UNIVERZITA J. E. PURKYNĚ V ÚSTÍ NAD LABEM
Přírodovědecká fakulta

Pokročilé numerické metody

KI/PNM?

RNDr. Petr Kubera, Ph.D.

Ústí nad Labem 2020

Kurz:	Pokročilé numerické metody
Obor:	Aplikovaná informatika (nmgr)
Klíčová slova:	numerické metody
Anotace:	Tento předmět nabízí studentům ucelený přehled metod numerické matematiky. Předmět dále rozvíjí znalosti a dovednosti z oblasti numerických metod, které by měli studenti získat v bakalářském studiu. Poskytuje hlubší vhled do problematiky a akcentuje metody numerické lineární algebry jakožto nástroje pro datovou analýzu a strojové učení. Studenti si osvojí jak teoretické znalosti, tak i praktické aplikace v rámci cvičení za použití volně dostupných knihoven pro Python a R.

Jazyková korektura nebyla provedena, za jazykovou stránku odpovídá autor.

Autor: RNDr. Petr Kubera, Ph.D.

Obsah

Úvodní slovo

Předmět je koncipován jako přehledový kurz v oblasti numerické matematiky. Předmět volně navazuje na předmět z bakalářského studia *Numerické metody*. Jedním z cílů předmětu je poskytnout přehled nástrojů numerické (nejen) lineární algebry, která je využita v metodách strojového učení, zpracování dat, redukce jejich dimensionality, či případně různých statistických testů časových řad (např. hledání kořenů char. polynomů)

Cílem je získat nejen obecné pochopení metod, ale znát i principy jejich implementace a hlavně schopnost využívat softwarové balíky. Vše proto budeme dělat za použití volně dostupných nástrojů a frameworků v Pythonu, případně R.

Splnění kurzu

Zápočet je možné získat za praktický test (řešení praktické úlohy na počítači). Zkouška je zaměřena na ověření teoretických znalostí.

1 Stabilita, zdroje zaokrouhlovacích chyb a jejich vliv

Protože řešení úloh numerické matematiky probíhá na počítači a obvykle v aritmetice s konečnou přesností, je nutné se věnovat vlivu zaokrouhlovacích chyb. Tento vliv je poměrně značný a může mít vliv na samotný proces výpočtu a to z toho důvodu, že obvykle jsou výpočty iterativní a zaokrouhlovací chyby se kumulují.

CÍLE KAPITOLY

- Reprezentace čísel v počítači - viz zde, v kapitole 1.
- Popis vlivu zaokrouhlovacích chyb na přesnost výpočtu - viz zde, zejména kapitola 1.
- Ukázka špatné podmíněnosti soustav lineárních rovnic je k dispozi zde.
- Problematika reprezentace čísel s plovoucí řádovou čárkou v Pythonu je diskutována zde.

KLÍČOVÁ SLOVA

zaokrouhlovací chyby, stabilita výpočtu

ÚKOLY

1. Prostudujte si následující typy problémů, které v "běžné" matematice nepůsobí problémy zde.
2. Dohledejte si na internetu nejméně tři příklady kumulace zaokrouhlovacích chyb a dané procesy naprogramujte.

OTÁZKY

1. Proč není většinou výhodné použít Taylorův rozvoj pro výpočet hodnoty dané funkce, např. exp.

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat princip problémů, které vznikají zaokrouhlováním.
- Uvést několik konkrétních nestabilních úloh.

ODKAZY NA LITERATURU

- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde, zejména kapitola 1.
- VONDRÁK, Vít a Lukáš POSPÍŠIL. Numerické metody 1. 2011. [online]: zde, zejména kapitola 1.

MÍSTO PRO VAŠE POZNÁMKY

2 Přímé metody pro řešení lineárních rovnic

Tato kapitola odpovídá bodu 2 a 3 sylabu předmětu.

Řešení lineárních rovnic je jednou ze základních úloh numerické matematiky. Potřeba řešit lineární rovnice vyvstává z technické praxe, kde je třeba řešit velmi velké systémy rovnic. S tím se také váže otázka numerické stability. Přímé metody poměrně často trpí na vliv zaokrouhlovacích chyb. Důvod je ten, že ač zde přímo není iterační proce řešení soustav, tak se iteračně konstruuje matice ve tvaru vhodném pro řešení a tento proces ovliňuje přesnost výpočtu. Existují uměle zkonsturované problémy, které bez nějakého stabilizačního mechanismu, např. pivotizace, pro větší počet rovnic přesně nespočítáte.

CÍLE KAPITOLY

- Gaussova eliminační metoda (GEM) - viz zde, v kapitole 2.
- Varianty LU rozkladu pro řešení l. rovnic - viz zde, v kapitole 2.
- Aplikace LU rozkladu na výpočet determinantu a inverze matice, třídiagonální systém - viz zde, v kapitole 2.
- Variantu LU rozkladu pro positivně definitní matice, tzv. Choleského faktorizace je diskutována zde zde.
- Konstrukce QR dekompozice je popsána zde, či pomocí jiného algoritmu zde.
- Balík pro numerickou lineární algebru v Pythonu *numpy.linalg* - viz zde.

KLÍČOVÁ SLOVA

přímé metody pro soustavy l. rovnic, LU rozklad, QR rozklad

ÚKOLY

1. Experimentálně ověřte vliv pivotizace pro GEM.
2. Implementujte všechny výše uvedené metody pro řešení lineárních rovnic a jejich soustav. Můžete se motivovat s implementací v jazyce C/C++ viz zde.
3. Diskutujte rychlost jednotlivých algoritmů pro soustavy o různé velikosti.
4. Seznamte se s *numpy.linalg* - viz zde.

OTÁZKY

1. Jaké jsou podmínky existence LU rozkladu ?
2. Jaký vliv má pivotizace na časovou náročnost GEM ?
3. Porovnejte časovou náročnost GEM s Gaussovou-Jordanovou eliminační metodou.
4. Jaký vliv mají zaokrouhlovací chyby na konstrukci QR rozkladu ?

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat princip daných metod a jejich výhody a nevýhody.
- Znát možnosti jejich elementární implementace.

ODKAZY NA LITERATURU

- LIMPOUCH, Jiří, Numerické metody lineární algebry, [online] zde
- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde, zejména kapitola 1.

MÍSTO PRO VAŠE POZNÁMKY

3 Iterační metody pro řešení lineárních rovnic

Tato kapitola odpovídá bodu 4 a 5 sylabu předmětu.

Klíčové pro iterační metody pro řešení soustav lineárních rovnic je to, že se konstruuje posloupnost aproximací řešení $\mathbf{x}^k$ tak, aby v limitě výsledná posoupnost byla řešením dané soustavy. Podmínka, aby limitou iteračního procesu bylo řešení dané soustavy se nazývá *podmínkou konvergence*. Během iteračního procesu tedy dochází ke zpřesňování řešení a každý krok iterace metody poskytuje nějaký odhad řešení. To jak rychle dochází k zpřesňování se značí rychlost konvergence. Pro každou z iteračních metod je třeba podmínky konvergence specifikovat zvlášt a hraje v tom roli i vlastnost matice soustavy, typicky její vlastní čísla.

Technika jak zlepšit konvergenční proces metody se nazývá předpodmiňování a v principu jde o převedení dané soustavy na jinou, modifikovanou, která má lepší vlastnosti z pohledu použité metody pro řešení.

CÍLE KAPITOLY

Pochopení základních principů metod :

- Základní pojmy iteračních metod: *lineární schéma*, *stacionarita*, *konzistence*, *konvergence* - viz zde, v kapitola třetí.
- Jacobiova, Gaussova-Seidelova metoda, SOR - viz zde, v kapitola 3.2. a zde.
- gradientní metoda a metoda sdružených gradientů - viz zde, v kapitola 3.3.
- princip předpodmiňování - viz zde, v kapitola 3.3.3

KLÍČOVÁ SLOVA

iterační metody pro soustavy l. rovnic, vlastnosti it. metod, předpodmiňování

ÚKOLY

1. Prostudujte implementaci metod v jazyce C/C++ viz zde.
2. Implementujte všechny výše uvedené metody pro řešení lineárních rovnic a jejich soustav. Věnujte se i podmínkám konvergence a hlavně rychlosti konvergence pro různě velké problémy ($n = 10, \ldots, 1000$). Tedy kromě toho, že metody naprogramujete, budete generovat soustavy rovnic a porovnávat čas běhu.
3. Seznamte se metodou scipy.linalg.solve().

OTÁZKY

1. Jaký vliv mají vlastní čísla matice soustavy na metody: Jacobiova, Gaussova-Seidelova metoda
2. Jaké jsou podmínky konvergence metod (proveďte diskusi proč to funguje) ?
3. Co to je předpodmiňování ?
4. Co to je číslo podmíněnosti matice a jak se spočítá ?

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat princip daných metod a jejich výhody a nevýhody a **možnosti použití**.
- Znát možnosti jejich elementární implementace.

ODKAZY NA LITERATURU

- LIMPOUCH, Jiří, Numerické metody lineární algebry, [online] zde
- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde, zejména kapitola 1.
- VONDRÁK, Vít a Lukáš POSPÍŠIL. Numerické metody 1. 2011. [online]: zde.

MÍSTO PRO VAŠE POZNÁMKY

4 Vlastní čísla a jejich výpočet

Tato kapitola odpovídá bodu 6 a 7 sylabu předmětu.

V předchozím textu jsme viděli, že vlastní čísla matice soustavy jsou důležitá pro konvergenci iteračních metod pro řešení dané soustavy. Další důležitou oblastí kde se setkáte s vlastními čísly matic je metoda PCA (Principal Component Analysis) při zpracování dat ve strojovém učení, viz další kapitola, nebo pokud je mi známo tak např. Google pagerank používá vlastní čísla pro určení bodového hodnocení stránky. Spolu s vlastnímy čísly je často potřebná též znalost vlastních vektorů.

Získávání vlastních čísel pomocí charakteristické rovnice jsou v případě velkých matic nevhodné, neboť vedou na řešení polynomiálních rovnic vysokého stupně. Zaměříme se proto na postup výpočtu jednoho vl.čísla - *částečný problém vlastních čísel* a všech vlastních čísel.

Pokud si nepamatujete, co přesně vlastní čísla jsou, tak kapitola 23 vám to připomene.

CÍLE KAPITOLY

- odhady vlastních čísel matice - Gershgorinovy kruhy, viz kapitola 4.1.1.
- výpočet vlastních čísel z charakteristického polynomu a Krylovova metoda - viz kapitola 4.1.
- částečný problém vlastních čísel - výpočet dominantního vl. čísla, viz kapitola 4.2.
- úplný problém vlastního čísla - QR iterace, Givensova rotace a Householderova transformace, viz kapitola 4.3.

KLÍČOVÁ SLOVA

vlastní čísla, vlastní vektory, odhady vl. čísel, Krylovova metoda, QR iterace,spektrum matice

ÚKOLY

1. Napište program pro odhad vlastních čísel matice pomocí Gershgorinových kruhů.
2. Implementujte Krylovovu metodu, pro řešení char. polynomu použijte numpy.roots.
3. Implementujte mocninnou metodu a jednu z metod pro úplný problém vlastních čísel.
4. Seznamte se s použitím metody scipy.linalg.eig.

OTÁZKY

1. Ukažte pomocí Geshgorinovy věty, že ostře diagonálně dominantní matice má kladná vlastní čísla.
2. Vysvětlete Krylovovu metodu a popište v čem spočívá její výhoda ?
3. Vysvětlete rozdíl mezi Givensovou rotací a Householderova transformací ?

OTÁZKY K ZAMYŠLENÍ

1. Jaký je problém při použití QR algoritmu pomocí Gramova-Schmidtova procesu?

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat princip daných metod a jejich výhody a nevýhody a **možnosti použití**.
- Znát možnosti jejich elementární implementace.
- Používat metodu scipy.linalg.eig().

ODKAZY NA LITERATURU

- LIMPOUCH, Jiří, Numerické metody lineární algebry, [online] zde.
- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde.
- VONDRÁK, Vít a Lukáš POSPÍŠIL. Numerické metody 1. 2011. [online]: zde.

MÍSTO PRO VAŠE POZNÁMKY

5 SVD rozklad

V předchozím textu jste se seznámili s použitím rozkladů na řešení lineárních rovnic. V předcházející kapitole jsme se naučili počítat vlastní čísla a vektory matice, jejich aplikací je možné získat další tzv. *spektrální rozklad* matice. Už v druhé kapitole jsme viděli, že ne vždy je možné rozklad dané (čtvercové) matice jednoznačně sestrojit. Dále i podmínka čtvercovosti matice může být omezující. Existuje však jeden rozklad matice A, tzv. *singulární rozklad* (SVD - Singular Value Decomposition), který můžete sestrojit pro každou reálnou matici.

Obdobně z lineární algebry víme, že pokud čtvercová matice není regulární, tak nelze sestrojit inverzní matici. Podobně existuje speciální typ inverze matice, tzv. *pseudoinverze* matice (Moore–Penroseovou pseudoinverze). Tuto pseudoinverzi je možné sestrojit pomocí SVD.

Nyní se budeme muset trochu ponořit i do lineární algebry.

CÍLE KAPITOLY

- spektrální rozklad matice, viz kapitola 24.
- singulární rozklad matice (SVD), viz kapitola 25.6 (Doporučuji však přečíst celou kapitolu).
- konstrukce SVD a různé použití SVD - špatně podmíněné soustavy, soustavy s žádným, nebo nekonečně mnoha řešeními, viz kapitola 2.6.
- aplikace SVD pro metdou nejmenších čtverců (regrese) -viz kapitola 15.4.
- pseudoinverze matice, konstrukce a aplikace na řešení soustavy rovnic, viz kapitola 25.7.

KLÍČOVÁ SLOVA

spektrální rozklad, singulární rozklad, pseudoinverze

ÚKOLY

1. Seznamte se s metodou scipy.linalg.svd().
2. Vyberte si některý z dostupných datasetů, viz zde a pomocí SVD proveďte lineární regresi.
3. Pomocí SVD implementujte řešič lineárních rovnic ve smyslu MNČ.
4. Pomocí metod výpočtu vlastních čísel (SVD) implementujte metodu PCA, viz např. zde.

OTÁZKY

1. Proč je lepší nepoužívat automaticky SVD pro řešení soustavy rovnic ?
2. Jaký je rozdíl mezi spektrálním a singulárním rozkladem ?
3. Jaká je geometrická interpretace SVD ?
4. jaká je souvislost SVD a čísla podmíněnosti matice?

SHRNUTÍ

Po prostudováni byste měli:

- Porozumět SVD rozkladu a možnostem jeho použití.
- Používat metodu scipy.linalg.svd().

ODKAZY NA LITERATURU

- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde.
- VONDRÁK, Vít a Lukáš POSPÍŠIL. Lineární Algebra. 2012. [online]: zde.

6 Iterační metody řešení nelineárních rovnic a jejich soustav

Hledání kořenů rovnice $f(x) = 0$ je častá úloha matematické analýzy a technických aplikací, pro kterou existuje celá řada názorných metod. V případě, že funkce $f(x)$ je polynom, bývá výhodné použít speciální metody, viz další kapitola. Některé z níže probraných metod je možné rozšířit na úlohu řešení soustavy nelineárních rovnic $\mathbf{F}(\mathbf{x}) = 0$, k tomu je však nutné mít příslušný aparát. Obecně se používá iteračních schémat, kdy polohu kořene buď omezujeme na nějakém intervalu (v 1D), nebo přímo konstrujeme posloupnost x^k, která konverguje k řešení. Abychom mohli metody použít, musí nás zajímat nejen jejich princip, ale hlavně *podmínky konvergence*, které jsou dány funkcí f. Dalším kritériem je rychlost konvergence, neboli *řád konvergence*. Obecně platí to, že neexistuje nějaká vždy fungující a přesto dostatečně rychlá metoda, spíše si musíme vybrat. Dalším kritériem je to, zda chceme/můžeme používat derivace dané funkce.

CÍLE KAPITOLY

- separace kořenů v 1D, viz kapitola 2.1.
- základní metody pro $f(x) = 0$ - bisekce a regula falsi a podmínky jejich konvergence, viz kapitola 2.2 a zde.
- Newtonova metoda a metoda prosté iterace pro $f(x) = 0$ a podmínky jejich konvergence, viz kapitola 2.3. a 2.4..
- Banachova věta o pevném bodu, řád konvergence, viz např. zde (projděte si i udávané příklady).
- Newtonova metoda a její varinty pro $\mathbf{F}(\mathbf{x}) = 0$, viz zde a kapitola 2.4.3

KLÍČOVÁ SLOVA

Newtonova metoda, metoda prosté iterace

ÚKOLY

1. Implementujte všechny výše uvedené metody pro 1D úlohu v Pythonu.
2. Řešte následující úlohy (volte přesnost $\epsilon < 1e^{-3}$), použijte všechny metody a diskutujte konvergenci:
   - $x + \sin(x) = 0, x > 0$

- $x \arctan x = 1$
- $x + \ln x = 0$

3. Pomocí Newtonovy metody řešte soustavu rovnic, vyzkoušejte různé počáteční podmínky $\mathbf{x}^0 = [0.1, 0.1], [1, 1], [10, 10]$:

$$\begin{aligned} \exp(x^2 + y^2) - 1 &= 0 \\ \exp(x^2 - y^2) - 1 &= 0 \end{aligned} \tag{6.1}$$

4. Rozhodněte o konvergenci metody prosté iterace pro úlohu 6.1.

OTÁZKY

1. Definujte řád konvergence metody. K čemu slouží ?
2. Vyslovte Banachovu větu o pevném bodě ?
3. Jak velká je chyba v $k-$tém kroku pomocí metody bisekce ?

OTÁZKY K ZAMYŠLENÍ

1. Jak je možné použít Banachovu větu o pevném bodě na konvergenci Newtonovy metody?
2. Zamyslete se na kombinací použití více metod zároveň pro zlepšení odhadu kořene.
3. Diskutujte vliv počátečního bodu na konvergenci Newtonovy metody v úloze 6.1.

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat principy fungování uvedených metod.
- Provést základní implementaci.
- Rozhodnout o vhodnosti použití probraných metod na danou úlohu.

ODKAZY NA LITERATURU

- LIMPOUCH, Jiří, Numerické metody, [online] zde.
- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde.
- VONDRÁK, Vít a Lukáš POSPÍŠIL. Numerické metody 1. 2011. [online]: zde.

7 Hledání kořenů polynomů

Při hlednání kořenů polynomů můžeme využít již dříve uvedených metod, nebo metod speciálních, které využívají některé specifické vlastnosti polynomů. Rozlišujeme, zda hledáme všechny kořeny daného polynomu, či kořeny mající nějakou speciální vlastnost, např. kořen největší v absolutní hodnotě (dominantní). Obecný postup hledání kořenů polynomu je následující:

1. Určení počtu a přibližné polohy jednotlivých kořenů polynomu $P(x)$.
2. Určení polohy konkrétního kořene x_i^*.
3. Vydělení původního polynomu $P(x)$ lineárním faktorem $(x - x_i^*)$, nebo případně kvadratickým faktorem, pokud máme k dispozici dva kořeny.
4. Opakujeme od bodu 1 s nově vzniklým polynomem.

Poznamenejme, že je nutné mít co nejpřesnější odhad kořene x_i^*, jinak polynom vzniklý dělením má jiné kořeny.

CÍLE KAPITOLY

- Zopakování základních vlastností polynomů, nalezenete zde a případně zde.
- Hornerovo schéma pro vyhodnocování a dělení polynomů, viz např. zde.
- Metody odhadu počtu kořenů a jejich vlastností naleznete zde. Detailněji:
    1. Descartovo znaménkové pravidlo zde.
    2. Budanova-Fourierova věta zde.
    3. Cauchyovo ohraničení kořenů zde.
    4. Sturmova věta zde.
- Müllerova metoda a Laguerrova metoda jsou uvedeny zde a kapitola 9.5.
- Bernoulliova metoda a další jsou zde.

KLÍČOVÁ SLOVA

kořeny polynomu, odhad kořenů, speciální metody

ÚKOLY

1. Implementujte metody vyhodnocení polynomů a dělení polynumů pomocí Hornerova schématu
2. Implementujte algoritmus pro odhad počtu kořenů pomocí uvedených vět.
3. Implementujte nejméně dvě metody pro výpočet kořenů polynomu.
4. Seznamte se s balíkem *numpy.polynomial* zde.

OTÁZKY

1. Jaká je časová náročnost Hornerova schématu ? Odůvodněte.
2. Formulujte věty nutné pro odhad kořenů.

OTÁZKY K ZAMYŠLENÍ

1. Diskutujte možnost rozšíření metod na hledání komplexních kořenů .
2. Jak se při redukci polynomů projeví komplexní kořen ?

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat principy fungování uvedených metod.
- Provést základní implementaci.
- Rozhodnout o vhodnosti použití probraných metod na danou úlohu.

ODKAZY NA LITERATURU

- LIMPOUCH, Jiří, Numerické metody, [online] zde.
- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde.

8 Numerická kvadratura

Výpočet určitého integrálu je úloha, která má časté praktické aplikace. Postup běžně používaný v matematice spočíval v nalezení *primitivní* funkce $F(x)$ k dané funkci $f(x)$ a použití Newtonovy-Leibnitzovy formule:

$$\int_a^b f(x)dx = F(b) - F(a). \tag{8.1}$$

Bohužel pro celou řadu důležitých funkcí, neumíme najít primitivní funkce, nebo jsou dané primitivní funkce výpočetně složité a vyplatí se tedy hodnotu určitého integrálu aproximovat pomocí numerických metod.

Jeden ze základních postupů spočívá v nahrazení integrované funkce pomocí interpolačního polynomu $L_n(x)$ (pokud si je napamatujete z bakalářského studia, tak viz kapitola 10). Ten zintegrujeme a dostaneme tzv. *Newtonovy-Cotesovy vzorce*. Jako vylepšení, minimálně co se týče přesnosti, lze považovat tzv. *Gaussovu kvadraturu*, kde se za uzlové body berou kořeny speciálních ortogonálních polynomů. Tyto vzorce můžeme též aplikovat tak, že původní interval $< a, b >$ rozdělíme na dílčí subintervaly a na každém z nich použijeme daný kvadraturní vzorec, takto získáme tzv. *složené kvadraturní vzorce*.

Další technikou používanou pro zpřesnění jsou *adaptivní techniky*, kde dochází ke zjemňování intervalů a tím zpřesnění výpočtu na základě odhadu chyby, např. *metoda poloviční kroku*.

Pro další zpřesnění výpočtu odhadu integrálů je možné používat jeho odhady pro různě velké kroky h (velikost dílčích intervalů) a tyto odhady zkombinovat do přesnější hodnoty. Toto dělá tzv. *Richardsonova extrapolace*, resp. *Rombergova kvadratura*.

Pro použití těchto vzorců je důležité, aby byla funkce $f(x)$ rozumně definována v uzlových bodech. Vzpomeňme si, že určitý integrál (konečný) mohou mít i funkce, které nabývají uvnitř intervalu $< a, b >$ hodnot $+\infty, -\infty$ (*singulární*). Nebo některá z mezí leží v nekonečnu. Je tedy nutné se věnovat i těmto úlohám.

Další důležitá aplikace je výpočet vícerozměrných integrálů, kde je možné buď získané vzorce zobecnit do více dimenzí. Což se děje za pomocí *Fubiniovy věty* a příslušného kvadraturního vzorce a numerická aproximace vícerozměrného integrálu se pak převádí na "posloupnost" jednorozměrných kvadraturních vzorců. Tento postup je však při větším počtu dimenzí velmi výpočetně náročný, proto je možné použít stochastické metody typu *Monte Carlo*.

CÍLE KAPITOLY

- Zopakování základních principů numerické kvadratury Newtonovy-Cotesovy vzorce a složené vzorce zde.
- Gaussova kvadratura (kapitola 4.5) zde. Problematika ortogonálních polynomů je probírána v kapitole 10.

- Richardsonova extrapolace obecný princip a její aplikace na integraci (Rombergova kvadratura) kapitola 4.3.
- Princip adaptivní integrace, popsán je např. paragraf 3 zde.
- Integrály se singularitou zde a kapitola 4.4 zde
- Vícerozměrné integrály 4.6 zde
- Principy a aplikace metody Monte Carlo (důkladně 7.6-7.8 zde) a zde kapitola 5.

KLÍČOVÁ SLOVA

kvadraturní vzorce, Gaussova kvadratura, Rombergova kvadratura, metoda Monte Carlo

ÚKOLY

1. Implementujte Gaussovu kvadraturu, vstupem je integrovaná funkce $f(x)$ a interval $< a, b >$
2. Implementujte Rombergovu kvadraturu, vstupem je integrovaná funkce $f(x)$ a interval $< a, b >$
3. Pomocí vašich implementací kvadraturních vzorců řešte (musíte substituovat, nebo vhodně rozdělit):
   - $\int_{-1}^{1} \frac{dx}{\sqrt{|x|}}$
   - $\int_{0}^{1} \frac{dx}{\sqrt[3]{x}}$
   - $\int_{0}^{\infty} e^{-x} dx$
4. Funkce $\Gamma(x)$ je definována $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$, určete $\Gamma(1), \Gamma(2), \Gamma(3)$.
5. Implementujte metodu Monte-Carlo a ověřte její funkčnost.
6. Odvoďte a implementujte schéma pro integraci na obdélníkové oblasti pomocí Gaussovy kvadratury.
7. Seznamte se s balíkem *scipy.integrate.quadrature* zde.

OTÁZKY

1. Proč je Gaussova kvadratura při stejném počtu vyhodnocení funkce přesnější než N-C vzorce?
2. Vysvětlete co je principem Rombergovy kvadratury.
3. Vysvětlete co je principem adaptivních metod. Jak se odhaduje chyba výpočtu ?
4. Vysvětlete princip metody Monte-Carlo.
5. Vysvětlete jak se chová chyba v případě metody Monte Carlo.

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat principy fungování uvedených metod.
- Provést základní implementaci.
- Rozhodnout o vhodnosti použití probraných metod na danou úlohu.
- Prakticky používat sw. knihovny v Pythonu.

ODKAZY NA LITERATURU

- LIMPOUCH, Jiří, Numerické metody, [online] zde.
- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde.
- VONDRÁK, Vít a Lukáš POSPÍŠIL. Numerické metody 1. 2011. [online] zde.

9 Řešení ODR a jejich soustav

V dalším uvažujme obyčejnou diferenciální rovnici prvního řádu

$$y'(x) = f(x, y(x)), \tag{9.1}$$

s počáteční podmínkou $y(x_0) = y_0$. Předpokládejme, že je splněna podmínka spojitosti funkce f ve všech proměnných, tento předpoklad souvisí s jednozačností řešitelnosti rovnice 9.1, viz *Cauchyho-Peanova věta* a *Picardova–Lindelöfova věta.* Analytickým řešením se rozumí funkce $y(x)$, která vyhovuje rovnici 9.1 spolu s počáteční podmínkou (p.p.), zatímco numerickým řešením rozumíme nalezení hodnot funkce $y(x_i)$ v uzlech diskretizace x_i. Při numerickém řešení ODR lze postupovat v podstatě dvěma směry. Prvním směrem je to, že v rovnici 9.1 nahradíme levou stranu pomocí některého ze schémat pro náhradu derivace. Dostáváme tak např. *Eulerovy metody*, *Rungeho-Kuttovy metody* (RK). Druhým směrem je možnost převedení rovnice 9.1 na tvar

$$y(x) = y_0 + \int_{x_0}^{x} f(t, y(t))dt, \tag{9.2}$$

a integrál na pravé straně nahradíme pomocí kvadraturních vzorců, tímto dostaneme tzv. *Adams-Bashforthovy metody* a *Adams-Moultonovy metody.*

V obou výše uvedených technikách, můžeme rozlišovat, zda po nahrazení levé, či pravé strany, hodnotu $y(x_i)$ *explicitně* spočteme z hodnot předcházejících $y(x_{i-1}), y(x_{i-2}), \ldots$. Nebo dostáváme numerické schéma (rovnici), kde hodnota $y(x_i)$ je na obou stranách rovnice $y(x_i) = g(y(x_i), y(x_{i-1}), y(x_{i-2}), \ldots)$. Schéma v druhém případě se pak nazývá *implicitní* a danou rovnici je nutné numericky řešit pro nalezení $y(x_i)$. Případně je možné obě schémata zkombinovat a získat tak metodu typu *prediktor-korektor.*

Dále rozlišujeme, zda hodnota $y(x_i)$ závisí na bezprostředně předcházející hodnotě $y(x_{i-1})$, takové schéma nazýváme *jednokrokové*, či na více hodnotách, v tomto případě jej nazýváme *vícekrokové.*

V případě, že potřebujeme řešit ODR vyššího řádu něž jedna, tak je možné je převést na *soustavu ODR* a tu řešit výše uvedenými metodami.

CÍLE KAPITOLY

- Jednokrokové explicitní metody (Eulerova metoda a RK metody) a jejich chyby kapitola 8.1 a zde.
- Bulirsch-Stoerova metoda kapitola 16.4. zde a zde.
- Vícekrokové metody a metoda prediktor-korektor kapitola 8.2 a zde.
- Soustavy diferenciálních rovnic kapitola 8.3
- *Stiff systémy* kapitola 16.6. zde a zde.

KLÍČOVÁ SLOVA

Eulerova metoda, RK metody, vícekrokové metody, soustavy ODR

ÚKOLY

1. Implementujte Eulerovu metodu (dopřednou) a RK2/RK4 metodu a metodu prediktor-korektor. Řešte pomocí nich rovnici $y'(x) = y + xy^2$, s p.p. $y(0) = 1$ (přesné řešení je $y(x) = \frac{1}{1-x}$). Zkoumejte vliv velikosti kroku h na chybu.
2. Implementujte model sluneční soustavy uvedený zde
3. Seznamte se s balíkem scipy.integrate.odeint.
4. Řešte pomocí výše uvedené knihovny tzv. Lotkův-Volterrův model zde.
5. Řešte pomocí výše uvedené knihovny epidemiologické modely (SI, SIR, SIRS) zde.

OTÁZKY

1. Odvoďte metodu Eulerovu metodu a metodu RK2 a vysvětlete jejich princip.
2. Co to je lokální a globální diskretizační chyba a řád metody ?
3. Vysvětlete rozdíl mezi explicitním a implicitním schématem ? Jaké jsou vlastnosti obou typů metod?
4. Vysvětlete použití metod prediktor-korektor?
5. Vysvětlete co jsou to stiff systémy.

OTÁZKY

1. Proč není vhodné používat metody nízkého řádu s velmi nízkým krokem *h*? Uveďte více důvodů.
2. Co to je *A-stabilita* ?

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat principy fungování uvedených metod.
- Provést základní implementaci.
- Rozhodnout o vhodnosti použití probraných metod na danou úlohu.
- Prakticky používat sw. knihovny v Pythonu.

ODKAZY NA LITERATURU

- LIMPOUCH, Jiří, Numerické metody, [online] zde.
- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde.
- VONDRÁK, Vít a Lukáš POSPÍŠIL. Numerické metody 1. 2011. [online] zde.

10 Aproximace a interpolace funkcí

.

Mějme dány dvojice hodnot x_i, y_i, pak *úlohou interpolace* rozumíme sestrojení takové fukce $f(x)$, pro kterou jsou splněny *podmínky interpolace* $f(x_i) = y_i, i = 0, \ldots, n$. V případě, že nám postačuje, aby $f(x_i)$ mělo pouze přibližnou hodnotu y_i (tedy $f(x_i) \approx y_i, i = 0, \ldots, n$), pak takovou úlohu budeme nazývat *úlohou aproximace*. Jednotlivé metody se liší v tom, jak je kvalita aproximace měřena, viz např. *metoda nejmenších čtverců*.

Obecně se pro náhradu funkce používají polynomy, *trigonometrické polynomy*, *racionální lomené funkce*, *spline funkce*[1].Interpolace, či aproximace se používá, chceme-li z nějakého důvodu nahradit původní funkci za jinou, která má "lepší" vlastnosti. Příkladem lepších vlastností může být např. náhrada původní funkce pomocí polynomu, který se nám snadno integruje.

CÍLE KAPITOLY

- Opakování - Lagrangeova a Newtonova interpolace kapitola 5.1-5.3.
- Opakování - aproximace a metoda nejmenších čtverců - kapitola 6-6.1..
- Problematika ortogonálních funkcí a polynomů a jejich konstrukce kapitola 6.2..
- Princip interpolace pomocí Čebyševových polynomů je uveden zde, detailněji včetně různých jejich dalších vlastností zde.
- Interpolace pomocí trigonometrických polynomů kapitola 5.4 a racionální lomenou funkcí zde .
- Interpolace pomocí spline funkcí kapitola 5.5 - 5.6.
- Padeova aproximace kapitola 5.12.
- Čebyševovská aproximace a princip Remezova algoritmu zde.

KLÍČOVÁ SLOVA

interpolace, aproximace, spline funkce,

ÚKOLY

1. Implementujte program pro interpolaci, kde vstupem je soustava dvojic bodů (x_i, y_i) a body, ve kterých nás interpolace zajímá. Program provede interpolaci pomocí interpolačního polynomu a pomocí kubické spline funkce. Výstupem bude vyhodnocený polynom v zadaných bodech. V případě polynomu použijte Aitkenovo-Nevillovo schéma pro vyhodnocení polynomu v bodě zde.

[1] Po částech spojité polynomy

2. Ověřte chování Vašeho programu na funkci $f(x) = \frac{1}{5x^2+1}$ (*Rungova úloha*). Volte ekvidistantní uzly na intervalu $< -1, 1 >$ a uzly které jsou kořeny Čebyševových polynomů. Diskutujte i vliv počtu uzlů interpolace.

3. Implementujte Bulirschův-Stoerův algoritmus pro interpolaci pomocí racionální lomené funkce kapitola 3.2.

4. Na datasetu s prodeji zde, proveďte extrapolaci pro odhad prodejů za poslední tři měsíce a porovnejte s realitou.

5. Seznamte se s balíkem scipy.interpolate.

6. Pomocí metody nejmenších čtverců za použití trigonometrických polynomů proložte následující data zde (cca 10 let denního měření teploty). Případně můžete použít funkci z scipy.optimize.curve_fit.

OTÁZKY

1. Na čem závisí chyba při Lagrangeově interpolaci a jak je jí možné zmenšit?

2. Porovnejte použití Lagrangeovy interpolace a kubického spline pro velký počet uzlových bodů.

3. V čem spočívá výhoda použití racionálních lomených funkcí před polynomy?

4. Jaké znáte způsoby měření chyby při aproximaci ?

5. Je možné pomocí metody nejmenších čtverců i interpolovat ?

SHRNUTÍ

Po prostudováni byste měli být schopni:

- Chápat principy fungování uvedených metod.
- Provést základní implementaci.
- Rozhodnout o vhodnosti použití probraných metod na danou úlohu.
- Prakticky používat sw. knihovny v Pythonu.

ODKAZY NA LITERATURU

- LIMPOUCH, Jiří, Numerické metody, [online] zde.
- PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde.
- VONDRÁK, Vít a Lukáš POSPÍŠIL. Numerické metody 1. 2011. [online] zde.

Literatura

[1] LIMPOUCH, Jiří, Numerické metody, [online] zde.

[2] PRESS, William H. Numerical recipes: the art of scientific computing. 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88407-5. [online] zde

[3] QUARTERONI, Alfio, Riccardo SACCO a Fausto SALERI. Numerical mathematics. New York: Springer, c2000. ISBN 0-387-98959-5.

[4] VONDRÁK, Vít a Lukáš POSPÍŠIL. Numerické metody 1. 2011. [online] zde.