Numerická matematika

Radek Kučera, Zuzana Morávková

Katedra matematiky a deskriptivní geometrie
Vysoká škola báňská – Technická Univerzita Ostrava

ISBN

Tento materiál vznikl jako součást řešení interního projektu IRP-FRVŠ 158/2015 *Inovace předmětu Numerická matematika na Fakultě strojní Vysoké školy báňské - Technické univerzitě Ostrava*. Představuje výukový text pro vedení základního kurzu numerické matematiky, která se v podobném rozsahu vyučuje nejen na Fakultě strojní. Je zaměřen na vysvětlení elementárních principů numerických metod zejména s ohledem na jejich algoritmizaci a zpracování na počítači. První kapitola je věnována problematice chyb při řešení numerických úloh, kdy stěžejní pojmy jsou ilustrovány v prostředí programu MATLAB. Až do podoby programů v MATLABu jsou dovedeny téměř všechny algoritmy pro řešení nelineárních rovnic z kapitoly druhé. V následujících kapitolách jsou už uvedeny jen algoritmy, počítačové programy by si měl čtenář vytvořit sám. Tento text doplňuje celá řada dalších studijních materiálů, které vznikly v rámci řešení uvedeného projektu.

OBSAH

KAPITOLA

1

NUMERICKÉ VÝPOČTY A CHYBY

1.1 Obsah předmětu

Numerickou úlohou máme na mysli srozumitelný a jednoznačný popis vztahu mezi **konečným** počtem vstupních a **konečným** počtem výstupních dat (reálných čísel). Konečnost vstupního a výstupního souboru je přitom velmi podstatná - ve svém důsledku znamená, že při řešení úlohy lze použít počítač. Postupy řešení numerických úloh se nazývají *numerické* nebo *počítačové metody*.

Numerické úlohy patří do skupiny úloh *diskrétních*. Matematické modely se však často zapisují jako *úlohy spojité*, u nichž se mezi vstupními nebo výstupními daty vyskytují spojité funkce. Chceme-li takové úlohy řešit numerickými metodami, musíme nejdříve najít způsob, jak tyto úlohy převést na úlohy diskrétní, tj. jak je *diskretizovat*.

Rozpoznat úlohu, kterou lze řešit numericky vyžaduje určitou zkušenost. Na několika jednoduchých příkladech si proto nejdříve ukážeme diskrétní a spojité úlohy, provedeme diskretizaci a vysvětlíme si základní pojmy jako je diskretizační parametr a řád diskretizace.

(1) Úloha vyřešit kvadratickou rovnici $ax^2 + bx + c = 0$, $a \neq 0$, je úloha diskrétní. Vstupní data jsou koeficienty a, b, c, výstupní data jsou reálná čísla $\alpha_1, \beta_1, \alpha_2, \beta_2$, která určují dva obecně komplexní kořeny $x_k = \alpha_k + i\beta_k$, $k = 1, 2$.

(2) Diskrétní úlohou je také soustava lineárních rovnic

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

kde $\mathbf{A} = (a_{ij})$ je daná čtvercová matice řádu n, $\mathbf{b} = (b_i)$ je daný sloupcový vektor o n složkách a $\mathbf{x} = (x_i)$ je sloupcový vektor neznámých také o n složkách. Například pro $n = 3$ můžeme takovou soustavu zapsat ve tvaru:

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

a nebo po jednotlivých rovnicích:

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

Vstupními daty jsou zde prvky matice a_{ij} a vektoru pravých stran b_i. Výstupními daty jsou složky x_i vektoru neznámých. Připomeňme ještě, že řešení může být jediné, může jich být nekonečně mnoho, nebo nemusí existovat. Přitom každý z těchto případů lze popsat pomocí konečného počtu reálných čísel.

(3) Úloha vypočítat určitý integrál

$$I = \int_a^b f(x)\,dx$$

je spojitá úloha, jelikož jedním ze vstupních dat je spojitá funkce f. Dalšími vstupními daty jsou integrační meze a, b a výstupní data představuje jediné reálné číslo I. Samotný pojem „spojitost funkce“, který zde chápeme velmi volně, může mít různý smysl (integrál existuje pro funkci, která je spojitá „po částech“, která není v některých bodech definovaná, atp.). Na tomto příkladě si předvedeme diskretizaci, kdy budeme pro jednoduchost předpokládat, že funkce f je spojitá na intervalu $\langle a, b\rangle$ podle klasické definice spojitosti (takže nevznikne problém s výpočtem funkční hodnoty).

Interval $\langle a, b\rangle$ si rozdělíme na n úseků o stejné délce h pomocí bodů x_i, $i = 0, 1, \ldots, n$, tak, že $x_i - x_{i-1} = h$, $x_0 = a$ a $x_n = b$. Pak můžeme psát:

$$I = \int_{x_0}^{x_1} f(x)\,dx + \int_{x_1}^{x_2} f(x)\,dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)\,dx.$$

Každý dílčí integrál nahradíme jeho přibližnou hodnotou

$$\int_{x_{i-1}}^{x_i} f(x)\,dx \approx hf\left(\frac{x_{i-1} + x_i}{2}\right)$$

a místo přesné hodnoty I budeme počítat její aproximaci:

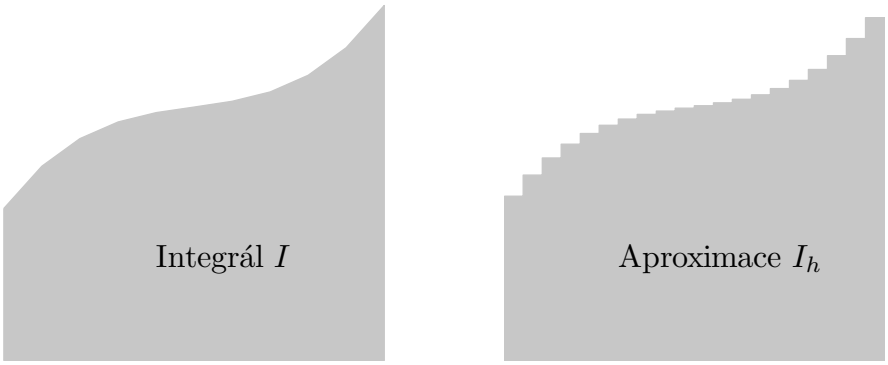
$$I_h = hf\left(\frac{x_0 + x_1}{2}\right) + hf\left(\frac{x_1 + x_2}{2}\right) + \cdots + hf\left(\frac{x_{n-1} + x_n}{2}\right). \tag{1.1}$$

Výpočet podle posledního vzorce je úloha diskrétní. Vstupními daty jsou funkční hodnoty $f\left(\frac{x_{i-1}+x_i}{2}\right)$, $i = 1, \ldots, n$ a velikost kroku h. Výstupem je přibližná hodnota integrálu I_h.

Smysl vzorce (1.1) ukazuje Obrázek 1.1, kde jsou hodnoty I a I_h znázorněny jako plochy příslušného obrazce. Odtud můžeme usoudit, že při menším kroku h bude I_h lépe aproximovat I, což lze zapsat pomocí limity takto:

$$\lim_{h \to 0^+} I_h = I. \tag{1.2}$$

Krok h zde přestavuje *diskretizační parametr*. Obecně platí, že řešení diskretizované úlohy se může přiblížit libovolně přesně k řešení původní spojité úlohy, pokud zvolíme dostatečně malý diskretizační parametr.



Obrázek 1.1: Znázornění integrálu I a jeho aproximace I_h.

Řádem diskretizace nazýváme kladné číslo p, pro které platí

$$|I - I_h| \leq Ch^p, \tag{1.3}$$

kde $C > 0$ je konstanta nezávislá na h. Výraz na levé straně nerovnosti je *velikost diskretizační chyby*, zatímco na pravé straně je *odhad diskretizační chyby*. Vztah (1.3) se u každé konkrétní diskretizace odvodí jejím rozborem. Je zřejmé, že diskretizační chyba bude při zmenšujícím se h klesat k nule tím rychleji, čím větší bude hodnota p. Diskretizace vyššího řádu je proto přesnější, jak názorně ukazuje Tabulka 1.1.

h	$p = 1$	$p = 2$	$p = 3$
0.1	0.1	0.01	0.001
0.01	0.01	0.0001	0.000001
0.001	0.001	0.000001	0.000000001

Tabulka 1.1: Odhady diskretizační chyby Ch^p pro $C = 1$.

Příklad 1.1.1 Pomocí vzorce (1.1) vypočtěte přibližnou hodnotu integrálu

$$I = \int_0^1 x^2\,dx$$

pro $h = 0.5, 0.25$ a 0.125. Z výsledků odhadněte, jaký je řád diskretizace.

Řešení: Přesná hodnota integrálu je $I = \frac{1}{3}$. Přibližné hodnoty vypočítáme takto:

$$\begin{aligned} I_{0.5} &= 0.5(0.25^2 + 0.75^2) = 0.3125, \\ I_{0.25} &= 0.25(0.125^2 + 0.375^2 + 0.625^2 + 0.875^2) = 0.328125, \\ I_{0.125} &= 0.125(0.0625^2 + 0.1875^2 + \cdots + 0.9375^2) = 0.33203125. \end{aligned}$$

Diskretizační chyby mají hodnotu:

$$\begin{aligned} E_{0.5} &= |I - I_{0.5}| = 0.02083333333333, \\ E_{0.25} &= |I - I_{0.25}| = 0.00520833333333, \\ E_{0.125} &= |I - I_{0.125}| = 0.00130208333333. \end{aligned}$$

Při odhadu řádu diskretizace budeme pro jednoduchost předpokládat, že v (1.3) nastane rovnost. Pro $h = 0.5$ pak dostáváme

$$\frac{E_{0.5}}{E_{0.25}} = \frac{Ch^p}{C(h/2)^p} = 2^p \quad \Longrightarrow \quad p = \log_2 \frac{E_{0.5}}{E_{0.25}} = 2.00000000000069.$$

Podobně pro $h = 0.25$ vypočítáme $p = 2.00000000000277$. Z těchto výsledků můžeme usoudit, že diskretizace podle vzorce (1.1) je druhého řádu. □

(4) Úloha najít funkci $y = y(x)$, která vyhovuje diferenciální rovnici

$$y' = x^2 - 0.2y \tag{1.4}$$

a splňuje počáteční podmínku $y(-2) = -1$, je spojitá úloha. Jak uvidíme později, diskretizace této úlohy bude v mnohém podobná postupu, který jsme provedli výše při diskretizaci integrálu.

Kontrolní otázky

Otázka 1. Jaký je rozdíl mezi diskrétní a spojitou úlohou?
Otázka 2. Co je to diskretizace? Jaký je význam diskretizačního parametru?
Otázka 3. Je přesnější diskretizace vysokého nebo nízkého řádu?

Úlohy k samostatnému řešení

1. Vyřešte rovnice: a) $x^2 + 3x + 1 = 0$; b) $x^2 + 2x + 1 = 0$; c) $x^2 + x + 1 = 0$.
2. Řešte následující soustavy lineárních rovnic:

$$a)\ \begin{pmatrix} 1 & 3 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \end{pmatrix}, \quad b)\ \begin{pmatrix} 1 & 2 \\ -1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ -3 \end{pmatrix},$$

$$c)\ \begin{pmatrix} 1 & 2 \\ -1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Připomeňte si jak lze podle hodnoty determinantu rozhodnout o existenci řešení?
3. Pomocí vzorce (1.1) vypočtěte přibližnou hodnotu integrálu

$$I = \int_{-1}^{1} x^2\,dx$$

pro $h = 1, 0.5$ a 0.25 a určete diskretizační chyby.
4. Zopakujte si analytické metody pro řešení diferenciální rovnice (1.4).

Výsledky úloh k samostatnému řešení

1. a) Dva reálné kořeny: $x_1 \doteq -2.6180$, $x_2 \doteq -0.3820$; b) jeden (dvojnásobný) reálný kořen: $x_1 = x_2 = -1$; c) dva komplexní kořeny: $x_1 \doteq -0.5 + i\,0.8660$, $x_2 \doteq -0.5 - i\,0.8660$.
2. a) $\mathbf{x} = (3, 1)^\top$, $\det \mathbf{A} = -7$; b) nekonečně mnoho řešení $\mathbf{x} = (3 - 2t, 3 - 2t)^\top$, $t \in R$, $\det \mathbf{A} = 0$; c) řešení neexistuje.
3. $I \doteq 0.6667$, $I_1 = 0.5$, $|I - I_1| \doteq 0.1667$, $I_{0.5} = 0.625$, $|I - I_{0.5}| \doteq 0.0417$, $I_{0.25} \doteq 0.6563$, $|I - I_{0.25}| \doteq 0.0104$.
4. Obecné řešení je $y(x) = 5x^2 - 50x + 250 + Ce^{-0.2x}$, řešení vyhovující počáteční podmínce je určeno konstantou $C = -248.688737$.

1.2 Zdroje chyb a podmíněnost úloh

Výsledky jsou téměř vždy ovlivněny nějakou chybou - snad jen s výjimkou školních příkladů, které jsou předem připraveny tak, aby "pěkně vyšly". To, že je ve výsledku obsažena chyba, však neznamená, že je něco špatně. Bez chyb to totiž většinou nejde. Je proto dobré vědět, jaké jsou nejčastější příčiny chyb.

Chyba matematického modelu vzniká v důsledku toho, že namísto skutečného technického, fyzikálního nebo jiného praktického problému, řešíme jeho matematický model. Ten může spočívat v řešení rovnic, výpočtu integrálů nebo třeba jde jen o stanovení funkční závislosti. Je-li řešení navrženého matematického modelu z nějakého důvodu složité nebo nemožné, provede se jeho aproximace jednodušší úlohou. Tím vznikne *chyba aproximační*. Jejím speciálním případem je *chyba diskretizační*, kterou jsme zmínili v předchozím odstavci.

Dalším zdrojem chyb je počítání s "nepřesnými" čísly, kam patří *chyby vstupních dat* a *chyby zaokrouhlovací*. Vstupními daty bývají naměřené veličiny, jejichž nepřesnost je dána rozlišovací schopností měřících zařízení, anebo to jsou výsledky předchozích výpočtů, které již zpravidla byly ovlivněny zaokrouhlením. Pro porozumění přesnosti počítačových výpočtů je potřeba mít představu o tom, jak s čísly pracuje počítač. Na dnešních běžných počítačích se počítá "na 16 desetinných míst", což souvisí s počítačovou konstantou nazývanou *počítačové epsilon*, jejíž standardní hodnota je 2.22×10^{-16}, jak uvidíme později.

U rozsáhlejšího výpočtu je často potřeba umět posoudit, jak se zaokrouhlovací chyby šíří v jeho průběhu. Důležité jsou dva scénáře: *(i)* vliv přenášených zaokrouhlovacích chyb se postupně tlumí, takže výsledek v podstatě neovlivní; *(ii)* zaokrouhlovací chyby se hromadí a vypočítaný výsledek částečně nebo úplně znehodnotí. V prvním případě hovoříme o výpočtu numericky *stabilním*, druhý případ představuje výpočet numericky *nestabilní*. Numerickou stabilitu může někdy ovlivnit výběrem vhodné metody řešení.

Kromě toho jsou výsledky ovlivněny také *chybami lidského faktoru*. Jedná se o chyby v počítačových programech, špatná zadání vstupních dat, nevhodnou volbu matematického modelu nebo nesprávný výběr metody řešení. Tyto chyby lze odstranit, pokud se nám ale podaří zjistit, co je špatně.

V tomto odstavci se budeme zabývat chybami zaokrouhlovacími.

Definice 1.2.1 Nechť $\bar{x}$ je přesná hodnota reálného čísla a x je jeho aproximace. Rozdíl

$$e(x) = \bar{x} - x$$

se nazývá absolutní chyba. Odhad absolutní chyby je číslo $\epsilon(x)$, pro které platí

$$|\bar{x} - x| \leq \epsilon(x). \tag{1.5}$$

Je-li $x \neq 0$, pak číslo

$$r(x) = \frac{\bar{x} - x}{x}$$

se nazývá relativní chyba. Odhad relativní chyby je číslo $\delta(x)$, pro které platí

$$\left| \frac{\bar{x} - x}{x} \right| \leq \delta(x).$$

Relativní chyba a její odhad se často udávají v procentech. Nerovnost (1.5) znamená $\bar{x} \in \langle x - \epsilon(x), x + \epsilon(x) \rangle$, což symbolicky zapisujeme $\bar{x} = x \pm \epsilon(x)$. Pokud nebude hrozit

nedorozumnění, budeme psát e, r, ϵ a δ místo $e(x)$, $r(x)$, $\epsilon(x)$ a $\delta(x)$.

Příklad 1.2.1 Číslo $x = 2.72$ je aproximace Eulerova čísla $\bar{x} = 2.7182818....$ Absolutní chyba je $e = -0.001718...$ a její odhad je například číslo $\epsilon = 0.002$, protože $|e| \leq \epsilon$. Proto $\bar{x} = 2.72 \pm 0.002$. Relativní chyba je $r = -0.00063168...$ a za její odhad můžeme vzít $\delta = 0.00064$, protože $|r| \leq \delta$.

Nyní ukážeme jak se chyby a jejich odhady přenášejí při provádění základních aritmetických operací. Budeme přitom předpokládat, že vykonáváme přesné aritmetické operace s nepřesnými čísly, tj. s aproximacemi, a že známe chyby, respektive jejich odhady.

Nechť

$$\bar{x}_i = x_i + e(x_i), \quad |e(x_i)| \leq \epsilon(x_i), \quad |r(x_i)| \leq \delta(x_i), \quad i = 1, 2.$$

(a) Je-li $u = x_1 + x_2$ *aproximace součtu* $\bar{u} = \bar{x}_1 + \bar{x}_2$, potom

$$\bar{u} = x_1 + e(x_1) + x_2 + e(x_2) = u + e(u),$$

kde

$$e(u) = e(x_1) + e(x_2),$$

a platí

$$|e(u)| \leq |e(x_1)| + |e(x_2)| \leq \epsilon(x_1) + \epsilon(x_2).$$

(b) Je-li $v = x_1 - x_2$ *aproximace rozdílu* $\bar{v} = \bar{x}_1 - \bar{x}_2$, potom

$$e(v) = e(x_1) - e(x_2)$$

a platí

$$|e(v)| \leq |e(x_1)| + |e(x_2)| \leq \epsilon(x_1) + \epsilon(x_2).$$

(c) Je-li $w = x_1 x_2$ *aproximace součinu* $\bar{w} = \bar{x}_1 \bar{x}_2$, potom

$$\bar{w} = (x_1 + e(x_1))(x_2 + e(x_2)) = w + x_1 e(x_2) + x_2 e(x_1) + e(x_1)e(x_2).$$

Odhad absolutní chyby, kterou dostaneme při součinu, určujeme výrazem

$$e(w) = x_1 e(x_2) + x_2 e(x_1).$$

Odtud pak

$$|e(w)| \leq |x_1|\epsilon(x_2) + |x_2|\epsilon(x_1).$$

(d) Je-li $z = x_1 / x_2$ *aproximace podílu* $\bar{z} = \bar{x}_1 / \bar{x}_2$, potom

$$\bar{z} = \frac{x_1 + e(x_1)}{x_2 + e(x_2)} = z + \frac{x_2 e(x_1) - x_1 e(x_2)}{x_2(x_2 + e(x_2))}.$$

Odhad absolutní chyby, kterou dostaneme při podílu, určujeme výrazem

$$e(z) = \frac{x_2 e(x_1) - x_1 e(x_2)}{x_2^2}.$$

Odtud platí

$$|e(z)| \leq \frac{|x_2|\epsilon(x_1) + |x_1|\epsilon(x_2)}{|x_2|^2}.$$

Pro relativní chyby můžeme z pravidel (a), (b), (c) a (d) odvodit:

(A) $$r(u) = \frac{1}{x_1+x_2}\left(x_1\frac{e(x_1)}{x_1} + x_2\frac{e(x_2)}{x_2}\right) = \frac{1}{x_1+x_2}\left(x_1 r(x_1) + x_2 r(x_2)\right),$$

$$|r(u)| \le \frac{1}{|x_1+x_2|}\left(|x_1|\delta(x_1) + |x_2|\delta(x_2)\right),$$

(B) $$r(v) = \frac{1}{x_1-x_2}\left(x_1\frac{e(x_1)}{x_1} - x_2\frac{e(x_2)}{x_2}\right) = \frac{1}{x_1-x_2}\left(x_1 r(x_1) - x_2 r(x_2)\right),$$

$$|r(v)| \le \frac{1}{|x_1-x_2|}\left(|x_1|\delta(x_1) + |x_2|\delta(x_2)\right),$$

(C) $$r(w) = \frac{x_1 e(x_2) + x_2 e(x_1)}{x_1 x_2} = r(x_2) + r(x_1),$$

$$|r(w)| \le \delta(x_2) + \delta(x_1),$$

(D) $$r(z) = \frac{x_2 e(x_1) - x_1 e(x_2)}{x_2^2}\cdot\frac{x_2}{x_1} = r(x_1) - r(x_2),$$

$$|r(z)| \le \delta(x_1) + \delta(x_2).$$

Poznámka

Při odčítání blízkých čísel (pravidlo (B)) má na velikost relativní chyby rozhodující vliv zlomek $1/(x_1-x_2)$, který ukazuje, že dochází ke ztrátě relativní přesnosti.

Příklad 1.2.2 Nechť $\bar{x}_1 = 758\,320$, $x_1 = 758\,330$, $\bar{x}_2 = 757\,940$ a $x_2 = 757\,930$. Určete ztrátu relativní přesnosti při odčítání.

Řešení: Protože $e(x_1) = -10, e(x_2) = 10$, můžeme položit

$$\left|\frac{-10}{758330}\right| \doteq 1.32\cdot 10^{-5} = \delta(x_1), \quad \left|\frac{10}{757930}\right| \doteq 1.32\cdot 10^{-5} = \delta(x_2).$$

Dále je $\bar{v} = \bar{x}_1 - \bar{x}_2 = 380$ a $v = x_1 - x_2 = 400$, a proto

$$\left|\frac{\bar{v}-v}{v}\right| = \left|\frac{-20}{400}\right| \doteq 5\cdot 10^{-2} = \delta(v).$$

Došlo ke ztrátě relativní přesnosti zhruba o tři řády. Podle pravidla (B) můžeme psát

$$|r(v)| \le \frac{758330}{400}(\delta(x_1)+\delta(x_2)) \le 1.9\cdot 10^3\cdot(\delta(x_1)+\delta(x_2)),$$

což naše zjištění potvrzuje. □

Nyní uvedeme některé pojmy, které souvisí s ukládáním a zaokrouhlováním čísel na počítači. Budeme vycházet ze zápisu reálného čísla v tzv. *pohyblivé řádové čárce*:

$$\textit{znaménko} \times \textit{mantisa} \times \textit{exponenciální část}.$$

Symbolem $\times$ zde označujeme násobení čísel. Při zápisu čísla musíme použít určitou číselnou soustavou. Tou je nejčastěji soustava *desítková* (decimální), *dvojková* (binární) a *šestnáctková* (hexadecimální). V paměti dnešních běžných počítačů se čísla ukládají téměř vždy v soustavě dvojkové, z čehož jsou odvozeny hodnoty významných počítačových konstant. Pro snadnější představu si všechny pojmy vysvětlíme v soustavě desítkové.

Například číslo -39.151 se zapíše jako $(-1) \times 0.39151 \times 10^{(+1)\times 2}$. Ve skutečnosti ještě konec mantisy a začátek exponentu obsahuje několik nul podle toho, jak velký paměťový prostor je pro uložení jednoho čísla k dispozici. Obecně vypadá zápis reálného čísla x takto:

$$x = (\pm 1) \times 0.x_1 x_2 \ldots x_t \times 10^b, \tag{1.6}$$

kde t je pevně zvolený počet číslic x_i v mantise a b je celočíselný exponent v určitých daných mezích, $b_{\min} \leq b \leq b_{\max}$. Aby byl zápis každého čísla jednoznačný, používá se *normalizovaná* mantisa, u níž je $x_1 \neq 0$. Nejmenší kladné reálné číslo, které lze takto vyjádřit, je $x_{\min} = 10^{b_{\min}-1}$. Hodnota $x_{\min}$ je mez pro *podtečení* (underflow) – při pokusu zapsat menší kladné číslo se uloží nula. Nula se ale reprezentuje jinak, protože normalizovanou mantisou ji nelze vyjádřit. Největší číslo, které lze zapsat pomocí (1.6), je $(1 - 10^{-t}) \times 10^{b_{\max}}$. Protože se $1 - 10^{-t}$ liší od 1 jen velmi nepatrně, nastavujeme mez pro *přetečení* (overflow) jako $x_{\max} = 10^{b_{\max}}$. Pokud dojde v MATLABu k pokusu zapsat číslo, které je větší než $x_{\max}$, uloží se hodnota `Inf`.

Všimněme si jaké největší chyby se můžeme dopustit, když pomocí (1.6) budeme zapisovat číslo $\bar{x}$ z intervalu $\langle x_{\min}, x_{\max}\rangle$. Obecně může mít takové číslo libovolně dlouhou mantisu:

$$\bar{x} = (\pm 1) \times 0.x_1 x_2 \ldots x_t x_{t+1} \ldots \times 10^b. \tag{1.7}$$

Odečtením (1.6) a (1.7) dostaneme odhad absolutní chyby:

$$|\bar{x} - x| = 0.x_{t+1} \ldots \times 10^{b-t} \leq 0.\bar{9} \times 10^{b-t} = 10^{b-t}.$$

Odtud lze vyčíst, že čísla ve tvaru (1.6) nejsou na intervalu $\langle x_{\min}, x_{\max}\rangle$ rozložena rovnoměrně, ale že jejich „výskyt je hustší“ při menších hodnotách. Zabývejme se dále relativní chybou. Snadno zjistíme, že

$$\left|\frac{\bar{x} - x}{x}\right| \leq \frac{10^{b-t}}{0.x_1 x_2 \ldots x_t \times 10^b} = \frac{10^{-t}}{0.x_1 x_2 \ldots x_t} \leq \frac{10^{-t}}{0.1} = 10^{1-t}.$$

Získali jsme odhad relativní chyby nezávislý na hodnotě čísla $\bar{x}$. Ten určujeme tzv. *počítačové epsilon*. Jestliže při zápisu čísla $\bar{x}$ do tvaru (1.6) budeme na číslici x_t zaokrouhlovat obvyklým způsobem, dostaneme odhad poloviční.

V následující definici přejdeme k soustavě s libovolným základem z tak, že v našich odhadech jednoduše změníme desítku na z .

Definice 1.2.2 Nechť z je přirozené číslo, které udává základ číselné soustavy. Počítačové epsilon je konstanta

$$\varepsilon(z,t) = \frac{1}{2} \times z^{1-t},$$

kde t je počet cifer normalizované mantisy. Konstanty, které určují meze pro podtečení a přetečení, mají tvar:

$$x_{\min}(z, b_{\min}) = z^{b_{\min}-1} \quad \text{a} \quad x_{\max}(z, b_{\max}) = z^{b_{\max}},$$

kde $b_{\min}$ a $b_{\max}$ jsou celočíselné hodnoty pro nejmenší a největší exponent.

Výpočty se v MATLABu provádějí standardně ve *dvojnásobné přesnosti* (double precision), kdy je pro uložení jednoho reálného čísla vyhrazeno 64 bitů (binary digits). Jeden bit se využívá pro znaménko, 52 bitů pro normalizovanou mantisu (tj. $t = 52$) a 11 bitů pro exponent (z toho si jeden bit vyžádá znaménko exponentu). Protože pro dvojkovou soustavu je $z = 2$, dostáváme:

$$\begin{aligned} \varepsilon(2,52) &= \frac{1}{2} \times 2^{-51} = 2.220446049250313 \times 10^{-16}, \\ x_{\min}(2,-2^{10}) &= 2^{-1025} = 2.781342323134002 \times 10^{-309}, \\ x_{\max}(2,2^{10}) &= 2^{1024} = 1.797693134862316 \times 10^{308}. \end{aligned}$$

Příklad 1.2.3 Prověřte v MATLABu hodnoty vypočítaných konstant.

Řešení: MATLAB má tyto hodnoty uloženy ve standardních proměnných `eps`, `realmin` a `realmax`:

```
eps = 2.220446049250313e-016
realmin = 2.225073858507201e-308
realmax = 1.797693134862316e+308
```

U hodnot `eps` a `realmax` pozorujeme dokonalou shodu. Hodnota `realmin` odpovídá 2^{-1022}, což je o něco větší číslo, než vyšlo nám. Způsobeno je to tím, že v našem výkladu jsme nešli do úplných detailů, např. jsme neřekli co s nulou nebo jak vypadá technika vykonávání základních aritmetických operace, která vše také trochu ovlivní; viz popis této problematiky v [9] str. 45-56. Všimněme si ještě něčeho dalšího. Když se pokusíme do nějaké proměnné přiřadit hodnotu konstanty `realmax`, dojde již k přetečení:

```
>> number = 1.797693134862316e+308
   number = Inf
```

Naopak trochu menší hodnoty než je `realmin` ještě zobrazit lze, protože se přitom využívá mechanismus pracující s nenormalizovanou mantisou:

```
>> number = 2.225073858507201*10^(-308-15)
   number = 1.976262583364986e-323
>> number = 2.225073858507201*10^(-308-16)
   number = 0
```

Poznámka

Reálná čísla lze v MATLABu ukládat také v *jednoduché přesnosti* (single precision), nebo speciálním způsobem jako čísla celá (integer). Šetří se tím paměť. Podívejte se do nápovědy na heslo `datatypes`. Způsob zápisu reálných čísel v tzv. *pevné řádové čárce*, o němž se lze dočíst ve některých starších textech, je již historický relikt.

V následujícím příkladu si ukážeme řešení úlohy, která je při jednom způsobu výpočtu numericky nestabilní, zatímco při druhém je numericky stabilní. Předpokládejme, že je našim úkol vypočítat hodnoty integrálů

$$y_i = \int_0^1 \frac{x^i}{x+5}\,dx \quad \text{pro} \quad i = 0,1,\ldots,8. \tag{1.8}$$

Nejdříve pomocí úpravy

$$y_i + 5y_{i-1} = \int_0^1 \frac{x^i + 5x^{i-1}}{x+5}\,dx = \int_0^1 x^{i-1}\frac{x+5}{x+5}\,dx = \int_0^1 x^{i-1}\,dx = \frac{1}{i}$$

odvodíme rekurentní vzorec

$$y_i = \frac{1}{i} - 5y_{i-1} \quad \text{pro} \quad i = 1,2,\ldots,8. \tag{1.9}$$

Při výpočtu budeme postupovat tak, že integrací vypočítáme y_0 a hodnoty dalších integrálů získáme pomocí (1.9). Zaokrouhlovat budeme na tři desetinná místa. Dostáváme:

$$\begin{aligned}
y_0 &= \int_0^1 \frac{1}{x+5}\,dx = [\ln(x+5)]_0^1 = 0.18232\ldots \doteq 0.182\\
y_1 &= 1 - 5y_0 = 1 - 5\cdot 0.182 = 0.090,\\
y_2 &= \frac{1}{2} - 5y_1 = \frac{1}{2} - 5\cdot 0.090 = 0.050,\\
y_3 &= \frac{1}{3} - 5y_2 = \frac{1}{3} - 5\cdot 0.050 \doteq 0.083,\\
y_4 &= \frac{1}{4} - 5y_3 = \frac{1}{4} - 5\cdot 0.083 = -0.165.
\end{aligned}$$

Takto vypočítaná hodnota y_4 je určitě nesprávná, protože všechny všechny integrály musí vyjít kladně. Správným výsledkem je $y_4 = 0.03427\ldots$. Výpočet podle vzorce (1.9) je numericky nestabilní.

Nestability se zbavíme vhodnější organizací výpočtu. Rekurentní vzorec (1.9) zapíšeme pro výpočet v opačném směru:

$$y_{i-1} = \frac{1}{5i} - \frac{1}{5}y_i \quad \text{pro} \quad i = 9,8,\ldots,1. \tag{1.10}$$

Startovací hodnotu y_9 vypočítáme z přibližné rovnosti $y_9 \doteq y_{10} = \frac{1}{50} - \frac{1}{5}y_9$, odkud vyjde $y_9 \doteq 0.017$. Rekurentním vzorcem (1.10) pak dostáváme:

$$\begin{aligned}
y_8 &= \frac{1}{45} - \frac{1}{5}y_9 = \frac{1}{45} - \frac{1}{5}\cdot 0.017 \doteq 0.019,\\
y_7 &= \frac{1}{40} - \frac{1}{5}y_8 = \frac{1}{40} - \frac{1}{5}\cdot 0.019 \doteq 0.021,\\
&\vdots\\
y_0 &= \frac{1}{5} - \frac{1}{5}y_1 = \frac{1}{5} - \frac{1}{5}\cdot 0.088 \doteq 0.182.
\end{aligned}$$

Hodnota y_0 je přesná (na tři desetinná místa), takže v tomto případě byl výpočet numericky stabilní.

Následující definicí zavádíme pojem, kterým lze numerickou stabilitu i nestabilitu posuzovat.

Definice 1.2.3 Uvažujme úlohu $\bar{y} = U(\bar{x})$, která k přesné vstupní hodnotě $\bar{x}$ vypočítá přesný výsledek $\bar{y}$. Nechť x je porušená vstupní hodnota a y je odpovídající výsledek, tj. $y = U(x)$. *Číslem podmíněnosti* úlohy U nazýváme kladné číslo C_U, pro které platí

$$|r(y)| = C_U |r(x)|,$$

kde $r(x)$ a $r(y)$ jsou relativní chyby vstupní a výstupní hodnoty.

Číslo podmíněnosti vyjadřuje citlivost úlohy na poruchu ve vstupních datech. Je-li $C_U \approx 1$, říkáme, že úloha U je *dobře podmíněná*. Je-li C_U velké, říkáme, že úloha U je *špatně podmíněná*. Pokud umíme určit jenom odhady relativních chyb, stanovíme číslo podmíněnosti přibližně:

$$C_U \approx \frac{\delta(y)}{\delta(x)}.$$

Podle čísla podmíněnosti můžeme posuzovat také citlivost úlohy na zaokrouhlovací chyby, které můžeme interpretovat jako důsledek (teoretické) počáteční poruchy. Při počítačových výpočtech lze za kritickou hodnotu čísla podmíněnosti označit převrácenou hodnotu počítačového epsilon. Pokud se číslo podmíněnosti nějaké úlohy přiblíží k této hodnotě, lze očekávat, že se nepodaří vypočítat smysluplné řešení.

Příklad 1.2.4 Určete číslo podmíněnosti úlohy U vypočítat hodnotu y_4 podle vzorců (1.9) při zaokrouhlování na tři desetinná místa.

Řešení: Dostáváme

$$\begin{aligned} r(y_0) &= \frac{0.18232\ldots - 0.182}{0.182} \doteq 0.001758, \\ r(y_4) &= \frac{0.03427\ldots + 0.165}{-0.165} \doteq -1.207, \\ C_U &= \frac{|r(y_4)|}{|r(y_0)|} \doteq 686.6. \end{aligned}$$

Všimněme si, že při výpočtu podle vzorce (1.9) se hodnota y_{i-1} násobí pěti, čímž dojde také k pětinásobnému zvětšení chyby. Vstupní porucha se v hodnotě y_4 projeví vynásobena číslem $5^4 = 625$, což zhruba odpovídá vypočítanému číslu podmíněnosti.

Druhým příkladem numericky nestabilního výpočtu bude řešení soustavy lineárních rovnic. Uvažujme nejdříve následující soustavu:

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

Snadno ověříme, že přesným řešením je vektor $\mathbf{x} = (1, 1)^\top$. Při výpočtu řešení v MATLABu dostanem:

```
>> A=[1,0.99;0.99,0.98];
>> b=[1.99;1.97];
>> x=A\b
   x =  1.000000000000000e+00
        9.999999999999999e-01
```

Výsledek je stejný až na formu zápisu výsledných hodnot. Zkusme ještě vyřešit soustavu lineárních rovnic s nepatrně pozměněnou pravou stranou:

$$\begin{aligned} x_1 + 0.99x_2 &= 1.9999, \\ 0.99x_1 + 0.98x_2 &= 1.9701. \end{aligned}$$

Výpočtem v MATLABu dostáváme:

```
>> A=[1,0.99;0.99,0.98];
>> b=[1.9999;1.9701];
>> x=A\b
   x =  -9.503000000001016e+01
         9.801000000001027e+01
```

Ačkoliv vypočítaný výsledek vypadá podezřele, přesné řešení jsme schopni rozpoznat. Je jím vektor $\mathbf{x} = (-95.03, 98.01)^\top$. Ten je ale zcela jiný než v předchozím případě, přestože změna v pravých stranách byla velmi malá a odpovídá zaokrouhlování, které se v takových situacích běžně provádí. Další nebezpečí se nachází na opačném konci mantis, kde se začínají výrazněji projevovat zaokrouhlovací chyby. Nestabilita zde souvisí s maticí soustavy, která je blízká matici singulární – má "téměř"lineárně závislé řádky. Lze snadno zkonstruovat úlohu, u níž se zaokrouhlovací chyby zvětší natolik, že bude problém s rozpoznáním správného výsledku (viz úloha níže).

Kontrolní otázky

Otázka 1. Jak se definuje absolutní a relativní chyba a jejich odhady?
Otázka 2. Jak se chovají chyby při provádění aritmetických operací?
Otázka 3. Co rozumíme pod pojmy počítačové epsilon, přetečení a podtečení a s jakými číselnými hodnotami souvisí?
Otázka 4. Jak se definuje číslo podmíněnosti úlohy a jaká je jeho kritická hodnota na počítači?

Úlohy k samostatnému řešení

1. Pro aproximaci $x = 3.14$ Ludolfova čísla $\bar{x} = 3.1415926...$ určete absolutní a relativní chybu a jejich odhady.
2. Pro data z Příkladu 1.2.2 určete relativní chyby při sčítání, odčítání a dělení.
3. Určete číslo podmíněnosti úlohy vypočítat y_0 podle vzorců (1.10).
4. Řešte v MATLABu následující soustavu lineárních rovnic a pokuste se určit její přesné řešení:

$$\begin{aligned} x_1 + 0.9999x_2 &= 1.999999, \\ 0.9999x_1 + 0.9998x_2 &= 1.999701. \end{aligned}$$

Výsledky úloh k samostatnému řešení

1. $e(x) = 0.0015926...$, $\epsilon(x) = 0.0016$, $r(x) = 0.000507197$, $\delta(x) = 0.00051$.
2. Pro $u = x_1 + x_2$ je $e(u) = 0$, $|e(u)| \leq 20$, $r(u) = 0$, $|r(u)| \leq 1.32 \cdot 10^{-5}$; pro $w = x_1 x_2$ je $e(w) = 3900$, $|e(w)| \leq 15162600$, $r(w) = 6.79 \cdot 10^{-9}$, $|r(w)| \leq 2.64 \cdot 10^{-5}$; pro $z = x_1 / x_2$ je $e(z) = -2.64 \cdot 10^{-5}$, $|e(z)| \leq 2.64 \cdot 10^{-5}$, $r(z) = -2.64 \cdot 10^{-5}$, $|r(z)| \leq 2.64 \cdot 10^{-5}$.
3. $y_9 = 0.0169264...$, $r(y_9) = (y_9 - 0.017)/0.017 \doteq -0.004329$,
$r(y_0) = (0.18232 - 0.1824)/0.1824 \doteq -4.01758$, $C_U = |r(y_0)|/|r(y_9)| \doteq 0.4061$.
4. Řešením vypočítaným v MATLABu je $\mathbf{x} = [-9797.029950778411, 9800.009950773489]^\top$.

KAPITOLA

2

ŘEŠENÍ NELINEÁRNÍCH ROVNIC

Budeme se zabývat numerickým řešením nelineárních rovnic:

$$f(x) = 0, \tag{2.1}$$

kde f je „rozumná“ funkce. Řešením je *kořen* $\bar{x}$, který po dosazení do funkce f dává nulu. O funkci je potřeba něco předpokládat. Například proto, aby vůbec nějaký kořen měla nebo aby bylo možno použít určitou metodu atp. Tyto předpoklady budeme uvádět postupně. Většina metod se bude týkat reálné funkce definované na nějakém intervalu, tj. $f : \langle a, b\rangle \mapsto \mathbb{R}$. Často je úloha postavena tak, že musíme určit i interval. Při zobecnění na soustavy nelineárních rovnic budeme používat vektorový zápis, kdy vektory a matice budeme značit tučně .

U některých rovnic (kvadratické, goniometrické atp.) se kořeny dají vypočítat pomocí vzorců. Stačí ale malá změna funkce a žádné vzorce nelze odvodit. Naproti tomu numerické metody lze použít téměř vždy. Jedná se o *metody iterační*, které vytvářejí posloupnost aproximací $\{x^k\}$ konvergující ke kořenu $\bar{x}$ v limitě:

$$\lim_{k\to\infty} x^k = \bar{x}. \tag{2.2}$$

„Limitování do nekonečna“ samozřejmě není možné a není ani potřeba. Kořen můžeme určit s dostatečně velkou přesností třeba jen po čtyřech iteracích.

U každé iterační metody musíme ošetřit její začátek a konec. Výpočet zahajujeme zadáním *počáteční aproximace* x^0. Čím blíže bude x^0 ke kořenu $\bar{x}$, tím méně iterací bude zpravidla zapotřebí. Pro ukončení výpočtu musíme určit, jak přesně aproximuje poslední x^k hodnotu kořene $\bar{x}$. K tomu používáme vhodně zvolené *ukončovací kritérium*. To by mělo zaručit, že absolutní nebo relativní chyba mezi x^k a $\bar{x}$ je nejvýše rovna zadané toleranci $\varepsilon > 0$. Protože posuzujeme vztah mezi známou hodnotou x^k a neznámou hodnotou $\bar{x}$, jsou ukončovací kritéria často jen *heuristickým* odhadem chyby, který může být v některých situacích nepřesný.

Kvalitu iterační metody posuzujeme podle rychlosti výpočtu. Většinou se sleduje jen počet iterací. To je ale nesprávné, pokud porovnávané metody mají výrazně odlišné výpočetními nároky v jednotlivých iteračních krocích. V takovém případě je rozumnější sledovat počet aritmetických operací nebo přímo výpočetní čas.

2.1 Určení počáteční aproximace

U úloh z praxe lze někdy počáteční aproximaci určit z kontextu. Jestliže například řešením rovnice máme vypočítat koncentraci nějakého roztoku, která je obvykle okolo 15%, pak tuto hodnotu zvolíme za počáteční aproximaci. Není-li taková informace k dispozici, musíme provést rozbor dané funkce. Výsledkem rozboru by měla být separace kořenů do dostatečně krátkých intervalů, z nichž pak počáteční aproximaci volíme. Ukážeme si několik obecných návodů, jak postupovat u funkcí jedné reálné proměnné. U vektorových funkcí je situace složitější, protože již u dvou rovnic pro dvě neznámé je grafem funkce množina bodů ve čtyřrozměrném prostoru, takže graf nelze jednoduše znázornit.

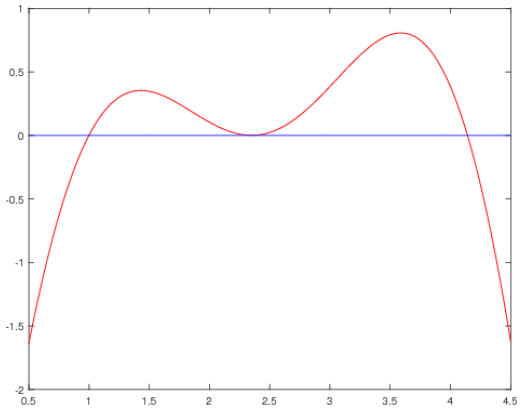
2.1.1 Graf funkce

Kreslení grafu je u funkcí jedné proměnné nejjednodušší způsob hledání kořenů. Musíme ale vědět, na jakém intervalu máme graf znázornit. Kořeny určíme jako průsečíky grafu s x-ovou osou.

Příklad 2.1.1 Nakreslete graf funkce $f(x) = (x-2.35)^2 \sin(x-1)$, zajímají-li nás kořeny z intervalu $\langle 0.5, 4.5\rangle$. Zjistěte kolik jich je a určete co nejpřesněji jejich hodnoty.

Řešení: Při kreslení v MATLABu postupujeme takto:

```
>> x=0.5:0.01:4.5;
>> y=(x-2.35).^2.*sin(x-1);
>> plot(x,y,'r-',x,0*y,'b-')
```



Obrázek 2.1: Graf funkce $f(x) = (x-2.35)^2 \sin(x-1)$ (červeně) a x-ová osa (modře).

Z grafu na Obrázku 2.1 určíme odhadem tři kořeny: $\bar{x}_1 \in \langle 0.5, 1.5\rangle$, $\bar{x}_2 \in \langle 2, 2.5\rangle$ a $\bar{x}_3 \in \langle 4, 4.5\rangle$. Pomocí funkce `Zoom in`, jejíž tlačítko se nachází v grafickém okně MATLABu, lze postupným zvětšováním odečítat hodnoty jednotlivých kořenů přesněji. Vše můžeme snadno kontrolovat, protože z funkčního předpisu lze určit, že přesné hodnoty kořenů jsou $\bar{x}_1 = 1$, $\bar{x}_2 = 2.35$, $\bar{x}_3 = \pi + 1$ a také to, že kořen $\bar{x}_2$ je dvojnásobný (je i kořenem první derivace $f'(x) = 2(x-2.35)\sin(x-1) + (x-2.35)^2\cos(x-1)$). □

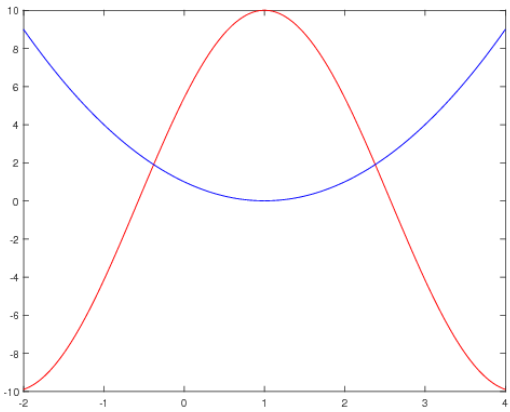
2.1.2 Rozklad funkce

Nechť $f(x) = h(x) - g(x)$. Rozkladem funkce rozumíme převod rovnice $f(x) = 0$ na ekvivalentní rovnici $h(x) = g(x)$. Na vhodném intervalu nakreslíme grafy funkcí g a h a určíme jejich průsečíky. Kořeny funkce f jsou x-ové souřadnice těchto průsečíků.

Příklad 2.1.2 Pomocí rozkladu funkce $f(x) = 10\cos(x-1) - x^2 + 2x - 1$ určete polohu jejich kořenů v intervalech délky nejvýše 0.1.

Řešení: Rovnici $f(x) = 0$ přepíšeme do tvaru $10\cos(x-1) = x^2 - 2x + 1$. Promyslíme-li si, jak vypadají grafy funkcí $h(x) = 10\cos(x-1)$ a $g(x) = x^2 - 2x + 1$, podaří se nám snadno zjistit, že kreslení je vhodné provést na intervalu $\langle -2, 4\rangle$:

```
>> x=-2:0.01:4;
>> hx=10.*cos(x-1); gx=x.^2-2*x+1;
>> plot(x,hx,'r-',x,gx,'b-')
```



Obrázek 2.2: Červeně $h(x) = 10\cos(x-1)$, modře $g(x) = x^2 - 2x + 1$.

Dva průsečíky na Obrázku 2.2 odpovídají kořenům $\bar{x}_1 \in \langle -1, 0\rangle$ a $\bar{x}_2 \in \langle 2, 3\rangle$. Zvětšováním pomocí `Zoom in` můžeme zjistit, že $\bar{x}_1 \in \langle -0.4, -0.3\rangle$ a $\bar{x}_2 \in \langle 2.3, 2.4\rangle$. □

2.1.3 Tabelace funkce

Tabelací funkce rozumíme výpočet tabulky funkčních hodnot. Předpokládáme, že daná funkce je spojitá. Výskyt kořene poznáme podle znaménkové změny u dvou sousedních funkčních hodnot nebo někdy přímo podle nulové funkční hodnoty.

Příklad 2.1.3 Tabelací funkce z Příkladu 2.1.1 určete intervaly délky nejvýše 0.1, které obsahují vždy jeden kořen.

Řešení: Při výpočtu v MATLABu postupujeme takto:

```
>> x=0.5:0.1:4.5;
>> fx=(x-2.35).^2.*sin(x-1);
>> tabulka=[x' fx']
```

Protože celá tabulka je dlouhá, zapíšeme do Tabulky 2.1 jen její zajímavé úseky. Jeden kořen určíme přesně: $\bar{x}_1 = 1$. Dvojnásobný kořen $\bar{x}_2$ ($= 2.35$) z tabulky nepoznáme. Podle znaménkové změny můžeme stanovit interval pro třetí kořen: $\bar{x}_3 \in \langle 4.1, 4.2 \rangle$. □

x	$f(x)$
...	...
0.9000	−0.2099
1.0000	0
1.1000	0.1560
...	...
2.3000	0.0024
2.4000	0.0025
...	...
4.1000	0.1273
4.2000	−0.1998
...	...

Tabulka 2.1: Tabelace funkce $f(x) = (x - 2.35)^2 \sin(x - 1)$.

Tento postup určování polohy kořenů se opírá o větu, ve které se pravidlo o znaménkové změně zapisuje jako záporný součin funkčních hodnot.

Věta 2.1.1 Nechť $f : \langle a, b \rangle \mapsto \mathbb{R}$ je funkce spojitá na intervalu $\langle a, b \rangle$ a platí

$$f(a)f(b) < 0. \tag{2.3}$$

Pak uvnitř intervalu (a, b) leží aspoň jeden kořen rovnice $f(x) = 0$.

Kontrolní otázky

Otázka 1. Co rozumíme pojmem separace kořenů funkce?
Otázka 2. Načrtněte aspoň dva podstatně odlišné obrázky znázorňující tvrzení Věty 2.1.1?
Otázka 3. Může interval (a, b) z Věty 2.1.1 obsahovat právě dva kořeny?

Úlohy k samostatnému řešení

1. Proveďte separaci kořenů rovnice: $x^2 - x - \frac{6}{7} \ln x = 0$.

Výsledky úloh k samostatnému řešení

1. Dva kořeny: $\bar{x}_1 \in (0.9, 0.91)$, $\bar{x}_2 = 1$.

2.2 Zkracování intervalu

Nejjednodušší metody pro řešením rovnice (2.1), kde $f : \langle a, b \rangle \mapsto \mathbb{R}$, jsou založeny na postupném zkracování intervalu $\langle a, b \rangle$. Opírají se o Větu 2.1.1, takže vyžadují pouze spojitost

funkce f a dodržování pravidla o znaménkové změně. Obsahuje-li výchozí interval více než jeden kořen, konvergují tyto metody k některému z nich.

Princip zkracování intervalu vysvětlíme nejprve obecně. Předpokládejme, že f je spojitá funkce na intervalu $\langle a^0, b^0\rangle \subseteq \langle a, b\rangle$ a platí $f(a^0)f(b^0) < 0$. Zvolíme bod $x^1 \in (a^0, b^0)$, který rozdělí výchozí interval na dvě části. Jako nový interval $\langle a^1, b^1\rangle$ vezmeme tu část původního intervalu, u níž zjistíme znaménkovou změnu funkčních hodnot v krajních bodech. Může se také stát, že bod x^1 bude kořenem. Rozhodujeme se proto takto:

- je-li $f(x^1) = 0$, potom $\bar{x} = x^1$ a výpočet ukončíme;
- je-li $f(a^0)f(x^1) < 0$, položíme $\langle a^1, b^1\rangle = \langle a^0, x^1\rangle$;
- je-li $f(x^1)f(b^0) < 0$, položíme $\langle a^1, b^1\rangle = \langle x^1, b^0\rangle$.

Pokud nenastane první případ, zopakujeme vše na intervalu $\langle a^1, b^1\rangle$, tj. zvolíme bod $x^2 \in (a^1, b^1)$, který může být kořenem, nebo pomocí něho určíme další interval $\langle a^2, b^2\rangle$ stejným způsobem. Pokračujeme-li tímto způsobem dál, může se nám podařit najít kořen $\bar{x}$ po konečném počtu iterací. To však nastane jen velmi zřídka. Většinou vytváříme posloupnosti $\{a^k\}$, $\{b^k\}$ a $\{x^k\}$ takové, že uvnitř každého intervalu (a^k, b^k) leží kořen $\bar{x}$. Abychom měli zaručeno, že posloupnost $\{x^k\}$ k němu konverguje (platí (2.2)), musíme volit body x^k vhodným způsobem.

2.2.1 Metoda půlení intervalu

Aproximaci kořene x^k, $k \geq 0$, vypočítáme jako střed intervalu $\langle a^k, b^k\rangle$ podle vzorce:

$$x^k = \frac{a^k + b^k}{2}. \tag{2.4}$$

Protože vytvářené intervaly se postupně půlí, konvergují jejich délky k nule. Posloupnost středů $\{x^k\}$ proto nutně konverguje ke kořenu $\bar{x}$.

Jako ukončovací kritérium použijeme nerovnost

$$\frac{b^k - a^k}{2} \leq \epsilon, \tag{2.5}$$

kde $\epsilon > 0$ je požadavek na přesnost. Protože kořen $\bar{x}$ leží uvnitř intervalu $\langle a^k, b^k\rangle$, liší se od středu x^k ne více, než je polovina délky intervalu. Při splnění (2.5) platí:

$$|\bar{x} - x^k| \leq \frac{b^k - a^k}{2} \leq \epsilon.$$

Hodnota x^k je tedy aproximací kořene $\bar{x}$ vyhovující odhadu absolutní chyby (1.5), takže můžeme psát: $\bar{x} = x^k \pm \epsilon$. Celý algoritmus zapíšeme takto:

Algoritmus: Půlení intervalu
Vstup: f, a^0, b^0, ϵ.
Pro $k = 0, 1, \ldots$ opakuj:
  $x^k := (a^k + b^k)/2$;
  je-li $f(x^k) = 0$, potom jdi na Výstup;
  je-li $f(a^k)f(x^k) < 0$, potom $a^{k+1} := a^k$, $b^{k+1} := x^k$;
  je-li $f(x^k)f(b^k) < 0$, potom $a^{k+1} := x^k$, $b^{k+1} := b^k$;
dokud $(b^k - a^k)/2 > \epsilon$.
Výstup: $\bar{x} = x^k \pm \epsilon$.

Příklad 2.2.1 Metodou půlení intervalu řešte rovnici:

$$10\cos(x-1) - x^2 + 2x - 1 = 0.$$

Vypočítejte kořen, který leží v intervalu $\langle 2.3, 2.4\rangle$ s přesností $\epsilon = 10^{-3}$.

Řešení: Na začátku výpočtu položíme $a^0 = 2.3$, $b^0 = 2.4$. Středem prvního intervalu je $x^0 = 2.35$. Tabulka 2.2 ukazuje průběh výpočtu. Symbolem „+“ nebo „−“ za číslem uvádíme znaménko funkční hodnoty v daném bodě. Všimněme si, že x^k nahrazuje a^k nebo b^k tak, aby byla zachována znaménková změna. Aproximace kořene s přesností ϵ je poslední číslo ve sloupci x^k. Výsledek zapíšeme takto: $\bar{x} = 2.378 \pm 10^{-3}$. □

k	a^k	b^k	x^k	$(b^k - a^k)/2$
0	2.3^+	2.4^-	2.35^+	0.05
1	2.35^+	2.4^-	2.375^+	0.025
2	2.375^+	2.4^-	2.3875^-	0.0125
3	2.375^+	2.3875^-	2.38125^-	0.00625
4	2.375^+	2.38125^-	2.378125^+	0.003125
5	2.378125^+	2.38125^-	2.3796875^-	0.0015625
6	2.378125^+	2.3796875^-	2.37890625^+	$0.00078125 < 10^{-3} = \epsilon$

Tabulka 2.2: Výpočet metodou půlení intervalu.

Příklad 2.2.2 Zapište metodu půlení intervalu v MATLABu jako funkci `puleni`. Spusťte ji z příkazové řádky a poté vygenerujte data pro Tabulku 2.2.

Řešení: V programu omezíme počet iterací vstupní proměnnou `maxk`.

```
function [x,k]=puleni(f,a,b,epsilon,maxk)
fa=f(a); fb=f(b)
for k=0:maxk
  x=(a+b)/2; pres=(b-a)/2; fx=f(x);
  if pres<=epsilon, break
  elseif fx==0, break
  elseif fa*fx<0, b=x; fb=fx;
  else a=x; fa=fx;
  end
end
```

Funkci f z příkladu 2.2.1 zadáme jako anonymní funkci (function handle):

```
>> f=@(x) 10*cos(x-1)-x^2 +2*x-1;
```

Spuštětní funkce `puleni` pak provedeme například takto:

```
>> [x,k]=puleni(f,2.3,2.4,1e-3,100)
    x = 2.3789
    k = 7
```

Pokud chceme sestavit Tabulku 2.2, odmažeme před spuštěním funkce `puleni` některé středníky. □

2.2.2 Metoda regula falsi

V intervalu $\langle a^k, b^k \rangle$, $k \geq 0$, vypočítáme aproximaci kořene x^k jako nulový bod přímky p, která prochází krajními body grafu funkce f, viz Obrázek 2.3.a. Přímka p je dána předpisem:

$$p(x) = f(a^k) + \frac{f(b^k) - f(a^k)}{b^k - a^k}(x - a^k)$$

a její nulový bod je určen rovnicí $p(x^k) = 0$. Odtud lze snadno odvodit vzorec:

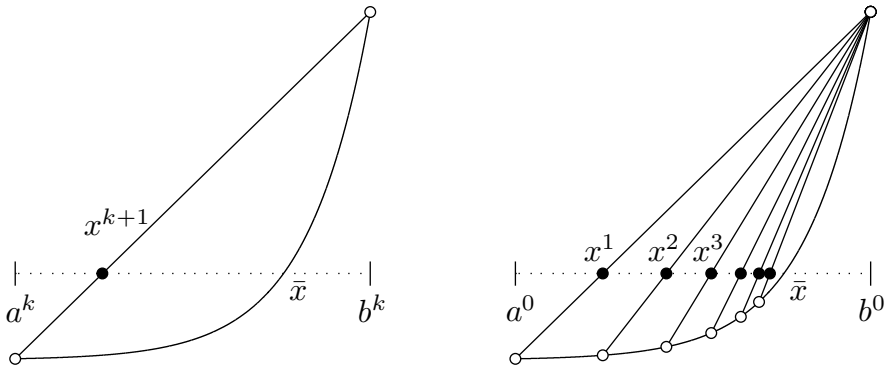
$$x^k = a^k - \frac{b^k - a^k}{f(b^k) - f(a^k)} f(a^k), \tag{2.6}$$

který se použije ve výpočtu.

Geometrický smysl metody regula falsi znázorňuje Obrázek 2.3.b. Ukončení iterací provedeme podle ukončovacího kritéria:

$$|x^k - x^{k-1}| \leq \epsilon, \tag{2.7}$$

kde $\epsilon > 0$ je požadavek na přesnost. Poznamenejme, že (2.7) je kritérium heuristické, protože neobsahuje přímou vazbu na kořen $\bar{x}$, a v některých situacích může výpočet ukončit předčasně. U rychle konvergujících metod však bývá dostatečně přesné, takže v takových případech můžeme psát: $\bar{x} = x^k \pm \epsilon$.



(a) (b)

Obrázek 2.3: (a) Jeden krok metody regula fasi; (b) několik kroků této metody.

V zápisu algoritmu zajistíme funkčnost ukončovacího kritéria (2.7) vhodně zvolenou hodnotou x^{-1}.

Algoritmus: Regula falsi
Vstup: f, a^0, b^0, ϵ, $x^{-1} := a^0$.
Pro $k = 0, 1, \ldots$ opakuj:
  $x^k := a^k - (b^k - a^k)/(f(b^k) - f(a^k))f(a^k)$;
  je-li $f(x^k) = 0$, potom jdi na Výstup;
  je-li $f(a^k)f(x^k) < 0$, potom $a^{k+1} := a^k$, $b^{k+1} := x^k$;
  je-li $f(x^k)f(b^k) < 0$, potom $a^{k+1} := x^k$, $b^{k+1} := b^k$;
dokud $|x^k - x^{k-1}| > \epsilon$.
Výstup: $\bar{x} = x^k \pm \epsilon$.

Příklad 2.2.3 Metodou regula falsi vyřešte rovnici z Příkladu 2.2.1

Řešení: Položíme opět $a^0 = 2.3$, $b^0 = 2.4$ a navíc $x^{-1} = 2.3$. V první iteraci vypočítáme:

$$x^0 := 2.3 - (2.4 - 2.3)f(2.3)/(f(2.4) - f(2.3)) \doteq 2.379095,$$
$$|x^0 - x^{-1}| = |2.379095 - 2.3| = 0.079095.$$

Tabulka 2.3 zachycuje celý výpočet, který se řídí podobnými pravidly jako u metody půlení intervalu. Výsledek zapíšeme $\bar{x} = 2.379 \pm 10^{-3}$, což je správný závěr vzhledem k rychlé konvergenci výpočtu. □

k	a^k	b^k	x^k	$\lvert x^k - x^{k-1}\rvert$
0	2.3^+	2.4^-	2.379095^+	0.079095
1	2.379095^+	2.4^-	2.379363^+	$0.000268 < 10^{-3} = \epsilon$

Tabulka 2.3: Výpočet metodou regula falsi.

Příklad 2.2.4 Zapište algoritmus metody regula falsi v MATLABu jako funkci `regfal`. Spusťte ji z příkazové řádky a poté vygenerujte data pro Tabulku 2.3.

Řešení: Postupujeme podobně jako u metody půlení intervalu.

```
function [x,k]=regfal(f,a,b,epsilon,maxk)
x0=a; fa=f(a); fb=f(b);
for k=0:maxk
  x=a-(b-a)/(f(b)-f(a))*f(a);
  pres=abs(x-x0); fx=f(x);
  if pres<=epsilon, break
  elaseif fx==0, break
  elseif fa*fx<0, b=x; fb=fx;
  else a=x; fa=fx;
  end
  x0=x;
end
```

Zbytek je podobný jako v Příkladu 2.2.2, budeme však muset nastavit dlouhý formát zobrazování čísel příkazem `format long`. □

Kontrolní otázky

Otázka 1. Odhadněte kolik iterací metody půlení intervalu potřebujeme, aby došlo ke zpřesnění o jedno desetinné místo?
Otázka 2. Pokuste se určit situace, kdy bude metoda regula falsi konvergovat rychle a kdy pomalu?
Otázka 3. Podrobně odvoďte vzorec (2.6).
Otázka 4. Proč nelze metodu regula falsi ukončovat podle kritéria (2.5)?

Úlohy k samostatnému řešení

1. Vypočtěte kořeny rovnice $x^2 - x - \frac{6}{7}\ln x = 0$ metodou půlení intervalu pro $\epsilon = 10^{-3}$.
2. Vypočtěte kořeny rovnice z předchozí úlohy metodou regula falsi.

Výsledky úloh k samostatnému řešení

1. Výpočet zahájíme na intervalu $\langle 0.9, 0.91\rangle$, po čtvrté iteraci je $\bar{x} = 0.9019 \pm 10^{-3}$.
2. Výpočet zahájíme na stejném intervalu, po čtvrté iteraci je $\bar{x} = 0.9021 \pm 10^{-3}$.

2.3 Newtonova metoda a její řád

V tomto odstavci odvodíme nejjednodušší variantu Newtonovy metody pro rovnici (2.1) a funkci $f : \langle a, b\rangle \mapsto \mathbb{R}$. Newtonova metoda patří mezi nejpopulárnější iterační metody pro řešení nelineárních rovnic. Kromě funkčních hodnot používá také hodnoty první derivace, takže její konvergence je velmi rychlá. Základní varianta Newtonovy metody konverguje *lokálně*, čímž máme na mysli, že konvergenci lze zaručit volbou počáteční aproximace x^0 „dostatečně blízko" u kořene $\bar{x}$. Stanovíme-li pro funkci f na intervalu $\langle a, b\rangle$ jisté silnější předpoklady, pak konvergence nastane pro každé x^0 z tohoto intervalu. V takovém případě hovoříme o konvergenci *globální*.

Předpokládejme, že známe aproximaci kořene x^{k-1} a chceme určit další (přesnější) aproximaci x^k. Rovnici (2.1) zapíšeme na okolí bodu x^{k-1} pomocí Taylorova rozvoje:

$$f(x^{k-1}) + (x - x^{k-1})f'(x^{k-1}) + (x - x^{k-1})^2 \frac{f''(\xi)}{2} = 0,$$

kde ξ je blíže neurčený bod mezi x a x^{k-1}. Tento tvar rovnice (2.1) *linearizujeme* tak, že vynecháme kvadratický člen na levé straně. Aproximaci x^k určíme z linearizované rovnice:

$$f(x^{k-1}) + (x^k - x^{k-1})f'(x^{k-1}) = 0. \tag{2.8}$$

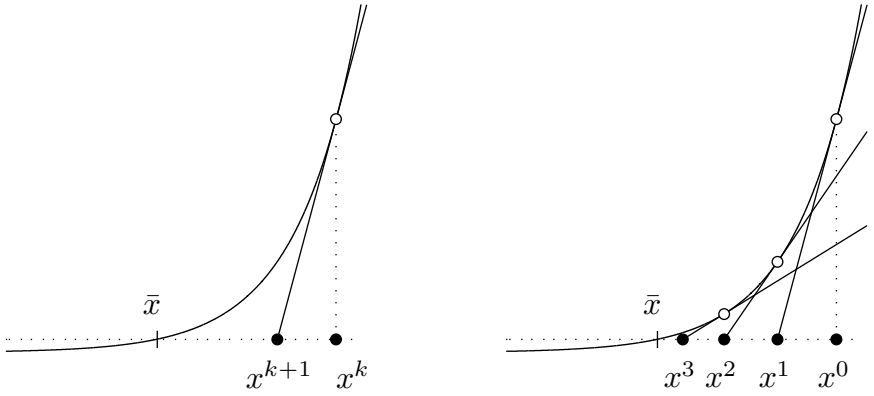
Odtud snadno odvodíme vzorec pro výpočet x^k:

$$x^k = x^{k-1} - \frac{f(x^{k-1})}{f'(x^{k-1})}. \tag{2.9}$$

Bod x^k je nulovým bodem tečny $t(x) = f(x^{k-1}) + (x - x^{k-1})f'(x^{k-1})$ ke grafu funkce f v bodě $(x^{k-1}, f(x^{k-1}))$, takže (2.8) lze zapsat také jako $t(x^k) = 0$, viz Obrázek 2.4.a. Několik po sobě následujících kroků Newtonovy metody je znázorněno na Obrázku 2.4.b.

Algoritmus Newtonovy metody vyžaduje zadat funkci f, její derivaci f', počáteční aproximaci x^0 a specifikovat požadavek na přesnost $\epsilon > 0$. Jako ukončovací kritérium zde volíme opět (2.7).

Algoritmus: Newtonova metoda
Vstup: f, f', x^0, ϵ.
Pro $k = 1, 2, \ldots$ opakuj:
  $x^k := x^{k-1} - f(x^{k-1})/f'(x^{k-1})$;
dokud $|x^k - x^{k-1}| > \epsilon$.
Výstup: $\bar{x} = x^k \pm \epsilon$.



(a) (b)

Obrázek 2.4: (a) Jeden krok Newtonovy metody; (b) několik kroků této metody.

Příklad 2.3.1 Newtonovou metodou vyřešte rovnici z Příkladu 2.2.1 pro $\epsilon = 10^{-6}$.

Řešení: Funkce $f(x) = 10\cos(x-1) - x^2 + 2x - 1$ má derivaci $f'(x) = -10\sin(x-1) - 2x + 2$. Newtonova metoda je dána předpisem:

$$x^k = x^{k-1} - \frac{10\cos(x^{k-1}-1) - (x^{k-1})^2 + 2x^{k-1} - 1}{-10\sin(x^{k-1}-1) - 2x^{k-1} + 2}, \quad k = 1, 2, \ldots$$

Počáteční aproximaci zvolme $x^0 = 2.4$. V první iteraci vypočítáme:

$$x^1 := 2.4 - \frac{10\cos(2.4-1) - 2.4^2 + 2\cdot 2.4 - 1}{-10\sin(2.4-1) - 2\cdot 2.4 + 2} = 2.37942798004,$$

$$|x^1 - x^0| = |2.37942798004 - 2.4| = 0.02057201996.$$

Výpočet zaznamenáváme v Tabulce 2.4. Výsledek zapíšeme jako $\bar{x} = 2.379364 \pm 10^{-6}$. □

k	x^k	$\lvert x^k - x^{k-1}\rvert$
1	2.37942798004	0.02057201996
2	2.37936459485	0.00006338519
3	2.37936459422	$0.00000000062 < 10^{-6} = \epsilon$

Tabulka 2.4: Výpočet Newtonovou metodou.

Příklad 2.3.2 Zapište algoritmus Newtonovy metody v MATLABu jako funkci `newton`. Spusťte ji z příkazové řádky a poté vygenerujte data pro Tabulku 2.4.

Řešení: Postupujeme podobně jako v předchozích příkladech.

```
function [x1,k]=newton(f,df,x0,epsilon,maxk)
for k=1:maxk
  x1=x0-f(x0)/(df(x0));
  pres=abs(x1-x0);
  if pres<=epsilon, break, end
  x0=x1;
end
```

Funkci f a její derivaci zadáme takto:

```
>> f=@(x) 10*cos(x-1)-x^2 +2*x-1;
>> df=@(x) -10*sin(x-1)-2*x +2;
```

Spuštění funkce `puleni` pak provedeme následovně:

```
>> [x,k]=newton(f,df,2.4,1e-6,100)
    x = 2.37936459422203
    k = 3
```

Data pro Tabulku 2.4 zobrazíme, když před spuštěním funkce `newton` smažeme některé středníky. □

V Tabulce 2.4 si můžeme všimnout velmi rychlé konvergence Newtonovy metody. Následující věta ukazuje, že se nejedná o náhodu.

Věta 2.3.1 (*O rychlosti konvergence*) Nechť funkce $f : \langle a,b\rangle \mapsto \mathbb{R}$ má spojitou druhou derivaci f'' a nenulovou první derivaci f' na intervalu $\langle a,b\rangle$. Nechť posloupnost $\{x^k\}$ počítaná podle vzorce (2.9) leží v intervalu $\langle a,b\rangle$ a konverguje k $\bar{x}$. Potom $\bar{x}$ je kořenem rovnice $f(x) = 0$ a platí

$$|\bar{x} - x^k| \leq C|\bar{x} - x^{k-1}|^2, \tag{2.10}$$

kde konstanta $C \geq 0$ nezávisí na k.

Důkaz: Ve vzorci (2.9) provedeme limitní přechod:

$$\bar{x} = \lim_{k\to\infty} x^k = \lim_{k\to\infty} \left(x^{k-1} - \frac{f(x^{k-1})}{f'(x^{k-1})} \right) = \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}.$$

Odtud dostaneme $f(\bar{x}) = 0$, takže $\bar{x}$ je kořenem rovnice (2.1). Protože předpokládáme spojitost f'', můžeme $f(\bar{x}) = 0$ zapsat pomocí Taylorova rozvoje v bodě x^{k-1} jako

$$f(x^{k-1}) + (\bar{x} - x^{k-1})f'(x^{k-1}) + (\bar{x} - x^{k-1})^2 \frac{f''(\xi)}{2} = 0,$$

kde ξ je bod mezi x^{k-1} a $\bar{x}$. Vydělíme-li tuto rovnost $f'(x^{k-1})$ a dosadíme ze vzorce (2.9), dostaneme:

$$\bar{x} - x^k + (\bar{x} - x^{k-1})^2 \frac{f''(\xi)}{2f'(x^{k-1})} = 0.$$

Nakonec označíme $C = \max_{\xi,x\in\langle a,b\rangle} |f''(\xi)/2f'(x)|$ a můžeme psát:

$$|\bar{x} - x^k| = \left| \frac{f''(\xi)}{2f'(x^{k-1})} \right| |\bar{x} - x^{k-1}|^2 \leq C\,|\bar{x} - x^k|^2.$$

□

Rychlost konvergence posloupnosti posuzujeme podle jejího řádu.

Definice 2.3.1 Nechť posloupnost $\{x^k\}$ konverguje k $\bar{x}$. Řekneme, že konvergence je řádu $p \geq 1$, jestliže existuje konstanta $C > 0$ nezávislá na k taková, že platí:

$$|\bar{x} - x^k| \leq C|\bar{x} - x^{k-1}|^p.$$

Ve Větě 2.3.1 jsme dokázali, že **Newtonova metoda je druhého řádu**. Tento teoretický výsledek potvrzuje Tabulka 2.4, kde se vždy mezi jednotlivými iteracemi (zhruba/aspoň) zdvojnásobí počet nul za desetinou tečkou. O metodách z předchozího odstavce (půlení intervalu a regula falsi), lze dokázat, že jsou prvního řádu.

Poznámka

Podrobnější analýzou důkazu Věty 2.3.1 lze ukázat, že Newtonova metoda bude konvergovat, jestliže počáteční aproximaci x^0 zvolíme v nějakém malém okolí $\bar{x}$.

Následující věta obsahuje určitý soubor předpokladů, při jejichž splnění konverguje Newtonova metoda na intervalu $\langle a, b\rangle$ globálně.

Věta 2.3.2 (*O globální konvergenci*) Nechť jsou pro funkci $f : \langle a, b\rangle \mapsto \mathbb{R}$ splněny následující předpoklady:
(i) f' je nenulová na intervalu $\langle a, b\rangle$;
(ii) f'' nemění znaménko v intervalu (a, b);
(iii) platí $f(a)f(b) < 0$;
(iv) platí $|f(a)/f'(a)| < b - a$ a $|f(b)/f'(b)| < b - a$.
Potom posloupnost $\{x^k\}$ počítaná podle vzorce (2.9) konverguje pro libovolnou počáteční aproximaci $x^0 \in \langle a, b\rangle$.

Poslední větu uvádíme bez důkazu, který má technický charakter. Vysvětlíme ale jaké nekonvergentní situace předpoklady *(i)*-*(iv)* vylučují. Předpoklad *(i)* zaručuje, že ve vzorci (2.9) nedojde k pokusu dělit nulou. Tečna je v takové případě rovnoběžná s x-sovou osou a buď to nemá nulový bod, nebo jich má nekonečně mnoho. Předpoklad *(ii)* vylučuje oscilace. Nesplňuj jej například funkce $f(x) = \sin x$ na intervalu $\langle -\pi/4, \pi/4\rangle$, takže kořen rovnice $\sin x = 0$ nelze pomocí základní varianty Newtonovy metody vypočítat.[1] Předpoklad *(iii)* zaručuje existenci kořene $\bar{x}$ podle Věty 2.1.1. Předpoklady *(iv)* zaručují, že posloupnost $\{x^k\}$ počítaná podle vzorce (2.9) bude ležet v intervalu $\langle a, b\rangle$.

Příklad 2.3.3 Ukažte, že pro rovnici z Příkladu 2.3.1 jsou na intervalu $\langle 2.3, 2.4\rangle$ splněny předpoklady Věty 2.3.2

Řešení: Funkce f a její derivace f' jsou uvedeny v Příkladu 2.3.1, druhá derivace má tvar $f''(x) = -10\cos(x-1) - 2$. V Tabulce 2.5 jsou uvedeny hodnoty f', f'', z nichž můžeme usoudit, že jsou splněny předpoklady *(i)* a *(ii)*.

Výpočtem dostáváme $f(2.3)f(2.4) \doteq -0.2564 < 0$, $|f(2.3)/f'(2.3)| \doteq 0.0805 < 0.1$ a $|f(2.4)/f'(2.4)| \doteq 0.0206 < 0.1$, což dokazuje splnění zbývajících předpokladů *(iii)* a *(iv)*. Počáteční aproximaci x^0 je proto možné zvolit na intervalu $\langle 2.3, 2.4\rangle$ libovolně . □

[1] Koře rovnice $\sin x = 0$ lze vypočítat například pomocí varianty Newtonovy metody s tlumením.

x	$f'(x)$	$f''(x)$
2.30	−12.2356	−4.6750
2.31	−12.2818	−4.5785
2.32	−12.3272	−4.4818
2.33	−12.3715	−4.3848
2.34	−12.4148	−4.2875
2.35	−12.4572	−4.1901
2.36	−12.4986	−4.0924
2.37	−12.5391	−3.9945
2.38	−12.5785	−3.8964
2.39	−12.6170	−3.7981
2.40	−12.6545	−3.6997

Tabulka 2.5: Tabelace první a druhé derivace.

Kontrolní otázky

Otázka 1. Jak se odvozuje vzorec (2.9) a jak ho lze nakresli?
Otázka 2. Zkuste graficky znázornit předpoklady *(i)-(iv)* z Věty 2.3.2.
Otázka 3. Jakého řádu je Newtonova metoda, jak se to zapíše a jak projeví ve výpočtu?

Úlohy k samostatnému řešení

1. Ověřte, že pro rovnici $x^2 - x - \frac{6}{7}\ln x = 0$ jsou na intervalu $\langle 0.9, 0.91\rangle$ splněny předpoklady *(i)-(iv)* z Věty 2.3.2.
2. Vypočtěte kořen z předchozí úlohy pomocí Newtonovy metody pro $\epsilon = 10^{-6}$.

Výsledky úloh k samostatnému řešení

1. Z tabelace $f'(x) = 2x - 1 - \frac{6}{7x}$ a $f''(x) = 2 + \frac{6}{7x^2}$ zjistíme, že na uvedeném intervalu je první derivace záporná a druhá derivace kladná. Přímým výpočtem dostaneme $f(0.9)f(0.91) \doteq -3.28 \cdot 10^{-7} < 0$, $|f(0.9)/f'(0.9)| \doteq 2.03 \cdot 10^{-3} < 0.01$ a $|f(0.91)/f'(0.91)| \doteq 8.71 \cdot 10^{-3} < 0.01$.
2. Začneme-li z $x^0 = 0.9$, dostaneme ve třetí iteraci $\bar{x} = 0.9020709 \pm 10^{-6}$.

2.4 Newtonova metoda pro soustavy nelineárních rovnic

Uvažujme soustavu n nelineárních rovnic pro n neznámých:

$$\left.\begin{aligned} f_1(x_1,\ldots,x_n) &= 0,\\ &\vdots\\ f_n(x_1,\ldots,x_n) &= 0, \end{aligned}\right\} \tag{2.11}$$

kde $f_i : \Omega \mapsto \mathbb{R}$, $i = 1,\ldots,n$, jsou funkce n-proměnných a $\Omega \subseteq \mathbb{R}^n$ je jejich definiční obor. Označíme-li vektor proměnných $\mathbf{x} = (x_1,\ldots,x_n)^\top \in \mathbb{R}^n$ a vektorovou funkci $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}),\ldots,f_n(\mathbf{x}))^\top$, $\mathbf{f} : \Omega \mapsto \mathbb{R}^n$, můžeme úlohu (2.11) zapsat ve vektorové podobě jako:

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}, \tag{2.12}$$

kde $\mathbf{0}$ je nulový vektor. Protože formálně jsou úlohy (2.12) a (2.1) stejné, lze očekávat, že i odvození Newtonovy metody bude podobné jako v Odstavci 2.3. Je potřeba jen použít Taylorův rozvoj pro vektorové funkce. Také lokální řád metody bude stejný, jak uvidíme na příkladech. Zaměříme se proto především na algoritmický popis metody a na jeho efektivní provedení.

Nejdříve ale metodu odvodíme. Linearizovaná rovnice tvořená prvními dvěma členy Taylorova rozvoje v bodě $\mathbf{x}^{k-1}$, z níž určujeme novou aproximaci $\mathbf{x}^k$ má tvar:

$$\mathbf{f}(\mathbf{x}^{k-1}) + (\mathbf{x}^k - \mathbf{x}^{k-1})\mathbf{J}_\mathbf{f}(\mathbf{x}^{k-1}) = \mathbf{0}, \tag{2.13}$$

kde $\mathbf{J}_\mathbf{f}(\mathbf{x}^{k-1}) \in \mathbb{R}^{n\times n}$ je Jakobiova matice k funkci $\mathbf{f}$ v bodě $\mathbf{x}^{k-1}$ (srovnej s (2.8)). Jakobiova matice $\mathbf{J}_\mathbf{f}(\mathbf{x})$ v bodě $\mathbf{x}$ je utvořena z parciálních derivací složek vektorové funkce $\mathbf{f}$:

$$\mathbf{J}_\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \dots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{pmatrix}.$$

Pokud je Jakobiova matice $\mathbf{J}_\mathbf{f}(\mathbf{x}^{k-1})$ regulární, můžeme z (2.13) vyjádřit $\mathbf{x}^k$:

$$\mathbf{x}^k = \mathbf{x}^{k-1} - \mathbf{J}_\mathbf{f}(\mathbf{x}^{k-1})^{-1}\mathbf{f}(\mathbf{x}^{k-1}). \tag{2.14}$$

Tento vzorec je analogií (2.9) pro jednu rovnici, takže pomocí něho můžeme napsat podobný algoritmus jako v Odstavci 2.3. Ve výpočtech se ale vzorec (2.14) používá zřídka. Obsahuje totiž inverzní matici, jejíž sestavení je u rozsáhlejších úloh značně zdlouhavé. Výpočet lze upravit tak, že místo násobení inverzní maticí se řeší soustava lineárních rovnic. Označme tzv. *Newtonův směr*:

$$\mathbf{d}^k = \mathbf{x}^k - \mathbf{x}^{k-1}.$$

Vzorec (2.14) teď můžeme přepsat jako

$$\mathbf{J}_\mathbf{f}(\mathbf{x}^{k-1})\mathbf{d}^k = -\mathbf{f}(\mathbf{x}^{k-1}), \tag{2.15}$$

což je ona soustava lineárních rovnic. Pro ukončovací kritérium budeme ještě potřebovat zobecnění absolutní hodnoty, které si definujeme jako tuto *vektorovou normu*:

$$\|\mathbf{x}\| = \max_{i=1,\dots,n} |x_i|, \quad \mathbf{x} \in \mathbb{R}^n.$$

Promyslete si smysl ukončovacího kritéria v následujícím zápisu algoritmu Newtonovy metody pro soustavy nelineárních rovnic.

Algoritmus: Newtonova metoda pro soustavy
Vstup: $\mathbf{f}$, $\mathbf{J}_\mathbf{f}$, $\mathbf{x}^0$, ϵ.
Pro $k = 1, 2, \dots$ opakuj:
  $\mathbf{J}_\mathbf{f}(\mathbf{x}^{k-1})\mathbf{d}^k = -\mathbf{f}(\mathbf{x}^{k-1})$;
  $\mathbf{x}^k := \mathbf{x}^{k-1} + \mathbf{d}^k$;
dokud $\|\mathbf{d}^k\| > \epsilon$.
Výstup: $\bar{\mathbf{x}} = \mathbf{x}^k \pm \epsilon$.

Nyní si výpočet vyzkoušíme pro soustavu dvou nelineárních rovnic. Příklad si musíme nejdříve připravit, protože - jak jsme už zmínili - Newtonova metoda konverguje lokálně a my proto potřebujeme dobrou počáteční aproximaci.

Příklad 2.4.1 Je dána soustava dvou nelineárních rovnic pro dvě neznáme:

$$\begin{aligned} 2x_1 + x_1^2 x_2 - 2x_2 - 2 &= 0, \\ x_1^2 - 3x_2 &= 0. \end{aligned}$$

Určete kolik existuje kořenů a kde asi leží.

Řešení: Každou rovnicí je určena rovinná křivka, kterou můžeme znázornit jako graf funkce. Vyjádřením x_2 dostaneme dvě funkce v proměnné x_1 označené g_1 a g_2:

$$g_1(x_1) = \frac{2(1-x_1)}{(x_1-\sqrt{2})(x_1+\sqrt{2})}, \quad g_2(x_1) = \frac{x_1^2}{3}.$$

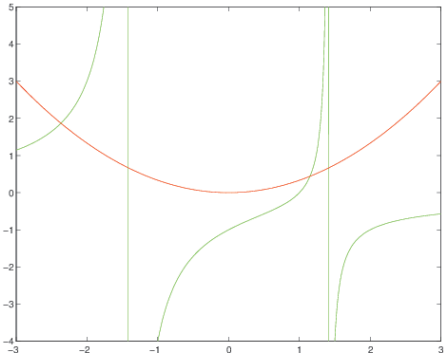
Funkce g_1 má tři větve oddělené singulárními body $-\sqrt{2}$ a $\sqrt{2}$, grafem funkce g_2 je parabola. Obě funkce nakreslíme na intervalu $\langle -3, 3\rangle$ těmito příkazy (nastavení os bylo třeba chvíli hledat):

```
>> x1=-3:0.01:3;
>> g1=2*(1-x1)./((x1-sqrt(2)).*(x1+sqrt(2)));
>> g2=x1.^2/3;
>> plot(x1,g1,'g',x1,g2,'r')
>> axis([-3 3 -4 5])
```

V Obrázku 2.5 je graf g_1 znázorněn zeleně a graf g_2 červeně. Z průsečíků grafů určíme dva kořeny $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2 \in \mathbb{R}^2$. Pomocí `Zoom in` lze určit jejich přibližnou polohu:

$$\begin{aligned} \bar{\mathbf{x}}_1 &\in \Omega_1 = \langle -2.4, -2.3\rangle \times \langle 1.8, 1.9\rangle, \\ \bar{\mathbf{x}}_2 &\in \Omega_2 = \langle 1.1, 1.2\rangle \times \langle 0.4, 0.5\rangle. \end{aligned}$$

Poznamenejme ještě, že svislé přímky v pozicích singulárních bodů jsou vytvořeny grafickou procedurou nadbytečně a nejsou součástí grafu funkce. □



Obrázek 2.5: Separace kořenů soustavy nelineárních rovnic z Příkladu 2.4.1.

Příklad 2.4.2 Newtonovou metodou vypočítejte kořen $\bar{\mathbf{x}}_2$ soustavy rovnic z Příkladu 2.4.1 pro $\epsilon = 10^{-6}$.

Řešení: V našem příkladě je $\mathbf{x} = (x_1, x_2)^\top$ a vektorová funkce $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^\top$ má složky $f_1(\mathbf{x}) = 2x_1 + x_1^2 x_2 - 2x_2 - 2$ a $f_2(\mathbf{x}) = x_1^2 - 3x_2$. Jakobiova matice má tvar (vypočtěte):

$$\mathbf{J}_\mathbf{f}(\mathbf{x}) = \begin{pmatrix} 2 + 2x_1x_2 & x_1^2 - 2 \\ 2x_1 & -3 \end{pmatrix}.$$

Vzorec (2.14) zapíšeme rozepsáním složek. Protože zápis je trochu nepřehledný, použijeme menší font a složky vektorů a matic oddělíme pomocí čar:

$$\begin{pmatrix} x_1^k \\ x_2^k \end{pmatrix} = \begin{pmatrix} x_1^{k-1} \\ x_2^{k-1} \end{pmatrix} - \left(\begin{array}{c|c} 2 + 2x_1^{k-1}x_2^{k-1} & (x_1^{k-1})^2 - 2 \\ \hline 2x_1^{k-1} & -3 \end{array}\right)^{-1} \begin{pmatrix} 2x_1^{k-1} + (x_1^{k-1})^2 x_2^{k-1} - 2x_2^{k-1} - 2 \\ \hline (x_1^{k-1})^2 - 3x_2^{k-1} \end{pmatrix}.$$

V MATLABu to bude přehlednější. Nejprve zadáme vektorovou funkci a Jakobiovu matici, pak počáteční aproximaci jako střed čtverce Ω_1 a nakonec zapíšeme vzorec (2.14):

```
>> f=@(x) [2*x(1)+x(1)^2*x(2)-2*x(2)-2;x(1)^2-3*x(2)];
>> Jf=@(x) [2+2*x(1)*x(2),x(1)^2-2;2*x(1),-3];
>> x0=[-2.35;1.85];
>> x1=x0-inv(Jf(x0))*f(x0), pres=max(abs(x1-x0)), x0=x1;
```

Opakováním poslední řádky provedeme výpočet. Průběh je zaznamenán v Tabulce 2.6. □

k	x_1^k	x_1^k	$\|\mathbf{x}^k - \mathbf{x}^{k-1}\|$
1	-2.367657765	1.868497165	0.018497165
2	-2.367458996	1.868287352	0.000209813
3	-2.367458970	1.868287325	$0.000000027 < 10^{-6} = \epsilon$

Tabulka 2.6: Řešení soustavy nelineární soustavu Newtonovou metodou.

I pro soustavy nelineárních rovnic je Newtonova metoda druhého řádu, jak je vidět z posledního sloupce Tabulky 2.6. Je to způsobeno tím, že linearizovaný vzorec (2.13) vznikl z Taylorova rozvoje opět vynecháním členů druhého a vyšších řádů.

Příklad 2.4.3 Zapište algoritmus Newtonovy metody pro soustavy nelineárních rovnic v MATLABu jako funkci `newton2`. Spusťte ji z příkazové řádky a poté vygenerujte data pro Tabulku 2.6.

Řešení: Postupujeme podobně jako v předchozích příkladech.

```
function [x1,k]=newton2(f,Jf,x0,epsilon,maxk)
for k=1:maxk
    d=-Jf(x0)\f(x0);
    x1=x0+d;
    pres=max(abs(d));
    if pres <=epsilon, break , end
    x0=x1;
end
```

Funkci $\mathbf{f}$ a Jakobiovu matici $\mathbf{J}_\mathbf{f}$ zadáme stejně jako v předchozím příkladu. Spuštění funkce `newton2` vypadá takto:

```
>> [x,k]=newton2(f,Jf,[-2.35;1.85],1e-6,100)
    x =  -2.367458970387554
          1.868287325489498
    k =  3
```

Data pro Tabulku 2.6 zobrazíme známým postupem. □

Kontrolní otázky

Otázka 1. Co je to Jakobiova matice?
Otázka 2. Jak se odvozuje Newtonova metoda pro soustavy a jakého je řádu?
Otázka 3. Co je to Newtonův směr a jak se počítá?

Úlohy k samostatnému řešení

1. Vypočítejte kořen $\bar{\mathbf{x}}_2$ z Příkladu 2.4.1 pro $\epsilon = 10^{-6}$.
2. Určete přibližnou polohu kořenů následující soustavy nelineárních rovnic a vypočítejte je Newtonovou metodou pro $\epsilon = 10^{-6}$:

$$\begin{aligned} 2\cos x + y^2 - 4x &= 0, \\ x^2 + 2xy - 5y &= 0. \end{aligned}$$

Výsledky úloh k samostatnému řešení

1. Pro $\mathbf{x}^0 = (1.15, 0.45)^\top$ dostaneme $\mathbf{x}^3 = (1.1494644 \pm 10^{-6}, 0.4404228 \pm 10^{-6})^\top$.
2. Existují dva kořeny, které lze separovat takto: $\bar{\mathbf{x}}_1 \in \Omega_1 = \langle 0.4503, 0.451\rangle \times \langle 0.04, 0.06\rangle$, $\bar{\mathbf{x}}_2 \in \Omega_2 = \langle 1.8, 1.9\rangle \times \langle 2.8, 2.9\rangle$. Jakobiova matice má tvar:

$$\mathbf{J}_\mathbf{f}(\mathbf{x}) = \begin{pmatrix} -2\sin x - 4 & 2y \\ 2x + 2y & 2x - 5y \end{pmatrix}.$$

Pro $\mathbf{x}_1^0 = (0.4507, 0.05)^\top$ dostaneme $\mathbf{x}_1^2 = (0.45068785 \pm 10^{-6}, 0.04955798 \pm 10^{-6})^\top$, pro $\mathbf{x}_2^0 = (1.85, 2.85)^\top$ dostaneme $\mathbf{x}_2^3 = (0.04955798 \pm 10^{-6}, 2.85118739 \pm 10^{-6})^\top$.

2.5 Metoda prosté iterace

Seznámíme se s obecnou metodu pro výpočet pevného bodu funkce. Metody z předchozích odstavců můžeme chápat jako její speciální varianty. Pojem pevného bodu hraje důležitou roli v různých oblastech matematického modelování.

Rovnici $f(x) = 0$ převedeme na ekvivalentní rovnici $x - g(x) = 0$, kde g je vhodná spojitá funkce. Místo původní rovnice budeme řešit rovnici v *iteračním tvaru*:

$$x = g(x). \tag{2.16}$$

Číslo $\bar{x}$, které je řešením rovnice (2.16), se nazývá *pevný bod funkce* g.

Věta 2.5.1 (*Brouwerova věta o pevném bodu*) Nechť g je spojitá funkce na intervalu $\langle a, b\rangle$, pro niž platí

$$g(x) \in \langle a, b\rangle \quad \forall x \in \langle a, b\rangle. \tag{2.17}$$

Pak na intervalu $\langle a, b\rangle$ existuje pevný bod funkce g.

Důkaz: Položme $f(x) = x - g(x)$. Pokud $f(a) = 0$ resp. $f(b) = 0$, pak je bevným bodem a, resp. b. Nechť $f(a) \neq 0$ a $f(b) \neq 0$. Protože $g(a) \in \langle a, b\rangle$, platí $f(a) = a - g(a) < 0$. Podobně lze ukázat $f(b) > 0$. Dohromady dostáváme $f(a)f(b) < 0$, takže existence pevného bodu plyne z Věty 2.1.1 □

O funkci g, která splňuje (2.17), říkáme, že zobrazuje interval $\langle a, b\rangle$ *do sebe*.

Nechť $x^0 \in \langle a, b\rangle$ je počáteční aproximace. *Metodou prostých iterací* nazýváme výpočet podle předpisu:

$$x^k = g(x^{k-1}), \quad k = 1, 2, \ldots \tag{2.18}$$

Jestliže posloupnost $\{x^k\}$ počítaná tímto postupem konverguje k číslu $\bar{x}$, pak limitním přechodem v (2.18) dostaneme, že $\bar{x}$ je pevným bodem funkce g. Výchozí rovnici $f(x) = 0$ můžeme převést na iterační tvar různými způsoby, ale jen některé vedou na konvergentní výpočet.

Příklad 2.5.4 Rovnici

$$f(x) \equiv e^x - x^2 + 1 = 0, \tag{2.19}$$

převeďte na iterační tvar a počítejte kořen, který leží v intervalu $\langle -1.2, -1.1\rangle$.

Řešení: Navrhneme tři iterační tvary:

$$\begin{aligned} x &= -\sqrt{e^x + 1} && \equiv g_a(x); \\ x &= x + (e^x - x^2 + 1) && \equiv g_b(x); \\ x &= x - \frac{e^x - x^2 + 1}{e^x - 2x} && \equiv g_c(x). \end{aligned}$$

Průběh výpočtů pro $x^0 = -1.1$ ukazuje Tabulka 2.7. Pro g_a výpočet konverguje pomalu, pro g_b výpočet diverguje a pro g_c výpočet konverguje rychle. □

Při studiu konvergence metody prostých iterací se používá pojem kontrakce.

Definice 2.5.1 Funkce g se nazývá *kontrakce na intervalu* $\langle a, b\rangle$, jestliže existuje konstanta L, $0 < L < 1$, taková, že

$$|g(x) - g(y)| \leq L|x - y| \quad \forall x, y \in \langle a, b\rangle. \tag{2.20}$$

k	$x^k = g_a(x^{k-1})$	$x^k = g_b(x^{k-1})$	$x^k = g_c(x^{k-1})$
0	−1.1	−1.1	−1.1
1	−1.1545003	−0.9771289	−1.1485105
2	−1.1468282	−0.5555196	−1.1477578
3	−1.1478861	+0.7096523	−1.1477576
4	−1.1477398	+3.2393301	−1.1477576
5	−1.1477600	+19.262693	
6	−1.1477572	∞	
7	−1.1477576		
8	−1.1477576		

Tabulka 2.7: Metoda prosté iterace.

Věta 2.5.2 Nechť g je spojitá kontrakce na intervalu $\langle a, b\rangle$, která zobrazuje tento interval do sebe. Pak na intervalu $\langle a, b\rangle$ existuje jediný pevný bod $\bar{x}$ funkce g. Navíc posloupnost $\{x^k\}$ vypočítaná podle předpisu (2.18) konverguje k $\bar{x}$ pro každou počáteční aproximaci $x^0 \in \langle a, b\rangle$.

Důkaz: Existence pevného bodu plyne z Věty 2.5.1. Jednoznačnost dokážeme sporem. Nechť $\tilde{x}$ je další pevný bod g. Pomocí (2.20) dostaneme

$$|\bar{x} - \tilde{x}| = |g(\bar{x}) - g(\tilde{x})| \leq L|\bar{x} - \tilde{x}|,$$

odkud $(1 - L)|\bar{x} - \tilde{x}| \leq 0$. Protože $1 - L > 0$, dostáváme $\bar{x} = \tilde{x}$.

Zbývá dokázat, že posloupnost $\{x^k\}$ vypočítaná podle předpisu (2.18) konverguje k $\bar{x}$. Podle (2.20) je

$$|x^k - \bar{x}| = |g(x^{k-1}) - g(\bar{x})| \leq L|x^{k-1} - \bar{x}|,$$

odkud plyne

$$|x^k - \bar{x}| \leq L^k|x^0 - \bar{x}|. \tag{2.21}$$

Protože $L \in (0, 1)$, je $\lim_{k\to\infty} L^k = 0$, a proto $\lim_{k\to\infty} |x^k - \bar{x}| = 0$. □

V konkrétních situacích je zpravidla nesnadné dokázat, že daná funkce g je kontrakce. Jednodušší je ověřovat následující silnější předpoklad:

$$\begin{aligned} &\text{nechť } g \text{ má v } (a, b) \text{ derivaci a} \\ &\exists L \in (0, 1) \text{ tak, že } |g'(\eta)| \leq L \quad \forall \eta \in (a, b). \end{aligned} \tag{2.22}$$

Věta 2.5.3 Nechť g je spojitá funkce na $\langle a, b\rangle$, která zobrazuje tento interval do sebe a splňuje (2.22). Pak platí tvrzení Věty 2.5.2.

Důkaz: Pomocí věty o střední hodnotě diferenciálního počtu dostáváme

$$|g(x) - g(y)| = |g'(\eta)|\,|x - y| \leq L|x - y|,$$

kde $\eta \in (x, y)$. Funkce g je proto kontrakce na intervalu $\langle a, b\rangle$ a Věta 2.5.3 je tak důsledkem Věty 2.5.2 □

Následující poznámka dává návod, jak rozhodnout o konvergenci metody prostých iterací.

Poznámka

Je-li číslo

$$M_g = \max_{x\in(a,b)} |g'(x)|$$

menší než jedna, pak můžeme položit $L = M_g$ a funkce g bude kontrakce na intervalu $\langle a, b\rangle$. Rychlost konvergence lze posoudit podle velikosti L. Vztah (2.21) totiž ukazuje, že výpočet bude konvergovat rychleji pro menší hodnoty L.

Příklad 2.5.5 Rozhodněte o konvergenci metody prostých iterací u iteračních tvarů z Příkladu 2.5.4.

Řešení: Derivováním g_a, g_b a g_c dostaneme

$$\begin{aligned} g_a'(x) &= -\frac{e^x}{2\sqrt{e^x+1}}, \\ g_b'(x) &= 1 + e^x - 2x, \\ g_c'(x) &= \frac{(e^x - x^2 + 1)(e^x - 2)}{(e^x - 2x)^2}. \end{aligned}$$

Tabulka 2.8 obsahuje absolutní hodnoty těchto derivací na intervalu $\langle -1.2, -1.1\rangle$. Odtud $M_{g_a} \doteq 0.1442$, $M_{g_b} \doteq 3.7012$ a $M_{g_c} \doteq 0.0323$. Protože $M_{g_b} > 1$ iterační tvar b) diverguje. Pro iterační tvary a) resp. c) můžeme položit $L_{g_a} = M_{g_a}$ resp. $L_{g_c} = M_{g_c}$, takže funkce g_a a g_b jsou kontrakce a metoda prostých iterací konverguje. Protože $L_{g_c} < L_{g_a}$, je konvergence rychlejší u iteračního tvaru c). □

x	$\lvert g_a'(x)\rvert$	$\lvert g_b'(x)\rvert$	$\lvert g_c'(x)\rvert$
−1.2000	0.1320	3.7012	0.0323
−1.1875	0.1335	3.6800	0.0248
−1.1750	0.1350	3.6588	0.0172
−1.1625	0.1365	3.6377	0.0094
−1.1500	0.1380	3.6166	0.0014
−1.1375	0.1395	3.5956	0.0067
−1.1250	0.1410	3.5747	0.0149
−1.1125	0.1426	3.5537	0.0233
−1.1000	0.1442	3.5329	0.0319

Tabulka 2.8: Posouzení rychlosti konvergence metody prosté iterace.

Kontrolní otázky

Otázka 1. Co nazýváme pevným bodem funkce? Jak se pevný bod počítá?
Otázka 2. Čím je zaručena konvergence metody prostých iterací?
Otázka 3. Jaký je vztah mezi metodou prosté iterace a Newtonovou metodou?

Úlohy k samostatnému řešení

1. Pro rovnici $x^2 - x - \frac{6}{7}\ln x = 0$ uvažujte iterační tvar $x = x^2 - \frac{6}{7}\log x \equiv g(x)$. Vypočítejte hodnotu konstanty M_g na intervalu $\langle 0.9, 0.91\rangle$.
2. U iteračního tvaru z předchozí úlohy vypočítejte pevný bod s přesností $\epsilon = 10^{-6}$.

Výsledky úloh k samostatnému řešení

1. Z tabelace $g'(x) = 2x - \frac{6}{7x}$ zjistíme, že $M_g = L \doteq 0.8781$.
2. Začneme-li z $x^0 = 0.9$, dostaneme ve 38-mé iteraci $x^{38} \doteq 0.9020659$ a platí $|x^{38} - x^{37}| \doteq 0.00000086 < \epsilon$.

KAPITOLA

3

SOUSTAVY LINEÁRNÍCH ROVNIC: PŘÍMÉ METODY

3.1 Formulace úlohy

Přímé metody řešení soustav lineárních rovnic jsou založeny na eliminaci neznámých. Výchozí myšlenka spočívá v tom, že z některé rovnice vyjádříme jednu neznámou a dosadíme ji do ostatních rovnic tak, aby soustava po eliminaci byla snáze řešitelná než soustava původní. Základní algoritmus tohoto typu je *Gaussova eliminační metoda*. V maticovém zápisu jí odpovídá *LU-rozklad matice*. Charakteristickým rysem přímých metod je výpočet (přesného) řešení po konečném počtu eliminací, tj. po konečném počtu aritmetických operací.

V předchozích odstavcích jsme řešili rovnici $f(x) = 0$. Nejjednodušším příkladem je lineární rovnice $ax = b$. Je–li $a \neq 0$, můžeme řešení zapsat ve tvaru $x = a^{-1}b$. Nyní se budeme zabývat zobecněním této úlohy:

> Nechť $\mathbf{A} = (a_{ij})$ je daná čtvercová matice řádu n s prvky a_{ij} a nechť $\mathbf{b} = (b_i)$ je daný sloupcový vektor s n prvky b_i, $i = 1, \ldots, n$, $j = 1, \ldots, n$; hledáme sloupcový vektor $\mathbf{x}$ takový, že

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.1}$$

Označíme–li x_i prvky vektoru $\mathbf{x}$, pak můžeme rovnici (3.1) zapsat jako soustavu lineárních rovnic:

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n. \end{aligned}\right\}$$

Pro existenci jediného řešení rovnice $ax = b$ musí být $a \neq 0$. Analogický předpoklad je potřeba i při řešení rovnice (3.1). Tímto předpokladem je nenulovost determinantu matice $\mathbf{A}$, tj. $\det \mathbf{A} \neq 0$. Matice $\mathbf{A}$, která má nenulový determinant se nazývá *regulární*, v opačném

případě se nazývá *singulární*. Ke každé regulární matici $\mathbf{A}$ existuje jediná *inverzní* matice $\mathbf{A}^{-1}$, pro niž platí $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, kde $\mathbf{I}$ je matice jednotková. Je-li matice $\mathbf{A}$ regulární, pak můžeme obě strany rovnice (3.1) vynásobit inverzní maticí $\mathbf{A}^{-1}$, tj. $\mathbf{A}^{-1}(\mathbf{A}\mathbf{x}) = \mathbf{A}^{-1}\mathbf{b}$. Protože $\mathbf{A}^{-1}(\mathbf{A}\mathbf{x}) = (\mathbf{A}^{-1}\mathbf{A})\mathbf{x} = \mathbf{I}\mathbf{x} = \mathbf{x}$, dostáváme

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}. \tag{3.2}$$

Tento vzorec dává návod jak vypočítat řešení.

Příklad 3.1.1 Pomocí vzorce (3.2) vyřešte soustavu lineárních rovnic:

$$\begin{aligned} x_1 + x_2 + x_3 &= 6, \\ 2x_1 + 4x_2 + 2x_3 &= 16, \\ -x_1 + 5x_2 - 4x_3 &= -3. \end{aligned} \tag{3.3}$$

Řešení: Protože

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 2 \\ -1 & 5 & -4 \end{pmatrix}, \quad \mathbf{A}^{-1} = \begin{pmatrix} \frac{13}{3} & -\frac{3}{2} & \frac{1}{3} \\ -1 & \frac{1}{2} & 0 \\ -\frac{7}{3} & 1 & -\frac{1}{3} \end{pmatrix}$$

(stačí ověřit $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$), dostáváme řešení

$$\mathbf{x} = \begin{pmatrix} \frac{13}{3} & -\frac{3}{2} & \frac{1}{3} \\ -1 & \frac{1}{2} & 0 \\ -\frac{7}{3} & 1 & -\frac{1}{3} \end{pmatrix} \begin{pmatrix} 6 \\ 16 \\ -3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

□

Poznámka

U rozsáhlejších soustav lineárních rovnic je použití vzorce (3.2) neekonomické, protože nalezení inverzní matice vyžaduje velké množství výpočtů.

Kontrolní otázky

Otázka 1. Kdy je matice regulární a kdy je singulární?
Otázka 2. Připomeňte si metody výpočtu determinantu a inverzní matice.

Úlohy k samostatnému řešení

1. Uvažujme soustavu lineárních rovnic

$$\begin{aligned} -x_1 - 3x_2 + 2x_3 &= -9, \\ -6x_1 - 19x_2 + 10x_3 &= -59, \\ 3x_1 + 9x_2 - 5x_3 &= 28. \end{aligned}$$

Vypočtěte inverzní matici a soustavu vyřešte pomocí vzorce (3.2).

Výsledky úloh k samostatnému řešení

1.

$$\mathbf{A} = \begin{pmatrix} -1 & -3 & 2 \\ -6 & -19 & 10 \\ 3 & 9 & -5 \end{pmatrix}, \quad \mathbf{A}^{-1} = \begin{pmatrix} 5 & 3 & 8 \\ 0 & -1 & -2 \\ 3 & 0 & 1 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}.$$

3.2 Gaussova eliminační metoda

Pro jednoduchost budeme uvažovat soustavy lineárních rovnic s regulární maticí. Nejdříve si Gaussovu eliminační metodu připomeneme na příkladu. Soustavu lineárních rovnic (3.3) můžeme zapsat ve tvaru:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \text{tj.} \quad \begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 2 \\ -1 & 5 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 16 \\ -3 \end{pmatrix}. \tag{3.4}$$

V první fázi eliminujeme v prvním sloupci. První rovnici vynásobíme číslem $m_{21} = -2$ resp. $m_{31} = 1$ a přičteme ke druhé resp. třetí rovnici. Dostaneme:

$$\mathbf{A}_1\mathbf{x} = \mathbf{b}_1, \quad \text{tj.} \quad \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 6 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 3 \end{pmatrix}.$$

Ve druhé fázi eliminujeme ve druhém sloupci. Druhou rovnici vynásobíme číslem $m_{32} = -3$ a přičteme ke třetí rovnici. Dostaneme:

$$\mathbf{A}_2\mathbf{x} = \mathbf{b}_2, \quad \text{tj.} \quad \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ -9 \end{pmatrix}$$

Poslední soustavu s *horní trojúhelníkovou maticí* budeme zapisovat jako $\mathbf{U}\mathbf{x} = \mathbf{y}$, tj. $\mathbf{U} = \mathbf{A}_2$, $\mathbf{y} = \mathbf{b}_2$. Pro lepší názornost použijeme zápis po rovnicích:

$$\begin{aligned} x_1 + x_2 + x_3 &= 6, \\ 2x_2 &= 4, \\ -3x_3 &= -9. \end{aligned} \tag{3.5}$$

Odtud $x_3 = 3$, $x_2 = 2$ a $x_1 = 1$.

V příkladu jsme viděli, že ve výpočet Gaussovy eliminační metody lze rozdělit ne dvě odlišné části:

- *dopředný chod* je úprava výchozí soustavy $\mathbf{A}\mathbf{x} = \mathbf{b}$ na soustavu $\mathbf{U}\mathbf{x} = \mathbf{y}$ s horní trojúhelníkovou maticí $\mathbf{U}$;
- *zpětný chod* je výpočet řešení ze soustavy $\mathbf{U}\mathbf{x} = \mathbf{y}$.

3.2.1 Zpětný chod

Uvažujme soustavu lineárních rovnic $\mathbf{U}\mathbf{x} = \mathbf{y}$ s horní trojúhelníkovou maticí $\mathbf{U} = (u_{ij})$, $u_{ij} = 0$, $i > j$, a vektorem pravé strany $\mathbf{y} = (y_i)$. Zapíšeme-li tuto soustavu po jednotlivých rovnicích, dostaneme

$$\begin{aligned} u_{11}x_1 + u_{12}x_2 + \cdots + u_{1n}x_n &= y_1, \\ u_{22}x_2 + \cdots + u_{2n}x_n &= y_2, \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ u_{nn}x_n &= y_n. \end{aligned}$$

Výpočet řešení $\mathbf{x} = (x_i)$ se provádí postupným dosazováním "od konce".

Algoritmus: Zpětný chod
Vstup: $\mathbf{U} = (u_{ij})$, $\mathbf{y} = (y_i)$.
$x_n := y_n / u_{nn}$.
Pro $i = n-1, \ldots, 1$ počítej i-tou neznámou:
$\quad x_i := (y_i - u_{in}x_n - \ldots - u_{ii+1}x_{i+1}) / u_{ii}$.
Výstup: $\mathbf{x} = (x_i)$.

Všimněme si počtu operací. Při výpočtu i-té neznámé potřebujeme provést jedno dělení a $n - i$ odčítání a násobení. Celkový počet operací je

$$\sum_{i=n}^{1} [1 + 2(n-i)] \;=\; n^2 \;=\; \mathcal{O}(n^2).$$

3.2.2 Dopředný chod

Dopředný chod Gaussovy eliminační metody má $n-1$ *fází*. V k-té fázi, $1 \le k \le n-1$, se provádí eliminace v k-tém sloupci matice. Pro jednoduchost budeme předpokládat, že není potřeba měnit pořadí řádků tak, jak tomu bylo v našem úvodním příkladu. Na tomto příkladu by si měl čtenář také ilustrovat všechny níže uvedené pojmy.

Prvky matice na začátku k–té fáze označíme $a_{ij}^{(k)}$ a prvky vektoru pravé strany $a_{in+1}^{(k)}$ (na začátku 1. fáze je $a_{ij}^{(1)} = a_{ij}$ a $a_{in+1}^{(1)} = b_i$). Eliminace provádíme v k-tém sloupci matice pod jejím diagonálním prvkem $a_{kk}^{(k)}$, kterému říkáme *hlavní prvek k-té fáze*. Nejdříve počítáme *multiplikátory k-té fáze*

$$m_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1, \ldots, n \tag{3.6}$$

a pak přičteme m_{ik}-násobek k-tého řádku k řádku i-tému, tj.

$$a_{ij}^{(k+1)} := a_{ij}^{(k)} + m_{ik}\, a_{kj}^{(k)}, \quad j = k+1, \ldots, n+1,$$

pro $i = k+1, \ldots, n+1$.

Algoritmus: Dopředný chod
Vstup: $\mathbf{A} = (a_{ij})$, $\mathbf{b} = (b_i)$.
$a_{ij}^{(1)} := a_{ij}$, $a_{in+1}^{(1)} := b_i$, $i, j = 1, \ldots, n$.
Pro $k = 1, \ldots, n-1$ proveď k-tou fázi:
  Pro $i = k+1, \ldots, n$ přičti m_{ik}-násobek k-tého řádku k i-tému řádku:
    $m_{ik} := -a_{ik}^{(k)} / a_{kk}^{(k)}$;
    Pro $j = k+1, \ldots, n+1$ proveď přičítání v j-tém sloupci:
      $a_{ij}^{(k+1)} := a_{ij}^{(k)} + m_{ik}\, a_{kj}^{(k)}$.
Polož $u_{ij} := a_{ij}^{(n)}$ pro $i \le j$, $u_{ij} := 0$ pro $i > j$, $y_i := a_{in+1}^{(n)}$, $i, j = 1, \ldots, n$.
Výstup: $\mathbf{U} = (u_{ij})$, $\mathbf{y} = (y_i)$.

Ze vzorce (3.6) je vidět, že hlavní prvek musí být nenulový. Pokud není, provedeme na začátku k-té fáze *výběr hlavního prvku*. Tomu se budeme věnovat v dalším odstavci. Všimněme si ještě počtu operací. V k-té fázi musíme provést $n-k$ dělení při výpočtu multiplikátorů. Kromě toho počítáme $(n-k)(n-k+1)$ prvků $a_{ij}^{(k+1)}$ pomocí dvou operací (násobení a sčítání). Celkový počet operací je proto

$$\sum_{k=1}^{n-1} [(n-k) + 2(n-k)(n-k+1)] \;=\; \frac{2}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{3}n.$$

Pro velká n převažuje v posledním výrazu člen $\frac{2}{3}n^3$. Říkáme, že dopředný chod Gaussovy eliminační metody vyžaduje $\mathcal{O}(\frac{2}{3}n^3)$ operací.

3.2.3 Výběr hlavního prvku

Cílem je vybrat hlavní prvek tak, aby jeho absolutní hodnota byla maximální. V první fázi dopředného chodu se za hlavní prvek vybere v absolutní hodnotě největší číslo z prvního sloupce matice. Eliminace se pak provedou ve všech řádcích neobsahujících hlavní prvek. V k-té fázi se celý výpočet omezí na řádky, v nichž dosud hlavní prvek nebyl vybrán. Nejprve se jako hlavní prvek vybere v absolutní hodnotě největší z čísel ležících v k-tém sloupci a příslušných řádcích a pak se provedou eliminace ve zbývajících řádcích.

Tento postup lze přehledně provádět pomocí přehazováním řádků. Do algoritmu dopředného chodu stačí na začátek k-té fáze vsunout následující doplněk:

$$\begin{cases} \text{Najdi } p,\ p \ge k, \text{ takové, že } |a_{pk}^{(k)}| = \max\{|a_{ik}^{(k)}|,\ i \ge k\}; \\ \text{Prohoď } p\text{-tý a } k\text{-tý řadek matice v } k\text{-té fázi.} \end{cases}$$

Algoritmus dopředného chodu s výběrem hlavního prvku lze provést pro každou regulární matici. Poznamenejme ještě, že přehazování řádků není nutné a v efektivních počítačových implementacích se neprovádí. Je jen potřeba uchovávat informaci o tom, který řádek obsahoval hlavní prvek v k-té fázi. Tato informace umožňuje definovat permutační matici, o které budeme mluvit později.

Příklad 3.2.1 Soustavu lineárních rovnic (3.4) řešte pomocí Gaussovy eliminační metody s výběrem hlavního prvku. Přehazujte přitom řádky.

Řešení: Výběr hlavního prvku a eliminace v první fázi:

$$\begin{pmatrix} 2 & 4 & 2 \\ 1 & 1 & 1 \\ -1 & 5 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16 \\ 6 \\ -3 \end{pmatrix}, \begin{pmatrix} 2 & 4 & 2 \\ 0 & -1 & 0 \\ 0 & 7 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16 \\ -2 \\ 5 \end{pmatrix}.$$

Výběr hlavního prvku a eliminace ve druhé fázi:

$$\begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16 \\ 5 \\ -2 \end{pmatrix}, \begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & 0 & -\frac{3}{7} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16 \\ 5 \\ -\frac{9}{7} \end{pmatrix}$$

Zpětným chodem vypočítáme $x_3 = 3$, $x_2 = 2$ a $x_1 = 1$. □

Kontrolní otázky

Otázka 1. Z jakých částí se skládá algoritmus GEM a jak jsou výpočetně náročné?
Otázka 2. Proč se provádí výběr hlavního prvku?

Úlohy k samostatnému řešení

1. Soustavu lineárních rovnic

$$\begin{aligned} -x_1 - 3x_2 + 2x_3 &= -9, \\ -6x_1 - 19x_2 + 10x_3 &= -59, \\ 3x_1 + 9x_2 - 5x_3 &= 28 \end{aligned}$$

řešte pomocí GEM bez výběru a s výběrem hlavního prvku.

Výsledky úloh k samostatnému řešení

1. Řešením je vektor $\mathbf{x} = (2, 3, 1)^\top$.

3.3 LU–rozklad

Ukážeme, že Gaussovu eliminační metodu lze na maticové úrovni zapsat jako LU-rozklad. K regulární čtvercové matici $\mathbf{A}$ budeme hledat dolní trojúhelníkovou matici $\mathbf{L}$ a horní trojúhelníkovou matici $\mathbf{U}$ takové, aby platilo

$$\mathbf{A} = \mathbf{LU}.$$

Začneme příkladem s maticí soustavy (3.4). Navážeme přitom na příklad ze začátku Odstavce 3.2, kdy jsme pomocí dopředného chodu vytvořili z $\mathbf{A}$ v první fázi $\mathbf{A}_1$ a ve druhé fázi $\mathbf{A}_2$:

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 2 \\ -1 & 5 & -4 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 6 & -3 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & -3 \end{pmatrix}.$$

Protože $\mathbf{A}_2$ je horní trojúhelníková matice, položíme $\mathbf{U} = \mathbf{A}_2$. Zbývá ukázat, jak vypadá dolní trojúhelníková matice $\mathbf{L}$. První fázi zapíšeme jako násobení maticí $\mathbf{M}_1$, kterou sestavíme z multiplikátorů $m_{21} = -2$ a $m_{31} = 1$:

$$\mathbf{A}_1 = \mathbf{M}_1\mathbf{A}, \text{ kde } \mathbf{M}_1 = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Podobně druhou fázi zapíšeme jako násobení maticí $\mathbf{M}_2$, která je určena multiplikátorem $m_{32} = -3$:

$$\mathbf{A}_2 = \mathbf{M}_2\mathbf{A}_1, \text{ kde } \mathbf{M}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{pmatrix}.$$

Dosazením dostaneme $\mathbf{U} = \mathbf{A}_2 = \mathbf{M}_2\mathbf{M}_1\mathbf{A}$ a odtud $\mathbf{A} = \mathbf{M}_1^{-1}\mathbf{M}_2^{-1}\mathbf{U}$. Zdá se, že matice $\mathbf{L}$ by mohl být součin $\mathbf{M}_1^{-1}\mathbf{M}_2^{-1}$. Musíme ale ještě ověřit, že se jedná o dolní trojúhelníkovou matici. Nejdříve si všimněme, že inverzní matice $\mathbf{M}_1^{-1}$ a $\mathbf{M}_2^{-1}$ mají tvar

$$\mathbf{M}_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix},$$

$$\mathbf{M}_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix}$$

(stačí ověřit, že platí $\mathbf{M}_1^{-1}\mathbf{M}_1 = \mathbf{I}$ a $\mathbf{M}_2^{-1}\mathbf{M}_2 = \mathbf{I}$). Vynásobením dostaneme

$$\mathbf{M}_1^{-1}\mathbf{M}_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & -m_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 3 & 1 \end{pmatrix}.$$

Proto můžeme položit $\mathbf{L} = \mathbf{M}_1^{-1}\mathbf{M}_2^{-1}$ a platí

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

Uvedený postup lze zobecnit pro matici libovolného řádu n.

Věta 3.3.1 Nechť $\mathbf{A}$ je matice řádu n, kterou lze dopředným chodem bez výběru hlavního prvku upravit na horní trojúhelníkovou matici $\mathbf{U}$. Nechť m_{ik}, $k = 1, \ldots, n-1$, $i = k+1, \ldots, n$ jsou multiplikátory k-té fáze, z nichž vytvoříme dolní trojúhelníkovou matici $\mathbf{L} = (l_{ik})$ tak, že $l_{ik} = -m_{ik}$, $i > k$, $l_{ii} = 1$ a $l_{ik} = 0$, $i < k$. Potom platí

$$\mathbf{A} = \mathbf{LU}.$$

Z předchozího odstavce víme, že dopředný chod *bez* výběru hlavního prvku nelze provést pro každou matici $\mathbf{A}$. Obecný tvar LU-rozkladu proto obsahuje ještě permutační matici, která popisuje přehazování řádků, k nimž dochází při výběru hlavního prvku.

Věta 3.3.2 Nechť $\mathbf{A}$ je regulární matice řádu n. Pak existují dolní trojúhelníková matice $\mathbf{L}$, horní trojúhelníková matice $\mathbf{U}$ a permutační matice $\mathbf{P}$ řádu n takové, že

$$\mathbf{PA} = \mathbf{LU}. \tag{3.7}$$

Důkaz: Princip důkazu je následující. Přehazování řádků v průběhu dopředného chodu se zaznamená v permutační matici $\mathbf{P}$. Jestliže se vrátíme na začátek a vytvoříme $\mathbf{PA}$ (tj. přehodíme řádky matice $\mathbf{A}$ tak, jak to vyžaduje průběh výpočtu), pak můžeme pro tuto matici najít její LU-rozklad podle Věty 3.3.1, protože přitom přehazování řádku již nebude potřeba. □

Při praktickém výpočtu LU-rozkladu (3.7) postupujeme například takto:

- vytvoříme pomocné matice $\tilde{\mathbf{U}} = \mathbf{A}$, $\tilde{\mathbf{P}} = \mathbf{I}$ a $\tilde{\mathbf{L}} = \mathbf{I}$;
- v matici $\tilde{\mathbf{U}}$ provádíme dopředný chod s výběrem hlavního prvku;
- v matici $\tilde{\mathbf{P}}$ přehazujeme řádky stejně jako v matici $\tilde{\mathbf{U}}$;
- do matice $\tilde{\mathbf{L}}$ zapíšeme v každé fázi multiplikátory (s opačnými znaménky) a při přehození řádků v $\tilde{\mathbf{U}}$ přehodíme v $\tilde{\mathbf{L}}$ řádky i sloupce;
- nakonec dostáváme $\mathbf{P} = \tilde{\mathbf{P}}$, $\mathbf{L} = \tilde{\mathbf{L}}$ a $\mathbf{U} = \tilde{\mathbf{U}}$.

Příklad 3.3.1 Vypočtěte LU-rozklad (3.7) pro matici

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

Řešení:

$$\tilde{\mathbf{U}} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 2 \\ -1 & 5 & -4 \end{pmatrix}, \quad \tilde{\mathbf{P}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \tilde{\mathbf{L}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Výběr hlavního prvku v první fázi:

$$\tilde{\mathbf{U}} = \begin{pmatrix} 2 & 4 & 2 \\ 1 & 1 & 1 \\ -1 & 5 & -4 \end{pmatrix}, \quad \tilde{\mathbf{P}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \tilde{\mathbf{L}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Eliminace v první fázi s multiplikátory $m_{21} = -\frac{1}{2}$ a $m_{31} = \frac{1}{2}$:

$$\tilde{\mathbf{U}} = \begin{pmatrix} 2 & 4 & 2 \\ 0 & -1 & 0 \\ 0 & 7 & -3 \end{pmatrix}, \quad \tilde{\mathbf{P}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \tilde{\mathbf{L}} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{pmatrix}.$$

Výběr hlavního prvku ve druhé fázi:

$$\tilde{\mathbf{U}} = \begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & -1 & 0 \end{pmatrix}, \quad \tilde{\mathbf{P}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \tilde{\mathbf{L}} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix}.$$

Eliminace ve druhé fázi s multiplikátorem $m_{32} = \frac{1}{7}$:

$$\tilde{\mathbf{U}} = \begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & 0 & -\frac{3}{7} \end{pmatrix}, \quad \tilde{\mathbf{P}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \tilde{\mathbf{L}} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{7} & 1 \end{pmatrix}.$$

Výsledek je $\mathbf{P} = \tilde{\mathbf{P}}$, $\mathbf{L} = \tilde{\mathbf{L}}$ a $\mathbf{U} = \tilde{\mathbf{U}}$. □

Kontrolní otázky

Otázka 1. Jak souvisí LU-rozklad s GEM?
Otázka 2. Jak se provádí výpočet LU-rozkladu?

Úlohy k samostatnému řešení

1. Pro matici

$$\mathbf{A} = \begin{pmatrix} -1 & -3 & 2 \\ -6 & -19 & 10 \\ 3 & 9 & -5 \end{pmatrix}.$$

vypočtěte LU-rozklad $\mathbf{A} = \mathbf{LU}$.
2. Pro matici z předchozí úlohy vypočtěte LU-rozklad $\mathbf{PA} = \mathbf{LU}$.
3. Jaká je výpočetní náročnost LU-rozkladu?

Výsledky úloh k samostatnému řešení

1.

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ 6 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} -1 & -3 & 2 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{pmatrix}.$$

2.

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{6} & -\frac{1}{3} & 1 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} -6 & -19 & 10 \\ 0 & -\frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{pmatrix}.$$

3. Výpočetní náročnost je zhruba stejná jako u dopředného chodu GEM. Objem výpočtu se zmenšil o úpravu vektoru pravé strany, což představuje $\mathcal{O}(n^2)$ operací. Platí $\mathcal{O}(\frac{2}{3}n^3) - \mathcal{O}(n^2) = \mathcal{O}(\frac{2}{3}n^3)$.

3.4 Použití LU-rozkladu

Pomocí LU-rozkladu lze řešit tradiční úlohy lineární algebry: řešení soustavy lineárních rovnic, výpočet inverzní matice, výpočet determinantu a řadu dalších úloh. V podstatě jde jen o použití Gaussovy eliminační metody, takže na první pohled se může zdát, že tím nezískáme nic nového. Ve skutečnosti je použití matic **L**, **U** a **P** velmi významné z metodického hlediska. Výpočty lze přehledně uspořádávat tak, aby výsledná podoba určitého algoritmu byla optimální z pohledu minimalizace výpočetních nároků. Další předností je fakt, že manipulace s LU-rozkladem představuje procedurální programování zapsané v terminologii matic. Algoritmy, které uvedeme níže, jsou kostry počítačových programů, kde stačí operace s maticemi nahradit příslušnou programovou procedurou.

3.4.1 Řešení soustav lineárních rovnic

Uvažujme soustavu lineárních rovnic

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

s regulární čtvercovou maticí řádu n a předpokládejme, že **P**, **L** a **U** jsou matice, které tvoří LU-rozklad $\mathbf{PA} = \mathbf{LU}$. Platí následující ekvivalence:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \mathbf{PA}\mathbf{x} = \mathbf{P}\mathbf{b} \quad \Leftrightarrow \quad \mathbf{LU}\mathbf{x} = \mathbf{P}\mathbf{b}.$$

Poslední rovnici rozložíme s využitím pomocné proměnné **y** na dvě rovnice

$$\mathbf{L}\mathbf{y} = \mathbf{P}\mathbf{b}, \quad \mathbf{U}\mathbf{x} = \mathbf{y}$$

a dostáváme následující algoritmus.

Algoritmus: Řešení soustav lineárních rovnic
Vstup: **A**, **b**.
Krok 1: Vypočti matice **P**, **L** a **U**, které tvoří LU-rozklad $\mathbf{PA} = \mathbf{LU}$.
Krok 2: Vyřeš soustavu lineárních rovnic $\mathbf{L}\mathbf{y} = \mathbf{P}\mathbf{b}$.
Krok 3: Vyřeš soustavu lineárních rovnic $\mathbf{U}\mathbf{x} = \mathbf{y}$.
Výstup: **x**.

Protože matice **L** a **U** jsou trojúhelníkové, stačí u kroků 2 a 3 provést $2\mathcal{O}(n^2)$ operací. Krok 1 je podstatně pracnější, vyžaduje totiž $\mathcal{O}(\frac{2}{3}n^3)$ operací. Podrobným rozborem se dá ukázat, že pracnost celého algoritmu je naprosto stejná jako pracnost Gaussovy eliminační metody.

Příklad 3.4.1 Pomocí LU-rozkladu $\mathbf{PA} = \mathbf{LU}$ řešte soustavu

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 2 \\ -1 & 5 & -4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 16 \\ -3 \end{pmatrix}.$$

Řešení: LU-rozklad pro matici této soustavy jsme vypočítali v Příkladu 3.3.1:

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{7} & 1 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & 0 & -\frac{3}{7} \end{pmatrix}. \tag{3.8}$$

Pro druhý krok algoritmu potřebujeme připravit pravou stranu pomocí prmutace

$$\mathbf{Pb} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 6 \\ 16 \\ -3 \end{pmatrix} = \begin{pmatrix} 16 \\ -3 \\ 6 \end{pmatrix}.$$

Máme tedy řešit soustavu

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{7} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 16 \\ -3 \\ 6 \end{pmatrix}.$$

Odtud postupně vypočítáme $y_1 = 16$, $y_2 = 5$ a $y_3 = -\frac{9}{7}$. Soustava ve třetím kroku algoritmu má tvar

$$\begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & 0 & -\frac{3}{7} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16 \\ 5 \\ -\frac{9}{7} \end{pmatrix}.$$

Z ní postupně vypočítáme řešení $x_3 = 3$, $x_2 = 2$ a $x_1 = 1$. □

3.4.2 Výpočet inverzní matice

Připomeňme, že pro inverzní matici platí $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$. Označíme-li $\mathbf{a}^{(i)}$ i–tý sloupec matice inverzní $\mathbf{A}^{-1}$ a $\mathbf{e}^{(i)}$ i–tý sloupec matice jednotkové $\mathbf{I}$, pak můžeme uvedenou rovnost zapsat jako $\mathbf{A}(\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(n)}) = (\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(n)})$ a po roznásobení jako

$$(\mathbf{A}\mathbf{a}^{(1)}, \ldots, \mathbf{A}\mathbf{a}^{(n)}) = (\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(n)}).$$

Odtud je zřejmé, že musí být splněny soustavy lineárních rovnic

$$\mathbf{A}\mathbf{a}^{(i)} = \mathbf{e}^{(i)}, \quad i = 1, \ldots, n.$$

Protože matice je u všech soustav stejná, stačí při jejich řešení vypočítat LU-rozklad jenom jednou.

Algoritmus: Výpočet inverzní matice
Vstup: **A**.
Krok 1: Vypočti matice **P**, **L** a **U**, které tvoří LU-rozklad $\mathbf{PA} = \mathbf{LU}$.
Pro $i = 1, \ldots, n$ vypočti i–tý sloupec inverzní matice:
  Krok 2: Vyřeš soustavu lineárních rovnic $\mathbf{Ly} = \mathbf{Pb}$, kde $\mathbf{b} = \mathbf{e}^{(i)}$;
  Krok 3: Vyřeš soustavu lineárních rovnic $\mathbf{Ux} = \mathbf{y}$ a polož $\mathbf{a}^{(i)} = \mathbf{x}$.
Výstup: $\mathbf{A}^{-1} = (\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(n)})$.

Výpočetní náročnost je $\mathcal{O}(\frac{2}{3}n^3)$ v kroku 1 a n-krát $2\mathcal{O}(n^2)$ v krocích 2 a 3. Celkem tedy vyžaduje algoritmus $\mathcal{O}(\frac{8}{3}n^3)$ operací, což je zhruba čtyřikrát víc než při řešení soustavy lineárních rovnic.

Příklad 3.4.2 Vypočtěte inverzní matici $\mathbf{A}^{-1}$ k matici

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

Řešení: LU-rozklad tvoří matice (3.8). Podle algoritmu dále počítáme postupně jednotlivé sloupce inverzní matice.
Pro $i = 1$ v kroku druhém řešíme $\mathbf{Ly} = \mathbf{P}(1,0,0)^\top$:

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{7} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \Longrightarrow \quad \mathbf{y} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Pro $i = 1$ v kroku třetím řešíme $\mathbf{Ux} = \mathbf{y}$:

$$\begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & 0 & -\frac{3}{7} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \Longrightarrow \quad \mathbf{x} = \begin{pmatrix} \frac{13}{3} \\ -1 \\ -\frac{7}{3} \end{pmatrix} = \mathbf{a}^{(1)}.$$

Pro $i = 2$ v kroku druhém řešíme $\mathbf{Ly} = \mathbf{P}(0,1,0)^\top$:

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{7} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \Longrightarrow \quad \mathbf{y} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{-3}{7} \end{pmatrix}.$$

Pro $i = 2$ v kroku třetím řešíme $\mathbf{Ux} = \mathbf{y}$:

$$\begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & 0 & -\frac{3}{7} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{-3}{7} \end{pmatrix} \quad \Longrightarrow \quad \mathbf{x} = \begin{pmatrix} -\frac{3}{2} \\ \frac{1}{2} \\ 1 \end{pmatrix} = \mathbf{a}^{(2)}.$$

Pro $i = 3$ v kroku druhém řešíme $\mathbf{Ly} = \mathbf{P}(0,0,1)^\top$:

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{7} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \Longrightarrow \quad \mathbf{y} = \begin{pmatrix} 0 \\ 1 \\ \frac{1}{7} \end{pmatrix}.$$

Pro $i = 3$ v kroku třetím řešíme $\mathbf{Ux} = \mathbf{y}$:

$$\begin{pmatrix} 2 & 4 & 2 \\ 0 & 7 & -3 \\ 0 & 0 & -\frac{3}{7} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \frac{1}{7} \end{pmatrix} \quad \Longrightarrow \quad \mathbf{x} = \begin{pmatrix} \frac{1}{3} \\ 0 \\ -\frac{1}{3} \end{pmatrix} = \mathbf{a}^{(3)}.$$

Vypočítali jsme inverzní matici:

$$\mathbf{A}^{-1} = (\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)}) = \begin{pmatrix} \frac{13}{3} & -\frac{3}{2} & \frac{1}{3} \\ -1 & \frac{1}{2} & 0 \\ -\frac{7}{3} & 1 & -\frac{1}{3} \end{pmatrix}.$$

□

3.4.3 Výpočet determinantu

Použijeme jedno ze základních pravidel pro počítání s determinanty, které říká, že determinant ze součinu (čtvercových) matic se rovná součinu jejich determinantů. Jestliže matice $\mathbf{P}$, $\mathbf{L}$ a $\mathbf{U}$ tvoří LU-rozklad $\mathbf{PA} = \mathbf{LU}$, pak můžeme psát

$$\det \mathbf{A} = (\det \mathbf{P})^{-1} \cdot \det \mathbf{L} \cdot \det \mathbf{U}.$$

Determinanty trojúhelníkových matic vypočítáme snadno jako součiny jejich diagonálních prvků. Determinant permutační matice je $+1$, resp. -1 podle toho, jestli vznikla z jednotkové matice sudým, resp. lichým počtem přehození řádků.

Příklad 3.4.3 Vypočítejte determinant matice $\mathbf{A}$ z Příkladu 3.3.1

Řešení: Pomocí výsledku Příkladu 3.3.1 dostáváme

$$\begin{aligned} \det \mathbf{L} &= 1 \cdot 1 \cdot 1 = 1, \\ \det \mathbf{U} &= 2 \cdot 7 \cdot \frac{-3}{7} = -6. \end{aligned}$$

Protože při výpočtu LU-rozkladu došlo ke dvěma záměnám řádků, bude $\det \mathbf{P} = 1$. Celkem je $\det \mathbf{A} = 1 \cdot 1 \cdot (-6) = -6$. □

Kontrolní otázky

Otázka 1. Jak se pomocí LU-rozkladu řeší soustava lineárních rovnic?
Otázka 2. Jak se pomocí LU-rozkladu počítá inverzní matice?
Otázka 3. Jak se pomocí LU-rozkladu počítá determinant?

Úlohy k samostatnému řešení

1. Soustavu lineárních rovnic

$$\begin{aligned} -x_1 - 3x_2 + 2x_3 &= -9, \\ -6x_1 - 19x_2 + 10x_3 &= -59, \\ 3x_1 + 9x_2 - 5x_3 &= 28 \end{aligned}$$

řešte pomocí LU-rozkladu $\mathbf{A} = \mathbf{LU}$.
2. Soustavu lineárních rovnic z první úlohy řešte pomocí LU-rozkladu $\mathbf{PA} = \mathbf{LU}$.

3. Pro matici

$$\mathbf{A} = \begin{pmatrix} -1 & -3 & 2 \\ -6 & -19 & 10 \\ 3 & 9 & -5 \end{pmatrix}$$

vypočtěte inverzní matici pomocí LU-rozkladu $\mathbf{A} = \mathbf{LU}$.
4. K předchozí matici vypočtěte inverzní matici pomocí LU-rozkladu $\mathbf{PA} = \mathbf{LU}$.
5. Vypočtěte determinant matice z 3. úlohy pomocí LU-rozkladu $\mathbf{A} = \mathbf{LU}$.
6. Vypočtěte determinant matice z 3. úlohy pomocí LU-rozkladu $\mathbf{PA} = \mathbf{LU}$.
7. Kolik operací je potřeba při výpočtu determinantu matice pomocí LU-rozkladu?

Výsledky úloh k samostatnému řešení

1. Dostaneme $\mathbf{y} = (-9, -5, 1)^\top$ a $\mathbf{x} = (2, 3, 1)^\top$.
2. Dostaneme $\mathbf{y} = (-59, -\frac{3}{2}, \frac{1}{3})^\top$ a $\mathbf{x} = (2, 3, 1)^\top$.
3. Pro $i = 1$ je $\mathbf{y} = (1, 6, -3)^\top$, pro $i = 2$ je $\mathbf{y} = (0, 1, 0)^\top$ a pro $i = 3$ je $\mathbf{y} = (0, 0, 1)^\top$. Inverzní matice má tvar

$$\mathbf{A}^{-1} = \begin{pmatrix} 5 & 3 & 8 \\ 0 & -1 & -2 \\ 3 & 0 & 1 \end{pmatrix}.$$

4. Pro $i = 1$ je $\mathbf{y} = (0, 0, 1)^\top$, pro $i = 2$ je $\mathbf{y} = (1, \frac{1}{2}, 0)^\top$ a pro $i = 3$ je $\mathbf{y} = (0, 1, \frac{1}{3})^\top$. Inverzní matice je stejná jako v předchozí úloze.
5. $\det \mathbf{L} = 1$, $\det \mathbf{U} = 1$ a $\det \mathbf{A} = 1$.
6. $\det \mathbf{P} = 1$, $\det \mathbf{L} = 1$, $\det \mathbf{U} = 1$ a $\det \mathbf{A} = 1$.
7. Zhruba $\mathcal{O}(\frac{2}{3}n^3)$ operací.

3.5 Maticové normy a podmíněnost matic

V první kapitole jsme Definicí 1.2.3 zavedli číslo podmíněnosti vyjadřující citlivost úlohy na různé typy poruch (chyby). Protože jsme uvažovali velmi jednoduché úlohy, stačilo posuzovat velikost chyby pomocí absolutní hodnoty. Nyní ukážeme jak se číslo podmíněnosti počítá u soustav lineárních rovnic. Budeme přitom potřebovat zobecnění absolutní hodnoty pro matice a vektory, které zavádí následující definice.

Definice 3.5.1 Norma matice je zobrazení, které každé matici $\mathbf{A} = (a_{ij})$ typu $m \times n$ přiřadí číslo $\|\mathbf{A}\|$ tak, že platí:

(i) $\|\mathbf{A}\| \geq 0$ a přitom $\|\mathbf{A}\| = 0$, právě když $\mathbf{A}$ je matice nulová;

(ii) $\|\alpha\mathbf{A}\| = |\alpha| \cdot \|\mathbf{A}\|$ pro každé reálné číslo α;

(iii) $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ pro každou matici $\mathbf{B}$ stejného typu jako je matice $\mathbf{A}$.

Základní maticové normy jsou:

- *řádková norma*: $\|\mathbf{A}\|_R = \max\limits_{i=1,\ldots,m} \sum\limits_{j=1}^{n} |a_{ij}|$;

- *sloupcová norma*: $\|\mathbf{A}\|_S = \max\limits_{j=1,\dots,n} \sum\limits_{i=1}^{m} |a_{ij}|$;
- *Frobeniova norma*: $\|\mathbf{A}\|_F = \sqrt{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} a_{ij}^2}$.

Příklad 3.5.1 Vypočítejte řádkovou, sloupcovou a Frobeniovu normu pro matici

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 2 \\ -1 & 5 & -4 \end{pmatrix}$$

a pro matici inverzní $\mathbf{A}^{-1}$.

Řešení: Dostáváme

$$\begin{aligned} \|\mathbf{A}\|_R &= \max\{1+1+1, 2+4+2, 1+5+4\} = 10, \\ \|\mathbf{A}\|_S &= \max\{1+2+1, 1+4+5, 1+2+4\} = 10, \\ \|\mathbf{A}\|_F &= \sqrt{1+1+1+4+16+4+1+25+16} = \sqrt{69} \doteq 8.3066. \end{aligned}$$

Matici inverzní $\mathbf{A}^{-1}$ známe z Příkladu 3.4.2. Pomocí tohoto výsledku dostaneme:

$$\|\mathbf{A}^{-1}\|_R = \frac{37}{6}, \quad \|\mathbf{A}^{-1}\|_S = \frac{23}{3}, \quad \|\mathbf{A}^{-1}\|_F \doteq 5.38.$$

□

Věta 3.5.1 Nechť $\mathbf{A}$ je matice typu $m \times n$ a $\mathbf{B}$ je matice typu $n \times p$. Pro řádkovou, sloupcovou a Frobeniovu normu platí:

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|. \tag{3.9}$$

Důkaz: Platnost tvrzení ukážeme pouze pro řádkovou normu, ostatní případy ponecháme jako cvičení. Prvky matice součinu $\mathbf{C} = \mathbf{AB}$ jsou určeny předpisem $c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}$. Proto

$$\begin{aligned} \|\mathbf{AB}\|_R &= \max_{i=1,\dots,m} \sum_{j=1}^{p} |c_{ij}| = \max_{i=1,\dots,m} \sum_{j=1}^{p} |\sum_{k=1}^{n} a_{ik}b_{kj}| \leq \\ &\leq \max_{i=1,\dots,m} \sum_{j=1}^{p}\sum_{k=1}^{n} |a_{ik}||b_{kj}| = \max_{i=1,\dots,m} \sum_{k=1}^{n} |a_{ik}| \sum_{j=1}^{p} |b_{kj}| \\ &\leq \max_{i=1,\dots,m} \sum_{k=1}^{n} |a_{ik}| \max_{l=1,\dots,n} \sum_{j=1}^{p} |b_{lj}| = \|\mathbf{A}\|_R\|\mathbf{B}\|_R. \end{aligned}$$

□

Definice 3.5.2 Číslo podmíněnosti regulární čtvercové matice $\mathbf{A}$ je definováno předpisem

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|.$$

Příklad 3.5.2 Vypočtěte číslo podmíněnosti matice z Příkladu 3.5.1 pomocí řádkové, sloupcové a Frobeniovy normy.

Řešení: S využitím výsledků Příkladu 3.5.1 dostaneme: $\kappa_R(\mathbf{A}) \doteq 61.67$, $\kappa_S(\mathbf{A}) \doteq 76.67$, $\kappa_F(\mathbf{A}) \doteq 44.69$. □

Věta 3.5.2 Nechť $\mathbf{A}$ je regulární čtvercová matice řádu n a nechť $\mathbf{b}$ a $\mathbf{x}$ jsou nenulové n-složkové vektory takové, že platí:

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

Dále nechť $\tilde{\mathbf{b}}$ a $\tilde{\mathbf{x}}$ jsou n-složkové vektory takové, že platí

$$\mathbf{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}.$$

Potom

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \kappa(\mathbf{A}) \frac{\|\mathbf{b} - \tilde{\mathbf{b}}\|}{\|\mathbf{b}\|}. \tag{3.10}$$

Důkaz: Zřejmě platí

$$\mathbf{b} = \mathbf{A}\mathbf{x} \quad \text{resp.} \quad \mathbf{x} - \tilde{\mathbf{x}} = \mathbf{A}^{-1}(\mathbf{b} - \tilde{\mathbf{b}})$$

a pomocí (3.9) odtud dostaneme

$$\|\mathbf{x}\|^{-1} \leq \|\mathbf{A}\|\|\mathbf{b}\|^{-1} \quad \text{resp.} \quad \|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\mathbf{A}^{-1}\|\|\mathbf{b} - \tilde{\mathbf{b}}\|.$$

Vynásobením těchto nerovností odvodíme tvrzení (3.10). □

Nerovnost (3.10) říká, že při velké hodnotě $\kappa(\mathbf{A})$ může malá porucha ve vektoru $\mathbf{b}$ vyvolat velkou změnu v řešení. Výpočty s maticí, která má velké číslo podmíněnosti, jsou zpravidla znehodnoceny kumulací zaokrouhlovacích chyb, jak ukazuje následující příklad.

Příklad 3.5.3 Vypočteme čísla podmíněnosti *Hilbertovy* matice

$$\mathbf{A} = \begin{pmatrix} 1 & 1/2 & 1/3 & \dots \\ 1/2 & 1/3 & 1/4 & \dots \\ 1/3 & 1/4 & 1/5 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

řádů $n = 5, 10, 15, 20, 25$ a pokusíme se vypočítat inverzní matici.

n	$\kappa(\mathbf{A})$	$\|\mathbf{A}\mathbf{A}^{-1} - \mathbf{I}\|$
5	4.8×10^{5}	1.4×10^{-11}
10	1.6×10^{13}	3.3×10^{-3}
15	1.1×10^{18}	2.8×10^{3}
20	2.5×10^{28}	2.6×10^{11}
25	1.0×10^{36}	1.3×10^{19}

Tabulka 3.1: Podmíněnost Hilbertovy matice $\mathbf{A}$.

Řešení: Čísla podmíněnosti jsou zaznamenána v Tabulce 3.1. Poslední sloupec tabulky ukazuje, jak se (na počítači) podařilo vypočítat inverzní matice. Je vidět, že pro řád $n = 15$ a vyšší jsou výsledky naprosto nesmyslné. □

Kontrolní otázky

Otázka 1. Jak se definuje norma matice?
Otázka 2. Co vyjadřuje číslo podmíněnosti matice?

Úlohy k samostatnému řešení

1. Pro matici

$$\mathbf{A} = \begin{pmatrix} -1 & -3 & 2 \\ -6 & -19 & 10 \\ 3 & 9 & -5 \end{pmatrix}.$$

vypočtěte číslo podmíněnosti pomocí řádkové, sloupcové a Frobeniovy normy.

Výsledky úloh k samostatnému řešení

1. $\|\mathbf{A}\|_R = 35$, $\|\mathbf{A}^{-1}\|_R = 16$, $\kappa_R(\mathbf{A}) = 560$; $\|\mathbf{A}\|_S = 31$, $\|\mathbf{A}^{-1}\|_S = 11$, $\kappa_S(\mathbf{A}) = 341$; $\|\mathbf{A}\|_F = 25.02$, $\|\mathbf{A}^{-1}\|_F = 10.63$, $\kappa_F(\mathbf{A}) = 265.96$.

KAPITOLA

4

SOUSTAVY LINEÁRNÍCH ROVNIC: ITERAČNÍ METODY

4.1 Příklad iteračního výpočtu

Iterační metody umožňují řešit soustavy lineárních rovnic pomocí postupného přibližování k přesnému řešení. Počítá se posloupnost vektorů aproximací $\{\mathbf{x}^{(k)}\}$ taková, že

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \bar{\mathbf{x}}, \quad \text{kde } \bar{\mathbf{x}} \text{ je řešením } \mathbf{A}\mathbf{x} = \mathbf{b}.$$

Výhody iteračních metod jsou tyto:

- *V každé iteraci známe aproximaci řešení* $\boldsymbol{x}^{(k)}$. Pokud je tato aproximace dostatečně přesná, pak výpočet ukončíme.
- *V každé iteraci je nejpracnější operací násobení matice a vektoru.* Jedná se o operaci, která je algoritmicky podstatně jednodušší než Gaussova eliminační metoda a lze ji snadno provést i pro rozsáhlé řídké matice, tj. pro matice s velkým počtem (neuložených) nulových prvků.
- *Iterační metody jsou méně citlivé na zaokrouhlovací chyby než metody přímé.* Na každou iteraci můžeme nahlížet jako na počáteční. Zaokrouhlovací chyby z předchozích iterací proto vymizí, pokud v dalším výpočtu dojde ke konvergenci. Některé speciální iterační metody byly navrženy pro zpřesnění výsledků vypočítaných pomocí přímých metod.

Zhruba platí následující dělení: přímé metody se používají, je-li matice soustavy malá ($1 \leq n \leq 100000$), plná a dobře podmíněná; iterační metody se používají pro velké soustavy ($n > 100000$) s řídkou maticí.

Nejdříve si ukážeme dva příklady iteračního řešení soustavy lineárních rovnic, v nichž uvidíme, že výpočet může konvergovat i divergovat. Budeme řešit soustavu:

$$\begin{aligned} 11x_1 + 2x_2 + x_3 &= 15, \\ x_1 + 10x_2 + 2x_3 &= 16, \\ 2x_1 + 3x_2 - 8x_3 &= 1, \end{aligned} \quad \text{resp.} \quad \begin{pmatrix} 11 & 2 & 1 \\ 1 & 10 & 2 \\ 2 & 3 & -8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 15 \\ 16 \\ 1 \end{pmatrix}. \tag{4.1}$$

Soustavu převedeme na tvar vhodný pro výpočet iterací, tzv. *iterační tvar*. Provádí se to například tak, že z každé rovnice vyjádříme jednu neznámou:

$$\begin{aligned} x_1 &= \tfrac{1}{11}(15 - 2x_2 - x_3), \\ x_2 &= \tfrac{1}{10}(16 - x_1 - 2x_3), \\ x_3 &= \tfrac{1}{8}(-1 + 2x_1 + 3x_2), \end{aligned} \tag{4.2}$$

tj.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & -\frac{2}{11} & -\frac{1}{11} \\ -\frac{1}{10} & 0 & -\frac{1}{5} \\ \frac{1}{4} & \frac{3}{8} & 0 \end{pmatrix}}_{\mathbf{C}_J} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \underbrace{\begin{pmatrix} \frac{15}{11} \\ \frac{8}{5} \\ -\frac{1}{8} \end{pmatrix}}_{\mathbf{d}_J}.$$

Jiná možnost:

$$\begin{aligned} x_1 &= 15 - 10x_1 - 2x_2 - x_3, \\ x_2 &= 16 - x_1 - 9x_2 - 2x_3, \\ x_3 &= -1 + 2x_1 + 3x_2 - 7x_3, \end{aligned} \tag{4.3}$$

tj.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\begin{pmatrix} -10 & -2 & -1 \\ -1 & -9 & -2 \\ 2 & 3 & -7 \end{pmatrix}}_{\mathbf{C}_B} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \underbrace{\begin{pmatrix} 15 \\ 16 \\ -1 \end{pmatrix}}_{\mathbf{d}_B}.$$

Takových převodů existuje zřejmě nekonečně mnoho, ale jenom některé povedou ke konvergentnímu výpočtu.

Z rovnic (4.2) a (4.3) dostaneme rekurentní vzorce připsáním iteračního indexu $k+1$ k neznámým na levé straně a k k neznámým na pravé straně. Dostáváme

$$\left.\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{11}(15 - 2x_2^{(k)} - x_3^{(k)}), \\ x_2^{(k+1)} &= \tfrac{1}{10}(16 - x_1^{(k)} - 2x_3^{(k)}), \\ x_3^{(k+1)} &= \tfrac{1}{8}(-1 + 2x_1^{(k)} + 3x_2^{(k)}), \end{aligned}\right\} \quad \text{tj.} \quad \mathbf{x}^{(k+1)} = \mathbf{C}_J\mathbf{x}^{(k)} + \mathbf{d}_J, \tag{4.4}$$

a

$$\left.\begin{aligned} x_1^{(k+1)} &= 15 - 10x_1^{(k)} - 2x_2^{(k)} - x_3^{(k)}, \\ x_2^{(k+1)} &= 16 - x_1^{(k)} - 9x_2^{(k)} - 2x_3^{(k)}, \\ x_3^{(k+1)} &= -1 + 2x_1^{(k)} + 3x_2^{(k)} - 7x_3^{(k)}, \end{aligned}\right\} \quad \text{tj.} \quad \mathbf{x}^{(k+1)} = \mathbf{C}_B\mathbf{x}^{(k)} + \mathbf{d}_B. \tag{4.5}$$

Nyní zvolíme počáteční aproximaci $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)})^\top$, např. $\mathbf{x}^{(0)} = (0,0,0)^\top$. Tyto hodnoty dosadíme do pravé strany rekurentních vzorců (4.4) a dostaneme

$$\mathbf{x}^{(1)} = \left(\tfrac{15}{11}, \ \tfrac{16}{10}, \ -\tfrac{1}{8} \right)^\top.$$

Jestliže takto pokračujeme dál, dostáváme

$$\begin{aligned} \mathbf{x}^{(2)} &= (1.0841, 1.4886, 0.81591)^\top, \\ \mathbf{x}^{(3)} &= (1.0188, 1.3284, 0.70426)^\top, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{aligned}$$

Provedeme-li několik dalších iterací, zjistíme, že se číslice na prvních desetinných místech začnou po chvíli opakovat. Dostaneme přitom vektor

$$\mathbf{x} = (1.0564, 1.3642, 0.65069)^\top,$$

o němž se můžeme domnívat, že je aproximací přesného řešení soustavy (4.1).

Jestliže analogicky počítáme podle rekurentních vzorců (4.5), dostaneme

$$\begin{aligned} \mathbf{x}^{(1)} &= (15, 16, -1)^\top, \\ \mathbf{x}^{(2)} &= (-166, -141, 84)^\top, \\ \mathbf{x}^{(3)} &= (1873, 1283, -1344)^\top, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{aligned}$$

Zde žádnou tendenci ke konvergenci nevidíme a je proto pravděpodobné, že posloupnost iterací diverguje.

Kontrolní otázky

Otázka 1. V čem spočívá základní rozdíl mezi přímými a iteračními metodami?
Otázka 2. Jaké jsou výhody a nevýhody přímých a iteračních metod?

Úlohy k samostatnému řešení

1. Pro soustavu lineárních rovnic

$$\begin{aligned} -x_1 - 3x_2 + 2x_3 &= -9, \\ -6x_1 - 19x_2 + 10x_3 &= -59, \\ 3x_1 + 9x_2 - 5x_3 &= 28 \end{aligned}$$

navrhněte dva iterační tvary pomocí postupů z odstavce 4.1.
2. U kterého z navržených iteračních tvarů výpočet konverguje?

Výsledky úloh k samostatnému řešení

1. Rekurentní vzorce pro první iterační tvar:

$$\begin{aligned} x_1^{(k+1)} &= 9 - 3x_2^{(k)} + 2x_3^{(k)}, \\ x_2^{(k+1)} &= \tfrac{1}{19}(59 - 6x_1^{(k)} + 10x_3^{(k)}), \\ x_3^{(k+1)} &= \tfrac{1}{5}(-28 + 3x_1^{(k)} + 9x_2^{(k)}); \end{aligned}$$

rekurentní vzorce pro druhý iterační tvar:

$$\begin{aligned} x_1^{(k+1)} &= 9 - 3x_2^{(k)} + 2x_3^{(k)}, \\ x_2^{(k+1)} &= 59 - 6x_1^{(k)} - 18x_2^{(k)} + 10x_3^{(k)}, \\ x_3^{(k+1)} &= -28 + 3x_1^{(k)} + 9x_2^{(k)} - 4x_3^{(k)}. \end{aligned}$$

2. Jestliže zkusíme výpočet provést, zjistíme, že dochází k divergenci v obou případech. Rozpoznáním konvergentního výpočtu z vlastností matice soustavy se budeme zabývat v dalších odstavcích.

4.2 Obecná iterační metoda

Ukážeme si obecný (lineární) iterační postup řešení soustav lineárních rovnic a uvedeme jeho dvě základní varianty nazývané Jakobiova a Gauss-Seidelova iterační metoda.

Uvažujme soustavu lineárních rovnic

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4.6}$$

s regulární čtvercovou maticí $\mathbf{A} = (a_{ij})$ řádu n, vektorem pravé strany $\mathbf{b} = (b_i)$ a vektorem neznámých $\mathbf{x} = (x_i)$. Soustavu (4.6) převedeme na ekvivalentní soustavu v *iteračním tvaru*:

$$\mathbf{x} = \mathbf{C}\mathbf{x} + \mathbf{d}, \tag{4.7}$$

kde $\mathbf{C}$ je *iterační matice* řádu n a $\mathbf{d}$ je sloupcový vektor. Musí přitom platit, že rovnice (4.6) a (4.7) mají stejná řešení.

Nechť $\mathbf{x}^{(0)}$ je daná počáteční aproximace. Iterační výpočet provádíme podle rekurentního vzorce

$$\mathbf{x}^{(k+1)} = \mathbf{C}\mathbf{x}^{(k)} + \mathbf{d}, \quad k = 0, 1, 2, \ldots \tag{4.8}$$

Jestliže posloupnost vektorů $\{\mathbf{x}^{(k)}\}$ konverguje k vektoru $\bar{\mathbf{x}}$, pak limitním přechodem v (4.8) dostaneme, že $\bar{\mathbf{x}}$ je řešením rovnice (4.7) a také (4.6).

Jak uvidíme později, volba počáteční aproximace neovlivní konvergenci, takže vektor $\mathbf{x}^{(0)}$ můžeme zvolit libovolně. Výpočet ukončíme, jestliže dvě poslední aproximace se od sebe liší ne více, než kolik udává požadovaná přesnost, tj. jestliže je splněno ukončovací kritérium

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \epsilon, \tag{4.9}$$

kde $\epsilon > 0$ je vhodné malé číslo a $\|\cdot\|$ je zvolená norma. V našich příkladech použijeme ukončovací kritérium s řádkovou normou.

Algoritmus: Obecná iterační metoda
Vstup: $\mathbf{C}$, $\mathbf{d}$, $\mathbf{x}^{(0)}$, ϵ.
Pro $k = 1, 2, \ldots$ opakuj:
    $\mathbf{x}^{(k+1)} := \mathbf{C}\mathbf{x}^{(k)} + \mathbf{d}$;
dokud $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_R > \epsilon$.
Výstup: $\bar{\mathbf{x}} = \mathbf{x}^{(k)} \pm \epsilon$.

4.2.1 Jakobiova iterační metoda

Jakobiovu metodu jsme si již ukázali při řešení soustavy (4.1) v Odstavci 4.1. Jsou to rekurentní vzorce (4.4). Nyní si je projdeme obecně.

Budeme předpokládat, že diagonální prvky matice soustavy (4.6) jsou nenulové, tj. $a_{ii} \neq 0$. Z i-té rovnice

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i, \quad i = 1, \ldots, n,$$

vyjádříme i-tou neznámou

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

Jakobiova iterační metoda je určena rekurentními vzorci

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} \right), \quad i = 1, \ldots, n, \tag{4.10}$$

pro $k = 0, 1, 2, \ldots$.

Všimněme si ještě, že vzorce (4.10) můžeme zapsat v obecném maticovém tvaru (4.8), jestliže položíme $\mathbf{C} = \mathbf{C}_J$ a $\mathbf{d} = \mathbf{d}_J$, kde

$$\mathbf{C}_J = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \cdots & -\frac{a_{2n}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0 & \cdots & -\frac{a_{3n}}{a_{33}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \cdots & 0 \end{pmatrix}, \quad \mathbf{d}_J = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \frac{b_3}{a_{33}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}. \tag{4.11}$$

Pomocí aditivního rozkladu matice $\mathbf{A} = (a_{ij})$,

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}, \tag{4.12}$$

kde $\mathbf{L} = (l_{ij})$, $l_{ij} = a_{ij}$, $i > j$, $l_{ij} = 0$, $i \leq j$, je dolní trojúhelníková část, $\mathbf{D} = (d_{ij})$, $d_{ii} = a_{ii}$, $d_{ij} = 0$, $i \neq j$, je diagonální část a $\mathbf{U} = (u_{ij})$, $u_{ij} = 0$, $i \geq j$, $u_{ij} = a_{ij}$, $i < j$, je horní trojúhelníková část, můžeme stručně psát

$$\mathbf{C}_J = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}), \quad \mathbf{d}_J = \mathbf{D}^{-1}\mathbf{b}.$$

Příklad 4.2.1 Soustavu lineárních rovnic (4.1) řešte pomocí Jakobiovy iterační metody s přesností $\epsilon = 10^{-4}$.

Řešení: Výpočet se provádí podle rekurentních vzorců (4.4). Začátek výpočtu jsme naznačili v odstavci 4.1. Nyní vše shrneme v Tabulce 4.1, kde kromě aproximací $\mathbf{x}^{(k)}$ uvádíme řádkové normy $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_R$. Naznačme ještě výpočet prvních dvou norem:

$$\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|_R = \max\{|\tfrac{15}{11} - 0|, |\tfrac{16}{10} - 0|, |-\tfrac{1}{8} - 0|\} = 1.6,$$
$$\|\mathbf{x}^{(2)} - \mathbf{x}^{(1)}\|_R = \max\{|1.0841 - \tfrac{15}{11}|, |1.4886 - \tfrac{16}{10}|, |0.8159 + \tfrac{1}{8}|\} = 0.9409,$$
atd.

Výpočet jsme ukončili po desáté iteraci, protože $\|\mathbf{x}^{(10)} - \mathbf{x}^{(9)}\|_R = 0.00005 \leq 10^{-4}$, a výsledek je $\bar{x}_1 = 1.0564 \pm 10^{-4}$, $\bar{x}_2 = 1.3642 \pm 10^{-4}$, $\bar{x}_3 = 0.6507 \pm 10^{-4}$. □

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_R$
0	0	0	0	—
1	1.3636	1.6000	−0.1250	1.60000
2	1.0841	1.4886	0.8159	0.94091
3	1.0188	1.3284	0.7043	0.16023
4	1.0581	1.3573	0.6279	0.07641
5	1.0598	1.3686	0.6485	0.02064
6	1.0558	1.3643	0.6532	0.00468
7	1.0562	1.3638	0.6506	0.00260
8	1.0565	1.3643	0.6505	0.00048
9	1.0565	1.3643	0.6507	0.00027
10	1.0564	1.3642	0.6507	0.00005

Tabulka 4.1: Iterace Jakobiovy iterační metody.

4.2.2 Gauss-Seidelova iterační metoda

Začneme příkladem. Pro řešení soustavy lineárních rovnic (4.1) jsme použili Jakobiovu metodu, která je určena rekurentními vzorci (4.4). Podle těchto vzorců se počítají v k-té iteraci složky nové aproximace $\mathbf{x}^{(k+1)}$ postupně, tj. nejdříve $x_1^{(k+1)}$ pak $x_2^{(k+1)}$ a nakonec $x_3^{(k+1)}$. Přitom se stále používají složky z předchozí aproximace, tj. $x_1^{(k)}$, $x_2^{(k)}$ a $x_3^{(k)}$. Tento postup můžeme snadno vylepšit. Stačí si uvědomit, že při výpočtu $x_2^{(k+1)}$ můžeme použít přesnější aproximaci $x_1^{(k+1)}$ namísto méně přesné $x_1^{(k)}$. Podobně můžme při výpočtu $x_3^{(k+1)}$ použít přesnější aproximace $x_1^{(k+1)}$ a $x_2^{(k+1)}$ namísto méně přesných $x_1^{(k)}$ a $x_2^{(k)}$. Původní rekurentní vzorce (4.1) se tak změní na tvar:

$$\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{11}(15 - 2x_2^{(k)} - x_3^{(k)}), \\ x_2^{(k+1)} &= \tfrac{1}{10}(16 - x_1^{(k+1)} - 2x_3^{(k)}), \\ x_3^{(k+1)} &= \tfrac{1}{8}(-1 + 2x_1^{(k+1)} + 3x_2^{(k+1)}), \end{aligned} \tag{4.13}$$

což je Gauss-Seidelova iterační metoda.

Příklad 4.2.2 Soustavu lineárních rovnic (4.1) řešte pomocí Gauss-Seidelovy iterační metody s přesností $\epsilon = 10^{-4}$.

Řešení: Výpočet podle vzorců (4.13) je zaznamenán v Tabulce 4.2. Výpočet jsme ukončili už po sedmé iteraci, protože $\|\mathbf{x}^{(7)} - \mathbf{x}^{(6)}\|_R = 0.00001 \leq 10^{-4}$, a výsledek je $\bar{x}_1 = 1.0564 \pm 10^{-4}$, $\bar{x}_2 = 1.3642 \pm 10^{-4}$, $\bar{x}_3 = 0.6507 \pm 10^{-4}$. □

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_R$
0	0	0	0	—
1	1.3636	1.4636	0.7648	1.46364
2	1.0280	1.3442	0.6361	0.33564
3	1.0614	1.3666	0.6528	0.03341
4	1.0558	1.3639	0.6504	0.00559
5	1.0565	1.3643	0.6507	0.00073
6	1.0564	1.3642	0.6507	0.00011
7	1.0564	1.3642	0.6507	0.00001

Tabulka 4.2: Iterace Gauss-Seidelovy iterační metody.

Poznámka

Z Příkladů 4.2.1 a 4.2.2 je vidět, že Gauss-Seidelova iterační metoda je rychlejší než metoda Jakobiova. Existují ale příklady, kdy Jakobiova iterační metoda konverguje, zatímco Gauss-Seidelova iterační metoda diverguje.

Pro obecnou soustavu (4.6), kde $a_{ii} \neq 0$, je *Gauss-Seidelova iterační metoda* určena rekurentními vzorci

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right), \quad i = 1, \ldots, n \tag{4.14}$$

pro $k = 0, 1, 2, \ldots$. Na první pohled ale není vidět, jak tyto vzorce zapsat v maticovém tvaru (4.8). Podívejme se proto ještě jednou na náš příklad. Jestliže v (4.13) převedeme na levou stranu všechny členy obsahující složky nové aproximace, dostaneme

$$\begin{aligned} 11x_1^{(k+1)} &= 15 - 2x_2^{(k)} - x_3^{(k)}, \\ x_1^{(k+1)} + 10x_2^{(k+1)} &= 16 - 2x_3^{(k)}, \\ -2x_1^{(k+1)} - 3x_2^{(k+1)} + 8x_3^{(k+1)} &= -1. \end{aligned}$$

Odtud vidíme, že $\mathbf{x}^{(k+1)}$ vznikne z $\mathbf{x}^{(k)}$ řešením soustavy lineárních rovnic s dolní trojúhelníkovou maticí $\mathbf{L} + \mathbf{D}$. Pomocí aditivního rozkladu (4.12) matice $\mathbf{A}$ to zapíšeme jako

$$(\mathbf{L} + \mathbf{D})\mathbf{x}^{(k+1)} = -\mathbf{U}\mathbf{x}^{(k)} + \mathbf{b}$$

a po úpravě

$$\mathbf{x}^{(k+1)} = -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U}\mathbf{x}^{(k)} + (\mathbf{L} + \mathbf{D})^{-1}\mathbf{b}.$$

Obecné rekurentní vzorce (4.8) představují Gauss-Seidelovu iterační metodu, když v nich položíme $\mathbf{C} = \mathbf{C}_{GS}$ a $\mathbf{d} = \mathbf{d}_{GS}$, kde

$$\mathbf{C}_{GS} = -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U}, \quad \mathbf{d}_{GS} = (\mathbf{L} + \mathbf{D})^{-1}\mathbf{b}.$$

Kontrolní otázky

Otázka 1. Jak vypadá obecné schéma iteračního řešení soustav lineárních rovnic?
Otázka 2. Jak se provádí výpočet u Jakobiovy a Gauss-Seidelovy iterační metody? Která z nich je rychlejší?

Úlohy k samostatnému řešení

1. Soustavu lineárních rovnic

$$\begin{aligned} 4x_1 - x_2 + 2x_3 &= -12, \\ 2x_1 + 5x_2 + x_3 &= 5, \\ x_1 + x_2 - 3x_3 &= -4 \end{aligned}$$

řešte pomocí Jakobiovy iterační metody s přesností $\epsilon = 10^{-2}$.
2. V předchozí úloze použijte při řešení Gauss-Seidelovu metodu.
3. Upravte obecný algoritmus pro iterační řešení soustav lineárních rovnic tak, aby vyjadřoval Jakobiovu resp. Gauss-Seidelovu iterační metodu.

Výsledky úloh k samostatnému řešení

1. Rekurentní vzorce pro Jakobiovu iterační metodu mají tvar

$$\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{4}(-12 + x_2^{(k)} - 2x_3^{(k)}), \\ x_2^{(k+1)} &= \tfrac{1}{5}(5 - 2x_1^{(k)} - x_3^{(k)}), \\ x_3^{(k+1)} &= \tfrac{1}{3}(4 + x_1^{(k)} + x_2^{(k)}). \end{aligned}$$

Při nulové počáteční aproximaci dojdeme na požadovanou přesnost v jedenácté iteraci; $x_1 = -2.9955 \pm 10^{-2}$, $x_2 = 2.0019 \pm 10^{-2}$, $x_3 = 1.0011 \pm 10^{-2}$.
2. Rekurentní vzorce pro Gauss-Seidelovu metodu mají tvar

$$\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{4}(-12 + x_2^{(k)} - 2x_3^{(k)}), \\ x_2^{(k+1)} &= \tfrac{1}{5}(5 - 2x_1^{(k+1)} - x_3^{(k)}), \\ x_3^{(k+1)} &= \tfrac{1}{3}(4 + x_1^{(k+1)} + x_2^{(k+1)}). \end{aligned}$$

Při nulové počáteční aproximaci dojdeme na požadovanou přesnost ve čtvrté iteraci; $x_1 = -2.9991 \pm 10^{-2}$, $x_2 = 1.9998 \pm 10^{-2}$, $x_3 = 1.0002 \pm 10^{-2}$.
3. Vstupní parametry $\mathbf{C}$ a $\mathbf{d}$ u původního algoritmu nahradíme za $\mathbf{A} = (a_{ij})$ a $\mathbf{b} = (b_i)$. Maticový výpočet nové iterace $\mathbf{x}^{(k+1)}$ zapíšeme rekurentními vzorci (4.10) resp. (4.14).

4.3 Vlastní čísla a vlastní vektory matic

Pro analýzu konvergence iteračních metod potřebujeme některé informace o vlastních číslech a vlastních vektorech. Na rozdíl od předchozích odstavců, zde musíme nutně pracovat s maticemi, které jsou singulární. Úlohu na výpočet vlastních čísel a vlastních vektorů připomeneme nejdříve pomocí příkladu.

Příklad 4.3.1 Uvažujme matici

$$\mathbf{A} = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}. \tag{4.15}$$

Určete taková čísla λ, pro která má soustava lineárních rovnic $\mathbf{Av} = \lambda\mathbf{v}$ nenulové řešení $\mathbf{v}$ a tato řešení vypočtěte.

Řešení: Soustavu $\mathbf{Av} = \lambda\mathbf{v}$ přepíšeme do tvaru $(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$:

$$\begin{pmatrix} 2-\lambda & 0 & 0 \\ 2 & 2-\lambda & 1 \\ 1 & 1 & 2-\lambda \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Odtud je vidět, že jedním z řešení je nulový vektor. Nás však zajímají řešení nenulová. Matice soustavy musí tedy mít více než jedno řešení, takže její determinant je nulový, tj. $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$. Pomocí tohoto vztahu můžeme určit vlastní čísla λ. Snadným výpočtem zjistíme, že platí

$$\det(\mathbf{A} - \lambda\mathbf{I}) = -\lambda^3 + 6\lambda^2 - 11\lambda + 6 = 0.$$

Dostali jsme algebraickou rovnici třetího stupně. Pouze pro její tři kořeny $\lambda_1 = 3$, $\lambda_2 = 2$ a $\lambda_3 = 1$ bude matice $\mathbf{A} - \lambda\mathbf{I}$ singulární a odpovídající soustavy lineárních rovnic budou mít nenulová řešení. Jedná se o tyto soustavy:

$$(\mathbf{A} - 3\mathbf{I})\mathbf{v} = \mathbf{0}, \quad (\mathbf{A} - 2\mathbf{I})\mathbf{v} = \mathbf{0}, \quad (\mathbf{A} - 1\mathbf{I})\mathbf{v} = \mathbf{0},$$

tj.

$$\begin{pmatrix} -1 & 0 & 0 \\ 2 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Vyřešením těchto soustav dostaneme

$$\mathbf{v}_1 = (0, r, r)^\top, \quad \mathbf{v}_2 = (s, -s, -2s)^\top, \quad \mathbf{v}_3 = (0, t, -t)^\top,$$

kde r, s a t jsou libovolná nenulová čísla. Vidíme, že každá soustava má nekonečně mnoho řešení. Konkrétní volbou r, s a t dostaneme například

$$\mathbf{v}_1 = (0, 1, 1)^\top, \quad \mathbf{v}_2 = (1, -1, -2)^\top, \quad \mathbf{v}_3 = (0, 1, -1)^\top.$$

Všimněme si ještě, že čísla λ_1, λ_2 a λ_3 jsou vzájemně různá a že vektory $\mathbf{v}_1$, $\mathbf{v}_2$ a $\mathbf{v}_3$ jsou lineárně nezávislé. □

Definice 4.3.1 Nechť $\mathbf{A}$ je čtvercová matice řádu n. Číslo λ (obecně komplexní), pro které má soustava

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \quad \text{resp.} \quad (\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$$

nenulové řešení, se nazývá *vlastní číslo* matice $\mathbf{A}$ a jemu odpovídající nenulové řešení $\mathbf{v} = (v_1, v_2, \ldots, v_n)^\top$ se nazývá *vlastní vektor* matice $\mathbf{A}$.

Je zřejmé, že číslo λ je vlastním číslem matice $\mathbf{A}$ právě tehdy, když je kořenem *charakteristického polynomu*

$$p_{\mathbf{A}}(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = (-1)^n\lambda^n + c_1\lambda^{n-1} + \cdots + c_{n-1}\lambda + c_n.$$

Odtud plyne, že každá čtvercová matice řádu n má právě n vlastních čísel, pokud každé vlastní číslo počítáme tolikrát, kolik činí násobnost příslušného kořene, a pokud bereme do úvahy i kořeny komplexní.

Poznámka

Vlastní čísla můžeme hledat jako řešení rovnice $p_{\mathbf{A}}(\lambda) = 0$ metodami z Kapitoly 2. Tento postup je však prakticky neproveditelný pro větší hodnoty n, protože výpočet koeficientů charakteristického polynomu c_i založený na determinantech vyžaduje velký počet aritmetických operací.

Snadno lze určit vlastní čísla u horní trojúhelníkové matice $\mathbf{U} = (u_{ij})$, $u_{ij} = 0$ pro $i > j$. Jsou to všechny diagonální prvky u_{ii}, protože z definice determinantu plyne, že charakteristický polynom má tvar

$$p_{\mathbf{U}}(\lambda) = (u_{11} - \lambda)(u_{22} - \lambda)\ldots(u_{nn} - \lambda).$$

Analogické tvrzení platí i pro dolní trojúhelníkovou matici.

Při vyšetřování konvergence iteračních metod budeme využívat následující větu

Věta 4.3.1 Nechť λ je vlastní číslo matice $\mathbf{A}$, které odpovídá vlastnímu vektoru $\mathbf{v}$, c je dané reálné číslo a k je číslo přirozené. Potom $c\lambda^k$ je vlastní číslo matice $c\mathbf{A}^k$, kterému odpovídá vlastní vektor $\mathbf{v}$.

Důkaz: Jestliže $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$, potom $c\mathbf{A}^k\mathbf{v} = c\lambda\mathbf{A}^{k-1}\mathbf{v} = \cdots = c\lambda^k\mathbf{v}$. □

Nyní si všimneme vlastních vektorů. V úvodním příkladu jsme viděli, že vlastní vektory odpovídající různým vlastním číslům jsou lineárně nezávislé. Toto tvrzení platí obecně, takže matice řádu n, může mít (nejvýše) n lineárně nezávislých vektorů. Tyto vektory pak tvoří bázi v prostoru n-složkových aritmetických vektorů. Následující příklad ukazuje, že matice nemusí mít vždy plný počet lineárně nezávislých vlastních vektorů.

Příklad 4.3.2 Určete vlastní čísla a vlastní vektory pro matice

$$\mathbf{A} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}.$$

Řešení: Všechny matice jsou trojúhelníkové a mají stejný charakteristický polynom

$$p_{\mathbf{A}}(\lambda) = p_{\mathbf{B}}(\lambda) = p_{\mathbf{C}}(\lambda) = p_{\mathbf{D}}(\lambda) = (2-\lambda)^3.$$

Číslo $\lambda = 2$ je tedy (trojnásobným) vlastním číslem všech čtyř matic. Postupem z Příkladu 4.3.1 zjistíme, že matice $\mathbf{A}$ má tři lineárně nezávislé vlastní vektory

$$\mathbf{v}_1 = (1,0,0)^\top, \quad \mathbf{v}_2 = (0,1,0)^\top, \quad \mathbf{v}_3 = (0,0,1)^\top$$

(každá nenulová lineární kombinace těchto vektorů je také vlastním vektorem matice $\mathbf{A}$). Pro matici $\mathbf{B}$ se podaří najít pouze dva lineárně nezávislé vlastní vektory $\mathbf{v}_1$ a $\mathbf{v}_3$. Matice $\mathbf{C}$ má opět dva vlastní vektory, nyní to jsou vektory $\mathbf{v}_1$ a $\mathbf{v}_2$. Konečně matice $\mathbf{D}$ má jediný vlastní vektor $\mathbf{v}_1$. □

V aplikacích se často vyskytují symetrické matice, pro něž platí následující tvrzení.

Věta 4.3.2 Nechť $\mathbf{A}$ je symetrická čtvercová matice, tj. $\mathbf{A} = \mathbf{A}^\top$. Potom platí:
(i) všechna vlastní čísla jsou reálná;
(ii) vlastní vektory odpovídající různým vlastním číslům jsou ortogonální;
(iii) k-násobnému vlastnímu číslu odpovídá k lineárně nezávislých vlastních vektorů, které lze zvolit tak, aby byly ortogonální.

Poznámka

Z věty plyne, že pro symetrickou matici řádu n můžeme vždy najít n ortogonálních vlastních vektorů. Protože ortogonální vektory jsou lineárně nezávislé, budou tvořit bázi v prostoru n-složkových aritmetických vektorů.

4.3.1 Výpočet vlastních čísel metodou LU-rozkladu

Ukážeme iterační metodu výpočtu vlastních čísel, která se v literatuře nazývá také LR-algoritmus. Jejím základem jsou vlastnosti podobných matic.

Definice 4.3.2 Dvě čtvercové matice $\mathbf{A}$ a $\mathbf{B}$ řádu n se nazývají *podobné*, jestliže existuje regulární čtvercová matice $\mathbf{C}$ taková, že platí $\mathbf{A} = \mathbf{C}^{-1}\mathbf{B}\mathbf{C}$.

Věta 4.3.3 Podobné matice mají stejná vlastní čísla.

Důkaz: Nechť λ je vlastní číslo matice $\mathbf{A}$ odpovídající vlastnímu vektoru $\mathbf{v}$, tj. platí $\mathbf{Av} = \lambda\mathbf{v}$. Potom

$$\underbrace{\mathbf{C}^{-1}\mathbf{AC}}_{\mathbf{B}}\underbrace{\mathbf{C}^{-1}\mathbf{v}}_{\tilde{\mathbf{v}}} = \mathbf{C}^{-1}\mathbf{Av} = \lambda\underbrace{\mathbf{C}^{-1}\mathbf{v}}_{\tilde{\mathbf{v}}}.$$

Odtud plyne, že λ je vlastní číslo matice $\mathbf{B}$ odpovídající vlastnímu vektoru $\tilde{\mathbf{v}}$. □

Metoda LU-rozkladu je založena na následujícím pozorování. Nechť $\mathbf{L}$ a $\mathbf{U}$ tvoří LU-rozklad matice $\mathbf{A}$ podle Věty 3.3.1, tj. platí $\mathbf{A} = \mathbf{LU}$. Definujme matici $\mathbf{A}_1 = \mathbf{UL}$. Protože

$$\mathbf{A}_1 = \mathbf{UL} = \mathbf{L}^{-1}\mathbf{LUL} = \mathbf{L}^{-1}\mathbf{AL},$$

vidíme, že matice $\mathbf{A}$ a $\mathbf{A}_1$ jsou podobné a mají proto stejná vlastní čísla. Analogicky můžeme k matici $\mathbf{A}_1$ vytvořit podobnou matici $\mathbf{A}_2$ atd. Dostaneme tak posloupnost podobných matic a budeme se zajímat o vlastní čísla limitní matice.

Algoritmus: Metoda LU-rozkladu
Položíme $\mathbf{A}_0 = \mathbf{A}$ a pro $k = 1, 2, \ldots$ dokud nezaznamenáme konvergenci opakujeme:
Krok 1: LU-rozklad matice $\mathbf{A}_{k-1}$, tj. určíme $\mathbf{L}_k$ a $\mathbf{U}_k$ tak, že $\mathbf{A}_{k-1} = \mathbf{L}_k\mathbf{U}_k$;
Krok 2: Vypočítáme součin $\mathbf{A}_k := \mathbf{U}_k\mathbf{L}_k$.

Všimněme si posloupností $\{\mathbf{A}_k\}$ a $\{\mathbf{U}_k\}$. Za jistých předpokladů lze dokázat, že tyto posloupnosti konvergují ke stejné limitní matici $\mathbf{M}$; viz [1]. Tato matice je nutně horní trojúhelníková a má stejná vlastní čísla jako $\mathbf{A}$. Hledaná vlastní čísla proto určíme jako diagonální prvky matice $\mathbf{M}$.

Příklad 4.3.3 Pomocí metody LU-rozkladu vypočtěte vlastní čísla matice

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \tag{4.16}$$

s přesností na dvě desetinná místa.

Řešení: Pro $\mathbf{A}_0 = \mathbf{A}$ určíme LU-rozklad

$$\mathbf{L}_0 = \begin{pmatrix} 1 & 0 & 0 \\ -0.50 & 1 & 0 \\ 0 & -0.67 & 1 \end{pmatrix}, \quad \mathbf{U}_0 = \begin{pmatrix} 2.00 & -1.00 & 0 \\ 0 & 1.50 & -1.00 \\ 0 & 0 & 1.33 \end{pmatrix}$$

a vynásobením dostaneme

$$\mathbf{A}_1 = \mathbf{U}_0\mathbf{L}_0 = \begin{pmatrix} 2.50 & -1.00 & 0 \\ -0.75 & 2.17 & -1.00 \\ 0 & -0.89 & 1.33 \end{pmatrix}.$$

Podobně pro $\mathbf{A}_1$ vypočítáme LU-rozklad

$$\mathbf{L}_1 = \begin{pmatrix} 1 & 0 & 0 \\ -0.30 & 1 & 0 \\ 0 & -0.48 & 1 \end{pmatrix}, \quad \mathbf{U}_1 = \begin{pmatrix} 2.50 & -1.00 & 0 \\ 0 & 1.87 & -1.00 \\ 0 & 0 & 0.86 \end{pmatrix}$$

a opět vynásobením dostaneme

$$\mathbf{A}_2 = \mathbf{U}_1\mathbf{L}_1 = \begin{pmatrix} 2.80 & -1.00 & 0 \\ -0.56 & 2.34 & -1.00 \\ 0 & -0.41 & 0.86 \end{pmatrix}.$$

Čísla pod diagonálou u matic $\mathbf{A}_k$ se začínají přibližovat k nule, což naznačuje, že výpočet bude pravděpodobně konvergovat. Jestliže takto pokračujeme dále, dostaneme

$$\mathbf{A}_{13} = \begin{pmatrix} 3.41 & -1.00 & 0 \\ 0.00 & 2.00 & -1.00 \\ 0 & 0.00 & 0.58 \end{pmatrix} \approx \mathbf{M}.$$

V dalších iteracích se čísla na diagonále (na prvních dvou desetinných místech) nemění. Přibližné hodnoty vlastních čísel jsou $\lambda_1 \doteq 3.41$, $\lambda_2 \doteq 2.00$ a $\lambda_1 \doteq 0.58$.

Kontrolní otázky

Otázka 1. Jak se definují vlastní čísla a vlastní vektory matic?
Otázka 2. Kolik vlastních čísel a vlastních vektorů má matice řádu n?
Otázka 3. Na jaké vlastnosti je založena metoda LU-rozkladu?

Úlohy k samostatnému řešení

1. Pomocí charakteristického polynomu vypočtěte vlastní čísla matice (4.16). Vypočtěte také vlastní vektory.
2. Metodou LU-rozkladu vypočtěte vlastní čísla matice

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{pmatrix}$$

s přesností na čtyři desetinná místa.

Výsledky úloh k samostatnému řešení

1. $p_{\mathbf{A}}(\lambda) = \lambda^3 - 6\lambda^2 + 10\lambda - 4$, $\lambda_1 = 3.414214$, $\lambda_2 = 2$, $\lambda_3 = 0.585786$. Vlastní vektory jsou například $\mathbf{v}_1 = (-1, \sqrt{2}, -1)^\top$, $\mathbf{v}_2 = (-\sqrt{2}, 0, \sqrt{2})^\top$, $\mathbf{v}_3 = (1, \sqrt{2}, 1)^\top$.
2. Vlastní čísla s požadovanou přesností jsou na diagonále matice $\mathbf{A}_{20}$; $\lambda_1 \doteq 3.7320$, $\lambda_2 \doteq 2.0000$, $\lambda_3 \doteq 0.2679$.

4.4 Konvergence iteračních metod

Nyní odvodíme podmínky, které zaručují konvergenci iteračních metod z Odstavce 4.2. Připomeňme, že při iteračním řešení převádíme soustavu lineárních rovnic $\mathbf{A}\mathbf{x} = \mathbf{b}$ na (ekvivalentní) soustavu v iteračním tvaru

$$\mathbf{x} = \mathbf{C}\mathbf{x} + \mathbf{d}. \tag{4.17}$$

Řešení se pak snažíme určit jako limitu posloupnosti $\{\mathbf{x}^{(k)}\}$, kterou počítáme podle rekurentního vzorce

$$\mathbf{x}^{(k+1)} = \mathbf{C}\mathbf{x}^{(k)} + \mathbf{d}. \tag{4.18}$$

Jestliže odečteme (4.17) a (4.18) a označíme přitom $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}$, dostaneme

$$\mathbf{e}^{(k+1)} = \mathbf{C}\mathbf{e}^{(k)}.$$

Tento vzorec ukazuje jak se chová *iterační chyba* $\mathbf{e}^{(k)}$. Opakovaným použitím vzorce dostaneme $\mathbf{e}^{(k+1)} = \mathbf{C}\mathbf{e}^{(k)} = \mathbf{C}^2\mathbf{e}^{(k-1)} = \cdots = \mathbf{C}^{k+1}\mathbf{e}^{(0)}$. Proto

$$\mathbf{e}^{(k)} = \mathbf{C}^k\mathbf{e}^{(0)}, \tag{4.19}$$

kde $\mathbf{e}^{(0)}$ je *počáteční chyba*, která je dána volbou počáteční aproximace $\mathbf{x}^{(0)}$.

Iterační výpočet bude konvergovat, jestliže $\lim_{k\to\infty} \mathbf{e}^{(k)} = \mathbf{0}$, tj. když se iterační chyba bude blížit k nulovému vektoru. Tuto limitu budeme vyšetřovat pomocí vzorce (4.19). Budeme přitom předpokládat, že iterační matice $\mathbf{C}$ má vlastní čísla $\lambda_1, \ldots, \lambda_n$, kterým odpovídají vlastní vektory $\mathbf{v}_1, \ldots, \mathbf{v}_n$ a ty tvoří bázi. Připomeňme, že taková situace nastane podle Věty 4.3.2 například tehdy, je-li $\mathbf{C}$ symetrická matice. Vektor $\mathbf{e}^{(0)}$ pak můžeme zapsat jako lineární kombinaci vlastních vektorů, tj. existují konstanty $c_1, \ldots, c_n$, pro něž platí

$$\mathbf{e}^{(0)} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n. \tag{4.20}$$

Jestliže dosadíme (4.20) do (4.19) dostaneme s pomocí Věty 4.3.1 vztah

$$\begin{aligned} \mathbf{e}^{(k)} &= c_1\mathbf{C}^k\mathbf{v}_1 + \cdots + c_n\mathbf{C}^k\mathbf{v}_n \\ &= c_1\lambda_1^k\mathbf{v}_1 + \cdots + c_n\lambda_n^k\mathbf{v}_n. \end{aligned} \tag{4.21}$$

Odtud je vidět, že pro každou volbu počáteční aproximace $\mathbf{x}^{(0)}$ bude

$$\lim_{k\to\infty} \mathbf{e}^{(k)} = \mathbf{0} \iff \lim_{k\to\infty} \lambda_i^k = 0 \quad \text{pro} \quad i = 1, \ldots, n.$$

Uvedené limity budou nulové, právě když $|\lambda_i| < 1$ pro $i = 1, \ldots, n$. Dokázali jsme následující tvrzení.

Věta 4.4.1 Nechť $\mathbf{C}$ je iterační matice, která má n lineárně nezávislých vlastních vektorů. Iterační metoda daná vzorcem (4.18) konverguje pro každou počáteční aproximaci $\mathbf{x}^{(0)}$, právě když všechna vlastní čísla matice $\mathbf{C}$ jsou v absolutní hodnotě menší než jedna.

Příklad 4.4.1 Určete vlastní čísla iteračních matic $\mathbf{C}_J$ a $\mathbf{C}_B$ z Odstavce 4.1 a porovnejte průběhy iteračních výpočtů s tvrzením poslední věty.

Řešení: Z charakteristického polynomu $p_{\mathbf{C}_J}(\lambda) = \lambda^3 + \frac{7}{88}\lambda - \frac{1}{80}$ určíme vlastní čísla matice $\mathbf{C}_J$: $\lambda_1 \doteq 0.1297$, $\lambda_2 \doteq -0.0649 + i0.3036$, $\lambda_3 \doteq -0.0649 - i0.3036$. Z absolutních hodnot $|\lambda_1| = \lambda_1$, $|\lambda_2| = |\lambda_3| \doteq 0.3104$ vidíme, že iterační výpočet musí být konvergentní, což je v souladu s naším pozorováním z Odstavce 4.1. Podobně z charakteristického polynomu $p_{\mathbf{C}_B}(\lambda) = \lambda^3 + 26\lambda^2 + 229\lambda + 683$ určíme vlastní čísla matice $\mathbf{C}_B$. Stačí si povšimnout, že jedno z vlastních čísel je $\lambda_1 \doteq -8.7373$. Protože $|\lambda_1| > 1$, nemůže iterační výpočet (obecně) konvergovat, což je rovněž v souladu s pozorováním z Odstavce 4.1. □

Viděli jsme, že o konvergenci iteračních metod lze rozhodnout pomocí vlastních čísel. Výpočet vlastních čísel je ale obvykle mnohem náročnější úloha než řešení soustavy lineárních rovnic. V další větě proto ukážeme jednodušší, i když slabší konvergenční podmínku.

Věta 4.4.2 Nechť $\mathbf{C}$ je iterační matice, která má n lineárně nezávislých vlastních vektorů. Iterační metoda daná vzorcem (4.18) konverguje pro každou počáteční aproximaci $\mathbf{x}^{(0)}$, jestliže pro některou normu platí $\|\mathbf{C}\| < 1$.

Důkaz: Nechť $\|\mathbf{C}\| < 1$ a nechť λ je libovolné vlastní číslo matice $\mathbf{C}$ odpovídající vlastnímu vektoru $\mathbf{v}$, tj. $\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$. Protože $|\lambda|\|\mathbf{v}\| = \|\lambda\mathbf{v}\| = \|\mathbf{C}\mathbf{v}\| \leq \|\mathbf{C}\|\|\mathbf{v}\|$, platí $|\lambda| \leq \|\mathbf{C}\| < 1$, takže všechna vlastní čísla matice $\mathbf{C}$ jsou v absolutní hodnotě menší než jedna. Iterační metoda proto konverguje podle Věty 4.4.1 □

U Jakobiovy a Gauss-Seidelovy iterační metody lze konvergenční podmínku z poslední věty formulovat pomocí matice soustavy $\mathbf{A}$. Používá se přitom terminologie z následující definice.

Definice 4.4.1 Řekneme, že čtvercová matice $\mathbf{A} = (a_{ij})$ řádu n je *ostře diagonálně dominantní*, jestliže platí

$$|a_{i1}| + \cdots + |a_{ii-1}| + |a_{ii+1}| + \cdots + |a_{in}| < |a_{ii}| \quad \text{pro} \quad i = 1, \ldots, n. \tag{4.22}$$

Příklad 4.4.2 Rozhodněte, která z následujících matic je ostře diagonálně dominantní:

$$\mathbf{A} = \begin{pmatrix} 11 & 2 & 1 \\ 1 & 10 & 2 \\ 2 & 3 & -8 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -8 & 2 & 1 \\ -2 & 1 & -3 \\ 1 & 1 & 3 \end{pmatrix}.$$

Řešení: Matice $\mathbf{A}$ je ostře diagonálně dominantní, protože platí $2 + 1 < 11$, $1 + 2 < 10$ a $2 + 3 < 8$. Matice $\mathbf{B}$ není ostře diagonálně dominantní, protože ve druhém řádku je $2 + 3 > 1$. □

Věta 4.4.3 Nechť $\mathbf{A}\mathbf{x} = \mathbf{b}$ je daná soustava lineárních rovnic. Jakobiova iterační metoda konverguje pro každou počáteční aproximaci $\mathbf{x}^{(0)}$, jestliže matice $\mathbf{A}$ je ostře diagonálně dominantní.

Důkaz: Nechť $\mathbf{A}$ je ostře diagonálně dominantní. Podmínku (4.22) může přepsat do tvaru

$$\frac{|a_{i1}|}{|a_{ii}|} + \cdots + \frac{|a_{ii-1}|}{|a_{ii}|} + \frac{|a_{ii+1}|}{|a_{ii}|} + \cdots + \frac{|a_{in}|}{|a_{ii}|} < 1 \quad \text{pro} \quad i = 1, \ldots, n.$$

Pro iterační matici $\mathbf{C}_J$ (viz (4.11)) to znamená, že součet absolutních hodnot prvků v každém řádku je menší než jedna. V řádkové normě proto platí $\|\mathbf{C}_J\|_R < 1$, takže konvergence Jakobiovy iterační metody tak plyne z Věty 4.4.2 □

Poznámka

Také Gauss-Seidelova iterační metoda konverguje, je-li matice soustavy ostře diagonálně dominantní. Důkaz je však o něco složitější; viz [1].

Poznámka

Konvergenci Jakobiovy i Gauss-Seidelovy iterační metody zajistíme tak, že řešenou soustavu předem upravíme na ekvivalentní soustavu s ostře diagonálně dominantní maticí.

Příklad 4.4.3 Soustavu lineárních rovnic

$$\begin{aligned} -8x_1 + 2x_2 + x_3 &= -1, \\ -2x_1 + x_2 - 3x_3 &= -9, \\ x_1 + x_2 + 3x_3 &= 12 \end{aligned}$$

upravte na tvar s ostře diagonálně dominantní maticí.

Řešení: Úpravy, které nemění řešení, jsou tři: záměna pořadí rovnic, vynásobení rovnice nenulovým číslem a přičtení nenulového násobku rovnice k jiné rovnici. V našem případě stačí přičíst třetí rovnici k rovnici druhé:

$$\begin{aligned} -8x_1 + 2x_2 + x_3 &= -1, \\ -x_1 + 2x_2 &= -3, \\ x_1 + x_2 + 3x_3 &= 12. \end{aligned}$$

Provedeme-li výpočet podle Jakobiovy nebo Gauss-Seidelovy iterační metody pro tuto soustavu, budeme mít zaručenu konvergenci. □

Kontrolní otázky

Otázka 1. Jaké podmínky zaručují konvergenci obecné iterační metody?
Otázka 2. Jak lze zajistit konvergenci u Jakobiovy a Gauss-Seidelovy iterační metody?

Úlohy k samostatnému řešení

1. Soustavu lineárních rovnic

$$\begin{aligned} 4x_1 + x_2 + x_3 &= 6, \\ x_1 + 4x_2 + x_3 &= 6, \\ 6x_1 + 6x_2 + 6x_3 &= 18 \end{aligned}$$

upravte na tvar s ostře diagonálně dominantní maticí.
2. Napište iterační matici pro Jakobiovu iterační metodu a vypočtěte její vlastní čísla.

Výsledky úloh k samostatnému řešení

1. Od třetí rovnice odečteme rovnici první i druhou. Dostaneme:

$$\begin{aligned} 4x_1 + x_2 + x_3 &= 6, \\ x_1 + 4x_2 + x_3 &= 6, \\ x_1 + x_2 + 4x_3 &= 6. \end{aligned}$$

2. Iterační matice má tvar:

$$\mathbf{C}_J = \begin{pmatrix} 0 & -1/4 & -1/4 \\ -1/4 & 0 & -1/4 \\ -1/4 & -1/4 & 0 \end{pmatrix}.$$

Z charakteristického polynomu $p_{\mathbf{C}_J}(\lambda) = \lambda^3 - \frac{3}{16}\lambda + \frac{1}{32}$ určíme kořeny $\lambda_1 = -\frac{1}{2}$, $\lambda_2 = \lambda_3 = \frac{1}{4}$.

KAPITOLA

5

INTERPOLACE A APROXIMACE FUNKCÍ

Často vzniká potřeba k dané funkci $f : \langle a, b\rangle \mapsto \mathbb{R}$ najít jiný funkční předpis $\varphi : \langle a, b\rangle \mapsto \mathbb{R}$ tak, aby se funkce f a φ od sebe příliš nelišily. Například vzorec popisující funkci f může být „složitý“ a vzorec pro φ může být jeho „jednodušší náhrada“. Často se také objevuje situace, kdy u funkce f známe jen její funkční hodnoty v určitých bodech (třeba jako výsledek měření). Sestavením funkce φ pak původní funkci f dodefinujeme na zbývajících částech intervalu $\langle a, b\rangle$.

V této kapitole budeme předpokládat, že jsou zadány uzly $x_i \in \langle a, b\rangle$ a v nich jsou předepsány funkční hodnoty $f_i = f(x_i)$ pro $i = 0, \ldots, n$. Budeme rozlišovat dvě úlohy.

Interpolační úloha: Hledáme funkci φ, pro niž platí

$$\varphi(x_i) = f_i, \quad i = 0, \ldots, n. \tag{5.1}$$

Aproximace metodou nejmenších čtverců: Hledáme funkci φ, pro niž je

$$\varphi(x_i) \approx f_i, \quad i = 0, \ldots, n, \tag{5.2}$$

kde přibližná rovnost „$\approx$“ je určena tak, aby součet druhých mocnin odchylek mezi předepsanými hodnotami f_i a předpokládanými hodnotami $\varphi(x_i)$ byl minimální (zápis vzorcem uvedeme později).

Jestliže tyto úlohy znázorníme graficky, bude graf funkce φ při řešení interpolační úlohy procházet skrze body (x_i, f_i), $i = 0, \ldots, n$, zatímco při řešení aproximační úlohy bude (obecně) procházet jejich blízkým okolím. Formulace obou úloh je ale zatím příliš obecná, protože jsme neřekli jakého typu má být funkce φ. Ukážeme tři volby: polynom, splajn (spline-funkce) a lineární kombinace obecných funkcí. Polynom je jednoduchý z hlediska provádění matematických operací (snadno se derivuje, integruje atp.), jeho graf však často osciluje. Lepší tvary grafu dostaneme pro splajny. Kombinace obecných funkcí se používá zpravidla v situacích, kdy je známo, jakou závislost daná data popisují (pro periodickou závislost je dobré použít funkce goniometrické, pro strmě rostoucí data se hodí funkce exponenciální atp.).

5.1 Interpolační polynom

Funkci φ v úloze (5.1) budeme hledat jako *interpolační polynom* stupně nejvýše n, tj. položíme $\varphi = p_n$, kde

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n. \tag{5.3}$$

Začneme příkladem.

Příklad 5.1.1 Jsou dány uzly $x_0 = -2$, $x_1 = -1$, $x_2 = 1$, $x_3 = 2$ a funkční hodnoty $f_0 = 10$, $f_1 = 4$, $f_2 = 6$, $f_3 = 3$. Určete interpolační polynom p_3.

Řešení: Hledaný polynom má obecný tvar

$$p_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3.$$

Koeficienty a_0, a_1, a_2, a_3 určíme tak, aby platilo (5.1). Každá interpolační rovnost určuje jednu rovnici:

$$\begin{array}{rcccccccccc}
p_3(-2) & = & 10 & \Rightarrow & a_0 & - & 2a_1 & + & 4a_2 & - & 8a_3 & = & 10, \\
p_3(-1) & = & 4 & \Rightarrow & a_0 & - & a_1 & + & a_2 & - & a_3 & = & 4, \\
p_3(1) & = & 6 & \Rightarrow & a_0 & + & a_1 & + & a_2 & + & a_3 & = & 6, \\
p_3(2) & = & 3 & \Rightarrow & a_0 & + & 2a_1 & + & 4a_2 & + & 8a_3 & = & 3.
\end{array}$$

Dostali jsme soustavu lineárních rovnic

$$\begin{pmatrix} 1 & -2 & 4 & -8 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 4 \\ 6 \\ 3 \end{pmatrix},$$

jejímž řešením (na tři desetinná místa) jsou koeficienty $a_0 = 4.500$, $a_1 = 1.917$, $a_2 = 0.500$ a $a_3 = -0.917$. Interpolační polynom má tvar

$$p_3(x) = 4.500 + 1.917x + 0.500x^2 - 0.917x^3.$$

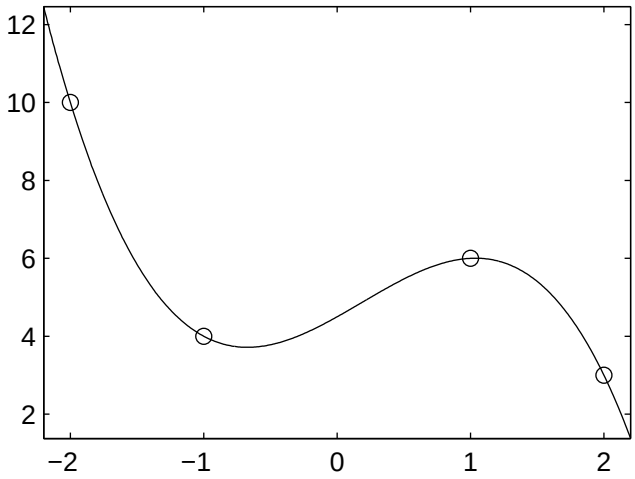
Jeho graf je na Obrázku 5.1. □

Rozborem postupu z příkladu dokážeme následující větu.

Věta 5.1.1 Nechť jsou dány vzájemně různé uzly x_i a funkční hodnoty f_i, $i = 0, \ldots, n$. Existuje právě jeden interpolační polynom stupně nejvýše n.

Důkaz: Dosazením obecného tvaru polynomu (5.3) do interpolačních rovností (5.1) dostaneme soustavu lineárních rovnic

$$a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_n x_i^n = f_i, \quad i = 0, \ldots, n,$$



Obrázek 5.1: Graf interpolačního polynomu p_3.

kterou lze zapsat pomocí matice jako

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}.$$

Matice této soustavy má nenulový (Vandermodův) determinant. Odtud plyne existence jediného řešení soustavy lineárních rovnic a také existence jediného interpolačního polynomu. □

Tento způsob vytvoření interpolačního polynomu umožňuje snadno analyzovat řešitelnost úlohy. Nevýhodou je potřeba řešit soustavu lineárních rovnic s (Vandermondovou) maticí, která je při větším počtu interpolovaných hodnot velmi špatně podmíněna.

5.1.1 Lagrangeův tvar interpolačního polynomu

Ukážeme postup, při němž se obejdeme bez řešení soustavy lineárních rovnic. Interpolační polynom budeme hledat ve tvaru

$$p_n(x) = f_0\varphi_0(x) + f_1\varphi_1(x) + \cdots + f_n\varphi_n(x). \tag{5.4}$$

Rovnosti $p_n(x_i) = f_i$, $i = 0, 1, \dots, n$ budou splněny, jestliže bude platit

$$\varphi_i(x_j) = \begin{cases} 1 & \text{pro } i = j, \\ 0 & \text{pro } i \neq j. \end{cases}$$

Z Věty 5.1.1 víme, že interpolační polonom je stupně nejvýše n, takže také všechny funkce φ_i musí být polynomy stupně nejvýše n. Uvedeným požadavkům vyhovuje následující definice:

$$\varphi_i(x) = \frac{(x - x_0)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_0)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)} \tag{5.5}$$

pro $i = 0, 1, \ldots, n$. Čitatel je totiž polynom, který nabývá nulových hodnot ve všech uzlech kromě uzlu x_i. V tomto uzlu pak nabývá nenulové hodnoty, která je obsažena ve jmenovateli zlomku, takže platí $\varphi_i(x_i) = 1$.

Polynomům φ_i, $i = 0, 1, \ldots, n$ se říká *Lagrangeova báze* interpolační úlohy a vzorec (5.4) se nazývá *Lagrangeův tvar* inteprolačního polynomu.

Příklad 5.1.2 Mějme dány uzly $x_0 = -2$, $x_1 = -1$, $x_2 = 1$, $x_3 = 2$ a funkční hodnoty $f_0 = 10$, $f_1 = 4$, $f_2 = 6$, $f_3 = 3$. Napište Lagrangeův tvar interpolačního polynomu.

Řešení: Nejdříve sestavíme Lagrangeovu bázi. Podle (5.5) je

$$\begin{aligned}
\varphi_0(x) &= \frac{(x+1)(x-1)(x-2)}{(-2+1)(-2-1)(-2-2)} = -\frac{1}{12}(x+1)(x-1)(x-2),\\
\varphi_1(x) &= \frac{(x+2)(x-1)(x-2)}{(-1+2)(-1-1)(-1-2)} = \frac{1}{6}(x+2)(x-1)(x-2),\\
\varphi_2(x) &= \frac{(x+2)(x+1)(x-2)}{(1+2)(1+1)(1-2)} = -\frac{1}{6}(x+2)(x+1)(x-2),\\
\varphi_3(x) &= \frac{(x+2)(x+1)(x-1)}{(2+2)(2+1)(2-1)} = \frac{1}{12}(x+2)(x+1)(x-1).
\end{aligned}$$

Dosazením do (5.4) dostaneme výsledek

$$\begin{aligned}
p_3(x) = &-\frac{5}{6}(x+1)(x-1)(x-2) + \frac{2}{3}(x+2)(x-1)(x-2) -\\
&-(x+2)(x+1)(x-2) + \frac{1}{4}(x+2)(x+1)(x-1).
\end{aligned}$$

□

Poznámka

Interpolační polynom je podle Věty 5.1.1 určen jednoznačně. Úpravou Lagrangeova tvaru interpolačního polynomu proto musíme nutně odvodit vzorec, který jsem vypočítali v Příkladu 5.1.1 (ověřte).

5.1.2 Newtonův tvar interpolačního polynomu

Newtonův tvar interpolačního polynomu je jistým kompromisem mezi předchozími dvěma postupy. Vyžaduje sice řešení soustavy lineárních rovnic, ta má ale trojúhelníkovou strukturu. Její řešení se převede na výpočet poměrných diferencí. Zápis interpolačního polynomu pomocí poměrných diferencí je pak analogie Taylorova polynomu, kdy informace o vytvářené funkci je rozprostřena do funkčních hodnot v různých uzlech (Taylorův polynom používá informaci z jednoho uzlu, kde jsou předepsány derivace různých řádů).

Uvažujme zápis polynomu ve tvaru:

$$p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\ldots(x - x_{n-1}). \qquad (5.6)$$

Jestliže dosadíme do interpolačních rovností $p_n(x_i) = f_i$, $i = 0, 1, \ldots, n$, dostaneme soustavu lineárních rovnic s dolní trojúhelníkovou maticí:

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & (x_1 - x_0) & 0 & \ldots & 0 \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \end{pmatrix}, \tag{5.7}$$

Odud můžeme postupně vyjádřit koeficienty a_k:

$$\begin{aligned} a_0 &= f_0, \quad a_1 = \frac{f_1 - a_0}{x_1 - x_0} = \frac{f_1 - f_0}{x_1 - x_0}, \\ a_2 &= \frac{f_2 - a_1(x_2 - x_0) - a_0}{(x_2 - x_0)(x_2 - x_1)} = \frac{\dfrac{f_2 - f_1}{x_2 - x_1} - \dfrac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0}, \end{aligned}$$

atd..

Výrazy na pravých stranách jsou poměrné diference, pro něž zavádíme značení v následující definici.

Definice 5.1.1 Nechť jsou dány vzájemně různé uzly x_i a funkční hodnoty f_i, $i = 0, \ldots, n$. Poměrné diference k-tého řádu $f[x_{i+k}, \ldots, x_i]$, $i = 0, 1, \ldots, n-k$ definujeme rekurentně:

- pro $k = 0$: $f[x_i] = f_i$;
- pro $k = 1$: $f[x_{i+1}, x_i] = \dfrac{f_{i+1} - f_i}{x_{i+1} - x_i}$;
- pro $k \leq n$: $f[x_{i+k}, \ldots, , x_i] = \dfrac{f[x_{i+k}, \ldots, x_{i+1}] - f[x_{i+k-1}, \ldots, x_i]}{x_{i+k} - x_i}$.

Porovnáním poměrných diferencí s koeficienty a_k vidíme, že

$$a_k = f[x_k, \ldots, x_0], \quad k = 0, 1, \ldots, n.$$

Dosazením do (5.6) dostaneme *Newtonův tvar* interpolačního polynomu:

$$p_n(x) = f_0 + f[x_1, x_0](x - x_0) + \cdots + f[x_n, \ldots, x_0](x - x_0) \ldots (x - x_{n-1}). \tag{5.8}$$

Při jeho sestavování potřebujeme vypočítat poměrné diference. Vše ukážeme v následujícím příkladu.

Příklad 5.1.3 Mějme dány uzly $x_0 = -2$, $x_1 = -1$, $x_2 = 1$, $x_3 = 2$ a funkční hodnoty $f_0 = 10$, $f_1 = 4$, $f_2 = 6$, $f_3 = 3$. Napište Newtonův tvar interpolačního polynomu.

Řešení: Potřebujeme vypočítat poměrné diference:

$$f[x_1, x_0], \; f[x_2, x_1, x_0], \; f[x_3, x_2, x_1, x_0].$$

Podle definice je

$$f[x_1,x_0] = \frac{4-10}{-1+2} = -6,$$

$$f[x_2,x_1,x_0] = \frac{f[x_2,x_1]-f[x_1,x_0]}{x_2-x_0} = \frac{\frac{6-4}{1+1}+6}{1+2} = \frac{7}{3},$$

$$f[x_3,x_2,x_1,x_0] = \frac{f[x_3,x_2,x_1]-f[x_2,x_1,x_0]}{x_3-x_0} = \frac{\frac{\frac{3-6}{2-1}-\frac{6-4}{1+1}}{2+1}-\frac{7}{3}}{2+2} = -\frac{11}{12}.$$

Dosazením do (5.8) dostaneme výsledek

$$p_3(x) = 10-6(x+2)+\frac{7}{3}(x+2)(x+1)-\frac{11}{12}(x+2)(x+1)(x-1).$$

Přehledně můžeme výpočet poměrných diferencí provést v tabulce (Tabulka 5.1), kde do prvních dvou sloupců zapíšeme zadané uzly a funkční hodnoty a v každém dalším sloupci pak vypočítáme všechny poměrné diference postupně se zvyšujících řádů. Pro napsání interpolačního polynomu potřebujeme z této tabulky hodnoty diferencí z prvního řádku. □

i	x_i	f_i	$f[x_{i+1},x_i]$	$f[x_{i+2},x_{i+1},x_i]$	$f[x_3,x_2,x_1,x_0]$
0	−2	10	−6	7/3	−11/12
1	−1	4	1	−4/3	
2	1	6	−3		
3	2	3			

Tabulka 5.1: Výpočet poměrných diferencí.

5.1.3 Interpolační chyba

Předpokládejme, že hodnoty f_i jsou funkčními hodnotami funkce f v uzlech x_i, tj. $f_i = f(x_i)$. Bude nás zajímat *interpolační chyba*

$$f(x)-p_n(x).$$

V uzlech x_i je interpolační chyba nulová, ale mimo uzly může být velká.

Věta 5.1.2 Nechť uzly x_i, $i = 0,1,\ldots,n$, jsou vzájemně různé a leží na intervalu $\langle a,b\rangle$. Nechť funkce f má na tomto intervalu $n+1$ spojitých derivací. Pak pro každé $x \in \langle a,b\rangle$ existuje $\xi = \xi(x)$ v (a,b) tak, že platí

$$f(x)-p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x), \tag{5.9}$$

kde $\pi_{n+1}(x) = (x-x_0)\ldots(x-x_n)$.

Důkaz: Pro $x = x_i$ je rovnost (5.9) splněna, protože obě její strany jsou nulové. Pro pevně zvolené $x \neq x_i$ definujme funkci

$$g(t) = f(t)-p_n(t)-\frac{\pi_{n+1}(t)}{\pi_{n+1}(x)}\left(f(x)-p_n(x)\right), \tag{5.10}$$

kde t je proměnná a x je parametr. Funkce g má zřejmě $n+2$ kořenů, kterými jsou všechny uzly $x_0, \ldots, x_n$ a x. Každá derivace funkce g má o jeden kořen méně, takže $(n+1)$-ní derivace má jediný kořen v nějakém bodě $\xi \in (a,b)$. Derivujeme-li $(n+1)$-krát výraz (5.10) (podle t) a použijeme přitom $p_n^{(n+1)}(t) = 0$ a $\pi_{n+1}^{(n+1)}(t) = (n+1)!$, dostaneme

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \frac{(n+1)!}{\pi_{n+1}(x)}\left(f(x) - p_n(x)\right).$$

Jestliže odtud vyjádříme interpolační chybu, odvodíme rovnost (5.9). □

Na průběh interpolační chyby v intervalu $\langle a,b\rangle$ má podstatný vliv tvar polynomu π_{n+1}, jak ukazuje následující příklad.

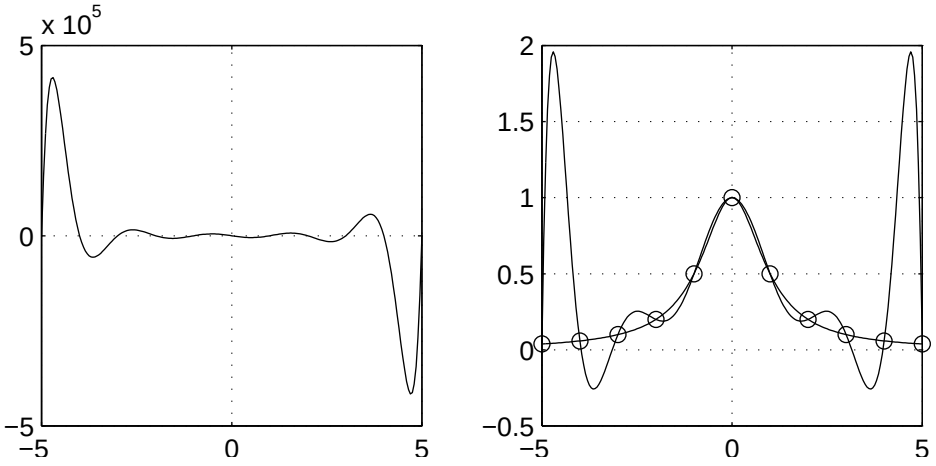
Příklad 5.1.4 (*Rungeho příklad*) Nakreslíme graf funkce

$$f(x) = \frac{1}{1+x^2}$$

a graf interpolačního polynomu odpovídajícího uzlům $x_i = -5+i$, $i = 0,1,\ldots,10$. Výsledek porovnáme s grafem polynomu

$$\pi_{11}(x) = (x+5)(x+4)\ldots(x-5).$$

Řešení: Obrázek 5.2.a ukazuje graf polynomu π_{11}. Z jeho průběhu lze usoudit, že největší interpolační chyby budou poblíž krajních uzlů $x_0 = -5$ a $x_{10} = 5$. Na obrázku 5.2.b vidíme, že graf interpolačního polynomu osciluje kolem grafu funkce f a že oscilace jsou největší právě na krajích intervalu $\langle -5,5\rangle$. Poznamenejme ještě, že při zvětšení počtu interpolačních uzlů nemusí dojít ke zmenšení interpolační chyby, ale naopak k jejímu zvětšení. □



(a) (b)

Obrázek 5.2: (a) Graf π_{11}; (b) grafy f (neoscilující) a p_{10} (oscilující).

Kontrolní otázky

Otázka 1. Jaké znáte metody pro sestavení interpolačního polynomu?

Otázka 2. Jakého stupně je interpolační polynom?
Otázka 3. Jak se chová interpolační chyba?

Úlohy k samostatnému řešení

1. Pro uzly $x_0 = -1$, $x_1 = 0$, $x_2 = 2$, $x_3 = 3$, $x_4 = 5$ a funkční hodnoty $f_0 = -2$, $f_1 = 1$, $f_2 = 0$, $f_3 = 2$, $f_4 = -1$ vypočtěte interpolační polynom ve tvaru (5.3).
2. Pro předchozí data vypočtěte Lagrangeův a Newtonův tvar interpolačního polynomu.

Výsledky úloh k samostatnému řešení

1. $p_4(x) = -\frac{3}{20}x^4 + \frac{11}{10}x^3 - \frac{109}{60}x^2 - \frac{1}{15}x + 1$.
2. Lagrangeův tvar: $p_4(x) = -\frac{1}{36}x(x-2)(x-3)(x-5) - \frac{1}{30}(x+1)(x-2)(x-3)(x-5) - \frac{1}{12}(x+1)x(x-2)(x-5) - \frac{1}{180}(x+1)x(x-2)(x-3)$;
Newtonův tvar: $p_4(x) = -2 + 3(x+1) - \frac{35}{30}(x+1)x + \frac{1}{2}(x+1)x(x-2) - \frac{3}{20}(x+1)x(x-2)(x-3)$.

5.2 Interpolační splajny

Viděli jsme, že graf interpolačního polynomu může oscilovat, což je pro mnohé praktické aplikace nepřijatelné. Tato situace nastává zpravidla při předepsání většího počtu dat, protože interpolační polynom je pak vysokého stupně. Zdá se proto rozumné řešit interpolační úlohu použitím funkcí, která jsou *počástech* polynomy nízkého stupně, a jejichž jednotlivé polynomické části na sebe navazují dostatečně hladce. Takovým funkcím se říká *splajn* (z angl. „spline“). Ukážeme dva nejčastěji používané splajny: lineární a kubický.

Abychom se vyhnuli zbytečným komplikacím v zápisu, budeme předpokládat, že uzly interpolace tvoří rostoucí posloupnost, tzn. že platí $x_0 < x_1 < \cdots < x_n$. Vzdálenost dvou sousedních uzlů označíme h_i, tj. $h_i = x_i - x_{i-1}, \quad i = 1, \ldots, n$.

5.2.1 Lineární splajn

Definice 5.2.1 Lineárním splajnem nazýváme funkci s_1, která je spojitá na intervalu $\langle x_0, x_n \rangle$ a na každém podintervalu $\langle x_{i-1}, x_i \rangle$, $i = 1, \ldots, n$, je polynomem prvního stupně.

Lineární *interpolační* splajn bude řešením interpolační úlohy (5.1), položíme-li $s_1(x_i) = f_i$, $i = 0, \ldots, n$. Zápis tohoto splajnu provedeme „po částech“, kdy na každém intervalu $\langle x_{i-1}, x_i \rangle$ zavedeme lokální proměnou t:

$$s_1(x) = f_{i-1}(1-t) + f_i t, \quad t = (x - x_{i-1})/h_i, \quad x \in \langle x_{i-1}, x_i \rangle \tag{5.11}$$

pro $i = 1, \ldots, n$. Výraz s lokální proměnnou t představuje Lagrangeův tvar lineárního interpolačního polynomu. Spojitost je zaručena tím, že pro interval $\langle x_{i-1}, x_i \rangle$ a $\langle x_i, x_{i+1} \rangle$ je v bodě x_i předepsaná stejná funkční hodnota f_i. Grafem lineárního splajnu je *lomená čára*.

Příklad 5.2.1 Mějme dány uzly $x_0 = -2$, $x_1 = -1$, $x_2 = 1$, $x_3 = 2$ a funkční hodnoty $f_0 = 10$, $f_1 = 4$, $f_2 = 6$, $f_3 = 3$. Napište lineární interpolační splajn.

Řešení: Zápis provedeme podle předpisu (5.11):

$$s_1(x) = \begin{cases} 10\varphi_0(t) + 4\varphi_1(t), & t = x + 2 \quad \text{pro } x \in \langle -2, -1\rangle, \\ 4\varphi_0(t) + 6\varphi_1(t), & t = (x+1)/2 \quad \text{pro } x \in \langle -1, 1\rangle, \\ 6\varphi_0(t) + 3\varphi_1(t), & t = x - 1 \quad \text{pro } x \in \langle 1, 2\rangle, \end{cases}$$

kde $\varphi_0(t) = 1 - t$, $\varphi_1(t) = t$. Graf je znázorněn na Obrázku 5.3. □

5.2.2 Kubický splajn

Definice 5.2.2 Kubickým splajnem nazýváme funkci s_3, která má na intervalu $\langle x_0, x_n\rangle$ dvě spojité derivace a na každém podintervalu $\langle x_{i-1}, x_i\rangle$, $i = 1, \ldots, n$ je polynomem třetího stupně.

Kubický *interpolační* splajn, který řeší interpolační úlohu (5.1), zapíšeme opět „po částech“:

$$\begin{aligned} s_3(x) \;=\; & f_{i-1}(1 - 3t^2 + 2t^3) + f_i(3t^2 - 2t^3) \\ & + m_{i-1}h_i(t - 2t^2 + t^3) + m_i h_i(-t^2 + t^3), \end{aligned} \tag{5.12}$$

kde $t = (x - x_{i-1})/h_i$, $x \in \langle x_{i-1}, x_i\rangle$ pro $i = 1, \ldots, n$. Tento předpis je navržen tak, aby platilo

$$s_3(x_{i-1}) = f_{i-1}, \quad s_3(x_i) = f_i, \tag{5.13}$$

$$s_3'(x_{i-1}) = m_{i-1}, \quad s_3'(x_i) = m_i. \tag{5.14}$$

O splnění vztahů (5.13) se můžeme snadno přesvědčit dosazením x_{i-1} a x_i. Tyto vztahy také zaručují spojitost. První derivaci s_3' vyjádříme z (5.12) pomocí pravidla o derivování složené funkce:

$$\begin{aligned} s_3'(x) \;=\; & f_{i-1}(-6t + 6t^2)/h_i + f_i(6t - 6t^2)/h_i \\ & + m_{i-1}(1 - 4t + 3t^2) + m_i(-2t + 3t^2). \end{aligned} \tag{5.15}$$

Dosazením x_{i-1} a x_i se můžeme přesvědčit o splnění vztahů (5.14). Tím je zaručena také spojitost první derivace s_3' na celém intervalu $\langle x_0, x_n\rangle$ pro libovolné hodnoty m_i, které mají význam prvních derivací v uzlech x_i. Spojitost druhé derivace vynutíme speciální volbou hodnot m_i. Budeme požadovat

$$\lim_{x \to x_i -} s_3''(x) = \lim_{x \to x_i +} s_3''(x) \tag{5.16}$$

ve vnitřních uzlech x_i, $i = 1, \ldots, n-1$. Potřebný výraz pro druhou derivaci vypočteme z (5.15) opět podle pravidla o derivování složené funkce:

$$\begin{aligned} s_3''(x) \;=\; & f_{i-1}(-6 + 12t)/h_i^2 + f_i(6 - 12t)/h_i^2 \\ & + m_{i-1}(-4 + 6t)/h_i + m_i(-2 + 6t)/h_i. \end{aligned} \tag{5.17}$$

Levou stranu v (5.16) vyjádříme z (5.17) pro $t = 1$:

$$\lim_{x \to x_i -} s_3''(x) = 6f_{i-1}/h_i^2 - 6f_i/h_i^2 + 2m_{i-1}/h_i + 4m_i/h_i. \tag{5.18}$$

Pravou stranu v (5.16) vyjádříme z (5.17) pro $t = 0$, když současně posuneme indexování:

$$\lim_{x \to x_i +} s''(x) = -6f_i/h_{i+1}^2 + 6f_{i+1}/h_{i+1}^2 - 4m_i/h_{i+1} - 2m_{i+1}/h_{i+1}. \tag{5.19}$$

Dosadíme-li (5.18) a (5.19) do (5.16), dostaneme po jednoduché úpravě

$$h_{i+1}m_{i-1} + 2(h_{i+1} + h_i)m_i + h_i m_{i+1} =$$
$$3\left[-\frac{h_{i+1}}{h_i}f_{i-1} + \left(\frac{h_{i+1}}{h_i} - \frac{h_i}{h_{i+1}}\right)f_i + \frac{h_i}{h_{i+1}}f_{i+1}\right], \quad i = 1, \ldots, n-1. \tag{5.20}$$

Tyto rovnosti tvoří soustavu $n-1$ rovnic pro $n+1$ neznámých m_i, $i = 0, 1, \ldots .n$. Abychom dostali jediné řešení, předepíšeme m_0 a m_n například takto (jako přibližné derivace):

$$m_0 = \frac{f_1 - f_0}{h_1}, \quad m_n = \frac{f_n - f_{n-1}}{h_n}. \tag{5.21}$$

Příklad 5.2.2 Mějme dány uzly $x_0 = -2$, $x_1 = -1$, $x_2 = 1$, $x_3 = 2$ a funkční hodnoty $f_0 = 10$, $f_1 = 4$, $f_2 = 6$, $f_3 = 3$. Napište kubický interpolační splajn.

Řešení: Nejdříve vypočítáme parametry m_i, $i = 0, 1, 2, 3$. Podle (5.21) je

$$m_0 = \frac{4 - 10}{-1 + 2} = -6, \quad m_3 = \frac{3 - 6}{2 - 1} = -3.$$

Soustava (5.20) má dvě rovnice:

$$\begin{aligned} 2(h_2 + h_1)m_1 + h_1 m_2 &= 3\left[-\frac{h_2}{h_1}f_0 + \left(\frac{h_2}{h_1} - \frac{h_1}{h_2}\right)f_1 + \frac{h_1}{h_2}f_2\right] - h_2 m_0, \\ h_3 m_1 + 2(h_3 + h_2)m_2 &= 3\left[-\frac{h_3}{h_2}f_1 + \left(\frac{h_3}{h_2} - \frac{h_2}{h_3}\right)f_2 + \frac{h_2}{h_3}f_3\right] - h_2 m_3, \end{aligned}$$

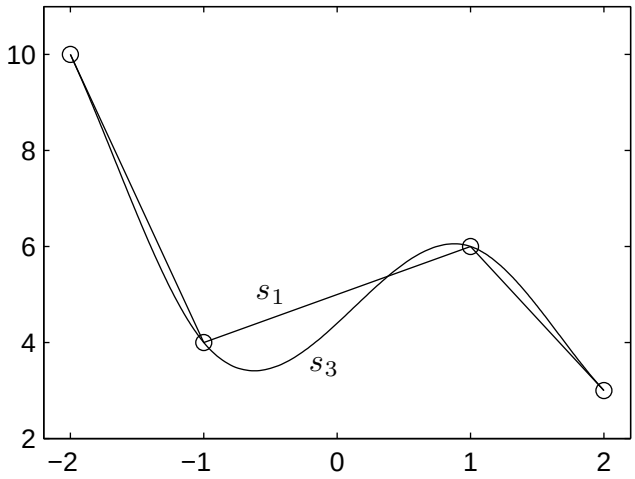
které můžeme psát jako

$$\begin{pmatrix} 6 & 1 \\ 1 & 6 \end{pmatrix}\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} -21 \\ -9 \end{pmatrix}.$$

Vyřešením dostaneme $m_1 = -\frac{234}{70}$, $m_2 = -\frac{66}{70}$. Výsledný splajn zapíšeme podle (5.12) po částech:

$$s_3(x) = \begin{cases} 10\varphi_1(t) + 4\varphi_2(t) - 6\varphi_3(t) - \frac{117}{35}\varphi_4(t), & t = x + 2 \text{ pro } x \in \langle -2, -1\rangle, \\ 4\varphi_1(t) + 6\varphi_2(t) - \frac{234}{35}\varphi_3(t) - \frac{66}{35}\varphi_4(t), & t = (x+1)/2 \text{ pro } x \in \langle -1, 1\rangle, \\ 6\varphi_1(t) + 3\varphi_2(t) - \frac{33}{35}\varphi_3(t) - 3\varphi_4(t), & t = x - 1 \text{ pro } x \in \langle 1, 2\rangle, \end{cases}$$

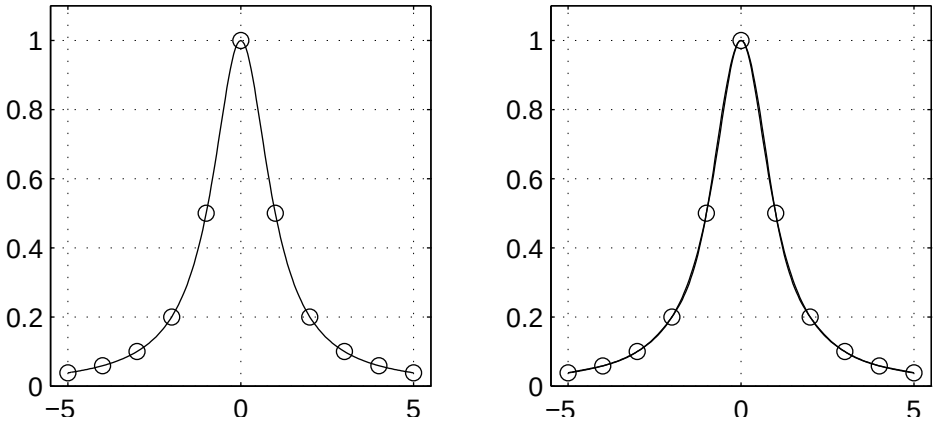
kde $\varphi_1(t) = 1 - 3t^2 + 2t^3$, $\varphi_2(t) = 3t^2 - 2t^3$, $\varphi_3(t) = t - 2t^2 + t^3$, $\varphi_4(t) = -t^2 + t^3$. Graf je znázorněn na Obrázku 5.3. □



Obrázek 5.3: Graf lineárního (s_1) a kubického interpolačního (s_3) splajnu.

Příklad 5.2.3 (*Rungeho příklad - pokračování*) Nakreslíme graf interpolačního kubického splajnu pro funkci f a uzly x_i z Příkladu 5.1.4 a porovnáme ho s grafem interpolačního polynomu.

Řešení: Na Obrázku 5.4 vidíme, že splajn s_3 neosciluje a je z tohoto pohledu mnohem rozumnější aproximací interpolované funkce f než interpolační polynom p_{10}; porovnej s Obrázkem 5.2.b. □



Obrázek 5.4: (a) funkce f; (b) Funkce f a kubický interpolační splajn s_3.

Kontrolní otázky

Otázka 1. Co je to splajn? Jak se definuje a počítá splajn lineární a kubický?
Otázka 2. Jak se chovají při interpolaci splajny v porovnání s polynomy?

Úlohy k samostatnému řešení

1. Pro uzly $x_0 = -1$, $x_1 = 0$, $x_2 = 2$, $x_3 = 3$, $x_4 = 5$ a funkční hodnoty $f_0 = -2$, $f_1 = 1$, $f_2 = 0$, $f_3 = 2$, $f_4 = -1$ sestavte lineární interpolační splajn.

2. Pro stejná data sestavte kubický interpolační splajn.

Výsledky úloh k samostatnému řešení

1.

$$s_1(x) = \begin{cases} -2\varphi_1(t) + \varphi_2(t), & t = x+1 \quad \text{pro } x \in \langle -1, 0\rangle, \\ \varphi_1(t), & t = x/2 \quad \text{pro } x \in \langle 0, 2\rangle, \\ 2\varphi_2(t), & t = x-2 \quad \text{pro } x \in \langle 2, 3\rangle, \\ 2\varphi_1(t) - 1\varphi_2(t), & t = (x-3)/2 \quad \text{pro } x \in \langle 3, 5\rangle, \end{cases}$$

kde $\varphi_1(t) = 1 - t$, $\varphi_2(t) = t$.

2. Krajní parametry jsou $m_0 = 3$, $m_4 = -\frac{3}{2}$, ostatní dostaneme ze soustavy

$$\begin{pmatrix} 6 & 1 & 0 \\ 1 & 6 & 2 \\ 0 & 2 & 6 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} \frac{21}{2} \\ \frac{21}{2} \\ 9 \end{pmatrix},$$

takže $m_1 = \frac{97}{62}$, $m_2 = \frac{69}{62}$, $m_3 = \frac{35}{31}$ a konečně

$$s_3(x) = \begin{cases} -2\varphi_1(t) + \varphi_2(t) + 3\varphi_3(t) + \frac{97}{62}\varphi_4(t), & t = x+1 \text{ pro } x \in \langle -1, 0\rangle, \\ \varphi_1(t) + \frac{97}{31}\varphi_3(t) + \frac{69}{31}\varphi_4(t), & t = x/2 \text{ pro } x \in \langle 0, 2\rangle, \\ 2\varphi_2(t) + \frac{69}{62}\varphi_3(t) + \frac{35}{31}3\varphi_4(t), & t = x-2 \text{ pro } x \in \langle 2, 3\rangle, \\ 2\varphi_1(t) - \varphi_2(t) + \frac{70}{31}\varphi_3(t) - 3\varphi_4(t), & t = (x-3)/2 \text{ pro } x \in \langle 3, 5\rangle, \end{cases}$$

kde $\varphi_1(t) = 1 - 3t^2 + 2t^3$, $\varphi_2(t) = 3t^2 - 2t^3$, $\varphi_3(t) = t - 2t^2 + t^3$, $\varphi_4(t) = -t^2 + t^3$.

5.3 Aproximace metodou nejmenších čtverců

V mnoha situacích, kdy je potřeba danou funkci f nahradit funkcí „jednodušší“, je nevhodné nebo vůbec nelze použít interpolaci. Jsou-li například funkční hodnoty nepřesné, přenese se tato nepřesnost i na celý interpolant. Často se také objevuje úloha, kdy se má velký počet naměřených dat „proložit“ přímkou nebo jinou jednoduchou funkcí. V těchto a v řadě dalších případů je rozumné použít metodu nejmenších čtverců.

Začněme příkladem.

Příklad 5.3.1 Mějme dány uzly $x_0 = -2$, $x_1 = -1$, $x_2 = 1$, $x_3 = 2$ a funkční hodnoty $f_0 = 10$, $f_1 = 4$, $f_2 = 6$, $f_3 = 3$. Najděte přímku

$$\varphi(x) = c_1 + c_2 x, \tag{5.22}$$

která je „blízko“ předepsaným hodnotám.

Řešení: Nejdříve musíme určit, co znamená „blízko“. V zápisu úlohy (5.2) jsme použili přibližné rovnosti „$\approx$“ a řekli jsme, jak jim budeme rozumět. Vyjádřeno vzorcem to znamená, že funkci φ chceme určit tak, aby minimalizovala výraz

$$\sum_{i=0}^{3} (\varphi(x_i) - f_i)^2.$$

Dosadíme-li sem jako φ předpis (5.22), dostaneme úlohu na výpočet minima funkce dvou proměnných:

$$\Psi(c_1, c_2) = \sum_{i=0}^{3} (c_1 + c_2 x_i - f_i)^2.$$

To už je ale úloha, kterou umíme řešit pomocí standardních postupů z diferenciálního počtu. Hodnoty c_1^*, c_2^*, pro něž funkce Ψ nabývá svého minima, vyhovují rovnicím

$$\frac{\partial \Psi}{\partial c_1}(c_1^*, c_2^*) = 0, \quad \frac{\partial \Psi}{\partial c_2}(c_1^*, c_2^*) = 0.$$

Výpočtem parciálních derivací odvodíme

$$2\sum_{i=0}^{3}(c_1^* + c_2^* x_i - f_i) = 0, \quad 2\sum_{i=0}^{3}(c_1^* + c_2^* x_i - f_i)x_i = 0,$$

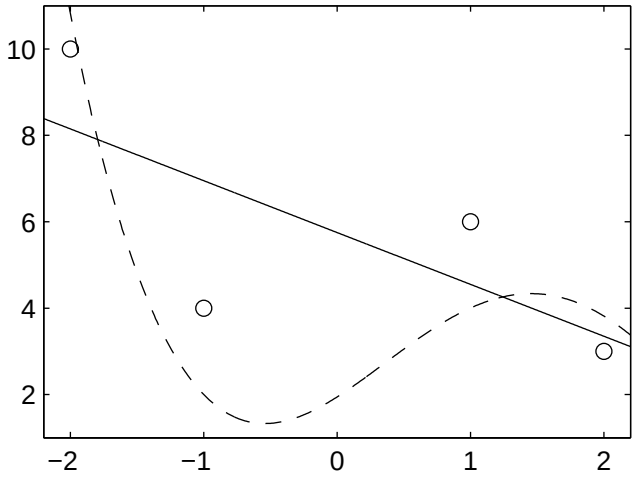
což je soustava lineárních rovnic

$$\begin{pmatrix} \sum_{i=0}^{3} 1 & \sum_{i=0}^{3} x_i \\ \sum_{i=0}^{3} x_i & \sum_{i=0}^{3} x_i^2 \end{pmatrix} \begin{pmatrix} c_1^* \\ c_2^* \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{3} f_i \\ \sum_{i=0}^{3} f_i x_i \end{pmatrix}, \text{ tj. } \begin{pmatrix} 4 & 0 \\ 0 & 10 \end{pmatrix} \begin{pmatrix} c_1^* \\ c_2^* \end{pmatrix} = \begin{pmatrix} 23 \\ -12 \end{pmatrix}.$$

Z ní vypočítáme $c_1^* = 23/4$ a $c_2^* = -6/5$. Přímka φ^*, která řeší naši úlohu, má tvar:

$$\varphi^*(x) = \frac{23}{4} - \frac{6}{5}x \tag{5.23}$$

a její graf je znázorněn na Obrázku 5.5. Podrobněji si tento postup rozebereme v dalším textu, z čehož mimo jiné vyplyne, že jsme opravdu našli minimum. □



Obrázek 5.5: Aproximace metodou nejmenších čtverců; přímka (5.23) plně; funkce (5.29) čárkovaně.

Postup z příkladu je jádrem obecné metody nejmenších čtverců pro úlohy (5.2). Budeme předpokládat, že je dán systém funkcí $\varphi_j = \varphi_j(x)$, $j = 1, \ldots, m$ a budeme uvažovat všechny funkce ve tvaru

$$\varphi(x) = c_1\varphi_1(x) + \cdots + c_m\varphi_m(x) = \sum_{j=1}^{m} c_j\varphi_j(x), \tag{5.24}$$

kde koeficienty $c_1, \ldots, c_m$ jsou libovolná reálná čísla. Funkci φ^*, pro niž platí

$$\sum_{i=0}^{n} (\varphi^*(x_i) - f_i)^2 \leq \sum_{i=0}^{n} (\varphi(x_i) - f_i)^2 \qquad \forall \varphi \tag{5.25}$$

nazýváme *aproximací podle metody nejmenších čtverců*. Její koeficienty $c_1^*, \ldots, c_m^*$ určíme jako minimum funkce

$$\Psi(c_1, \ldots, c_m) = \sum_{i=0}^{n} (\sum_{j=1}^{m} c_j \varphi_j(x_i) - f_i)^2, \tag{5.26}$$

které vyhovuje rovnicím

$$\frac{\partial \Psi}{\partial c_k}(c_1^*, \ldots, c_m^*) = 0, \quad k = 1, \ldots, m. \tag{5.27}$$

Vyjádříme-li parciální derivace

$$\frac{\partial \Psi}{\partial c_k} = 2 \sum_{i=0}^{n} (\sum_{j=1}^{m} c_j \varphi_j(x_i) - f_i) \varphi_k(x_i)$$

a dosadíme je do (5.27), dostaneme po jednoduché úpravě soustavu lineárních rovnic

$$\sum_{j=1}^{m} \left(\sum_{i=0}^{n} \varphi_j(x_i) \varphi_k(x_i) \right) c_j = \sum_{i=0}^{n} f_i \varphi_k(x_i), \quad k = 1, \ldots, m \tag{5.28}$$

pro výpočet koeficientů $c_1^*, \ldots, c_m^*$. Soustava (5.28) se nazývá *soustava normálních rovnic*.

Věta 5.3.1 Nechť jsou dány vzájemně různé uzly x_i a funkční hodnoty f_i, $i = 0, \ldots, n$. Nechť je dán systém funkcí φ_j, $j = 1, \ldots, m$, které jsou lineárně nezávislé na množině uzlů. Potom existuje jediná funkce φ^*, která splňuje (5.25) a její koeficienty $c_1^*, \ldots, c_m^*$ jsou řešením soustavy normálních rovnic (5.28).

Důkaz: V bodě $c_1^*, \ldots, c_m^*$, který vyhovuje rovnicím (5.27), nabývá funkce Ψ minima, jestliže matice druhých derivací je symetrická a pozitivně definitní (kladná). Druhé derivace jsou určeny vzorci

$$\frac{\partial^2 \Psi}{\partial c_k \partial c_l} = 2 \sum_{i=0}^{n} \varphi_l(x_i) \varphi_k(x_i), \quad k, l = 1, \ldots, m,$$

z nichž je symetrie vidět na první pohled (prohozením indexů k a l se nic nezmění). Nechť $d_1, \ldots, d_m$ jsou čísla ne všechna současně nulová. Potom

$$\sum_{k=1}^{m} \sum_{l=1}^{m} d_k d_l \frac{\partial^2 \Psi}{\partial c_k \partial c_l} = 2 \sum_{i=0}^{n} \Big(\sum_{l=1}^{m} d_l \varphi_l(x_i) \Big) \Big(\sum_{k=1}^{m} d_k \varphi_k(x_i) \Big) = 2 \sum_{i=0}^{n} \tilde{\varphi}(x_i)^2 > 0,$$

kde $\tilde{\varphi}(x) = \sum_{k=1}^{m} d_k \varphi_k(x)$, takže matice druhých derivací je pozitivně definitní. Odtud také plyne, že matice soustavy normálních rovnic je konstantní vzhledem k $c_1, \ldots, c_m$ a regulární. Proto má (5.28) jediné řešení $c_1^*, \ldots, c_m^*$, které definuje jedinou funkci φ^* řešící naši úlohu (5.25). □

Při aproximaci metodou nejmenších čtverců se musíme nejdříve rozhodnout pro nějaký lineárně nezávislý systém funkcí φ_j, $j = 1, \ldots, m$. Poté stačí sestavit a vyřešit soustavu normálních rovnic (5.28).

Příklad 5.3.2 Napište normální soustavu lineárních rovnic odpovídající systému funkcí

$$\varphi_1(x) = e^{-x}, \quad \varphi_2(x) = \sin x.$$

Aproximujte data z Příkladu 5.3.1

Řešení: Soustava normálních rovnic má v tomto případě tvar

$$\begin{pmatrix} \sum_{i=0}^{n} e^{-2x_i} & \sum_{i=0}^{n} e^{-x_i}\sin x_i \\ \sum_{i=0}^{n} e^{-x_i}\sin x_i & \sum_{i=0}^{n} \sin^2 x_i \end{pmatrix} \begin{pmatrix} c_1^* \\ c_2^* \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{n} f_i e^{-x_i} \\ \sum_{i=0}^{n} f_i \sin x_i \end{pmatrix}.$$

Po dosazení dostaneme

$$\begin{pmatrix} 62.1409 & -8.5736 \\ -8.5736 & 3.0698 \end{pmatrix} \begin{pmatrix} c_1^* \\ c_2^* \end{pmatrix} = \begin{pmatrix} 87.3770 \\ -4.6821 \end{pmatrix}$$

a odtud vypočítáme $c_1^* = 1.9452$, $c_2^* = 3.9076$, tj.

$$\varphi^*(x) = 1.9452e^{-x} + 3.9076 \sin x. \tag{5.29}$$

Graf je znázorněn na Obrázku 5.5.

Kontrolní otázky

Otázka 1. Kdy je vhodné použít metodu nejmenších čtverců?
Otázka 2. Graficky znázorněte smysl výrazu pro součet druhých mocnin odchylek?
Otázka 3. Co je to soustava normálních rovnic a jak vznikne?
Otázka 4. Co se stane, když v (5.24) a (5.25) bude $m = n + 1$?

Úlohy k samostatnému řešení

1. Napište soustavu normálních lineárních rovnic pro systém funkcí $\varphi_1(x) = 1$, $\varphi_2(x) = x$, $\varphi_3(x) = x^2$.
2. Data $x_0 = -1$, $x_1 = 0$, $x_2 = 2$, $x_3 = 3$, $x_4 = 5$ a $f_0 = -2$, $f_1 = 1$, $f_2 = 0$, $f_3 = 2$, $f_4 = -1$ aproximujte metodou nejmenších čtverců pomocí systému funkcí z předchozí úlohy.

Výsledky úloh k samostatnému řešení

1. Normální soustava má tvar:

$$\begin{pmatrix} \sum_{i=0}^{n} 1 & \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 \\ \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 & \sum_{i=0}^{n} x_i^4 \end{pmatrix} \begin{pmatrix} c_1^* \\ c_2^* \\ c_3^* \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{n} f_i \\ \sum_{i=0}^{n} f_i x_i \\ \sum_{i=0}^{n} f_i x_i^2 \end{pmatrix}.$$

2. $\varphi^*(x) = -0.2835x^2 + 1.2359x - 0.0130$.

KAPITOLA

6

NUMERICKÉ INTEGROVÁNÍ A DERIVOVÁNÍ

Při řešení řady praktických úloh je potřeba určit hodnotu určitého integrálu z nějaké funkce. V mnoha případech lze integraci provést pomocí analytických postupů, které se učí v základech integrálního počtu. Tak například snadno zjistíme, že integrály

$$\int_0^1 e^x \, dx \quad \text{a} \quad \int_0^\pi \cos x \, dx$$

mají hodnotu $e - 1$ a 0. Stačí však malá změna u integrované funkce a analytický výpočet se stane složitým nebo dokonce neproveditelným. Pokuste se vypočítat integrály

$$\int_0^1 e^{x^2} \, dx \quad \text{a} \quad \int_0^\pi \cos x^2 \, dx$$

a uvidíte jak daleko se vám podaří dojít! Je-li integrovaná funkce zadána tabulkou - například jako výsledek měření - pak se o analytickou integraci nelze ani pokusit.

V této kapitole se budeme nejprve zabývat numerickým integrováním. Úloha bude formulována obecně. Budeme předpokládat, že je dána spojitá funkce $f : \langle a, b \rangle \mapsto \mathbb{R}$ a máme aspoň přibližně určit číselnou hodnotu určitého integrálu

$$I = \int_a^b f(x) \, dx. \tag{6.1}$$

Numerické integrační vzorce lze znázornit geometricky, jako přibližnou velikost plochy určenou integrálem I (nakreslete si obrázek). Tyto vzorce používají jen několik funkčních hodnot, takže je lze použít téměř pro jakoukoliv funkci f, dokonce i pro funkce zadané tabulkou.

Při numerickém výpočtu derivace $f'(x)$ budeme předpokládat, že funkce f je na nějakém „rozumném“ okolí bodu x spojitě diferencovatelná. Užitečný bude opět geometrický význam derivace jakožto směrnice tečny ke grafu funkce f v bodě x. Vzorce pro numerické derivování počítají tuto směrnici přibližně z několika okolních funkčních hodnot. Jak uvidíme, narozdíl od numerického integrování přitom nelze dosáhnout příliš malé celkové chyby.

6.1 Newton-Cotesovy vzorce

Ze základního kurzu integrálního počtu víme, že je velmi jednoduché integrovat polynomy. Integrační vzorce pro obecnou funkci f zde proto odvodíme tak, že namísto ní budeme integrovat její interpolační polynom p_n:

$$I = \int_a^b f(x)\,dx \approx \int_a^b p_n(x)\,dx. \tag{6.2}$$

Pro jednoduchost sestavíme polynom p_n pomocí uzlů x_i rozložených na intervalu $\langle a, b\rangle$ rovnoměrně:

$$x_i = a + i\tau, \quad i = 0, 1, \ldots, n,$$

kde $\tau = (b-a)/n$. Interpolační polynom zapíšeme v Lagrangeově tvaru (viz Odstavec 5.1.1):

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\varphi_i(x), \quad \text{kde} \quad \varphi_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}.$$

Dosazením $p_n(x)$ do pravé strany přibližné rovnosti (6.2) dostaneme *Newton-Cotesovy vzorce*:

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n} w_i f(x_i), \tag{6.3}$$

kde

$$w_i = \int_a^b \varphi_i(x)\,dx, \quad i = 0, 1, \ldots, n \tag{6.4}$$

jsou *integrační váhy*. V další textu si podrobně odvodíme často používané Newton-Cotesovy vzorce pro interpolační polynomy nízkého stupně $n = 0, 1$ a 2 (zkuste si pak tyto vzorce nakreslit).

a) Obdélníkové pravidlo. Položíme $n = 0$, $x_0 = (a+b)/2$ a integrujeme konstantní polynom

$$p_0(x) = f\left(\frac{a+b}{2}\right).$$

Dostaneme:

$$\int_a^b f(x)\,dx \approx (b-a) f\left(\frac{a+b}{2}\right) = I_{Obd}. \tag{6.5}$$

b) Lichoběžníkové pravidlo. Položíme $n = 1$, $x_0 = a$, $x_1 = b$ a integrujeme lineární polynom

$$p_1(x) = \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b).$$

Dostaneme:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}\left(f(a) + f(b)\right) = I_{Lich}. \tag{6.6}$$

c) Simpsonovo pravidlo. Položíme $n = 2$, $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$ a integrujeme

kvadratický polynom $p_2(x)$. Integrační váhy zde snadno vypočítáme pomocí substituce $x = (b-a)t/2 + (a+b)/2$:

$$\begin{aligned} w_0 &= \int_a^b \varphi_0(x)\,dx = \int_a^b \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\,dx \\ &= \frac{b-a}{4}\int_{-1}^{1} t(t-1)\,dt = \frac{b-a}{6}; \end{aligned}$$

podobně dostaneme $w_1 = 4(b-a)/6$, $w_2 = (b-a)/6$ a dohromady máme:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) = I_{Simps}. \tag{6.7}$$

V příštích odstavcích budeme vzorce (6.5)-(6.7) nazývat jako jednoduché, protože z nich budeme vytvářet vzorce složené.

Příklad 6.1.1 Pomocí Newton-Cotesových vzorců (6.5), (6.6) a (6.7) vypočtěte přibližné hodnoty integrálu

$$I = \int_{-1}^{1} e^x\,dx.$$

Řešení: Máme $a = -1$, $b = 1$ a $f(x) = e^x$. Pomocí obdélníkového pravidla (6.5) je

$$I_{Obd} = 2e^0 = 2.$$

Lichoběžníkové pravidlo (6.6) dává

$$I_{Lich} = \frac{2}{2}\left(e^{-1} + e^1\right) = 3.086161.$$

Simpsonovým pravidlem (6.7) vypočteme

$$I_{Simps} = \frac{2}{6}\left(e^{-1} + 4e^0 + e^1\right) = 2.362054.$$

Přesná hodnota integrálu je $I = e^1 - e^{-1} = 2.350402$. □

Poznámka

Nejpřesnější hodnoty jsme dosáhli pomocí Simpsonova pravidla. To nás může vést k domněnce, že ještě přesnějších hodnot dosáhneme, použijeme-li Newton-Cotesovy vzorce odvozené z interpolačních polynomů vyšších stupnů. Obecně to však neplatí. V Příkladu 5.1.4 jsme viděli, že interpolační polynom vysokého stupně může být velmi špatnou aproximací funkce. Newton-Cotesovy vzorce dávají v takovém případě nesmyslné výsledky.

Následující věta se týká vyjádření integrační chyby.

Věta 6.1.1 *(i)* Nechť funkce f má na intervalu $\langle a, b\rangle$ spojitou druhou derivaci. Pak platí:

$$\begin{aligned} I - I_{Obd} &= \frac{f''(\xi)}{24}(b-a)^3, \\ I - I_{Lich} &= -\frac{f''(\xi)}{12}(b-a)^3. \end{aligned} \tag{6.8}$$

(ii) Nechť funkce f má na intervalu $\langle a, b\rangle$ spojitou čtvrtou derivaci. Pak platí:

$$I - I_{Simps} = -\frac{f^{(4)}(\xi)}{90}(b-a)^3.$$

Ve všech případech je ξ blíže neurčený bod z intervalu (a, b).

Důkaz: Důkaz ukážeme pouze pro lichoběžníkové pravidlo (6.6). Vyjádření interpolační chyby z Věty 5.1.2 má pro interpolační polynom p_1 s uzly $x_0 = a$, $x_1 = b$ tvar:

$$f(x) - p_1(x) = \frac{f''(\tilde{\xi})}{2}(x-a)(x-b).$$

Integrací a užitím věty o střední hodnotě integrálního počtu dostaneme

$$\begin{aligned} I - I_{Lich} &= \frac{f''(\xi)}{2}\int_a^b (x-a)(x-b)\,dx \\ &= \frac{f''(\xi)}{2}\int_0^{b-a} t(t+a-b)\,dt = -\frac{f''(\xi)}{12}(b-a)^3, \end{aligned}$$

kde jsme pro zjednodušení výpočtu použili substituci $x = t + a$. □

Kontrolní otázky

Otázka 1. Jak vzniknou Newton-Cotesovy vzorce?
Otázka 2. Jaké jsou základní pravidla pro numerický výpočet integrálu?
Otázka 3. Znázorněte graficky smysl obdélníkového a lichoběžníkového pravidla.
Otázka 4. Proč není rozumné používat Newton-Cotesovy vzorce odvozené z interpolačních polynomů vysokého stupně?

Úlohy k samostatnému řešení

1. Pomocí obdélníkového, lichoběžníkového a Simpsonova pravidla vypočtěte přibližné hodnoty integrálů

$$\text{a)}\ \int_0^1 e^{x^2}\,dx; \qquad \text{b)}\ \int_0^{\pi} \cos x^2\,dx.$$

Výsledky úloh k samostatnému řešení

1. a) $I_{Obd} = 1.284025$, $I_{Lich} = 1.859141$, $I_{Simps} = 1.475731$; b) $I_{Obd} = 1.961189$, $I_{Lich} = -0.675916$, $I_{Simps} = 1.082154$.

6.2 Složené vzorce a dvojný přepočet

V předchozím odstavci jsme odvodili tři *jednoduché* vzorce: jednoduché obdélníkové pravidlo (6.5), jednoduché lichoběžníkové pravidlo (6.6) a jednoduché Simpsonovo pravidlo (6.7). Pro přesnější výpočty se z těchto vzorců vytvářejí vzorce *složené*. Nyní si předvedeme jejich odvození.

Interval $\langle a, b\rangle$ rozdělíme pomocí uzlů $x_i = a + ih$, $i = 0, 1, \ldots m$ s krokem $h = (b-a)/m$ na m rovnoměrných úseků. Toto dělení použijeme pro následujíc rozklad integrálu I:

$$I = \int_a^b f(x)\,dx = \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} f(x)\,dx. \tag{6.9}$$

Jestliže každý dílčí integrál za sumací nahradíme vždy jednoduchým obdélníkovým nebo jednoduchým lichoběžníkovým pravidlem, dostaneme odpovídající pravidlo složené.

a) Složené obdélníkové pravidlo. Jednoduché obdélníkové pravidlo (6.5) použijeme takto:

$$\int_{x_{i-1}}^{x_i} f(x)\,dx \approx hf\left(\frac{x_{i-1}+x_i}{2}\right).$$

Dosazením do (6.9) dostáváme:

$$I_{SO} = h\sum_{i=1}^{m} f\left(\frac{x_{i-1}+x_i}{2}\right). \tag{6.10}$$

To pravidlo jsme si už odvodili a vyzkušeli v Odstavci 1.1, viz vzorec (1.1).

b) Složené lichoběžníkové pravidlo. Jednoduché lichoběžníkové pravidlo (6.6) použijeme takto:

$$\int_{x_{i-1}}^{x_i} f(x)\,dx \approx \frac{h}{2}\left(f(x_{i-1}) + f(x_i)\right).$$

Dosazením do (6.9) dostáváme:

$$\begin{aligned} I_{SL} &= \frac{h}{2}\left(f(x_0) + 2f(x_1) + \cdots + 2f(x_{m-1}) + f(x_m)\right) \\ &= \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{m-1} f(x_i) + f(x_m)\right). \end{aligned} \tag{6.11}$$

c) Složené Simpsonovo pravidlo. Protože jednoduché Simpsonovo pravidlo obshuje tři funkční hodnoty, pozměníme rozklad integrálu I z (6.9). Interval $\langle a, b\rangle$ nyní rozdělíme na $2m$ rovnoměrných úseků (sudý počet), přičemž je $h = (b-a)/(2m)$ a uzly jsou $x_i = a + ih$, $i = 0, 1, \ldots 2m$. Pak můžeme psát:

$$I = \int_a^b f(x)\,dx = \sum_{i=1}^{m} \int_{x_{2i-2}}^{x_{2i}} f(x)\,dx. \tag{6.12}$$

Jednoduché Simpsonovo pravidlo (6.7) použijeme takto:

$$\int_{x_{2i-2}}^{x_{2i}} f(x)\,dx \approx \frac{2h}{6}\left(f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})\right).$$

Dosazením do (6.12) dostáváme:

$$\begin{aligned} I_{SS} &= \frac{h}{3}\left(f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+2f(x_4)+\cdots+f(x_{2m})\right) \\ &= \frac{h}{3}\left(f(x_0)+4\sum_{i=1}^{m} f(x_{2i-1})+2\sum_{i=1}^{m-1} f(x_{2i})+f(x_{2m})\right). \end{aligned} \tag{6.13}$$

Příklad 6.2.1 Pomocí složených pravidel (6.10), (6.11) a (6.13) vypočtěte přibližné hodnoty integrálu

$$I=\int_{-1}^{1} e^x\,dx.$$

Interval $\langle -1,1\rangle$ přitom rozdělte na čtyři části.

Řešení: (a) Položíme $m=4$, takže krok bude $h=0.5$ a dostaneme uzly $x_0=-1$, $x_1=-0.5$, $x_2=0$, $x_3=0.5$ a $x_4=1$. Složené obdélníkového pravidlo (6.10) má tvar:

$$I_{SO}=0.5[e^{-0.75}+e^{-0.25}+e^{0.25}+e^{0.75}]\doteq 2.326096.$$

(b) Složené lichoběžníkového pravidlo (6.11) použijeme takto:

$$I_{SL}=\frac{0.5}{2}[e^{-1}+2e^{-0.5}+2e^{0}+2e^{0.5}+e^{1}]\doteq 2.399166.$$

(c) U složeného Simpsonova pravidla (6.13) vezmeme $m=2$, což znamená, že použijeme stejné dělení intervalu jako v předchozích případech (protože $h=2/(2m)=0.5$). Dostaneme:

$$I_{SS}=\frac{0.5}{3}[e^{-1}+4e^{-0.5}+2e^{0}+4e^{0.5}+e^{1}]\doteq 2.351195.$$

Připomeňme, že přesná hodnota integrálu je $I=2.350402$. □

V následující větě uvádíme řád složených integračních pravidel.

Věta 6.2.1 *(i)* Nechť funkce f má na intervalu $\langle a,b\rangle$ spojitou druhou derivaci. Složené obdélníkové a složené lichoběžníkové pravidlo jsou druhého řádu, takže platí:

$$\begin{aligned} |I-I_{SO}| &\le C_1h^2, \\ |I-I_{SL}| &\le C_2h^2. \end{aligned}$$

(ii) Nechť funkce f má na intervalu $\langle a,b\rangle$ spojitou čtvrtou derivaci. Složené Simpsonovo pravidlo je čtvrtého řádu, takže platí:

$$|I-I_{SS}| \le C_3h^4.$$

Kladné konstanty C_1, C_2 a C_3 jsou blíže neurčená nezáporná čísla, která nezávisí na velikosti kroku h.

Důkaz: Omezíme se opět jen na lichoběžníkové pravidlo. Stačí si uvědomit, že složené lichoběžníkové pravidlo obsahuje m-krát jednoduché lichoběžníkové pravidlo Výraz pro

chybu tedy odhadneme tak, že m-krát sečteme známý výraz pro chybu jednoduchého pravidla (6.8). Označíme $M = \max_{\zeta\in\langle a,b\rangle} |f''(\zeta)|$ a postupně dostáváme:

$$|I - I_{SL}| = \left|\sum_{i=1}^{m} -\frac{f''(\xi_i)}{12}h^3\right| \leq \sum_{i=1}^{m}\left|\frac{f''(\xi_i)}{12}h^3\right| \leq \frac{M}{12}h^2\sum_{i=1}^{m} h = \frac{M(b-a)}{12}h^2 = C_2h^2,$$

kde $C_2 = M(b-a)/12$. □

Příklad 6.2.2 Pomocí složených integračních pravidel (6.10), (6.11) a (6.13) vypočtěte přibližné hodnoty integrálu

$$I = \int_{-1}^{1} e^x\,dx$$

s krokem $h = 0.5$, $h = 0.25$, $h = 0.125$ a $h = 0.0625$ a porovnejte je s přesnou hodnotou.

Řešení: Přesná hodnota je $I = 2.350402387$. Přibližné hodnoty a jejich porovnání s přesnou hodnotou uvádíme v Tabulce 6.1. Z tabulky je vidět, že složené obdélníkové a složené lichoběžníkové pravidlo se chovají podobně, zatímco složené Simpsonovo pravidlo dává výsledky o několik řádů přesnější. □

h	I_{SO}	$\lvert I - I_{SO}\rvert$	I_{SL}	$\lvert I - I_{SL}\rvert$	I_{SS}	$\lvert I - I_{SS}\rvert$
0.5	2.326096	0.024306	2.399166	0.048764	2.351195	0.000792
0.25	2.344293	0.006110	2.362631	0.012229	2.350453	0.000051
0.125	2.348873	0.001530	2.353462	0.003060	2.350406	0.000003
0.0625	2.350020	0.000383	2.351167	0.000765	2.350402	0.000000

Tabulka 6.1: Složené integrační vzorce.

Při výpočtu přibližné hodnoty integrálu se obvykle požaduje, aby chyba mezi přesnou hodnotou I a její aproximací $I(h)$ vypočítanou některým složeným integračním pravidlem s krokem h byla nejvýše rovna odhadu $\epsilon > 0$. Přesnou hodnotu I ale neznáme. Situace je podobná jako u iteračních metod, kdy jsme takovouto chybu posuzovali ukončovacím kritériem. Zde si iterační výpočet musíme nejdříve zavést. Postupujeme tak, že počítáme stále přesnější hodnoty integrálů pro zmenšující se kroky h a zjišťujeme, jak se od sebe liší. Z praktických důvodů používá půlení kroku h. O dosažení požadované přesnosti můžeme rozhodnout podle „ukončovacího kritéria“:

$$|I(h) - I(2h)| \leq \epsilon. \tag{6.14}$$

Této metodě se říká *dvojný přepočet*.

Algoritmus: Dvojný přepočet
Vstup: f, a, b, m, ϵ.
Polož $h := (b-a)/m$ a vypočti $I(h)$.
Opakuj
    polož $h := h/2$;
    vypočti $I(h)$;

dokud $|I(h) - I(2h)| > \epsilon$.
Výstup: $I = I(h) \pm \epsilon$.

Příklad 6.2.3 Pomocí dvojného přepočtu vypočtěte přibližnou hodnotu integrálu

$$I = \int_{-1}^{1} e^x \, dx$$

s přesností $\epsilon = 0.0001$. Použijte složené lichoběžníkové pravidlo a začněte s dělením integračního intervalu na 4 úseky.

Řešení: Průběh výpočtu je uveden v Tabulce 6.2. Jeho výsledkem je $I = 2.3504 \pm 0.0001$. □

m	h	$I(h)$	$\lvert I(2h) - I(h)\rvert$
4	0.5000000	2.3991662	—
8	0.2500000	2.3626313	0.0365349
16	0.1250000	2.3534620	0.0091693
32	0.0625000	2.3511674	0.0022945
64	0.0312500	2.3505936	0.0005737
128	0.0156250	2.3504502	0.0001434
256	0.0078125	2.3504143	$0.0000358 < 0.0001$

Tabulka 6.2: Dvojný přepočet.

Kontrolní otázky

Otázka 1. Jak se odvozují složené integrační vzorce?
Otázka 2. Jakého řádu jsou základní složená integrační pravidla?
Otázka 3. Vysvětlete princip dvojného přepočtu. Nakreslete k tomu obrázek.

Úlohy k samostatnému řešení

1. Pomocí složeného obdélníkového, lichoběžníkového a Simpsonova pravidla vypočtěte přibližné hodnotu integrálů

$$\text{a)} \int_0^1 e^{x^2} \, dx \quad \text{pro } h = 0.1; \qquad \text{b)} \int_0^{\pi} \cos x^2 \, dx \quad \text{pro } h = \pi/10.$$

2. Pomocí dvojného přepočtu vypočtěte hodnoty integrálů z předchozí úlohy s přesností $\epsilon = 0.0001$ a použijte přitom složené lichoběžníkové pravidlo.

Výsledky úloh k samostatnému řešení

1. a) $I_{SO} = 1.460393$, $I_{SL} = 1.467175$, $I_{SS} = 1.462681$; b) $I_{SO} = 0.553752$, $I_{SL} = 0.588876$, $I_{SS} = 0.566030$.
2. Požadované přesnosti dosáhneme: a) pro $m = 128$, kdy $I = 1.462679 \pm 10^{-4}$; b) pro $m = 512$, kdy $I = 0.565702 \pm 10^{-4}$.

6.3 Extrapolace při výpočtu integrálu

Efektivnější použití dvojného přepočtu s opírá o extrapolační vzorce, které pomocí jednoduché početní operace umožňují zvýšit řád použité integrační metody. Tyto vzorce nejdříve odvodíme. Na hodnotu $I(h)$ lze nahlížet jako na funkci v proměnné h. Pro tyto funkci můžeme napsat Taylorův rozvoj. Odhad chybu podle Věty 6.2.1 ukazuje, že v tomto rozvoji se budou vyskytovat mocniny h odpovídající řádu příslušného integračního pravidla p a vyšší, tj.

$$I(h) = I + Ch^p + \mathcal{O}(h^r), \tag{6.15}$$

kde $r > p$. Jestliže vzorec (6.15) napíšeme pro $2h$ dostaneme

$$I(2h) = I + 2^p Ch^p + \mathcal{O}(h^r), \tag{6.16}$$

protože $\mathcal{O}((2h)^r) = \mathcal{O}(h^r)$. Vyeliminujeme-li z rovností (6.15) a (6.16) výraz Ch^p dostáváme

$$I = I(h) + \frac{I(h) - I(2h)}{2^p - 1} + \mathcal{O}(h^r). \tag{6.17}$$

Odtud vidíme že:
(i) hodnota

$$I_1(h) = I(h) + \frac{I(h) - I(2h)}{2^p - 1} \tag{6.18}$$

je lepší aproximací I než $I(h)$, protože je vyššího řádu r;
(ii) výraz

$$E(h) = \frac{I(h) - I(2h)}{2^p - 1} \tag{6.19}$$

je aproximace chyby řádu p přibližné hodnoty integrálu $I(h)$, kterou můžeme použít také jako (pesimistický) odhad chyby aproximace $I_1(h)$. Postup zvyšování přesnosti podle vzorce (6.17) se nazývá *Richardsonova extrapolace*. Pomocí extrapolačních vzorců (6.17) a (6.18) jednoduše upravíme algoritmus dvojného přepočtu z Odstavce 6.2. Předposlední řádek bude obsahovat zápis „dokud $|E(h)| > \epsilon$“ a na posledním bude „Výstup: $I := I_1(h) \pm \epsilon$“.

Poznámka

V algoritmu potřebujeme znát řád p integračního pravidla. Podle Věty 6.2.1 je $p = 2$ pro složené obdélníkové a lichoběžníkové pravidlo a $p = 4$ pro složené Simpsonovo pravidlo.

Příklad 6.3.1 Pomocí dvojného přepočtu s extrapolací vypoččítejte přibližnou hodnotu integrálu

$$I = \int_{-1}^{1} e^x \, dx$$

s přesností $\epsilon = 10^{-4}$. Použijete složené lichoběžníkové pravidlo a začněte pro $m = 4$.

Řešení: Pro $p = 2$ mají extrapolační vzorce tvar

$$I_1(h) = I(h) + \frac{I(h) - I(2h)}{3} \quad \text{a} \quad E(h) = \frac{I(h) - I(2h)}{3}.$$

Průběh výpočtu je zaznamenán v Tabulce 6.3, kde jsou první tři sloupce stejné jako v Tabulce 6.2. Třetí sloupec obsahuje třetinové hodnoty, takže k ukončení dojde o jeden řádek dříve, což ale znamená ušetření zhruba poloviny výpočtů (proč?). Výpočet dokončíme tak, že z posledních dvou hodnot ve sloupci $I(h)$ vypočítáme zpřesněnou aproximaci

$$I_1(h) = 2.3504502 + \frac{2.3504502 - 2.3505936}{3} = 2.3504024.$$

Dostali jsme výsledek $I = 2.3504024 \pm 0.0001$. Porovnáním s přesnou hodnotou integrálu $I = 2.350402387...$ vidíme, že dosažená přesnost je velmi vysoká. □

m	h	$I(h)$	$\lvert E(h)\rvert$
4	0.5000000	2.3991662	—
8	0.2500000	2.3626313	0.0121783
16	0.1250000	2.3534620	0.0030564
32	0.0625000	2.3511674	0.0007649
64	0.0312500	2.3505936	0.0001913
128	0.0156250	2.3504502	0.0000478

Tabulka 6.3: Dvojný přepočet s extrapolací pomocí lichoběžníkového pravidla.

Příklad 6.3.2 V předchozím příkladu použijte složené Simpsonovo pravidlo.

Řešení: Použijeme extrapolační vzorce

$$I_1(h) = I(h) + \frac{I(h) - I(2h)}{15} \quad \text{a} \quad E(h) = \frac{I(h) - I(2h)}{15}.$$

Průběh výpočtu je zaznamenán v Tabulce 6.4. Pomocí extrapolace vypočítáme zpřesněnou aproximaci

$$I_1(h) = 2.3504530 + \frac{2.3504530 - 2.3511948}{15} = 2.350403.$$

Požadované přesnosti jsme dosáhli, i když celková přesnost je o něco menší než v předchozím výpočtu. Důležitá je ale celková výpočetní efektivita výpočtu, potřebovali jsem jenom devět funkčních hodnot integrované funkce. □

m	h	$I(h)$	$\lvert E(h)\rvert$
4	0.5000000	2.3511948	—
8	0.2500000	2.3504530	0.000049

Tabulka 6.4: Dvojný přepočet s extrapolací pomocí Simpsonova pravidla.

Kontrolní otázky

Otázka 1. Jak vzniknou extrapolační vzorce a jak se používají?
Otázka 2. Kolik funkčních hodnot bylo potřeba pro přibližný výpočet integrálu v Příkladech 6.2.3-6.3.2.

Úlohy k samostatnému řešení

1. Pomocí extrapolace a složeného Simpsonovo pravidla vypočtěte přibližnou hodnotu integrálů

$$\text{a)} \quad \int_0^1 e^{x^2}\,dx; \qquad \text{b)} \quad \int_0^\pi \cos x^2\,dx$$

s přesností $\epsilon = 0.0001$.

Výsledky úloh k samostatnému řešení

1. Požadované přesnosti dosáhneme: a) pro $m = 8$, kdy $I = 1.462658 \pm 10^{-4}$; b) pro $m = 32$, kdy $I = 0.565689 \pm 10^{-4}$.

6.4 Numerické derivování

Ukážeme, jak přibližně počítat hodnoty derivací $f'(x)$ a $f''(x)$ ze známých funkčních hodnot $f(x-h)$, $f(x)$ a $f(x+h)$, kde $h > 0$ je zadaný (malý) krok. Budeme postupovat podobně jako při odvozování Newton-Cotesových vzorců v Odstavci 6.1. K funkci f sestavíme interpolační polynom p_n a ten pak derivujeme místo této funkce. Použijeme přitom Newtonův tvar interpolačního polynomu (5.8).

1) Pro uzly $x_0 = x$ a $x_1 = x + h$ sestavíme lineární interpolační polynom a vyjádříme jeho první derivaci:

$$p_1(t) = f(x) + f[x+h, x](t-x) \quad \Rightarrow \quad p_1'(t) = f[x+h, x].$$

Z definice poměrných diferencí dostaneme

$$p_1'(x) = \frac{f(x+h) - f(x)}{h} \approx f'(x). \tag{6.20}$$

2) Pro uzly $x_0 = x - h$, $x_1 = x$ a $x_2 = x + h$ sestavíme kvadratický interpolační polynom:

$$p_2(t) = f(x-h) + f[x, x-h](t-x+h) + f[x+h, x, x-h](t-x+h)(t-x).$$

První a druhá derivace mají tvar

$$\begin{aligned} p_2'(t) &= f[x, x-h] + f[x+h, x, x-h](2t - 2x + h), \\ p_2''(t) &= 2f[x+h, x, x-h]. \end{aligned}$$

Dosazením $t = x$ a úpravou dostáváme

$$\begin{aligned} p_2'(x) &= \frac{f(x) - f(x-h)}{h} + \frac{f(x+h) - 2f(x) + f(x-h)}{2h^2} h \\ &= \frac{f(x+h) - f(x-h)}{2h} \approx f'(x) \end{aligned} \tag{6.21}$$

a

$$p_2''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \approx f''(x). \tag{6.22}$$

Následující příklad ukazuje, že použití těchto vzorců je snadné - stačí do nich dosadit.

Příklad 6.4.1 Vypočtěte přibližné hodnoty první a druhé derivace pro funkci $f(x) = \sin x$ v bodě $x = 1$ s krokem $h = 0.01$ pomocí vzorců (6.20), (6.21) a (6.22). Porovnejte je s přesnými hodnotami. Zaokrouhlujte přitom na šest desetinných míst.

Řešení: Budeme potřebovat funkční hodnoty $\sin 1 = 0.841471$, $\sin 1.01 = 0.846832$ a $\sin 0.99 = 0.836026$. Podle vzorec (6.20), resp. (6.21) dostaneme:

$$f'(1) \approx p_1'(1) = \frac{0.846832 - 0.841471}{0.01} = 0.536100,$$

resp.

$$f'(1) \approx p_2'(1) = \frac{0.846832 - 0.836026}{2 \cdot 0.01} = 0.540300.$$

Přesná hodnota je $f'(1) = \cos 1 = 0.540302$, takže druhá přibližná hodnota je přesnější. Podle vzorce (6.22) vypočítáme druhou derivaci:

$$f''(1) \approx p_2''(1) = \frac{0.846832 - 2 \cdot 0.841471 + 0.836026}{0.01^2} = -0.840000.$$

Nyní je přesná hodnota $f''(1) = -\sin 1 = -0.841471$. □

Následující věta ukazuje řád jednotlivých vzorců.

Věta 6.4.1 Nechť je funkce f dostatečně hladká (má v okolí bodu x spojitou druhou, třetí, resp. čtvrtou derivaci). Pak platí:

$$\begin{aligned} p_1'(x) - f'(x) &= h\frac{f''(\xi)}{2}, \\ p_2'(x) - f'(x) &= h^2\frac{f^{(3)}(\xi)}{3}, \\ p_2''(x) - f''(x) &= h^2\frac{f^{(4)}(\xi)}{12}. \end{aligned} \tag{6.23}$$

Ve všech případech je ξ blíže neurčený bod z okolí x.

Důkaz: Nejjednodušší způsob jak získat tato tvrzení je použít Taylorův rozvoj. Například pro odvození (6.23) stačí vyjádřit první derivaci z

$$f(x+h) = f(x) + hf'(x) + h^2\frac{f''(\xi)}{2}.$$

□

Výpočet přibližných hodnot derivací mohou podstatně ovlivnit zaokrouhlovací chyby. Jmenovatele vzorců totiž obsahují parametr h, který musí být malý, abychom dostali dostatečně přesnou aproximaci derivace s malou diskretizační chybou. Současně ovšem malé

číslo ve jmenovateli zlomku zvětšuje zaokrouhlovací chyby v čitateli. Největší dosažitelná přesnost je proto jistým kompromisem mezi chybou diskretizační, která při zmenšujícím se h klesá, a zaokrouhlovací, která přitom roste. Na příkladu vzorce (6.20) si ukážeme analýzu tohoto problému.

Označme $E_{celk}(h)$ horní odhad celkové chyby, který vznikne jako součet odhadu chyby diskretizační $e(h)$ a chyby zaokrouhlovací $z(h)$:

$$E_{celk}(h) = e(h) + z(h).$$

Podle (6.23) je $e(h) = Ch$, kde $C > 0$. Dále budeme předpokládat, že při výpočtu každé funkční hodnoty $f(x)$ dostaneme vlivem zaokrouhlení porušenou hodnotu $f^*(x)$ a že velikost této poruchy lze odhadnou kontantou κ, která souvisí s úrovní zaokrouhlování (hodnotou počítačovým epsilon):

$$|f^*(x) - f(x)| \leq \kappa.$$

Pro vzorec (6.20) tak dostáváme následující odhad:

$$\left| \frac{f(x+h) - f(x)}{h} - \frac{f^*(x+h) - f^*(x)}{h} \right| \leq$$

$$\leq \frac{1}{h} \left(|f^*(x) - f(x)| + |f^*(x+h) - f(x+h)| \right) \leq \frac{2\kappa}{h} = z(h).$$

Odhad celkové chyby má tedy tvar:

$$E_{celk}(h) = Ch + \frac{2\kappa}{h},$$

viz Obrázek 6.1. Snadno lze odvodit (derivováním podle h), že funkce $E_{celk}(h)$ má jediné minimum v bodě

$$h_{opt} = \sqrt{\frac{2\kappa}{C}} \tag{6.24}$$

a odpovídající hodnota $E_{celk}(h_{opt})$ představuje nejmenší horní odhad celkové chyby. Při výpočtu derivace podle (6.20) tedy dojde pro h menší než h_{opt} paradoxně ke ztrátě přesnosti.

Příklad 6.4.2 Vypočtěte přibližnou hodnotu první derivace funkce $f(x) = \sin x$ v bodě $x = 1$ s krokem $h = 0.01$ a $h = 0.001$ pomocí vzorce (6.20). Zaokrouhlujte přitom na čtyři desetinná místa. Výsledky pak diskutujte vzhledem k hodnotě optimálního kroku h_{opt}.

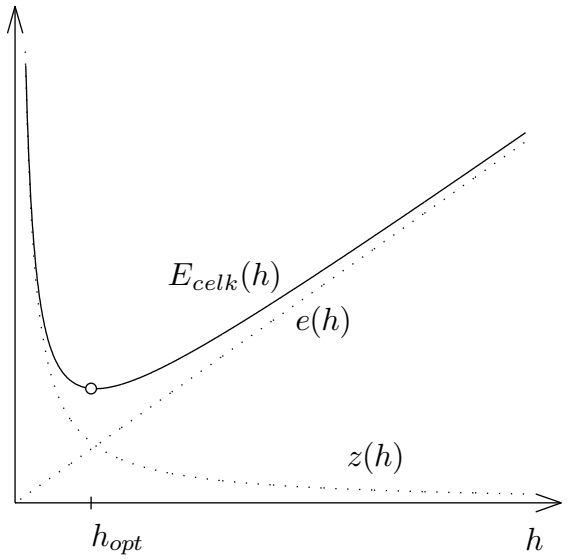
Řešení: V našem případě je $\kappa = 0.5 \cdot 10^{-4}$ a

$$\left| \frac{f''(\xi)}{2} \right| = \left| \frac{-\sin \xi}{2} \right| \leq 0.5 = C.$$

Dosazením do vzorce (6.24) vypočítáme $h_{opt} = 0.014142$. Oba naše kroky $h = 0.01$ i $h = 0.001$ spadají do intervalu $\langle 0, h_{opt} \rangle$, v němž dominuje zaokrouhlovací chyba. Z průběhu odhadu celkové chyby na tomto intervalu plyne, že přibližná hodnot derivace vypočítaná pro menší krok $h = 0.001$ bude méně přesná. Výpočtem dostáváme tyto hodnoty:

$$f'(1) \approx \frac{\sin(1.01) - \sin(1)}{0.01} \doteq \frac{0.8468 - 0.8415}{0.01} = 0.53,$$

$$f'(1) \approx \frac{\sin(1.001) - \sin(1)}{0.001} \doteq \frac{0.8420 - 0.8415}{0.001} = 0.50.$$



Obrázek 6.1: Odhad celkové chyby $E_{celk}(h)$.

Jejich porovnání s přesnou hodnotou $f'(1) = \cos 1 = 0.540302$ potvrzuje předchozí analýzu. □

Kontrolní otázky

Otázka 1. Jak se odvozují vzorce pro přibližný výpočet derivace?
Otázka 2. Znázorněte graficky smysl vzorců pro výpočet první derivace.
Otázka 3. Jak ovlivňují výpočet derivací zaokrouhlovací chyby?
Otázka 4. Odvoďte podrobně vztah (6.24).

Úlohy k samostatnému řešení

1. Vypočtěte přibližně první a druhou derivaci funkce $f(x) = \cos x$ v bodě $x = 1.5$ s krokem $h = 0.01$.

Výsledky úloh k samostatnému řešení

1. $f'(1.5) \approx -0.997832$ podle vzorce (6.20); $f'(1.5) \approx -0.997478$ podle vzorce (6.21); $f''(1.5) \approx -0.070737$ podle vzorce (6.22).

KAPITOLA

7

OBYČEJNÉ DIFERENCIÁLNÍ ROVNICE: POČÁTEČNÍ ÚLOHY

Diferenciální rovnice jsou nástrojem modelování ve fyzice, v chemii, v biologii, v technických i sociálních vědách a v mnoha dalších oblastech. Jen velmi málo diferenciálních rovnic lze řešit analyticky, takže numerické metody jsou většinou jedinou možností jak vypočítat řešení. Vývoj výkonných numerických algoritmů spojený s masivním využitím počítačů se u mnoha typů diferenciálních rovnic stal v posledních desetiletích předmětem intenzivního výzkumu.

V této kapitole se budeme věnovat počátečním úlohám pro obyčejné diferenciální rovnice prvního řádu, tzv. Cauchyově úloze. Nejdříve si připomeneme některé analytické metody na řešení jisté konkrétní úlohy, kterou budeme používat jako testovací příklad. Pak postupně projdeme jednotlivé numerické metody a na řešení testovacího příkladu si budeme demonstrovat jejich vlastnosti - zejména dosažení co největší přesnosti při co nejmenším objemu výpočtů. Ve standardních počítačových knihovnách (například v MATLABu) lze mnohé z těchto metody nalézt s dalším vylepšením, jako je adaptivní kontrola přesnosti nebo ošetření numerické nestability. Tato problematika ale překračuje rámec našeho textu.

7.1 Formulace úlohy

Budeme se zabývat numerickým výpočtem spojitě diferencovatelné funkce $y = y(x)$, která na intervalu $\langle a, b \rangle$ vyhovuje rovnici

$$y'(x) = f(x, y(x)), \tag{7.1}$$

kde $f = f(x, y)$ je daná pravá strana. Rovnice (7.1) je *obyčejná diferenciální rovnice prvního řádu*. K této rovnici připojujeme *počáteční podmínku*, což je rovnice ve tvaru

$$y(a) = c, \tag{7.2}$$

kde c je předepsané reálné číslo. Dvojice rovnic (7.1), (7.2) se nazývá *Cauchyova úloha*. Existence řešení závisí na vlastnostech pravé strany f. O této funkci budeme v celé kapitole

předpokládat, že je spojitá v první proměnné a splňuje *Lipschitzovu podmínku* ve druhé proměnné, tj. existuje konstanta $L > 0$ (nezávislá na x a y) taková, že platí

$$|f(x,y) - f(x,z)| \leq L|y - z| \quad \forall x \in \langle a,b\rangle \quad \forall y,z \in \mathbb{R}. \tag{7.3}$$

Za těchto předpokladů má Cauchyova úloha jediné řešení.

Příklad 7.1.1 Prozkoumejte otázku existence a jednoznačnosti u Cauchyovy úlohy

$$y' = y^2, \quad y(0) = 1$$

na intervalu $\langle 0,2\rangle$.

Řešení: Pravá strana diferenciální rovnice je dána funkcí $f(x,y) = y^2$. V první proměnné je funkce f konstantní, takže i spojitá. Pro druhou proměnou dostáváme:

$$\frac{f(x,y) - f(x,z)}{y - z} = \frac{y^2 - z^2}{y - z} = y + z.$$

Protože výraz $y + z$ může nabývat libovolně velké hodnoty, nemůže existovat konstanta L vyhovující vztahu (7.3). Funkce f proto nesplňuje Lipschitzovu podmínku ve druhé proměnné, a proto nemusí existovat jediné řešení. Snadným výpočtem lze zjistit, že funkce $y(x) = 1/(1-x)$ je řešením ale jen na intervalu $\langle 0,1)$, na celém intervalu $\langle 0,2\rangle$ řešení neexistuje (nakreslete si $y(x)$). □

Ve druhém příkladu si připomeneme některé analytické postupy řešení. Úlohu z tohoto příkladu použijeme budeme používat jako modelový příklad při testování numerických metod.

Příklad 7.1.2 Ověřte, že Cauchyova úloha

$$y' = x^2 - 0.2y, \quad y(-2) = -1 \tag{7.4}$$

má jediné řešení na intervalu $\langle -2,3\rangle$. Toto řešení vypočtěte pomocí analytických metod.

Řešení: Pravá strana $f(x,y) = x^2 - 0.2y$ je kvadratickou funkcí v první proměnné, takže je v této proměnné spojitá. Pro druhou proměnou dostáváme:

$$\frac{f(x,y) - f(x,z)}{y - z} = \frac{x^2 - 0.2y - x^2 + 0.2z}{y - z} = -0.2.$$

Odtud vidíme, že (7.3) je splněno například pro konstantu $L = 0.2$. Řešení zadané Cauchyovy úlohy je proto jediné. Při jeho výpočtu nejdříve vyřešíme homogenní diferenciální rovnici $u' = -0.2u$ pomocí separace proměnných. Postup je následující:

$$\frac{du}{dx} = -0.2u \;\Rightarrow\; \frac{du}{u} = -0.2\,dx \;\Rightarrow\; \int \frac{du}{u} = \int -0.2\,dx \;\Rightarrow$$

$$\ln|u| = -0.2x + C_1 \;\Rightarrow\; |u| = e^{-0.2x+C_1} \;\Rightarrow$$

$$u(x) = Ce^{-0.2x}, \quad C > 0.$$

Pro výpočet partikulárního řešení y_p rovnice (7.4) použijeme metodu variace konstant. Předpokládáme, že řešení bude ve tvaru $y_p(x) = C(x)e^{-0.2x}$. Dosazením do diferenciální rovnice (7.4) dostaneme:

$$C'(x)e^{-0.2x} - 0.2C(x)e^{-0.2x} = x^2 - 0.2C(x)e^{-0.2x} \;\Rightarrow\; C'(x) = x^2e^{0.2x}$$

a odtud integrací per partes vypočítáme:

$$C(x) = \int x^2e^{0.2x}\,dx = e^{0.2x}(5x^2 - 50x + 250),$$

takže $y_p(x) = 5x^2 - 50x + 250$. Obecné řešení dané diferenciální rovnice lze zapsat ve tvaru $y(x) = y_p(x) + u(x)$, tedy $y(x) = 5x^2 - 50x + 250 + Ce^{-0.2x}$. Nakonec z počáteční podmínky určíme konstantu C:

$$-1 = y(-2) = 20 + 100 + 250 + Ce^{0.4} \;\Rightarrow\; C = -248.688737.$$

Řešení naší výchozí úlohy má tvar:

$$y(x) = 5x^2 - 50x + 250 - 248.018417e^{-0.2x}. \tag{7.5}$$

□

Poznámka

Cauchyovu úlohu (7.1), (7.2) můžeme chápat vektorově tak, že hledáme vektorovou funkci $y(x) = (y_1(x), \ldots, y_n(x))^\top$ pro pravou stranu $f(x, y) = (f_1(x, y), \ldots, f_n(x, y))^\top$ a počáteční vektor $c = (c_1, \ldots, c_n)^\top$. Všechny numerické metody, s nimiž se seznámíme v této kapitole, lze použít i pro soustavy byčejných diferenciálních rovnic prvního řádu.

Poznámka

Ani omezení na rovnice prvního řádu není příliš podstatné. Z teorie diferenciálních rovnic je totiž známo, že počáteční úlohu pro rovnici vyššího řádu lze substitucí převést na soustavu rovnic prvního řádu.

Kontrolní otázky

Otázka 1. Jak vypadá obecná formulace Cauchyovy úlohy a kdy má tato úloha jediné řešení?
Otázka 2. Zopakujte si analytické metody výpočtu řešení u Cauchyovy úlohy.

7.2 Eulerova metoda

Odvození numerických metod pro řešení Cauchyovy úlohy je založeno na diskretizaci. Postupně se vytváří rostoucí posloupnost uzlů $x_0 = a$, x_1, x_2, ... a pro ně se rovněž postupně počítají hodnoty $y_0 = c, y_1, y_2, \ldots$ aproximující přesné řešení $y(x_0), y(x_1), y(x_2), \ldots$. Vzorce pro tyto výpočty mají rekurentní charakter. Pro jednoduchost budeme předpokládat, že uzly jsou rozloženy rovnoměrně s krokem h a platí $x_n = b$, takže $h = (b - a)/n$ a $x_i = a + ih$ pro $i = 0, 1, \ldots, n$.

Při odvozování vzorců můžeme postupovat například tak, že diferenciální rovnici (7.1) zapíšeme pro uzel x_i a nahradíme přesnou hodnotu řešení $y(x_i)$ její aproximací y_i. Derivaci na levé straně pak vyjádříme pomocí vzorce numerického derivování (6.20):

$$y'(x_i) = f(x_i, y(x_i)) \quad \approx \quad \frac{y_{i+1} - y_i}{h} = f(x_i, y_i).$$

Odtud jsme schopni při známých hodnotách x_i a y_i vypočítat y_{i+1}. Tímto jednoduchým postupem jsme odvodili rekurentní vzorce pro *Eulerovu metodu*:

$$\left.\begin{array}{rcl} y_0 &=& c, \\ y_{i+1} &=& y_i + hf(x_i, y_i), \quad i = 0, 1, \ldots, n-1. \end{array}\right\} \tag{7.6}$$

Příklad 7.2.1 Počáteční úlohu (7.4) řešte pomocí Eulerovy metody s krokem $h = 1$ a 0.5. Výsledky porovnejte s přesným řešením.

Řešení: V našem případě je $x_0 = -2$, $y(-2) = y_0 = -1$ a $f(x, y) = x^2 - 0.2y$. Pro $h = 1$ dostáváme:

$$\begin{aligned} y_1 &= -1 + 1 \cdot ((-2)^2 - 0.2 \cdot (-1)) = 3.2, \\ y_2 &= 3.2 + 1 \cdot ((-1)^2 - 0.2 \cdot 3.2) = 3.56, \\ y_3 &= 2.848, \quad y_4 = 3.2784, \quad y_5 = 6.6227. \end{aligned}$$

Podobně pro $h = 0.5$ vypočítáme:

$$\begin{aligned} y_1 &= -1 + 0.5 \cdot ((-2)^2 - 0.2 \cdot (-1)) = 1.1, \\ y_2 &= 1.1 + 0.5 \cdot ((-1.5)^2 - 0.2 \cdot 1.1) = 2.115, \\ y_3 &= 2.4035, \quad \ldots, \quad y_{10} = 7.4988. \end{aligned}$$

Obě přibližná řešení jsou uvedena v Tabulce 7.1, kde porovnáváme s přesným řešením $y(x)$ daným vzorcem (7.5). Z tabulky je vidět, že řešení vypočítané s menším krokem h je přesnější. Tento závěr potvrzuje i Obrázek 7.1, kde jsou přibližná řešení znázorněna jako lomené čáry. Přesné řešení $y(x)$ je zde znázorněno tečkovaně. Budeme-li postupně zmenšovat krok h (zvětšovat n), bude docházet k „vyhlazování“ lomených čar a k jejich přibližování k řešení přesnému $y(x)$. □

Poznámka

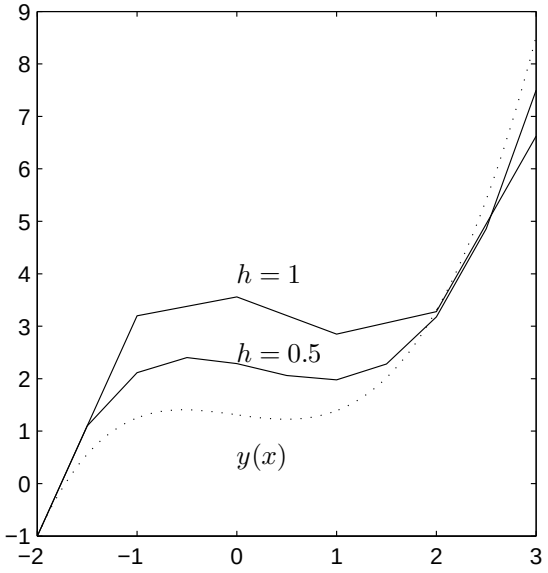
Eulerova metoda je *prvního řádu*, konverguje proto velmi pomalu. Chceme-li dosáhnout malé diskretizační chyby, musíme použít velmi malý krok h. Metody vyššího řádu vypočítají stejně kvalitní řešení s větším krokem h, což podstatně snižuje výpočetní nároky.

Kontrolní otázky

Otázka 1. Odvoďte Eulerovu metodu. Jakého je řádu?

$h = 1$				$h = 0.5$			
i	x_i	y_i	$\lvert y_i - y(x_i)\rvert$	i	x_i	y_i	$\lvert y_i - y(x_i)\rvert$
0	-2	-1	0	0	-2	-1	0
				1	-1.5	1.1	0.5447
1	-1	3.2	1.9491	2	-1	2.1150	0.8641
				3	-0.5	2.4035	0.9971
2	0	3.56	2.2487	4	0	2.2882	0.9769
				5	0.5	2.0593	0.8322
3	1	2.848	1.4571	6	1	1.9784	0.5875
				7	1.5	2.2806	0.2637
4	2	3.2784	0.0206	8	2	3.1775	0.1214
				9	2.5	4.8598	0.5529
5	3	6.6227	1.8940	10	3	7.4988	1.0179

Tabulka 7.1: Eulerova metoda pro $h = 1, 0.5$.



Obrázek 7.1: Přibližná řešení vypočítaná Eulerovou metodou s $h = 1, 0.5$ a přesné řešení $y(x)$ (tečkovaně).

Úlohy k samostatnému řešení

1. Eulerovou metodou řešte diferenciální rovnici $y' = 1/(x^2+1) - 0.1y$ s počáteční podmínkou $y(-2) = -1$ na intervalu $\langle -2, 3\rangle$ s krokem $h = 0.5$.

Výsledky úloh k samostatnému řešení

1. $y_0 = -1$, $y_1 = -0.85$, $y_2 = -0.6537$, $y_3 = -0.3710$, $y_4 = 0.0476$, $y_5 = 0.5452$, $y_6 = 0.9179$, $y_7 = 1.1220$, $y_8 = 1.2198$, $y_9 = 1.2588$, $y_{10} = 1.2648$.

7.3 Jednokrokové metody vyššího řádu

Rekurentní vzorce pro řešení Cauchyovy úlohy uvedené v tomto odstavci jsou řádu vyššího než jedna. Při jejich použití vypočítáme srovnatelně přesné řešení jako u Eulerovy metody mnohem rychleji.

Heunova metoda je druhého řádu. Je určena rekurentními vzorci:

$$\left.\begin{aligned} y_0 &= c, \\ k_1 &= hf(x_i, y_i), \\ k_2 &= hf(x_i + h, y_i + k_1), \\ y_{i+1} &= y_i + \tfrac{1}{2}(k_1 + k_2), \quad i = 0, \ldots, n-1. \end{aligned}\right\} \tag{7.7}$$

Rungeova-Kuttova metoda (RK4) je čtvrtého řádu. Je určena rekurentními vzorci:

$$\left.\begin{aligned} y_0 &= c, \\ k_1 &= hf(x_i, y_i), \\ k_2 &= hf(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1), \\ k_3 &= hf(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2), \\ k_4 &= hf(x_i + h, y_i + k_3), \\ y_{i+1} &= y_i + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad i = 0, \ldots, n-1. \end{aligned}\right\} \tag{7.8}$$

Obě tyto metody jsou jednokrokové. Máme tím namysli, že při výpočtu y_{i+1} potřebujeme použít hodnotu přibližného řešení y_i jen z jednoho předchozího rekurentního kroku. Ze vzorců je také vidět, že zvýšení řádu je spojeno s větším objemem výpočtů v jednom kroku: u metody druhého řádu se počítá hodnota pravé strany f ve dvou bodech, u metody čtvrtého řádu pak ve čtyřech bodech.

Příklad 7.3.1 Cauchyovu úlohu (7.4) řešte pomocí Heunovy metody a pomocí metody RK4 s krokem $h = 1$. Výsledky porovnejte s přesným řešením a s řešením vypočítaným Eulerovou metodou.

Řešení: Opět vyjdeme z $x_0 = -2$, $y(-2) = y_0 = -1$ a $f(x, y) = x^2 - 0.2y$. Naznačíme

první dva kroky Heunovy metody:

$$\begin{aligned} k_1 &= 1\cdot((-2)^2-0.2\cdot(-1)) = 4.2,\\ k_2 &= 1\cdot((-2+1)^2-0.2\cdot(-1+4.2)) = 0.36,\\ y_1 &= -1+\tfrac{1}{2}(4.2+0.36) = 1.28,\\ k_1 &= 1\cdot((-1)^2-0.2\cdot 1.28) = 0.744,\\ k_2 &= 1\cdot((-1+1)^2-0.2\cdot(1.28+0.744)) = -0.4048,\\ y_2 &= 1.28+\tfrac{1}{2}(0.744+(-0.4048)) = 1.4496,\\ &\text{atd.} \end{aligned}$$

Obě přibližná řešení zapíšeme do Tabulky 7.2 a porovnáme je s přesným řešením $y(x)$ určeným vzorcem (7.5). Porovnáním se sloupcem $h = 1$ v Tabulce 7.1 vidíme, že dosažené výsledky jsou přesnější než u Eulerovy metody, přičemž nejpřesnější výsledky dává metoda RK4, viz také Obrázek 7.2. □

Heunova metoda				Metoda RK4			
i	x_i	y_i	$\lvert y_i - y(x_i)\rvert$	i	x_i	y_i	$\lvert y_i - y(x_i)\rvert$
0	−2	−1	0	0	−2	−1	0
1	−1	1.2800	0.0291	1	−1	1.2508	0.0001
2	0	1.4496	0.1383	2	0	1.3112	0.0001
3	1	1.6887	0.2978	3	1	1.3910	0.0001
4	2	3.7847	0.4858	4	2	3.2994	0.0004
5	3	9.2035	0.6867	5	3	8.5175	0.0008

Tabulka 7.2: Jedokrokové metody vyššího řádu.

Kontrolní otázky

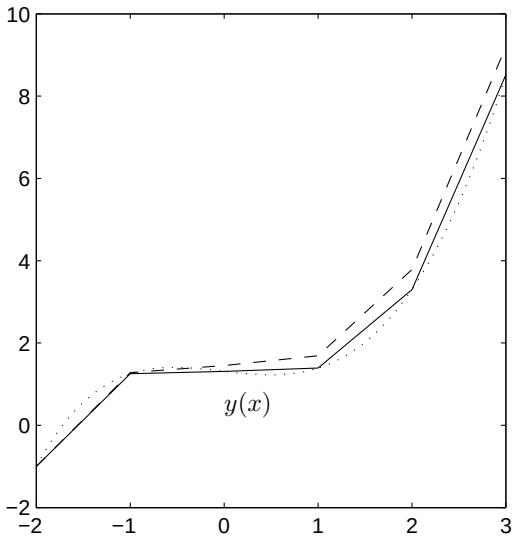
Otázka 1. Jaké znáte jednokrokové metody vyššího řádu? Co je jejich výhodou oproti metodám prvního řádu?
Otázka 2. Jaký je vztah mezi řádem metody a výpočetní náročností jednoho kroku?

Úlohy k samostatnému řešení

1. Diferenciální rovnici $y' = y(1+\sin x) - x^3$, $y(1) = 0$ řešte na intervalu $\langle 1, 2\rangle$ Heunovou metodou a metodou RK4 s krokem $h = 0.2$.

Výsledky úloh k samostatnému řešení

1. Heunova metoda: $y_0 = 0$, $y_1 = -0.3114$, $y_2 = -0.9732$, $y_3 = -2.2320$, $y_4 = -4.4495$, $y_5 = -8.1186$; metoda RK4: $y_0 = 0$, $y_1 = -0.3210$, $y_2 = -1.0087$, $y_3 = -2.3257$, $y_4 = -4.6586$, $y_5 = -8.5351$.



Obrázek 7.2: Přibližné řešení vypočítané metodou RK4 (plně), Heunovou metodou (čárkovaně) a přesné řešení (tečkovaně).

7.4 Vícekrokové metody

V tomto odstavci si ukážeme odvození některých vícekrokových metod. Zavedeme také pojmy, pomocí nichž lze vyjádřit vztah mezi řádem metody a složitostí výpočtu.

Definice 7.4.1 Rekurentní metoda pro řešení Cauchyovy úlohy se nazývá k-kroková, jestliže při výpočtu y_{i+1} potřebujeme v jednom rekurentním kroku použít hodnoty přibližného řešení z k předcházejících kroků, tj. hodnoty

$$y_i, y_{i-1}, \ldots, y_{i-k+1}.$$

Je zřejmé, že u k-krokové metody potřebujeme pro zahájení výpočtu použít hodnoty přibližného řešení

$$y_0, y_1, \ldots, y_{k-1},$$

kterým se říká *počáteční úsek délky* k. Jeho výpočet se provádí vhodnou jednokrokovou metodou, která by měla být aspoň stejného řádu, jako je používaná k-kroková metoda, aby nedošlo ke zbytečné ztrátě přesnosti.

Všechny dosud uvedené metody byly jednokrokové. Příkladem dvoukrokové metody je *metoda skákající žáby*:

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i), \tag{7.9}$$

kterou lze odvodit podobně jako Eulerovu metodu ze vzorce numerického derivování (6.21). Tato metoda je druhého řádu, protože stejného řádu je vzorec numerického derivování (6.21).

Definice 7.4.2 Rekurentní metoda pro řešení Cauchyovy úlohy se nazývá l-bodová, pokud při výpočtu y_{i+1} potřebujeme v jednom rekurentním kroku vypočítat hodnoty pravé strany f v l různých bodech.

Z rekurentních vzorců vyplývá, že Eulerova metoda (7.6) je jednobodová, Heunova metoda (7.7) je dvoubodová a RK4 metoda (7.8) je čtyřbodová. U jednokrokových metod je tedy dosažení vyššího řádu spojeno se zvětšením objemu výpočtů v jednom rekurentním kroku. U vícekrokových metod lze dosáhnout vyššího řádu, ačkoliv metoda zůstane jednobodová. Například metoda skákající žáby (7.9) je jednobodová a druhého řádu.

Všechnu dosud uvedené metody byly *explicitní*. Znamená to, že v jejich rekurentních vzorcích se y_{i+1} vyskytovalo pouze na levé straně vzorce. Z výpočetního hlediska to představuje příznivou situaci, kdy „stačí dosadit". U vícekrokových metod se často používají vzorce, které jsou *implicitní*, u nichž se y_{i+1} vyskytuje i na pravé straně vzorce jako argument funkce f. Při použití těchto vzorců je situace složitější, v každém rekurentním kroku je potřeba „řešit rovnici".

7.4.1 Adamsovy-Bashforthovy metody

Tímto názvem rozumíme skupinu explicitních vícekrokových rekurentních vzorců různých řádů, které lze odvodit integrací interpolačního polynomu sestaveného pro derivaci řešení diferenciální rovnice. Ukážeme si odvození tříkrokového vzorce.

Nechť $y(x)$ je řešením Cauchyovy úlohy a označme $y'_i = y'(x_i)$. Derivaci $y'(x)$ můžeme vyjádřit ve tvaru

$$y'(x) = p_2(x) + e(x), \tag{7.10}$$

kde $p_2(x)$ je Newtonův tvar interpolačního polynomu pro uzly x_i, x_{i-1}, x_{i-2} a $e(x)$ je interpolační chyba (viz (5.8) a (5.9)):

$$\begin{aligned} p_2(x) &= y'_i + y'[x_{i-1}, x_i](x - x_i) + y'[x_{i-2}, x_{i-1}, x_i](x - x_i)(x - x_{i-1}), \\ e(x) &= \frac{y^{(4)}(\tilde{\xi})}{3!}(x - x_i)(x - x_{i-1})(x - x_{i-2}). \end{aligned}$$

V (7.10) budeme integrovat přes interval $\langle x_i, x_{i+1}\rangle$:

$$\begin{aligned} \int_{x_i}^{x_{i+1}} y'(x)\,dx &= y(x_{i+1}) - y(x_i), \\ \int_{x_i}^{x_{i+1}} p_2(x)\,dx &= y'_i \cdot h + y'[x_{i-1}, x_i] \cdot \frac{h^2}{2} + y'[x_{i-2}, x_{i-1}, x_i] \cdot \frac{5h^3}{6} \\ &= y'_i \cdot h + \frac{y'_i - y'_{i-1}}{h} \cdot \frac{h^2}{2} + \frac{y'_i - 2y'_{i-1} + y'_{i-2}}{2h^2} \cdot \frac{5h^3}{6} \\ &= \frac{h}{12}(23y'_i - 16y'_{i-1} + 5y'_{i-2}), \\ \int_{x_i}^{x_{i+1}} e(x)\,dx &= \frac{y^{(4)}(\xi)}{3!} \cdot \frac{9}{4} \cdot h^4 = \frac{3}{8} \cdot h^4 \cdot y^{(4)}(\xi). \end{aligned}$$

Dohromady dostáváme:

$$y(x_{i+1}) - y(x_i) = \frac{h}{12}(23y'_i - 16y'_{i-1} + 5y'_{i-2}) + \frac{3}{8}h^4 y^{(4)}(\xi). \tag{7.11}$$

Vynecháním posledního členu vznikne rekurentní vzorec, který lze použít pro přibližný výpočet $y(x_{i+1})$. Stačí si uvědomit, že přibližné hodnoty derivací y'_i, y'_{i-1}, y'_{i-2} lze vypočítat

dosazením do pravé strany f diferenciální rovnice (7.1). Označíme-li $y_i \approx y(x_i)$ a $f_i = f(x_i, y_i) \approx y'_i$, obdržíme tento rekurentní vzorec:

$$y_{i+1} = y_i + \frac{h}{12}(23f_i - 16f_{i-1} + 5f_{i-2}).$$

Jedná se o tříkrokovou metodu třetího řádu a současně je to metoda jednobodová! Stačí si uvědomit, že při výpočtu y_{i+1} není nutné počítat hodnoty f_{i-1} a f_{i-2}, protože k tomu již došlo v předchozích krocích. Všimněme si ještě řádu diskretizační chyby této metody. *Lokální chyba*, které se dopouštíme v každém rekurentní kroku je čtvrtého řádu, viz h^4 v posledním členu v (7.11). Pro posouzení celkové přesnosti je ale důležitější *globální chyba*, která je řádu o jedna menšího, protože v průběhu rekurentního výpočtu dochází ke kumulaci lokálních chyb (v důkazu této vlastnosti se jedno h ztratí v rovnosti $nh = b - a$, podobně jako tomu bylo v důkazu Věty 6.2.1 při numerické integraci). Ztráta přesnosti o jeden řád je typická pro numerické řešení počátečních úloh.

Analogickým postupem lze odvodit další Adamsovy-Bashforthovy vzorce lišící se počtem kroků a řádem. Obvykle se používají vzorce nejvýše čtyřkrokové. Zde je jejich přehled:

$$y_{i+1} = y_i + hf_i, \tag{7.12}$$

$$y_{i+1} = y_i + \frac{h}{2}(3f_i - f_{i-1}), \tag{7.13}$$

$$y_{i+1} = y_i + \frac{h}{12}(23f_i - 16f_{i-1} + 5f_{i-2}), \tag{7.14}$$

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}). \tag{7.15}$$

Příklad 7.4.1 Počáteční úlohu (7.4) řešte pomocí tříkrokové Adamsovy-Bashforthovy metody s krokem $h = 1$. Výsledky porovnejte s přesným řešením.

Řešení: Počáteční úsek vypočítáme metodou RK4. Jsou to první tři hodnoty x_i a y_i z druhé části Tabulky 7.2, tj. $x_0 = -2$, $x_1 = -1$, $x_2 = 0$ a $y_0 = -1$, $y_1 = 1.2508$, $y_2 = 1.3112$. Protože je $f(x, y) = x^2 - 0.2y$, vypočítáme $f_0 = (-2)^2 - 0.2 \cdot (-1) = 4.2000$, $f_1 = (-1)^2 - 0.2 \cdot 1.2508 = 0.7498$ a $f_2 = (0)^2 - 0.2 \cdot 1.3112 = -0.2622$. Pomocí vzorce (7.14) pak provedeme všechny další výpočty:

$$\begin{aligned}
y_3 &= -1.3112 + \tfrac{h}{12}(23f_2 - 16f_1 + 5f_0) = 1.5588,\\
f_3 &= 1^2 - 0.2 \cdot 1.5588 = 0.6882,\\
y_4 &= 1.5588 + \tfrac{h}{12}(23f_3 - 16f_2 + 5f_1) = 3.5400,\\
f_4 &= 2^2 - 0.2 \cdot 3.5400 = 3.2920,\\
y_5 &= 3.5400 + \tfrac{h}{12}(23f_4 - 16f_3 + 5f_2) = 8.8227.
\end{aligned}$$

Výsledky jsou uvedeny v první části Tabulky 7.3. □

7.4.2 Adamsovy-Moultonovvy metody

Jedná se o skupinu rekurentních vícekrokových vzorců, které jsou implicitní. Odvodí se podobně jako vzorce pro Adamsovy-Bashforthovy metody - integruje se interpolační polynom pro derivaci řešení diferenciální rovnice, nyní na intervalu $\langle x_{i-1}, x_i \rangle$ a pak se posune indexování. Uveďme zde pouze přehled vzorců:

$$y_{i+1} = y_i + h f_{i+1}, \tag{7.16}$$

$$y_{i+1} = y_i + \frac{h}{2}(f_{i+1} + f_i), \tag{7.17}$$

$$y_{i+1} = y_i + \frac{h}{12}(5f_{i+1} + 8f_i - f_{i-1}), \tag{7.18}$$

$$y_{i+1} = y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}). \tag{7.19}$$

Například (7.18) představuje dvoukrokovou metodu třetího řádu. O tom, kolika je bodová, je předčasné hovořit, protože záleží na způsobu použití vzorce. Jak už jsme zmínili, tyto vzorce vyžadují v každém rekurentním kroku řešit rovnici pro neznámou y_{i+1}. Iterační způsob řešení si ukážeme v dalším odstavci. Zde si předvedeme jednodušší použití, kdy se do implicitního vzorce dosadí pravá strana diferenciální rovnice f a vyjádřením y_{i+1} se vzorec převede na explicitní tvar. Tento postup lze použít například, je-li pravá strana f lineární ve druhé proměnné.

Příklad 7.4.2 Počáteční úlohu (7.4) řešte pomocí dvoukrokové Adamsova-Moultonovy metody pro $h = 1$. Výsledky porovnejte s přesným řešením.

Řešení: Do pravé strany vzorce (7.18) dosadíme $f_{i+1} = f(x_{i+1}, y_{i+1}) = x_{i+1}^2 - 0.2y_{i+1}$:

$$y_{i+1} = y_i + \frac{h}{12}(5(x_{i+1}^2 - 0.2y_{i+1}) + 8f_i - f_{i-1})$$

a vyjadříme y_{i+1}:

$$y_{i+1} = 12(y_i + \frac{h}{12}(5x_{i+1}^2 + 8f_i - f_{i-1}))/(12 + h).$$

Tento přepis Adamsova-Moultonova vzorce je dvoukroková metoda třetího řádu. Pro počáteční úsek vezmeme hodnotu z metody RK4, tj. použijeme x_i a y_i z prvních dvou řádků Tabulky 7.2. Výsledky jsou uvedeny ve druhé části Tabulky 7.3. □

7.4.3 Metody prediktor-korektor

Princip odvození metod typu *prediktor-korektor* si ukážeme na dvoukrokovém Adamsově-Moultonově vzorci:

$$y_{i+1} = y_i + \frac{h}{12}(5f(x_{i+1}, y_{i+1}) + 8f_i - f_{i-1}).$$

Předpokládejme, že známe hodnoty y_i, y_{i-1}, y_{i-2} a chceme vypočítat y_{i+1}. Na uvedený vzorec můžeme nahlížet jako na rovnici v iteračním tvaru (2.16), pro jejíž vyřešení je přirozené použít metodu prostých iterací (2.18). Tato úvaha vede na následující výpočetní postup:

Ad.-Bash. 3. řádu				Ad.-Moul. 3. řádu			
i	x_i	y_i	$\lvert y_i - y(x_i)\rvert$	i	x_i	y_i	$\lvert y_i - y(x_i)\rvert$
0	-2	-1	0	0	-2	-1	0
1	-1	1.2508	0.0001	1	-1	1.2508	0.0001
2	0	1.3112	0.0001	2	0	1.2929	0.0183
3	1	1.5588	0.1679	3	1	1.3613	0.0296
4	2	3.5400	0.2411	4	2	3.2628	0.0362
5	3	8.8227	0.3060	5	3	8.4773	0.0394

Tabulka 7.3: Adamsova-Bashforthova a Adamsova-Moultonova metoda.

Pro $i = 1, \ldots, n-1$ *vypočti* y_{i+1} *takto:*

(a) urči počáteční odhad y_{i+1}^0,

(b) počáteční odhad zpřesňuj pomocí iterací:

$$y_{i+1}^{k+1} = y_i + \tfrac{h}{12}(5f(x_{i+1}, y_{i+1}^k) + 8f_i - f_{i-1}), \quad k = 0, 1, 2, \ldots.$$

Krok *(a)* je *prediktor*, krok *(b)* je *korektor*. Metoda prediktor-korektor uvedená níže používá explicitní Adamsův-Bashforthův vzorec třetího řádu (7.14) jako prediktor. V korektoru se počítá jen jedno iterační zpřesnění. Pro stručný zápis použijeme symbol přiřazení „:=“ pro změnu hodnoty proměnné, což umožní vynechat iterační index k. Dostáváme následující algoritmus:

$$\left.\begin{aligned} &y_0 = c;\\ &y_1, y_2 \text{ vypočti pomocí jednokrokové metody;}\\ &\text{Pro } i = 2, \ldots, n-1 \text{ vypočti } y_{i+1} \text{ takto:}\\ &\quad (P)\ \ y_{i+1} := y_i + \tfrac{h}{12}(23f_i - 16f_{i-1} + 5f_{i-2}),\\ &\quad (E)\ \ f_{i+1} := f(x_{i+1}, y_{i+1}),\\ &\quad (C)\ \ y_{i+1} := y_i + \tfrac{h}{12}(5f_{i+1} + 8f_i - f_{i-1}),\\ &\quad (E)\ \ f_{i+1} := f(x_{i+1}, y_{i+1}). \end{aligned}\right\} \tag{7.20}$$

Uvedený algoritmus je tříkroková, dvoubodová metoda třetího řádu označovaná jako PECE. Podobně se používá označení PE(CE)^k, P(EC)^k nebo P(EC)^{k}E, pro varianty algoritmů prediktor-korektor s k vnitřními kroky a s různou organizací doprovodných výpočtů.

Příklad 7.4.3 Počáteční úlohu (7.4) řešte pomocí PECE metody (7.20) s krokem $h = 1$. Výsledky porovnejte s přesným řešením.

Řešení: Počáteční úsek vypočítáme metodou RK4. Jsou to první tři hodnoty x_i a y_i z druhé části Tabulky 7.2, tj. $x_0 = -2$, $x_1 = -1$, $x_2 = 0$ a $y_0 = -1$, $y_1 = 1.2508$, $y_2 = 1.3112$. Pomocí těchto hodnot vypočítáme $f_0 = 4.2000$, $f_1 = 0.7498$ a $f_2 = -0.2622$. U dalších výpočtů

podle (7.20) uvádíme pouze pořadí v jakém vznikají jednotlivé hodnoty:

$$y_3 := 1.5588,\ f3 := 0.6882,\ y_3 := 1.3607,\ f3 := 0.7279,$$
$$y_4 := 3.4178,\ f4 := 3.3164,\ y_4 := 3.2496,\ f4 := 3.3501,$$
$$y_5 := 8.5908,\ f5 := 7.2818,\ y_5 := 8.4564,\ f5 := 7.3087.$$

Výsledky jsou uvedeny v Tabulce 7.4. □

Poznámka

Porovnáním výsledků v Tabulkách 7.3 a 7.4 je vidět, že implicitní metoda je podstatně přesnější než explicitní metoda stejného řádu. Algoritmus prediktor-korektor představuje „nepřesnou“ implicitní metodu, takže přesnost je o něco menší.

i	x_i	y_i	$\lvert y_i - y(x_i)\rvert$
0	-2	-1	0.0000
1	-1	1.2508	0.0001
2	0	1.3112	0.0001
3	1	1.3607	0.0302
4	2	3.2496	0.0493
5	3	8.4564	0.0603

Tabulka 7.4: Metoda PECE založená Adamsových-Bashforthových a Adamsových-Moultonových vzorcích třetího řádu.

Kontrolní otázky

Otázka 1. V čem spočívá hlavní přínos vícekrokových metod oproti metodám jednokrokovým?
Otázka 2. Jak se odvozují Adamsovy-Bashforthovy a Adamsovy-Moultonovy vzorce?
Otázka 3. Na čem je založena konstrukce algoritmu prediktor-korektor?

Úlohy k samostatnému řešení

1. Diferenciální rovnici $y' = y(1 + \sin x) - x^3, y(1) = 0$ řešte na intervalu $\langle 1, 2\rangle$ s krokem $h = 0.2$ pomocí Adamsovy-Bashforthovy metody čtvrtého řádu.
2. Vzorec Adamsovy-Moultonovvy metody čtvrtého řádu přepište na explicitní tvar pro diferenciální rovnici z předchozí úlohy a proveďte výpočet.
3. Sestavte algoritmy prediktor korektor PECE a PECEC založené na Adamsových-Bashforthových a Adamsových-Moultonových vzorcích čtvrtého řádu a vyřešte diferenciální rovnici z první úlohy.

Výsledky úloh k samostatnému řešení

1. Počáteční úsek vypočítáme metodou RK4, pak $y_4 = -4.6497$, $y_5 = -8.5164$.
2. Odvodíme $y_{i+1} = 8(y_i + \frac{h}{24}(-9x_{i+1}^3 + 19f_i - 5f_{i-1} + f_{i-2}))/(8 - 3h(1 + \sin x_{i+1}))$. Pomocí y_0, y_1, y_2 z RK4 vypočítáme: $y_3 = -2.3270$, $y_4 = -4.6615$, $y_5 = -8.5396$.
3. Algoritmus PECEC má následující podobu:

$$
\left.
\begin{array}{l}
y_0 = c; \\
y_1, y_2, y_3 \textit{ vypočti pomocí jednokrokové metody (RK4)}; \\
\textit{Pro } i = 3, \ldots, n-1 \textit{ vypočti } y_{i+1} \textit{ takto:} \\
\quad (P)\ \ y_{i+1} := y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}), \\
\quad (E)\ \ f_{i+1} := f(x_{i+1}, y_{i+1}), \\
\quad (C)\ \ y_{i+1} := y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}), \\
\quad (E)\ \ f_{i+1} := f(x_{i+1}, y_{i+1}), \\
\quad (C)\ \ y_{i+1} := y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}).
\end{array}
\right\}
$$

Algoritmus PECE dostaneme vynecháním posledního kroku (C). Počáteční úsek vypočítáme opět metodou RK4. Výsledky podle PECE: $y_4 = -4.6581$, $y_5 = -8.5342$; výsledky podle PECEC: $y_4 = -4.6594$, $y_5 = -8.5360$.

LITERATURA

[1] Čermák, L., Hlavička, R.: *Numerické metody I.* Vysoké učení technické, Brno, 2015.

[2] Kubíček, M., Dubcová, M., Janovská, D.: *Numerické metody a algoritmy.* Vysoká škola chemicko-technologická, Praha, 2008.

[3] Míka, S., Brandner, M.: *Numerické metody I.* Západočeská univerzita, Plzeň, 2000.

[4] Přikryl, P., Brandner, M.: *Numerické metody II.* Západočeská univerzita, Plzeň, 2000.

[5] Sülli, E., Mayers, D.: *An introduction to numerical analysis.* Cambridge University Press, Cambridge, 2003. (anglicky)

[6] Van Loan, Ch., F.: *Introduction to Scientific Computing.* Prentice-Hall, New Jersey, 2000. (anglicky)

[7] Vitásek, E.: *Numerické metody.* SNTL - Nakladatelství technické literatury, Praha, 1987.

[8] Vondrák, V., Pospíšil, L.: *Numerické metody I.* Vysoká škola báňská-Technická univerzita, Ostrava, 2011.

[9] Quateroni, A., Sacco, R., Saleri, F.: *Numerical Mathematics (Texts in Applied Mathematics).* Springer, Berlin, 2007. (anglicky)