro diferenciální rovnice je tedy nalézt aproximace $y_0, \dots, y_n$ řešení $y(x_0), \dots, y(x_n)$

$$y_i \approx y(x_i), \quad i = 1, \dots, n$$

na síti bodů $\{x_0, x_1, \dots, x_n\}$,

$$\begin{aligned} x_{i+1} - x_i &= h_i \\ x_i &= x_0 + \sum_{j=0}^{i-1} h_j, \end{aligned}$$

kde h_i se nazývá *krok numerické metody*.

Numerické metody pro řešení obyčejných diferenciálních úloh dělíme na dvě základní třídy:

1. *jednokrokové*

K výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají pouze předchozí hodnotu aproximace y_i a mají obecný předpis

$$y_{i+1} = \Phi(y_i, x_i, f, h_i).$$

2. *vícekrokové*

K výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají k předchozích aproximacím $y_i, \dots, y_{i-k+1}$ a mají obecný předpis

$$y_{i+1} = \Psi(y_i, y_{i-1}, \dots, y_{i-k+1}, x_i, \dots, x_{i-k+1}, f).$$

Je-li $h_i := h = \text{const.}$ nazýváme numerickou metodu *s pevným krokem*, v opačném případě *s proměnlivým krokem*.



Obsah

362. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8.1. Jednokrokové metody

Nechť je dána obyčejná diferenciální rovnice prvního řádu s počáteční podmínkou

$$\begin{aligned}y'(x) &= f(x, y(x)) \\ y(x_0) &= y_0 .\end{aligned}\tag{8.2}$$

Naším úkolem je nalézt diskrétní řešení na intervalu $\langle a, b \rangle$, resp. na síti bodů

$$a = x_0 < x_1 < \cdots < x_n = b.$$



Obsah

363. strana ze 415



Zavřít dokument

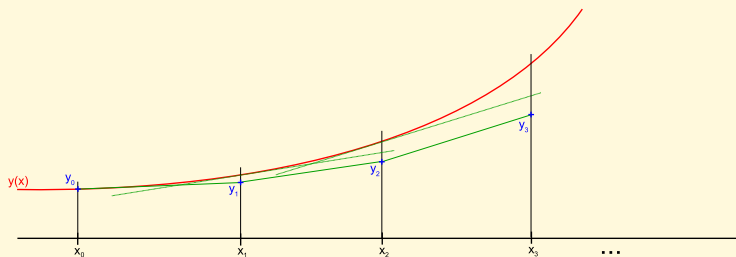
Celá obrazovka / Okno

8.1.1. Eulerova metoda

Nejjednodušší metodou pro numerické řešení úlohy (8.2) je tzv. *Eulerova metoda*. Přepokládejme, že v bodě x_0 je předepsána počáteční podmínka y_0 . Postupně budeme procházet intervaly diskretizace

$$\langle x_i, x_{i+1} \rangle, \quad i = 0, 1, \dots, n-1,$$

přičemž vždy vypočte aproximaci řešení y_{i+1} na základě již spočteného y_i .



Obr. 8.3: Princip Eulerovy metody.

Konkrétně hodnoty funkce y na intervalu $\langle x_i, x_{i+1} \rangle$ aproximujeme přímkou vycházející z bodu $[x_i, y_i]$

$$y(x) \approx y(x_i) + (x - x_i)y'(x_i), \quad \text{pro } x \in \langle x_i, x_{i+1} \rangle.$$

Jelikož pro směrnici platí $y'(x_i) = f(x_i, y(x_i))$ (dle rovnice (8.2)), pak dosazením získáme

$$y(x) \approx y(x_i) + (x - x_i)f(x_i, y(x_i)).$$



Obsah

364. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Nahradíme-li přesné hodnoty $y(x_i)$ jejich aproximacemi y_i , pak dostáváme předpis pro nalezení nové aproximace y_{i+1} v následujícím tvaru

$$\begin{aligned}y_{i+1} &= y_i + (x_{i+1} - x_i)f(x_i, y_i) \\y_{i+1} &= y_i + h_i f(x_i, y_i), \quad \text{pro } i = 0, 1, \dots, n-1.\end{aligned}$$

Tím jsme získali jednoduchý iterační předpis pro výpočet aproximace řešení.

[Obsah](#)

365. strana ze 415

[Zavřít dokument](#)[Celá obrazovka/Okno](#)

Poznámka 8.4. Tento algoritmus lze odvodit také z vzorce pro numerickou derivace. Je-li

$$y'(x) \approx \frac{y(x+h) - y(x)}{h} \quad (\text{pro dostatečně malé } h),$$

pak dle rovnice (8.2)

$$y'(x) = f(x, y(x)) \Rightarrow \frac{y(x+h) - y(x)}{h} = f(x, y(x))$$

a úpravou získáme předešlý iterační předpis.

Poznámka 8.5. Tento iterační předpis lze odvodit i z Taylorova polynomu

$$\begin{aligned} y(x_{i+1}) &= y(x_i + h_i) = y(x_i) + y'(x_i)h_i + \frac{y''(\xi)}{2!}h_i^2, \quad \xi \in (x_i, x_{i+1}) \\ y(x_{i+1}) &\approx y(x_i) + y'(x_i)h_i \\ y_{i+1} &= y_i + h_i f(x_i, y_i) \end{aligned}$$

Algoritmus 26

EULEROVA METODA

Input: $\{x_i\}_{i=0}^n, f(x, y), y_0$

```
1: for  $i = 1, 2, \dots, n-1$  do
2:    $h_i := x_{i+1} - x_i$ 
3:    $y_{i+1} := y_i + h_i f(x_i, y_i)$ 
4: end for
```

Output: $\{y_i\}_{i=0}^n$



Obsah

366. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Při studiu aproximační chyby, které se při použití metody dopouštíme, rozlišujeme dva základní typy:

- *Lokální diskretizační chyba*

Jedná se o chybu, které se dopouštíme v jednom kroku metody (tj. na jednom intervalu $\langle x_i, x_{i+1} \rangle$) s vyloučením zaokrouhlovacích chyb. Jelikož hodnotu $y(x_{i+1})$ aproximujeme hodnotou Taylorova polynomu prvního řádu, tj. $y(x_{i+1}) \approx T_1(x_i)$, dopouštíme se chyby řádově h_i^2 . Jinak řečeno

$$y(x_{i+1}) = y(x_i) + h_i f(x_i, y(x_i)) + d_i = T_1(x_i) + d_i, \quad d_i = \mathcal{O}(h_i^2),$$

kde d_i je lokální diskretizační chyba, které se dopouštíme na každém intervalu $\langle x_i, x_{i+1} \rangle$.

- *Globální diskretizační chyba*

Pod touto chybou rozumíme rozdíl přesného řešení a aproximace numerickou metodou v bodě x_i , tj.

$$e_i = y(x_i) - y_i.$$

Očividně tato chyba akumuluje všechny předchozí zaokrouhlovací i lokální diskretizační chyby $d_0, d_1, \dots, d_{n-1}$.

[Obsah](#)

367. strana ze 415

[Zavřít dokument](#)[Celá obrazovka / Okno](#)



Definice 8.6. Jestliže

$$b := x_n = x_0 + nh, \quad h := \frac{b-a}{n}, \quad n \in \mathbb{N},$$

je-li

$$\lim_{n \rightarrow 0} y_n = y(b),$$

pak říkáme, že je metoda *konvergentní*.

Definice 8.7. Je-li $d_i = \mathcal{O}(h_i^{p+1})$, pak říkáme, že je numerická *metoda řádu p* , tj.

$$\exists C = \text{const.} : |d_i| \leq Ch_i^{p+1} \text{ pro } h_i \rightarrow 0.$$

Poznámka 8.8. Obecně lze dokázat následující tvrzení

- metody alespoň 1. řádu jsou konvergentní
- pro globální diskretizační chybu v bodě x s pevným krokem h platí

$$e(x, h) = \mathcal{O}(h^p)$$

kde p je řád metody

- Eulerova metoda je řádu 1

Obsah

368. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 8.9. Řešte počáteční úlohu pro obyčejnou diferenciální rovnici

$$\begin{aligned}y' &= \frac{1+y^2}{xy(1+x^2)} \\ y(1) &= 1\end{aligned}$$

na intervalu $\langle 1, 5 \rangle$ pomocí Eulerovy metody.

Řešení. Analytické řešení úlohy (viz [3],[13]) má předpis

$$y(x) = \sqrt{\frac{3x^2 - 1}{1 + x^2}}.$$

Do algoritmu Eulerovy metody dosadíme pravou stranu diferenciální rovnice

$$f(x, y) = \frac{1 + y^2}{xy(1 + x^2)},$$

pak získáme pro různou volbu diskretizačního parametru h různě přesná řešení, viz Obrázek 8.4.

Z obrázku lze vidět, že pro $h \rightarrow 0$ numerické řešení konverguje k řešení přesnému.



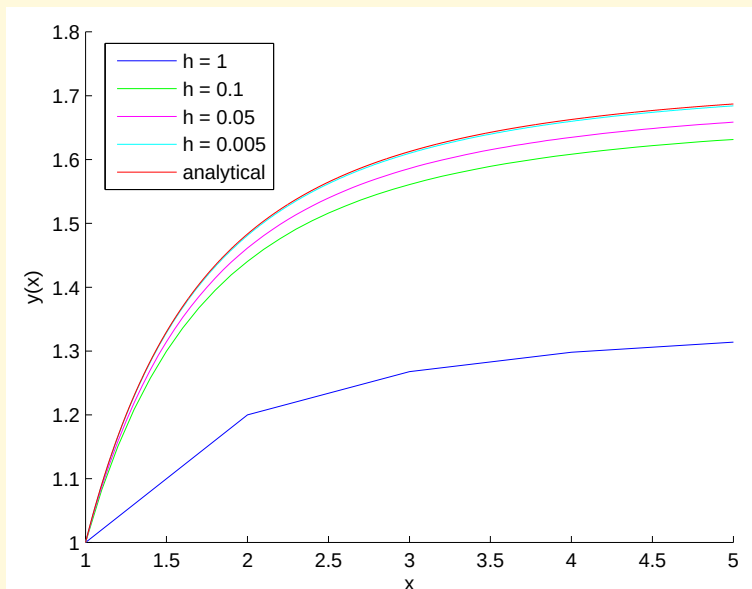
Obsah

369. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 8.4: Řešení Eulerovou metodou při použití různých diskretizačních parametrů.



Obsah

370. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

8.1.2. Metody Runge-Kutta

Pokusme se nyní vylepšit Eulerovu metodu. Pro výpočet y_i využijme odhady derivací ve více bodech. Pro celý krok potom použijeme vážený průměr těchto derivací s váhami α_i , konkrétně

$$y_{i+1} = y_i + h_i(\alpha_1 k_1 + \cdots + \alpha_r k_r) = y_i + h_i \sum_{j=1}^r \alpha_j k_j ,$$

přičemž pro odhady derivací platí

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ &\vdots \\ k_j &= f(x_i + \lambda_j h_i, y_i + \mu_j h_i k_{j-1}) \quad \text{pro } j > 1 \end{aligned}$$

Pro speciální volby parametrů α, λ a μ dostáváme následující metody.

[Obsah](#)

371. strana ze 415

[Zavřít dokument](#)[Celá obrazovka / Okno](#)

8.1.2.1. Metody Runge-Kutta 2. řádu

1. Pro volbu $r = 2$, $\alpha_1 = 0$, $\alpha_2 = 1$, $\lambda_2 = \mu_2 = \frac{1}{2}$ získáme předpis:

$$\begin{aligned} y_{i+1} &= y_i + h_i k_2 \\ k_1 &= f(x_i, y_i) \\ k_2 &= f\left(x_i + \frac{1}{2}h_i, y_i + \frac{1}{2}h_i k_1\right) \end{aligned}$$



Obr. 8.5: Metoda Runge-Kutta 2. řádu, 1. varianta.

Tato volba má svou geometrickou interpretaci, viz Obrázek 8.5. Funkci y na intervalu $\langle x_i, x_{i+1} \rangle$ aproximujeme přímkou vycházející z bodu $[x_i, y_i]$ se směrnici $y'_{i+\frac{1}{2}}$, tj. se směrnici rovnou derivaci v bodě uprostřed intervalu.

Porovnáním s Taylorovým polynomem lze dokázat, že odvozená metoda je řádu 2.



Obsah

372. strana ze 415



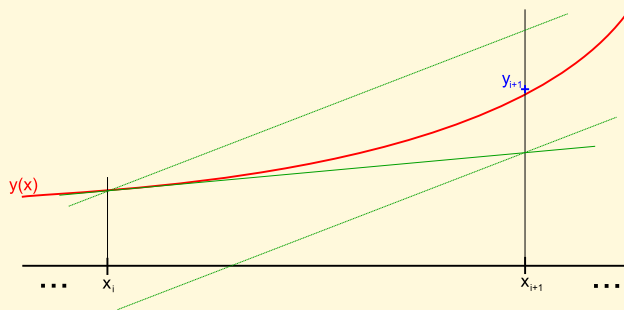
Zavřít dokument

Celá obrazovka / Okno

2. Volbou $r = 2$, $\alpha_1 = \alpha_2 = \frac{1}{2}$, $\lambda_2 = \mu_2 = 1$ získáme předpis:

$$\begin{aligned} y_{i+1} &= y_i + h_i \frac{k_1 + k_2}{2} \\ k_1 &= f(x_i, y_i) \\ k_2 &= f(x_{i+1}, y_i + h_i k_1) \end{aligned}$$

Tato metoda se také občas nazývá *Crank-Nicolsonova metoda*.



Obr. 8.6: Metoda Runge-Kutta 2. řádu, 2. varianta.

Funkci y na intervalu $\langle x_i, x_{i+1} \rangle$ aproximujeme přímkou vycházející z bodu $[x_i, y_i]$ se směrnicí rovnu aritmetickému průměru derivací v krajních bodech intervalu (viz Obrázek 8.6).

Obdobně jako u předchozí metody lze ukázat, že Crank-Nicolsonova metoda je řádu 2.



Obsah

373. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8.1.2.2. Metoda Runge-Kutta 4. řádu

Snad nejhojněji se používá metoda 4. řádu, která je dána předpisem.

$$\begin{aligned}y_{i+1} &= y_i + h_i \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \\k_1 &= f(x_i, y_i) \\k_2 &= f\left(x_i + \frac{1}{2}h_i, y_i + \frac{1}{2}h_i k_1\right) \\k_3 &= f\left(x_i + \frac{1}{2}h_i, y_i + \frac{1}{2}h_i k_2\right) \\k_4 &= f(x_{i+1}, y_i + h_i k_3)\end{aligned}$$

Bylo by možné odvodit i metody vyšších řádů, ale zde bychom již odkázali na literaturu [24] a [25], kde je možné tyto metody nalézt.



Obsah

374. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Příklad 8.10. Řešte počáteční úlohu

$$\begin{aligned} 12x^2 + 3y + 54 &= (60 + y')x \\ y(1) &= 1 \end{aligned}$$

na intervalu $\langle 1, 10 \rangle$ metodou Runge-Kutta druhého a čtvrtého řádu.

Řešení. Úloha má analytické řešení

$$y(x) = x^3 - 12x^2 + 30x - 18.$$

Tento předpis vyhovuje jak diferenciální rovnici, tak počáteční podmínce, navíc dle Picardovy-Lindelöfovy věty (viz [13]) existuje právě jedno řešení.

Nejdříve budeme muset upravit danou diferenciální rovnici do tvaru odpovídajícímu rovnici ((8.2)).

$$\begin{aligned} 12x^2 + 3y + 54 &= 60x + y'x \\ 3y + 12x^2 - 60x + 54 &= y'x \\ y' &= \frac{3y + 12x^2 - 60x + 54}{x} =: f(x, y). \end{aligned}$$

Pro numerické jsme volili ekvidistantní dělení s diskretizačním parametrem

$$h = \frac{b - a}{n} = \frac{9}{n}$$

Na obrázcích 8.7 a 8.8 naleznete srovnání přesnosti řešení pro různé volby počtu dělení intervalu na n podintervalů.. ▲

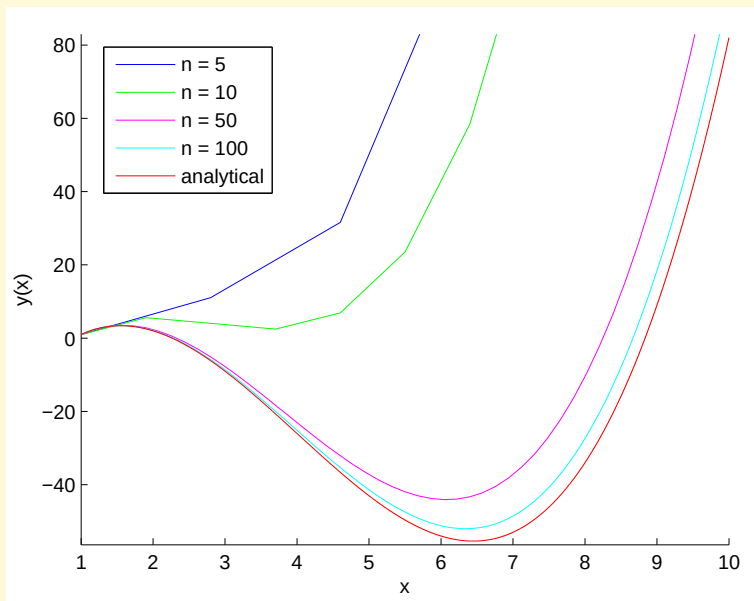
Obsah

375. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 8.7: Řešení metodou Runge-Kutta 2.řádu.

Poznámka 8.11. U výše uvedených Rungových-Kuttových metod se řád metody vždy rovná použitému počtu odhadů derivace. Tak to ale vždy není, například při pěti odhadech derivace se dá sestavit pouze metoda čtvrtého řádu. I z toho důvodu patří metoda čtvrtého řádu k velice oblíbeným.



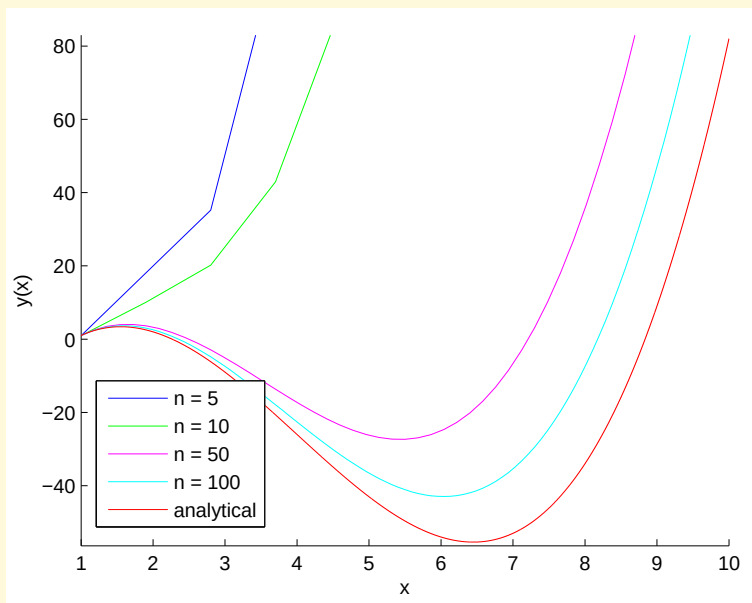
Obsah

376. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 8.8: Řešení metodou Runge-Kutta 4.řádu.

Obsah

377. strana ze 415

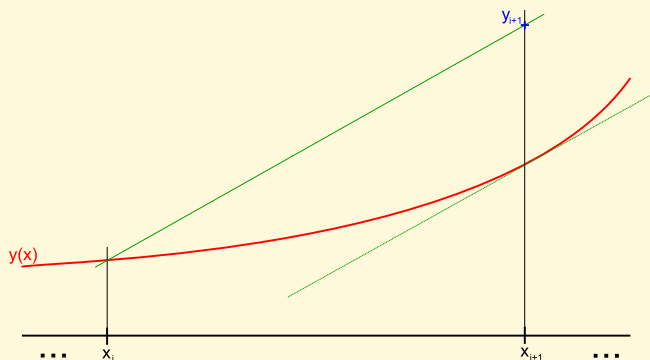


Zavřít dokument

Celá obrazovka/Okno

8.1.2.3. Implicitní metody

Hledanou funkci y aproximujeme na intervalu $\langle x_i, x_{i+1} \rangle$ úsečkou vychazející z bodu $[x_i, y_i]$ se směrnici y'_{i+1} , viz Obrázek 8.9.



Obr. 8.9: Princip implicitní metody.

Tedy tato metoda bude mít předpis

$$y_{i+1} = y_i + h_i f(x_{i+1}, y_{i+1}) .$$

Tzn., že jak na levé tak na pravé straně předpisu se vyskytuje neznámá y_{i+1} . V každém kroku musíme tedy řešit obecně nelineární rovnici $y_{i+1} = G(y_{i+1})$.

K tomu lze použít některou numerickou metodu (např. prosté iterace).



Obsah

378. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 8.12. Řešte okrajovou úlohu

$$\begin{aligned} 12x^2 + 3y + 54 &= (60 + y')x \\ y(1) &= 1 \end{aligned}$$

na intervalu $\langle 1, 10 \rangle$ implicitní Eulerovou metodou.

Řešení. Stejně jako v předchozím příkladě nejdříve nalezneme předpis pravé strany diferenciální rovnice

$$f(x, y) = \frac{3y + 12x^2 - 60x + 54}{x}$$

a pak dosazením do předpisu implicitní metody

$$y_{i+1} = y_i + h \cdot f(x_{i+1}, y_{i+1})$$

získáme rovnici

$$y_{i+1} = y_i + h \frac{3y_{i+1} + 12x_{i+1}^2 - 60x_{i+1} + 54}{x_{i+1}},$$

kterou budeme muset vyřešit v každé iteraci. Ještě předtím, než začneme uvažovat, kterou numerickou metodu použijeme pro řešení této dílčí úlohy, pokusíme se rovnici vyřešit analyticky. Získáme předpis

$$y_{i+1} = \frac{1}{x_{i+1} - 3h} [x_{i+1}y_i + h \cdot (12x_{i+1}^2 - 60x_{i+1} + 54)]$$

Jelikož původní rovnice byla lineární, neznámou y_{i+1} lze vyjádřit pomocí proměnných y_i a x_{i+1} .



Obsah

379. strana ze 415

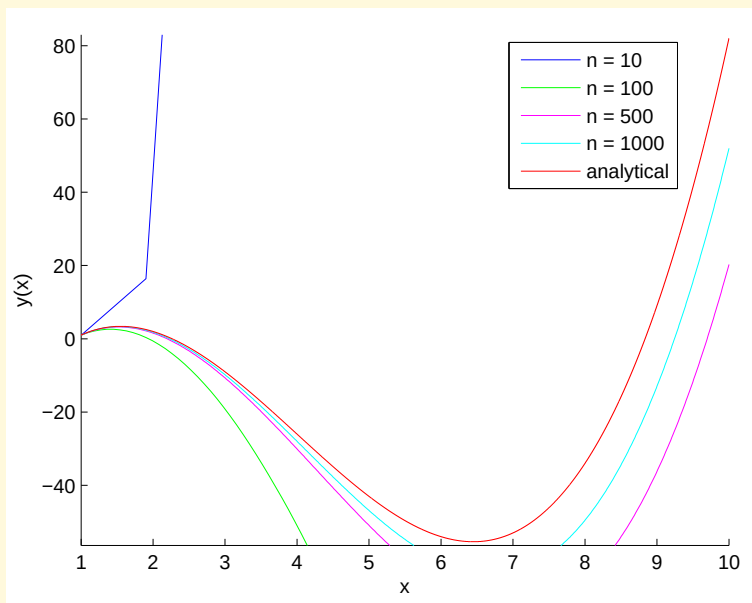


Zavřít dokument

Celá obrazovka/Okno

Řešení úlohy je vykresleno na Obrázku 8.10.

Obecně však tato rovnice může být nelineární, pak bychom museli volit například některou z metod uvedených v kapitole 2.



Obr. 8.10: Řešení implicitní metodou.



Obsah

380. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Poznámka 8.13. Všechny výše uvedené metody lze použít i pro soustavy diferenciálních rovnic, jak si ukážeme v kapitole 8.3.



Obsah

381. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8.2. Vícekrokové metody

Na rozdíl od jednokrokových metod budeme v případě více-krokových metod používat k výpočtu nové aproximace y_{i+1} k předchozích hodnot $y_i, y_{i-1}, \dots, y_{i-k+1}$, $k \geq 1$. V takovém případě budeme mluvit o tzv. k -krokové metodě.

Pro jednoduchost budeme uvažovat ekvidistantní síť bodů

$$a = x_0 < x_1 < x_2 < \dots < x_n = b \quad (8.3)$$

s diskretizačním parametrem h , tj. $\forall i = 0, \dots, n : x_i = x_0 + ih$.

Dále označme

$$f_i \stackrel{\text{ozn}}{=} f(x_i, y_i) .$$

Definice 8.14. *Lineární k -krokovou metodou* pro řešení počátečních úloh pro obyčejné diferenciální rovnice rozumíme metodu danou předpisem

$$\sum_{j=0}^k \alpha_j y_{i+j} = h \sum_{j=0}^k \beta_j f_{i+j}, \quad i = 0, 1, \dots, n-k \quad (8.4)$$

kde $\alpha_0, \dots, \alpha_k, \beta_0, \dots, \beta_k \in \mathbb{R}$ jsou dané konstanty splňující

$$|\alpha_0| + |\beta_0| > 0, \quad \alpha_k = 1 .$$



Obsah

382. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 8.15. Pokud zvolíme $k = 1$ (jednokrokovou metodu), lze rozepsat rovnici (8.4) na tvar

$$\alpha_0 y_i + \alpha_1 y_{i+1} = h(\beta_0 f_i + \beta_1 f_{i+1}) .$$

Pak volbou parametrů

$$\begin{array}{rcl} \alpha_0 & = & -1 \qquad \beta_0 = 1 \\ \alpha_1 & = & 1 \qquad \beta_1 = 0 \end{array}$$

získáme

$$-y_i + y_{i+1} = h f_i .$$

Tedy Eulerovu metodu.

Poznámka 8.16. K tomu abychom mohli předpis použít, musíme znát hodnoty $y_0, y_1, \dots, y_{k-1}$ (při výpočtu y_k použijeme těchto k předchozích hodnot). Tyto hodnoty nelze získat použitím k -krokové metody, proto pro jejich výpočet použijeme některou z jednokrokových metod.



Obsah

383. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 8.17. • Je-li $\beta_k = 0$, dostáváme metodu *explicitní*, tj. člen $f(x_{i+k}, y_{i+k})$ se v rovnici nevyskytuje a neznámou y_{i+k} lze vyjádřit přímo z předpisu

$$y_{i+k} = - \sum_{j=0}^{k-1} \alpha_j y_{i+j} + h \sum_{j=0}^{k-1} \beta_j f(x_{i+j}, y_{i+j}) .$$

• Je-li $\beta_k \neq 0$, dostáváme metodu *implicitní*, tj.

$$y_{i+k} + \sum_{j=0}^{k-1} \alpha_j y_{i+j} = h f(x_{i+k}, y_{i+k}) + h \sum_{j=0}^{k-1} \beta_j f(x_{i+j}, y_{i+j}) .$$

To znamená, že dostáváme v každém kroku úlohu

$$y_{i+k} = G(y_{i+k}) ,$$

kterou můžeme řešit některou numerickou metodou, např. metodou prostých iterací (viz 2.3)

$$y_{i+k}^{j+1} = G(y_{i+k}^j), \quad j = 0, 1, \dots$$

Otázkou však zůstává, jak volit parametry $\alpha_0, \dots, \alpha_k$ a $\beta_0, \dots, \beta_k$.



Obsah

384. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8.2.1. Metody založené na numerické integraci

Jelikož

$$y'(x) = f(x, y(x)) ,$$

pak integrací přes interval $\langle x_i, x_{i+1} \rangle$ získáme

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} f(x, y(x)) dx ,$$

a úpravou

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x)) dx .$$

Vyjádřeno ve formě aproximací

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y(x)) dx . \quad (8.5)$$

Integrál na pravé straně budeme počítat numericky na síti bodů $\{x_{i-k+1}, \dots, x_{i+1}\}$.



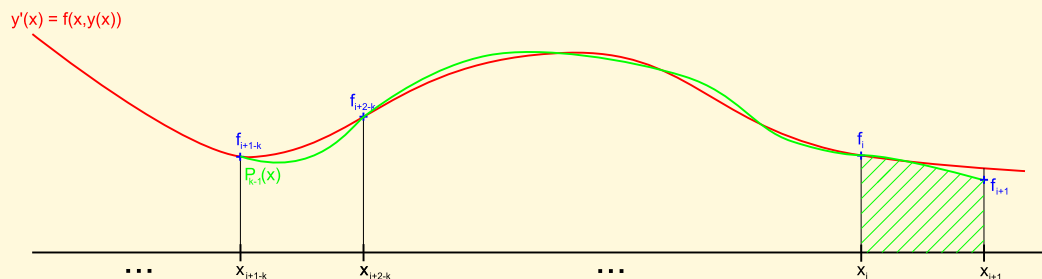
Obsah

385. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 8.11: Princip využití numerické integrace při řešení diferenciální rovnice.

Pro tento výpočet můžeme použít Newtonovy-Cotesovy formule (viz 7.2). K tomu však budeme potřebovat i bod $[x_{i+1}, f_{i+1}]$.

Existují dva způsoby jak tento bod získat:

- *extrapolací* (např. Adams-Bashforthovy metody) - pokud při integraci použijeme Lagrangeův interpolační polynom $P_{k-1}(x)$ stupně $k-1$ na síti $\{x_{i-k+1}, \dots, x_i\}$, lze extrapolovat hodnotu

$$f_{i+1} \approx P_{k-1}(x_{i+1})$$

a následně ji použít pro výpočet integrálu.

- *implicitně* (např. Adams-Moultonovy metody) - potřebnou hodnotu nebudeme vyčíslovat, naopak ji ponecháme jako neznámou. Pak získáme implicitní rovnici, kterou lze řešit např. prostými iteracemi.

Obsah

386. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

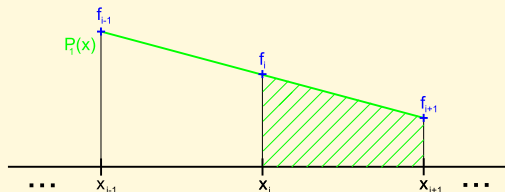
8.2.1.1. Adams-Bashforthovy metody

Odvodíme vzorec pro 2-krokovou Adams-Bashforthovu metodu.

Volme ekvidistantní síť uzlů (8.3) s krokem h . Nechť máme k dispozici aproximace řešení y_i, y_{i-1} (tj. máme k dispozici i f_i, f_{i-1}) a našim úkolem je nalézt předpis pro y_{i+1} ve tvaru (8.5).

Pro numerický výpočet požadovaného integrálu budeme potřebovat přibližnou hodnotu f_{i+1} . Nejdříve sestojíme interpolační polynom $P_1(x)$ prvního stupně na síti x_{i-1}, x_i , tj.

$$P_1(x) = a_0 + a_1x \quad \text{přičemž} \quad P_1(x_{i-1}) = f_{i-1} \quad \wedge \quad P_1(x_i) = f_i.$$



Obr. 8.12: Princip Adams-Bashforthovy dvoukrokové metody

Z požadovaných interpolačních vlastností získáme předpis

$$P_1(x) = \frac{x_{i-1}f_i - x_if_{i-1}}{x_{i-1} - x_i} + \frac{f_{i-1} - f_i}{x_{i-1} - x_i}x = \frac{1}{h} [x_if_{i-1} - x_{i-1}f_i + (f_i - f_{i-1})x].$$

Následně extrapolací spočteme hledanou hodnotu

$$f_{i+1} \approx P_1(x_{i+1}) = \frac{1}{h} [x_if_{i-1} - x_{i-1}f_i + (f_i - f_{i-1})x_{i+1}] = 2f_i - f_{i-1}.$$

Pak využijeme Newton-Cotesova vzorce pro integraci na intervalu $\langle x_i, x_{i+1} \rangle$ (Lichoběžníkové pravidlo)

$$\int_{x_i}^{x_{i+1}} f(x, y(x)) dx \approx \frac{x_{i+1} - x_i}{2} [f_i + (2f_i - f_{i-1})] = \frac{h}{2} (3f_i - f_{i-1})$$

a dosadíme do předpisu (8.5). Získáváme dvoukrokovou metodu ve tvaru

$$y_{i+1} = y_i + \frac{h}{2} (3f_i - f_{i-1}) .$$

Poznámka 8.18. Vícekrokové metody tohoto typu lze odvodit podobným způsobem. Například pro $k = 3$ a $k = 4$ získáme tyto předpisy

$$\begin{aligned} k = 3 \quad y_{i+1} &= y_i + \frac{1}{12} h (23f_i - 16f_{i-1} + 5f_{i-2}) \\ k = 4 \quad y_{i+1} &= y_i + \frac{1}{24} h (55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}) \end{aligned}$$



Obsah

388. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8.2.1.2. Adams-Moultonovy metody

Odvoďme vzorec pro 2-krokovou Adams-Moultonovu metodu.

Princip je stejný jako u Adams-Bashfortových formulí - interpolační polynom sestojíme však na síti bodů x_i, x_{i+1} .

V tomto případě se jedná o Lagrangeův interpolační polynom prvního stupně s předpisem

$$P_1(x) = a_0 + a_1x \quad \text{přičemž} \quad P_1(x_i) = f_i \quad \wedge \quad P_1(x_{i+1}) = f_{i+1} .$$

Z požadovaných interpolačních vlastností získáme předpis

$$P_1(x) = \frac{x_i f_{i+1} - x_{i+1} f_i}{x_i - x_{i+1}} + \frac{f_i - f_{i+1}}{x_i - x_{i+1}} x = \frac{1}{h} [x_{i+1} f_i - x_i f_{i+1} + (f_{i+1} - f_i)x] .$$

Nyní spočteme integrál potřebný pro předpis iterační metody (8.5). K jeho výpočtu použijeme Lichoběžníkové pravidlo

$$\int_{x_i}^{x_{i+1}} f(x, y(x)) \approx \int_{x_i}^{x_{i+1}} P_1(x) dx = \frac{h}{2} (f_i + f_{i+1}) .$$

Pak dosazením do předpisu (8.5)

$$y_{i+1} = y_i + \frac{h}{2} (f(x_{i+1}, y_{i+1}) + f_i) ,$$

kde y_{i+1} je neznámá na levé i pravé straně. V každé iteraci proto budeme řešit obecně nelineární soustavu

$$y_{i+1} = G(y_{i+1}) .$$



Obsah

389. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Poznámka 8.19. Vícekrokové metody tohoto typu lze odvodit podobným způsobem. Například pro $k = 3$ a $k = 4$ získáme tyto předpisy

$$\begin{aligned} k = 3 \quad y_{i+1} &= y_i + \frac{1}{12}h(5f(x_{i+1}, y_{i+1}) + 8f_i - f_{i-1}) , \\ k = 4 \quad y_{i+1} &= y_i + \frac{1}{24}h(9f(x_{i+1}, y_{i+1}) + 19f_i - 5f_{i-1} + f_{i-2}) . \end{aligned}$$



Obsah

390. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8.3. Soustavy diferenciálních rovnic a diferenciální rovnice vyšších řádů

8.3.1. Soustavy obyčejných diferenciálních rovnic

Definice 8.20. *Soustavou m obyčejných diferenciálních rovnic nazveme soubor m rovnic o m neznámých funkcí $y_1, \dots, y_m : \mathbb{R} \rightarrow \mathbb{R}$*

$$\begin{aligned} y_1'(x) &= f_1(x, y_1(x), y_2(x), \dots, y_m(x)) \\ y_2'(x) &= f_2(x, y_1(x), y_2(x), \dots, y_m(x)) \\ &\vdots \\ y_m'(x) &= f_m(x, y_1(x), y_2(x), \dots, y_m(x)) \end{aligned}$$

Poznámka 8.21. Soustavu lze zapsat vektorově

$$Y'(x) = F(x, Y(x))$$

kde

$$Y(x) = \begin{bmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_m(x) \end{bmatrix}, \quad F(x, Y(x)) = \begin{bmatrix} f_1(x, Y(x)) \\ f_2(x, Y(x)) \\ \vdots \\ f_m(x, Y(x)) \end{bmatrix}$$

Řešením rozumíme diferencovatelné funkce $Y(x)$ na intervalu $\langle a, b \rangle$, které splňují na $\langle a, b \rangle$ výše uvedené rovnice.

Na řešení této vektorové obyčejné diferenciální rovnice můžeme použít některou z již uvedených numerických metod.



Obsah

391. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8.3.2. Diferenciální rovnice vyšších řádů

Řešení libovolné diferenciální rovnice m -tého řádu

$$y^{(m)}(x) = f(x, y(x), y'(x), \dots, y^{(m-1)}(x))$$

je ekvivalentní s řešením soustavy

$$\begin{aligned} y'(x) &= u_1(x) \\ u_1'(x) &= u_2(x) \\ &\vdots \\ u_{m-1}'(x) &= f(x, y(x), u_1(x), \dots, u_{m-1}(x)) \end{aligned}$$

Tuto soustavu pak můžeme stejně jako v předešlé kapitole přepsat do vektorového tvaru

$$Y'(x) = F(x, Y(x))$$

a na její řešení můžeme opět využít numerických metod popsanych v kapitolách 8.1 a 8.2.



Obsah

392. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Příklad 8.22. Uvažujme matematické kyvadlo - hmotný bod zavěšený na tenkém vlákně zanedbatelné hmotnosti. Zanedbáme také odpor vzduchu při pohybu i tření v závěsu. Funkcí $\varphi(t)$ označme výchylku kyvadla z rovnovážné polohy v čase t .

Dá se ukázat (viz [12]), že tato úloha vede na obyčejnou diferenciální rovnici druhého řádu s okrajovými podmínkami

$$\begin{aligned}\varphi''(t) &= -\frac{g}{l} \sin \varphi(t) \\ \varphi(0) &= \varphi_0 \\ \varphi'(0) &= v_0\end{aligned}$$

kde

- g je tíhové zrychlení
- l je délka vlákna
- φ_0 je počáteční výchylka v čase $t = 0$
- v_0 je počáteční rychlost v čase $t = 0$

Zvolme konkrétní hodnoty $g = 9,8 \text{ m} \cdot \text{s}^{-2}$, $l = 1 \text{ m}$, $\varphi_0 = 0$, $v_0 = 1 \text{ m} \cdot \text{s}^{-1}$.

Řešení. Pokud při řešení využijeme substituci

$$u(t) = \varphi'(t),$$

tak získáme soustavu s okrajovými podmínkami

$$\begin{aligned}u'(t) &= -10 \sin \varphi(t) & u(0) &= 1 \\ \varphi'(t) &= u(t) & \varphi(0) &= 0\end{aligned}$$



Obsah

393. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Soustavu lze zapsat vektorově

$$Y'(t) = F(t, Y(t)), \quad Y(t) = \begin{bmatrix} u(t) \\ \varphi(t) \end{bmatrix}, \quad F(t, Y(t)) = \begin{bmatrix} \sin \varphi(t) \\ u(t) \end{bmatrix}$$

a použít Eulerovu metodu s počáteční hodnotou a iteračním krokem (pro jednoduchost volme konstantní diskretizační krok h)

$$\begin{bmatrix} u_0 \\ \varphi_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} u_{i+1} \\ \varphi_{i+1} \end{bmatrix} = \begin{bmatrix} u_i \\ \varphi_i \end{bmatrix} + h \begin{bmatrix} -\sin \varphi_i \\ u_i \end{bmatrix}$$

Při různé volbě diskretizačního kroku získáme různá řešení, která jsou zobrazena na Obrázku 8.13. Nicméně dle konvergence Eulerovy metody lze s jistotou tvrdit, že čím menší tento parametr zvolíme, tím přesnější řešení získáme.

Je očividné, že použití Eulerovy metody je numericky nestabilní. Předchozí výsledky naznačují, že kyvadlo postupně zvětšuje své výchyly při každém dalším kmitu, což je v rozporu s fyzikálními vlastnostmi matematického kyvadla.

Při použití metody Runge-Kutta 2.řádu, získáme pro různou volbu diskretizačního parametru h získáme řešení, která jsou na obrázku 8.14. Z obrázku je patrné, že metoda Runge-Kutta je mnohem stabilnější.

Vedlejším produktem obou metod je průběh derivace funkce $\varphi(t)$, tedy průběh funkce rychlosti pohybu kyvadla. ▲



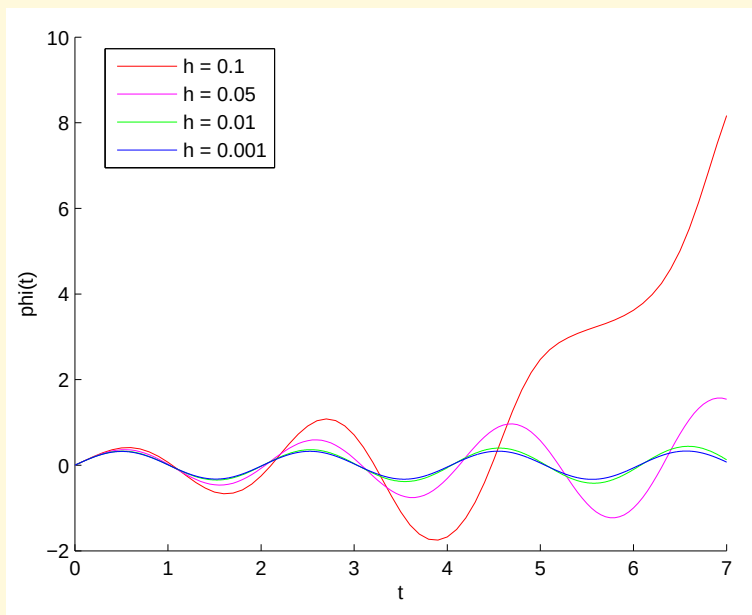
Obsah

394. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Obr. 8.13: Řešení úlohy matematického kyvadla Eulerovou metodou

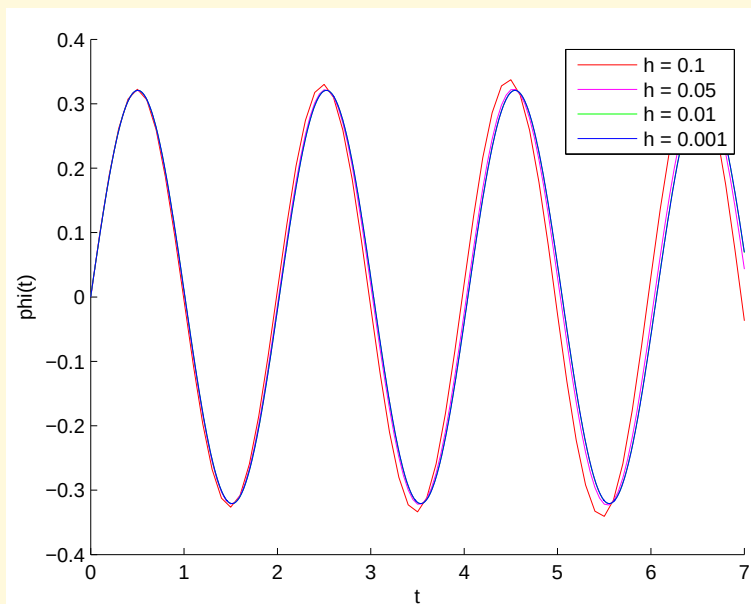
Obsah

395. strana ze 415



Zavřít dokument

Celá obrazovka/Okno



Obr. 8.14: Řešení úlohy matematického kyvadla metodou Runge-Kutta 2.řádu

Obsah

396. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

8.4. Test

Následující test zahájíte kliknutím na tlačítko „Zacátek testu“. U otázek, kde lze volit jen jednu odpověď (test nedovolí zaškrtnout více odpovědí), je správná odpověď bodována počtem bodů uvedených v závorce u zadání a špatná odpověď je bodována 0 body. U otázek, kde lze volit více správných odpovědí, je součet bodů správných odpovědí uveden v závorce u zadání a za každou špatnou odpověď bude odečten jeden bod.

1. (4b.) Princip numerického řešení obyčejných diferenciálních rovnic prvního řádu spočívá
v nalezení přibližného předpisu spojitého řešení,
v aproximaci analytického řešení polynomem,
v určení přibližné hodnoty řešení v bodech diskretizace uvažovaného intervalu,
v interpolaci analytického řešení vhodnou spline funkcí.



Obsah

397. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

**2. (4b.)** Jednokrokové numerické metody pro řešení obyčejných diferenciálních rovnic

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají hodnoty ve všech předchozích aproximacích,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají hodnoty ve dvou předchozích aproximacích,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají extrapolaci řešení v předchozích aproximacích,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají pouze předchozí hodnotu aproximace x_i .

3. (4b.) Vícekrokové numerické metody pro řešení obyčejných diferenciálních rovnic

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají hodnoty ve všech předchozích aproximacích,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají hodnoty v několika předchozích aproximacích,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají extrapolaci řešení ve všech předchozích aproximacích,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají pouze předchozí hodnotu aproximace x_i .

Obsah

398. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

4. (4b.) Eulerova metoda pro řešení obyčejných diferenciálních rovnic má předpis

$$y_{i+1} = y_i + f(x_i, y_i)/h_i,$$

$$y_{i+1} = y_i + h_i f(x_i, y_i),$$

$$y_{i+1} = y_i + h_i/f(x_i, y_i),$$

$$x_{i+1} = (f(x_{i+1}) - f(x_i))/(x_{i+1} - x_i).$$

5. (4b.) Pod pojmem *lokální diskretizační chyba* numerické metody pro numerické řešení obyčejné diferenciální rovnice rozumíme

chybu, které se dopouštíme v jednom kroku metody včetně zaokrouhlovacích chyb,

chybu, které se dopouštíme v jednom kroku metody s vyloučením zaokrouhlovacích chyb,

rozdíl přesného řešení a aproximace numerickou metodou v bodech diskretizace,

rozdíl přesného řešení a aproximace numerickou metodou na jednotlivých intervalech diskretizace.



Obsah

399. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

6. (4b.) Pod pojmem *globální diskretizační chyba* numerické metody pro numerické řešení obyčejné diferenciální rovnice rozumíme

chybu, které se dopouštíme v jednom kroku metody včetně zaokrouhlovacích chyb,

chybu, které se dopouštíme v jednom kroku metody s vyloučením zaokrouhlovacích chyb,

rozdíl přesného řešení a aproximace numerickou metodou v bodech diskretizace,

rozdíl přesného řešení a aproximace numerickou metodou na jednotlivých intervalech diskretizace.

7. (4b.) Numerická metoda je řádu p , pokud pro lokální diskretizační chybu platí $d_i = \mathcal{O}(h_i^{p+1})$ a existuje konstanta C taková, že pro $h_i \rightarrow 0$ platí

$$|d_i| \leq C^{p+1} \sqrt[p]{h_i},$$

$$|d_i| \geq C^{p+1} \sqrt[p]{h_i},$$

$$|d_i| \geq C h_i^{p+1},$$

$$|d_i| \leq C h_i^{p+1}.$$



Obsah

400. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



8. (4b.) Metody Runge-Kutta

jsou vícezkrokové metody,

využívají pro výpočet řešení v aktuální iteraci odhady derivací ve více bodech,

využívají pro výpočet řešení v aktuální iteraci odhad derivace v pouze v předchozím kroce,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívají extrapolaci řešení v předchozích aproximacích.

9. (4b.) Geometrické odvození Crank-Nicolsonovy metody lze popsat takto:

Funkci y na intervalu $\langle x_i, x_{i+1} \rangle$ aproximujeme přímkou vycházející z bodu $[x_i, y_i]$ se směrnici $y'_{i+\frac{1}{2}}$, tj. se směrnici rovnou derivaci v bodě uprostřed intervalu.

Funkci y na intervalu $\langle x_i, x_{i+1} \rangle$ aproximujeme přímkou vycházející z bodu $[x_i, y_i]$ se směrnici rovnou aritmetickému průměru derivací v krajních bodech intervalu.

Funkci y na intervalu $\langle x_i, x_{i+1} \rangle$ aproximujeme přímkou vycházející z bodu $[x_{i+1}, y_i]$ se směrnici rovnou aritmetickému průměru derivací v krajních bodech intervalu.

Funkci y na intervalu $\langle x_i, x_{i+1} \rangle$ aproximujeme přímkou vycházející z bodu $[x_{i+1}, y_{i+1}]$ se směrnici $y'_{i+\frac{1}{2}}$, tj. se směrnici rovnou derivaci v bodě uprostřed intervalu.

Obsah

401. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

10. (4b.) Implicitní metoda pro numerické řešení obyčejné diferenciální rovnice

vede na problém řešení obecně nelineární rovnice v každém kroku,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívá extrapolaci řešení v předchozích aproximacích,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívá pouze předchozí hodnotu aproximace x_i ,

vede na problém řešení soustavy lineárních rovnic v každém kroku,

k výpočtu nové aproximace y_{i+1} v bodě x_{i+1} využívá hodnoty ve všech předchozích aproximacích.

[Obsah](#)

402. strana ze 415

[Zavřít dokument](#)[Celá obrazovka / Okno](#)

11. (4b.) Lineární k -krokovou metodou pro řešení počátečních úloh pro obyčejné diferenciální rovnice rozumíme metodu danou předpisem

$$\sum_{j=0}^k \alpha_j y_{i+j} = h \sum_{j=0}^k \beta_j f_{i+j}, \quad i = 0, 1, \dots, n-k \qquad \sum_{j=0}^k \alpha_j x_{i+j} = h \sum_{j=0}^k \beta_j f_{i+j}, \quad i = 0, 1, \dots, n-k$$

$$\sum_{j=0}^k \alpha_j y_{i+j} = h \sum_{j=0}^k \beta_j x_{i+j}, \quad i = 0, 1, \dots, n-k \qquad \sum_{j=0}^k \alpha_j y_{i+j} = h \sum_{j=0}^k \beta_j f_{i+j}/x_{i+j}, \quad i = 0, 1, \dots, n$$

kde $\alpha_0, \dots, \alpha_k, \beta_0, \dots, \beta_k \in \mathbb{R}$ jsou dané konstanty splňující

$$|\alpha_0| + |\beta_0| > 0, \quad \alpha_k = 1.$$



Obsah

403. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

12. (4b.) Adams-Bashforthovy metody a Adams-Moultonovy metody jsou numerické metody

založené na numerické aproximaci,

založené na numerické extrapolaci,

založené na numerické derivaci,

založené na numerické integraci.

13. (4b.) Diferenciální rovnici vyššího řádu

nelze numericky řešit,

lze přímo řešit numerickou metodou pro řešení ODR prvního řádu,

lze aproximovat diferenciální rovnici prvního řádu,

lze řešit pomocí převodu na soustavu diferenciálních rovnic prvního řádu.



Obsah

404. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

Správně zodpovězené otázky:

Získané body:

Procento úspěšnosti:

Pro ukončení testu je třeba kliknout na tlačítko „Konec testu“. Opravení testu se provede kliknutím na tlačítko „Výsledky“, správné odpovědi budou označeny zeleně a červeně budou označeny odpovědi chybné. Při použití otázky s tvořenou odpovědí se správná odpověď zobrazí v rámečku umístěném vpravo dole v navigačním panelu po kliknutí na tlačítko „Odpověď“.



Obsah

405. strana ze 415



Zavřít dokument

Celá obrazovka / Okno



Literatura

- [1] Kozubek T. – Brzobohatý T. – Jarošová M. – Hapla V. – Markopoulos A. *Lineární algebra s Matlabem*. Matematika pro inženýry 21. století, 2012.
- [2] Dostál, Z. – Vondrák, V. – Lukáš, D. *Lineární algebra*. Matematika pro inženýry 21. století, 2012.
- [3] Krajc, B. – Beremlijski, P. *Obyčejné diferenciální rovnice*. Matematika pro inženýry 21. století, 2012.
- [4] Bouchala, J. *Úvod do funkcionální analýzy*. Matematika pro inženýry 21. století, 2012.
- [5] Dostál, Z. *Optimal Quadratic Programming Algorithms*. Springer, 2009.
- [6] Dostál, Z. *Lineární Algebra*. skriptum VŠB-TU Ostrava , 2000.
- [7] Děmidovič, B.P. – Maron, J.A. *Základy numerické matematiky*. SNTL, 1966.
- [8] Horová, I. – Zelinka, J. *Numerické metody*. Přírodovědecká fakulta Masarykovy univerzity v Brně, 2008.

Obsah

406. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

- [9] Nocedal, J. – Wright, S. J. *Numerical Optimization*. Springer, 1999.
- [10] Greguš, M. – Švec, M. – Šeda, V. *Obyčejné diferenciální rovnice*. SNTL, 1985.
- [11] Blata, Z. *Numerické metody řešení diferenciálních rovnic, diplomová práce*. Univerzita Tomáše Bati v Zlíně , 2009.
- [12] Vlček, J. *Matematické modelování, sylabus k předmětu*. VŠB-TU Ostrava , 2009.
- [13] Bouchala, J. *Matematika III pro bakalářské studium*. VŠB-TU Ostrava , 2000.
- [14] Bouchala, J. *Matematická analýza 1*. VŠB-TU Ostrava , 2005.
- [15] Wilkinson, J.H. *Algebraic Eigenvalue Problem*. Clarendon Press, Oxford , 1965.
- [16] Schatzman, M. *A simple proof of convergence of the QR algorithm for normal matrices without shifts*. IMA Preprint Series 720 , 1990.
- [17] Sojka, E. *Digitální zpracování a analýza obrazů*. Skriptum VŠB-TU Ostrava , 2000.
- [18] Golub, G. H. – Van Loan, Ch. F. *Matrix computations*. The Johns Hopkins University Press , 1996.
- [19] Laub, A. J. *Matrix Analysis for Scientists and Engineers*. SIAM, Philadelphia, PA, 2005.
- [20] Chen, K. *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, 2005.
- [21] Vitásek, E. *Numerické metody*. SNTL Praha, 1987.



Obsah

407. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

- [22] Benzi, M. *Preconditioning Techniques for Large Linear Systems: A Survey*. Journal of Computational Physics 182, 418–477, 2002.
- [23] Horn, R. A. – Johnson, Ch. J. *Topics in matrix analysis*. Cambridge University Press, 1991.
- [24] Quarteroni, A. – Sacco, R. – Saleri, F. *Numerical Mathematics*. Springer, 2000.
- [25] Sewell, G. *The Numerical Solution of Ordinary and Partial Differential Equations, Second Edition*. John Wiley & Sons, 2005.



Obsah

408. strana ze 415



Zavřít dokument

Celá obrazovka/Okno

Rejstřík

algoritmus

Eulerova metoda, 366
 interpolace polynomem, 226
 Krylovova metoda, 165
 kubická spline funkce, 273
 Lagrangeův interpolační polynom, 236 ■
 metoda prostých iterací, 66
 metoda půlení intervalu, 55
 metoda sdružených gradientů, 133
 nejmenších čtverců, 303
 Newtonova metoda, 73
 Newtonova metoda pro soustavy, 80
 Newtonovy-Cotesovy vzorce, 341
 Newtonův interpolační polynom, 243
 trigonometrická interpolace, 254

aproximace

metoda nejmenších čtverců, 300

princip, 294

čísel na počítači, 33, 37
 čísla, 24

charakteristický polynom, 158
 třídiagonální matice, 168

chyba

absolutní, 24
 aproximace, 23
 funkční hodnoty, 28
 matematického modelu, 23
 mocniny, 27
 numerické metody, 23
 násobku, 26
 obecný odhad, 29
 podílu, 27
 relativní, 24
 rozdílu, 27



Obsah

409. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

- součinu, 27
 - součtu, 26
 - ve vstupních datech, 24
 - zaokrouhlovací, 24
 - šíření, 26
- dominantní vlastní číslo, 171
- dvojnásobná aritmetika, 36
- floating point system, 36
- Gaussovy-Legendrovy vzorce, 345
- Gramův-Schmidtův proces, 130
- interpolace
 - kubickými spline funkcemi, 266
 - Lagrangeův interpolační polynom, 231
 - lineárními spline funkcemi, 260
 - Newtonův interpolační polynom, 239
 - polynomem, 224
 - příklad, 227
 - trigonometrická, 252
- interpolační podmínky, 224
- iterační metoda, 102
- kontraktivní zobrazení, 64
- korektní úloha, 16
- Krylovovy prostory, 133
- kubická spline funkce, 266
 - algoritmus, 273
 - chyba, 272
 - konstrukce, 268
 - okrajové podmínky, 270
 - příklad, 274
- kvadratická funkce, 116
- lineární spline funkce, 261
 - algoritmus, 262
 - konstrukce, 261
 - příklad, 263
- matice
 - Grammova, 302
 - ortogonální, 183
 - podobné, 182
 - rovinné rotace, 194
 - rovinného zrcadlení, 203
 - třídiagonální, 168
 - Vandermondova, 229
- metoda
 - Adams-Bashforthova, 387
 - Adams-Moultonova, 389
 - Eulerova



Obsah

410. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

- algoritmus, 366
- princip, 364
- příklad, 369
- Gaussova-Seidelova, 106
- Givensova, 194
 - algoritmus, 199
 - příklad, 197
- Householderova, 203
- implicitní
 - princip, 378
 - příklad, 378
- iterační, 102
- Jacobiova, 106
- jednokroková, 363
- konvergentní, 103
- konzistentní, 103
- Krylovova, 163
 - algoritmus, 165
 - princip, 163
- Lanczosova, 207
 - algoritmus, 211
 - princip, 207
- lineární, 102, 104
 - chyba, 104
 - konvergence, 105
 - příklad, 111, 113
- lineární k -kroková, 382
- mocninná
 - algoritmus, 174
 - konvergence, 173
 - normovaná, 175
 - princip, 172
 - příklad, 176
- nejmenších čtverců, 294, 300
 - algoritmus, 303
 - princip, 297, 300
 - příklad, 304, 308
 - volba metriky, 299
- největšího spádu
 - konvergence, 124
 - princip, 119
 - příklad, 125
- Newtonova
 - algoritmus, 73
 - konvergence, 76
 - modifikace, 77, 78
 - princip, 73
 - pro soustavy, 79
- prostých iterací
 - algoritmus, 66



Obsah

411. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

- chyba, 72
- konvergence, 64, 67
- princip, 64
- pro soustavy, 72
- předpodmínění, 136
- půlení intervalu, 54
 - algoritmus, 55
 - konvergence, 57
 - princip, 55
- QR, 184
 - konvergence, 184
 - příklad, 190, 191
- Runge-Kutta
 - 2. řádu, 372
 - 4. řádu, 374
 - princip, 371
 - příklad, 375
- sdužených gradientů
 - algoritmus, 133
 - konvergence, 135
 - princip, 128
- sečen, 78
- SOR, 106
- tečen, 75
- vícekroková, 382
- Newtonovy-Cotesovy vzorce, 334
- norma, 94
 - energetická, 124
 - Frobeniova, 99
 - maticová, 100
 - příklad, 94
 - vektorová, 94
- normální rovnice, 302
- numerická derivace, 324
 - příklad, 330
- numerická integrace, 343
 - extrapolací, 346
 - Gaussovy-Legendrovy vzorce, 345
 - nevlastní integrály, 347
 - Newtonovy-Cotesovy vzorce, 334
 - algoritmus, 341
 - lichoběžníkové pravidlo, 337
 - obdélníkové pravidlo, 335
 - Simpsonovo pravidlo, 339
 - ve složeném tvaru, 341
- obecný odhad chyby, 29
- obrácená úloha, 31
- pevný bod, 64
- podmíněná úloha, 17



Obsah

412. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

polynom

aproximační, 300

charakteristický, 158

definice, 224

interpolační, 224

Lagrangeův, 232

Newtonův, 239, 242

Legendrův, 313

ortogonální, 311

Čebyševův, 238, 312

poměrné difference, 241

počítačová jednotka, 37

počítačové epsilon, 37

příklad

lineární metoda, 111, 113

předpokládání, 136

Choleského faktorizace, 140

Jacobiho, 140

SSOR, 140

převod mezi soustavami, 34

příklad

absolutní chyba, 25, 30

Eulerova metoda, 369

Geršgorinova věta, 161

Givensova metoda, 197

interpolace polynomem, 227

Krylovova metoda, 166

kubická spline funkce, 274

Lagrangeův interpolační polynom, 233

lineární spline funkce, 263

matematické kyvadlo, 393

metoda nejmenších čtverců, 304, 308

metoda největšího spádu, 125

metoda prostých iterací, 70

metoda půlení intervalu, 58

metoda Runge-Kutta, 375

nestabilní metoda, 19, 21

Newtonova metoda pro soustavy, 81

Newtonův interpolační polynom, 244,
249

numerická derivace, 330

obrácená chyba, 32

podmíněnost úlohy, 18

převod mezi soustavami, 34

QR, 190, 191

relativní chyba, 25, 30

rychlost numerických řešičů soustav,
144

soustava diferenciálních rovnic, 393

stabilní úloha, 21



Obsah

413. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

- trigonometrická interpolace, 255
- třídiagonální matice, 169
- vlastní čísla, 159
- QR rozklad, 184
- Rayleighův podíl, 178
- reprezentace čísel na počítači, 33, 37
- semilogaritmický tvar s normalizovanou man-
tisou, 33
- separace kořenů, 47
 - grafická, 49
 - tabelací funkce, 51
- soustava
 - binární, 34
 - dekadická, 34
 - diferenciálních rovnic, 391
- spline
 - kubický, 266
 - lineární, 261
- systém s pohyblivou řádovou čárkou, 36
- vektor kořenů soustavy, 52
- věta
 - Cayley-Hamiltonova, 163
 - chyba Newtonova-Cotesova vzorce, 334
 - Geršgorinova, 160
 - konvergentní lineární metoda, 105
 - konzistentní lineární metoda, 104
 - o pevném bodě, 65
 - postačující podmínky, 66
 - posun spektra, 180
 - redukce spektra, 181
 - vektorová a maticová norma, 100
- zobrazení
 - kontraktivní, 64
- základní aritmetické operace, 26
- číslo
 - aproximace, 24
 - dominantní vlastní číslo, 171
 - numerické stability, 22
 - podmíněnost matice, 101
 - podmíněnosti úlohy, 17
 - převod, 34
 - Rayleighův podíl, 178
 - reprezentace, 33, 37
 - Schwarzova konstanta, 173
 - vlastní číslo, 158
 - řád konvergence, 53
- řád konvergence, 53



Obsah

414. strana ze 415



Zavřít dokument

Celá obrazovka / Okno

úloha

korektní, **16**

obrácená, **31**

podmíněnost, **17**

stabilní, **21**



Obsah

415. strana ze 415



Zavřít dokument

Celá obrazovka / Okno