Počítačové metody v teoretické fyzice I

Numerika I — skripta z přednáškových poznámek

Obsah

1 Reprezentace čísel. Zaokrouhlovací chyba. Stabilita. Podmíněnost **7**
- 1.1 Reprezentace čísel v počítači . . . 7
- 1.2 Zaokrouhlovací chyba . . . 8
  - 1.2.1 Vliv zaokrouhlovacích chyb na výsledek . . . 8
  - 1.2.2 Neplatnost některých axiomů reálné aritmetiky . . . 8
  - 1.2.3 Odečítání blízkých čísel (cancellation problem) . . . 9
  - 1.2.4 Součet řady „velkých“ čísel, jehož výsledek je „malý“ . . . 9
- 1.3 Stabilita algoritmu . . . 10
- 1.4 Podmíněnost úlohy . . . 10
- 1.5 Normy na vektorových prostorech a normy matic . . . 12
  - 1.5.1 Maticová norma indukovaná vektorovou normou . . . 14
  - 1.5.2 Obecné maticové normy . . . 17

2 Řešení nelineárních rovnic. Urychlování konvergence. Hledání kořenů **19**
- 2.1 Řešení nelineárních rovnic pomocí iterací . . . 19
- 2.2 Rychlost konvergence a řád iterační metody . . . 20
- 2.3 Urychlování konvergence lineárních metod . . . 21
  - 2.3.1 Aitkenova Δ^2 metoda . . . 21
  - 2.3.2 Archimedův výpočet čísla π . . . 21
  - 2.3.3 Aitkenova–Steffensenova metoda . . . 24
- 2.4 Hledání kořenů algebraické rovnice $g(x) = 0$ . . . 25
  - 2.4.1 Vždy konvergentní metody . . . 25
  - 2.4.2 Iterační metody vyššího řádu . . . 27
- 2.5 Řešení rovnic pro více proměnných . . . 30

3 Aproximace funkcí. Polynomiální interpolace. Splajny **33**
- 3.1 Aproximace funkcí v počítači . . . 33
  - 3.1.1 Interpolace . . . 33
  - 3.1.2 Metoda nejmenších čtverců . . . 33
  - 3.1.3 Čebyševova aproximace . . . 34
  - 3.1.4 Volba normy a role polynomů . . . 34
- 3.2 Polynomiální interpolace . . . 34
- 3.3 Aitkenovo–Nevillovo schéma . . . 35
- 3.4 Chyba polynomiální interpolace a Čebyševovy uzly . . . 35
  - 3.4.1 Proč zrovna kořeny Čebyševova polynomu? . . . 36
- 3.5 Hermiteova interpolace . . . 37
- 3.6 Interpolace pomocí splajnů . . . 37
  - 3.6.1 Interpolační kubický splajn . . . 38
  - 3.6.2 Přirozený kubický splajn procházející body (x_i, f_i) . . . 38

4 Ortogonální polynomy **41**
4.1 Ortogonální polynomy . . . 41
4.2 Rekurentní vztah . . . 41
4.3 Christoffelova–Darbouxova identita . . . 42
4.4 Kořeny ortogonálních polynomů . . . 43

5 Numerická derivace **45**
5.1 Reprezentace funkce a výpočet derivace . . . 45
5.2 Derivace interpolačního polynomu . . . 45
5.2.1 Přímá derivace bázových polynomů . . . 46
5.2.2 Řešení soustavy lineárních rovnic z Taylorových rozvojů . . . 46
5.2.3 Fornbergovy rekurentní vztahy . . . 46
5.2.4 Rovnoměrné sítě . . . 48
5.2.5 Richardsonova extrapolace . . . 48
5.3 Optimální volba kroku h . . . 48
5.4 Vzorce pro derivaci a integraci pomocí operátorů na polynomech . . . 49
5.4.1 Derivace . . . 50
5.4.2 Integrace . . . 51

6 Numerická integrace **53**
6.1 Kvadraturní vzorce . . . 53
6.2 Newtonovy–Cotesovy vzorce . . . 54
6.2.1 Odvození vzorců pomocí operátorů na polynomech . . . 54
6.2.2 Lichoběžníkové a Simpsonovo pravidlo . . . 55
6.2.3 Složené kvadraturní vzorce . . . 55
6.3 Eulerův–Maclaurinův sumační vzorec . . . 56
6.4 Využití Eulerova–Maclaurinova vzorce . . . 57
6.4.1 Odvození složených Newtonových–Cotesových vzorců vyššího řádu . . . 57
6.4.2 Rombergův algoritmus . . . 58
6.5 Gaussova kvadratura . . . 59
6.5.1 Odvození . . . 59
6.5.2 Výpočet kořenů x_j a vah w_j . . . 60
6.5.3 Chyba Gaussovy kvadratury . . . 60
6.5.4 Základní Gaussovy kvadratury . . . 60
6.6 Integrály se singularitami . . . 60
6.7 Poznámky k vícerozměrným integrálům . . . 61
6.8 Poznámky k singulárním integrálům . . . 61

7 Numerické řešení ODR – Základní pojmy. Rungeovy-Kuttovy metody **63**
7.1 Počáteční a okrajová úloha . . . 63
7.1.1 Matematická formulace počáteční úlohy pro ODR . . . 63
7.2 Existence a jednoznačnost řešení . . . 64
7.2.1 Problém špatné podmíněnosti . . . 64
7.3 Numerické řešení ODR — základní pojmy . . . 65
7.3.1 Jednokrokové a vícekrokové metody . . . 66
7.3.2 Explicitní a implicitní metody . . . 66
7.4 Lokální a globální diskretizační chyba . . . 66
7.5 Jednokrokové metody vyšších řádů . . . 67
7.5.1 Metoda Taylorova rozvoje . . . 68
7.6 Rungeovy–Kuttovy metody . . . 68
7.6.1 Motivace — metoda RK 2. řádu . . . 68
7.6.2 Obecný tvar a Butcherova tabulka . . . 69

7.6.3 Podmínky na řád metody . . . 70

8 Numerické řešení ODR – Lineární vícekrokové metody. Stabilita a konvergence 73
8.1 Lineární vícekrokové metody . . . 73
8.1.1 Standardní vícekrokové metody . . . 73
8.1.2 Určení koeficientů pomocí operátorového počtu . . . 74
8.2 Stabilita a konvergence . . . 75
8.3 Konvergence explicitních jednokrokových metod . . . 75
8.3.1 Explicitní Eulerova metoda je konvergentní . . . 76
8.3.2 Explicitní Rungeovy–Kuttovy metody jsou konvergentní . . . 77
8.4 Řešení homogenních lineárních diferenčních rovnic . . . 78
8.5 Stabilita a konvergence lineárních vícekrokových metod . . . 79

9 Numerické řešení ODR – Absolutní stabilita. Systémy se silným tlumením. Adaptivní metody 81
9.1 Absolutní stabilita . . . 81
9.1.1 Oblast stability lineárních vícekrokových metod . . . 82
9.1.2 Hranice oblasti stability . . . 83
9.1.3 Jednokrokové metody typu Runge–Kutta . . . 84
9.2 Stabilita pro systém ODR . . . 85
9.2.1 Volba kroku . . . 87
9.3 ODR se silným tlumením (stiff ODE) . . . 87
9.4 Adaptivní metody . . . 88
9.4.1 Tentýž postup s krokem h a $2h$ . . . 89
9.4.2 Vnořené Rungeovy–Kuttovy metody . . . 89
9.4.3 Bulirschova–Stoerova metoda . . . 89

10 Numerická lineární algebra. Úvod. LU rozklad 91
10.1 Numerická lineární algebra . . . 91
10.1.1 Základní rozklady . . . 91
10.1.2 Rychlost a stabilita . . . 92
10.1.3 Doporučení pro praxi . . . 92
10.2 Gaussova eliminace a LU dekompozice (rozklad) . . . 93
10.2.1 Pivotizace . . . 95
10.2.2 Řádková pivotizace a LU rozklad . . . 95
10.2.3 Algoritmus LU rozkladu s řádkovou pivotizací . . . 96
10.2.4 Počet operací . . . 96
10.2.5 Řešení soustavy $Ax = b$ pomocí LU rozkladu . . . 97
10.3 Choleského rozklad . . . 97
10.4 Stabilita Gaussovy eliminace a LU rozkladu . . . 98

11 Numerická lineární algebra. QR a SVD rozklad 101
11.1 QR rozklad (faktorizace) . . . 101
11.1.1 Existence a jednoznačnost . . . 101
11.1.2 Použití na řešení soustavy $Ax = b$ . . . 102
11.1.3 Výpočet QR rozkladu . . . 102
11.1.4 Přímá konstrukce pomocí Gramovy–Schmidtovy ortogonalizace . . . 102
11.1.5 Modifikovaná Gramova–Schmidtova metoda . . . 103
11.1.6 Trojúhelníková versus ortogonální triangularizace . . . 103
11.1.7 Householderova triangularizace . . . 104
11.1.8 Givensovy rotace . . . 106

11.1.9 Stabilita Householderovy triangularizace . . . 106
11.2 Singulární rozklad matice (SVD) . . . 107
11.2.1 Základní vlastnosti SVD rozkladu . . . 108
11.2.2 Výpočet SVD rozkladu (nástin) . . . 109

12 Numerická lineární algebra. Řešení přeurčené soustavy 111
12.1 Řešení přeurčeného systému $Ax = b$ metodou nejmenších čtverců . . . 111
12.1.1 Normální rovnice . . . 112
12.1.2 QR rozklad . . . 113
12.1.3 SVD rozklad . . . 113
12.1.4 Porovnání metod . . . 113

13 Jacobiho metoda pro vlastní čísla reálné symetrické matice 115
13.1 Základní myšlenka . . . 115
13.2 Nulování mimodiagonálních prvků obecné matice . . . 115
13.3 Numericky stabilní formulace . . . 116
13.4 Konvergence a náročnost . . . 117
13.5 Vlastní vektory . . . 117

14 Rychlá Fourierova transformace a její využití 119
14.1 Fourierova transformace . . . 119
14.1.1 Definice a volba normalizace . . . 119
14.1.2 Parsevalova rovnost a konvoluce . . . 119
14.1.3 Vlastnosti Fourierovy transformace . . . 120
14.2 Diskrétní Fourierova transformace . . . 120
14.2.1 Vzorkování a Nyquistova kritická frekvence . . . 120
14.2.2 Problém aliasingu . . . 121
14.2.3 Odvození diskrétní Fourierovy transformace . . . 121
14.2.4 Inverzní diskrétní Fourierova transformace . . . 122
14.3 Rychlá Fourierova transformace (FFT) . . . 122
14.4 Výpočet Fourierovy transformace pomocí DFT (FFT) . . . 123
14.5 Časový vývoj v kvantové mechanice pomocí split-operator metody . . . 124

A Normy vektorů a matic 125
A.1 Normy na vektorových prostorech . . . 125
A.2 Maticová norma indukovaná vektorovými normami . . . 126
A.2.1 Příklad . . . 128
A.2.2 Obecné vztahy pro maticové normy . . . 128

Kapitola 1

Reprezentace čísel. Zaokrouhlovací chyba. Stabilita. Podmíněnost

1.1 Reprezentace čísel v počítači

Nejběžnějším způsobem reprezentace čísel v počítači je reprezentace pomocí *pohyblivé řádové čárky* (floating-point arithmetic), daná standardem IEEE. Řada čísel, která v desítkové soustavě zapíšeme snadno, přitom není reprezentována přesně. Například čísla $\pm 0{,}01$ přesně reprezentovat nelze, neboť

$$\begin{aligned} \pm 0{,}01 \text{ (DEC)} &= \pm 0{,}000\ 000\ 101\ 000\ 111\ 101\ldots \text{ (BIN)} \\ &= \pm \underbrace{1.01000111101\ \ldots}_{t \text{ cifer, určuje přesnost (precision)}} \times 2^{-7} \text{ (BIN)} \\ &= \pm\, m \times \beta^{e-t+1}. \end{aligned}$$

Zde m značí *mantisu* (significand), β je *základ* (base) – obvykle 2, někdy 10 nebo 16 – a e je *exponent*.

Poznámka 1.1.1. Anglický termín *precision* je třeba nezaměňovat s termínem *accuracy*.

Mantisa je celočíselná (ovšem může být někdy brána i jako desetinné číslo),

$$m = d_1 d_2 \cdots d_t, \tag{1.1}$$

kde d_i nabývají hodnot $0 \le d_i < \beta$, ovšem $d_1 \neq 0$.

Číslo 0,01 je ve 32-bitovém IEEE standardu, tedy v single precision, neboli v `REAL*4` (4 byty), reprezentováno takto:

$$0{,}01 \approx \underbrace{\boxed{0}}_{\substack{\text{znaménkový} \\ \text{bit } s\text{: } (-1)^s}} \quad \underbrace{\boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{0}\boxed{0}\ \boxed{0}}_{\substack{127+e \\ (8 \text{ bitů})}} \quad \underbrace{\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{0}\boxed{1}\boxed{1}\ \boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{1}\ \boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{0}}_{\substack{\text{mantisa } m \text{ bez první 1} \\ (23 \text{ bitů})}}$$

To odpovídá zhruba 7 platným místům. V double precision je na exponent vyhrazeno 11 bitů a na mantisu 52 bitů, což dává zhruba 16 platných míst v desítkové soustavě.

Rozsah exponentů je $e_{min} \le e \le e_{max}$, přičemž pro single precision je $e_{min} = -126$ a $e_{max} = 127$,

$$(2^{-126} \approx 1{,}18 \cdot 10^{-38},\ 2^{127} \approx 3{,}4 \cdot 10^{38}),$$

a pro double precision je $e_{min} = -1022$ a $e_{max} = 1023$,

$$(2^{-1022} \approx 2{,}2 \cdot 10^{-308},\ 2^{1023} \approx 1{,}8 \cdot 10^{308}).$$

Poznámka 1.1.2. Speciální exponenty $e = -127$ (samé nuly) a $e = 128$ (samé jedničky) se používají na tzv. subnormální čísla a též pro $\pm\infty$ a NaN.

1.2 Zaokrouhlovací chyba

Pokud reálné číslo $x \in \mathbb{R}$ leží v rozsahu systému F čísel s pohyblivou řádovou čárkou, pak

$$fl(x) = x\,(1+\delta)\ ,\ \text{kde}\ \ |\delta| < u = \frac{\varepsilon_M}{2}, \tag{1.2}$$

kde $fl(x)$ značí číslo x vyjádřené nejbližším číslem s pohyblivou řádovou čárkou, tj. $f(x) \in F$, a u je tzv. unit roundoff,

$$u = \tfrac{1}{2}\beta^{1-t},$$

zatímco ε_M je tzv. *strojové epsilon*. Maximální relativní chyba vyjádření libovolného čísla x v rozsahu F je tedy dána hodnotou u.

Je-li $fl(x) = x + \Delta x = x\,(1+\varepsilon_x)$, hovoříme o Δx jako o *zaokrouhlovací chybě* a o ε_x jako o *relativní zaokrouhlovací chybě*.

Poznámka 1.2.1. Přestože je reprezentace čísel pomocí pohyblivé řádové čárky zdaleka nejběžnější, není jediná. Existují i další systémy, jako např. pevná řádová čárka (fixed-point) nebo logaritmický číselný systém (logarithmic number system) a podobně (viz Wikipedia nebo Higham: Accuracy and Stability ...).

1.2.1 Vliv zaokrouhlovacích chyb na výsledek

Při výpočtech na počítači je nutné mít na paměti, že zaokrouhlovací chyba je všudypřítomná:

- čísla na vstupu jsou zaokrouhlena;
- výsledek libovolné aritmetické operace je zaokrouhlen.

Poznámka 1.2.2. Dobrý systém by měl mít implementovány tyto operace tak, že výsledek bude

$$fl\,(x\ op\ y) = (x\ op\ y)\,(1+\delta)\ ,\ |\delta| < u \tag{1.3}$$

pro $op = +, -, *, /$ a pro druhou odmocninu též.

Numerická matematika se z velké části zabývá právě vlivem zaokrouhlovací chyby na správnost výpočtu.

1.2.2 Neplatnost některých axiomů reálné aritmetiky

Pozor: některé axiomy reálné aritmetiky nejsou v systémech s pohyblivou řádovou čárkou splněny.

Například *výsledek závisí na pořadí sčítanců*, a to i tehdy, jsou-li všechny sčítance kladné.

Příklad 1.2.3. Uvažujme jednoduchou sumu (viz notebook `roundoff.errors.nb`)

$$S_n = 1 + \sum_{k=1}^{n} \frac{1}{k^2+k} = 1 + \sum_{k=1}^{n} \left(\frac{1}{k} - \frac{1}{k+1}\right) = 2 - \frac{1}{n+1}. \tag{1.4}$$

Sčítáme-li odpředu, objeví se zaokrouhlovací chyby, zvláště pro velká n. Sčítáme-li odzadu, tedy od nejmenších příspěvků po největší, zaokrouhlovací chyby se neprojeví.

Dále *rovnice nemusí mít jednoznačné řešení*, případně *nemusí mít žádné řešení*, i když v přesné aritmetice existuje jediné.

Příklad 1.2.4. Rovnice $1 + x = 1$ má řešení $x = 0$, ovšem v double precision bude tato rovnice splněna pro libovolné $x \lesssim 10^{-17}$, které patří do systému čísel double precision.

Příklad 1.2.5. Rovnice $f(x) = 0$ většinou nemá v daném systému čísel s pohyblivou řádovou čárkou žádné řešení. Lze hledat pouze řešení, pro které je $|f(x)|$ nejmenší, případně menší než jisté ε, když nám stačí určitá přesnost.

1.2.3 Odečítání blízkých čísel (cancellation problem)

Jde o odečítání blízkých, a to ne nutně velkých, čísel. Operace odečítání může vnést do numerických výpočtů značné chyby, jsou-li odečítaná čísla blízká.

Máme-li

$$fl\,(x_1) = x_1 + \Delta x_1 \qquad \text{a} \qquad fl\,(x_2) = x_2 + \Delta x_2,$$

pak

$$fl\,(x_1 \pm x_2) = x_1 \pm x_2 + \Delta x_1 \pm \Delta x_2 + \varepsilon\,(x_1 \pm x_2), \tag{1.5}$$

kde $\varepsilon \lesssim \varepsilon_M$, a relativní chyba je

$$\delta = \left| \frac{fl\,(x_1 \pm x_2) - (x_1 \pm x_2)}{x_1 \pm x_2} \right| \leq \frac{|\Delta x_1| + |\Delta x_2|}{|x_1 \pm x_2|} + |\varepsilon|. \tag{1.6}$$

Pokud je například $x_1 \approx 1$, $x_2 \approx 1$, $\Delta x_1 \approx 10^{-8}$ a $\Delta x_2 \approx 10^{-8}$ a v přesné aritmetice $x_1 - x_2 \approx 10^{-4}$, pak $\delta \approx 2 \cdot 10^{-4}$.

Často se s tímto problémem nedá mnoho dělat, ale někdy lze ze znalosti problému změnit výpočet tak, že se odečítání vyhneme.

Příklad 1.2.6. Výraz $\sqrt{x+1} - \sqrt{x}$ lze vyčíslit též pomocí

$$\frac{1}{\sqrt{x+1} + \sqrt{x}},$$

a pro velká x pak velká zaokrouhlovací chyba nevzniká.

Příklad 1.2.7. Řešení kvadratické rovnice $ax^2 + bx + c = 0$ spočtené pomocí

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1.7}$$

může být zatíženo chybou, pokud $b^2 \gg |4ac|$, neboť pak $\sqrt{b^2 - 4ac} \approx |b|$ a jeden z kořenů dostaneme odečítáním dvou blízkých čísel. Lze to obejít tak, že tento kořen spočteme jako $x_2 = \frac{c}{ax_1}$.

1.2.4 Součet řady „velkých“ čísel, jehož výsledek je „malý“

Tento jev se označuje jako *smearing problém*. Jde o skryté odečítání blízkých čísel, neboť blízké mohou být až mezivýsledky. Opět jej lze někdy obejít jiným přístupem k danému problému.

Příklad 1.2.8. Výpočet e^x pomocí Taylorovy řady

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots$$

je pro $x > 0$ víceméně v pořádku, jen pro větší x máme pomalou konvergenci. Ovšem pro $x < 0$ výpočet selže, neboť výsledek je malé číslo jako součet velkých čísel s opačnými znaménky. Trik: pro $x < 0$ lze psát $e^x = \dfrac{1}{e^{-x}}$.

Často tento problém nastává u řad typu binomického rozvoje.

Příklad 1.2.9. Učebnicový příklad ilustrující řešení přes rekurentní vztahy. Uvažujme integrál

$$I_n(a) = \int_0^1 \frac{x^n}{x+a}\,dx = (\text{trik } x^n = [(x+a) - a]^n) \tag{1.8}$$

$$= \sum_{k=0}^{n-1} (-a)^k \binom{n}{k} \left[(1+a)^{n-k} - a^{n-k} \right] \frac{1}{n-k} + (-a)^n \ln \frac{a+1}{a}. \tag{1.9}$$

Použití této řady v double precision dá například pro $a = 10$ a $n \gtrsim 10$ řešení se značnými chybami, nebo vyjdou přímo očividné nesmysly ($I_n < 0$). Koeficienty jsou například pro $n = 10$ a $k = 5$ řádu 10^{11}, a přitom výsledek leží v $0 < I_n < 1$!

Řešení tohoto problému spočívá ve využití rekurentního vztahu

$$I_n = \int_0^1 \frac{x^{n-1}(x+a-a)}{x+a}\,dx = \int_0^1 x^{n-1}\,dx - a\int_0^1 \frac{x^{n-1}}{x+a}\,dx = \frac{1}{n} - aI_{n-1}. \tag{1.10}$$

Dopředná rekurence, kdy startujeme s $I_0 = \ln\frac{a+1}{a}$, ovšem *není stabilní*! Libovolná chyba (i zaokrouhlovací) na začátku bude v n-tém kroku násobena faktorem $(-a)^n$ a výsledky pro $n \geq 15$ znehodnotí úplně.

Co to zkusit obráceně? Máme

$$I_{n-1} = \frac{1}{a}\left(\frac{1}{n} - I_n\right), \tag{1.11}$$

ovšem jak začít? Lze začít libovolně, neboť počáteční chyba bude po k iteracích potlačována faktorem $\frac{1}{a^k}$. Například pro $n = 10$ a $a = 10$ zvolíme $I_{n+k} = 1$ pro vhodně velké $k \gtrsim n + \log\frac{1}{\varepsilon_M}$ a dostaneme I_n s přesností ε_M.

1.3 Stabilita algoritmu

Definice 1.3.1 (Stabilita algoritmu)**.** Řekneme, že algoritmus výpočtu veličiny $f = f(x)$, závisející na vstupních datech x, je *stabilní*, pokud jeho provedením v aritmetice s pohyblivou řádovou čárkou dostaneme pro všechna x výsledek $\tilde{f}$, který se od přesného f liší s relativní chybou řádu $O(\varepsilon_M)$, tj. při zmenšení ε_M lze chybu odhadnout výrazem $C\varepsilon_M$, kde konstanta C nezávisí na x.

V praxi navíc požadujeme, aby pro konkrétní ε_M (např. pro $\varepsilon_M \sim 10^{-16}$ v double precision) nebylo $C \sim \varepsilon_M^{-1}$, tj. $\sim 10^{16}$, neboť pak by chyba byla srovnatelná s výsledkem!

Příklad 1.3.2. Při výpočtu $\sqrt{x+1} - \sqrt{x}$ lze pro velká kladná x odhadnout absolutní chybu jako $2\sqrt{x}\,\varepsilon_M$, a tedy relativní chyba bude

$$\delta \leq \frac{2\sqrt{x}\,\varepsilon_M}{\sqrt{x+1} - \sqrt{x}} = 2\varepsilon_M\sqrt{x}\left(\sqrt{x+1} + \sqrt{x}\right) \sim 4\varepsilon_M x, \tag{1.12}$$

což nelze odhadnout výrazem $C\varepsilon_M$, neboť pro velká x tento výraz roste. Odtud plyne, že výpočet je nestabilní.

Ovšem pro $\dfrac{1}{\sqrt{x+1} + \sqrt{x}}$ dostaneme pro odhad relativní chyby jmenovatele

$$\delta \leq \frac{2\sqrt{x}\,\varepsilon_M}{\sqrt{x+1} + \sqrt{x}} \sim \varepsilon_M \tag{1.13}$$

a pro celý výraz tedy také ε_M, takže jde o stabilní výpočet.

1.4 Podmíněnost úlohy

V některých úlohách může výsledek silně záviset na vstupních datech, a to bez ohledu na použitý algoritmus. Klasickým příkladem takové úlohy je určení kořenů polynomu zadaného jeho koeficienty, které představují vstupní data.

Příklad 1.4.1. Polynom

$$p(x) = x^{10} - 10x^9 + 45x^8 - \ldots - 10x + 1 = (x-1)^{10}$$

má desetinásobný kořen $x = 1$. Avšak $q(x) = p(x) - 10^{-10}$, kde se pouze a_0 změnilo o 10^{-10}, má kořeny

$$x_k = 1 + \frac{1}{10}\,e^{\frac{2\pi i k}{10}} \qquad \text{pro } k = 0, \ldots, 9,$$

a tedy změna výsledku je $\sim 10^{-1}$ (v přesné aritmetice!).

Obecně uvažujme problém, který řešíme, jako funkci $f : X \to Y$, kde X je normovaný vektorový prostor vstupních dat a Y je normovaný vektorový prostor řešení, s tím, že f je obecně (a též obvykle) nelineární.

Definice 1.4.2 (Dobrá a špatná podmíněnost úlohy)**.** Řekneme, že *úloha* (problém) *je dobře podmíněná* (well-conditioned), když malá změna vstupních dat x vede k malým změnám výstupních dat $f(x)$. Naopak *úloha je špatně podmíněná* (ill-conditioned), když malá změna x vede k velké změně $f(x)$.

Dobře podmíněnou úlohou je například výpočet vlastního integrálu či určení vlastních čísel a vektorů hermitovské matice. Špatně podmíněnou úlohou je hledání kořenů polynomu, řešení soustavy lineárních rovnic pro nevhodné matice či počáteční úloha pro diferenciální rovnice.

Podmíněnost se kvantifikuje buď pomocí *absolutního čísla podmíněnosti*

$$\hat{\kappa}(x) = \lim_{\varepsilon \to 0} \sup_{\|\delta x\| \leq \varepsilon} \frac{\|\delta f\|}{\|\delta x\|}, \tag{1.14}$$

kde δx je malá změna x a $\delta f = f(x + \delta x) - f(x)$, nebo pomocí *relativního čísla podmíněnosti*

$$\kappa(x) = \lim_{\varepsilon \to 0} \sup_{\|\delta x\| \leq \varepsilon} \frac{\|\delta f\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|}. \tag{1.15}$$

Pro diferencovatelné f je $\hat{\kappa}(x)$ dáno pomocí jakobiánu zobrazení $f(x)$:

$$J_{ij}(x) = \frac{\partial f_i}{\partial x_j}\Big|_x, \qquad \delta f \approx J(x)\, \delta x,$$

a tedy $\hat{\kappa}(x) = \|J(x)\|$, kde maticová norma $J(x)$ je indukovaná normami v X a Y. Dále pak

$$\kappa(x) = \frac{\|J(x)\|}{\dfrac{\|f(x)\|}{\|x\|}}. \tag{1.16}$$

Relativní číslo podmíněnosti je obvykle důležitější při výpočtech v pohyblivé řádové čárce, kde pracujeme s relativní chybou $\sim \varepsilon_M$.

Problém je (typicky) dobře podmíněný pro $\kappa \lesssim 10^2$ a špatně podmíněný pro $\kappa \gtrsim 10^{10}$; přesná hodnota κ ovšem závisí na zvolené normě.

Příklad 1.4.3. Výpočet $f(x) = \sqrt{x}$ je dobře podmíněná úloha, neboť

$$\kappa = \frac{\dfrac{1}{2\sqrt{x}}}{\dfrac{\sqrt{x}}{x}} = \frac{1}{2}.$$

Ovšem pro $f(x) = x_1 - x_2$ dostaneme $J = (1, -1)$ a pro $\|\cdot\|_\infty$ normu pak

$$\kappa = \frac{2}{\dfrac{|x_1 - x_2|}{\max\{|x_1|, |x_2|\}}} = \frac{2 \max\{|x_1|, |x_2|\}}{|x_1 - x_2|}, \tag{1.17}$$

což může být libovolně velké pro $x_1 \sim x_2$.

Poznámka 1.4.4. Zde $\|J\|_\infty = 2$ je maximum součtů řádkových prvků (viz poznámky k vektorovým a maticovým normám níže).

Příklad 1.4.5. Pro hledání kořenů polynomu jde $\kappa \to \infty$, neboť pro $\delta x = 10^{-10}$ je

$$\frac{\|\delta f\|}{\|\delta x\|} \approx 10^9,$$

pro $\delta x = 10^{-20}$ je

$$\frac{\|\delta f\|}{\|\delta x\|} \approx 10^{18}$$

a tak dále. Podíl $\frac{\|\delta f\|}{\|\delta x\|}$ tedy roste nade všechny meze, a přitom $\|f(x)\| \sim 10^2$ a $\|x\| \sim 10^0$.

1.5 Normy na vektorových prostorech a normy matic

Definice 1.5.1 (Norma)**.** Norma $\|\cdot\|$ je funkce obecně z $\mathbb{C}^m \to \mathbb{R}$ splňující pro $\forall x, y \in \mathbb{C}^m$ a $\alpha \in \mathbb{C}$:

1. $\|x\| \geq 0$ a $\|x\| = 0$ právě, když $x = 0$;
2. $\|x + y\| \leq \|x\| + \|y\|$;
3. $\|\alpha x\| = |\alpha| \|x\|$.

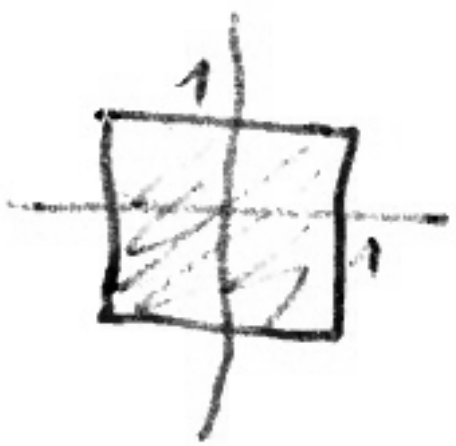
Uveďme p-normy na $\mathbb{C}^m$ a jejich jednotkové koule $\{x \in \mathbb{C}^m,\ \|x\| \leq 1\}$.

$$\|x\|_1 = \sum_{i=1}^{m} |x_i| \tag{1.18}$$

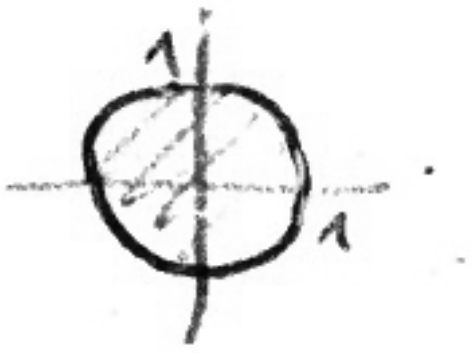


Obrázek 1.1: Jednotková koule 1-normy v rovině – kosočtverec (otočený čtverec) s vrcholy na osách, vnitřek vyšrafovaný; na svislé ose je vyznačena 1, na vodorovné ose 1.

$$\|x\|_\infty = \max_{1 \leq i \leq m} |x_i| \tag{1.19}$$



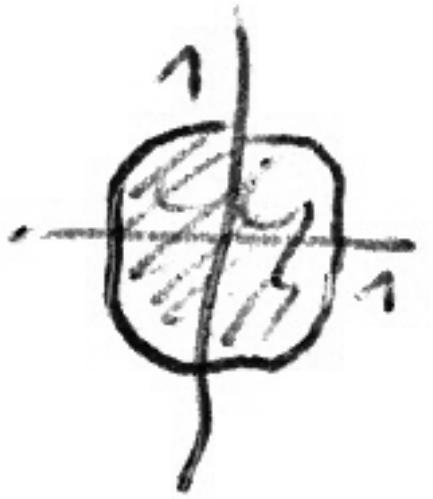
Obrázek 1.2: Jednotková koule ∞-normy – čtverec se stranami rovnoběžnými s osami, vnitřek vyšrafovaný; na svislé ose je vyznačena 1, na vodorovné ose 1.



Obrázek 1.3: Jednotková koule 2-normy – kružnice se středem v počátku, vnitřek vyšrafovaný; na svislé ose je vyznačena 1, na vodorovné ose 1.

$$\|x\|_2 = \left(\sum_{i=1}^{m} |x_i|^2 \right)^{1/2} \tag{1.20}$$

$$\|x\|_p = \left(\sum_{i=1}^{m} |x_i|^p \right)^{1/p} \tag{1.21}$$



Obrázek 1.4: Jednotková koule p-normy – čtverec se zaoblenými rohy (tvar mezi kružnicí a čtvercem), vnitřek vyšrafovaný; na svislé ose je vyznačena 1, na vodorovné ose 1.

Dále se používají vážené p-normy

$$\|x\|_W = \|Wx\|_p = \left(\sum_{i=1}^{m} |w_i x_i|^p \right)^{1/p}, \tag{1.22}$$

kde W je diagonální matice.

Tento pojem lze zobecnit na libovolné nesingulární matice W.

1.5.1 Maticová norma indukovaná vektorovou normou

Uvažujme matici $A \in \mathbb{C}^{m\times n}$ jako zobrazení z vektorového prostoru s normou $\|\cdot\|_{(n)}$ do prostoru s normou $\|\cdot\|_{(m)}$. Pak indukovaná maticová norma $\|A\|_{(m,n)}$ je nejmenší C, pro které platí

$$\|Ax\|_{(m)} \le C\|x\|_{(n)},$$

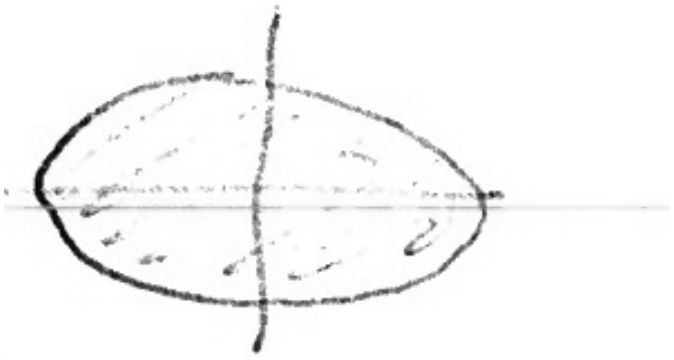
neboli

$$\|A\|_{(m,n)} = \sup_{\substack{x\in\mathbb{C}^n \\ x\neq 0}} \frac{\|Ax\|_{(m)}}{\|x\|_{(n)}} = \sup_{\substack{x\in\mathbb{C}^n \\ \|x\|_{(n)}=1}} \|Ax\|_{(m)}. \tag{1.23}$$

Příklad 1.5.2. Uvažujme matici

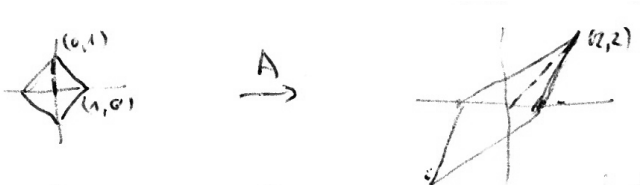
$$A = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}.$$

Obecně A zobrazí $e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ na $\begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ a $e_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ na $\begin{pmatrix} a_{12} \\ a_{22} \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$.



Obrázek 1.5: Jednotková koule vážené normy – elipsa protažená ve vodorovném směru, se středem v počátku os, vnitřek vyšrafovaný.

Pro 1-normu dostaneme $\|A\|_1 = 4$.



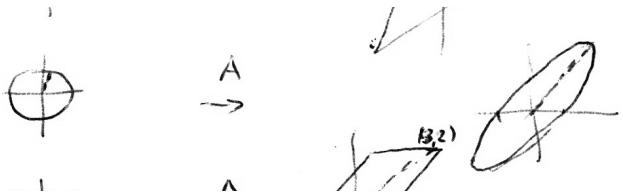
Obrázek 1.6: Vlevo jednotková koule 1-normy (kosočtverec s mírně prohnutými stranami) v osách, s popisky vrcholů $(0,1)$ a $(1,0)$; uprostřed šipka s popiskem A; vpravo obraz – kosodélník (rovnoběžník) v osách s vyznačenou čárkovanou úhlopříčkou a popiskem vrcholu $(2,2)$.

Pro 2-normu se supremum nabývá pro jednotkový vektor

$$x = \begin{pmatrix} \sqrt{\frac{1}{2} - \frac{7}{2\sqrt{65}}} \\ \sqrt{\frac{1}{2} + \frac{7}{2\sqrt{65}}} \end{pmatrix},$$

a tedy

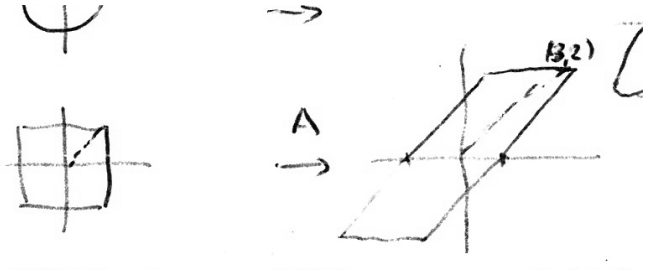
$$\|A\|_2 = \|Ax\|_2 = \sqrt{\frac{1}{2}\left(9 + \sqrt{65}\right)} \doteq 2{,}9208. \tag{1.24}$$



Obrázek 1.7: Vlevo jednotková koule 2-normy (kružnice) v osách; uprostřed šipka s popiskem A; vpravo obraz – protáhlá elipsa nakloněná zhruba pod 45° v osách, s čárkovanou hlavní poloosou.

Pro ∞-normu je $\|A\|_\infty = 3$.

Poznámka 1.5.3. Uvidíme později, že $\|A\|_2 = \max \sigma_i$, kde σ_i jsou singulární čísla.



Obrázek 1.8: Vlevo jednotková koule ∞-normy (čtverec) v osách s čárkovanou úhlopříčkou; uprostřed šipka s popiskem A; vpravo obraz – protáhlý kosodélník (rovnoběžník) nakloněný doprava, v osách, s čárkovanou úhlopříčkou a popiskem vrcholu $(3, 2)$.

Příklad 1.5.4. Pro libovolnou diagonální matici

$$D = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_m \end{pmatrix}$$

se jednotková koule transformuje na hyperelipsu s poloosami o délce $|d_i|$, a tedy

$$\|D\|_2 = \max_{1\le i\le n} |d_i| = \|D\|_p. \tag{1.25}$$

Platí, že $\|A\|_1$ je maximum součtů absolutních hodnot sloupcových prvků, tedy

$$\|A\|_1 = \max_{1\le j\le n} \|a_j\|_1, \quad \text{kde } a_j \text{ je } j\text{-tý sloupec matice,} \tag{1.26}$$

neboť

$$\|Ax\|_1 = \Big\| \sum_{j=1}^{n} x_j a_j \Big\|_1 \le \sum_{j=1}^{n} |x_j| \, \|a_j\|_1 \le \|x\|_1 \max_{1\le j\le n} \|a_j\|_1,$$

což platí i pro $\|x\|_1 = 1$. Přitom jsme využili rozkladu

$$Ax = \Big(a_1 \Big| a_2 \Big| \cdots \Big| a_n\Big) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \sum_{j=1}^{n} x_j a_j.$$

Volbou $x = e_j$, pro které je $\|a_j\|_1$ maximální, dostáváme rovnost, a tedy

$$\|A\|_1 = \sup_{\|x\|_1=1} \|Ax\|_1 = \max_{1\le j\le n} \|a_j\|_1. \tag{1.27}$$

Podobně $\|A\|_\infty$ je maximum součtů řádkových prvků, tedy

$$\|A\|_\infty = \max_{1\le i\le n} \|a_i^*\|_1, \quad \text{kde } a_i^* \text{ je } i\text{-tý řádek,} \tag{1.28}$$

neboť pro $\|x\|_\infty = 1$ je

$$\|Ax\|_\infty = \max_{1\le i\le m} \Big| \sum_{j=1}^{n} a_{ij} x_j \Big| \le \max_i \sum_{j=1}^{n} |a_{ij}|,$$

neboť $|x_j| \le 1$. Rovnost přitom nastává pro jeden z vektorů $x = (\pm 1, \pm 1, \cdots \pm 1)$.

Poznámka 1.5.5. Následující nerovnosti by studenti měli znát z lineární algebry. Hölderova nerovnost říká, že

$$|x^*y| \leq \|x\|_p \|y\|_q, \quad \text{kde} \quad 1 \leq p, q \leq \infty \quad \text{a} \quad \frac{1}{p} + \frac{1}{q} = 1, \tag{1.29}$$

přičemž x^*y je skalární součin. Speciálním případem je Cauchyova–Schwarzova nerovnost

$$|x^*y| \leq \|x\|_2 \|y\|_2. \tag{1.30}$$

Pro odhad normy součinu dvou matic platí

$$\|AB\|_{(\ell,n)} \leq \|A\|_{(\ell,-)} \|B\|_{(-,n)}, \tag{1.31}$$

ale obecně zde není rovnost. Například $\|A^n\| \leq \|A\|^n$, ale pro $n \geq 2$ obecně neplatí

$$\|A^n\| = \|A\|^n.$$

1.5.2 Obecné maticové normy

Obecné maticové normy splňují stejné podmínky jako normy vektorů, jen nyní pro matice – jako bychom měli $m \times n$ rozměrný vektorový prostor.

Nejpoužívanější je Frobeniova norma (též Hilbertova–Schmidtova norma)

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}, \tag{1.32}$$

tj. jde o 2-normu v $m \times n$ rozměrném prostoru. I zde platí odhad pro součin

$$\|AB\|_F \leq \|A\|_F \|B\|_F. \tag{1.33}$$

Poznámka 1.5.6. Lze ukázat, že $\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2}$.

Kapitola 2

Řešení nelineárních rovnic. Urychlování konvergence. Hledání kořenů

2.1 Řešení nelineárních rovnic pomocí iterací

Metodu, kterou se zde budeme zabývat, lze také najít pod názvem metoda postupných aproximací.

Rovnici typu $x = f(x)$, kde $f(x)$ je obecně nelineární funkce, či dokonce operátor, lze velmi často řešit pomocí *metody prostých iterací*. Zvolíme počáteční odhad $x = x_0$ a iterujeme pomocí

$$x_{n+1} = f(x_n). \tag{2.1}$$

Zda pro $n \to \infty$ konvergujeme, tj. $x_n \to x_P$, kde x_P řeší $x_P = f(x_P)$, závisí obecně jak na problému (funkci f), tak na volbě x_0.

Věta 2.1.1 (O pevném bodě kontrahujícího zobrazení (Banachova věta, fixed point)). *Nechť f je spojité zobrazení definované na oblasti D a nechť f zobrazuje D na sebe, tj. $f(D) \subset D$, a splňuje* lipschitzovskou podmínku *kontrahujícího zobrazení, tj. $\exists L < 1$ takové, že pro $\forall x_1, x_2 \in D$ platí (ve vhodné normě)*

$$\|f(x_1) - f(x_2)\| \leq L \|x_1 - x_2\|. \tag{2.2}$$

Pak

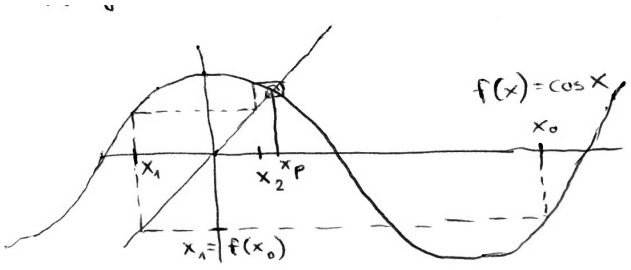
1. *$f(x)$ má na D právě jeden pevný bod $x_P = f(x_P)$,*
2. *posloupnost $x_n = f(x_{n-1})$ konverguje k x_P pro libovolné počáteční $x_0 \in D$, tj. $\lim_{n\to\infty} \|x_n - x_P\| = 0$,*
3. *pro odhad chyby v n-té iteraci platí*

$$\|x_n - x_P\| \leq \frac{L^n}{1 - L} \|x_1 - x_0\|. \tag{2.3}$$

Důkaz. Viz např. J. Segethová: Základy numerické matematiky. □

Pro $F \in C^1\langle a, b\rangle$ tedy metoda konverguje, pokud $|f'(x)| < 1$ pro $\forall x \in \langle a, b\rangle$.

Pro rovnici $x = \cos x$ dostáváme dokonce řešení pro libovolné x_0, tj. vždy konvergujeme k $x_P \doteq 0{,}739\ldots$ Otázkou zůstává, jak rychle.



Obrázek 2.1: Graf funkce $f(x) = \cos x$ spolu s přímkou $y = x$ a schodovitou (pavučinovou) konstrukcí iterací. Popisky: $f(x) = \cos x$, x_0, x_1, x_2, x_P, $x_1 = f(x_0)$; čárkované spojnice mezi osou a grafem naznačují postup iterací ke kroužkem vyznačenému průsečíku.

2.2 Rychlost konvergence a řád iterační metody

Rychlost konvergence (speed of convergence) je dána tzv. *řádem iterační metody* (order of convergence). Označíme-li chybu v n-tém kroku $e_n = x_n - x_P$, pak pro diferencovatelnou funkci f dostáváme

$$\begin{aligned} e_{n+1} = x_{n+1} - x_P = f(x_n) - f(x_P) = f(x_P + e_n) - f(x_P) = \\ = f'(x_P)\, e_n + \frac{1}{2} f''(x_P)\, e_n^2 + \dots \qquad \text{pro malé } e_n. \end{aligned}$$

V závislosti na tom, kolik derivací je nulových v bodě x_P, dostáváme metody různých řádů. Pro *lineární metody* bude $f'(x_P) \neq 0$, a tedy $e_{n+1} \approx q\, e_n$, kde $q = f'(x_P)$. Počet správných cifer tedy roste lineárně s počtem kroků, neboť $e_n \sim e_0 q^n$, a proto

$$\underbrace{\log|e_n| - \log|e|_{n+1}}_{\text{počet platných cifer získaných v jedné iteraci}} = \big[n - (n+1)\big] \log|q| = -\log|q|. \tag{2.4}$$

Pro *kvadratické metody* bude $f'(x_P) = 0$ a $f''(x_P) \neq 0$, a tedy

$$e_{n+1} \approx \tfrac{1}{2} f''(x_P)\, e_n^2.$$

Nyní

$$\log|e_{n+1}| = \log|q| + 2\log|e_n|, \tag{2.5}$$

a počet platných cifer se v každé iteraci minimálně zdvojnásobí. Podobně je tomu i pro metody vyšších řádů.

Definice 2.2.1 (Řád metody)**.** Řád metody je nenulové α, pro nějž existuje konečná, nenulová limita

$$\lim_{n\to\infty} \frac{|x_{n+1} - x_P|}{|x_n - x_P|^\alpha} = C < \infty \qquad \left(= \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} \right). \tag{2.6}$$

Říkáme též, že iterační metoda má řád konvergence α v bodě x_P.

- C samozřejmě závisí na $f(x)$.
- α je jednoznačně určeno díky $c \neq 0$ a $\alpha \geq 1$.

Existují i obecnější metody, například vícekrokové

$$F_n(x_n, x_{n-1}, \dots, x_{n-s+1}) = x_{n+1},$$

a pokud $F_n = F$, jde o stacionární metodu. Existují také vždy konvergentní metody fungující pro libovolné $f(x)$ a x_0; ty jsou ovšem typicky pomalejší a používají se jako startovací.

Příklad 2.2.2. Hledání řešení rovnice $x = \cos x$ prostou iterací je lineární metoda ($\alpha = 1$), neboť $\frac{\mathrm{d}}{\mathrm{d}x}\cos x\big|_{x=x_P} \neq 0$. Přesněji

$$\left.\frac{\mathrm{d}\cos x}{\mathrm{d}x}\right|_{x=x_P} = -\sin x_P \doteq -0{,}6736\ldots \approx -\frac{2}{3},$$

a tedy $e_{n+1} \approx -\frac{2}{3}e_n$, z čehož vidíme, že zhruba po šesti iteracích se zmenší chyba o jeden řád, neboť $\left(\frac{2}{3}\right)^6 \sim \frac{1}{10}$.

2.3 Urychlování konvergence lineárních metod

2.3.1 Aitkenova Δ^2 metoda

Pro lineární metody lze pro dostatečně velké n psát chybu jako $e_n \simeq cq^n$, a pro tři po sobě jdoucí aproximace tak máme (alespoň přibližně)

$$\begin{aligned} x_n - x_P &\simeq cq^n, \\ x_{n+1} - x_P &\simeq cq^{n+1}, \\ x_{n+2} - x_P &\simeq cq^{n+2}, \end{aligned}$$

což jsou tři rovnice pro tři neznámé x_P, c a q. Vyjádříme

$$q = \frac{x_{n+1} - x_P}{x_n - x_P} = \frac{x_{n+2} - x_P}{x_{n+1} - x_P}, \tag{2.7}$$

z čehož

$$x_P \simeq \frac{x_n x_{n+2} - x_{n+1}^2}{x_n - 2x_{n+1} + x_{n+2}} = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n}. \tag{2.8}$$

Druhý tvar je lepší z hlediska zaokrouhlovacích chyb, neboť $x_n x_{n+2}$ a x_{n+1}^2 mohou být blízká čísla.

Neboli

$$x_P \approx x'_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n} \quad \text{, kde } \Delta \text{ je operátor dopředné diference,} \tag{2.9}$$

přičemž právě od $\Delta^2 x_n$ ve jmenovateli pochází název metody.

Pozor, výsledek je opět jen přibližný, takže dostáváme novou posloupnost x'_n, pro kterou lze ukázat, že

$$\lim_{n\to\infty} \frac{x'_n - x_P}{x_n - x_P} = 0, \tag{2.10}$$

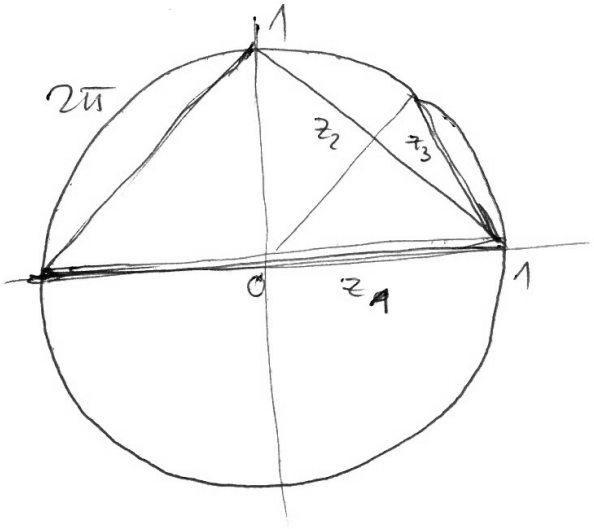
a tedy konverguje k x_P rychleji než x_n. Obecně však jde opět o lineární metodu, ovšem s $q' < q$.

Poznámka 2.3.1. Později se setkáme s podobnou „extrapolací“ ze známého chování chyby u numerické derivace a integrace.

2.3.2 Archimedův výpočet čísla π

Číslo π postupně aproximujeme obvodem mnohoúhelníku vepsaného do kružnice.

$$\begin{aligned} 2\pi &\doteq 2z_1 && \Rightarrow && \pi &\doteq 2 \\ 2\pi &\doteq 4z_2 && \Rightarrow && \pi &\doteq 2\sqrt{2} \\ 2\pi &\doteq 8z_3 && \Rightarrow && \pi &\doteq 4\sqrt{2-\sqrt{2}} \end{aligned}$$



Obrázek 2.2: Kružnice o poloměru 1 se středem O s vyznačenými vodorovnou a svislou osou. Do kružnice je vepsán postupně zjemňovaný mnohoúhelník (čtverec, osmiúhelník). Popisky: 2π u oblouku vlevo nahoře, 1 u horního průsečíku se svislou osou, 1 u pravého průsečíku s vodorovnou osou, O ve středu, z_1 pro vodorovnou tětivu (průměr), z_2 pro tětivu z horního bodu do pravého bodu a z_3 pro kratší tětivu u pravého okraje.

a tak dále.

Obecně dostaneme rekurentní vztah pro z_k. Nejprve

$$x = 1 - \sqrt{1 - \left(\frac{z_k}{2}\right)^2}, \tag{2.11}$$

a tedy

$$\begin{aligned} z_{k+1} &= \sqrt{x^2 + \left(\frac{z_k}{2}\right)^2} = \\ &= \sqrt{1 - 2\sqrt{1 - \left(\frac{z_k}{2}\right)^2} + \left(1 - \left(\frac{z_k}{2}\right)^2\right) + \left(\frac{z_k}{2}\right)^2}, \end{aligned}$$

takže

$$z_{k+1} = \sqrt{2}\sqrt{1 - \sqrt{1 - \left(\frac{z_k}{2}\right)^2}}. \tag{2.12}$$

Protože $2\pi \doteq 2^k z_k = 2y_k$, bude

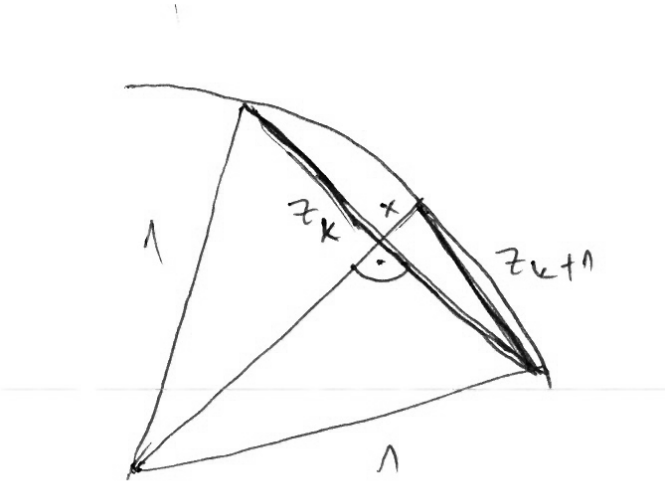
$$\pi \doteq y_{k+1} = 2^k z_{k+1} = \frac{2^{k+1}}{\sqrt{2}}\sqrt{1 - \sqrt{1 - \left(\frac{y_k}{2^k}\right)^2}}. \tag{2.13}$$

Zde je ovšem problém s odčítáním dvou blízkých čísel, neboť $\frac{y_k}{2^k} \to 0$ pro velká k.

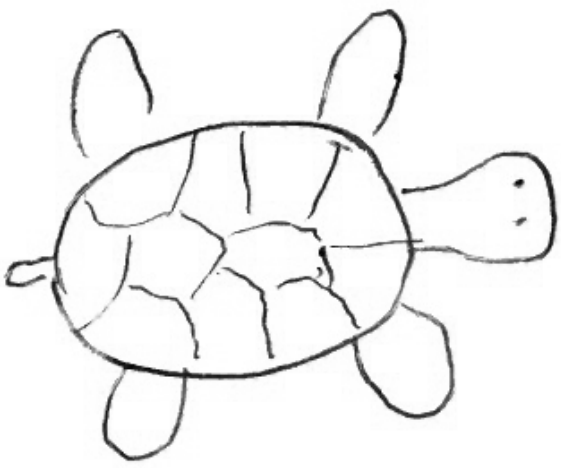
Proto výraz vynásobíme zlomkem $\frac{\sqrt{1+\sqrt{\cdots}}}{\sqrt{1+\sqrt{\cdots}}}$ a dostaneme

$$\tilde{y}_{k+1} = \frac{2^{k+1}}{\sqrt{2}} \cdot \frac{\sqrt{1 - \left(1 - \left(\frac{\tilde{y}_k}{2^k}\right)^2\right)}}{\sqrt{1 + \sqrt{1 - \left(\frac{\tilde{y}_k}{2}\right)^2}}} = \tilde{y}_k \sqrt{\frac{2}{1 + \sqrt{1 - \left(\frac{\tilde{y}_k}{2^k}\right)^2}}}, \tag{2.14}$$

což je stabilní výpočet s chybou zhruba 10^{-14} po 25 iteracích.



Obrázek 2.3: Výsek kružnice s obloukem; ze středu (vlevo dole) vedou dvě úsečky délky 1 ke dvěma bodům na oblouku. Tětiva mezi krajními body oblouku je označena z_k, kratší tětiva (přes polovinu oblouku) je označena z_{k+1}. Krátká úsečka x leží na prodloužení poloměru mezi tětivou z_k a obloukem, kolmost je vyznačena značkou pravého úhlu.

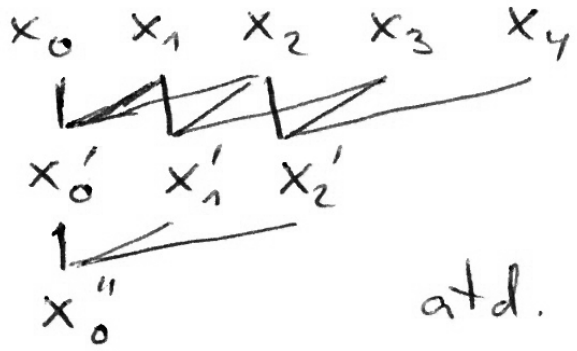


Obrázek 2.4: Perokresba želvy shora (krunýř s políčky, hlava, ocásek a čtyři nohy) – doplňková kresba bez popisků.

2.3.3 Aitkenova–Steffensenova metoda

Použijeme-li opakovaně Aitkenovu Δ^2 metodu, dostaneme posloupnosti

$$x_n \to x'_n \to x''_n \to \dots$$



Obrázek 2.5: Schéma opakovaného použití Aitkenovy metody: horní řada bodů x_0, x_1, x_2, x_3, x_4, z níž šikmé (cikcakové) šipky vedou na druhou řadu x'_0, x'_1, x'_2, a z ní dále šipkami na x''_0; vpravo dole popisek „atd.“

Alternativně lze uvažovat posloupnosti $x_0^{(k)}$, $x_1^{(k)}$ a $x_2^{(k)}$, které získáme následovně (nyní se mění horní index):

1. na začátku položíme $x_0^{(0)} = x_0$, $x_1^{(0)} = x_1 = f(x_0)$, $x_2^{(0)} = x_2 = f(x_1)$;
2. pro $k = 0, 1, 2, \dots$ spočteme vždy $x_0^{(k)}$ z Aitkenovy Δ^2 metody, tj.

$$x_0^{(k+1)} = x_0^{(k)} - \frac{\left(x_1^{(k)} - x_0^{(k)}\right)^2}{x_2^{(k)} - 2x_1^{(k)} + x_0^{(k)}}, \tag{2.15}$$

ovšem $x_1^{(k)} = f\left(x_0^{(k)}\right)$ a $x_2^{(k)} = f\left(x_1^{(k)}\right)$.

Souhrnně lze psát jedinou posloupnost $x_0^{(k)}$ danou vztahem

$$x_0^{(k+1)} = x_0^{(k)} - \frac{\left[f\left(x_0^{(k)}\right) - x_0^{(k)}\right]^2}{f\left(f\left(x_0^{(k)}\right)\right) - 2f\left(x_0^{(k)}\right) + x_0^{(k)}}, \tag{2.16}$$

o které si ukažme, že *konverguje kvadraticky* k x_p.

Nyní totiž neiterujeme f, ale novou funkci

$$\tilde{f}(x) = x - \frac{[f(x) - x]^2}{f(f(x)) - 2f(x) + x}. \tag{2.17}$$

Označme $q = f'(x_p)$, pak

$$f(x_p + \varepsilon) = x_p + \varepsilon q + \mathcal{O}(\varepsilon^2),$$
$$f(f(x_p + \varepsilon)) = x_p + \varepsilon q^2 + \mathcal{O}(\varepsilon^2),$$

a dostaneme

$$\tilde{f}(x_p + \varepsilon) = x_p + \varepsilon - \frac{(x_p + \varepsilon q - x_p - \varepsilon)^2}{x_p + \varepsilon q^2 - 2x_p - 2\varepsilon q + x_p + \varepsilon} + \mathcal{O}(\varepsilon^2)$$
$$= x_p + \varepsilon - \varepsilon \frac{(q-1)^2}{(q-1)^2} + \mathcal{O}(\varepsilon^2) \;=\; x_p + \mathcal{O}(\varepsilon^2).$$

Neboli $\tilde{f}(x_p) = x_p$, jde tedy o pevný bod $\tilde{f}$, a zároveň $\tilde{f}'(x_p) = 0$, což znamená kvadratickou konvergenci.

2.4 Hledání kořenů algebraické rovnice $g(x) = 0$

2.4.1 Vždy konvergentní metody

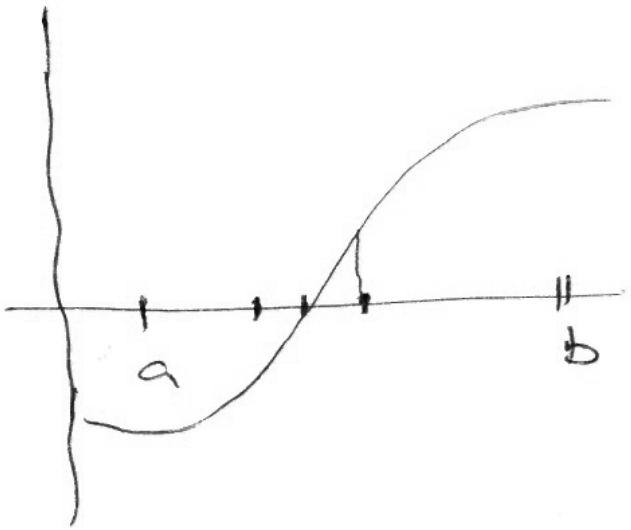
Uvažujme reálné funkce na intervalu $\langle a, b\rangle$. Základním předpokladem je, že $g(x)$ je spojitá na $\langle a, b\rangle$ a $g(a)\, g(b) < 0$, aby zde měla kořen. Standardní „klasické“ metody jsou následující.

a) Metoda bisekce (půlení intervalu)

Konstruujeme dvě posloupnosti a_n a b_n:

$$a_0 = a, \quad b_0 = b,$$
$$\text{pro } n = 0, 1, 2 \ldots \qquad x_n = \frac{a_n + b_n}{2},$$
$$\text{je-li } g(a_n)\, g(x_n) \le 0 : \; b_{n+1} = x_n, \; a_{n+1} = a_n,$$
$$\text{jinak } b_{n+1} = b_n, \; a_{n+1} = x_n,$$

a to dokud $|g(x_n)| \not\geq \varepsilon$.



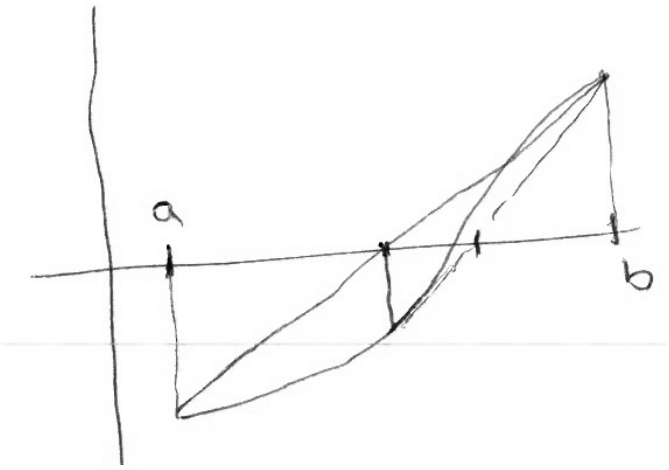
Obrázek 2.6: Graf rostoucí funkce protínající vodorovnou osu; na ose jsou vyznačeny čárkami dělicí body, svislá čárka vlevo je osa. Popisky: a pod osou u levé značky, b pod osou u pravé dvojité značky. U jednoho z dělicích bodů je svislá spojnice od osy ke křivce.

b) Metoda regula falsi (metoda sečen)

Opět konstruujeme a_n a b_n s $a_0 = a$, $b_0 = b$, ale nyní proložíme přímku:

$$\text{pro } n = 0, 1, \ldots \qquad x_n = \frac{a_n\, g(b_n) - b_n\, g(a_n)}{g(b_n) - g(a_n)}. \tag{2.18}$$

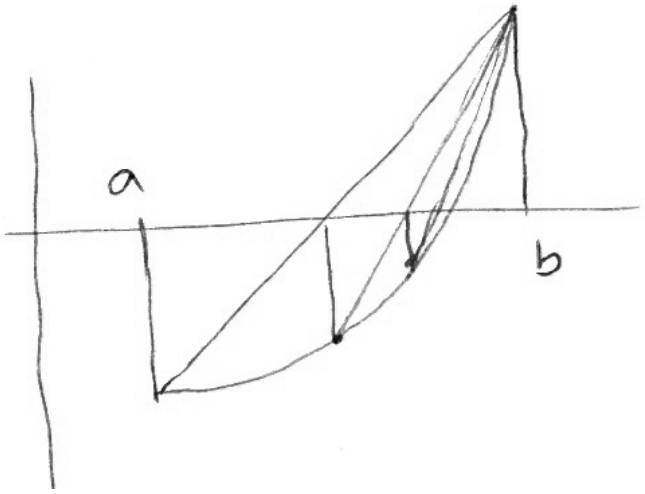
Podmínky pro výběr nového intervalu jsou stejné jako u bisekce.



Obrázek 2.7: Rostoucí konvexní křivka protínající šikmou vodorovnou osu, s vepsanými sečnami spojujícími krajní body; svislé úsečky vedou od osy ke křivce v mezilehlých bodech. Popisky: a vlevo nad osou, b vpravo pod osou.

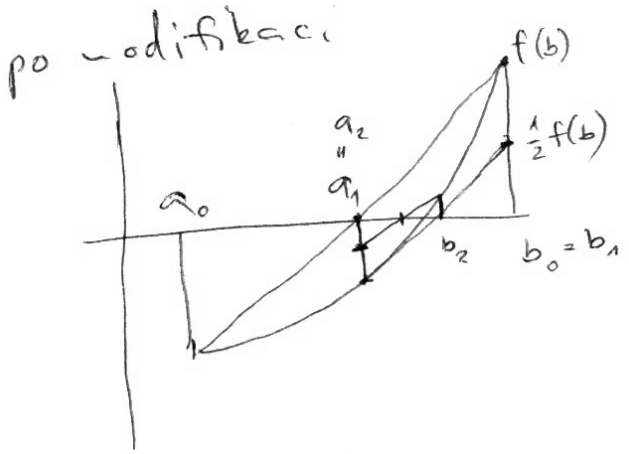
c) Modifikovaná metoda regula falsi

Uvedeme ji pouze obrázkově, viz obrázky 2.8 a 2.9.



Obrázek 2.8: Stav „před“: rostoucí konvexní křivka protínající vodorovnou osu, z pravého krajního bodu (nad osou) vychází svazek sečen do postupných bodů na křivce vlevo dole; svislé úsečky od osy dolů ke křivce. Popisky: a vlevo nad osou, b vpravo pod osou.

Podrobnosti o konvergenci a detaily algoritmů viz J. Segethová: Základy numerické matematiky, kap. 6.3.



Obrázek 2.9: Stav „po modifikaci“: táž konvexní křivka se sečnami, ale funkční hodnota v pevném krajním bodě je půlena, takže sečny se sklánějí a body se rychleji blíží kořeni; šipky u svislých úseček naznačují postup. Popisky: a_0, a_1, a_2 (mezi a_2 a a_1 je drobná dvojitá značka, patrně „=“), b_2, $b_0 = b_1$, $f(b)$ vpravo nahoře a $\frac{1}{2}f(b)$ vpravo níže.

2.4.2 Iterační metody vyššího řádu

Pišme

$$x = f(x) = x - \phi(x)\, g(x), \tag{2.19}$$

kde $\phi(x)$ je vhodná funkce. Je-li x_p kořen, tj. $g(x_p) = 0$, pak je též pevným bodem funkce $f(x)$, a to za předpokladu, že jde o kontrahující zobrazení.

Funkce $\phi(x)$ nesmí mít kořen na intervalu, kde hledáme x_p, a navíc musí platit $0 < |\phi(x)| < \infty$, tedy musí být omezená.

Jinou možností je například $f(x) = x - F(g(x))$, kde

$$F(0) = 0, \quad F(y) \neq 0 \text{ pro } y \neq 0. \tag{2.20}$$

a) Newtonova metoda (metoda tečen)

Pokud má být $x = f(x) = x - \phi(x)\, g(x)$ metodou 2. řádu, musí být

$$f'(x_p) = 0 = 1 - \phi'(x_p) \underbrace{g(x_p)}_{\| \; 0} - \phi(x_p)\, g'(x_p), \tag{2.21}$$

neboť x_p je kořen rovnice $g(x) = 0$.

Z toho plyne $\phi(x_p) = \dfrac{1}{g'(x_p)}$, neboli zvolíme $\phi(x) = \dfrac{1}{g'(x)}$, což dává

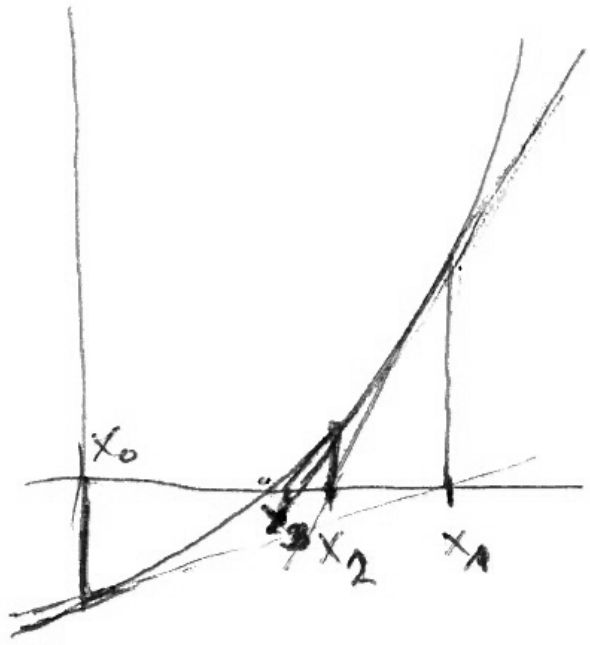
$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}. \tag{2.22}$$

A protože

$$f'(x_p) = \frac{g(x_p)\, g''(x_p)}{[g'(x_p)]^2} = 0,$$

vyžadujeme $g'(x_p) \neq 0$ a existenci $g''(x_p)$.

Newtonova metoda nekonverguje vždy, ale pokud je na určitém intervalu $|f'(x)| < 1$, pak ano. Často funguje, pokud $g''(x)$ kolem kořene x_p nemění znaménko. Je nutný „nástřel“ pomocí nějaké vždy konvergentní metody a pak lze rychle doiterovat Newtonovou metodou.



Obrázek 2.10: Náčrt Newtonovy metody (metody tečen): rostoucí konvexní křivka protínající vodorovnou osu, svislá osa vlevo; z bodu x_1 je vedena svislice ke křivce a tečna, která protíná osu v x_2, odtud další svislice a tečna ke x_3; iterace se blíží ke kořeni. Popisky zleva doprava na vodorovné ose: x_0 u svislé osy vlevo, x_3 (silně přepisovaný, rozmazaný), x_2, x_1.

Speciálně pro $g(x) = x^2 - C = 0$, kdy hledáme $\sqrt{C}$, dostaneme

$$x_{n+1} = x_n - \frac{x_n^2 - C}{2x_n} = \frac{1}{2}\left(x_n + \frac{C}{x_n}\right). \tag{2.23}$$

b) Dvoubodová metoda sečen

Neznáme-li derivaci $g'(x)$, nebo je-li obtížné ji spočítat, lze ji aproximovat, například

$$g'(x_k) = \frac{g(x_k) - g(x_{k-1})}{x_k - x_{k-1}},$$

a tedy

$$x_{n+1} = \frac{x_{n-1}\, g(x_n) - x_n\, g(x_{n-1})}{g(x_n) - g(x_{n-1})}. \tag{2.24}$$

Lze ukázat (viz např. Isaacson, Keller: Analysis of Numerical Methods), že pro $g(x)$ splňující $g'(x) \neq 0$ na intervalu, kde hledáme kořen, a pro $g''(x)$, které nemění znaménko, je konvergence řádu

$$\alpha = \frac{\sqrt{5}+1}{2} \simeq 1{,}618, \tag{2.25}$$

tedy zlatého řezu. Ukáže se totiž, že $e_{n+1} \leq e_n e_{n-1}$, a pro počáteční chybu $\varrho = \max(e_0, e_1)$ dostaneme $e_2 \leq \varrho^2$, $e_3 \leq \varrho^3$, $e_4 \leq \varrho^5$ a tak dále; exponenty tvoří Fibonacciho posloupnost.

Protože však na jednu iteraci potřebujeme vyčíslit g pouze jednou, může být tato metoda rychlejší než Newtonova, kde potřebujeme i derivaci.

Kořeny polynomů

Pro hledání kořenů polynomů lze použít výše uvedené metody. Newtonovu metodu lze použít i pro hledání komplexních kořenů; jen pro vícenásobné kořeny dostáváme lineární konvergenci. Pokud známe násobnost kořene p, iterujeme

$$x_{n+1} = x_n - p\,\frac{g(x_n)}{g'(x_n)}. \tag{2.26}$$

Dále je to například Laguerreova metoda, která využívá 2. derivace, též Skowron, Gould:

c) Iterační metoda pomocí Padého aproximace

Vyjdeme z racionální aproximace jisté funkce $f(x)$

$$f(x) \simeq \frac{a_0 + a_1 x + \cdots + a_N x^N}{b_0 + b_1 x + \ldots b_M x^M} = P_{II}^{[N/M]}(x) = \frac{P^N(x)}{Q^M(x)}, \tag{2.27}$$

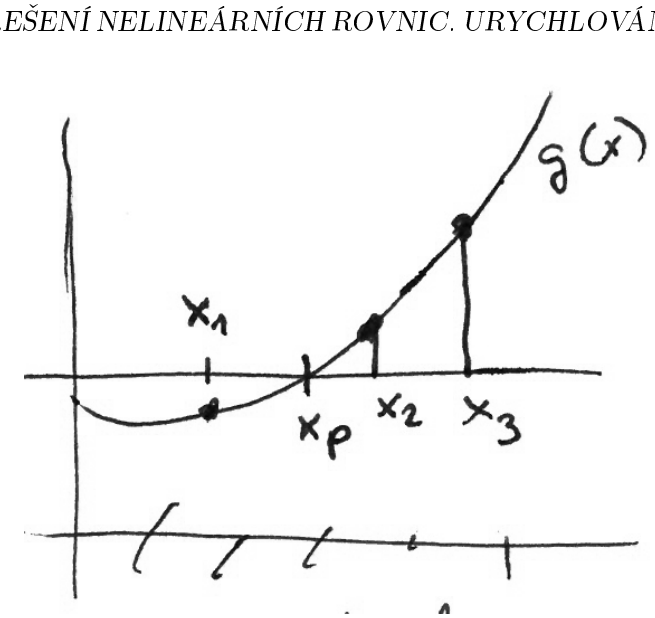
s tím, že se obvykle položí $b_0 = 1$. Aproximace $P_I^{[N/M]}(x)$ by byla Padého aproximace získaná porovnáním s Taylorovým rozvojem.

K určení a_i a b_j je potřeba $N + M + 1$ bodů, což dává soustavu lineárních rovnic

$$P^N(x_k) = f(x_k)\, Q^M(x_k). \tag{2.28}$$

Použijeme-li $P_{II}^{[1/1]}(x)$ k aproximaci funkce $g(x)$, jejíž kořen hledáme, a určíme-li kořen jako $x_p \simeq -\dfrac{a_0}{a_1}$, pak opakováním tohoto postupu dostaneme iterační metodu

$$x_n = \frac{x_{n-1}\,\dfrac{(x_{n-2} - x_{n-3})}{F(x_{n-1})} + x_{n-2}\,\dfrac{x_{n-3} - x_{n-1}}{F(x_{n-2})} + x_{n-3}\,\dfrac{x_{n-1} - x_{n-2}}{F(x_{n-3})}}{\dfrac{x_{n-2} - x_{n-3}}{F(x_{n-1})} + \dfrac{x_{n-3} - x_{n-1}}{F(x_{n-2})} + \dfrac{x_{n-1} - x_{n-2}}{F(x_{n-3})}}. \tag{2.29}$$



Obrázek 2.11: Nahoře graf funkce $g(x)$ v soustavě souřadnic – křivka s minimem vlevo pod osou, protínající vodorovnou osu a rostoucí vpravo nahoru; na ose jsou vyznačeny body x_1, x_p, x_2, x_3, přičemž body křivky nad x_2 a x_3 jsou spojeny svislicemi s osou a bod nad x_1 leží pod osou. Pod tím druhý malý náčrtek: vodorovná přímka protnutá svislou osou s několika šikmými značkami a tečkou, tedy body v komplexní rovině.

Neboli jde o vážený průměr bodů x_{n-1}, x_{n-2} a x_{n-3} pomocí vah

$$w_{n-1} = \frac{x_{n-2} - x_{n-3}}{F(x_{n-1})} \quad \text{atd.}$$

2.5 Řešení rovnic pro více proměnných

Banachova věta o pevném bodě platí stále, jen x je nyní vektor a $f(x)$ je vektorová funkce. Iterujeme

$$x_i^{n+1} = f_i(x_1^n, \dots x_D^n) \quad \text{pro } i = 1, 2, \dots, D. \tag{2.30}$$

Pro chybu nyní máme

$$e^{n+1} = x^{n+1} - x_{(P)} = f(x^n) - f(x_{(P)}) = J(x_{(P)})\, e^n + O\big(\|e^n\|^2\big), \tag{2.31}$$

kde $J(x)$ je jakobián zobrazení f, tj.

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_D}{\partial x_1} & \dots & \frac{\partial f_D}{\partial x_D} \end{pmatrix}. \tag{2.32}$$

Má-li být $f(x)$ kontrahující zobrazení, musí být všechna vlastní čísla J menší než 1 (v absolutní hodnotě) pro $\forall x$ z uvažované oblasti.

Řád konvergence α je dán vztahem

$$\lim_{n\to\infty} \frac{\|x^{n+1} - x_{(P)}\|}{\|x^n - x_{(P)}\|^\alpha} = C < \infty, \tag{2.33}$$

a je-li $J(x_{(P)}) \neq 0$, budeme mít lineární konvergenci, atd.

Chceme-li vyřešit soustavu $g(x) = 0$ pro vektorovou funkci g, pak můžeme hledat pevný bod zobrazení

$$f(x) = x - A(x)\, g(x), \tag{2.34}$$

kde $A(x)$ je nesingulární matice pro $\forall x$ z jistého okolí bodu $x_{(P)}$, který je řešením $g(x_{(P)}) = 0$.

Newtonova metoda plyne z požadavku, aby byl jakobián nulový, tj.

$$0 = \frac{\partial f_i}{\partial x_j} = \delta_{ij} - \sum_k \left(\frac{\partial A_{ik}}{\partial x_j} \underbrace{g_k}_{\substack{\| \\ 0 \text{ pro } x=x_{(P)}}} + A_{ik} \frac{\partial g_k}{\partial x_j} \right) \quad \text{v } x = x_{(P)}, \tag{2.35}$$

a tedy

$$\sum_{k=1}^{D} A_{ik} \frac{\partial g_k}{\partial x_j} = \delta_{ij}, \tag{2.36}$$

neboli musíme iterovat

$$x^{n+1} = x^n - J_g(x^n)^{-1} g(x^n) \quad \text{, kde} \quad J_g(x)_{ij} = \frac{\partial g_i(x)}{\partial x_j}, \tag{2.37}$$

nebo zapsáno jako soustava rovnic

$$J_g(x^n)(x^n - x^{n+1}) = g(x^n).$$

Někdy se používá tzv. relaxace, kdy na pravé straně $x = f(x)$ použijeme již modifikované složky x^n:

$$x_i^{n+1} = f_i(x_1^{n+1}, \ldots x_{i-1}^{n+1}, x_i^n, \ldots x_D^n). \tag{2.38}$$

Kapitola 3

Aproximace funkcí. Polynomiální interpolace. Splajny

3.1 Aproximace funkcí v počítači

Funkce je v počítači nutné reprezentovat diskrétně. Možností, jak to udělat, je celá řada: na gridu, v určité bázi (polynomy apod.), řetězovým zlomkem atd. Nejčastěji se používají polynomy, případně racionální funkce, neboť je lze snadno vyčíslit pomocí základních operací.

I když ovšem použijeme např. polynomy, máme stále několik možností, jak danou funkci aproximovat. Nejběžnější metody aproximace jsou následující tři.

3.1.1 Interpolace

Hodnoty funkce (případně její derivace) jsou zadány v bodech $x_0, \ldots, x_n$ a hledáme funkci $\phi(x)$ z určité třídy funkcí (nejčastěji polynomů) takovou, že

$$\phi(x_i) = f(x_i), \quad \text{pro } i = 0, \ldots, n, \tag{3.1}$$

což je *Lagrangeova interpolace*, případně navíc

$$\phi^{(k)}(x_i) = f^{(k)}(x_i), \quad \text{pro } i = 0, \ldots, n, \tag{3.2}$$

což je *Hermiteova interpolace*. Takto získaná funkce aproximuje původní funkci $f(x)$ mezi body x_i (interpolace), případně i vně intervalu $\langle x_0, x_n \rangle$ (extrapolace).

3.1.2 Metoda nejmenších čtverců

Opět hledáme aproximaci $\phi(x)$ funkce $f(x)$ zadané v bodech $x_0, \ldots, x_n$, ovšem nyní takovou, že

$$\sum_{i=0}^{n} w_i \left(f(x_i) - \phi(x_i)\right)^2 = \min_{\substack{\text{přes} \\ \forall \psi \text{ z jisté třídy funkcí}}} \sum_{i=0}^{n} w_i \left(f(x_i) - \psi(x_i)\right)^2, \tag{3.3}$$

kde $w_i > 0$ jsou váhy jednotlivých funkčních hodnot.

Někdy lze minimalizovat i integrální výraz

$$\int_a^b w(x) \left(f(x) - \phi(x)\right)^2 \mathrm{d}x = \min_{\psi} \int_a^b w(x) \left(f(x) - \phi(x)\right)^2 \mathrm{d}x, \tag{3.4}$$

kde $w(x)$ je (skoro všude) kladná na intervalu $\langle a, b \rangle$, přičemž $w \in L^2(\langle a, b \rangle)$.

3.1.3 Čebyševova aproximace

Hledáme $\phi(x)$ takové, že

$$\max_{x\in\langle a,b\rangle} |\phi(x) - f(x)| \leq \max_{x\in\langle a,b\rangle} |\psi(x) - f(x)| \quad \text{pro } \forall\psi \tag{3.5}$$

z prostoru, ve kterém hledáme ϕ.

3.1.4 Volba normy a role polynomů

Volbou různých norem, případně seminorem, kterými poměřujeme blízkost funkcí, dostáváme různé aproximace. Nejčastěji se používá aproximace polynomy (případně funkcemi, které jsou po částech polynomy, tedy splajny) do určitého stupně n.

Poznámka 3.1.1. Z analýzy znáte jednak Taylorův rozvoj, jednak platí Weierstrassova věta.

Věta 3.1.2 (Weierstrassova věta)**.** *Nechť $\langle a,b\rangle$ je konečný uzavřený interval a f je spojitá funkce na $\langle a,b\rangle$, tj. $f \in C(\langle a,b\rangle)$, a nechť $\varepsilon > 0$ je libovolně malé. Pak existuje polynom $p(x)$ takový, že*

$$|f(x) - p(x)| < \varepsilon \quad \textit{pro } \forall x \in \langle a,b\rangle.$$

V obou případech ovšem nic nevíme o stupni n polynomu, chceme-li aproximaci s určitou přesností. Postačuje přitom pouhá spojitost: i funkci jako $|x|$ lze aproximovat polynomem libovolně přesně.

Budeme se zabývat především interpolací; k metodě nejmenších čtverců se vrátíme, až se budeme zabývat přeurčenými systémy lineárních rovnic.

3.2 Polynomiální interpolace

Polynomiální interpolace je nejjednodušší metodou aproximace funkce zadané v bodech $(x_i, f(x_i))$, $i = 0, 1, \ldots, n$. Jde ovšem o špatně podmíněnou úlohu a zvláště pro velká n můžeme dostat velmi „divoké“ polynomy (Rungeův jev). Využívá se také např. při odvozování vztahů pro numerickou aproximaci derivace funkce a integraci funkcí (viz později).

Věta 3.2.1 (Existence a jednoznačnost interpolačního polynomu)**.** *Máme-li $f(x)$ zadanou v $n+1$ bodech $(x_i, f(x_i))$, $i = 0, \ldots, n$, které jsou navzájem různé, pak existuje jediný polynom stupně n*

$$L_n(x) = a_0 + a_1 x + \cdots + a_n x^n, \tag{3.6}$$

který prochází všemi těmito body, tj. $L_n(x_i) = f(x_i)$.

Důkaz. Pro koeficienty a_i lze sestavit soustavu $n+1$ rovnic

$$Va = \underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix}}_{\text{tzv. Vandermondeova matice,}} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}, \quad \text{kde } f_i = f(x_i), \tag{3.7}$$

pro niž platí

$$\det V = \prod_{i>j} (x_i - x_j) \neq 0 \quad \text{pro } x_i \neq x_j \text{ pro } \forall i, j,$$

a tedy soustava má jediné řešení. □

Poznámka 3.2.2. Determinant $\det V$ je totiž polynomem v x_i a má nuly tam, kde je determinant nulový, tj. právě když pro jisté i a j platí $x_i = x_j$; takových možností je $n(n+1)/2$.

Polynom $L_n(x)$ se obvykle nazývá *Lagrangeův interpolační polynom* a obvykle se uvádí ve tvaru

$$L_n(x) = \sum_{i=0}^{n} f_i \ell_i(x), \quad \text{kde } \ell_i(x) = \frac{\prod\limits_{j=0, j\neq i}^{n} (x - x_j)}{\prod\limits_{j=0, j\neq i}^{n} (x_i - x_j)}, \quad \text{pro něž } \ell_i(x_j) = \delta_{ij}. \tag{3.8}$$

3.3 Aitkenovo–Nevillovo schéma

Chceme-li spočítat $L_n(x)$ v jednom nebo v malém počtu bodů, není nutné počítat koeficienty $L_n(x)$. Lze místo toho použít Aitkenovo–Nevillovo schéma, které počítá $L_n(\tilde{x})$ pro konkrétní $\tilde{x}$ přímo z hodnot x_i a f_i pomocí tabulky

			$\xrightarrow{k}$						
	stupeň	0		1		2		3	
	x_0	$f_0 = P_0^{(0)}$	$\searrow$						
	x_1	$f_1 = P_0^{(1)}$	$\rightarrow$	$P_1^{(01)}$	$\searrow$				
$i\downarrow$	x_2	$f_2 = P_0^{(2)}$	$\rightarrow$	$P_1^{(12)}$	$\rightarrow$	$P_2^{(012)}$	$\searrow$		
	x_3	$f_3 = P_0^{(3)}$	$\rightarrow$	$P_1^{(23)}$	$\rightarrow$	$P_2^{(123)}$	$\rightarrow$	$P_3^{(0123)}$	
	$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$	$\ddots$

Na i-tém řádku a v k-tém sloupci je hodnota

$$P_k^{(i-k,\ldots,i)} = L_k^{(i-k,\ldots,i)}(\tilde{x}) \tag{3.9}$$

Lagrangeova interpolačního polynomu v $x = \tilde{x}$ k-tého stupně daného body $(x_{i-k}, f_{i-k}), \ldots, (x_i, f_i)$. Přitom používáme rekurentní vztah

$$L_{k+1}^{(i-(k+1),\ldots,i-1,i)}(x) = \frac{(x - x_i) L_k^{(i-(k+1),\ldots,i-1)}(x) + (x_{i-(k+1)} - x) L_k^{(i-k,\ldots,i)}(x)}{x_{i-(k+1)} - x_i} \tag{3.10}$$

pro $k = 0, 1, \ldots, n$ a $i = 0, 1, 2, \ldots, i \geq k$.

Jde o vážený průměr dvou sousedních polynomů, který zajišťuje, že L_{k+1} prochází body $x_{i-(k+1)}, \ldots, x_i$. Například pro $P_3^{(0123)}$ se váhy kombinují takto:

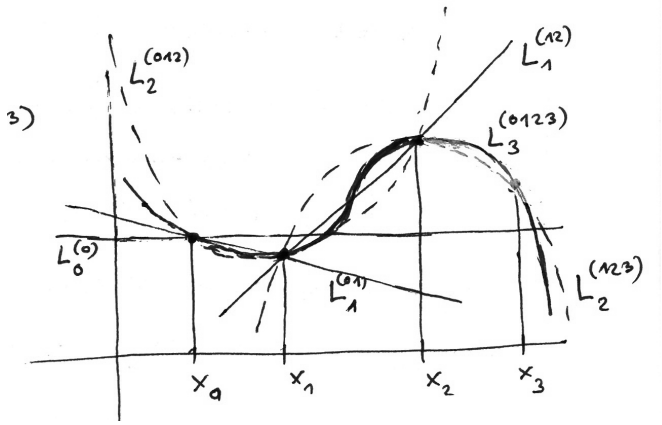
$$\begin{array}{lll} P_2^{(012)} & \searrow & \dfrac{(x - x_3)}{(x_0 - x_3)} \\ & & \qquad P_3^{(0123)} \\ P_2^{(123)} & \nearrow & \dfrac{(x_0 - x)}{(x_0 - x_3)} \end{array}$$

3.4 Chyba polynomiální interpolace a Čebyševovy uzly

Věta 3.4.1 (Chyba polynomiální interpolace). *Nechť $x_0, \ldots, x_n$ jsou různé body z $\langle a, b\rangle$ a $f \in C^{n+1}(\langle a, b\rangle)$. Pak pro každé $x \in \langle a, b\rangle$ existuje $\xi \in \langle a, b\rangle$ takové, že*

$$f(x) - L_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x), \tag{3.11}$$

kde $\omega_{n+1}(x) = \prod_{j=0}^{n} (x - x_j)$.



Obrázek 3.1: Interpolační polynomy vznikající v Aitkenově–Nevillově schématu pro uzly x_0, x_1, x_2, x_3: téměř vodorovná přímka $L_0^{(0)}$, přímky $L_1^{(01)}$ a $L_1^{(12)}$, čárkované paraboly $L_2^{(012)}$ a $L_2^{(123)}$ a tlustě vytažená kubika $L_3^{(0123)}$.

Důkaz. Viz Segethová. Využívá se Rolleova věta pro funkci $\tilde{\varphi}(x) = f(x) - L_n(x) - \tilde{k}\,\omega_{n+1}(x)$, kde

$$\tilde{K} = \frac{f(\tilde{x}) - L_n(\tilde{x})}{\omega_{n+1}(\tilde{x})} \quad \text{pro } \tilde{x} \neq x_i.$$

□

Chyba tedy prostřednictvím $\omega_{n+1}(x)$ závisí na rozložení bodů! Pro ekvidistantní rozložení x_i může nabývat mnohem větších hodnot než pro jiná rozložení, zvláště než pro optimální rozložení

$$x_i^{\text{čeb}} = \cos\frac{\pi}{2}\,\frac{2i+1}{n+1} \qquad \text{na int. } \langle -1, 1\rangle, \tag{3.12}$$

což jsou kořeny Čebyševova polynomu T_{n+1}, neboli po přeškálování na interval $\langle a, b\rangle$ pro

$$x_i^{\text{opt}} = \frac{a+b}{2} + \frac{(b-a)}{2}\cos\frac{\pi\,(2i+1)}{2\,(n+1)}, \tag{3.13}$$

kdy

$$\omega_{n+1}(x) = \left(\frac{b-a}{2}\right)^{n+1} t_{n+1}\left(\left(x - \frac{a+b}{2}\right)\frac{2}{b-a}\right),$$

kde t_{n+1} je Čebyševův polynom s koeficientem 1 u x^{n+1}, a maximum je

$$\max_{x\in\langle a,b\rangle} |\omega_{n+1}(x)| = \frac{(b-a)^{n+1}}{2^{2n+1}}. \tag{3.14}$$

3.4.1 Proč zrovna kořeny Čebyševova polynomu?

Čebyševův polynom $t_n(x) = \dfrac{1}{2^{n-1}}T_n(x)$ s koeficientem 1 u x^n má totiž tu význačnou vlastnost, že minimalizuje ∞-normu na $\langle -1, 1\rangle$ mezi polynomy p_n stupně n, které mají u x^n koeficient 1.

Věta 3.4.2 (Minimalita Čebyševova polynomu)**.** *Pro každý polynom p_n stupně n s koeficientem 1 u x^n platí*

$$\max_{x\in\langle -1,1\rangle} |p_n(x)| \geq \max_{x\in\langle -1,1\rangle} |t_n(x)| = \frac{1}{2^{n-1}}. \tag{3.15}$$

Důkaz. Z vyjádření $t_n(x) = \frac{1}{2^{n-1}} \cos(n \arccos x)$ (viz poznámky o ortogonálních polynomech) plyne

$$|t_n(x)| \leq \frac{1}{2^{n-1}} \quad \text{pro } x \in \langle -1, 1 \rangle.$$

Přitom body, kde platí rovnost, jsou ty, pro které bude $n \arccos x_k = k\pi$, $k = 0, \ldots, n$, neboli

$$x_k = \cos \frac{k\pi}{n}, \qquad \text{a tedy} \qquad t_n(x_k) = \frac{(-1)^k}{2^{n-1}}.$$

Nechť nyní existuje $q_n(x)$ mezi všemi $p_n(x)$ takový, že

$$\max_{x \in \langle -1,1 \rangle} |q_n(x)| < \frac{1}{2^{n-1}}.$$

Pak $r(x) = t_n(x) - q_n(x)$ má stupeň menší než n a je nenulový, neboť v bodech x_k je nenulový a navíc má v x_k stejné znaménko jako t_n, a tedy by měl mít n kořenů, což je spor s předpokladem, že má stupeň $< n$. □

3.5 Hermiteova interpolace

Kromě funkčních hodnot zadáváme i derivace do řádu $\alpha_i - 1$ v bodě x_i. Dostaneme tak polynom $H_m(x)$ m-tého stupně, přičemž $m = \sum_{i=0}^{n} \alpha_i - 1$. Koeficienty $H_m(x)$ by šlo opět určit řešením soustavy $m+1$ rovnic pro $m+1$ neznámých.

Pro $f \in C^{m+1}(\langle a, b \rangle)$ je chyba dána vztahem

$$f(x) - H_m(x) = \frac{f^{(m+1)}(\xi)}{(m+1)!} \omega_{m+1}(x) \quad \text{pro jisté } \xi \in \langle a, b \rangle, \tag{3.16}$$

kde nyní

$$\omega_{m+1}(x) = (x - x_0)^{\alpha_0} (x - x_1)^{\alpha_1} \ldots (x - x_n)^{\alpha_n}.$$

Pro $\alpha_0 = \alpha_1 = \cdots = \alpha_n = 1$ by šlo o standardní Lagrangeovu interpolaci; pro $n = 0$ a $\alpha_0 = m + 1$ dostaneme Taylorův rozvoj v okolí x_0 do řádu m.

Speciálně pro $\alpha_0 = \cdots = \alpha_n = 2$, kdy zadáváme funkční hodnoty a první derivace, lze psát

$$H_{2n+1}(x) = \sum_{i=0}^{n} s_i(x) f_i + \sum_{i=0}^{n} t_i(x) f_i', \tag{3.17}$$

kde

$$\begin{aligned} s_i(x) &= \left[1 - 2(x - x_i)\ell_i'(x_i)\right] \ell_i^2(x) && \Rightarrow && s_i(x_j) = \delta_{ij}, && s_i'(x_j) = 0, \\ t_i(x) &= (x - x_i)\ell_i^2(x) && \Rightarrow && t_i(x_j) = 0, && t_i'(x_j) = \delta_{ij}. \end{aligned}$$

3.6 Interpolace pomocí splajnů

Namísto jediného interpolačního polynomu procházejícího body $(x_i, f_i)_{i=0,\ldots,n}$, který může značně oscilovat (Rungeův jev), budeme hledat po částech polynomy nižších řádů na každém podintervalu $\langle x_{i-1}, x_i \rangle$, $i = 1, \ldots, n$. Přitom budeme chtít, aby byla výsledná funkce (splajn) spojitá i s derivacemi do jistého řádu.

Název splajn pochází z anglického *spline*, které označuje dřevěný pás používaný k prokládání křivek mezi body při konstrukcích lodí.

Definice 3.6.1 (Obyčejný splajn řádu k)**.** Obyčejný splajn řádu k pro uzly $x_0 < x_1 < \cdots < x_n$ je funkce, která je na $\forall\langle x_{i-1}, x_i\rangle$, $i = 1, \ldots, n$, polynomem stupně nejvýše k a která má na celém intervalu $\langle x_0, x_n\rangle$ spojité derivace do řádu $k-1$.

Splajny se používají nejen k interpolaci, ale také např. ke konstrukci bází s kompaktním nosičem (B-splajny) pro řešení PDR apod.

V praxi se využívají především kubické splajny pro své dobré „vyhlazovací“ vlastnosti,

$$\int_{x_0}^{x_n} |S''(x)|^2 \mathrm{d}x \leq \int_{x_0}^{x_n} |\psi''(x)|^2 \mathrm{d}x \quad \text{pro } \forall \psi \in C^{(n)}_{\langle x_0, x_n\rangle} \text{ pro něž } \psi(x_i) = \underset{i=0,\ldots,n}{S(x_i)},$$

a zároveň pro relativně jednoduchou konstrukci.

Výhodou oproti Lagrangeovým interpolačním polynomům je, že pro $f \in C^4_{\langle x_0, x_n\rangle}$ splajny stejnoměrně konvergují k interpolované funkci f, pokud zmenšujeme $\max\limits_{i=1,\ldots,n} |x_{i-1} - x_i|$. Pro ekvidistantní uzly lze ukázat, že existuje konstanta C nezávislá na x taková, že

$$|f(x) - S(x)| \leq Ch^4 \quad \text{pro } \forall x \in \langle x_0, x_n\rangle. \tag{3.18}$$

3.6.1 Interpolační kubický splajn

Definice 3.6.2 (Interpolační kubický splajn)**.** Interpolační kubický splajn je splajn 3. řádu procházející interpolovanými body (x_i, f_i), $i = 0, \ldots, n$.

Interpolační kubický splajn není určen jednoznačně: máme $4n-2$ podmínek

$$\begin{aligned} S_i(x_{i-1}) &= f_{i-1}, \\ S_i(x_i) &= f_i, \\ S_i'(x_i) &= S_{i+1}'(x_i), \\ S_i''(x_i) &= S_{i+1}''(x_i), \end{aligned} \tag{3.19}$$

přičemž poslední dvě podmínky se berou bez krajních bodů x_0 a x_n, ale $4n$ neznámých koeficientů polynomů stupně 3 na každém $\langle x_{i-1}, x_i\rangle$. Jsou tedy nutné dodatečné 2 podmínky.

Přirozený (natural) interpolační kubický splajn je dán podmínkami

$$S''(x_0) = S''(x_n) = 0. \tag{3.20}$$

Obyčejný (clamped, upnutý, tj. na okrajích nasměrujeme) interpolační kubický splajn je dán podmínkami

$$S'(x_0) = f'(x_0), \quad S'(x_n) = f'(x_n). \tag{3.21}$$

Poznámka 3.6.3. Splajn s těmito okrajovými podmínkami přechází na koncích do rovné čáry.

Používají se i další varianty, např. periodický splajn:

$$\begin{aligned} S'(x_0) &= S'(x_n) \\ S''(x_0) &= S''(x_n) \end{aligned} \quad + \quad \begin{aligned} &\text{musí být} \\ &S(x_0) = S(x_n). \end{aligned}$$

3.6.2 Přirozený kubický splajn procházející body (x_i, f_i)

Bylo by možné z podmínek (3.19) sestavit $4n$ rovnic, ovšem efektivnější je jít na to nepřímo:

a) určíme druhé derivace $z_i = S''(x_i)$ z tridiagonální soustavy lineárních rovnic;

b) určíme $S(x)$ ze známých x_i, f_i a z_i, $i = 0, \ldots, n$.

Rovnice pro druhé derivace z_i

Pokud by byly druhé derivace z_i známé, pak $S_i''(x)$ je lineární funkce

$$S_i''(x) = \frac{z_i}{h_i}(x - x_{i-1}) + \frac{z_{i-1}}{h_i}(x_i - x), \qquad \text{kde } h_i = x_i - x_{i-1}, \quad i = 1, \dots, n. \tag{3.22}$$

Dvojí integrací dostaneme (s vhodně zvolenými integračními konstantami)

$$S_i(x) = \frac{z_i}{6h_i}(x - x_{i-1})^3 + \frac{z_{i-1}}{6h_i}(x_i - x)^3 + C_i(x - x_{i-1}) + D_i(x_i - x), \tag{3.23}$$

kde integrační konstanty C_i a D_i se určí z podmínek $S_i(x_{i-1}) = f_{i-1}$ a $S_i(x_i) = f_i$, takže

$$C_i = \frac{f_i}{h_i} - \frac{z_i h_i}{6}, \qquad D_i = \frac{f_{i-1}}{h_i} - \frac{z_{i-1} h_i}{6}.$$

Známe-li tedy z_i, můžeme určit $S(x)$ z rovnice (3.23) pro $x \in \langle x_{i-1}, x_i \rangle$, i když je vhodnější vyčíslit ji jako

$$S_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \tag{3.24}$$

kde

$$\begin{aligned} a_i &= f_{i-1}, \\ b_i &= -\frac{h_i}{3}\left(z_{i-1} + \frac{z_i}{2}\right) + \frac{f_i - f_{i-1}}{h_i}, \\ c_i &= z_{i-1}/2, \\ d_i &= \frac{z_i - z_{i-1}}{6h_i}. \end{aligned}$$

Takto zkonstruované $S_i(x)$ zatím splňuje $S_i(x_{i-1}) = f_{i-1}$, $S_i(x_i) = f_i$, $i = 0, \dots, n$, a

$$S_i''(x_i) = S_{i+1}''(x_i), \qquad i = 1, \dots, n-1.$$

Nyní použijeme podmínku $S_i'(x_i) = S_{i+1}'(x_i)$, $i = 1, \dots, n-1$, k určení z_i. Dostaneme

$$S_i'(x) = \frac{z_i}{2h_i}(x - x_{i-1})^2 - \frac{z_{i-1}}{2h_i}(x_i - x)^2 + C_i - D_i$$

a tedy

$$z_{j-1}h_j + 2z_j(h_i + h_{i+1}) + z_{j+1}h_{i+1} = 6(b_{i+1} - b_j), \tag{3.25}$$

kde

$$b_i = \frac{f_i - f_{i-1}}{h_i} \qquad \text{pro} \qquad i = 1, \dots, n-1. \tag{3.26}$$

Navíc položíme $z_0 = 0$ a $z_n = 0$, čímž dostaneme tridiagonální (a navíc symetrický) systém

$$\begin{pmatrix} U_1 & h_2 & & & & 0 \\ h_2 & U_2 & h_3 & & & \\ & h_3 & \ddots & & & \\ & & & \ddots & & \\ 0 & & & h_{n-2} & U_{n-2} & h_{n-1} \\ & & & & h-1 & U_{n-1} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ \vdots \\ z_{n-1} \end{pmatrix} = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ \vdots \\ v_{n-1} \end{pmatrix}, \tag{3.27}$$

kde

$$\begin{aligned} U_i &= 2(h_i + h_{i+1}), \\ v_i &= 6(b_{j+1} - b_i). \end{aligned}$$

Systém lze řešit přímou Gaussovou eliminací na $\mathcal{O}(n)$ operací: hodnotu U_i „nahradíme“ hodnotou

$$\begin{aligned} U_i &\longleftarrow U_i - \frac{h_i^2}{U_{i-1}}, \\ V_i &\longleftarrow V_i - \frac{h_i V_{i-1}}{U_{i-1}}, \end{aligned}$$

a to pro $i = 2, \ldots, n-1$, a zpětnou substitucí dostaneme

$$\begin{aligned} z_{n-1} &= \frac{V_{n-1}}{U_{n-1}}, \\ z_i &= \frac{V_i - h_{i+1} z_{i+1}}{U_i}, \qquad i = n-2, n-3, \ldots, 1. \end{aligned}$$

Vše je v pořádku, neboť $U_i > h_{i+1} > 0$. Platí totiž $U_1 > h_2 > 0$ a indukcí

$$U_i = 2(h_{i+1} + h_i) - \frac{h_i^2}{U_{i-1}} > 2(h_{i+1} + h_i) - h_i > h_{i+1} > 0.$$

Kapitola 4

Ortogonální polynomy

4.1 Ortogonální polynomy

Zde uvedeme jen základní výsledky pro ortogonální polynomy (dále též OP) jedné proměnné. Existuje ovšem i zobecnění na polynomy ve více proměnných, viz např. Yuan Xu: *On Orthogonal Polynomials in Several Variables.*

Definice 4.1.1 (Ortogonální polynomy na intervalu $\langle a, b\rangle$ s vahou $v(x)$)**.** Ortogonální polynomy na intervalu $\langle a, b\rangle$ s vahou $v(x)$ jsou posloupností polynomů $\{p_n(x)\}_{n=0}^{\infty}$ takovou, že $p_n(x)$ je stupně n a všechny jsou navzájem ortogonální vzhledem k váhové funkci $v(x)$, která je kladná skoro všude a pro kterou existují integrály (momenty)

$$\int_a^b x^k v(x)\,\mathrm{d}x < +\infty, \qquad k = 0, 1, 2, \ldots,$$

tj. platí

$$(p_n, p_m) \equiv \int_a^b p_n(x)\, p_m(x)\, v(x)\,\mathrm{d}x = 0 \quad \text{pro } m \neq n. \tag{4.1}$$

Že taková posloupnost polynomů existuje, lze ukázat přímou konstrukcí pomocí Gramovy–Schmidtovy ortogonalizace aplikované na polynomy $1, x, x^2, \ldots$ se skalárním součinem (p, q). Díky linearitě je polynom $p_n(x)$ ortogonální k *libovolnému* polynomu stupně $m < n$.

4.2 Rekurentní vztah

Věta 4.2.1 (Rekurentní vztah pro ortogonální polynomy)**.** *Polynomy $p_n(x)$ normalizované tak, že koeficient u x^n je 1, splňují rekurentní vztah*

$$p_{n+1}(x) = (x - \delta_{n+1})\, p_n(x) - \gamma_{n+1}^2\, p_{n-1}(x) \tag{4.2}$$

pro $n \geq 0$, přičemž položíme $p_0(x) = 1$ a $p_{-1}(x) = 0$, kde

$$\delta_{n+1} = \frac{(xp_n, p_n)}{(p_n, p_n)} \qquad a \qquad \gamma_{n+1}^2 = \begin{cases} \dfrac{(p_n, p_n)}{(p_{n-1}, p_{n-1})} > 0 & \text{pro } n > 0, \\ 0 & \text{pro } n = 0. \end{cases} \tag{4.3}$$

Důkaz. Součin $xp_n(x)$ musí být lineární kombinací pouze tří ortogonálních polynomů,

$$xp_n(x) = a_{n-1}p_{n-1}(x) + a_n p_n(x) + a_{n+1}p_{n+1}(x), \tag{4.4}$$

neboť $(xp_n, p_m) = (p_n, xp_m) = 0$ pro $m < n-1$. Navíc $a_{n+1} = 1$ (aby u x^{n+1} byl koeficient 1) a koeficienty a_{n-1} a a_n se už jednoduše určí projekcí na p_{n-1} a p_n. Přitom

$$a_{n+1} = 1 = \frac{(p_{n+1}, xp_n)}{(p_{n+1}, p_{n+1})}.$$

□

Tento rekurentní vztah umožňuje přímo konstruovat ortogonální polynomy na základě tzv. *momentů*

$$\mu_k = \int_a^b x^k v(x)\,\mathrm{d}x, \qquad k = 0, 1, 2, \ldots \tag{4.5}$$

Například

$$p_1(x) = (x - \delta_1)\,p_0(x) = x - \frac{(xp_0, p_0)}{(p_0, p_0)} = x - \frac{\mu_1}{\mu_0}$$

a dále

$$p_2(x) = (x - \delta_2)\,p_1(x) - \gamma_2^2\,p_0(x),$$

kde

$$\delta_2 = \frac{(xp_1, p_1)}{(p_1, p_1)} = \frac{\mu_3\mu_0^2 - 2\mu_1\mu_2\mu_0 + \mu_1^3}{\mu_0^2\mu_2 - \mu_1^2\mu_0},$$

$$\gamma_2^2 = \frac{(p_1, p_1)}{(p_0, p_0)} = \frac{\mu_2\mu_0 - \mu_1^2}{\mu_0^2},$$

a tak dále.

Poznámka 4.2.2. Odtud je patrná úzká spojitost s momentovou teorií, která studuje tzv. momentové funkcionály $\mathcal{L}[f]$ na prostorech polynomů dané obecně komplexními čísly $\mu_k = \mathcal{L}[x^k]$.

4.3 Christoffelova–Darbouxova identita

Věta 4.3.1 (Christoffelova–Darbouxova identita)**.** *Nechť polynomy $\{p_n(x)\}_{n=0}^{\infty}$ splňují rekurentní vztah*

$$p_{n+1}(x) = (x - \delta_{n+1})\,p_n(x) - \lambda_{n+1}\,p_{n-1}(x), \tag{4.6}$$

tj. $\lambda_{n+1} = \gamma_{n+1}^2$, kde $\lambda_n > 0$ pro $n \geq 1$ a $\lambda_1 = \mu_0$ (což platí pro $v(x)$ kladné skoro všude). Pak

$$\sum_{k=0}^{n} \frac{p_k(x)\,p_k(u)}{\lambda_1\lambda_2\cdots\lambda_{k+1}} = (\lambda_1\lambda_2\cdots\lambda_{n+1})^{-1}\,\frac{p_{n+1}(x)\,p_n(u) - p_n(x)\,p_{n+1}(u)}{x - u}. \tag{4.7}$$

Důkaz. Tvrzení plyne z (4.6) tak, že odečteme

$$(*)(x)\cdot p_n(u) - (*)(u)\cdot p_n(x),$$

kde $(*)$ značí vztah (4.6) a $(*)(u)$ znamená tentýž vztah, v němž $x \to u$. Tím dostaneme

$$p_k(x)\,p_k(u) = \frac{p_{k+1}(x)\,p_k(u) - p_k(x)\,p_{k+1}(u)}{x - u} - \lambda_{k+1}\,\frac{p_k(x)\,p_{k-1}(u) - p_{k-1}(x)\,p_k(u)}{x - u}. \tag{4.8}$$

Vynásobením $(\lambda_1\lambda_2\cdots\lambda_{k+1})^{-1}$ a sečtením těchto rovnic pro $k = 0, \ldots, n$ dostaneme Christoffelovu–Darbouxovu identitu (4.7). □

V limitě $x \to u$ odtud plyne

$$\sum_{k=0}^{n} \frac{p_k^2(x)}{\lambda_1\lambda_2\cdots\lambda_{k+1}} = \frac{p'_{n+1}(x)\,p_n(x) - p'_n(x)\,p_{n+1}(x)}{\lambda_1\lambda_2\cdots\lambda_{n+1}}, \tag{4.9}$$

kde

$$p'_{n+1}(x) = \lim_{x\to u} \frac{p_{n+1}(x) - p_{n+1}(u)}{x-u}$$

a podobně pro $p'_n(x)$.

Důsledek 4.3.2. *Pro $\forall x \in \langle a,b\rangle$ platí*

$$p'_{n+1}(x)\,p_n(x) - p'_n(x)\,p_{n+1}(x) > 0. \tag{4.10}$$

4.4 Kořeny ortogonálních polynomů

Věta 4.4.1 (Poloha kořenů)**.** *Všechny kořeny x_j, $j = 1,\ldots,n$, polynomu $p_n(x)$ jsou reálné, jednoduché a leží v intervalu (a,b).*

Důkaz. Označme

$$q(x) = \prod_{j\in L}(x - x_j),$$

kde L jsou indexy kořenů $x_j \in (a,b)$, které jsou navzájem různé a liché násobnosti (což znamená, že $p_n(x)$ v x_j mění znaménko). Pak součin $p_n(x)q(x)$ na $\langle a,b\rangle$ znaménko nemění (nakreslete si příklad :-), a tedy

$$(p_n, q) = \int_a^b p_n(x)\,q(x)\,v(x)\,\mathrm{d}x \neq 0.$$

Z toho ovšem plyne, že $q(x)$ musí být polynom stupně n (neboť všechny polynomy stupně $m < n$ jsou ortogonální na $p_n(x)$), a musí tedy mít n kořenů v intervalu (a,b), a tedy i $p_n(x)$. □

Věta 4.4.2 (Prokládání kořenů)**.** *Kořeny $p_n(x)$ leží mezi kořeny $p_{n+1}(x)$.*

Důkaz. Do vztahu (4.10), tj. $p'_{n+1}(x)\,p_n(x) - p'_n(x)\,p_{n+1}(x) > 0$, dosadíme kořen $x_k^{(n+1)}$ polynomu $p_{n+1}(x)$. Dostaneme

$$p'_{n+1}\Big(x_k^{(n+1)}\Big)\,p_n\Big(x_k^{(n+1)}\Big) > 0 \qquad \text{pro } k = 1,2,\ldots,n+1. \tag{4.11}$$

Z toho plyne, že $p_n(x)$ musí změnit znaménko mezi dvěma kořeny $p_{n+1}(x)$, a má tedy kořen mezi $x_k^{(n+1)}$ a $x_{k+1}^{(n+1)}$. □

Kořeny $x_k^{(n)}$ lze hledat buď přímo díky předchozímu výsledku, například kombinací metody půlení intervalu a rychlého dokonvergování Newtonovou metodou, nebo pomocí diagonalizace tridiagonální matice

$$J_N = \begin{pmatrix} \delta_1 & \gamma_2 & & & \\ \gamma_2 & \delta_2 & \gamma_3 & & 0 \\ & \gamma_3 & \delta_3 & & \\ & & & \ddots & \gamma_N \\ 0 & & & \gamma_N & \delta_N \end{pmatrix}, \tag{4.12}$$

neboť $\det(\lambda\mathbf{1} - J_N)$ splňuje stejný rekurentní vztah jako $p_n(x)$:

$$\det(\lambda - J_N) = \det(\lambda - J_{N-1})\,(\lambda - \delta_N) - \gamma_N^2\,\det(\lambda - J_{N-2}). \tag{4.13}$$

Kořeny $p_N(x)$ jsou tedy zároveň vlastními čísly matice J_N.

Poznámka 4.4.3. Existují speciální velmi rychlé algoritmy pro diagonalizaci tridiagonálních matic, takže takto se kořeny často hledají.

Kapitola 5

Numerická derivace

5.1 Reprezentace funkce a výpočet derivace

Výpočet derivace v počítači závisí na reprezentaci funkce v počítači. Často máme například funkce reprezentované jako rozvoj do určité (analytické) báze $\phi_i(x)$,

$$f(x) \simeq \sum_{i=1}^{n} c_i \phi_i(x), \tag{5.1}$$

a derivace pak můžeme počítat přímo pomocí

$$\left.\frac{\mathrm{d}^m f(x)}{\mathrm{d}x^m}\right|_{x=z} \approx \sum_{i=1}^{n} c_i \left.\frac{\mathrm{d}^m \phi_i(x)}{\mathrm{d}x^m}\right|_{x=z}, \tag{5.2}$$

pokud lze derivace báze rozumně počítat (např. pro polynomy apod.).

Pokud máme funkci zadanou body a funkčními hodnotami, pak je nutné funkci nejprve aproximovat, a to buď pomocí (5.1), např. metodou nejmenších čtverců, nebo interpolací, nejčastěji polynomiální. Teprve tuto aproximaci pak derivujeme.

I tehdy, když můžeme funkci $f(x)$ vyčíslit v libovolném bodě, ale ne už její derivaci (např. je-li $f(x)$ řešením složitého problému), není vhodné používat nejjednodušší aproximace derivací jako např.

$$\left.\frac{\mathrm{d}f(x)}{\mathrm{d}x}\right|_{x=x_0} \approx \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_0 + h) - f(x_0)}{h}, \tag{5.3}$$

kdy volíme x_1 co nejblíže x_0, neboli volíme h malé. Vznikají totiž *problémy se zaokrouhlovací chybou* při odečítání dvou blízkých čísel. Chyba se v tomto případě chová jako $\mathcal{O}(h)$ (viz později), a potřebujeme-li vysokou přesnost, je nutné volit velmi malé h; zaokrouhlovací chyba pak může výsledek hodně ovlivnit.

Je tedy vhodné použít *vícebodové aproximace*, u kterých se chyba chová jako $\mathcal{O}(h^p)$ s $p > 1$, a není pak nutné volit tak malá h.

5.2 Derivace interpolačního polynomu

Mějme $f(x)$ zadanou v bodech $x_0, \ldots, x_n$ a k její aproximaci použijme polynomiální interpolaci

$$f(x) \approx L_n(x) = \sum_{i=0}^{n} \ell_{n,i}(x)\, f(x_i),$$

kde $\ell_{n,i}(x)$ lze vyjádřit jako

$$\ell_{n,i}(x) = \frac{\omega_n(x)}{(x - x_i)\, \omega_n'(x_i)},$$

přičemž

$$\omega_n(x) = \prod_{j=0}^{n}(x - x_j) \quad \text{a} \quad \omega_n'(x) = \sum_{i=0}^{n}\prod_{\substack{j=0\\ j\neq i}}^{n}(x - x_j),$$

a tedy

$$\omega_n'(x_k) = \prod_{\substack{j=0\\ j\neq k}}^{n}(x_k - x_j).$$

Derivováním $L_n(x)$ dostaneme aproximaci

$$\left.\frac{\mathrm{d}^m f}{\mathrm{d}x^m}\right|_{x=z} \approx \sum_{i=0}^{n} f(x_i)\left.\frac{\mathrm{d}^m \ell_{n,i}(x)}{\mathrm{d}x^m}\right|_{x=z} = \sum_{i=0}^{n} f(x_i)\, C_{n,i}^m(z). \tag{5.4}$$

Koeficienty $C_{n,i}^m(z)$ závisí jak na počtu použitých bodů n, tak na bodu z, ve kterém derivaci počítáme. Lze je určit různými způsoby, které rozebereme v následujících odstavcích.

5.2.1 Přímá derivace bázových polynomů

První možností je přímá derivace $\ell_{n,i}(x)$. Není příliš efektivní, neboť vyžaduje konstrukci polynomů $\ell_{n,i}(x)$.

5.2.2 Řešení soustavy lineárních rovnic z Taylorových rozvojů

Druhou možností je řešení soustavy lineárních rovnic, které vychází z Taylorových rozvojů $f(x)$ v bodech x_i. Ani tento postup není příliš efektivní; jeho výhodou ovšem je, že spolu s koeficienty dostáváme i chybové členy.

Příklad 5.2.1. Rozvineme-li

$$f(x_0) = f(z) + \frac{f'(z)}{1!}(x_0 - z) + \frac{f''(z)}{2!}(x_0 - z)^2 + \mathcal{O}\big((x_0 - z)^3\big),$$
$$f(x_1) = f(z) + f'(z)(x_1 - z) + \frac{f''(z)}{2!}(x_0 - z)^2 + \mathcal{O}\big((x_1 - z)^3\big)$$

a vyřešíme-li tuto soustavu vůči $f(z)$ a $f'(z)$, dostaneme pro $f(z)$ lineární interpolaci funkce $f(x)$,

$$f(z) = \frac{x_1 - z}{x_1 - x_0}f(x_0) + \frac{z - x_0}{x_1 - x_0}f(x_1) + \frac{f''(z)}{2}(x_1 - z)(z - x_0), \tag{5.5}$$

a pro derivaci

$$f'(z) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} - \frac{f''(z)}{2}\underbrace{(x_1 + x_0 - 2z)}_{\substack{\pm h \text{ pro } z=x_0 \text{ nebo } z=x_1\\ 0 \text{ pro } z=\frac{x_0+x_1}{2}}}. \tag{5.6}$$

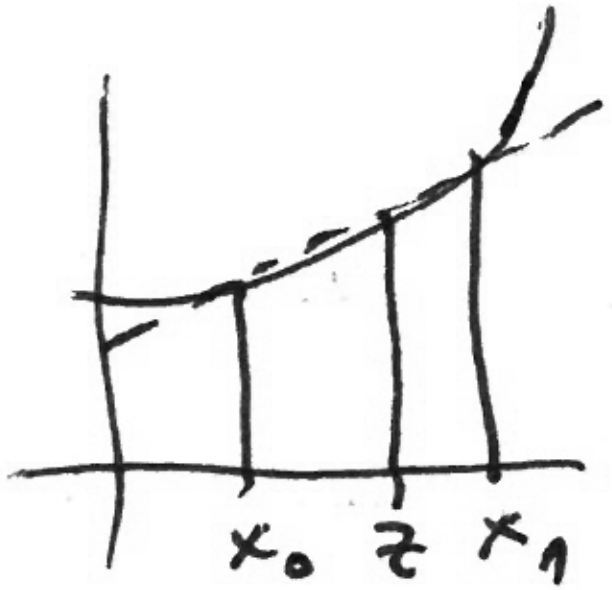
Pro z uprostřed intervalu tedy vedoucí chybový člen vymizí a chyba je $\mathcal{O}(h^2)$.

Více viz demonstrace v Mathematice.

5.2.3 Fornbergovy rekurentní vztahy

Efektivní výpočet koeficientů $C_{n,i}^m(z)$ navrhl Fornberg v Math. Comp. 51 (1988) 699. Vychází z rekurentních vztahů pro $\omega_n(x)$,

$$\omega_n(x) = (x - x_n)\,\omega_{n-1}(x),$$
$$\omega_n'(x) = (x - x_n)\,\omega_{n-1}'(x) + \omega_{n-1}(x),$$



Obrázek 5.1: Náčrt souřadnicových os s rostoucí křivkou $f(x)$ a čárkovanou přímkou (lineární interpolací) protínající ji; na ose x jsou svislými čarami vyznačeny body x_0, z, x_1.

z nichž plynou vztahy

$$\begin{aligned} \ell_{n,i}(x) &= \frac{(x-x_n)}{(x_i-x_n)}\,\ell_{n-1,i} \quad \text{pro } 0 \le i \le n-1, \\ \text{a} \quad \ell_{n,n}(x) &= \frac{\omega_{n-2}(x_{n-1})}{\omega_{n-1}(x_n)}\,(x-x_{n-1})\,\ell_{n-1,n-1}\,. \end{aligned} \tag{5.7}$$

Vyjádříme-li nyní $\ell_{n,i}(x)$ pomocí Taylorova rozvoje v bodě z jako

$$\ell_{n,i}(x) = \sum_{m=0}^{n} \frac{C_{n,i}^m(z)}{m!}\,(x-z)^m$$

a dosadíme do (5.7), pak porovnáním členů u $(x-z)^k$ nakonec dostaneme

$$C_{n,i}^m = \frac{1}{x_n - x_i}\Big[(x_n - z)\,C_{n-1,i}^m - m\,C_{n-1,i}^{m-1}\Big] \quad \text{pro } 0 \le i \le n-1 \text{ a } n \ge m \ge 1 \tag{5.8}$$

a

$$C_{n,n}^m = \frac{\omega_{n-2}(x_{n-1})}{\omega_{n-1}(x_n)}\Big[m\,C_{n-1,n-1}^{m-1} - (x_{n-1}-z)\,C_{n-1,n-1}^m\Big], \tag{5.9}$$

a speciálně

$$\begin{aligned} C_{n,i}^0 &= \frac{x_n - z}{x_n - x_i}\,C_{n-1,i}^0 \qquad \text{a } C_{0,0}^0(z) = 1, \\ C_{n,n}^0 &= \frac{\omega_{n-2}(x_{n-1})}{\omega_{n-1}(x_n)}\,(z-x_{n-1})\,C_{n-1,n-1}^0, \end{aligned}$$

kde $\omega_{-1}(x) = 1$ a $C_{n,i}^m = 0$ pro $m > n$.

Poznámka 5.2.2. Tyto rekurentní vztahy používá Mathematica.

Chyba takto získaných aproximací je obecně $\mathcal{O}(h^{n+1-m})$ pro m-tou derivaci a $(n+1)$-bodovou formuli, ale u symetrických vztahů je obvykle o řád lepší.

5.2.4 Rovnoměrné sítě

Pro rovnoměrné sítě (ekvidistantní grid), kdy

$$x_i = x_0 + ih \quad , \quad i = 1, \ldots, n, \tag{5.10}$$

jsou koeficienty jednodušší a zvláště pro derivace v bodech x_i jsou pouze funkcemi kroku h (viz později).

5.2.5 Richardsonova extrapolace

K odvození přesnějších vztahů pro derivaci lze použít i tzv. *Richardsonovu extrapolaci*, kdy obecně ze známého chování chyby extrapolujeme k $h \to 0$.

Příklad 5.2.3. Centrovaná diference v bodě x_i aproximuje první derivaci s chybou $\mathcal{O}(h^2)$. Označíme-li D_h operátor centrované diference s krokem h, můžeme psát

$$D_h f(x_i) = \frac{f_{i+1} - f_{i-1}}{2h} = f'(x_i) + ch^2 + \mathcal{O}(h^4),$$
$$D_{2h} f(x_i) = \frac{f_{i+2} - f_{i-2}}{4h} = f'(x_i) + c\,4h^2 + \mathcal{O}(h^4),$$

kde c je neznámá konstanta. Odtud

$$\begin{aligned} f'(x_i) + \mathcal{O}(h^4) &= \frac{4D_h f(x_i) - D_{2h} f(x_i)}{3} \\ &= \frac{4}{6h}\left(f_{i-1} - f_{i-1}\right) - \frac{1}{12h}\left(f_{i+2} - f_{i-2}\right) \\ &= \frac{1}{12h}\left(-f_{i+2} + 8f_{i+1} - 8f_{i-1} + f_{i-2}\right). \end{aligned} \tag{5.11}$$

Výraz (5.11) je speciálním případem obecné Richardsonovy extrapolace

$$\left.\begin{aligned} A_h &= A + Ch^k + \mathcal{O}(h^{k+1}) \\ A_{2h} &= A + C(2h)^k + \mathcal{O}(h^{k+1}) \end{aligned}\right\} \quad \Rightarrow \quad A + \mathcal{O}(h^{k+1}) = \frac{2^k A_h - A_{2h}}{2^k - 1}. \tag{5.12}$$

5.3 Optimální volba kroku h

Na přesnost výpočtu derivace mají vliv dvě chyby.

1. *Chyba aproximace* – závisí na použitém vzorci a je typicky $\mathcal{O}(h^k)$; např. $f_0' \approx \frac{f_1 - f_0}{h} + \mathcal{O}(h)$, přesněji je vedoucí člen roven $\frac{hf''}{2}$.

2. *Chyba zaokrouhlovací* – odhad absolutní chyby odečítání $f(x+h) - f(x)$ je $\frac{2\varepsilon|f|}{h}$, kde ε je strojové epsilon.

Optimální volba h je tedy taková, že jsou obě chyby srovnatelné:

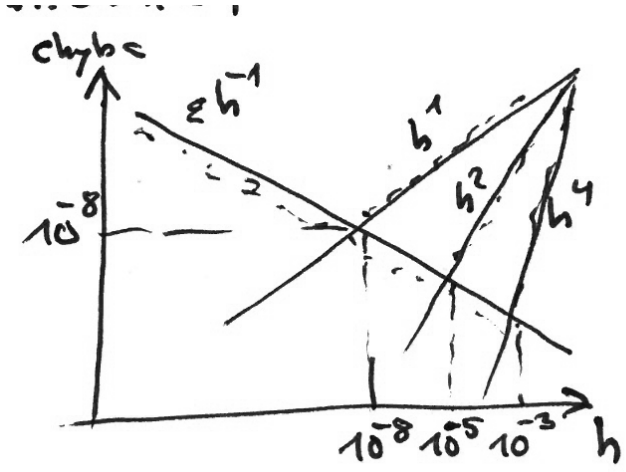
$$\frac{h_{\text{opt}}|f''|}{2} \simeq \frac{2\varepsilon|f|}{h_{\text{opt}}} \quad \Rightarrow \quad h_{\text{opt}} \simeq 2\sqrt{\frac{|f|}{|f''|}}\,\sqrt{\varepsilon}, \tag{5.13}$$

což pro dvojitou přesnost dává $h_{\text{opt}} \sim 10^{-8}$.

Poznámka 5.3.1. Výraz $\sqrt{|f|/|f''|}$ představuje jistou charakteristickou škálu, na které se f mění.

Podobně dostaneme

$$\begin{aligned} \mathcal{O}(h^2) \to h_{\text{opt}} \approx \sqrt[3]{\varepsilon} \approx 10^{-5} \quad &\Rightarrow \quad h^2 \approx 10^{-10}, \\ \mathcal{O}(h^4) \to h_{\text{opt}} \approx \sqrt[5]{\varepsilon} \approx 10^{-3} \quad &\qquad h^4 \approx 10^{-12}. \end{aligned}$$



Obrázek 5.2: Log-log graf závislosti chyby na kroku h. Svislá osa je popsána „chyba“, vodorovná „h“. Klesající přímka je označena εh^{-1}, tři rostoucí přímky jsou označeny h^1, h^2 a h^4. Na svislé ose je vyznačena hodnota 10^{-8} (čárkovaná vodorovná čára), na vodorovné ose hodnoty 10^{-8}, 10^{-5}, 10^{-3} v místech průsečíků klesající přímky s jednotlivými rostoucími přímkami (čárkované svislice).

5.4 Vzorce pro numerickou derivaci a integraci na rovnoměrné síti pomocí operátorů na polynomech

Definujme na prostoru polynomů do určitého stupně n následující operátory:

$I\,p(x) = p(x)$	identita,
$E\,p(x) = p(x+h)$	posunutí,
$\Delta\,p(x) = p(x+h) - p(x)$	dopředná diference,
$\nabla\,p(x) = p(x) - p(x-h)$	zpětná diference,
$\delta\,p(x) = p\big(x+\frac{h}{2}\big) - p\big(x-\frac{h}{2}\big)$	centrovaná diference.

Poznámka 5.4.1. Operátory E, Δ, ∇ a δ jsou závislé na h, ale tuto závislost obvykle vyznačovat nebudeme.

Jde o lineární operátory a standardně jsou k operátoru A definovány i operátory $A^2, \ldots, A^k$, dále $A^0 = I$ a A^{-1} jakožto operátor, pro nějž $A^{-1}(Ap(x)) = p(x)$ pro libovolný polynom. Pro některé operátory, jako jsou diference, však operace $A^{-1}p(x)$ není jednoznačná a výsledek je určen až na konstantu.

Exponenciála je dána Taylorovým rozvojem, například

$$e^{\Delta} = I + \Delta + \frac{\Delta^2}{2} + \ldots,$$

a obdobně jiné funkce. Pro polynomy stupně nejvýše n ovšem platí

$$\Delta^m p_n(x) = \nabla^m p_n(x) = \delta^m p_n(x) = 0 \quad \text{pro } m > n, \tag{5.14}$$

neboť se vždy sníží řád polynomu minimálně o 1, a tedy Taylorova řada je konečná.

Protože $\Delta = E - I$, platí

$$\Delta^n = (E-I)^n = \sum_{k=0}^{n} (-1)^k \binom{n}{k} E^{n-k},$$

neboli, označíme-li $x_i = x_0 + ih$, $i = 1, \ldots, n$, a $p_i = p(x_i)$,

$$\begin{aligned}\Delta p_i &= p_{i+1} - p_i,\\ \Delta^2 p_i &= p_{i+2} - 2p_{i+1} + p_i,\\ \Delta^3 p_i &= p_{i+3} - 3p_{i+2} + 3p_{i+1} - p_i \qquad \text{atd.}\end{aligned}$$

Obdobně pro $\nabla = I - E^{-1}$ máme

$$\nabla^n = (I - E^{-1})^n = \sum_{k=0}^{n} (-1)^k \binom{n}{k} E^{-k}$$

a

$$\begin{aligned}\nabla p_i &= p_i - p_{i-1},\\ \nabla^2 p_i &= p_i - 2p_{i-1} + p_i \qquad \text{atd.}\end{aligned}$$

5.4.1 Derivace

Zavedeme operátor derivace $D\,p(x) = p'(x)$. Pomocí Taylorova rozvoje máme

$$\begin{aligned}(I + \Delta)p(x) = E\,p(x) = p(x+h) &= p(x) + h\,p'(x) + \frac{h^2}{2!}p''(x) + \ldots\\ &= \left(I + hD + \frac{(hD)^2}{2!} + \ldots\right) p(x) = e^{hD}\,p(x),\end{aligned}$$

odkud

$$hD = \ln(I + \Delta) = \Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \cdots + (-1)^{k+1}\frac{\Delta^k}{k} + \ldots \tag{5.15}$$

Obdobně pro zpětnou diferenci je $e^{-hD} = I - \nabla$, a tedy

$$hD = \nabla + \frac{\nabla^2}{2} + \frac{\nabla^3}{3} + \ldots \tag{5.16}$$

Aplikujeme-li tyto vztahy na interpolační polynom funkce $f(x)$ jdoucí body $f_i = f(x_i)$, $x_i = x_0 + ih$, a omezíme-li se na prvních několik členů, dostaneme vzorce pro přibližnou derivaci. Například dopředná konečná diference dává

$$Df_i = f'(x_i) \cong \frac{\Delta f_i}{h} = \frac{f_{i+1} - f_i}{h} \qquad \text{s chybou } \mathcal{O}(h),$$

respektive

$$\begin{aligned}Df_i \cong \frac{1}{h}\left(\Delta - \frac{\Delta^2}{2}\right) f_i &= \frac{f_{i+1} - f_i}{h} - \frac{f_{i+2} - 2f_{i+1} + f_i}{2h}\\ &= \frac{-f_{i+2} + 4f_{i+1} - 3f_i}{2h} \qquad \text{s chybou } \mathcal{O}(h^2).\end{aligned}$$

Pro centrovanou diferenci vezmeme

$$\delta_{2h} = \Delta + \nabla = e^{hD} - e^{-hD} = 2\sinh(hD),$$

a tedy

$$hD = \sinh^{-1}\frac{\Delta + \nabla}{2} = \frac{\Delta + \nabla}{2} - \frac{1}{6}\left(\frac{\Delta + \nabla}{2}\right)^3 + \ldots \tag{5.17}$$

5.4.2 Integrace

Nyní definujeme operátor integrace

$$\begin{aligned} J\,p(x) &= \int_x^{x+h} p(t)\,\mathrm{d}t = h\int_0^1 E^s p(x)\,\mathrm{d}s = h\int_0^1 e^{s\ln E}\,\mathrm{d}s\; p(x) \\ &= h\,\frac{E-I}{\ln E}\,p(x) = \frac{h\Delta}{\ln(I+\Delta)}\,p(x), \end{aligned} \tag{5.18}$$

kde jsme použili substituci $t = x + sh$, $\mathrm{d}t = h\,\mathrm{d}s$, formálně integrovali přes s a využili vztahu $E^s p(x) = p(x + sh)$.

Můžeme tedy psát

$$J\,p(x) = \frac{h\Delta}{hD}\,p(x) = \frac{\Delta}{D}\,p(x) \qquad \text{a tedy} \qquad JD = DJ = \Delta.$$

Opět Taylorovým rozvojem dostaneme

$$J = \frac{h\Delta}{\ln(I+\Delta)} = h\left(I + \frac{1}{2}\Delta - \frac{1}{12}\Delta^2 + \frac{1}{24}\Delta^3 - \frac{19}{720}\Delta^4 + \dots\right). \tag{5.19}$$

Lze při tom využít trik $\ln(I+\Delta) = \Delta(I-R)$, kde $R = \frac{1}{2}\Delta - \frac{1}{3}\Delta^2 + \dots$, takže

$$J = \frac{h}{I-R} = h\left(1 + R + R^2 + \dots\right).$$

Opět ponecháním prvních členů máme aproximaci integrálu přes interpolační polynom dané funkce, například

$$\int_{x_0}^{x_1} f(x)\,\mathrm{d}x = Jf(x)\Big|_{x_0} \cong hf_0 + \frac{h}{2}(f_1 - f_0) = \frac{h(f_1+f_0)}{2} + \mathcal{O}(h^3). \tag{5.20}$$

Kapitola 6

Numerická integrace

6.1 Kvadraturní vzorce

Chceme-li spočítat určitý integrál, případně integrál s váhovou funkcí

$$I[f(x)] = \int_a^b f(x)\,\mathrm{d}x, \qquad \text{příp.} \qquad I[f(x)] = \int_a^b w(x) f(x)\,\mathrm{d}x, \tag{6.1}$$

numericky, aproximujeme jej určitým kvadraturním vzorcem

$$I_n[f] = \sum_{j=0(1)}^{n} w_j\, f(x_j), \tag{6.2}$$

kde w_j jsou tzv. váhy a x_j uzly kvadraturního vzorce.

Poznámka 6.1.1. Zápis $j = 0(1)$ znamená, že u některých typů kvadratur je vhodné číslovat od 0 (Newtonovy–Cotesovy vzorce) a jindy od 1 (Gaussovy vzorce).

Existují i zobecnění, která uvažují též derivace $f'(x_j)$ a příslušné váhy w'_j. Přímo se jimi zabývat nebudeme, ale využití je například pro určení chyby Gaussova vzorce.

Obvykle požadujeme, aby $I_n[f]$ dával přesné výsledky pro polynomy do určitého stupně s, který pak nazýváme *řádem kvadraturního vzorce*. Podle toho, které parametry ponecháme volné, dostáváme tři základní typy kvadratur.

A) Pokud zafixujeme x_j, obvykle $x_j = x_0 + jh$, $j = 0, \ldots, n$, pak máme $n+1$ volných parametrů w_j, které můžeme určit položením

$$I_n[x^k] = I[x^k] = \mu_k, \qquad k = 0, \ldots, n.$$

To vede na $Vw = \mu$, kde V je opět Vandermondeova matice, a pro $x_i \neq x_j$ pro $\forall i \neq j$ máme jednoznačné řešení. Takovéto kvadratury nazýváme *Newtonovy–Cotesovy vzorce*.

B) Pokud bereme též x_j jako volné parametry, pak budeme mít x_j a w_j, $j = 1, \ldots, n$, tj. $2n$ parametrů, které by mělo jít v principu určit z podmínek

$$I_n[x^k] = \mu_k, \qquad k = 0, \ldots, 2n-1.$$

Jde však o soustavu nelineárních rovnic, a proto se obvykle postupuje jinak, viz níže. Tyto kvadratury se nazývají *Gaussovy*.

C) Jsou též kvadratury, které fixují pouze některé uzly. Například *Lobattova kvadratura* fixuje krajní body a, b, a tedy integruje přesně polynomy stupně $2n-1$, máme-li $x_0 = a$, $x_1, \ldots, x_n = b$.

6.2 Newtonovy–Cotesovy vzorce

Newtonovy–Cotesovy vzorce jsou numerické kvadratury, které *integrují přesně polynomy do řádu* n, máme-li zadány uzly $x_j = x_0 + jh$, $j = 0, \dots, n$.

Máme-li funkci f zadánu v bodech x_j, pak Lagrangeův interpolační polynom $L_n(x)$ těmito body je stupně n, a tedy

$$I[f(x)] \simeq \int_a^b L_n(x)\,\mathrm{d}x = I_n[f(x)], \tag{6.3}$$

a tedy

$$\sum_{j=0}^{n} w_j f(x_j) = \sum_{j=0}^{n} f(x_j) \underbrace{\int_a^b \ell_j(x)\,\mathrm{d}x}_{w_j}, \qquad \text{kde } \ell_j(x) = \prod_{i\neq j} \frac{x - x_i}{x_j - x_i}. \tag{6.4}$$

Chybu této aproximace lze určit integrací chybového členu Lagrangeova interpolačního polynomu. Ukazuje se (viz např. Segethová, Ralston a podobně), že pro n sudé integrují Newtonovy–Cotesovy vzorce přesně i polynomy stupně $(n+1)$, a chyba je následující.

1) Pro n sudé, nechť $f \in C^{n+2}\langle a, b\rangle$; pak existuje $\xi \in \langle a, b\rangle$ takové, že

$$\int_a^b f(x)\,\mathrm{d}x - \int_a^b L_n(x)\,\mathrm{d}x = \frac{-f^{(n+2)}(\xi)}{(n+2)!} \int_a^b x\,\omega_{n+1}(x)\,\mathrm{d}x, \tag{6.5}$$

kde $\omega_{n+1}(x) = \prod_{j=0}^{n}(x - x_j)$.

2) Pro n liché, nechť $f \in C^{n+1}\langle a, b\rangle$; pak existuje $\xi \in \langle a, b\rangle$ takové, že

$$I[f(x)] - I[L_n(x)] = \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_a^b \omega_{n+1}(x)\,\mathrm{d}x. \tag{6.6}$$

6.2.1 Odvození vzorců pomocí operátorů na polynomech

Definujme operátor integrace

$$Jp(x) = \int_x^{x+h} p(t)\,\mathrm{d}t. \tag{6.7}$$

Provedeme substituci $t = x + sh$, $\mathrm{d}t = h\,\mathrm{d}s$, a využijeme, že $E^s p(x) = p(x + sh)$. Formální integrací přes s dostaneme

$$\begin{aligned} Jp(x) &= h \int_0^1 E^s p(x)\,\mathrm{d}s = h \int_0^1 e^{s \ln E}\,\mathrm{d}s\; p(x) = h\,\frac{E - I}{\ln E}\,p(x) = \frac{h\Delta}{\ln(I + \Delta)}\,p(x) \\ &= \frac{h\Delta}{hD}\,p(x) = \frac{\Delta}{D}\,p(x). \end{aligned}$$

Dostáváme mimo jiné vztah $JD = DJ = \Delta$.

Vzorce pro numerickou kvadraturu dostaneme Taylorovým rozvojem v Δ do určitého stupně:

$$J = \frac{h\Delta}{\ln(I + \Delta)} = h\left(I + \frac{1}{2}\Delta - \frac{1}{12}\Delta^2 + \frac{1}{24}\Delta^3 - \frac{19}{720}\Delta^4 + \dots\right), \tag{6.8}$$

například trikem $\ln(I + \Delta) = \Delta(I - R)$, kde

$$R = \frac{1}{2}\Delta - \frac{1}{3}\Delta^2 + \frac{1}{4}\Delta^3 - \dots \qquad \text{a} \qquad J = \frac{h}{I - R} = h\left(I + R + R^2 + \dots\right).$$

6.2.2 Lichoběžníkové a Simpsonovo pravidlo

Z rozvoje (6.8) dostáváme kvadraturní vzorce. Například ponecháním členů do Δ dostaneme

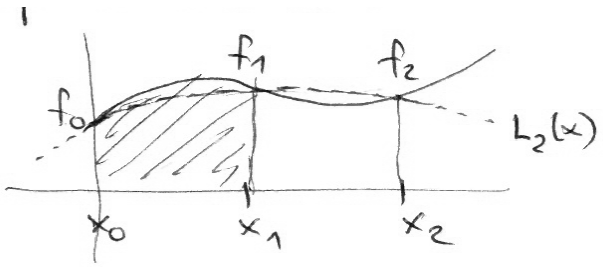
$$\int_{x_0}^{x_1} f(x)\,\mathrm{d}x = Jf(x)\Big|_{x_0} \simeq hf_0 + \frac{h}{2}(f_1 - f_0) = \frac{h(f_1+f_0)}{2} - \frac{f''(\xi)h^3}{12}, \tag{6.9}$$

což je *lichoběžníkové pravidlo*.

Přidáním dalšího členu máme

$$\int_{x_0}^{x_1} f(x)\,\mathrm{d}x \simeq h\left(\frac{5}{12}f_0 + \frac{2}{3}f_1 - \frac{1}{12}f_2\right). \tag{6.10}$$

Poznámka 6.2.1. Tento vzorec není úplně ideální: parabola $L_2(x)$ je proložena třemi body, ale integruje se jen přes první podinterval.



Obrázek 6.1: Náčrt paraboly $L_2(x)$ proložené body f_0, f_1, f_2 v uzlech x_0, x_1, x_2 nad vodorovnou osou; šrafovaná je plocha pod křivkou mezi x_0 a x_1.

Sečteme-li (6.10) se vzorcem pro sousední podinterval

$$\int_{x_1}^{x_2} f(x)\,\mathrm{d}x \simeq h\left(-\frac{1}{12}f_0 + \frac{2}{3}f_1 + \frac{5}{12}f_2\right), \tag{6.11}$$

dostaneme

$$\int_{x_0}^{x_2} f(x)\,\mathrm{d}x \simeq \frac{h}{3}(f_0 + 4f_1 + f_2) - \frac{f^{(4)}(\xi)}{90}h^5, \tag{6.12}$$

tedy *Simpsonovo pravidlo*.

Obdobně bychom mohli získat vyšší kvadraturní vzorce, například *Booleovo pravidlo*

$$\int_{x_0}^{x_4} f(x)\,\mathrm{d}x = \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) + \mathcal{O}\left(h^7\right). \tag{6.13}$$

Obecně se ovšem nevyplatí zvyšovat n kvůli numerickým nestabilitám: například pro $n > 7$ jsou některé váhy záporné, což vede k odečítání blízkých čísel a podobně. Navíc tyto vzorce konvergují pro $n \to \infty$ jen pro omezenou třídu funkcí.

6.2.3 Složené kvadraturní vzorce

V praxi se užívají tzv. *složené kvadraturní vzorce* (též *rozšířené*), kdy použijeme určité kvadraturní pravidlo na podintervalech, do kterých rozdělíme interval $\langle a, b\rangle$.

Například *složené lichoběžníkové pravidlo* je

$$\int_a^b f(x)\,\mathrm{d}x \simeq h\left(\frac{f_0}{2} + \sum_{i=1}^{n-1} f_i + \frac{f_n}{2}\right) + \mathcal{O}\left(h^2\right) \tag{6.14}$$

a složené Simpsonovo pravidlo (pro n sudé)

$$\int_a^b f(x)\,\mathrm{d}x \simeq \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 4f_{n-1} + f_n) + \mathcal{O}\left(h^4\right). \tag{6.15}$$

Poznámka 6.2.2. Řád chyby je zde o řád nižší než pro samotný integrál přes $\langle x_0, x_1\rangle$, a to kvůli sčítání chyb z jednotlivých podintervalů.

Existují i tzv. otevřené Newtonovy–Cotesovy vzorce, kdy $x_0 > a$, $x_n < b$ a vyčíslujeme integrál na $\langle a, b\rangle$. To je třeba udělat, pokud $f(a)$ či $f(b)$ divergují, ale přesto je integrál konečný. Vzhledem k malé přesnosti oproti Gaussovým kvadraturám, které jsou též v tomto smyslu „otevřené“, se však často nepoužívají.

6.3 Eulerův–Maclaurinův sumační vzorec

Pro chybu složeného lichoběžníkového pravidla si dokážeme následující vztah, který je důležitý pro extrapolaci pro $h \to 0$, a tedy pro Rombergův algoritmus.

Věta 6.3.1 (Eulerův–Maclaurinův sumační vzorec). *Nechť $f(x) \in C^{2k+2}\langle x_0, x_n\rangle$. Potom platí*

$$\begin{aligned} h\left(\frac{f_0}{2} + \sum_{i=1}^{n-1} f_i + \frac{f_n}{2}\right) = \int_{x_0}^{x_0+nh} f(x)\,\mathrm{d}x + a_1\left[f'(x_n) - f'(x_0)\right]h^2 \\ + a_2\left[f'''(x_n) - f'''(x_0)\right]h^4 + \dots \\ + a_k\left[f^{(2k-1)}(x_n) - f^{(2k-1)}(x_0)\right]h^{2k} + \dots, \end{aligned} \tag{6.16}$$

kde konstanty $a_1, a_2, \dots$ jsou koeficienty Taylorova rozvoje funkce $\dfrac{x}{2\tanh\frac{x}{2}}$.

Konstanty a_k lze vyjádřit též jako $a_k = \dfrac{B_{2k}}{(2k)!}$, kde B_n jsou Bernoulliova čísla

$$B_0 = 1,\ B_1 = \frac{1}{2},\ B_2 = \frac{1}{6},\ B_3 = 0,\ B_4 = -\frac{1}{30},\ B_6 = \frac{1}{42}.$$

Důkaz. Upravujme rozdíl

$$\begin{aligned} &h\left[\sum_{i=0}^{n-1} f_i + \frac{1}{2}(f_n - f_0)\right] - \int_{x_0}^{x_0+h} f(x)\,\mathrm{d}x - \int_{x_0+h}^{x_0+2h} f(x)\,\mathrm{d}x - \dots - \int_{x_0+(n-1)h}^{x_0+nh} f(x)\,\mathrm{d}x = \\ &= h\left[\underbrace{\left(E^0 + E^1 + \dots + E^{n-1}\right)}_{(E^n-I)/(E-I)} f_0 + \frac{1}{2}(E^n - I) f_0\right] \\ &\quad - \left[Jf(x_0) + E^1 Jf(x_0) + \dots + E^{n-1} Jf(x_0)\right] = \\ &= h(E^n - I)\left(\frac{1}{\Delta} + \frac{I}{2}\right) f_0 - (E^n - I)\left(\frac{J}{\Delta}\right) f(x_0) = \\ &= h(E^n - I)\left(\frac{1}{2\tanh\frac{hD}{2}} - \frac{1}{hD}\right) f_0 = \\ &= \frac{1}{12}h^2\left(f'(x_n) - f'(x_0)\right) - \frac{1}{720}h^4\left(f'''(x_n) - f'''(x_0)\right) + \dots \end{aligned}$$

Přitom jsme využili, že $\dfrac{J}{\Delta} = \dfrac{1}{D}$, a dále vedlejšího výpočtu (s $E = e^{hD}$)

$$\begin{aligned} \frac{I}{2} + \frac{1}{\Delta} = \frac{I}{2} + \frac{1}{e^{hD} - I} = \frac{e^{hD} + I}{2\left(e^{hD} - I\right)} = \frac{1}{2\tanh\frac{hD}{2}} \\ = \frac{1}{hD} + \frac{1}{12}hD - \frac{1}{720}(hD)^3 + \frac{1}{30240}(hD)^5 - \dots \end{aligned}$$

□

Poznámka 6.3.2. Operace $\frac{1}{D}$ a $\frac{1}{\Delta}$ nejsou jednoznačně definovány, jde o třídu funkcí lišících se konstantou: $\frac{1}{D}p(x)$ jsou polynomy $P(x)$ takové, že $DP(x) = p(x)$, a totéž tedy platí pro $P(x)+c$. Vždy jsou ale v kombinaci s $(E^n - E^0)$, takže výsledek už je jednoznačný.

6.4 Využití Eulerova–Maclaurinova vzorce

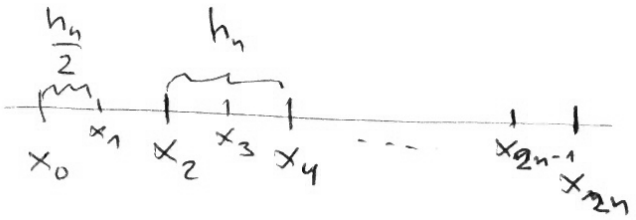
6.4.1 Odvození složených Newtonových–Cotesových vzorců vyššího řádu

Složené Newtonovy–Cotesovy vzorce vyššího řádu lze odvodit pomocí Richardsonovy extrapolace. Označme $I_n^0[f]$ lichoběžníkové pravidlo pro $n+1$ bodů. Platí tedy

$$\begin{aligned} I_n^0[f] &= I[f] + a_1 \left[f_n' - f_0'\right] h_n^2 + \mathcal{O}(h^4) \\ &= I[f] + c\, h_n^2 + \mathcal{O}(h_n^4) \end{aligned}$$

a zároveň máme

$$I_{2n}^0[f] = I[f] + c \left(\frac{h_n}{2}\right)^2 + \mathcal{O}(h_n^4).$$



Obrázek 6.2: Číselná osa s vyznačeným dělením na body $x_0, x_1, x_2, x_3, x_4, \ldots, x_{2n-1}, x_{2n}$; svorka nad úsekem x_0–x_1 je popsána $\frac{h_n}{2}$, svorka nad úsekem x_2–x_4 je popsána h_n.

Vhodnou kombinací obou vztahů dostaneme

$$I_{2n}^1[f] = \frac{4\, I_{2n}^n - I_n^0}{3} = I[f] + \mathcal{O}(h_n^4), \tag{6.17}$$

a tedy

$$\begin{aligned} I_{2n}^1[f] &= \frac{4}{3}\left[\frac{h_n}{2}\left(\frac{f_0}{2} + \sum_{i=1}^{2n-1} f_i + \frac{f_{2n}}{2}\right)\right] - \frac{1}{3} h_n \left(\frac{f_0}{2} + \sum_{k=1}^{n-1} f_{2k} + \frac{f_{2n}}{2}\right) = \\ &= \frac{\left(\frac{h_n}{2}\right)}{3}\left[f_0 + 4f_1 + 2f_2 + \cdots + 4f_{2n-1} + f_{2n}\right], \end{aligned}$$

což je Simpsonovo pravidlo.

Obecně platí

$$\left.\begin{aligned} I_n^{k-1} &= I + c\, h^{2k} + \mathcal{O}(h^{2k+2}) \\ I_{2n}^{k-1} &= I + c \left(\frac{h}{2}\right)^{2k} + \mathcal{O}(h^{2k+2}) \end{aligned}\right\} \quad \Rightarrow \quad I_{2n}^k = \frac{4^k I_{2n}^{k-1} - I_n^{k-1}}{4^k - 1} = I + \mathcal{O}(h^{2k+2}).$$

Pro Booleovo pravidlo tak dostáváme

$$I_{2n}^2 = \frac{16\,I_{2n}^1 - I_n^1}{15} = \frac{16}{15}\frac{\left(\frac{h}{2}\right)}{3}\left(f_0 + 4f_1 + 2f_2 + \cdots + 2f_{2n-2} + 4f_{2n-1} + f_{2n}\right)$$
$$- \frac{1}{15}\frac{h}{3}\left(f_0 + \quad 4f_2 + 2f_4 + \cdots + 4f_{2n-2} \quad + \quad f_{2n}\right) =$$
$$= \left(\frac{h}{2}\right)\frac{2}{45}\left(7f_0 + 32f_1 + 12f_2 + \underbrace{\cdots}_{+32f_3+14f_4+32f_5+\ldots} + 32f_{2n-1} + 7f_{2n}\right).$$

6.4.2 Rombergův algoritmus

Richardsonovu extrapolaci lze použít přímo k výpočtu integrálu. Začneme s $h_1 = b-a$ a počítáme

$$I_1^0[f] = (b-a)\frac{f_1 + f_0}{2},$$

a dále

$$I_{2n}^0[f] = \frac{1}{2}I_n^0[f] + h_{2n}\sum_{k=1}^{n} f(x_{2k-1}), \tag{6.18}$$

kde faktor $\frac{1}{2}$ plyne z toho, že $h_{2n} = \frac{h_n}{2}$. Hodnoty I_{2n}^k pro $k > 0$ pak dopočítáme pomocí Richardsonovy extrapolace.

Dostáváme tabulku

počet bodů	$h \diagdown$ chyba	$\mathcal{O}(h^2)$		$\mathcal{O}(h^4)$		$\mathcal{O}(h^6)$	$\cdots$	$\mathcal{O}(h^{2k+2})$
2	$b-a$	I_1^0						
$2^1+1=3$	$\frac{b-a}{2}$	I_2^0	$\to$	I_2^1				
$2^2+1=5$	$\frac{b-a}{2^2}$	I_4^0	$\to$	I_4^1	$\to$	I_4^2		$\cdots$
	$\vdots$	$\vdots$						
2^k+1	$\frac{b-a}{2^k}$	$I_{2^k}^0$	$\to$	$I_{2^k}^1$	$\to$	$I_{2^k}^2$	$\to \cdot \to$	$I_{2^k}^k$

V tabulce navíc vedou šikmé šipky $\searrow$ vždy z posledního prvku jednoho řádku do dalšího sloupce na řádku pod ním, tj. $I_1^0 \searrow I_2^1$, $I_2^1 \searrow I_4^2$ a tak dále. Poslední šipka vedoucí do $I_{2^k}^k$ odpovídá odstranění chyby $\mathcal{O}(h^{2k})$.

Poznámka 6.4.1.

a) Má-li $f(x)$ charakteristickou škálu, na které se mění, je vhodné začínat rovnou s I_n^0, kde n je zvoleno tak, aby $h_n = \frac{b-a}{n}$ bylo srovnatelné s touto škálou.

b) Schéma funguje až do k-tého sloupce, má-li funkce $f(x)$ $2k-1$ spojitých derivací; jinak je nutné zvyšovat počet bodů.

e) Pokud jsou derivace na kraji nulové či stejné, pak Richardsonova extrapolace selhává, protože se chyba chová jinak; je tedy nutné to hlídat. Například u hladké periodické funkce je integrál přes periodu vhodné aproximovat přímo pomocí I_n^0 a zvětšovat n, neboť chyba klesá rychleji než jakákoli mocnina h^m.

d) Rombergův algoritmus může být vhodnější než Gaussova kvadratura (viz níže), pokud chceme dosáhnout určité přesnosti a dopředu nevíme, kolik bodů n za tím účelem u Gaussovy kvadratury zvolit, neboť přepočítávání Gaussových x_n, w_n je časově náročné.

6.5 Gaussova kvadratura

Uvažujme nyní integrál s vahou

$$I[f(x)] = \int_a^b f(x)\, v(x)\, \mathrm{d}x, \tag{6.19}$$

kde váha $v(x)$ je skoro všude kladná na $\langle a, b\rangle$, a předpokládejme, že $I[x^k]$ je konečný pro $\forall k$.

Tento integrál chceme aproximovat kvadraturním vzorcem

$$I[f(x)] \simeq \sum_{j=1}^{n} w_j\, f(x_j) \equiv I_n[f(x)] \tag{6.20}$$

tak, aby $I_n[x^k] = I[x^k] = \mu_k$ pro $k = 0, \ldots, 2n-1$, tj. aby integrál z polynomu do stupně $2n-1$ byl přesný. Máme $2n$ podmínek $I_n[x^k] = \mu_k$ pro $2n$ neznámých $x_1, \ldots, x_n$; $w_1, \ldots, w_n$, avšak tyto podmínky jsou nelineární.

Ukážeme si však obecně, že pro zvolenou váhu $v(x)$ musí být x_j *kořeny ortogonálního polynomu* $P_n(x)$ *pro tuto váhu* $v(x)$, a pak už stačí jen určit váhy w_j, což je lineární problém. Podle váhy pak například dostáváme Gaussovu–Legendreovu kvadraturu pro $v(x) = 1$ a podobně pro další váhy.

6.5.1 Odvození

Předpokládejme, že už známe x_j, a zvolme w_j tak, aby šlo o Newtonův–Cotesův vzorec, tj. aby $I_n[x^k] = \mu_k$ pro $k = 0, \ldots, n-1$.

Nyní každý polynom $p(x)$ stupně nejvýše $2n-1$ lze psát ve tvaru

$$p(x) = q(x)\, P_n(x) + r(x), \tag{6.21}$$

kde $P_n(x)$ je ortogonální polynom dané váhy a $q(x)$ a $r(x)$ jsou nejvýše stupně $n-1$.

Pro integrál $p(x)$ dostaneme

$$I[p(x)] = \underbrace{I[q(x)\, P_n(x)]}_{=0} + I[r(x)] = I_n[r(x)],$$

neboť $P_n(x)$ je ortogonální na $\forall$ polynomy stupně $< n$ a I_n integruje přesně polynomy stupně $n-1$. Ale zároveň má být přesné

$$I[p(x)] = \underbrace{\sum_{k=1}^{n} w_k\, q(x_k)\, P_n(x_k)}_{\text{musí být } 0} + I_n[r(x)].$$

Označená suma tedy musí být nula pro $\forall p(x)$, a tedy pro libovolné $q(x)$, což může být jedině tehdy, pokud $P_n(x_k) = 0$ pro $\forall k$. Odtud plyne, že x_k *musí být kořeny* $P_n(x)$.

Poznámka 6.5.1. Váhy lze vyjádřit jako

$$w_k = \int_a^b \ell_k(x)\, v(x)\, \mathrm{d}x$$

podobně jako u Newtonových–Cotesových vzorců.

Lze ukázat, že váhy w_k jsou vždy kladné pro $v(x)$ kladné skoro všude a že integrál konverguje pro dostatečně hladké funkce. Neboť pro $p(x) = \ell_i(x)^2$ stupně $2n-2$, kde

$$\ell_i(x) = \frac{\prod\limits_{i \neq j} (x - x_j)}{\prod\limits_{i \neq j} (x_i - x_j)},$$

máme

$$0 < I[\ell_i(x)^2] = \sum_{k=1}^{n} \ell_i(x_k)^2\, w_k = \sum_{k=1}^{n} \delta_{ik}\, w_k = w_i.$$

6.5.2 Výpočet kořenů x_j a vah w_j

Nejčastěji se postupuje pomocí diagonalizace tridiagonální matice

$$J_n = \begin{pmatrix} \delta_1 & \gamma_2 & & & \\ \gamma_2 & \delta_2 & \ddots & & 0 \\ & \ddots & \ddots & & \\ & & & & \gamma_n \\ 0 & & & \gamma_n & \delta_n \end{pmatrix}, \tag{6.22}$$

neboť $\det(\lambda - J_n)$ splňuje stejné rekurentní vztahy jako $p_n(x)$:

$$\det(\lambda - J_n) = \det(\lambda - J_{n-1})(\lambda - \delta_n) - \gamma_n^2 \det(\lambda - J_{n-2}).$$

Kořeny x_j jsou tedy též vlastní čísla J_n.

Lze ukázat, že

$$w_j^{(n)} = \left(v_1^{(j)}\right)^2, \qquad j = 1, \ldots, n,$$

kde $v_1^{(j)}$ je první prvek j-tého vlastního vektoru odpovídajícího vlastní hodnotě $x_j^{(n)}$, pokud je normovaný tak, že

$$v^{(j)T} v^{(j)} = (p_0, p_0) = \int_a^b v(x)\,\mathrm{d}x = \mu_0.$$

6.5.3 Chyba Gaussovy kvadratury

Pokud $f \in C^{2n}\langle a, b\rangle$, pak

$$I[f(x)] - I_n[f(x)] = \frac{f^{(2n)}(\xi)}{(2n)!}\,(p_n, p_n) \tag{6.23}$$

pro určité $\xi \in \langle a, b)$.

Existují i kvadratury, které fixují krajní body ($a = x_1$, $b = x_n$); to jsou Gaussovy–Lobattovy kvadratury.

6.5.4 Základní Gaussovy kvadratury

	$\langle a, b\rangle$	$w(x)$
Gaussova–Legendreova kv.	$\langle -1, 1\rangle$	1
Gaussova–Čebyševova kv.	$\langle -1, 1\rangle$	$1/\sqrt{1-x^2}$ nebo $\sqrt{1-x^2}$
Gaussova–Jacobiho kv.	$\langle -1, 1\rangle$	$(1-x)^\alpha(1+x)^\beta$
Gaussova–Laguerreova kv.	$\langle 0, \infty\rangle$	$x^\alpha e^{-x}$
Gaussova–Hermiteova kv.	$\langle -\infty, \infty\rangle$	e^{-x^2}

6.6 Integrály se singularitami

Postupovat lze buď (pokud víme, kde a jaká singularita je) rozdělením na dva intervaly a použitím vhodné Gaussovy kvadratury se singulární vahou, nebo (je-li například singularita v proměnném bodě) lze občas singularitu odečíst.

Chceme-li například spočítat

$$Hf(y) = \text{v.p.} \int_{-\infty}^{\infty} \frac{f(x)}{y-x}\,\mathrm{d}x$$

(tzv. Hilbertova transformace) a známe-li $Hg(y)$ pro vhodnou funkci $g(x)$, pak pro $\tilde{f}(x) = f(x)/g(x)$ platí

$$Hf(y) = \text{v.p.} \int_{-\infty}^{\infty} \frac{\tilde{f}(x)\,g(x)}{y-x}\,\mathrm{d}x = \int_{-\infty}^{\infty} \frac{\tilde{f}(x)-\tilde{f}(y)}{y-x}\,g(x)\,\mathrm{d}x + \tilde{f}(y)\,Hg(y), \tag{6.24}$$

přičemž podíl $\dfrac{\tilde{f}(x)-\tilde{f}(y)}{y-x}$ už může být hladká funkce.

6.7 Poznámky k vícerozměrným integrálům

Vícerozměrné integrály jsou mnohem obtížnější než 1D integrály:

a) hranice může být velmi komplikovaná (nekonvexní apod.);

b) počet vyčíslení funkce může velice rychle narůstat, chceme-li vysokou přesnost (pokud v 1D potřebujeme 10 bodů, tak ve 3D 10^3 a obecně 10^N).

Proto je důležité se nejprve zamyslet:

a) Je možné část integrálu spočítat analyticky? Pokud ano, určitě to udělat.

b) Je potřeba vysoká přesnost? Pokud ne, pak je asi nejvhodnější metoda Monte Carlo (viz letní semestr), která je vhodná i pro komplikovanou hranici. Bohužel chyba je řádu $\dfrac{1}{\sqrt{n}}$, kde n je počet bodů, ovšem nezávisle na dimenzi.

Pokud je třeba vysoká přesnost, pak buď

a) integruj opakovaně v 1D, nebo

b) použij vícerozměrné Gaussovy kvadratury, je-li to možné.

Příklad 6.7.1 (Integrál přes obsah kruhu)**.** Integrál přes obsah kruhu lze zapsat jako

$$I = \int_{-1}^{1} \mathrm{d}x \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \mathrm{d}y\, f(x,y), \tag{6.25}$$

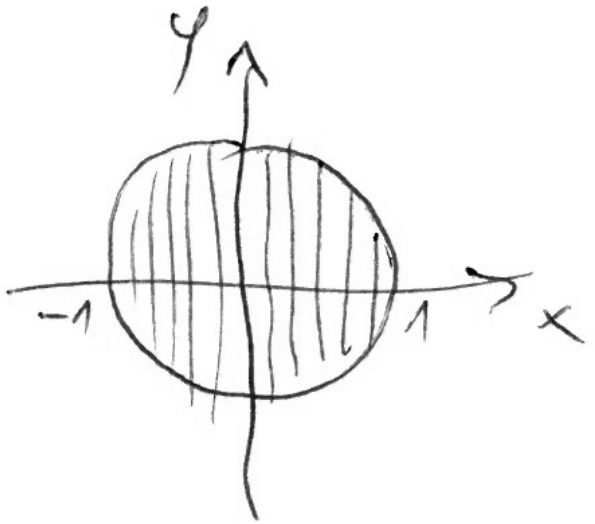
tj. nejprve integrace přes y ve „vhodných“ bodech x s ohledem na vnější integrál.

6.8 Poznámky k singulárním integrálům

Užitečné bývá použití vhodné substituce, například

$$\int_a^b f(x)\,\mathrm{d}x = \int_{1/b}^{1/a} \frac{1}{t^2}\, f\!\left(\frac{1}{t}\right)\mathrm{d}t, \qquad \text{pro } ab > 0, \tag{6.26}$$

a podobně; viz Numerical Recipes.



Obrázek 6.3: Souřadné osy x a y; kolem počátku je nakreslen kruh protínající osu x v bodech -1 a 1, jeho vnitřek je vyšrafován svislými čárami (naznačení integrace přes y při pevném x).

Kapitola 7

Numerické řešení ODR – Základní pojmy. Rungeovy-Kuttovy metody

7.1 Počáteční a okrajová úloha

Z hlediska numerického je zásadní rozdíl mezi řešením obyčejných diferenciálních rovnic (ODR) s počátečními a s okrajovými podmínkami. Rozlišujeme proto dva typy úloh:

1) *počáteční úloha* (initial value problem) — máme zadánu funkční hodnotu a případně derivace v jednom bodě (typicky v určitém čase) a iteračně hledáme řešení v následných bodech (časech);

2) *okrajová úloha* (boundary value problem) — funkce je zadána v různých bodech a hledáme řešení mezi těmito body. Okrajovou úlohu řešíme obvykle převedením na soustavu lineárních rovnic diskretizací nebo rozvojem do báze.

Zde se budeme zabývat počáteční úlohou; okrajové úlohy budou probírány v letním semestru.

7.1.1 Matematická formulace počáteční úlohy pro ODR

Nalezněte $u(t) : \langle 0, T\rangle \to \mathbb{C}^d$ splňující rovnici

$$u'(t) \equiv \frac{du}{dt} = f(u(t), t) \qquad \text{s poč. podmínkou } u(t=0) = u_0, \tag{7.1}$$

kde f je zadaná funkce (obecně vektorová)

$$f : \mathbb{C}^d \times \langle 0, T\rangle \to \mathbb{C}^d,$$

která může být (a obvykle je) nelineární.

Poznámka 7.1.1. Obecně tedy uvažujeme systém ODR, neboť $u(t) \in \mathbb{C}^d$.

Poznámka 7.1.2. Díky tomu jsou zahrnuty i počáteční úlohy pro ODR vyšších řádů, neboť stačí zavést proměnné $u_1(t) = u(t)$, $u_2(t) = u'(t)$ atd.

Poznámka 7.1.3. Bez újmy na obecnosti pokládáme počáteční čas $t_0 = 0$; vždy lze posunout.

Poznámka 7.1.4. Pro $f(u, t) = f(t)$ jde o integraci

$$u(t) = u_0 + \int_0^t f(t')\, dt', \tag{7.2}$$

proto se můžete setkat i s pojmem numerická integrace ODR.

7.2 Existence a jednoznačnost řešení

Otázka existence a jednoznačnosti řešení patří do matematické analýzy, viz příslušnou přednášku. Standardní Cauchyho předpoklady pro existenci a jednoznačnost řešení jsou:

a) f je spojitá;

b) f je stejnoměrně Lipschitzovská vzhledem k u, tj. existuje konstanta $L > 0$ taková, že

$$\|f(u,t) - f(v,t)\| \leq L \|u - v\| \quad \text{pro } \forall\, u, v \in \mathbb{C}^d \text{ a } t \in \langle 0, T), \tag{7.3}$$

přičemž $\|\cdot\|$ je vhodná vektorová norma.

Poznámka 7.2.1. Pokud existuje $\frac{\partial f}{\partial u}$, pak musí být všude omezená konstantou L.

Příklad 7.2.2. Uvažujme rovnici

$$u'(t) = u^2 = f(u), \qquad u(0) = 1.$$

Zde f není Lipschitzovská, neboť pro

$$|u^2 - v^2| = |u + v|\,|u - v|$$

má platit $|u^2 - v^2| \leq L\,|u - v|$, a tedy by L muselo být nekonečné. Řešením je

$$u(t) = \frac{1}{1 - t},$$

které v konečném čase $t = 1$ diverguje, a pro $t > 1$ tedy neexistuje.

Příklad 7.2.3. Uvažujme rovnici

$$u'(t) = \sqrt{u} = f(u), \qquad u(0) = 0.$$

I zde f není Lipschitzovská, neboť

$$\frac{\partial f}{\partial u} = \frac{1}{2\sqrt{u}} \to \infty \quad \text{pro } u \to 0.$$

Řešení ovšem existuje, ale není jednoznačné, neboť úlohu splňuje jak $u(t) = 0$, tak $u(t) = \frac{t^2}{4}$.

Lipschitzova konstanta L říká, jak moc se řešení pro různé počáteční podmínky rozbíhají, nebo naopak sbíhají. Oba extrémy (velká rozbíhavost, nebo naopak rychlá sbíhavost) způsobují numerické problémy.

7.2.1 Problém špatné podmíněnosti

Existují rovnice, kde libovolně blízké počáteční podmínky mohou vést na dramaticky rozdílná řešení; mnohdy se jedná o chaotické systémy.

Příklad 7.2.4 (Fyzické kyvadlo)**.** Fyzické kyvadlo popsané rovnicí

$$\ddot{\varphi}(t) = -\frac{g}{\ell} \sin \varphi(t) \tag{7.4}$$

má pro počáteční podmínky

$$\varphi(0) = \pi \pm \varepsilon \quad \text{a} \quad \dot{\varphi}(0) = 0$$

zcela jiné, i když jednoznačné řešení. Lipschitzova konstanta je

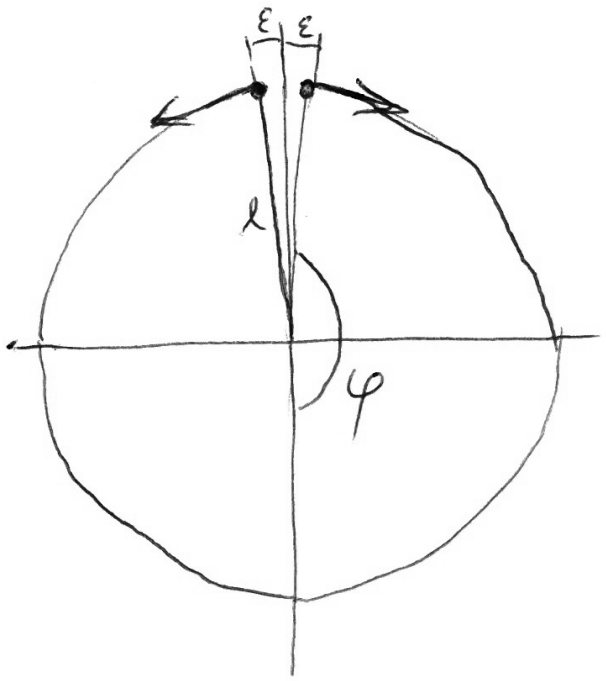
$$L = \max\left(1, \frac{g}{\ell}\right),$$

neboť pro

$$u = \begin{pmatrix} \varphi \\ \dot\varphi \end{pmatrix} \quad \text{splňující} \quad \frac{du}{dt} = \frac{d}{dt}\begin{pmatrix} \varphi \\ \dot\varphi \end{pmatrix} = \begin{pmatrix} \dot\varphi \\ -\frac{g}{\ell}\sin\varphi \end{pmatrix} = f(u)$$

dostaneme (označme si $v = \begin{pmatrix} \psi \\ \dot\psi \end{pmatrix}$)

$$\begin{aligned} \|f(u) - f(v)\|_\infty &= \max\left(|\dot\varphi - \dot\psi|,\ \frac{g}{\ell}\,|\sin\varphi - \sin\psi|\right) \le \\ &\le \max\left(|\dot\varphi - \dot\psi|,\ \frac{g}{\ell}\,|\varphi - \psi|\right) \le \\ &\le \max\left(1, \frac{g}{\ell}\right) \max\left(|\dot\varphi - \dot\psi|,\ |\varphi - \psi|\right) = \\ &= \max\left(1, \frac{g}{\ell}\right) \|u - v\|_\infty. \end{aligned}$$



Obrázek 7.1: Kružnice opisovaná kyvadlem se svislou a vodorovnou osou procházejícími středem otáčení. Ve svislém směru nahoru jsou vyznačeny dvě blízké polohy kyvadla odchýlené o ε na obě strany, z nichž vycházejí šipky opačným směrem (doleva a doprava) podél kružnice. Rameno kyvadla je označeno ℓ, úhel φ je vyznačen obloukem u středu.

7.3 Numerické řešení ODR — základní pojmy

Typicky hledáme řešení $u(t)$ ODR

$$u'(t) = f(u(t), t), \qquad u(t_0) = u_0$$

pouze v diskrétních časech $t_n = t_0 + nk_n$, tj. hledáme *přibližné řešení*

$$v^n \simeq u(t_n), \tag{7.5}$$

přičemž *časový krok* k_n je často konstantní (k), i když existují např. adaptivní metody, které mění krok pro dosažení vyšší přesnosti.

7.3.1 Jednokrokové a vícekrokové metody

Metody se obvykle dělí na jednokrokové a vícekrokové.

U *jednokrokových metod* závisí řešení v^{n+1} v každém kroku pouze na v^n. Výhodou je, že lze snadno změnit krok, a že jsou vhodné pro problémy s nespojitými funkcemi. Nevýhodou je, že pro vyšší přesnost je nutné počítat $f(u,t)$ i v mezibodech mezi t_n a t_{n+1} (víceúrovňové metody). Příkladem jsou *Rungeovy–Kuttovy metody*.

U *vícekrokových metod* se v^{n+1} počítá z předchozích s kroků (s-kroková metoda), tj. z hodnot $v^n, v^{n-1}, \ldots, v^{n-s+1}$. Výhodou je, že k výpočtu v^{n+1} typicky stačí jedno vyčíslení pravé strany $f(u,t)$, neboť ostatní hodnoty jsou známy z předchozích kroků; tyto metody tedy mohou být rychlejší při vyšší přesnosti. Nevýhodou je náročnější inicializace (potřebujeme na začátku s počátečních hodnot — lze např. použít Rungeovu–Kuttovu metodu) a obtížná změna kroku. Předpokladem je spojitost vyšších derivací. Příkladem jsou *lineární vícekrokové metody*.

7.3.2 Explicitní a implicitní metody

Dále dělíme metody na explicitní a implicitní. U *explicitních* metod nepotřebujeme k nalezení v^{n+1} hodnotu $f(v^{n+1}, t_{n+1})$. U *implicitních* metod závisí v^{n+1} na $f(v^{n+1}, t_{n+1})$; je tedy nutné řešit (typicky nelineární) rovnici pro v^{n+1}, k čemuž slouží iterační metody.

Někdy se používají tzv. *metody prediktor–korektor*, kdy do implicitního schématu dosazujeme za v^{n+1} v argumentu $f(v^{n+1}, t_{n+1})$ odhad určený z explicitní metody.

Výhodou implicitních metod je *vyšší stabilita* (viz později).

7.4 Lokální a globální diskretizační chyba

V každém kroku délky k se dopouštíme *lokální chyby*, kterou můžeme určit z Taylorova rozvoje. Její první člen bude řádu $O(k^{p+1})$, kde p nazýváme *řád přesnosti metody*.

Příklad 7.4.1 (Explicitní Eulerova metoda)**.** Nejjednodušší metodou je tzv. *explicitní Eulerova metoda*, která aproximuje

$$\frac{du(t_n)}{dt} \approx \frac{v^{n+1} - v^n}{k} = f(v^n, t_n), \tag{7.6}$$

a tedy

$$v^{n+1} = v^n + k\, f(v^n, t_n). \tag{7.7}$$

Lokální diskretizační chybu určíme tak, že za v^n položíme $u(t)$ a za v^{n+1} hodnotu $u(t+k)$ a určíme z Taylorova rozvoje chybu aproximace $u(t+k)$ pomocí daného rekurentního vztahu, tj.

$$u(t+k) - u(t) - k\,\underbrace{f(u(t),t)}_{u'(t)} = u(t) + ku'(t) + \frac{k^2}{2}u''(t) + \ldots - u(t) - ku'(t) =$$

$$= \frac{k^2}{2}u''(t) + O(k^3) = O(k^{p+1}),$$

neboli $p = 1$ pro explicitní Eulerovu metodu.

Pokud je lokální chyba řádu $O(k^{p+1})$, lze za poměrně obecných předpokladů jak pro jednokrokové, tak pro vícekrokové metody ukázat, že *globální chyba bude řádu* $O(k^p)$, pokud je daná metoda stabilní (tj. nedochází k růstu chyb v následných iteracích).

Takový výsledek lze očekávat, neboť pro určení v^n potřebujeme $\frac{t_n}{k}$ kroků s lokální chybou $O(k^{p+1})$, a tedy celková chyba by měla být

$$O\left(\frac{t_n}{k}\, k^{p+1}\right) \approx O(k^p).$$

V praxi se samozřejmě projevuje též *zaokrouhlovací chyba*, a tedy podobně jako u numerické derivace existuje i zde jistá optimální volba kroku k minimalizující obě chyby zároveň. Opět platí, že pro vyšší přesnost jsou potřeba metody vyšších řádů.

Příklad 7.4.2 (Metoda leap-frog)**.** Metoda *leap-frog* (též midpoint) je dvoukroková metoda druhého řádu. Aproximujeme derivaci v bodě t_n pomocí centrované diference

$$\frac{du(t_n)}{dt} \approx \frac{v^{n+1} - v^{n-1}}{2k} \approx f(v^n, t_n), \tag{7.8}$$

neboli

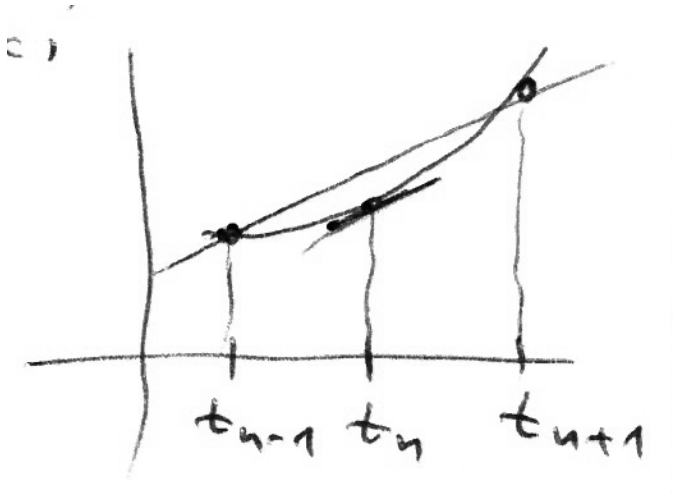
$$v^{n+1} = v^{n-1} + 2k\, f(v^n, t_n). \tag{7.9}$$

A tedy

$$u(t+k) - u(t-k) - 2k\, f(u(t), t) =$$

$$\begin{aligned} &= u(t) + ku'(t) + \frac{k^2}{2}u''(t) + \frac{k^3}{3!}u'''(t) + O(k^4) \\ &\quad - u(t) + ku'(t) - \frac{k^2}{2}u''(t) + \frac{k^3}{3!}u'''(t) + O(k^4) \quad - 2ku'(t) = \\ &= \frac{k^3}{3}u'''(t) + O(k^4) = O(k^{p+1}) \implies p = 2, \end{aligned}$$

neboť se navzájem vykrátí členy $u(t)$ s $-u(t)$, členy $\frac{k^2}{2}u''(t)$ s $-\frac{k^2}{2}u''(t)$ a členy $ku'(t) + ku'(t)$ s $-2ku'(t)$.



Obrázek 7.2: Náčrt k metodě leap-frog v osách (svislá osa vlevo, vodorovná osa času). Rostoucí křivka prochází třemi vyznačenými body nad dělicími body t_{n-1}, t_n, t_{n+1}; body v t_{n-1} a t_{n+1} spojuje přímka (sečna), v bodě t_n je nakreslen krátký úsek tečny. Od bodů na křivce vedou svislé spojnice k osám s ryskami popsanými t_{n-1}, t_n, t_{n+1}.

7.5 Jednokrokové metody vyšších řádů

Obecná jednokroková metoda je

$$v^{n+1} = v^n + k\, \Phi(v^n, t_n, k) \tag{7.10}$$

a její lokální diskretizační chyba je

$$u(t+k) - u(t) - k\,\Phi(u(t), t, k) = O(k^{p+1}). \tag{7.11}$$

Aby metoda byla alespoň řádu $p = 1$, musí platit

$$\Phi(u(t), t, 0) = f(u(t), t). \tag{7.12}$$

Vhodnou volbou Φ dostáváme metody vyšších řádů.

7.5.1 Metoda Taylorova rozvoje

U metody Taylorova rozvoje je nutné vyčíslit derivace $f(u(t), t)$. Plyne přímo dosazením za

$$u'(t) = f(u(t), t), \qquad u''(t) = \frac{\partial f}{\partial u}\,u'(t) + \frac{\partial f}{\partial t} \quad \text{atd.}$$

do rozvoje

$$u(t+k) = u(t) + ku'(t) + \frac{k^2}{2}u''(t) + \cdots,$$

čímž dostaneme

$$v^{n+1} = v^n + k \overbrace{\left[f(v^n, t_n) + \frac{k}{2}\left(\frac{\partial f}{\partial t}\Big|_{(v^n,t_n)} + f(v^n, t_n)\frac{\partial f}{\partial u}\Big|_{(v^n,t_n)}\right) + \cdots\right]}^{\Phi(v^n, t_n, k)}. \tag{7.13}$$

7.6 Rungeovy–Kuttovy metody

Rungeovy–Kuttovy metody jsou lepší v tom, že u nich není nutné počítat derivace f.

7.6.1 Motivace — metoda RK 2. řádu

Nejprve „nastřelíme“ do $t = t_n + \frac{k}{2}$ pomocí explicitní Eulerovy metody

$$v^{n+\frac{1}{2}} = v^n + \frac{k}{2}f(v^n, t_n) \tag{7.14}$$

a pak aproximujeme derivaci v $t = t_n + \frac{k}{2}$ jako

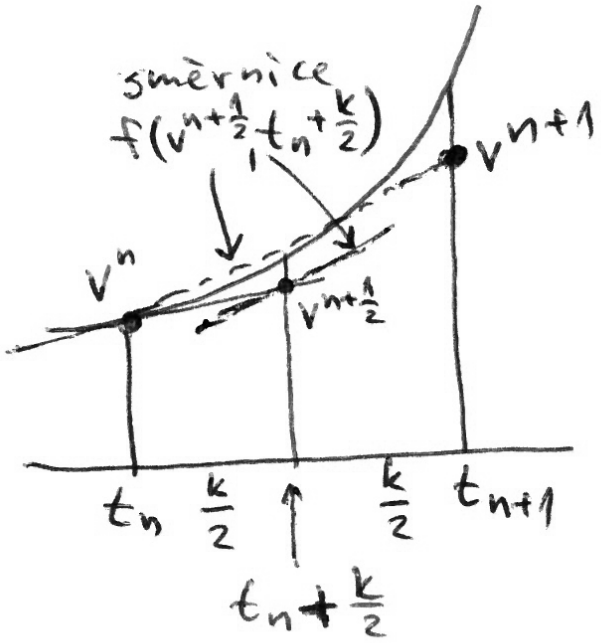
$$\underbrace{\frac{v^{n+1} - v^n}{k}}_{\substack{\text{centrovaná diference v bodě } t_n + \frac{k}{2}}} = f\left(v^{n+\frac{1}{2}}, t_n + \frac{k}{2}\right) = f\left(v^n + \frac{k}{2}f(v^n, t_n),\, t_n + \frac{k}{2}\right). \tag{7.15}$$

Jde o metodu 2. řádu, neboť

$$u(t+k) - u(t) - k\,f\left(u(t) + \frac{k}{2}f(u(t), t),\, t + \frac{k}{2}\right) =$$

$$= u(t) + ku'(t) + \frac{k^2}{2}u''(t) + O(k^3) \; - u(t) - kf(u(t), t)$$
$$- \frac{k^2}{2}\left(\frac{\partial f}{\partial u}f(u(t), t) + \frac{\partial f}{\partial t}\right) + O(k^3) \;=\; O(k^3).$$

Poznámka 7.6.1. Výsledný člen $O(k^3)$ už není nulový, ač jsme to neukázali.



Obrázek 7.3: Náčrt jednoho kroku RK metody 2. řádu: hladká rostoucí křivka řešení, na ní vyznačené body v^n (v t_n), $v^{n+\frac{1}{2}}$ (uprostřed) a v^{n+1} (v t_{n+1}), spojené úsečkami. Čárkovaně je naznačena tečna v v^n a plnou čarou sečna se směrnicí ve středním bodě. Popisky: „směrnice $f(v^{n+\frac{1}{2}}, t_n + \frac{k}{2})$“ se dvěma šipkami mířícími ke sklonům úseček, dále v^n, $v^{n+\frac{1}{2}}$, v^{n+1}. Na vodorovné ose zdola t_n, $\frac{k}{2}$, $\uparrow$, $\frac{k}{2}$, t_{n+1}, pod šipkou uprostřed $t_n + \frac{k}{2}$.

7.6.2 Obecný tvar a Butcherova tabulka

Obecnou metodu s r mezikroky (úrovněmi) lze psát ve tvaru

$$\begin{cases} y_1 = v^n + k \sum_{j=1}^{r} a_{1j}\, f(y_j, t_n + c_j k) \\ \quad \vdots \\ y_r = v^n + k \sum_{j=1}^{r} a_{rj}\, f(y_j, t_n + c_j k) \end{cases} \tag{7.16}$$

a

$$v^{n+1} = v^n + k \sum_{j=1}^{r} b_j\, f(y_j, t_n + c_j k), \tag{7.17}$$

což je ekvivalentní zápisu pomocí tzv. Butcherovy tabulky

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1r} \\ \vdots & \vdots & \ddots & \vdots \\ c_r & a_{r1} & \dots & a_{rr} \\ \hline & b_1 & \dots & b_r \end{array}$$

Většinou se používají explicitní metody. Pokud je libovolné $a_{ij} \neq 0$ pro $j \geq i$, pak jde o implicitní metodu.

Butcherova tabulka pro metodu 2. řádu (viz výše) je

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

a pro metodu 4. řádu

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

což je ekvivalentní předpisu

$$\begin{aligned} y_1 &= v^n, \\ y_2 &= v^n + \frac{k}{2} f(y_1, t_n)\,, \\ y_3 &= v^n + \frac{k}{2} f\left(y_2, t_n + \frac{k}{2}\right), \\ y_4 &= v^n + k\, f\left(y_3, t_n + \frac{k}{2}\right) \end{aligned}$$

a

$$\begin{aligned} v^{n+1} = v^n + \frac{k}{6}\Big[& f(y_1, t_n) + 2f\left(y_2, t_n + \frac{k}{2}\right) \\ & + 2f\left(y_3, t_n + \frac{k}{2}\right) + f(y_4, t_n + k)\Big]. \end{aligned} \tag{7.18}$$

7.6.3 Podmínky na řád metody

Aby bylo $p \geq 1$, tj. aby platilo $\Phi(v^n, t_n, 0) = f(v^n, t_n)$, musí zřejmě platit

$$\sum_{j=1}^{r} b_j = 1. \tag{7.19}$$

Pro $p \geq 2$ navíc dostaneme z Taylorova rozvoje (ponecháme pouze členy do k^2)

$$u(t+k) - u(t) - k\sum_{j=1}^{r} b_j\, f\left(u(t) + k\sum_{m=1}^{r} a_{jm} \underbrace{f\,(u(t), t)}_{\text{zde zahozeny členy } O(k)},\ t + c_j k\right) =$$

$$\begin{aligned} &= ku'(t) && + \frac{k^2}{2}u''(t) \ + O(k^3) \\ &\quad - k\sum_{j=1}^{r} b_j f(u(t), t) && - k^2 \sum_{j=1}^{r} b_j \left(\sum_{m=1}^{r} a_{jm}\frac{\partial f}{\partial u} f(u(t), t) \ + c_j \frac{\partial f}{\partial t}\right). \end{aligned}$$

Členy $ku'(t)$ a $-k\sum_{j=1}^{r} b_j f(u(t), t)$ se navzájem ruší právě tehdy, když

$$1 = \sum_{j=1}^{r} b_j,$$

a poslední člen $-k^2 \sum_{j=1}^{r} b_j(\dots)$ má být roven členu s druhou derivací, tj.

$$\frac{u''(t)}{2} = \frac{1}{2}\left(\frac{\partial f}{\partial u} f(u(t),t) + \frac{\partial f}{\partial t}\right).$$

A tedy podmínky pro $p \geq 2$ jsou

$$\sum_{m=1}^{r} a_{jm} = c_j \qquad \text{a} \qquad \sum_{j=1}^{r} b_j c_j = \frac{1}{2}. \tag{7.20}$$

Obecně existuje více metod určitého řádu (i pro explicitní metody), a proto se přidávají dodatečné podmínky, jako je stabilita, minimální koeficient u lokální chyby apod.

Podrobně viz Butcher: *Numerical Methods for ODE*, Wiley 2008.

Kapitola 8

Numerické řešení ODR – Lineární vícekrokové metody. Stabilita a konvergence

8.1 Lineární vícekrokové metody

Obecnou s-krokovou lineární metodu (anglicky *linear multistep method*) lze psát ve tvaru

$$\sum_{j=0}^{s} \alpha_j v^{n+j} = k \sum_{j=0}^{s} \beta_j f\left(v^{n+j}, t_{n+j}\right). \tag{8.1}$$

Obvykle se volí $\alpha_s = 1$ a musí být $\alpha_0 \neq 0$ nebo $\beta_0 \neq 0$, aby skutečně šlo o s-krokovou metodu. Pokud je $\beta_s = 0$, je metoda explicitní; pro $\beta_s \neq 0$ je implicitní.

Lokální diskretizační chyba je dána Taylorovým rozvojem

$$\sum_{j=0}^{s} \left[\alpha_j u(t+jk) - k\,\beta_j\, u'(t+jk)\right] = C_0 u(t) + C_1 k u'(t) + C_2 k^2 u''(t) + \dots \tag{8.2}$$

Pokud $C_0 = C_1 = \dots = C_p = 0$ a $C_{p+1} \neq 0$, máme metodu řádu p.

Dostaneme tak podmínky

$$C_0 = \alpha_0 + \alpha_1 + \dots + \alpha_s = \sum_{j=0}^{s} \alpha_j = 0,$$

$$C_1 = (\alpha_1 + 2\alpha_2 + \dots + s\,\alpha_s) - (\beta_0 + \beta_1 + \dots + \beta_s) = \sum_{j=0}^{s} (j\alpha_j - \beta_j) = 0,$$

což jsou obě nutné podmínky pro $p \geq 1$. Obecně je podmínka na m-tý koeficient tvaru

$$C_m = \sum_{j=0}^{s} \left(\frac{j^m}{m!}\,\alpha_j - \frac{j^{m-1}}{(m-1)!}\,\beta_j \right) = 0. \tag{8.3}$$

Vidíme, že by mohla existovat s-kroková metoda s řádem přesnosti $2s$ (volíme $\alpha_s = 1$ a zbývá $2s+1$ neznámých), ovšem objeví se problémy se stabilitou, ke kterým se dostaneme později. Dahlquist ukázal, že stabilní s-krokové lineární metody mohou mít řád přesnosti nejvýše $s+2$.

8.1.1 Standardní vícekrokové metody

Standardně se používají tři skupiny metod:

1) explicitní Adamsovy–Bashforthovy metody,

2) implicitní Adamsovy–Moultonovy metody,

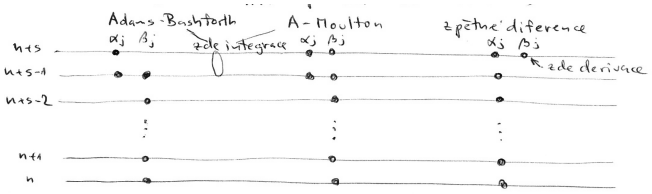
3) implicitní zpětné diference.

První dvě skupiny vycházejí z integrace interpolačního polynomu $q(t)$, totiž

$$v^{n+s} - v^{n+s-1} = \int_{t_{n+s-1}}^{t_{n+s}} q(t)\,\mathrm{d}t, \tag{8.4}$$

kde polynom $q(t)$ prochází body $\{t_{n+i}, f_{n+i}\}$, a to pro $i = 0, \ldots, s-1$ v případě Adamsových–Bashforthových metod a pro $i = 0, \ldots, s$ v případě Adamsových–Moultonových metod.

Třetí skupina, tedy implicitní zpětné diference, vychází z aproximace derivace v bodě t_{n+s} pomocí vztahu $kD = -\ln(I - \nabla)$.



Obrázek 8.1: Schéma pěti vodorovných časových hladin označených shora dolů $n+s$, $n+s-1$, $n+s-2$, $n+1$, n (mezi $n+s-2$ a $n+1$ je svislá trojtečka naznačující vynechané hladiny), na nichž jsou ve třech svislých blocích vyznačeny tečkami nenulové koeficienty. První blok je nadepsán „Adams–Bashforth“ se sloupci α_j, β_j; tečky α_j jsou na hladinách $n+s$ a $n+s-1$, tečky β_j na hladinách $n+s-1, n+s-2, \ldots, n+1, n$. Druhý blok „A-Moulton“ má sloupce α_j, β_j; tečky α_j na $n+s$ a $n+s-1$, tečky β_j na $n+s, n+s-1, n+s-2, \ldots, n+1, n$. Třetí blok „zpětné diference“ má sloupce α_j, β_j; tečky α_j na $n+s, n+s-1, n+s-2, \ldots, n+1, n$, tečka β_j pouze na $n+s$. Mezi prvním a druhým blokem je nakreslena svislá elipsa mezi hladinami $n+s$ a $n+s-1$ a popiska „zde integrace“, k níž vedou dvě čáry od nadpisů „Adams–Bashforth“ a „A-Moulton“. Vpravo šipka směřuje k tečce β_j na hladině $n+s$ s popiskou „zde derivace“.

8.1.2 Určení koeficientů pomocí operátorového počtu

Koeficienty α_j a β_j by šlo určit i z podmínek na přesnost metod, tj. z $c_0 = \alpha_0 + \cdots + \alpha_s = 0$ a tak dále. Elegantněji to však lze udělat pomocí Taylorova rozvoje následujícím trikem: použijeme-li operátor posunutí E a operátor derivace D, můžeme operátor daný lineární vícekrokovou metodou

$$Lu(t_n) = \sum_{j=0}^{r} \alpha_j\, u(t_{n+j}) - k \sum_{j=0}^{r} \beta_j\, u'(t_{n+j}) \tag{8.5}$$

psát jako

$$L = \rho(E) - kD\,\sigma(E), \quad \text{kde } \rho(z) = \sum_{j=0}^{r} \alpha_j z^j \quad \text{a } \sigma(z) = \sum_{j=0}^{s} \beta_j z^j. \tag{8.6}$$

Ve členu $kD\,\sigma(E)$ přitom lze operátory E a D libovolně přehazovat, neboť spolu komutují.

Pomocí vztahu $E = e^{kD}$ pak dostaneme

$$L = \rho(e^{kD}) - kD\,\sigma(e^{kD}) = \rho(e^x) - x\,\sigma(e^x) = c_0 + c_1 x + \ldots, \tag{8.7}$$

neboli lineární vícekroková metoda má řád přesnosti p právě tehdy, pokud Taylorův rozvoj funkce $\rho(e^x) - x\,\sigma(e^x)$ začíná členem x^{p+1}.

Případně můžeme položit $e^x = z$, neboli $x = \ln z$ (nyní pro $x \to 0$ máme $z \to 1$), a po vydělení $\sigma(z)$ lze psát (pokud $\sigma(1) \neq 0$)

$$\frac{\rho(z)}{\sigma(z)} = \ln z + \mathcal{O}\big((z-1)^{p+1}\big) \qquad \text{pro } z \to 1. \tag{8.8}$$

Poznámka 8.1.1. Podmínky pro přesnost $p \geq 1$ lze psát též jako

$$\rho(1) = \sum_{j=0}^{s} \alpha_j = 0 \qquad \text{a} \qquad \rho'(1) = \sigma(1).$$

Příklad 8.1.2 (Adamsovy–Bashforthovy metody)**.** Zde je požadavek $\rho(z) = z^s - z^{s-1}$, a tedy $\sigma(z)$ lze určit jako Taylorův rozvoj funkce

$$\frac{\rho(z)}{\ln z} = \frac{z^s - z^{s-1}}{\ln z}$$

v $z \to 1$ do řádu $s-1$. Pro Adamsovy–Moultonovy metody se postupuje obdobně, jen se rozvíjí do řádu s.

8.2 Stabilita a konvergence

Řekneme, že určitá metoda je *konvergentní*, pokud se pro časový krok $k \to 0$ blížíme ke správnému řešení, a že je *stabilní*, pokud při dané metodě nedochází ke zvětšování lokálních diskretizačních a zaokrouhlovacích chyb.

Přestože metoda může být stabilní a konvergentní pro dostatečně malý časový krok k, nemusí být stabilní pro konečná (a i rozumně velká) k. Proto se zavádějí pojmy jako *zero-stability* (pro $k \to 0$) nebo *absolute stability*, tedy absolutní stabilita, pro konečná k a podobně. Některé z nich si definujeme později.

Definice 8.2.1 (Konvergence s-krokové metody)**.** s-kroková metoda je *konvergentní*, pokud pro libovolnou ODR $u'(t) = f(u,t)$, kde $f(u,t)$ je spojitá a stejnoměrně Lipschitzovská v u, a pro libovolné počáteční podmínky $u(t_0) = U_0$ splňující

$$\lim_{k \to 0} v^i(k) = U_0 \quad \text{pro } i = 0, 1, \ldots, s-1 \tag{8.9}$$

platí

$$\lim_{\substack{k \to 0 \\ Nk = T}} v^N = U(T) \tag{8.10}$$

pro libovolné $T > 0$, pro které má ODR jednoznačné řešení.

Poznámka 8.2.2. K této definici se hodí tři poznámky:

1) Podmínka (8.9) je přirozená, máme-li spojité řešení.

2) Pro pevné T jde $N \to \infty$.

3) Existují i metody, které jsou konvergentní jen pro určité ODR či pro určité počáteční podmínky. My se budeme zabývat jen metodami konvergujícími pro libovolný počáteční problém.

8.3 Konvergence explicitních jednokrokových metod

Ač si ukážeme, že za poměrně obecných podmínek jde o konvergentní metody, nemusí být tyto metody nutně stabilní pro konečné k. K oblastem absolutní stability se dostaneme později.

8.3.1 Explicitní Eulerova metoda je konvergentní

Uvažujme nejprve lineární problém

$$u'(t) = \lambda u(t) + g(t), \qquad u(t_0) = u_0. \tag{8.11}$$

Eulerova metoda iteruje

$$\begin{aligned} v^{n+1} &= v^n + k\left[\lambda v^n + g(t_n)\right] = \\ &= (1 + k\lambda)v^n + kg(t_n) \end{aligned}$$

a z Taylorova rozvoje máme

$$u(t_{n+1}) = (1 + k\lambda)u(t_n) + kg(t_n) + \tfrac{1}{2}k^2u'(t_n) + \mathcal{O}(k^3), \tag{8.12}$$

přičemž poslední dva členy $\frac{1}{2}k^2u'(t_n) + \mathcal{O}(k^3)$ dohromady tvoří lokální diskretizační chybu $k\tau^n$, která je vždy přítomna pro konzistentní metodu s $p \geq 1$.

Odečtením dostaneme

$$e^{n+1} = v^{n+1} - u(t_{n+1}) = (1 + k\lambda)e^n - k\tau^n, \tag{8.13}$$

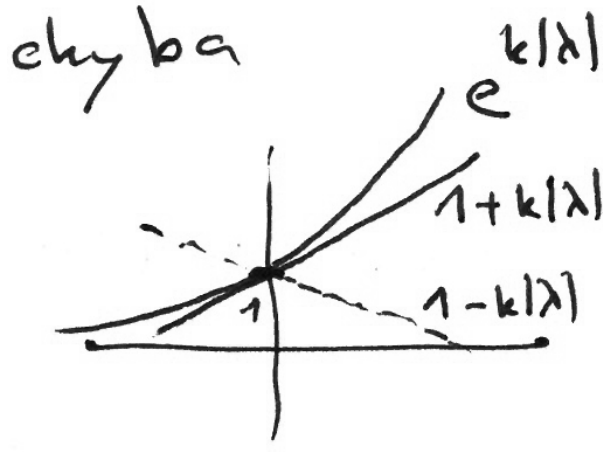
kde e^{n+1} je globální chyba v čase t_{n+1}.

Poznámka 8.3.1. Horní index n u e^n, v^n či τ^n je skutečně index, nikoli mocnina.

Rozepsáním rekurence (8.13) dostaneme

$$e^n = (1 + k\lambda)^n e^0 - k\sum_{m=1}^{n}(1 + k\lambda)^{n-m}\tau^{m-1}, \tag{8.14}$$

kde e^0 je počáteční zaokrouhlovací chyba.



Obrázek 8.2: Náčrt souřadných os s vyznačenou hodnotou 1 na svislé ose; z tohoto bodu vycházejí tři křivky — nejstrmější plná křivka označená $e^{k|\lambda|}$, plná přímka označená $1 + k|\lambda|$ a klesající čárkovaná čára označená $1 - k|\lambda|$.

Využijeme-li odhad (pro kladné k a libovolné λ)

$$(1 + k\lambda)^{n-m} \leq e^{(n-m)k|\lambda|} \leq e^{nk|\lambda|} \leq e^{|\lambda|T} \tag{8.15}$$

platný pro $nk \leq T$, tedy pro všechny časy $0 \leq t_n \leq T$, dostaneme

$$|e^n| \leq e^{|\lambda|T} \left[|e^0| + k \sum_{m=1}^{n} |\tau^{m-1}| \right] \leq e^{|\lambda|T} \left(|e^0| + kn \max_{n=1}^{n} |\tau^{m-1}| \right) \leq$$
$$\leq e^{|\lambda|T} \left(|e^0| + T\|\tau\|_\infty \right),$$

kde $\|\tau\|_\infty$ je alespoň $\mathcal{O}(k)$.

Pro $k \to 0$ jde tedy druhý člen k nule a první závisí na počáteční zaokrouhlovací chybě a na typu úlohy. Pokud $e^0 = 0$, máme konvergenci pro $k \to 0$ a jde o metodu řádu 1, neboť globální chyba je $\mathcal{O}(k)$.

Pro nelineární problém dostaneme obdobně

$$e^{n+1} = e^n + k\,[f(v^n, t_n) - f(u(t_n), t_n)] - k\tau^n \tag{8.16}$$

a díky stejnoměrné Lipschitzovosti můžeme psát

$$|f(v^n, t_n) - f(u(t_n), t_n)| \leq L\,|v^n - u(t_n)| = L\,|e^n|. \tag{8.17}$$

Máme tedy nyní odhad

$$|e^{n+1}| \leq (1 + kL)\,|e^n| + k\,|\tau^n|, \tag{8.18}$$

neboli

$$|e^n| \leq (1 + kL)^n |e^0| + k \sum_{m=1}^{n} (1 + kL)^{n-m} |\tau^{m-1}|, \tag{8.19}$$

a konečně

$$|e^n| \leq e^{LT} \left(|e^0| + T\|\tau\|_\infty \right), \tag{8.20}$$

kde člen $T\|\tau\|_\infty$ je $\mathcal{O}(k)$, stejně jako v předchozím případě.

8.3.2 Explicitní Rungeovy–Kuttovy metody jsou konvergentní

Obecně lze pro tyto metody psát

$$v^{n+1} = v^n + k\,\Phi(v^n, t_n, k), \tag{8.21}$$

kde Φ závisí na pravé straně $f(u, t)$ a stejně jako u f předpokládáme, že je spojitá a stejnoměrně Lipschitzovská, ovšem s jinou konstantou L'.

Příklad 8.3.2 (Rungeova–Kuttova metoda 2. řádu)**.** Rungeova–Kuttova metoda 2. řádu má

$$\Phi = f\left(v^n + \tfrac{1}{2}kf(v^n, t_n), t_n\right),$$

z čehož dostaneme $L' = L + \frac{1}{2}kL^2$, a pro $k \to 0$ je $L' \to L$.

Pro chybu nyní dostaneme (obdobně jako u Eulerovy metody)

$$e^{n+1} = e^n + k\,[\Phi(v^n, t_n, k) - \Phi(u(t_n), t_n, k)] - k\tau^n, \tag{8.22}$$

a tedy

$$|e^{n+1}| \leq |e^n|(1 + kL') + k\,|\tau^n|. \tag{8.23}$$

Dále už postupujeme stejně jako výše, čímž dostaneme konvergenci.

8.4 Řešení homogenních lineárních diferenčních rovnic

Uvažujme diferenční rovnici k-tého řádu

$$a_k f_{n+k} + \cdots + a_1 f_{n+1} + a_0 f_n = 0, \tag{8.24}$$

kde a_i jsou reálná čísla ($a_k \neq 0$) a f_n je zadána pro $n = 0, 1, \ldots, k-1$.

Obecně, tedy bez počátečních hodnot, existuje k nezávislých řešení. Jde totiž o k vektorů tvaru

$$(f_n, \ldots, f_{n+k}),$$

které jsou kolmé na vektor koeficientů $(a_0, \ldots, a_k)$. Bázi tohoto k-rozměrného prostoru řešení lze dostat následujícím trikem.

S využitím $f_{n+i} = E^i f_n$ přepíšeme rovnici (8.24) jako

$$p(E) f_n = 0, \tag{8.25}$$

kde

$$p(\lambda) = a_k \lambda^k + \cdots + a_1 \lambda + a_0 = a_k (\lambda - \lambda_1)^{\alpha_1} \ldots (\lambda - \lambda_m)^{\alpha_m} \tag{8.26}$$

a kde $\lambda_1, \ldots, \lambda_m$ jsou kořeny polynomu $p(\lambda)$ s násobnostmi $\alpha_1, \ldots, \alpha_m$ splňujícími $\alpha_1 + \alpha_2 + \cdots + \alpha_m = k$.

Rovnici (8.24) tedy můžeme psát též jako

$$p(E) f_n = a_k (E - \lambda_1)^{\alpha_1} \ldots (E - \lambda_m)^{\alpha_m} f_n = 0,$$

a protože $E\lambda^n = \lambda^{n+1} = \lambda\,\lambda^n$, bude $f_n = \lambda^n$ řešením rovnice (8.24), pokud je λ kořenem charakteristického polynomu $p(\lambda)$.

Pokud je násobnost všech kořenů $\alpha_i = 1$, pak máme k nezávislých řešení. Pro násobné kořeny obecně platí

$$(E - \lambda)^{\alpha_i} q_{\alpha_i - 1}(n)\, \lambda^n = 0, \tag{8.27}$$

kde $q_N(n)$ je libovolný polynom N-tého stupně. Platí totiž

$$(E - \lambda)\, q_N(n) \lambda^n = q_N(n+1)\lambda^{n+1} - q_N(n)\lambda^{n+1} = \tilde{q}_{N-1}(n)\lambda^{n+1},$$

a tedy postupně snižujeme stupeň q až na nulu, takže pro $(E - \lambda)^{N+1}$ máme

$$(E - \lambda)^{N+1} q_N(n) \lambda^n = 0. \tag{8.28}$$

Jako α_i nezávislých řešení pro násobný kořen λ_i tedy můžeme vzít

$$\lambda_i^n,\ n\lambda_i^n,\ \ldots,\ n^{\alpha_i - 1}\lambda^n. \tag{8.29}$$

Řešení rovnice (8.24) s počátečními podmínkami pro $n = 0, 1, \ldots, k-1$ pak dostaneme jako lineární kombinaci takovýchto bázových funkcí. Koeficienty tohoto rozkladu do báze určíme řešením soustavy lineárních rovnic daných počátečními hodnotami $f_0, \ldots, f_{k-1}$.

Příklad 8.4.1 (Fibonacciho posloupnost)**.** Uvažujme

$$f_{n+2} = f_{n+1} + f_n, \qquad f_0 = 0, \quad f_1 = 1. \tag{8.30}$$

Charakteristický polynom je

$$p(\lambda) = \lambda^2 - \lambda - 1 \qquad \text{s kořeny } \lambda_{1,2} = \frac{1 \pm \sqrt{5}}{2},$$

a tedy řešení bude mít tvar

$$f_n = A\lambda_1^n + B\lambda_2^n,$$

kde A a B jsou dány rovnicemi

$$f_0 = 0 = A + B,$$
$$f_1 = 1 = A\,\frac{1+\sqrt{5}}{2} + B\,\frac{1-\sqrt{5}}{2},$$

z nichž plyne $A = -B = \frac{\sqrt{5}}{5}$. Řešením je tedy

$$f_n = \frac{\sqrt{5}}{5}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]. \tag{8.31}$$

8.5 Stabilita a konvergence lineárních vícekrokových metod

Zde je situace složitější a ani konzistentní metoda ($p \geq 1$) nemusí být stabilní, a tedy konvergentní.

Příklad 8.5.1 (Nestabilní dvoukroková metoda)**.** Explicitní lineární vícekroková metoda s nejvyšším řádem přesnosti $p = 2s - 1 = 3$ je dána vztahem

$$v^{n+2} = -4v^{n+1} + 5v^n + k\left[4f(v^{n+1}, t_{n+1}) + 2f(v^n, t_n)\right]. \tag{8.32}$$

Poznámka 8.5.2. Zde máme čtyři neznámé $\alpha_0, \alpha_1, \beta_0, \beta_1$ a při volbě $\alpha_2 = 1$ pro ně dostáváme podmínky

$$\begin{aligned} 1 + \alpha_1 + \alpha_0 &= 0,\\ 2 + \alpha_1 - \beta_0 - \beta_1 &= 0,\\ 2 + \tfrac{1}{2}\alpha_1 - \beta_1 &= 0,\\ \tfrac{4}{3} + \tfrac{1}{6}\alpha_1 - \tfrac{1}{2}\beta_1 &= 0, \end{aligned}$$

z nichž plyne $\alpha_0 = -5$, $\alpha_1 = 4$, $\beta_0 = 2$, $\beta_1 = 4$.

Použijeme-li tuto metodu na rovnici $u'(t) = 0$, $u(0) = 0$, pak pro inicializaci $v^0 = v^1 = 0$ je řešení v pořádku. Avšak pro $v^0 = 0$, $v^1 = k$, což je inicializace splňující podmínku pro konvergenci, dostaneme

$$v^n = \frac{k}{6}\left(1 - (-5)^n\right), \tag{8.33}$$

a tedy nepřesnosti z předchozích kroků jsou násobeny faktorem 5, což je exponenciální růst.

Poznámka 8.5.3. Charakteristický polynom je zde $p(\lambda) = \lambda^2 + 4\lambda - 5$ s kořeny $\lambda_1 = 1$ a $\lambda_2 = -5$, takže $v^n = A \cdot 1^n + B(-5)^n$. Z počátečních podmínek máme rovnice

$$v^0 = 0 = A + B,$$
$$v^1 = k = A - 5B,$$

z nichž plyne $A = -B = \frac{k}{6}$.

Příklad 8.5.4 (Implicitní dvoukroková metoda řádu 3)**.** Uvažujme metodu

$$v^{n+2} - 2v^{n+1} + v^n = \frac{k}{2}\left[f(v^{n+2}, t_{n+2}) - f(v^n, t_n)\right]. \tag{8.34}$$

Nyní je $p(\lambda) = \lambda^2 - 2\lambda + 1 = (\lambda - 1)^2$; opět metodu aplikujeme na $u'(t) = 0$. Obecné řešení má tvar $v^n = A \cdot 1^n + Bn1^n$ a počáteční podmínky

$$v^0 = 0 = A,$$
$$v^1 = k = B$$

dávají řešení $v^n = kn$. Metoda je tedy opět nestabilní, ale jen polynomiálně.

Pokud bychom měli vícenásobný kořen λ_i, pro který by platilo $|\lambda_i| < 1$, pak bychom měli stabilní metodu, neboť pro libovolné p platí

$$\lim_{n\to\infty} n^p \lambda_i^n = 0. \tag{8.35}$$

Vidíme tedy, že kořeny charakteristického polynomu

$$\rho(\lambda) = \sum_{j=0}^{s} \alpha_j \lambda^j \tag{8.36}$$

lineární vícekrokové metody musí být v absolutní hodnotě menší než jedna pro vícenásobné kořeny, zatímco pro jednonásobné kořeny může být $|\lambda_i| \leq 1$.

Takto se definuje stabilita lineární vícekrokové metody; jde o tzv. zero-stability.

Definice 8.5.5 (Stabilita s-krokové lineární metody)**.** s-kroková lineární metoda je *stabilní*, pokud je řešení

$$\rho(E)v^n = 0$$

omezené pro $n \to \infty$, neboli pokud $|\lambda_i| \leq 1$ a pro vícenásobné kořeny ρ navíc $|\lambda_i| < 1$.

Příklad 8.5.6 (Adamsovy metody)**.** Adamsovy–Bashforthovy i Adamsovy–Moultonovy metody jsou stabilní v tomto smyslu, neboť obecně

$$\rho(\lambda) = \lambda^s - \lambda^{s-1} \implies \lambda_1 = 1, \quad \lambda_2 = \lambda_3 = \cdots = \lambda_s = 0,$$

přičemž $\lambda_1 = 1$ je tzv. principální (hlavní) kořen.

Věta 8.5.7 (Dahlquistův teorém ekvivalence)**.** *Lineární vícekroková metoda pro $u'(t) = f(u(t), t)$ je konvergentní právě tehdy, když je konzistentní ($p \geq 1$) a stabilní ve výše uvedeném smyslu.*

Stačí tedy zkoumat stabilitu, tj. kořeny $\rho(\lambda)$, abychom ověřili, že metoda konverguje.

Obecně lze též ukázat (rovněž Dahlquist), že každá stabilní lineární s-kroková metoda může mít řád p nejvýše

$$\begin{array}{ll} s+2 & \text{pro } s \text{ sudé,} \\ s+1 & \text{pro } s \text{ liché,} \end{array} \quad \text{a } s \text{ pro explicitní metody.} \tag{8.37}$$

Zpětné diference jsou stabilní pro $1 \leq s \leq 6$.

Kapitola 9

Numerické řešení ODR – Absolutní stabilita. Systémy se silným tlumením. Adaptivní metody

9.1 Absolutní stabilita

Metoda, která je stabilní pro $k \to 0$, může být nestabilní pro konečné k, a to nezávisle na tom, zda jde o jedno- nebo vícekrokovou metodu.

Příklad 9.1.1. Uvažujme ODR

$$u'(t) = \lambda u(t)$$

s počáteční podmínkou $u(t_0) = u_0$. Jejím řešením je $u(t) = u_0 e^{\lambda(t-t_0)}$. Pro Eulerovu metodu máme odhad

$$e^{n+1} = (1 + k\lambda)\, e^n - k\tau^n, \tag{9.1}$$

takže k růstu chyby nedochází, pokud

$$|1 + k\lambda| \leq 1, \tag{9.2}$$

což platí pro $-2 \leq k\lambda \leq 0$, tj. pouze pro $\lambda \leq 0$ a pro $k \leq \left|\frac{2}{\lambda}\right|$.

Poznámka 9.1.2. Odhad

$$|e^n| \leq e^{|\lambda|T}\, T\, \|\tau\|_\infty = \mathcal{O}(k)$$

stále platí pro $k \to 0$, ale výraz $e^{|\lambda|T}T$ může být velký, a tedy k musí být velmi malé, aby byl tento odhad rozumný.

Pro vyšetřování stability numerického řešení ODR se proto zavádějí pojmy *absolutní stabilita* a *oblast absolutní stability*, které říkají, pro jaká konečná k je metoda stabilní. Jsou založené na vyšetřování stability pro lineární problém

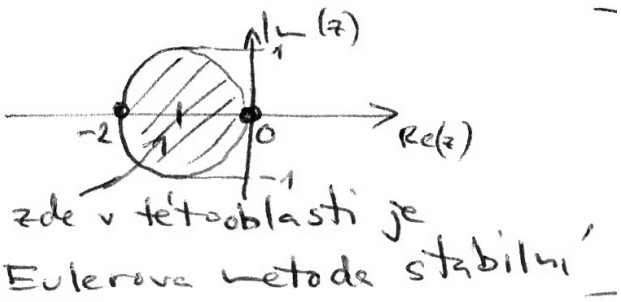
$$u'(t) = \lambda u(t) + g(t), \qquad u(t_0) = u_0, \tag{9.3}$$

a obecně hledáme oblast v komplexní rovině, pro kterou je metoda stabilní, s tím, že komplexní proměnnou je součin $z = k\lambda$.

Příklad 9.1.3. Pro Eulerovu metodu máme podmínku

$$|1 + k\lambda| = |1 + z| \leq 1,$$

tedy oblastí stability je kruh o poloměru 1 se středem v bodě -1 (viz obrázek 9.1).



Obrázek 9.1: Komplexní rovina s osami Im(z) (svislá) a Re(z) (vodorovná); vyšrafovaný kruh o poloměru 1 se středem v -1, procházející body -2 a 0. V této oblasti je Eulerova metoda stabilní.

Pro obecnou lineární vícekrokovou metodu, kterou pro problém $u'(t) = \lambda u(t)$ můžeme psát ve tvaru

$$\sum_{j=0}^{s} \alpha_j v^{n+j} = k \sum_{j=0}^{s} \beta_j \lambda v^{n+j} \quad \text{neboli} \quad \sum_{j=0}^{s} (\alpha_j - z\beta_j)\, v^{n+j} = 0, \tag{9.4}$$

bude podmínkou stability omezenost řešení této lineární diferenční rovnice.

9.1.1 Oblast stability lineárních vícekrokových metod

Definice 9.1.4 (Absolutní stabilita)**.** Řekneme, že *lineární vícekroková metoda je absolutně stabilní* pro konečné $z = k\lambda$, pokud každé řešení diferenční rovnice

$$\pi_z(E)\, v^n = \big(\varrho(E) - z\sigma(E)\big)\, v^n = 0 \tag{9.5}$$

je omezené, neboli pokud pro všechny kořeny tzv. *polynomu stability*

$$\pi_z(\xi) = \varrho(\xi) - z\sigma(\xi) = \sum_{j=0}^{s} \alpha_j \xi^j - z \sum_{j=0}^{s} \beta_j \xi^j \tag{9.6}$$

platí $|\xi_i| \le 1$, přičemž $|\xi_i| = 1$ může nastat jen pro jednoduché kořeny.

Definice 9.1.5 (Oblast absolutní stability)**.** *Oblast (absolutní) stability* je oblast v komplexní rovině z, kde je lineární vícekroková metoda absolutně stabilní, tj.

$$S = \{z \in \mathbb{C};\ \pi_z(\xi) \text{ má kořeny } |\xi_i| \le 1 \text{ a } |\xi_i| = 1 \text{ platí pro jednoduché kořeny}\}. \tag{9.7}$$

Poznámka 9.1.6. Pro $z = 0$ máme podmínku na zero-stabilitu, a tedy metoda je stabilní pro $k \to 0$, pokud $z = 0 \in S$.

Příklad 9.1.7.

1. Eulerova metoda $v^{n+1} - v^n = k\lambda v^n$:

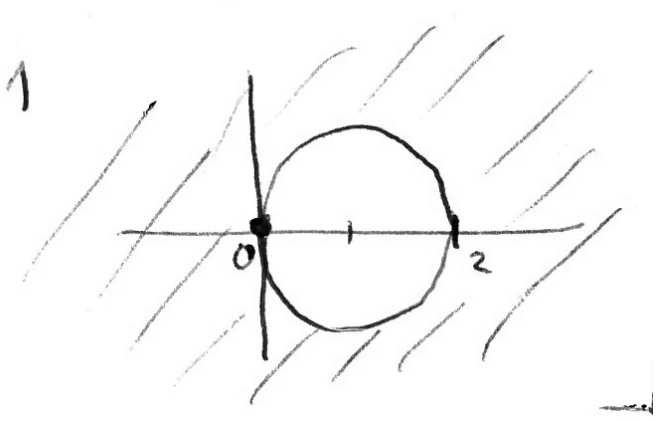
$$\left.\begin{aligned} \varrho(\xi) &= \xi - 1 \\ \sigma(\xi) &= 1 \end{aligned}\right\} \quad \pi_z(\xi) = \xi - 1 - z \ \Rightarrow\ |\xi| = |1 + z| \le 1$$

   jako výše.

2. Implicitní Eulerova metoda $v^{n+1} - v^n = k\lambda v^{n+1}$:

$$\left.\begin{aligned} \varrho(\xi) &= \xi - 1 \\ \sigma(\xi) &= \xi \end{aligned}\right\} \quad \pi_z(\xi) = \xi - 1 - z\xi \ \Rightarrow\ |\xi| = \frac{1}{|1 - z|} \le 1,$$

   neboli $|1 - z| \ge 1$. Tedy např. pro $\lambda \le 0$ je metoda stabilní pro libovolné $k > 0$ (viz obrázek 9.2).



Obrázek 9.2: Komplexní rovina z s kružnicí se středem v bodě 1 a poloměrem 1, procházející body 0 a 2. Vyšrafovaná oblast vně kružnice je oblastí stability implicitní Eulerovy metody, $|1-z| \geq 1$.

9.1.2 Hranice oblasti stability

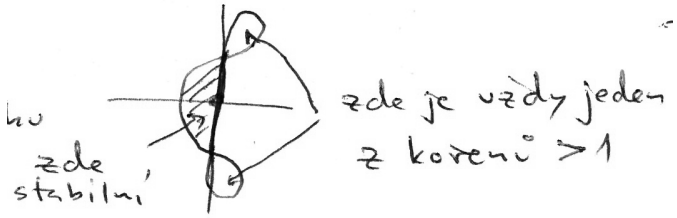
Obecně lze pro lineární vícekrokové metody určit hranici oblasti stability S tak, že kořen s $|\xi_i| = 1$ musí mít tvar $e^{i\theta}$, a tedy na hranici musí mít alespoň jeden kořen tento tvar. Z podmínky $\varrho(e^{i\theta}) - z\sigma(e^{i\theta}) = 0$ dostáváme body

$$z = \frac{\varrho(e^{i\theta})}{\sigma(e^{i\theta})}, \tag{9.8}$$

a měníme-li $\theta \in \langle 0, 2\pi \rangle$, dostaneme křivku v komplexní rovině, která může být hranicí oblasti stability.

Problém ještě je, že vnitřek či vnějšek oblasti ohraničené touto křivkou může, ale nemusí patřit do oblasti stability S dané metody; je nutné to vyzkoušet alespoň pro jeden bod z dané podoblasti.

Příklad 9.1.8. Adamsova–Bashforthova metoda 4. řádu dá křivku znázorněnou na obrázku 9.3.



Obrázek 9.3: Uzavřená křivka (hranice oblasti stability) Adamsovy–Bashforthovy metody 4. řádu v komplexní rovině; nahoře i dole má malé smyčky. Úzká šrafovaná část uvnitř křivky vlevo od imaginární osy je oblastí stability, vpravo je vždy jeden z kořenů větší než 1.

Příklad 9.1.9. Pro midpoint metodu (též leap-frog)

$$v^{n+1} - v^{n-1} = 2\lambda k\, v^n \tag{9.9}$$

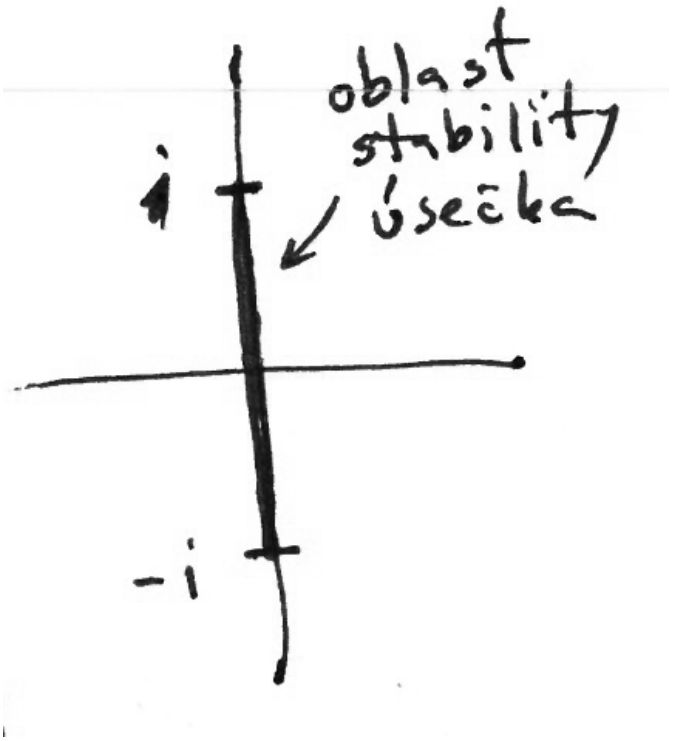
je

$$\left.\begin{aligned} \varrho(\xi) &= \xi^2 - 1 \\ \sigma(\xi) &= 2\xi \end{aligned}\right\} \Rightarrow \pi_z(\xi) = \xi^2 - 2z\xi - 1 = 0$$

s kořeny

$$\xi_{1,2} = \frac{2z \pm \sqrt{4z^2+4}}{2} = z \pm \sqrt{z^2+1}.$$

Pro $z = i\alpha$, $|\alpha| < 1$, máme $\xi_{1,2} = i\alpha \pm \sqrt{1-\alpha^2}$, a tedy $|\xi_{1,2}| = 1$, ale jde o různé kořeny. Pro $z = \pm i$ budou $\xi_1 = \xi_2 = \pm i$ (dvojnásobné kořeny). Pro libovolné jiné z bude jeden z kořenů v absolutní hodnotě větší než 1. Oblastí stability je tedy pouze úsečka na imaginární ose (viz obrázek 9.4).



Obrázek 9.4: Oblast stability midpoint metody: úsečka na imaginární ose mezi body i a $-i$.

V blízkosti $z = 0$ je obecně vždy jeden kořen

$$\xi_1(z) = e^z + \mathcal{O}(z^{p+1}), \tag{9.10}$$

avšak $|e^z| < 1$ pouze pro $\mathrm{Re}(z) < 0$, a proto lze očekávat, že oblast stability bude nalevo od $z = 0$ (ovšem existují výjimky, jako je midpoint metoda výše).

9.1.3 Jednokrokové metody typu Runge–Kutta

Pro *jednokrokové metody* typu Runge–Kutta dostaneme pro ODR $u'(t) = \lambda u(t)$ obecně

$$v^{n+1} = R(z)\, v^n, \qquad z = k\lambda, \tag{9.11}$$

kde $R(z)$ je obvykle polynom pro explicitní metody a racionální funkce pro implicitní metody. Přitom platí $R(z) = e^z + \mathcal{O}(z^{p+1})$ pro $z \to 0$ (Taylorův rozvoj, resp. Padé aproximace e^z).

Příklad 9.1.10. Pro Rungeovu–Kuttovu metodu 2. řádu a ODR $u'(t) = \lambda u(t)$ dostaneme

z obecného předpisu

$$\left.\begin{aligned} y_1 &= v^n \\ y_2 &= v^n + \frac{k}{2} f(u^n) \\ v^{n+1} &= v^n + k f(y_2) \end{aligned}\right\} \Rightarrow \begin{aligned} v^{n+1} &= v^n + k\lambda\left(v^n + \frac{k}{2}\lambda v^n\right) = \\ &= \underbrace{\left(1 + z + \frac{z^2}{2}\right)}_{e^z + \mathcal{O}(z^3)} v^n. \end{aligned}$$

Pokud $|R(z)| \leq 1$, je metoda stabilní a oblast absolutní stability je ohraničena konturou danou rovností $|R(z)| = 1$ (viz ukázky v Mathematice).

9.2 Stabilita pro systém ODR

Máme-li soustavu m rovnic, obecně nelineárních, můžeme odhadnout, zda bude určitá metoda stabilní pro určitý krok k v určitém čase $\tilde{t}$, pomocí linearizace. Zapíšeme-li řešení jako $u(t) = \tilde{u} + w(t)$, kde $\tilde{u} = u(\tilde{t})$, pak pro $w(t)$ dostaneme

$$\begin{aligned} w'(t) = u'(t) &= f(\tilde{u} + w(t), t) = \\ &= f(\tilde{u}, t) + J(\tilde{u}, \tilde{t})\, w(t) + \mathcal{O}(\|w\|^2), \end{aligned} \tag{9.12}$$

kde jakobián $J_{ij} = \dfrac{\partial f_i}{\partial u_j}$ je vyčíslený v $\tilde{u}$ a $\tilde{t}$. Neboli $w(t)$ splňuje přibližně lineární systém ODR

$$w'(t) = Aw(t) + b \qquad \big(A = J(\tilde{u}, \tilde{t})\big), \tag{9.13}$$

a stabilita určité metody bude záviset na A (obdoba λ).

Je-li matice A systému $u'(t) = Au(t)$ diagonalizovatelná, tj. $A = R\Lambda R^{-1}$, kde Λ je diagonální matice s vlastními čísly $\underbrace{(\lambda_1, \ldots, \lambda_m)}_{\text{vlastní čísla}}$ na diagonále a

$$R = (r_1 | r_2 | \ldots | r_m)$$

je matice vlastních vektorů, pak je systém $u'(t) = Au(t)$ ekvivalentní systému

$$R^{-1}u'(t) = (R^{-1}AR)\, R^{-1}u(t), \tag{9.14}$$

neboli $v'(t) = \Lambda v(t)$, což je m nezávislých rovnic. Pro každou z nich musí platit $k\lambda_i \in S$ dané metody, aby byla stabilní.

Příklad 9.2.1 (Chemická kinetika)**.** Uvažujme reakci $A \overset{\beta_1}{\to} B \overset{\beta_2}{\to} C$, kde β_1 a β_2 jsou rychlostní konstanty pro přeměnu A na B, resp. B na C, které se mohou řádově lišit. Systém je popsán rovnicemi (u_1 pro množství A, u_2 pro B atd.)

$$\begin{aligned} u_1'(t) &= -\beta_1 u_1(t) \\ u_2'(t) &= \beta_1 u_1(t) - \beta_2 u_2(t) \\ u_3'(t) &= \beta_2 u_2(t) \end{aligned} \qquad \text{a tedy} \quad A = \begin{pmatrix} -\beta_1 & 0 & 0 \\ \beta_1 & -\beta_2 & 0 \\ 0 & \beta_2 & 0 \end{pmatrix}$$

s vlastními čísly

$$\begin{aligned} \lambda_1 &= -\beta_1 \\ \lambda_2 &= -\beta_2 \\ \lambda_3 &= 0. \end{aligned}$$

Pro $\beta_1 \neq \beta_2$ je řešení

$$u_j(t) = c_{j1} e^{-\beta_1 t} + c_{j2} e^{-\beta_2 t} + c_{j3} \tag{9.15}$$

s konstantami c_{ji} závislými na počátečních podmínkách. Například Eulerova metoda bude stabilní, pokud $k \max(\beta_1, \beta_2) \leq 2$, a podobně pro další metody.

Příklad 9.2.2 (Tlumený harmonický oscilátor)**.** Rovnici

$$y''(t) = -ay(t) - \underbrace{by'(t)}_{\text{tlumení}}$$

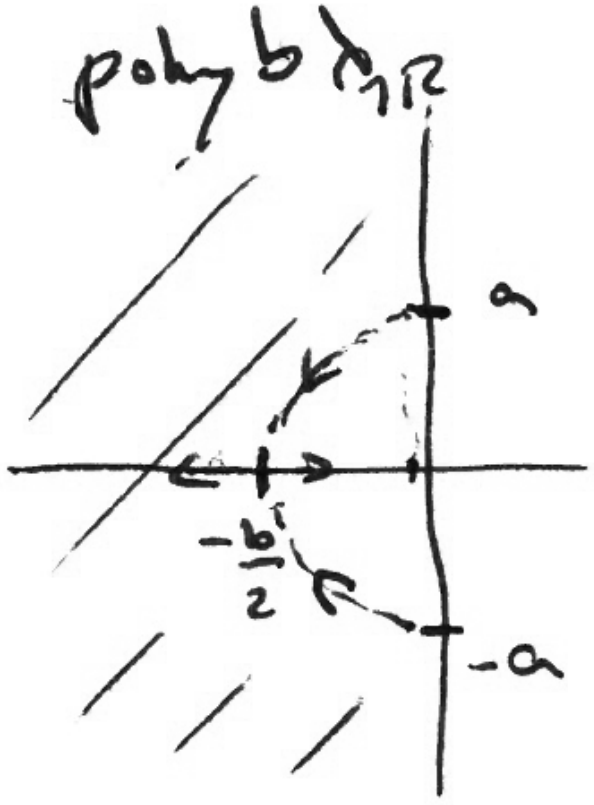
převedeme substitucí

$$\left.\begin{aligned} u_1(t) &= y(t) \\ u_2(t) &= y'(t) \end{aligned}\right\} \Rightarrow u'(t) = Au(t), \text{ kde } A = \begin{pmatrix} 0 & 1 \\ -a & -b \end{pmatrix}.$$

Nyní

$$\lambda_{1,2} = \frac{1}{2}\left(-b \pm \sqrt{b^2 - 4a}\right). \tag{9.16}$$

Pro silné tlumení (velké b) je vhodná Eulerova metoda a podobné metody, ovšem pro slabé tlumení už ne. Pro $b = 0$ dostaneme $\lambda_{1,2} = \pm\sqrt{-a} = \pm i\sqrt{|a|}$, kdy je vhodná midpoint metoda.



Obrázek 9.5: Pohyb vlastních čísel $\lambda_{1,2}$ tlumeného oscilátoru v komplexní rovině: vyšrafovaná levá polorovina $\mathrm{Re}(z) < 0$, na reálné ose je vyznačen bod $-\frac{b}{2}$ se šipkami doleva a doprava, na imaginární ose body a a $-a$, z nichž vedou čárkované šipky k bodu $-\frac{b}{2}$.

Ideální metoda by byla taková, která by měla $S = \{z \in \mathbb{C} : \mathrm{Re}(z) \leq 0\}$; tou je *lichoběžníková* (nebo též Crankova–Nicolsonové) metoda

$$v^{n+1} = v^n + \frac{k}{2}\left[f(v^{n+1}, t_{n+1}) + f(v^n, t_n)\right] \qquad \text{(implicitní metoda 2. řádu)}, \tag{9.17}$$

pro kterou

$$\left.\begin{aligned} \varrho(\xi) &= \xi - 1 \\ \sigma(\xi) &= \frac{\xi + 1}{2} \end{aligned}\right\} \quad \pi_z(\xi) = \xi - 1 - \frac{z}{2}(\xi + 1) = 0 \;\Rightarrow\; |\xi| = \left|\frac{2+z}{2-z}\right| \leq 1,$$

a rovnost nastává pro $z = i\alpha$, $\alpha \in \mathbb{R}$.

9.2.1 Volba kroku

Volba kroku k závisí jednak na tom, jakou požadujeme přesnost (danou globální diskretizační chybou), tj. volíme $k \leq k_{\text{acc}}$, kde k_{acc} odpovídá požadované přesnosti. Jednak závisí na oblasti absolutní stability dané metody, kdy musíme volit $k \leq k_{\text{stab}}$, kde k_{stab} je maximální krok, pro nějž je ještě metoda pro danou úlohu stabilní. Typicky je pro explicitní metody k_{stab} konečné, ale pro implicitní metody může být $k_{\text{stab}} = \infty$.

Pokud $k_{\text{stab}} \ll k_{\text{acc}}$, je vhodné zvolit jinou metodu, protože jinak potřebujeme příliš mnoho kroků pro dosažení určité přesnosti.

9.3 ODR se silným tlumením (stiff ODE)

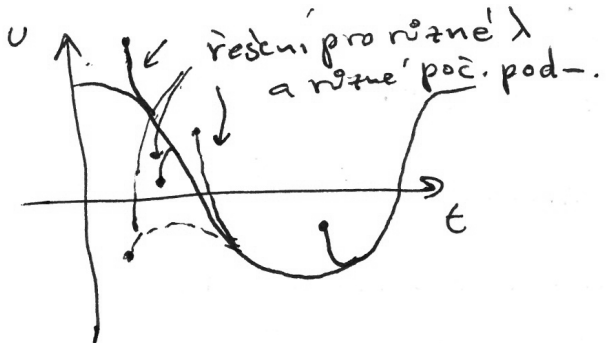
Příklad 9.3.1. Uvažujme ODR

$$u'(t) = \lambda\big(u(t) - \cos t\big) - \sin t, \qquad u(0) = 1. \tag{9.18}$$

Pro zadanou počáteční podmínku je řešením $u(t) = \cos t$ pro libovolné λ. Pro obecnou počáteční podmínku $u(t_0) = u_0$ bude řešení

$$u(t) = e^{\lambda(t-t_0)}(u_0 - \cos t_0) + \cos t. \tag{9.19}$$

Pro $\lambda \ll 0$ nastává velmi rychlé tlumení. I pro $u(0) = 1$, kdy je řešení nezávislé na λ, rozhoduje o stabilitě součin $k\lambda$: i drobná odchylka způsobí nestabilitu a chyba se nekontrolovaně zvětšuje, pokud není k dostatečně malé.



Obrázek 9.6: Graf s osami u (svislá) a t (vodorovná). Silně vytažená křivka je průběh typu kosinu; k ní se z různých počátečních bodů rychle přimykají tenčí křivky, tedy řešení pro různá λ a různé počáteční podmínky.

Charakteristiky silného tlumení jsou následující:

1. velmi *různé časové škály* (velmi rychlé změny v řešení oproti pomalejším);
2. nutnost použít velmi malé k, zvláště u explicitních metod, k dosažení přesnosti kvůli stabilitě, $k < k_{\text{stab}} < k_{\text{acc}}$;
3. nefunkčnost explicitních metod pro rozumné k, takže je nutné používat např. zpětné diference a podobné implicitní metody.

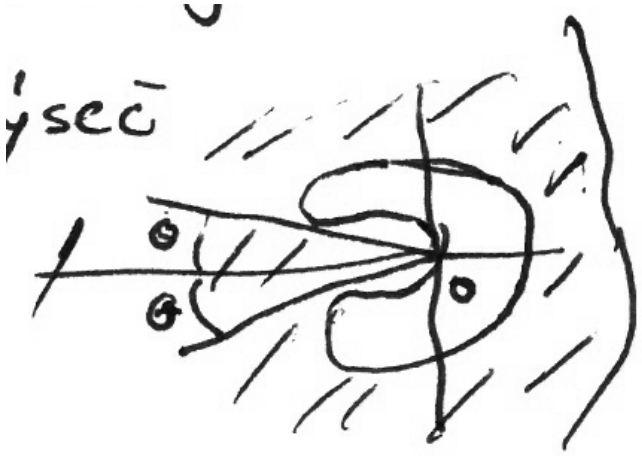
Někdy se pro charakterizaci silného tlumení používá poměr

$$\frac{\max|\lambda_i|}{\min|\lambda_i|}, \tag{9.20}$$

tj. poměr maximálního a minimálního vlastního čísla matice A systému $u'(t) = Au(t)$. Existují ovšem i případy, kdy je tento poměr malý, a přesto jde o problém se silným tlumením. Viz např. Higham, Trefethen: Stiffness of ODEs, BIT 33 (1993) 285, kde je ukázán příklad, kdy A není tzv. normální a je nutné zkoumat pseudospektra místo vlastních čísel A.

Na druhou stranu, jsou-li λ_i blízko imaginární osy a velká, nemusí jít nutně o problém se silným tlumením; spíše jsou řešením rychle oscilující funkce, na něž se používají speciální metody.

Protože jde o důležitý problém a jsou pro něj vhodné metody, které mají oblast stability S takovou, že obsahuje celou polorovinu $\mathrm{Re}(z) \leq 0$, nebo alespoň co největší její část, zavádí se pojem tzv. A-stability (obsahuje-li S celou polorovinu), případně $A(\Theta)$-stability (obsahuje-li S výseč, viz obrázek 9.7). Lze např. ukázat, že A-stabilní lineární vícekroková metoda může mít řád nejvýše 2 (viz lichoběžníková metoda).



Obrázek 9.7: Náčrt komplexní roviny k pojmu $A(\Theta)$-stability: kolem počátku je nakreslena uzavřená nepravidelná křivka, zbytek roviny vlevo je vyšrafován; z počátku vychází doleva klínovitá výseč ohraničená dvěma přímkami s polovičními úhly Θ nad osou a Θ pod osou.

Příklad 9.3.2 (Metody zpětné diference)**.** Pro systém $u'(t) = Au(t)+g(t)$ lze s-krokovou metodu

$$\sum_{j=0}^{s} \alpha_j v^{n+j} = k\,\beta_s\big(Av^{n+s} + g(t_{n+s})\big) \tag{9.21}$$

přepsat (pro $\alpha_s = 1$) na

$$(\mathbb{1} - k\beta_s A)\, v^{n+s} = k\beta_s\, g(t_{n+s}) - \sum_{j=0}^{s-1} \alpha_j v^{n+j}, \tag{9.22}$$

což je systém lineárních rovnic pro neznámý vektor v^{n+s}. Oblast stability těchto metod obsahuje celou polopřímku $z < 0$, avšak pro vyšší řád přesnosti s se stále více zmenšuje úhel Θ, vyšetřujeme-li $A(\Theta)$-stabilitu.

9.4 Adaptivní metody

Potřebujeme-li zajistit určitou přesnost řešení, je někdy nezbytné měnit krok podle toho, jak se chyba chová. K tomu slouží odhad lokální chyby pomocí současného použití dvou metod s různými řády přesnosti. Používají se dvě standardní strategie.

9.4.1 Tentýž postup s krokem h a $2h$

Jednu metodu použijeme s krokem h a $2h$ a rozdíl těchto dvou řešení slouží k odhadu chyby.

Příklad 9.4.1. Uvažujme metodu Runge–Kutta 4. řádu: spočteme y_1 pomocí kroku $2h$ a y_2 pomocí dvou kroků o délce h. Pak pro přesné řešení platí

$$\begin{aligned} u(x+2h) &= y_1 + (2h)^5 C + \mathcal{O}(h^6) \quad \leftarrow \text{krok } 2h \\ u(x+2h) &= y_2 + 2h^5 C + \mathcal{O}(h^6) \quad \leftarrow \text{krok } h, \end{aligned} \tag{9.23}$$

z čehož plyne, že $\Delta \equiv y_2 - y_1$ se bude chovat jako $\mathcal{O}(h^5)$. Pokud chceme dosáhnout určité lokální přesnosti Δ_0, porovnáme obě řešení a případně změníme krok pomocí

$$\tilde{h} = h \left| \frac{\Delta_0}{\Delta} \right|^{0,2}. \tag{9.24}$$

9.4.2 Vnořené Rungeovy–Kuttovy metody

Byly též objeveny tzv. vnořené (embedded) Rungeovy–Kuttovy metody, které se liší pouze vektorem b, tj. koeficienty určujícími v^{n+1} z v^n a z mezikroků, a které jsou 4. a 5. řádu. Stačí nám pak 6 vyčíslení $f(u,t)$ na jeden krok (oproti například 11 u metody Runge–Kutta 4. řádu s kroky h a $2h$). Opět je rozdíl řešení řádu $\mathcal{O}(h^5)$ a úprava kroku se provádí jako výše. Viz např. Cashova–Karpova metoda v Mathematice.

9.4.3 Bulirschova–Stoerova metoda

V současnosti je asi nejefektivnější metodou, pokud potřebujeme vysokou přesnost, *Bulirschova–Stoerova metoda*. Kombinuje midpoint metodu na podintervalech $\frac{h}{n}$, kde se n volí různě, a pak se extrapoluje podobně jako při Richardsonově extrapolaci u derivací či při Rombergově integraci. I zde se využívá toho, že právě pro midpoint metodu se v chybě objevují jen sudé mocniny h^{2k}.

Podrobnosti viz Numerical Recipes, kap. 16, a další knihy o numerickém řešení ODR.

Kapitola 10

Numerická lineární algebra. Úvod. LU rozklad

10.1 Numerická lineární algebra

Mnoho úloh ve fyzice (zvláště pokud je systém popsán parciálními diferenciálními rovnicemi) vede nakonec k numerickému řešení jedné ze tří základních úloh lineární algebry:

1. řešení soustavy lineárních rovnic $Ax = b$;
2. problém hledání vlastních čísel a vektorů

$$Ax = \lambda x, \quad \text{neboli hledáme rozklad } A = V\Lambda V^{-1}; \tag{10.1}$$

3. řešení přeurčené soustavy $Ax = b$ vedoucí na metodu nejmenších čtverců.

Při řešení těchto úloh se s výhodou používají různé faktorizace (rozklady, dekompozice) příslušné matice A na součin dvou až tří speciálních matic, pomocí kterých už pak může být určitá úloha snadno (a rychle) řešitelná.

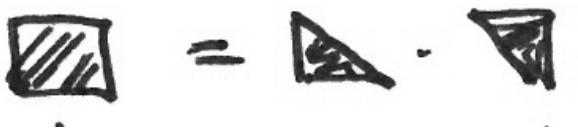
10.1.1 Základní rozklady

Základní rozklady, kterými se budeme podrobněji zabývat, jsou následující tři.

a) LU rozklad (dekompozice), používaný k řešení úlohy 1):

$$A = L \cdot U, \tag{10.2}$$

kde L je dolní trojúhelníková a U je horní trojúhelníková matice.



Obrázek 10.1: Schéma LU rozkladu pomocí tvarů matic: plný šrafovaný čtverec (matice A) se rovná dolnímu trojúhelníku (L) vynásobenému horním trojúhelníkem (U).

b) QR rozklad, používaný pro úlohy 2) a 3), ale lze jej použít i pro 1), jen je dvakrát pomalejší než LU rozklad:

$$A = Q \cdot R, \tag{10.3}$$



Obrázek 10.2: Schéma QR rozkladu pomocí tvarů matic: šrafovaný obdélník (A) se rovná vyššímu úzkému šrafovanému obdélníku (Q) vynásobenému malým čtvercem s vyšrafovanou horní trojúhelníkovou částí (R).

kde Q je ortogonální (obecně $m \times n$) matice a R je horní trojúhelníková matice.

c) SVD — singulární rozklad matice (singular value decomposition):

$$A = U \cdot \Sigma \cdot V^t, \tag{10.4}$$

kde Σ je diagonální matice obsahující tzv. singulární hodnoty a U a V jsou ortogonální (unitární) matice. Tento rozklad lze použít téměř na cokoli, jen je pro praktické výpočty mnohem pomalejší než předchozí rozklady, ovšem je nejstabilnější.



Obrázek 10.3: Schéma SVD pomocí tvarů matic: úzký šrafovaný obdélník (A) se rovná širšímu šrafovanému obdélníku (U) vynásobenému obdélníkem s vyznačenou diagonálou a nulami (kroužky) mimo ni (Σ) a malým šrafovaným čtvercem (V^t).

10.1.2 Rychlost a stabilita

V praktických problémech rozhoduje především **rychlost** (efektivita) určité metody řešení a dále **stabilita** a přesnost; správnost řešení jde většinou ruku v ruce se stabilitou.

Bohužel velmi často jsou stabilnější metody méně efektivní a je nutný určitý kompromis založený na zkušenosti z praxe. Například pro řešení $Ax = b$ je pořadí efektivity

$$LU < QR < SVD, \tag{10.5}$$

ale pokud jde o stabilitu, je to naopak. V praxi se ale ukazuje, že pro většinu systémů $Ax = b$ vzešlých např. z diskretizace PDR je z hlediska přesnosti bezpečné použít LU rozklad.

10.1.3 Doporučení pro praxi

Přestože si ukážeme, jak některé metody fungují a jak je lze využít k řešení problémů 1), 2) a 3), v praxi **velmi doporučuji** použít některou z dostupných knihoven, případně programy jako Mathematica nebo Matlab, kde jsou všechny výše uvedené metody a problémy vyladěné jak z hlediska efektivity, tak z hlediska přesnosti (viz pár odkazů na mé webové stránce).

- Zvláště upozorňuji na knihovnu LAPACK a její implementaci v **Intel MKL knihovně**, kde je provedena optimalizace nejen z hlediska algoritmů, ale i z hlediska hardwaru (využití rychlé paměti na procesorech).

- Též je potřeba uvážit, zda váš konkrétní problém nevede na jeden z řady speciálních typů matic (tridiagonální, pásové, řídké apod.), pro které existují specializované metody, často řádově efektivnější než metody pro obecné matice.
- Někdy (zvláště pro tzv. řídké matice) se s výhodou používají iterační metody (implementované např. v knihovnách PARDISO či JADAMILU) probírané v letním semestru.

10.2 Gaussova eliminace a LU dekompozice (rozklad)

Gaussova eliminace převádí systém $Ax = b$, kde $A \in \mathbb{C}^{m\times m}$, na systém $Ux = \bar{b}$, kde U je horní Δ matice a $\bar{b}$ je modifikované b, pomocí operací eliminujících prvky pod diagonálou.

Postupné řádkové eliminace v jednom sloupci jsou ovšem ekvivalentní násobení matice A dolní Δ maticí:

$$A^{(1)} = L_1 A, \tag{10.6}$$

neboli

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{m1} & \cdots & \cdots & a_{mm} \end{pmatrix}}_{A = A^{(0)}} \xrightarrow{L_1} \underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ 0 & \widetilde{a}_{22} & \cdots & \widetilde{a}_{2m} \\ \vdots & \vdots & & \vdots \\ 0 & \widetilde{a}_{m2} & \cdots & \widetilde{a}_{mm} \end{pmatrix}}_{A^{(1)}}, \text{ kde } L_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -\ell_{21} & 1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \\ \vdots & \vdots & \ddots & 0 \\ -\ell_{m1} & 0 & \cdots 0 & 1 \end{pmatrix}$$

a

$$\ell_{j1} = a_{j1}/a_{11},\ j > 1. \tag{10.7}$$

Obdobně pro eliminaci k-tého sloupce použijeme

$$L_k = \begin{pmatrix} 1 & 0 & \cdots 0 & \cdots & \cdots & 0 \\ 0 & \ddots & & & & \vdots \\ & \ddots & 0 & & & \\ \vdots & & 0 & 1 & \ddots & \\ & & & -\ell_{k+1,k} & \ddots & 0 \\ 0 & \cdots 0 & \cdots & -\ell_{mk} & 0 & 1 \end{pmatrix} \quad \text{kde } \ell_{jk} = \frac{a_{jk}^{(k-1)}}{a_{kk}^{(k-1)}} \quad \text{pro } j > k, \tag{10.8}$$

přičemž sloupec obsahující jedničku a prvky $-\ell_{k+1,k}, \ldots, -\ell_{mk}$ je k-tý sloupec a $a_{jk}^{(k-1)}$ jsou prvky matice $A^{(k-1)} = L_{k-1} \cdots L_1 A$.

Dostáváme nakonec $L_{m-1}L_{m-2}\cdots L_1 A = U$, neboli

$$A = (L_{m-1}\cdots L_1)^{-1}U = L_1^{-1}\cdots L_{m-1}^{-1}U = LU, \tag{10.9}$$

kde U je horní Δ matice a L je dolní Δ matice, neboť

$$L_k^{-1} = \begin{pmatrix} 1 & 0 & \cdots 0 & \cdots & \cdots & 0 \\ 0 & \ddots & & & & \vdots \\ & & 1 & 0 & & \vdots \\ \vdots & & 0 & 1 & \ddots & \\ & & & \ell_{k+1,k} & \ddots & 0 \\ 0 & \cdots & 0 & \ell_{mk} & 0 & 1 \end{pmatrix} \quad \text{a navíc} \quad L_1^{-1}L_2^{-1} = \begin{pmatrix} 1 & & & \\ \ell_{21} & 1 & & 0 \\ \vdots & \ell_{32} & 1 & \\ & & \ddots & \\ \ell_{m1} & \ell_{m2} & 0 & 1 \end{pmatrix} \quad \text{atd.}$$

Jde tedy pouze o změnu znamének v ℓ_{jk}.

Poznámka 10.2.1. Matici L_k lze zapsat pomocí „vnějšího součinu“

$$L_k = I - \ell_k e_k^T, \tag{10.10}$$

kde

$$\ell_k e_k^T = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \ell_{k+1,k} \\ \vdots \\ \ell_{m,k} \end{pmatrix} (0 \dots 0\ 1\ 0 \dots 0)$$

a jednička v řádkovém vektoru je na k-tém místě. Protože skalární součin $e_k^T \ell_k = 0$, bude

$$(I - \ell_k e_k^T)(I + \ell_k e_k^T) = I,$$

a tedy $L_k^{-1} = I + \ell_k e_k^T$. Dále platí

$$L_k^{-1} L_{k+1}^{-1} = (I + \ell_k e_k^T)(I + \ell_{k+1} e_{k+1}^T) = I + \ell_k e_k^T + \ell_{k+1} e_{k+1}^T$$

atd.

Poznámka 10.2.2. Protože má L na diagonále jedničky, lze L a U uložit do jediné matice; většinou se nahrazuje přímo matice A jejím LU rozkladem.

Poznámka 10.2.3. Takto přímočaře nelze postup použít, neboť je nestabilní; nutná je tzv. pivotizace.

Příklad 10.2.4 (Nestabilita Gaussovy eliminace bez pivotizace)**.** Řešme $Ax = b$ pro

$$A = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Přímo:

$$\begin{aligned} 10^{-20} x_1 + x_2 &= 1\,, \\ x_1 + x_2 &= 0 \end{aligned} \qquad \Rightarrow \qquad x_1 = \frac{1}{10^{-20} - 1} \doteq -1\,, \quad x_2 = -x_1 \doteq 1.$$

LU dekompozice dává

$$L = \begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{pmatrix} \quad \Rightarrow \quad L \cdot U = A = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix}.$$

Ovšem v double precision:

$$\tilde{L} = \begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix} \qquad \tilde{U} = \begin{pmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{pmatrix} \quad \Rightarrow \quad \tilde{L}\tilde{U} = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 0 \end{pmatrix} \neq A.$$

Pro některé pravé strany (např. $b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$) to stále dá správné řešení, ale ne vždy! Právě pro $b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ dostaneme

$$\tilde{L}^{-1} = \begin{pmatrix} 1 & 0 \\ -10^{20} & 1 \end{pmatrix} \qquad \tilde{U}^{-1} = \begin{pmatrix} 10^{20} & 1 \\ 0 & -10^{-20} \end{pmatrix} \quad \Rightarrow$$

$$\Rightarrow \quad \tilde{x} = \tilde{U}^{-1}\tilde{L}^{-1} b = \tilde{U}^{-1} \begin{pmatrix} 1 \\ -10^{20} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \neq \begin{pmatrix} -1 \\ 1 \end{pmatrix},$$

neboť Gaussova eliminace dělá něco jiného než dosazení za $x_2 = -x_1$:

$$\begin{array}{l|l} - \cdot 10^{20} & 10^{-20}x_1 + x_2 = 1 \\ & x_1 + x_2 = 0 \\ \hline & 10^{-20}x_1 + x_2 = 1 \\ & (1 - 10^{20})x_2 = -10^{20} \quad \Rightarrow \end{array}$$

Po zaokrouhlení odtud plyne

$$-10^{20}x_2 = -10^{20} \quad \Rightarrow \quad x_2 = 1 \quad !$$

a tedy $10^{-20}x_1 + 1 = 1 \quad \Rightarrow \quad x_1 = 0 \quad !$

Ovšem **s pivotizací** bude

$$\begin{array}{l|l} \cdot 10^{-20} & x_1 + x_2 = 0 \\ & 10^{-20}x_1 + x_2 = 1 \\ \hline & x_1 + x_2 = 0 \\ & 0 + (1 - 10^{-20})x_2 = 1 \quad \Rightarrow \end{array}$$

a po zaokrouhlení

$$1x_2 = 1 \quad \Rightarrow \quad x_2 = 1,$$

a tedy nyní $x_1 = -x_2 = -1$.

10.2.1 Pivotizace

Pivotizace je výběr největšího prvku, kterým dělíme. Nejen tehdy, pokud se na diagonále objeví nula, ale i tehdy, když je tam velmi malé číslo relativně k ostatním, je nutné kvůli stabilitě přehodit řádky (a případně i sloupce) a pak pokračovat v Gaussově eliminaci.

Rozlišujeme dvě varianty:

- **řádková pivotizace** — hledáme největší $a_{jk}^{(k-1)}$ pro $j \geq k$ a přehazujeme řádky;
- **plná pivotizace** — hledáme největší $a_{j\ell}^{(k-1)}$ pro $j \geq k$ a $\ell \geq k$ a přehazujeme nejen řádky, ale i sloupce.

Plná pivotizace ovšem potřebuje $O(m^3)$ operací porovnání oproti $O(m^2)$ operacím v řádkové pivotizaci, a proto se používá pivotizace řádková.

Poznámka 10.2.5. Zde $O(m^3)$ je celkový počet operací v průběhu celé Gaussovy eliminace.

10.2.2 Řádková pivotizace a LU rozklad

Prohazování řádků lze provést pomocí permutačních matic, např. matice

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

přehodí 2. a 4. řádek matice 4×4.

Lze tedy psát

$$L_{m-1}P_{m-1}L_{m-2}P_{m-2}\ldots L_1P_1A = U \tag{10.11}$$

s příslušnými permutačními maticemi P_k pro řádkovou pivotizaci v k-tém kroku. Toto lze ovšem přepsat pomocí

$$L'_k = P_{m-1}\ldots P_{k+1}L_kP_{k+1}^{-1}\ldots P_{m-1}^{-1} \quad , \quad L'_{m-1} = L_{m-1} \tag{10.12}$$

jako

$$\underbrace{\left(L'_{m-1}L'_{m-2}\dots L'_1\right)}_{L^{-1}}\underbrace{\left(P_{m-1}P_{m-2}\dots P_1\right)}_{P}A = U, \tag{10.13}$$

neboli $PA = LU$. Kromě matic L a U je tedy nutné ještě uložit celkovou permutační matici P (lze ji uložit jako vektor).

Matice L'_k má oproti L_k pouze přeuspořádané prvky ℓ_{jk}, neboť $P_k^{-1} = P_k$ a P_k násobící A zprava přehazuje místo řádků příslušné sloupce; matici L_k přitom násobíme pouze maticemi P_j pro $j > k$, takže se přehodí sloupce jednotkové podmatice a pak to přehození řádků vrátí.

Poznámka 10.2.6. Obdobně lze přepsat i úplnou pivotizaci, a to jako $PAQ = LU$, kde P i Q jsou permutační matice. V praxi se však kvůli efektivitě nepoužívá.

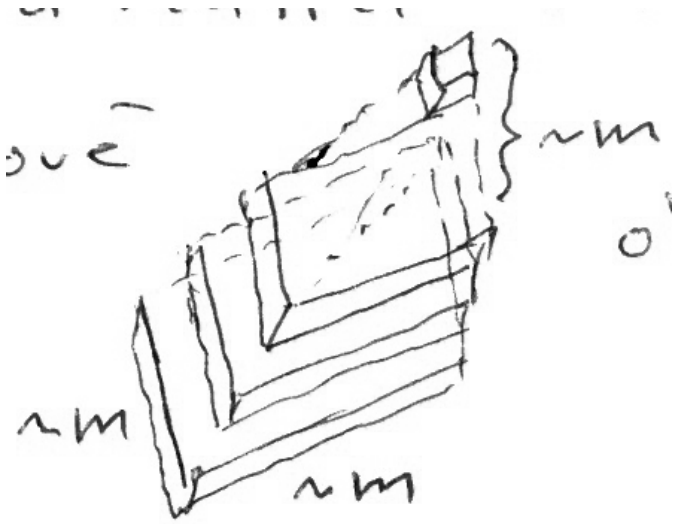
10.2.3 Algoritmus LU rozkladu s řádkovou pivotizací

$U = A$, $L = I$, $P = I$ (inicializace)
pro $k = 1, m-1$ (eliminujeme $m-1$ sloupců)
  najdi $i \geq k$, pro které je $|U_{ik}|$ maximální (pivotizace)
  $U_{k,k:m} \longleftrightarrow U_{i,k:m}$
  $\ell_{k,1:k-1} \longleftrightarrow \ell_{i,1:k-1}$
  $P_{k,:} \longleftrightarrow P_{i,:}$
  pro $j = k+1, m$
    $\ell_{jk} = U_{jk}/U_{kk}$
    $U_{j,k:m} = U_{j,k:m} - \ell_{jk}U_{k,k:m}$

Tři řádky s prohozením představují přehození řádků, pokud $i \neq k$. Poslední dva řádky jsou odečtení dvou řádků, tedy časově nejnáročnější část algoritmu.

10.2.4 Počet operací

Operace $+, -, *, /$ uvažujeme na stejné úrovni. Když eliminujeme jeden (k-tý) sloupec, musíme upravit podmatici $(m-k) \times (m-k)$, a sloupců je $m-1$; obrazově tedy dostáváme situaci na obrázku 10.4.



Obrázek 10.4: Perspektivní kresba stupňovité „pyramidy“ složené z vrstev postupně se zmenšujících čtvercových podmatic; hrany základny i výška jsou popsány $\sim m$, $\sim m$, $\sim m$.

Objem této pyramidy je $\sim \frac{1}{3}m^3$, ovšem vždy musíme dělat dvě operace $(-, *)$, viz poslední řádek algoritmu. Takže celkem je potřeba

$$\sim \tfrac{2}{3}m^3 \tag{10.14}$$

operací (tzv. flops, tedy floating point operations). Řádková pivotizace je $\sim m^2$.

Zápis $\sim \frac{2}{3}m^3$ znamená, že

$$\lim_{m\to\infty} \frac{\text{počet operací}}{\frac{2}{3}m^3} = 1.$$

10.2.5 Řešení soustavy $Ax = b$ pomocí LU rozkladu

Jakmile máme rozklad $PA = LU$, můžeme řešit ekvivalentní soustavu $PAx = Pb$ (tj. přeuspořádáme pravou stranu) a řešíme $Ly = Pb$ a pak $Ux = y$ přímo, neboť jde o trojúhelníkové matice (tzv. zpětná substituce).

Obě rovnice mají náročnost $\sim m^2$, proto se vyplatí udělat LU rozklad zvláště tehdy, pokud se matice A nemění, ale máme různé pravé strany.

10.3 Choleského rozklad

Jde o obdobu LU rozkladu, ale pro hermitovské, pozitivně definitní matice A, tj. $A = A^\dagger$ a $x^\dagger Ax > 0$ pro libovolné $x \neq 0$. Z pozitivní definitnosti např. plyne, že všechna vlastní čísla jsou kladná, a též že jsou kladné diagonální prvky.

Díky hermitovosti můžeme při Gaussově eliminaci nejen nulovat první sloupec násobením dolní Δ maticí zleva, ale také první řádek násobením hermitovsky sdruženou maticí zprava. Matici A lze tedy (pro $a_{11} > 0$) rozložit takto:

$$A = \begin{pmatrix} a_{11} & w^\dagger \\ w & K \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ \frac{w}{\alpha} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & K - \frac{ww^\dagger}{\alpha^2} \end{pmatrix} \begin{pmatrix} \alpha & w^\dagger/\alpha \\ 0 & 1 \end{pmatrix} = R_1^\dagger A_1 R_1, \quad \text{kde } \alpha = \sqrt{a_{11}}, \tag{10.15}$$

přičemž $K - \frac{ww^\dagger}{\alpha^2}$ je opět pozitivně definitní hermitovská matice.

Matice $A_1 = (R_1^\dagger)^{-1} A R_1^{-1}$ je též pozitivně definitní hermitovská matice. Navíc platí, že libovolná podmatice obsahující levý horní nebo pravý dolní roh takovéto matice je též hermitovská a pozitivně definitní: násobením $X^\dagger A_1 X$, kde

$$X = \begin{pmatrix} 0 \\ \mathbf{1}_{n\times n} \end{pmatrix},$$

takovou podmatici vybereme.

Takto můžeme pokračovat dále rozklady podmatic, až zůstane uprostřed jednotková matice, tj. máme

$$A = R_1^\dagger R_2^\dagger \cdots R_m^\dagger R_m \cdots R_2 R_1 = R^\dagger R, \quad \text{kde } r_{jj} > 0 \tag{10.16}$$

a R je horní Δ matice. Tento rozklad je jednoznačný.

Díky hermitovosti matice A stačí uložit jen její horní trojúhelníkovou část a upravovat ji na R:

$$\begin{aligned} &R = A \\ &\text{pro } k = 1, m \\ &\quad \text{pro } j = k+1, m \\ &\qquad R_{j,j:m} = R_{j,j:m} - R_{k,j:m}\, R^*_{kj}/R_{kk} \\ &\quad R_{k,k:m} = R_{k,k:m}/\sqrt{R_{kk}} \end{aligned}$$

Výraz R^*_{kj}/R_{kk} se vyplatí uložit do pomocné proměnné, pak jsou potřeba jen dvě operace; nyní je ovšem „objem“ pouze $\frac{1}{6}m^3$, takže celkem je potřeba $\sim \frac{1}{3}m^3$ operací.

10.4 Stabilita Gaussovy eliminace a LU rozkladu

Bez pivotizace lze ukázat, že spočtené $\tilde{L}$ a $\tilde{U}$ splňují

$$\tilde{L}\tilde{U} = A + \delta A, \quad \text{kde ovšem } \frac{\|\delta A\|}{\|L\|\,\|U\|} = \mathcal{O}(\varepsilon_M), \tag{10.17}$$

pro určité δA. Ve jmenovateli tedy není $\|A\|$, ale záleží na velikosti L a U. Pokud $\|A\| \ll \|L\|$ nebo $\|A\| \ll \|U\|$, pak může být $\|\delta A\|$ poměrně velké a spočtené $\tilde{L}\tilde{U}$ se může dost lišit od původní matice A.

Říkáme, že **LU rozklad není zpětně stabilní**; důsledkem je, že řešení $\tilde{L}\tilde{U}x = b$ pak může být velmi nepřesné.

Pro **LU rozklad s pivotizací** lze ukázat, že je zpětně stabilní, tj.

$$\tilde{L}\tilde{U} = \tilde{P}A + \delta A, \quad \text{kde } \frac{\|\delta A\|}{\|A\|} = \mathcal{O}(\rho\,\varepsilon_M), \tag{10.18}$$

přičemž kvůli zaokrouhlovacím chybám nemusí jít u $\tilde{P}$ nutně o matici P.

Nyní již $\|\delta A\|$ porovnáváme s $\|A\|$, ale objevuje se tzv. **růstový faktor** (growth factor)

$$\rho = \frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|}, \tag{10.19}$$

který může být ve výjimečných případech až 2^m (viz ukázka v Mathematice), ale v praxi se s tímto případem s velkou pravděpodobností nesetkáte; u náhodně generovaných matic je spíše řádu $\sqrt{m}$.

Poznámka 10.4.1. **Algoritmus** $\tilde{f}$ je pro problém $f = f(x)$ **zpětně stabilní**, pokud pro $\forall x$ platí $\tilde{f}(x) = f(\tilde{x})$ pro určité $\tilde{x}$ splňující

$$\frac{\|\tilde{x} - x\|}{\|x\|} = \mathcal{O}(\varepsilon_M),$$

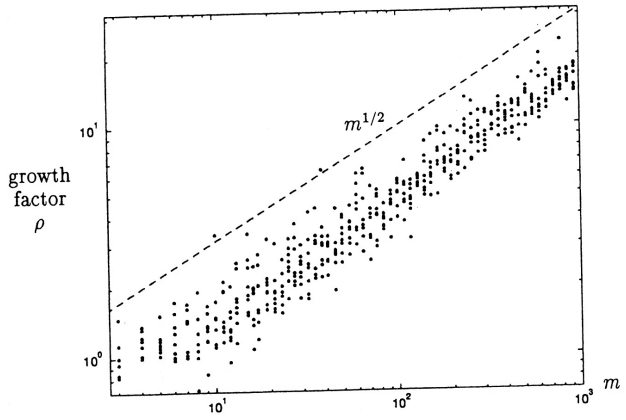
neboli „Zpětně stabilní algoritmus dá správnou odpověď na skoro správnou otázku.“

Navíc pro **přesnost zpětně stabilního algoritmu** platí (viz např. Trefethen, Bau III: Numerical linear algebra)

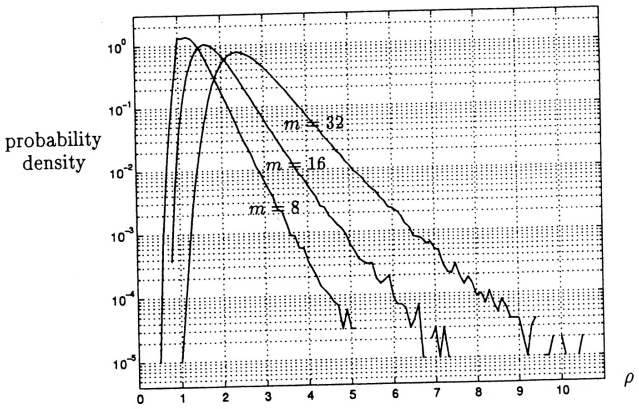
$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = \mathcal{O}(\varkappa(x)\,\varepsilon_M), \tag{10.20}$$

kde $\varkappa(x)$ je číslo podmíněnosti, pro matice $\varkappa(A) = \|A\|\,\|A^{-1}\|$.

Poznámka 10.4.2. Následující dva obrázky jsou převzaty ze strany 168 uvedené knihy (část IV, Systems of Equations); jejich popisky jsou ponechány v původním anglickém znění.



Obrázek 10.5: *Figure 22.1. Growth factors for Gaussian elimination with partial pivoting applied to 496 random matrices (independent, normally distributed entries) of various dimensions. The typical size of ρ is of order $m^{1/2}$, much less than the maximal possible value 2^{m-1}.* — bodový graf (log-log) růstového faktoru v závislosti na rozměru matice; mrak bodů stoupá zleva doprava, nad ním čárkovaná přímka označená $m^{1/2}$. Svislá osa „growth factor ρ“ s ryskami 10^0, 10^1; vodorovná osa „m“ s ryskami 10^1, 10^2, 10^3.



Obrázek 10.6: *Figure 22.2. Probability density distributions for growth factors of random matrices of dimensions $m = 8$, 16, 32, based on sample sizes of one million for each dimension. The density appears to decrease exponentially with ρ. The chatter near the end of each curve is an artifact of the finite sample sizes.* — semilogaritmický graf s tečkovanou mřížkou, tři křivky hustoty pravděpodobnosti s ostrým náběhem a exponenciálním poklesem, na koncích rozkmitané. Svislá osa „probability density“ s ryskami 10^0 až 10^{-5}; vodorovná osa „ρ“ s ryskami 0 až 10; křivky označené $m = 32$, $m = 16$, $m = 8$.

Kapitola 11

Numerická lineární algebra. QR a SVD rozklad

11.1 QR rozklad (faktorizace)

Základní myšlenka QR rozkladu je následující: sloupce matice A jsou vektory, jejichž lineární obal tvoří obor hodnot $R(A)$ zobrazení Ax, volíme-li x libovolně. Neboli

$$(a_1 \,|\, a_2 \,|\, \ldots \,|\, a_n) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = x_1 \begin{pmatrix} a_1 \end{pmatrix} + x_2 \begin{pmatrix} a_2 \end{pmatrix} + \cdots + x_n \begin{pmatrix} a_n \end{pmatrix}, \tag{11.1}$$

kde A je matice $m \times n$, $m \geq n$.

Vektory $a_1, \ldots, a_n$ dávají postupně podprostory (lineární obaly)

$$\mathcal{L}(a_1)\,,\ \mathcal{L}(a_1, a_2)\,, \ldots \qquad \mathcal{L}(a_1, \ldots, a_n) = R(A),$$

přičemž v nich můžeme postupně konstruovat ortonormální bázi $q_1, \ldots, q_n$ (za předpokladu úplné hodnosti matice A) takovou, že $\mathcal{L}(a_1) = \mathcal{L}(q_1)$, $\mathcal{L}(a_1, a_2) = \mathcal{L}(q_1, q_2)$ atd.

Vektory a_i pak můžeme vyjádřit v bázi q_i například takto:

$$\begin{aligned} a_1 &= r_{11} q_1 \\ a_2 &= r_{12} q_1 + r_{22} q_2 \\ &\vdots \\ a_n &= r_{1n} q_1 + \cdots + r_{nn} q_n \end{aligned} \qquad (\Longleftrightarrow) \qquad (a_1 \,|\, \ldots \,|\, a_n) = (q_1 \,|\, \ldots \,|\, q_n) \begin{pmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ 0 & r_{22} & \ldots & r_{2n} \\ \vdots & & \ddots & \\ 0 & \ldots & 0 & r_{nn} \end{pmatrix}, \tag{11.2}$$

tj.

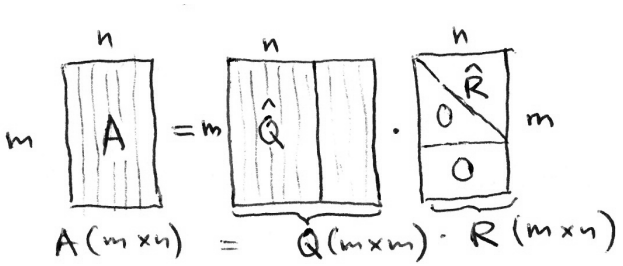
$$A = \hat{Q} \cdot \hat{R}, \tag{11.3}$$

kde $\hat{Q}$ je obecně část (pro $m > n$) ortonormální matice Q splňující $Q^\dagger Q = Q Q^\dagger = \mathbf{1}$ a $\hat{R}$ je horní trojúhelníková matice.

Tomuto rozkladu se říká *redukovaný QR rozklad* (reduced). Pokud $\hat{Q}$ doplníme na Q libovolnými vektory ortogonálními na $q_1, \ldots, q_n$ a $\hat{R}$ doplníme řádky s nulami, dostaneme *plný QR rozklad*.

11.1.1 Existence a jednoznačnost

Lze ukázat, že každá matice $A \in \mathbb{C}^{m \times n}$, $m \geq n$, má QR rozklad, a tedy i rozklad $\hat{Q}\hat{R}$; plyne to z přímé konstrukce pomocí Gramovy–Schmidtovy ortogonalizace.



Obrázek 11.1: Schéma plného QR rozkladu pomocí blokových obdélníků. Vlevo svisle šrafovaný obdélník s popiskem A, nad ním n, vlevo od něj m; následuje „$= m$“ a širší svisle šrafovaný čtverec rozdělený svislou čárou, v levé části popisek $\hat{Q}$, nad ním n; poté tečka násobení a vysoký obdélník rozdělený vodorovnou čárou – v horní části diagonála oddělující 0 (vlevo dole) a $\hat{R}$ (vpravo nahoře), v dolní části 0; nad ním n, vpravo od něj m. Pod druhým a třetím blokem jsou svorky. Pod obrázkem: $A(m \times n) = Q(m \times m) \cdot R(m \times n)$.

Je-li navíc hodnost $A = n$ (tedy maximální), pak existuje právě jeden rozklad $\hat{Q} \cdot \hat{R}$ s $r_{jj} > 0$, tj. s r_{jj} reálným a kladným, čímž fixujeme „fázi“ vektoru q_j.

Pokud je hodnost $A < n$, pak pro určité j budou vektory $q_1, \ldots, q_j$ lineárně závislé, pak bude $r_{jj} = 0$ a jako q_j volíme libovolný normovaný vektor kolmý na $q_1, \ldots, q_{j-1}$.

11.1.2 Použití na řešení soustavy $Ax = b$

Máme-li matici A typu $(m \times m)$ a její rozklad $Q \cdot R$, pak řešení x dostaneme přímo zpětnou substitucí pro

$$Rx = Q^\dagger b, \tag{11.4}$$

neboť $Q^\dagger$ je inverzní k Q.

Poznámka 11.1.1. U některých metod (např. u Householderovy triangularizace) se Q přímo nekonstruuje – z A dostáváme rovnou R – a v průběhu výpočtu R se počítá i $Q^\dagger b$.

11.1.3 Výpočet QR rozkladu

K výpočtu QR rozkladu se používá několik metod; u každé uvádíme i počet operací.

1) Přímá Gramova–Schmidtova ortogonalizace, $\sim 2mn^2$ operací. Není obecně stabilní, ztrácí se ortogonalita kvůli zaokrouhlovacím chybám.

2) Modifikovaná Gramova–Schmidtova ortogonalizace, $\sim 2mn^2$ operací. Je o něco lepší, ale také ne zcela přesná.

3) **Householderova triangularizace**, $\sim 2mn^2 - \frac{2}{3}n^3$ operací. Jde o nejlepší metodu, je zpětně stabilní.

4) Givensovy rotace, $\sim 3mn^2$ operací. Jsou o něco pomalejší, ale mají své uplatnění např. v iteračních metodách (viz např. metoda GMRES v letním semestru).

11.1.4 Přímá konstrukce pomocí Gramovy–Schmidtovy ortogonalizace

Z matice A typu $m \times n$ dává tato metoda redukovaný QR rozklad. Ortogonalizujeme postupně a_2 na a_1, pak a_3 na a_1 a a_2 atd., a zároveň normalizujeme. Předpokládáme přitom plnou hodnost A; jinak je nutno testovat, zda $r_{jj} = 0$, a případně volit q_j jinak.

Algoritmus je následující:

$$\begin{aligned}
&\text{pro } j = 1, \ldots, n \\
&\quad q_j = a_j \\
&\quad \text{pro } i = 1, \ldots, j-1 \\
&\qquad r_{ij} = q_i^\dagger a_j \\
&\qquad q_j = q_j - r_{ij} q_i \\
&\quad r_{jj} = \|q_j\|_2 \\
&\quad q_j = q_j / r_{jj}
\end{aligned}$$

Dvojice řádků $r_{ij} = q_i^\dagger a_j$ a $q_j = q_j - r_{ij} q_i$ provádí ortogonalizaci: pokud je a_j blízko q_i, jde o odečítání dvou blízkých čísel (vektorů), a tedy $\Rightarrow$ ztráta přesnosti a ortogonality. Poslední dva řádky, $r_{jj} = \|q_j\|_2$ a $q_j = q_j / r_{jj}$, provádějí normalizaci.

Vnitřní část algoritmu představuje $\sim 4m$ operací a opakuje se $\sim \frac{n^2}{2}$-krát, celkem tedy $O(2mn^2)$ operací.

11.1.5 Modifikovaná Gramova–Schmidtova metoda

Ve standardní Gramově–Schmidtově metodě jsou jednotlivá q_j dána jako $P_j a_j$, tj. projektorem

$$P_j = I - q_1 q_1^\dagger - \ldots - q_{j-1} q_{j-1}^\dagger \tag{11.5}$$

na podprostor kolmý na $\mathcal{L}(q_1, \ldots, q_{j-1}) = \mathcal{L}(a_1, \ldots, a_{j-1})$ (alespoň v přesné aritmetice).

Díky ortogonalitě vektorů q_i lze tento projektor napsat jako součin projektorů

$$P_j = P_{\perp q_{j-1}} P_{\perp q_{j-2}} \ldots P_{\perp q_2} P_{\perp q_1} \quad , \text{ kde } P_{\perp q_i} = I - q_i q_i^\dagger. \tag{11.6}$$

Takto je to numericky výhodnější: P_j má hodnost $m-(j-1)$, kdežto $P_{\perp q_i}$ mají stále hodnost $m-1$.

Algoritmus modifikované Gramovy–Schmidtovy metody vypadá takto:

$$\begin{aligned}
&\text{pro } i = 1, \ldots, n \\
&\quad q_i = a_i \qquad\qquad \text{(inicializace všech } q_i\text{)} \\
&\text{pro } i = 1, \ldots, n \\
&\quad r_{ii} = \|q_i\|_2 \\
&\quad q_i = q_i / r_{ii} \\
&\quad \text{pro } j = i+1, \ldots, n \\
&\qquad r_{ij} = q_i^\dagger q_j \\
&\qquad q_j = q_j - r_{ij} q_i
\end{aligned}$$

Dvojice řádků $r_{ii} = \|q_i\|_2$ a $q_i = q_i / r_{ii}$ provádí normalizaci (opět neošetřujeme, zda $r_{ii} = 0$); jde přitom opět o $\sim 4m$ operací. Poslední dva řádky, $r_{ij} = q_i^\dagger q_j$ a $q_j = q_j - r_{ij} q_i$, představují postupnou ortogonalizaci všech vektorů q_j najednou – i zde dochází ke ztrátě přesnosti, ale menší. Celkem je i zde potřeba $O(2mn^2)$ operací.

11.1.6 Trojúhelníková versus ortogonální triangularizace

Poznámka 11.1.2. Gramova–Schmidtova ortogonalizace je vlastně *trojúhelníková ortogonalizace*, neboť ji jde přepsat jako

$$AR_1 R_2 \ldots R_n = \hat{Q}, \tag{11.7}$$

kde

$$R_1 = \begin{pmatrix} \frac{1}{r_{11}} & -\frac{r_{12}}{r_{11}} & \cdots & -\frac{r_{1n}}{r_{11}} & \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & 0 \\ 0 & \cdots & 0 & & 1 \end{pmatrix}$$

a obecně

$$R_i = \begin{pmatrix} 1 & & & 0 & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \frac{1}{r_{ii}} & -\frac{r_{i\,i+1}}{r_{ii}} & \cdots & -\frac{r_{in}}{r_{ii}} \\ & 0 & & 1 & & 0 \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix},$$

přičemž $\hat{R} = R_n^{-1} R_{n-1}^{-1} \dots R_1^{-1}$.

Myšlenka Householderovy a Givensovy metody je opačná: jde v nich o *ortogonální triangularizaci*.

11.1.7 Householderova triangularizace

Nyní budeme převádět A na horní $\triangle$ tvar pomocí ortogonálních matic $Q_1, \dots, Q_n$, tj.

$$\underbrace{Q_n \dots Q_1}_{Q^\dagger} A = R, \tag{11.8}$$

a dostaneme tak *plný QR rozklad*.

Matice $Q_1, \dots, Q_n$ musí být takové, aby nulovaly prvky pod diagonálou, tj.

$$A \xrightarrow{Q_1} \underbrace{\begin{pmatrix} x & x & x & \cdots & x \\ 0 & x & & & \vdots \\ \vdots & x & & & \\ \vdots & \vdots & & & \vdots \\ 0 & x & \cdots & & x \end{pmatrix}}_{Q_1 A} \xrightarrow{Q_2} \underbrace{\begin{pmatrix} x & x & x & \cdots & x \\ 0 & x & x & & \vdots \\ 0 & 0 & x & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & x & \cdots & x \end{pmatrix}}_{Q_2 Q_1 A} \quad \text{atd.}$$

kde x zde značí obecně nenulovou hodnotu.

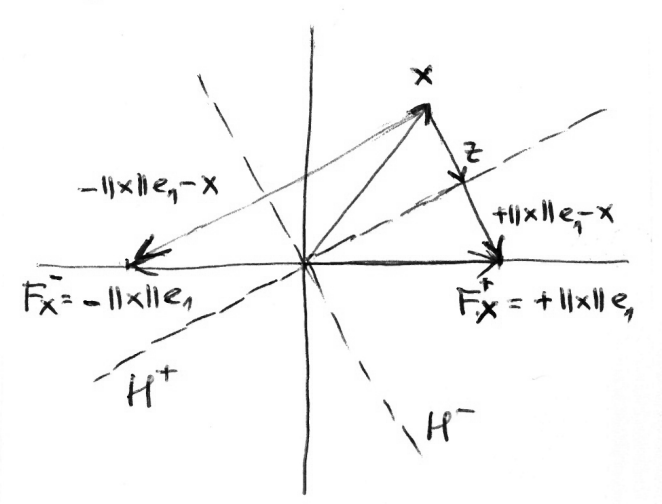
To lze provést např. *Householderovým zrcadlením*

$$Q_k = \begin{pmatrix} I & 0 \\ 0 & F \end{pmatrix} \begin{matrix} \} \, k-1 \text{ řádků} \\ \} \, m-(k-1) \text{ řádků} \end{matrix} \tag{11.9}$$

(první blok tvoří $k-1$ řádků, druhý blok $m-(k-1)$ řádků), kde F je takové, aby z vektoru tvořeného prvky od diagonály dolů v k-tém sloupci udělalo násobek jednotkového vektoru e_1:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} \xrightarrow{F} \begin{pmatrix} \|x\|_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \|x\|_2 e_1. \tag{11.10}$$

Matice F tedy musí provést zrcadlení v „nadrovině“ $H^+ \perp v = \|x\| e_1 - x$, případně v „nadrovině“ $H^- \perp v = -\|x\| e_1 - x$, pokud je x příliš blízko $\|x\| e_1$.



Obrázek 11.2: Householderovo zrcadlení v rovině – vodorovná a svislá osa, dvě čárkovaně vyznačené nadroviny H^+ a H^- procházející počátkem. Z počátku vede vektor x šikmo vzhůru doprava; z jeho hrotu vede šipka doprava dolů k bodu na vodorovné ose označenému $F^+x = +\|x\|e_1$, tato šipka je popsána $+\|x\|e_1 - x$ a je na ní vyznačen bod z. Druhá šipka vede z hrotu x doleva k bodu na vodorovné ose označenému $F^-x = -\|x\|e_1$ a je popsána $-\|x\|e_1 - x$.

Projekce do H^+ je dána projektorem

$$Px = \left(I - \frac{vv^\dagger}{v^\dagger v}\right) x = x - z,$$

ovšem my potřebujeme, aby $Fx = x - 2z$, neboli

$$Fx = \left(I - 2\,\frac{vv^\dagger}{v^\dagger v}\right) x. \tag{11.11}$$

Vektor v se obvykle volí podle znaménka x_1, tj. $v = -\mathrm{sign}(x_1)\|x\|e_1 - x$. Protože však F na znaménku vektoru v nezávisí, lze vzít jednoduše

$$v = \mathrm{sign}(x_1)\|x\|e_1 + x \qquad (\text{a např. } \mathrm{sign}(0) = 1). \tag{11.12}$$

Algoritmus pak upravuje A na R takto:

pro $k = 1, \ldots, n$

$x = A_{k:m,k}$	(proměnlivá délka x!)
$v_k = \mathrm{sign}(x_1)\|x\|_2 e_1 + x$	(volba v)
$v_k = v_k / \|v_k\|_2$	(normalizace v)
$A_{k:m,k:n} = A_{k:m,k:n} - 2v_k \left(v_k^\dagger A_{k:m,k:n}\right)$	

Dvojku v posledním řádku lze přitom „schovat“ do v_k.

Poslední řádek je nejnáročnější částí algoritmu: upravuje se $\sim (m-k)\cdot(n-k)$ prvků, vždy 4 operace. Celkový počet upravovaných prvků odpovídá objemu klínovitého tělesa na obrázku 11.3, tedy $\frac{1}{2}mn^2 - \frac{1}{6}n^3$, což krát 4 operace dává celkem $\sim 2mn^2 - \frac{2}{3}n^3$ operací.

Matice Q se často nekonstruuje, ale počítá se buď $Q^\dagger b$ (kde $Q^\dagger = Q_n \ldots Q_1$)

$$\begin{aligned} Q^\dagger b: \quad & \text{pro } k = 1, \ldots n \\ & b_{k:m} = b_{k:m} - 2v_k \left(v_k^\dagger b_{k:m}\right) \end{aligned}$$



Obrázek 11.3: Perspektivní náčrt klínovitého tělesa (zkosený hranol) znázorňujícího objem upravované části matice. Popisky: vlevo dole m, vpravo nahoře u horní hrany $m-n$, vpravo n, dole n.

nebo Qb (neboť $Q = Q_1 \dots Q_n$, jelikož $Q_i^\dagger = Q_i$)

$$\begin{aligned} Qb: \quad & \text{pro } k = n, \dots, 1 \\ & b_{k:m} = b_{k:m} - 2v_k \left(v_k^\dagger b_{k:m} \right) \end{aligned}$$

Zde je nutné uchovat vektory v_k! Obě tyto úpravy $b_{k:m}$ představují už jen $\sim 4mn$ operací.

Pokud potřebujeme přímo matici Q, působíme takto na jednotkovou matici, tj. na vektory e_j.

11.1.8 Givensovy rotace

Givensovy rotace používají místo zrcadlení otočení, ale nebudeme je zde podrobněji probírat.

11.1.9 Stabilita Householderovy triangularizace

Lze ukázat, že *Householderova triangularizace je zpětně stabilní*, tj. pro spočtené $\tilde{Q}$ a $\tilde{R}$ platí pro určité $\delta A \in \mathbb{C}^{m \times n}$

$$\tilde{Q} \cdot \tilde{R} = A + \delta A \ , \qquad \frac{\|\delta A\|}{\|A\|} = \mathcal{O}(\varepsilon_M). \tag{11.13}$$

Též řešení soustavy $Ax = b$ je stabilní, pokud ji řešíme pomocí $Q \cdot R = A$ a následně $y = Q^\dagger b$ a $x = R^{-1} y$.

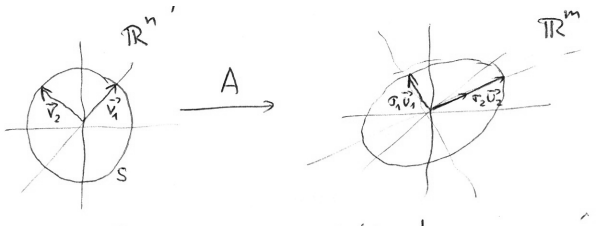
Pozor ovšem na podmíněnost úlohy: obecně budou jak $\tilde{Q}$ a $\tilde{R}$, tak $\tilde{x}$ mít chybu $\mathcal{O}(\kappa(A)\varepsilon_M)$, kde

$$\kappa(A) = \|A\|\|A^{-1}\| = \frac{\sigma_1}{\sigma_m}, \tag{11.14}$$

přičemž σ_1 a σ_m jsou největší a nejmenší singulární hodnoty.

11.2 Singulární rozklad matice (SVD)

Singulární rozklad matice (Singular Value Decomposition – SVD) vychází z následující představy. Matice A typu $m \times n$ zobrazuje vektory x z $\mathbb{R}^n$ do prostoru $\mathcal{R}(A) \subset \mathbb{R}^m$, přitom jednotkové vektory z $\mathbb{R}^n$ (1-sféru) zobrazí na hyperelipsu v $\mathbb{R}^m$.



Obrázek 11.4: Vlevo kružnice (jednotková sféra) v $\mathbb{R}^n$ se dvěma šipkami z počátku označenými $\vec{v}_2$ a $\vec{v}_1$, kružnice je popsána S, prostor je označen $\mathbb{R}^n$. Uprostřed vodorovná šipka s popiskem A. Vpravo natočená elipsa v prostoru označeném $\mathbb{R}^m$ se dvěma šipkami z počátku podél hlavních poloos, označenými $\sigma_1\vec{u}_1$ a $\sigma_2\vec{u}_2$.

Přitom některé směry mohou být degenerované, $\sigma_i = 0$ (pro matice, které nemají hodnost n).

Lze najít ortonormální bázi v $\mathbb{R}^n$ (vektory $\vec{v}_1, \vec{v}_2 \dots$) takovou, že

$$A\vec{v}_i = \sigma_i \vec{u}_i, \tag{11.15}$$

kde $\vec{u}_i$ jsou jednotkové vektory ve směrech hlavních poloos hyperelipsy a σ_i budou seřazeny, tj. $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

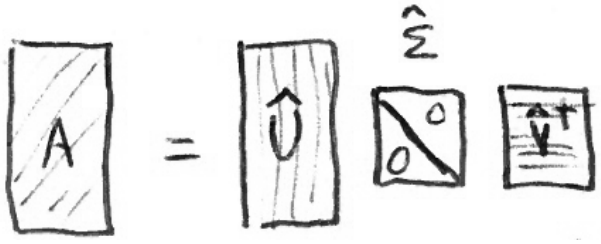
Maticově zapsáno

$$\overset{m\times n}{A} \overset{n\times n}{\left(\begin{array}{c|c|c|c} v_1 & v_2 & \dots & v_n \end{array} \right)} = \overset{m\times n}{\left(\begin{array}{c|c|c|c} u_1 & u_2 & \dots & u_n \end{array} \right)} \begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_n \end{pmatrix},$$

neboli

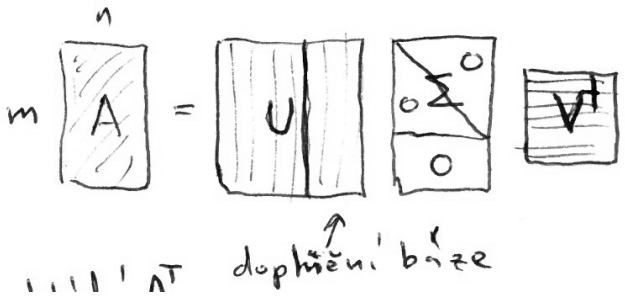
$$A\hat{V} = \hat{U}\hat{\Sigma} \qquad \Rightarrow \qquad A = \hat{U}\hat{\Sigma}\hat{V}^\dagger. \tag{11.16}$$

Přitom $\hat{V}$ je unitární a $\hat{U}$ splňuje $\hat{U}^\dagger\hat{U} = \mathbf{1}_{n\times n}$.



Obrázek 11.5: Malé blokové schéma rozkladu: vysoký šrafovaný obdélník s A, rovnítko, vysoký obdélník se svislými pruhy a $\hat{U}$, nad ním doprava popisek $\hat{\Sigma}$ nad malým čtvercem s diagonální čarou a dvěma nulami mimo diagonálu, a nakonec malý čtverec s vodorovnými pruhy a $\hat{V}^\dagger$.

Hodnoty σ_i jsou tzv. singulární hodnoty (na rozdíl od vlastních čísel jsou reálné), u_i jsou levé singulární vektory a v_i jsou pravé singulární vektory.



Obrázek 11.6: Blokové schéma plného rozkladu. Vlevo šrafovaný obdélník s A, vlevo od něj popisek m, nad ním popisek n; rovnítko; široký obdélník se svislými pruhy a U, uvnitř silná svislá čára dělící původní a doplněné sloupce; dále obdélník s diagonální čarou a Σ, s O vpravo nahoře, O vlevo a O v dolní části pod vodorovnou čarou; nakonec čtverec s vodorovnými pruhy a $V^\dagger$. Zespoda míří šipka k dělící čáře v bloku U s popiskem „doplnění báze".

Doplněním báze v $\mathbb{R}^m$ dostaneme tzv. plný singulární rozklad; každá matice je tedy diagonální ve vhodné bázi.

Pro $m < n$ se rozkládá A^T.

Věta 11.2.1 (Existence a jednoznačnost SVD rozkladu)**.** *Každá matice A typu $m \times n$ má SVD rozklad. Navíc $\{\sigma_i\}$ jsou dány jednoznačně, a pokud jsou navzájem různé, pak i $\{u_i\}$ a $\{v_i\}$ jsou dány jednoznačně až na komplexní fázi (± 1 u reálných matic).*

Důkaz. Indukcí: nechť existuje SVD pro matici $(m-1) \times (n-1)$. Položme

$$\sigma_1 = \|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sup_{\|x\|_2 = 1} \|Ax\|_2.$$

Určitě existují v_1 a u_1 takové, že

$$\|v_1\|_2 = 1\,, \quad \|u_1\|_2 = 1 \quad \text{a} \quad Av_1 = \sigma_1 u_1.$$

Doplněním v_1 a u_1 na ortonormální bázi (libovolně) dostaneme unitární matice U_1 a V_1, pro které

$$U_1^\dagger A V_1 = S = \begin{pmatrix} \sigma_1 & w^\dagger \\ 0 & B \end{pmatrix},$$

a platí

$$\left\| S \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\|_2 \geq \sigma_1^2 + w^\dagger w = \left(\sigma_1^2 + w^\dagger w\right)^{1/2} \left\| \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\|_2.$$

Odtud $\|S\|_2 \geq \left(\sigma_1^2 + w^\dagger w\right)^{1/2}$, ale U_1, V_1 jsou unitární, a tedy $\|S\|_2 = \|A\|_2 = \sigma_1$, z čehož plyne $w = 0$; zbytek plyne indukcí. □

11.2.1 Základní vlastnosti SVD rozkladu

SVD vyjadřuje možnou změnu báze v $\mathbb{R}^m$ a $\mathbb{R}^n$ tak, aby

$$b = Ax \quad \text{přešlo na} \quad b' = \Sigma x', \text{ kde } x' = V^\dagger x \text{ a } b' = U^\dagger b. \tag{11.17}$$

Rozdíl oproti vlastním hodnotám a vektorům je následující. Ty mají smysl jen pro čtvercové matice a rozklad $A = X\Lambda X^{-1}$ existuje jen pro nedefektní matice (viz později), přičemž oba prostory jsou ve stejné bázi. Pokud je matice **hermitovská** ($A = A^\dagger$), **pak** lze λ_i volit tak, že $\sigma_i = |\lambda_i|$.

Hodnost matice A je rovna hodnosti Σ (U i V mají plnou hodnost), a tedy

$$\Rightarrow \ \#\text{ nenulových } \sigma_i \ \text{ určuje hodnost } A.$$

Pokud je r nenulových σ_i, pak $\mathcal{R}(A) = \mathcal{L}(u_1, \ldots, u_r)$ a $\mathrm{Ker}(A) = \mathrm{null}(A) = \mathcal{L}(v_{r+1}, \ldots, v_n)$.

Dále platí

$$\|A\|_2 = \sigma_1 = \max \sigma_i = \|\Sigma\|_2 \qquad \text{a} \qquad \|A\|_F = \left(\sum_{i,j} |a_{ij}|^2 \right)^{1/2} = \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2}. \tag{11.18}$$

Sestrojme matici $A^\dagger A$ typu $n \times n$; pak

$$A^\dagger A = (U\Sigma V^\dagger)^\dagger (U\Sigma V^\dagger) = V(\Sigma^\dagger \Sigma) V^\dagger, \tag{11.19}$$

a tedy $A^\dagger A$ a $\Sigma^\dagger \Sigma$ mají stejná vlastní čísla $\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2$.

Pro matici A typu $m \times m$ je

$$\det(A) = \prod_{i=1}^{m} \sigma_i. \tag{11.20}$$

Rozepsáním Σ na součet matic tvaru

$$\begin{pmatrix} 0 & & 0 \\ & \sigma_i & \\ 0 & & 0 \end{pmatrix}$$

dostaneme (přes počet r nenulových σ_i)

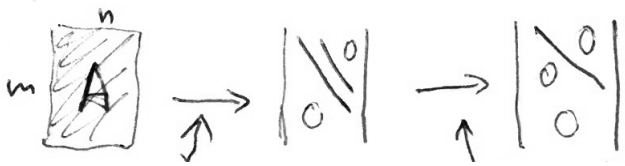
$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^\dagger, \tag{11.21}$$

odkud $\Rightarrow$ aproximace A omezením na velká σ_i, tedy „nejlepší aproximace A“ pomocí matice určité hodnosti.

11.2.2 Výpočet SVD rozkladu (nástin)

Přímočaré by bylo najít vlastní čísla $A^\dagger A = V \Lambda V^\dagger$ a spočítat Σ jako odmocninu Λ, pak vyřešit $U\Sigma = AV$ (např. QR faktorizací). Ukazuje se ovšem, že jde o nestabilní algoritmus, pokud některá $\sigma_k \ll \|A\|$.

Stabilní algoritmus je založen na převedení na bidiagonální matici.

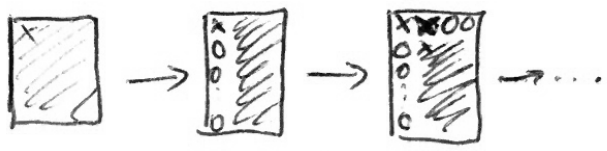


Obrázek 11.7: Schéma převodu matice na bidiagonální tvar – vyšrafovaný obdélník označený A s popisky rozměrů m (vlevo) a n (nahoře), šipkou převeden na matici s šikmým pruhem a nulami (popisky 0 vpravo nahoře a 0 vlevo dole), a další šipkou na matici s bidiagonální strukturou (popisky 0 vlevo nahoře, 0 vpravo nahoře a 0 dole). Zespoda k oběma šipkám míří šipky od komentářů pod obrázkem.

První přeměnu obstarává Golubova–Kahanova bidiagonalizace, $O(mn^2)$, přesněji $4mn^2 - \frac{4}{3}n^3$ operací. Druhou přeměnu pak velmi efektivní iterační metody, $O(n^2)$, založené na QR faktorizaci nebo na metodě rozděl a panuj.

Pro bidiagonalizaci se používají např. Householderovy reflexe zleva i zprava.

Pro $m \gg n$ to jde i lépe, až $2mn^2 + 2n^3$ operací.



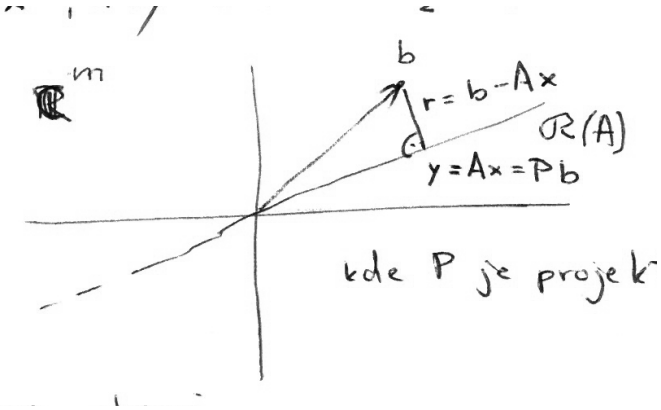
Obrázek 11.8: Posloupnost tří malých obdélníkových matic spojených šipkami (a dále „$\rightarrow \ldots$“), znázorňující postupné vynulování prvků Householderovými reflexemi – v první matici je vlevo nahoře křížek a zbytek vyšrafovaný, ve druhé je v prvním sloupci $\times, 0, 0, \ldots, 0$ a zbytek vyšrafovaný, ve třetí přibývá v prvním řádku $\times, \times, 0, 0$ a v prvním sloupci $\times, 0, 0, \ldots, 0$.

Kapitola 12

Numerická lineární algebra. Řešení přeurčené soustavy

12.1 Řešení přeurčeného systému $Ax = b$ metodou nejmenších čtverců

Úlohou je nalézt takové x, aby norma $\|Ax - b\|_2$ byla nejmenší. Geometricky jde o situaci v prostoru $\mathbb{C}^m$, kterou znázorňuje obrázek 12.1: hledáme kolmý průmět vektoru b do oboru hodnot $\mathcal{R}(A)$, přičemž reziduum $r = b - Ax$ je na tento podprostor kolmé. Rozměry jednotlivých činitelů soustavy ukazuje obrázek 12.2.



Obrázek 12.1: Geometrické schéma metody nejmenších čtverců – dvě osy (svislá a vodorovná) a šikmá přímka procházející počátkem znázorňující podprostor. Z počátku vede šikmá čárkovaná šipka k bodu označenému b, z něj svisle dolů kolmice (vyznačen pravý úhel) na šikmou přímku. Popisky: b, $r = b - Ax$, $\mathcal{R}(A)$, $y = Ax = Pb$, vlevo nahoře $\mathbb{C}^m$.

Zde P je projektor na $\mathcal{R}(A)$.

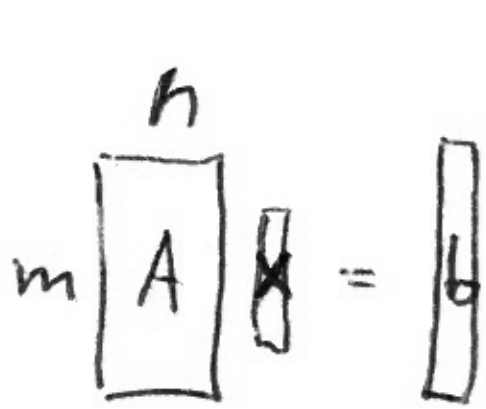
Příklad 12.1.1. Typickým příkladem je lineární či polynomiální regrese, obecně proložení dat lineární kombinací určitých funkcí

$$f(x) = c_1\phi_1(x) + \cdots + c_n\phi_n(x), \tag{12.1}$$

přičemž máme zadány dvojice (x_i, f_i) pro $i = 1, \ldots, m$. Pokud $m > n$, dostáváme přeurčenou soustavu a hledáme $c_1, \ldots, c_n$ takové, aby

$$\sum_{i=1}^{m} |f(x_i) - f_i|^2$$

bylo nejmenší.



Obrázek 12.2: Rozměrové schéma soustavy – obdélník s písmenem A (nad ním rozměr n, vlevo od něj rozměr m) násobený úzkým vysokým obdélníkem s x, rovná se úzkému vysokému obdélníku s b.

Existují tři základní metody hledání x:

1. řešením normálních rovnic,
2. pomocí QR rozkladu,
3. pomocí SVD rozkladu.

12.1.1 Normální rovnice

Normální rovnice mají tvar

$$A^\dagger A x = A^\dagger b. \tag{12.2}$$

Pokud má A plnou hodnost n, pak ji má také $A^\dagger A$, tuto matici lze invertovat, a dostáváme tak jednoznačné řešení

$$x = \underbrace{(A^\dagger A)^{-1} A^\dagger}_{\substack{\text{pseudoinverze } A^+ \text{ matice } A \\ \text{jde o matici } n\times m}} b. \tag{12.3}$$

Věta 12.1.2 (Charakterizace řešení ve smyslu nejmenších čtverců)**.** *Nechť $A \in \mathbb{C}^{m\times n}$, kde $m \geq n$, a $b \in \mathbb{C}^m$. Vektor $x \in \mathbb{C}^n$ minimalizuje $\|r\|_2 = \|b - Ax\|_2$ právě tehdy, když*

$$A^\dagger r = 0 \iff A^\dagger A x = A^\dagger b \iff Pb = Ax. \tag{12.4}$$

První z podmínek v (12.4) je vyjádřením kolmosti rezidua r na $\mathcal{R}(A)$. Druhá z nich z ní plyne dosazením za r a třetí je zřejmá z Pythagorovy věty.

Projektor P lze vyjádřit tak, že vztah pro řešení

$$x = (A^\dagger A)^{-1} A^\dagger b$$

vynásobíme zleva maticí A, čímž dostaneme

$$P = A(A^\dagger A)^{-1} A^\dagger, \tag{12.5}$$

pokud má A hodnost n.

Matice $A^\dagger A$ je symetrická a pozitivně definitní, lze tedy použít Choleského faktorizaci.

Náročnost je následující: sestavení $A^\dagger A$ stojí mn^2 flops (díky symetrii, jinak dvakrát tolik), výpočet $A^\dagger b$ stojí $2mn$ flops a Choleského řešení soustavy $A^\dagger A x = c$ stojí $\frac{1}{3}n^3$ flops. Celkem tedy

$$\sim mn^2 + \tfrac{1}{3}n^3 \text{ flops}. \tag{12.6}$$



Obrázek 12.3: Schéma redukovaného QR rozkladu – vysoký obdélník rozdělený svislými čarami na sloupce (nad ním rozměr n, vlevo m) násobený menším čtvercem (nad ním n, vlevo n) s nulou 0 v dolním trojúhelníku a šrafovaným horním trojúhelníkem.

12.1.2 QR rozklad

Použijeme redukovaný QR rozklad matice A, tedy $A = \hat{Q}\hat{R}$, jehož schéma je na obrázku 12.3.

Vektory v $\hat{Q}$ tvoří bázi v $\mathcal{R}(A)$, a proto je projektor $P = \hat{Q}\hat{Q}^\dagger$. Tedy

$$\hat{Q}\hat{R}x = Ax = Pb = \hat{Q}\hat{Q}^\dagger b, \qquad \text{neboli} \qquad \hat{R}x = \hat{Q}^\dagger b. \tag{12.7}$$

Matice $\hat{R}$ je horní trojúhelníková, takže poslední soustavu vyřešíme zpětnou substitucí, pokud je A nesingulární.

Náročnost QR faktorizace pomocí Householderových reflexí je

$$\sim 2mn^2 - \tfrac{2}{3}n^3 \text{ flops}, \tag{12.8}$$

což je pro $m \gg n$ dvakrát více než u metody normálních rovnic, zatímco pro $m \sim n$ jsou obě náročnosti srovnatelné.

12.1.3 SVD rozklad

Umíme-li spočítat redukovaný SVD rozklad $A = \hat{U}\hat{\Sigma}\hat{V}^\dagger$, pak $P = \hat{U}\hat{U}^\dagger$, a tedy

$$\hat{U}\hat{\Sigma}V^\dagger x = Ax = Pb = \hat{U}\hat{U}^\dagger b,$$
$$\hat{\Sigma}V^\dagger x = \hat{U}^\dagger b,$$

odkud

$$x = \hat{V}\hat{\Sigma}^{-1}\hat{U}^\dagger b, \tag{12.9}$$

pokud jsou všechna $\sigma_i \neq 0$.

Náročnost je typicky

$$\sim 2mn^2 + 11n^3, \tag{12.10}$$

jde tedy obecně o nejpomalejší z uvedených metod.

12.1.4 Porovnání metod

Metoda normálních rovnic je obecně nejrychlejší, avšak ne vždy stabilní. Doporučeným algoritmem je proto druhá metoda, tedy řešení pomocí QR rozkladu. Jsou-li ovšem některé singulární hodnoty σ_i blízké nule, tj. je-li matice A téměř singulární, pak je nejstabilnější třetí metoda využívající SVD rozklad.

Poznámka 12.1.3. Pokud A nemá hodnost n, je nutné specifikovat dodatečnou podmínku, aby bylo x jednoznačné, například požadovat, aby $\|x\|_2$ byla nejmenší. V těchto případech je obecně stabilní pouze SVD rozklad.

Kapitola 13

Jacobiho metoda pro vlastní čísla reálné symetrické matice

13.1 Základní myšlenka

Jacobiho metoda je základní jednoduchá metoda pro výpočet vlastních čísel reálných symetrických matic. Její základní myšlenkou je *nulování mimodiagonálních prvků pomocí rotace*, tedy pomocí ortogonální transformace.

Příklad 13.1.1 (Matice 2×2)**.** Matici typu 2×2 lze diagonalizovat přesně. Máme

$$A = \begin{pmatrix} a & d \\ d & b \end{pmatrix}$$

a hledáme matici

$$J = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \qquad \begin{aligned} c &= \cos\varphi, \\ s &= \sin\varphi, \end{aligned}$$

takovou, aby platilo

$$J^T A J = \begin{pmatrix} x & 0 \\ 0 & y \end{pmatrix}.$$

Dostaneme podmínku

$$d\,(c^2 - s^2) - cs\,(b - a) = 0, \tag{13.1}$$

neboli

$$\operatorname{tg} 2\varphi = \frac{2d}{b-a}. \tag{13.2}$$

13.2 Nulování mimodiagonálních prvků obecné matice

Pro větší matice nulujeme prvky pod diagonálou a nad diagonálou iteračně, a to pomocí matic typu

$$P = \begin{pmatrix} 1 & \ddots & & & & 0 \\ & & c & 0 & s & \\ & & 0 & 1 & 0 & \\ & & -s & 0 & c & \\ 0 & & & & \ddots & 1 \end{pmatrix}. \tag{13.3}$$

Použitím takovéto matice sice vynulujeme prvek na příslušném řádku a v příslušném sloupci, ale zároveň zrušíme nuly z předchozích iterací. Přesto to funguje: provádíme-li toto nulování postupně několikrát pro všechny mimodiagonální prvky, dostaneme nakonec diagonální matici.

Nulujeme-li prvek a_{pq} (a tedy též a_{qp}) na p-tém řádku a v q-tém sloupci, změní se p-tý a q-tý řádek a sloupec takto:

$$\begin{aligned} a'_{rp} &= a'_{pr} = c\,a_{rp} - s\,a_{rq} \\ a'_{rq} &= a'_{qr} = c\,a_{rq} + s\,a_{rp} \end{aligned} \qquad \text{pro } r \neq p \text{ a } r \neq q, \tag{13.4}$$

a dále

$$\begin{aligned} a'_{pp} &= c^2 a_{pp} + s^2 a_{qq} - 2cs\,a_{pq}, \\ a'_{qq} &= s^2 a_{pp} + c^2 a_{qq} + 2cs\,a_{pq}, \\ a'_{pq} &= a'_{qp} = (c^2 - s^2)\,a_{pq} + cs\,(a_{pp} - a_{qq}) = 0. \end{aligned}$$

Z poslední podmínky plyne, že musí být

$$\operatorname{cotg}(2\varphi) = \frac{c^2 - s^2}{2cs} = \frac{a_{qq} - a_{pp}}{2a_{pq}}. \tag{13.5}$$

13.3 Numericky stabilní formulace

Tato základní verze se ovšem kvůli stabilitě nepoužívá. Zaokrouhlovací chyby se projeví méně, pokud vše přepíšeme následujícím způsobem.

Označme

$$t = \frac{s}{c}, \qquad \Theta = \operatorname{cotg}(2\varphi) = \frac{c^2 - s^2}{2sc} = \frac{1 - t^2}{2t}. \tag{13.6}$$

Veličinu t lze vyjádřit přímo pomocí $\Theta = \dfrac{a_{qq} - a_{pp}}{2a_{pq}}$ řešením kvadratické rovnice

$$t^2 + 2t\Theta - 1 = 0 \quad \Rightarrow \quad t = -\Theta \pm \sqrt{\Theta^2 + 1}, \tag{13.7}$$

přičemž se ukazuje, že stabilnější volbou je menší ze dvou kořenů:

$$t = \begin{cases} -\Theta + \sqrt{\Theta^2 + 1} = \dfrac{1}{\Theta + \sqrt{\Theta^2 + 1}} & \text{pro } \Theta > 0, \\ -\Theta - \sqrt{\Theta^2 + 1} = \dfrac{-1}{-\Theta + \sqrt{\Theta^2 + 1}} & \text{pro } \Theta < 0, \end{cases} \tag{13.8}$$

tedy dohromady

$$t = \frac{\operatorname{sign}(\Theta)}{|\Theta| + \sqrt{\Theta^2 + 1}}. \tag{13.9}$$

Poznámka 13.3.1. Vyjádření kořenů pomocí zlomků v (13.8) se používá kvůli odečítání dvou blízkých čísel ve jmenovateli pro velké Θ. Je-li Θ příliš velké, takže by Θ^2 přeteklo, klademe $t = \dfrac{1}{2\Theta}$.

Hodnoty s a c lze z t vyjádřit takto:

$$t^2 = \frac{1 - c^2}{c^2} \quad \Rightarrow \quad c = \frac{1}{\sqrt{t^2 + 1}}\,, \quad s = tc. \tag{13.10}$$

S využitím vztahu $a_{qq} = a_{pp} + \dfrac{c^2 - s^2}{cs}\,a_{pq}$ dostaneme nakonec

$$\begin{aligned} a'_{pq} &= a'_{qp} = 0 & \qquad a'_{rp} &= a'_{pr} = a_{rp} - s\,(a_{rq} + \tau\,a_{rp}) \\ a'_{pp} &= a_{pp} - t\,a_{pq} & \qquad a'_{rq} &= a'_{qr} = a_{rq} + s\,(a_{rp} - \tau\,a_{rq}) \\ a'_{qq} &= a_{qq} + t\,a_{pq} & \qquad \text{kde } \tau &= \frac{s}{1 + c}\,. \end{aligned}$$

13.4 Konvergence a náročnost

Lze spočítat, že suma mimodiagonálních prvků

$$S = \sum_{r \neq s} |a_{rs}|^2 \tag{13.11}$$

se změní na

$$S' = S - 2|a_{pq}|^2, \tag{13.12}$$

tedy se zmenší. Postupně tak zmenšujeme S na nulu; je-li a_{pq} „malé“, vynechává se.

Jak rychle konvergence probíhá? Pokud bychom vždy volili největší $a_{pq} > \frac{S}{n^2}$, pak by platilo

$$S' < \left(1 - \frac{1}{n^2}\right) S \quad \Rightarrow \quad S^{(k)} < \left(1 - \frac{1}{n^2}\right)^k S, \tag{13.13}$$

a protože každý krok vyžaduje $\mathcal{O}(n)$ operací, bude celkem potřeba $\mathcal{O}(n^3)$ operací. Obecně je ovšem tato diskuze o něco delší.

13.5 Vlastní vektory

Vlastní vektory lze získat jako sloupce matice

$$V = \mathbb{1} \cdot P_1 P_2 \dots P_k, \tag{13.14}$$

kterou počítáme postupnou změnou matice $V = \mathbb{1}$ pomocí vztahů

$$\begin{aligned} v'_{rs} &= v_{rs}\ , \quad s \neq p\,,\ s \neq q, \\ v'_{rp} &= c\, v_{rp} - s\, v_{rq} = v_{rp} - s\,(v_{rq} + \tau v_{rp}), \\ v'_{rq} &= s\, v_{rp} + c\, v_{rq} = v_{rq} + s\,(v_{rp} - \tau v_{rq}). \end{aligned}$$

Kapitola 14

Rychlá Fourierova transformace a její využití

14.1 Fourierova transformace

14.1.1 Definice a volba normalizace

Definice 14.1.1 (Fourierova transformace)**.** Pro libovolnou funkci $u \in L^2(\mathbb{R})$ je Fourierův obraz dán vztahem

$$\hat{u}(\xi) = (\mathcal{F}u)(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i\xi x}\, u(x)\, \mathrm{d}x, \qquad \xi \in \mathbb{R}. \tag{14.1}$$

Obraz $\hat{u}$ je opět z $L^2(\mathbb{R})$ a funkci $u(x)$ lze z $\hat{u}(\xi)$ získat zpět pomocí *inverzní Fourierovy transformace*

$$u(x) = (\mathcal{F}^{-1}\hat{u})(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\xi x}\, \hat{u}(\xi)\, \mathrm{d}\xi, \qquad x \in \mathbb{R}. \tag{14.2}$$

Poznámka 14.1.2. Občas se v literatuře liší faktor $\frac{1}{\sqrt{2\pi}}$, případně je použito $e^{-i2\pi\xi x}$. Obecně je-li

$$\hat{u}(\xi) = \frac{a}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-ic\xi x}\, u(x)\, \mathrm{d}x \qquad \text{a} \qquad u(x) = \frac{b}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ic\xi x}\, \hat{u}(\xi)\, \mathrm{d}\xi, \tag{14.3}$$

pak musí být

$$\frac{ab}{|c|} = 1,$$

což plyne z identity

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ic\xi(x-x')}\, \mathrm{d}\xi = \frac{1}{|c|}\, \delta(x - x').$$

14.1.2 Parsevalova rovnost a konvoluce

Pro obecnou volbu konstant (14.3) má Parsevalova rovnost tvar

$$\int_{-\infty}^{\infty} f(x)\, g^*(x)\, \mathrm{d}x = \frac{|c|}{|a|^2} \int_{-\infty}^{\infty} \hat{f}(\xi)\, \hat{g}^*(\xi)\, \mathrm{d}\xi \qquad \text{neboli} \qquad \|f\|_2 = \sqrt{\frac{|c|}{|a|^2}}\, \|\hat{f}\|_2, \tag{14.4}$$

takže speciálně pro $a = 1$ a $c = 1$ dostáváme

$$\int_{-\infty}^{\infty} f(x)\, g^*(x)\, \mathrm{d}x = \int_{-\infty}^{\infty} \hat{f}(\xi)\, \hat{g}^*(\xi)\, \mathrm{d}\xi \qquad \text{neboli} \qquad \|f\|_2 = \|\hat{f}\|. \tag{14.5}$$

Pro konvoluci

$$(u * v)(x) = \int_{-\infty}^{\infty} u(x-y)\, v(y)\, \mathrm{d}y = \int_{-\infty}^{\infty} u(y)\, v(x-y)\, \mathrm{d}y \tag{14.6}$$

platí (je-li $u \in L^2$ a $v \in L^1$ nebo obráceně)

$$\widehat{u * v}(\xi) = \hat{u}(\xi)\,\hat{v}(\xi). \tag{14.7}$$

14.1.3 Vlastnosti Fourierovy transformace

1. Linearita: $\widehat{u+v}(\xi) = \hat{u}(\xi) + \hat{v}(\xi)$ a $\widehat{cu}(\xi) = c\,\hat{u}(\xi)$.
2. Translace: pokud $x_0 \in \mathbb{R}$, pak $\widehat{u(x+x_0)}(\xi) = e^{i\xi x_0}\,\hat{u}(\xi)$.
3. Modulace: pokud $\xi_0 \in \mathbb{R}$, pak $\widehat{e^{i\xi_0 x}u(x)}(\xi) = \hat{u}(\xi - \xi_0)$.
4. Dilatace: pokud $c \in \mathbb{R}$, $c \neq 0$, pak $\widehat{u(cx)}(\xi) = \hat{u}(\xi/c)\,/|c|$.
5. Sdružení: $\widehat{u^*}(\xi) = \hat{u}(-\xi)^*$.
6. Derivace: $\widehat{u_x}(\xi) = i\xi\,\hat{u}(\xi)$.

14.2 Diskrétní Fourierova transformace

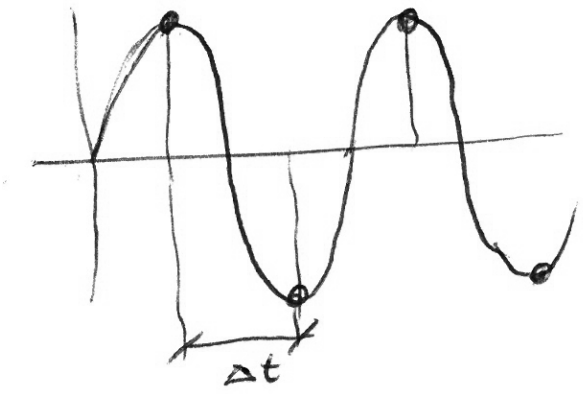
14.2.1 Vzorkování a Nyquistova kritická frekvence

Uvažujme funkci $h(t)$ zadanou pouze s jistou vzorkovací frekvencí v bodech $t_n = n\,\Delta t$, kde $n = \ldots, -3, -2, -1, 0, 1, 2, 3, \ldots$. Hodnota $\frac{1}{\Delta t}$ přitom odpovídá počtu bodů (sampling rate) za jednu sekundu.

Nejvyšší frekvence, kterou lze takto popsat, je dána *Nyquistovou kritickou frekvencí.*

Definice 14.2.1 (Nyquistova kritická frekvence)**.**

$$f_c = \frac{1}{2\Delta t}. \tag{14.8}$$



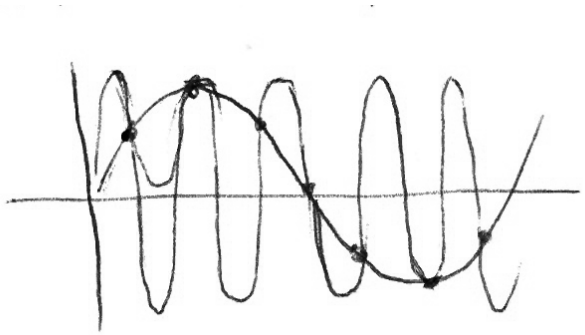
Obrázek 14.1: Sinusovka s vyznačenými vzorkovacími body v jejích maximech a minimech; rozestup dvou sousedních vzorků je označen jako Δt.

Máme tedy dva body na jednu periodu, což je minimum nutné k zachycení této frekvence, tj. vlny $\sin 2\pi f_c t$.

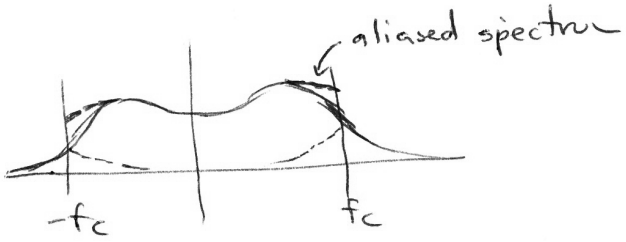
14.2.2 Problém aliasingu

- Pokud Fourierův obraz obsahuje vyšší frekvence f než f_c, dochází k aliasingu neboli příspěvek vyšších frekvencí se objeví v nižších.

- Vzorkování s krokem Δt totiž nerozliší frekvence f_1 a f_2, pokud se liší o násobek $\frac{1}{\Delta t}$, což je šířka frekvenčního rozsahu $\langle -f_c, f_c \rangle$.



Obrázek 14.2: Rychle kmitající sinusovka vzorkovaná v bodech vyznačených tečkami; vzorky leží na pomalé sinusovce nakreslené silnější čarou, která je aliasem původního rychlého signálu.



Obrázek 14.3: Spektrum se svislými čarami v bodech $-f_c$ a f_c: čárkovaně skutečné spektrum přesahující za $\pm f_c$, plnou čarou vzniklé přeložené spektrum („aliased spectrum“), které je uvnitř intervalu vyšší.

Je tedy nutné vzorkovat podle nejvyšší frekvence, případně použít filtry apod.

14.2.3 Odvození diskrétní Fourierovy transformace

Máme-li funkci $h(t)$ vzorkovánu v časech $t_k = k\,\Delta t$, $k = 0, 1, \ldots, N-1$, tj. máme posloupnost $h_k = h(t_k)$, pak její Fourierův obraz můžeme aproximovat, ovšem jen v určitých frekvencích: z N bodů lze určit nejvýše N nezávislých frekvencí mezi $-f_c$ a f_c, a sice

$$f_n = \frac{n}{N\Delta t}, \quad \begin{array}{ccc} n = -\frac{N}{2}, & \ldots, & \frac{N}{2} \\ \Updownarrow & & \Updownarrow \\ -f_c & & +f_c \end{array}$$

Jde o $N+1$ frekvencí, ale hodnota pro $-f_c$ a f_c bude stejná.

Máme tedy

$$\begin{aligned} H(f_n) &= \int_{-\infty}^{\infty} h(t)\, e^{2\pi i f_n t}\, \mathrm{d}t \simeq \\ &\simeq \sum_{k=0}^{N-1} h_k\, e^{2\pi i f_n t_k}\ \Delta t = \\ &= \Delta t \underbrace{\sum_{k=0}^{N-1} h_k\, e^{2\pi i k n/N}}_{H_n} = \Delta t\, H_n. \end{aligned}$$

I když jsme brali

$$n = -\frac{N}{2}, \ldots, \frac{N}{2},$$

hodnota H_n se nezmění, přičteme-li k n číslo N, takže

$$H_{-n} = H_{N-n}. \tag{14.9}$$

Vztah

$$H_n = \sum_{k=0}^{N-1} h_k\, e^{2\pi i k n/N} \tag{14.10}$$

je tzv. *diskrétní Fourierova transformace* posloupnosti h_k; povšimněme si, že nezávisí na Δt. Proto se obvykle hledá H_n pro $n = 0, \ldots, N-1$, tedy pro stejný rozsah indexů jako u k.

Je ovšem nutné dát pak pozor na to, které frekvence odpovídají kterému H_n. Obvykle tedy H_0 přísluší nulové frekvenci, hodnoty $H_1, \ldots, H_{N/2-1}$ odpovídají kladným frekvencím $0 < f < f_c$, hodnoty $H_{N/2+1}$ až H_{N-1} odpovídají záporným frekvencím $-f_c < f < 0$ a $H_{N/2}$ odpovídá oběma hodnotám f_c i $-f_c$.

14.2.4 Inverzní diskrétní Fourierova transformace

Inverzní diskrétní Fourierova transformace, která dává přesně h_k z H_n, pak je

$$h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n\, e^{-2\pi i k n/N}. \tag{14.11}$$

Od (14.10) se liší jen znaménkem v exponentu a faktorem $\frac{1}{N}$, takže ji lze počítat prakticky stejně.

14.3 Rychlá Fourierova transformace (FFT)

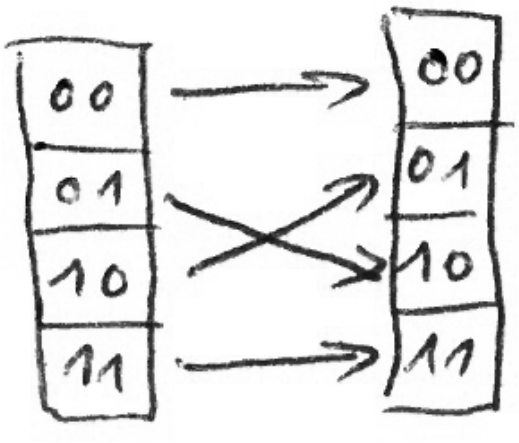
I když je diskrétní Fourierova transformace na první pohled metodou řádu $\mathcal{O}(N^2)$, lze ji pro $N = 2^x$ provést v řádu $\mathcal{O}(N \log_2 N)$ operací, neboť podle Danielsona a Lanczose platí

$$\begin{aligned} H_n &= \sum_{k=0}^{N-1} e^{2\pi i k n/N}\, h_k = \\ &= \sum_{k=0}^{N/2-1} e^{2\pi i n(2k)/N}\, h_{2k} + \sum_{k=0}^{N/2-1} e^{2\pi i n(2k+1)/N}\, h_{2k+1} = \\ &= \sum_{k=0}^{N/2-1} e^{2\pi i n k/(N/2)}\, h_{2k} + W^n \sum_{n=0}^{N/2-1} e^{2\pi i n k/(N/2)}\, h_{2k+1} = \\ &= \underbrace{H_n^e}_{\text{DFT sudých}} + \underbrace{W^n H_n^o}_{\text{DFT lichých vzorků } h_k}. \end{aligned}$$

Tento princip pak použijeme rekurzivně na H_n^e a H_n^o, až bude

$$H_n^{eoeeo\cdots oe} = h_k$$

pro jisté k odpovídající vzoru $eoee\cdots$. A sice obrátíme-li vzor $eoeeo\ldots oe$ binárně, kde $e = 0$ a $o = 1$, dostaneme odpovídající k. Je tedy nutno posloupnost přeuspořádat pomocí *bit reversal* (přehození bitů), například takto.



Obrázek 14.4: Přeuspořádání pomocí bit reversal pro dvoubitové vzory: 00 → 00, 01 → 10, 10 → 01, 11 → 11.

Přeuspořádání lze provést v řádu $\mathcal{O}(N \log_2 N)$ operací. Potom se výsledky kombinují zpět pomocí Danielsonova–Lanczosova vztahu, a to $\log_2 N$-krát, vždy v řádu N operací. Podrobnosti algoritmu viz Numerical Recipes.

Existují i verze pro $N \neq 2^x$, ale jejich efektivita je horší a záleží na faktorizaci čísla N.

14.4 Výpočet Fourierovy transformace pomocí DFT (FFT)

Nechť máme program, který vrací

$$H_n = \sum_{k=0}^{N-1} h_k \, e^{2\pi i k n/N}, \qquad n = 0, \ldots, N-1, \tag{14.12}$$

a chceme pomocí něj (přibližně) spočítat například

$$I(\omega_n) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\omega_n t} \, h(t) \, \mathrm{d}t \tag{14.13}$$

pomocí hodnot h_k v bodech $t_k = t_{\min} + k\,\Delta t$, $k = 0, \ldots, N-1$.

Za předpokladu, že h_k jsou mimo body t_k nulové, dostaneme pomocí lichoběžníkového pravidla

$$I(\omega_m) \cong \frac{1}{\sqrt{2\pi}} \sum_{r=0}^{N-1} h_r \, e^{i\omega_m t_r} \, \Delta t,$$

kde

$$t_r = t_{\min} + r\,\Delta t,$$
$$\omega_m = \frac{2\pi m}{N \Delta t}, \qquad m = -\tfrac{N}{2}, \ldots, \tfrac{N}{2},$$

a tedy

$$I(\omega_m) \cong \frac{\Delta t}{\sqrt{2\pi}}\, e^{i\omega_m t_{\min}} \sum_{r=0}^{N-1} h_r\, e^{2\pi i r m/N}. \tag{14.14}$$

Faktorem $\frac{\Delta t}{\sqrt{2\pi}}\, e^{i\omega_m t_{\min}}$ je přitom nutno výsledek transformace násobit, přičemž suma v (14.14) je

$$\tilde{H}_m = \begin{cases} H_{m+N} & \text{pro } m < 0, \\ H_m & \text{pro } m \geq 0. \end{cases}$$

Navíc je třeba dávat pozor na umístění záporných frekvencí v H_n.

14.5 Časový vývoj v kvantové mechanice pomocí split-operator metody

Časový vývoj je dán evolučním operátorem $U = e^{-iHt}$, tedy

$$|\psi(t)\rangle = e^{-iHt}|\psi_0\rangle, \qquad (\hbar = 1 \text{ — atomové jednotky}), \tag{14.15}$$

kde $H = T + V$, přičemž T je operátor kinetické energie a V potenciální energie. Tyto dva operátory spolu nekomutují, a tedy $e^{-iHt} \neq e^{-iTt}e^{-iVt}$.

Do řádu $\mathcal{O}(\Delta t^2)$ pro malé Δt ovšem platí

$$e^{-iH\Delta t} = e^{-\frac{iV\Delta t}{2}}\, e^{-iT\Delta t}\, e^{-\frac{iV\Delta t}{2}} + \mathcal{O}(\Delta t^3), \tag{14.16}$$

přičemž člen $e^{-iT\Delta t}$ je v p-reprezentaci roven $e^{-i\frac{p^2}{2m}\Delta t}$. Jde tedy o pouhé násobení: působení $e^{-iH\Delta t}$ lze pomocí přechodů mezi $\psi^x(x)$ a $\psi^p(p) = \langle p|\psi\rangle$ převést na násobení buď $e^{-\frac{iV\Delta t}{2}}\psi^x(x)$, nebo $e^{-i\frac{p^2}{2m}\Delta t}\psi^p(p)$.

Je-li $\psi^p(p)$ určeno pomocí diskrétní (rychlé) Fourierovy transformace, je nutno dát pozor na to, jakým p násobíme.

Přechod mezi $\psi^x(x)$ a $\psi^p(p)$ je dán Fourierovými vztahy

$$\psi^p(p) = \frac{1}{(2\pi)^{1/2}} \int_{-\infty}^{\infty} e^{-ipx}\psi^x(x)\,\mathrm{d}x \qquad \text{(inverzní FT)}, \tag{14.17}$$

$$\psi^x(x) = \frac{1}{(2\pi)^{1/2}} \int_{-\infty}^{\infty} e^{ipx}\psi^p(p)\,\mathrm{d}p. \tag{14.18}$$

Přecházíme-li tam a zpět, konstantní faktory nehrají roli, takže stačí násobit obraz z diskrétní Fourierovy transformace vhodným $e^{-i\frac{p^2}{2m}\Delta t}$, a sice s hodnotami

$$p_j = \frac{2\pi j}{N\Delta x} \qquad \text{pro } j = 0, \ldots, \tfrac{N}{2} - 1,$$
$$p_j = \frac{2\pi}{N\Delta x}(j - N) \qquad \text{pro } j = \tfrac{N}{2}, \ldots, N - 1.$$

Zobecnění na metody vyššího řádu pomocí symetrické kompozice viz J. Chem. Phys. **150** (2019) 204113.

Příloha A

Normy vektorů a matic

A.1 Normy na vektorových prostorech

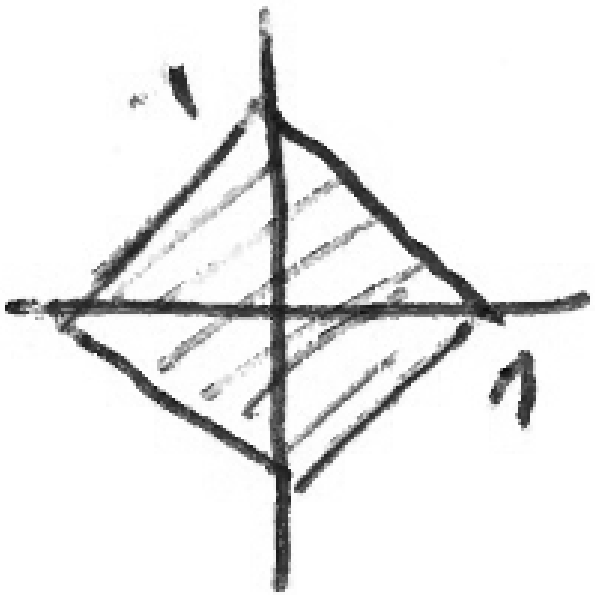
Definice A.1.1 (Norma)**.** Norma $\|\cdot\|$ je zobrazení (funkce) obecně z $\mathbb{C}^m$ do $\mathbb{R}$, které pro $\forall x, y \in \mathbb{C}^m$ a $\forall \alpha \in \mathbb{C}$ splňuje:

1. $\|x\| \geq 0$, přičemž $\|x\| = 0$ právě tehdy, když $x = 0$;
2. $\|x + y\| \leq \|x\| + \|y\|$ (trojúhelníková nerovnost);
3. $\|\alpha x\| = |\alpha| \, \|x\|$.

Na prostoru $\mathbb{C}^m$ se obvykle používají tzv. p-normy. Ty mají následující jednotkové „koule“, tedy množiny $\{x \in \mathbb{C}^m : \|x\| \leq 1\}$.

Pro $p = 1$ je

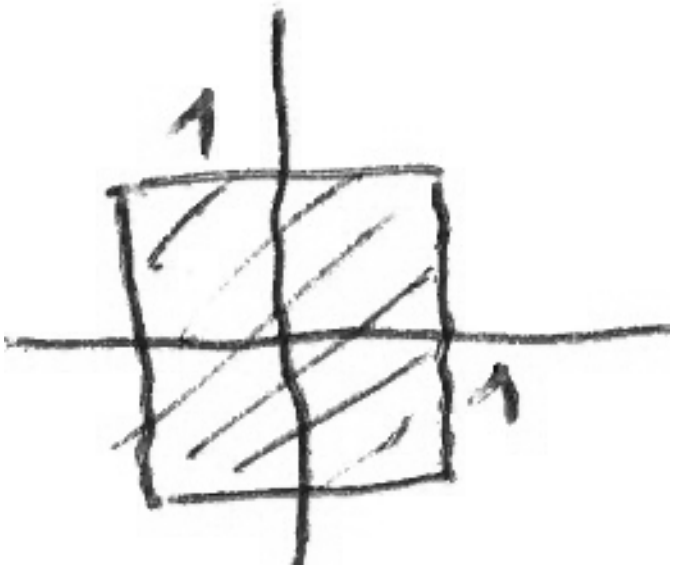
$$\|x\|_1 = \sum_{i=1}^{m} |x_i|. \tag{A.1}$$



Obrázek A.1: Šrafovaná jednotková koule 1-normy v rovině — kosočtverec (otočený čtverec) se středem v počátku, nakreslený přes dvě osy; na svislé ose nahoře a na vodorovné ose vpravo je popisek „1“.

Pro $p = \infty$ je

$$\|x\|_\infty = \max_{1 \le i \le m} |x_i|. \tag{A.2}$$



Obrázek A.2: Šrafovaná jednotková koule ∞-normy v rovině — čtverec se středem v počátku, nakreslený přes dvě osy; popisky „1“ u svislé osy nahoře a u vodorovné osy vpravo.

Pro $p = 2$ je

$$\|x\|_2 = \left(\sum_{i=1}^{m} |x_i|^2 \right)^{1/2}. \tag{A.3}$$

Pro obecné p pak

$$\|x\|_p = \left(\sum_{i=1}^{m} |x_i|^p \right)^{1/p}. \tag{A.4}$$

Používají se případně i vážené p-normy

$$\|x\|_{w,p} = \|Wx\|_p = \left(\sum_{i=1}^{m} |w_i x_i|^p \right)^{1/p}, \tag{A.5}$$

kde W je diagonální matice.

Lze uvažovat i obecnější, libovolnou nesingulární matici W, kdy je však nutno definici upravit.

A.2 Maticová norma indukovaná vektorovými normami

Uvažujme matici $A \in \mathbb{C}^{m \times n}$ jako zobrazení z vektorového prostoru s normou $\|\cdot\|_{(n)}$ do prostoru s normou $\|\cdot\|_{(m)}$; obvykle jde o stejnou normu, jen v různě dimenzionálních prostorech.

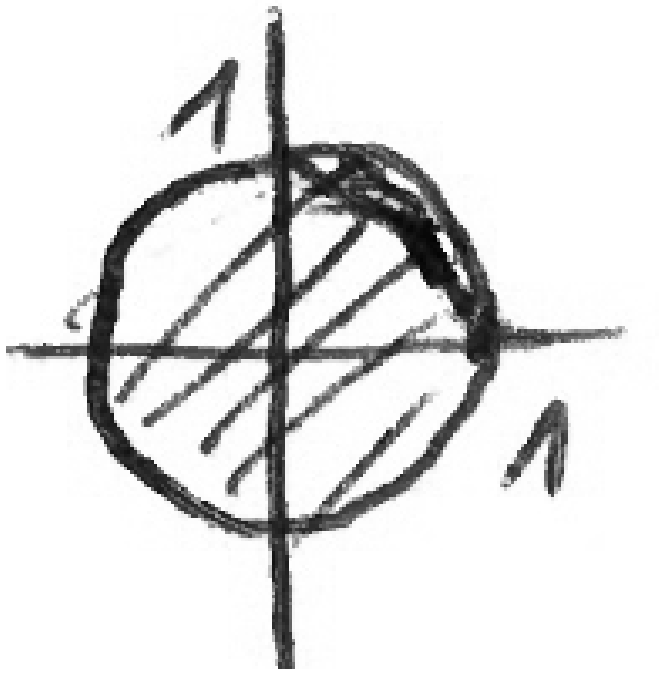
Definice A.2.1 (Indukovaná maticová norma)**.** Indukovaná maticová norma $\|A\|_{(m,n)}$ je nejmenší $C \in \mathbb{R}$ takové, že platí

$$\|Ax\|_{(m)} \le C \|x\|_{(n)},$$

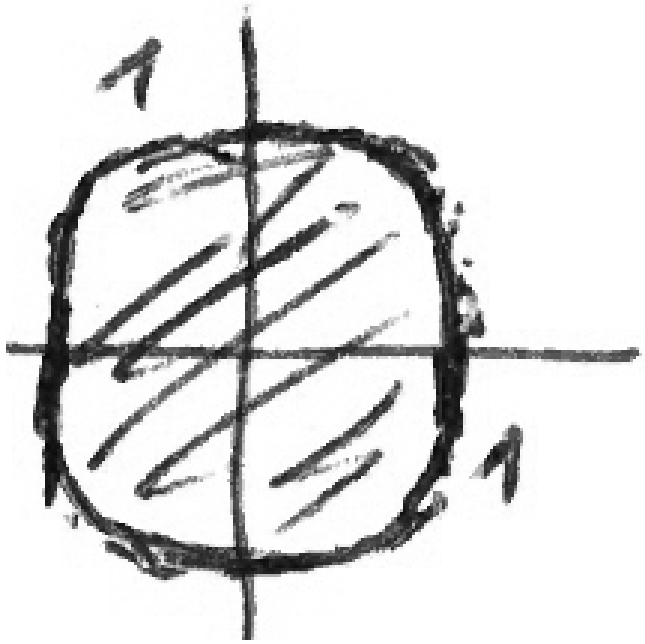
neboli

$$\|A\|_{(m,n)} = \sup_{\substack{x \in \mathbb{C}^n \\ x \neq 0}} \frac{\|Ax\|_{(m)}}{\|x\|_{(n)}} = \sup_{\substack{x \in \mathbb{C}^n \\ \|x\|_{(n)} = 1}} \|Ax\|_{(m)}. \tag{A.6}$$

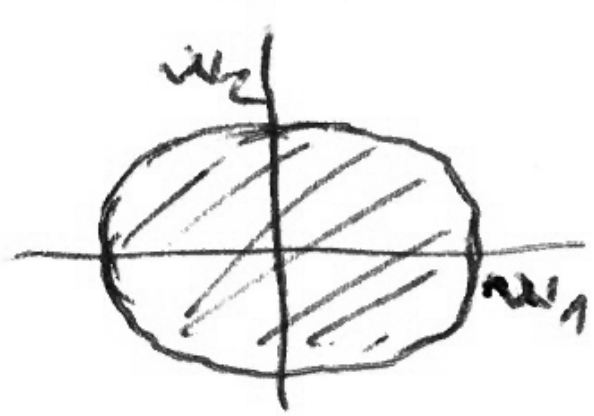
Indukovaná norma tedy vyjadřuje, jak moc se při působení matice A „prodlouží“ jednotkové vektory — přičemž se obecně změní i jejich směr, ba i počet složek.



Obrázek A.3: Šrafovaná jednotková koule 2-normy v rovině — kružnice (kruh) se středem v počátku přes dvě osy; popisky „1“ u svislé osy nahoře a u vodorovné osy vpravo.



Obrázek A.4: Šrafovaná jednotková koule obecné p-normy v rovině — zaoblený čtverec (mezi kruhem a čtvercem) se středem v počátku přes dvě osy; popisky „1“ u svislé osy nahoře a u vodorovné osy vpravo.



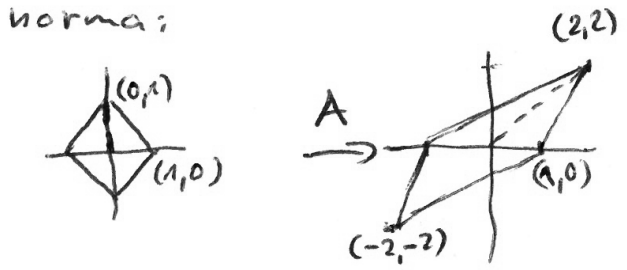
Obrázek A.5: Šrafovaná elipsa (protáhlá vodorovně) se středem v počátku přes dvě osy — jednotková koule vážené normy; svislá osa je popsána w_2, vodorovná osa vpravo w_1.

A.2.1 Příklad

Příklad A.2.2. Určeme normy matice

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

Pro 1-normu dostaneme $\|A\|_1 = 4$.



Obrázek A.6: Vlevo jednotková koule 1-normy v rovině — kosočtverec s vrcholy označenými $(0,1)$ a $(1,0)$, uvnitř svisle vyznačený vektor; šipka s popiskem A vede k pravému obrázku, kde je obraz kosočtverce při zobrazení A — kosodélník s vyznačenými vrcholy $(2,2)$ a $(-2,-2)$, na vodorovné ose bod $(1,0)$, uvnitř čárkovaná úhlopříčka.

Pro 2-normu je

$$\|A\|_2 = \sqrt{\frac{9+\sqrt{65}}{2}} \approx 2{,}9208,$$

a to pro bod

$$\left(\sqrt{\frac{1}{2} - \frac{7}{2\sqrt{65}}},\ \sqrt{\frac{1}{2} + \frac{7}{2\sqrt{65}}}\right).$$

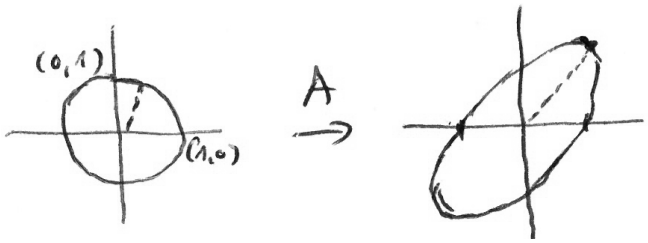
Pro ∞-normu je $\|A\|_\infty = 3$.

A.2.2 Obecné vztahy pro maticové normy

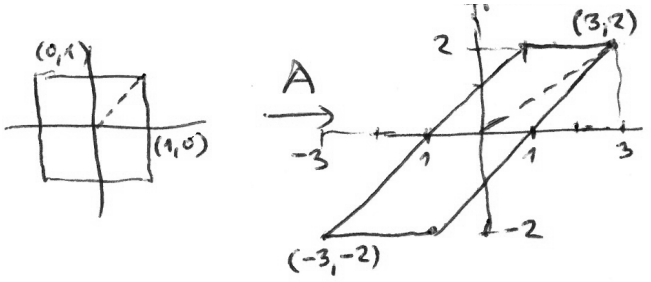
Obecně platí následující tvrzení.

1) 1-norma. Hodnota $\|A\|_1$ je maximum ze součtů absolutních hodnot prvků ve sloupcích, tedy

$$\|A\|_1 = \max_{1 \le j \le n} \|a_j\|_1, \tag{A.7}$$



Obrázek A.7: Vlevo jednotková kružnice 2-normy s popisky $(0,1)$ a $(1,0)$ a čárkovaným poloměrem; šipka s popiskem A vede k pravému obrázku, kde je obrazem elipsa protažená v šikmém směru, s vyznačeným krajním bodem vpravo nahoře a čárkovanou hlavní poloosou.



Obrázek A.8: Vlevo jednotková koule ∞-normy — čtverec s popisky $(0,1)$ a $(1,0)$ a čárkovanou úhlopříčkou; šipka s popiskem A vede k pravému obrázku, kde je obrazem kosodélník s vyznačenými vrcholy $(3,2)$ a $(-3,-2)$, osy jsou popsané hodnotami -3, 1, 3 na vodorovné ose a 2, -2 na svislé, uvnitř čárkovaná úhlopříčka.

kde a_j je j-tý sloupec matice A. Platí totiž

$$\|Ax\|_1 = \Big\| \sum_{j=1}^{n} x_j a_j \Big\|_1 \le \sum_{j=1}^{n} |x_j| \, \|a_j\|_1 \le \|x\|_1 \cdot \max_{1\le j\le n} \|a_j\|_1, \tag{A.8}$$

a volbou $x = e_j$ pro to j, pro něž je $\|a_j\|_1$ maximální, dostaneme rovnost, a tedy

$$\|A\|_1 = \sup_{\|x\|_1=1} \|Ax\|_1 = \max_{1\le j\le n} \|a_j\|_1.$$

2) ∞-norma. Podobně lze ukázat, že $\|A\|_\infty$ je maximum ze součtů absolutních hodnot prvků v řádcích, tedy

$$\|A\|_\infty = \max_{1\le i\le m} \|a_i^*\|_1, \tag{A.9}$$

kde a_i^* je i-tý řádek matice A.

3) 2-norma. Hodnotu $\|A\|_2$ lze určit z maximálního tzv. *singulárního* čísla, tj. jako $\max\{\sigma_i\}_{i=1}^m$, kde singulární čísla se určují pomocí tzv. SVD rozkladu (viz později).

4) Norma součinu. Pro odhad normy součinu dvou matic platí

$$\|AB\|_{(\ell,n)} \le \|A\|_{(\ell,m)} \cdot \|B\|_{(m,n)}, \tag{A.10}$$

a speciálně

$$\|A^n\| \le \|A\|^n, \tag{A.11}$$

kde obecně nejde o rovnost.