FAKULTA STROJNÍHO INŽENÝRSTVÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Numerické metody

Garant předmětu:
doc. RNDr. Libor Čermák, CSc.

Autoři textu:
Mgr. Irena Růžičková
RNDr. Rudolf Hlavička, CSc.

Obsah

I NUMERICKÉ METODY 7

1 Chyby při numerických výpočtech 7
1.1 Zdroje a typy chyb . . . 7
1.2 Definice chyb . . . 7
1.3 Zaokrouhlování. Šíření chyb při výpočtu . . . 8
1.4 Podmíněnost numerických úloh a numerická stabilita algoritmů . . . 10
1.5 Otázky a příklady ke cvičení . . . 11

2 Exkurze do funkcionální analýzy 12
2.1 Metrický prostor . . . 12
2.2 Úplný metrický prostor . . . 13
2.3 Pevný bod zobrazení, iterační proces . . . 14
2.4 Normovaný vektorový prostor . . . 16
2.5 Otázky a příklady ke cvičení . . . 19

3 Numerické řešení soustavy lineárních rovnic 20
3.1 Přímé metody . . . 20
3.1.1 Cramerovo pravidlo . . . 21
3.1.2 Gaussova eliminační metoda . . . 21
3.1.3 Eliminace s výběrem hlavního prvku . . . 24
3.2 Iterační metody . . . 25
3.2.1 Jacobiho metoda . . . 25
3.2.2 Gauss-Seidelova metoda . . . 30
3.3 Otázky a příklady ke cvičení . . . 33

4 Numerické metody řešení nelineárních rovnic 35
4.1 Numerické metody řešení jedné nelineární rovnice . . . 35
4.1.1 Metoda půlení intervalu . . . 36
4.1.2 Metoda regula falsi . . . 39
4.1.3 Metoda sečen . . . 40
4.1.4 Newtonova metoda (metoda tečen) . . . 41
4.1.5 Metoda prosté iterace . . . 44
4.2 Numerické metody řešení soustav nelineárních rovnic . . . 47
4.2.1 Metoda prosté iterace . . . 48
4.2.2 Newtonova metoda . . . 50
4.3 Otázky a příklady ke cvičení . . . 54

5 Aproximace funkcí 57
5.1 Interpolace algebraickými polynomy . . . 57
5.1.1 Existence a jednoznačnost interpolačního polynomu . . . 57

5.1.2 Konstrukce interpolačního polynomu, Lagrangeův interpolační polynom . . . 58
5.1.3 Newtonův interpolační polynom . . . 59
5.1.4 Odhad chyby . . . 64
5.2 Interpolace pomocí splajnů . . . 65
5.3 Metoda nejmenších čtverců . . . 69
5.4 Otázky a příklady ke cvičení . . . 74

6 Numerické derivování a integrování 77
6.1 Numerické derivování . . . 77
6.1.1 Některé často používané vzorce pro numerické derivování . . . 77
6.2 Numerické integrování . . . 79
6.2.1 Newton-Cotesovy vzorce . . . 80
6.2.2 Složené kvadraturní vzorce . . . 82
6.3 Otázky a příklady ke cvičení . . . 86

7 Nepodmíněná optimalizace 89
7.1 Jednorozměrné algoritmy . . . 89
7.1.1 Metody intervalové redukce . . . 91
7.1.2 Metody interpolace . . . 98
7.2 Vícerozměrné algoritmy . . . 102
7.2.1 Simplexová metoda [Nelder, Mead 1965] . . . 102
7.2.2 Metody redukce dimenze . . . 103
7.2.3 Newtonova metoda . . . 106

8 Numerické řešení diferenciálních rovnic 107
8.1 Počáteční úlohy . . . 108
8.1.1 Eulerova metoda . . . 108
8.1.2 Typy a vlastnosti metod pro řešení počátečních úloh, lokální a globální chyba . . . 109
8.1.3 Modifikace Eulerovy metody . . . 112
8.1.4 Rungovy-Kuttovy metody . . . 113
8.1.5 Odhad chyby. Metoda poloviční kroku . . . 114
8.1.6 Vícekrokové metody . . . 116
8.1.7 Vícekrokové metody založené na numerické integraci . . . 118
8.1.8 Metody prediktor-korektor . . . 119
8.1.9 Řešení soustav diferenciálních rovnic . . . 120
8.1.10 Řešení diferenciálních rovnic vyššího řádu . . . 122
8.2 Okrajové úlohy . . . 123
8.2.1 Metoda konečných diferencí . . . 123
8.2.2 Metoda střelby . . . 129
8.3 Otázky a příklady ke cvičení . . . 131

9 Výsledky **133**
9.1 Výsledky cvičení 1.5 133
9.2 Výsledky cvičení 2.5 133
9.3 Výsledky cvičení 3.3 133
9.4 Výsledky cvičení 4.3 133
9.5 Výsledky cvičení 5.4 134
9.6 Výsledky cvičení 6.3 135
9.7 Výsledky cvičení 8.3 136

Seznam obrázků

2.1 Příklad metriky . . . 13
2.2 Příklad metriky . . . 13
2.3 Pevné body reálné funkce . . . 15
2.4 Funkce, která je kontraktivní . . . 15
2.5 Funkce, která není kontraktivní . . . 15
4.1 Podmínka $f(a) \cdot f(b) < 0$ zaručí existenci kořene. . . . 36
4.2 K příkladu 4.1 - separace kořenů rovnice . . . 37
4.3 Metoda půlení intervalu . . . 37
4.4 Metoda regula falsi . . . 39
4.5 Metoda sečen . . . 40
4.6 Metoda sečen může divergovat. . . . 40
4.7 Newtonova metoda . . . 41
4.8 Newtonova metoda může divergovat . . . 41
4.9 Fraktál vzniklý řešením rovnice $z^4 = 1$ . . . 44
4.10 Fraktál vzniklý řešením rovnice $z - e^z = 0$ . . . 44
4.11 Metoda prosté iterace . . . 45
4.12 Metoda prosté iterace může divergovat . . . 45
4.13 Grafický význam řešení dvou nelineárních rovnic . . . 48
4.14 K příkladu 4.8 - odhad polohy kořenů. . . . 52
5.1 Funkce a interpolační polynom . . . 58
5.2 K příkladu 5.1: Zadané body a výsledný interpolační polynom . . . 59
5.3 K příkladu 5.2: Srovnání funkce a interpolačního polynomu . . . 61
5.4 Nevhodná aproximace interpolačním polynomem . . . 65
5.5 Nahrazení funkce lineárním splajnem . . . 66
5.6 Přirozený kubický splajn . . . 67
5.7 Nahrazení funkce $f(x) = \frac{1}{1+x^2}$ přirozeným kubickým splajnem. . . . 68
5.8 Metoda nejmenších čtverců . . . 70
5.9 K příkladu 5.6: zadané body a nalezená přímka . . . 73
6.1 Ilustrace ke vzorci 6.2 . . . 78
6.2 Ilustrace ke vzorci 6.4 . . . 78
6.3 Připomenutí významu určitého integrálu . . . 79
6.4 Obdélníková metoda . . . 80
6.5 Lichoběžníková metoda . . . 80
6.6 Simpsonova metoda . . . 81
6.7 Složené lichoběžníkové pravidlo . . . 82
7.1 Hledání intervalu, kde f(x) je unimodální . . . 90
7.2 Redukce intervalu . . . 92
7.3 Metoda parabolické interpolace . . . 101
7.4 Funkce $f(x) = sin(xy) + cos(x + y)$, kterou minimalizujeme . . . 102
7.5 Reflexe . . . 103
7.6 Kontrakce . . . 103
7.7 Simplexová metoda pro funkci 7.13 . . . 104

7.8 Metoda největšího spádu pro funkci 7.13 105
8.1 Přesné a přibližné řešení diferenciální rovnice 107
8.2 Směrové pole 110
8.3 Přibližné řešení diferenciální rovnice Eulerovou metodou 110
8.4 Globální a lokální chyba 111
8.5 První modifikace Eulerovy metody 113
8.6 Druhá modifikace Eulerovy metody 113
8.7 Počáteční úloha 123
8.8 Okrajová úloha 123
8.9 K příkladu 8.10 - nalezené přibližné řešení. 127

Úvod

V praxi má velký význam matematické modelování a simulace nejrůznějších procesů. Při tom je potřeba řešit různé matematické úlohy, mnoho dějů je např. popsáno diferenciálními rovnicemi. Nalezení přesného řešení takovýchto problémů bývá často náročné, někdy i úplně nemožné. Často je lepší nehledat řešení v uzavřeném tvaru, ale pomocí konečného počtu kroků určitého postupu najít řešení přibližné. K tomu právě slouží numerické metody.
I hledání přibližného řešení bývá ovšem dosti pracné a jen málo úloh lze s uspokojivou přesností vyřešit „ručně". Proto jsou numerické metody těsně spjaty s programováním a rozkvět některých oblastí numerických metod přišel teprve s rozvojem výpočetní techniky.

V první části těchto skript se studenti mohou seznámit se základními a nejjednoduššími numerickými metodami pro řešení lineárních a nelineárních rovnic, aproximaci funkcí, numerické derivování a integrování a pro řešení diferenciálních rovnic.
Snažila jsem se o srozumitelnost a současně o zachování matematické přesnosti. Zkušenější čtenáři mi snad prominou jistou nepořádnost v uvádění předpokladů.

Poznámka k řešeným příkladům
Všechny mezivýsledky v příkladech řešených v těchto skriptech jsou zapisovány po zaokrouhlení. Při dalším výpočtu však byly použity původní, přesnější hodnoty. Proto se může stát, že bude-li někdo tyto příklady přepočítávat a použije k tomu mezivýsledky zde uvedené, může dojít k výsledkům poněkud odlišným.

Část I
NUMERICKÉ METODY

1 Chyby při numerických výpočtech

Cíl kapitoly

Protože základem numerických metod je získávání přibližných výsledků, je nutné mít vždy představu, jaký rozdíl může být mezi přesným řešením dané úlohy a řešením získaným použitou numerickou metodou.
Cílem této kapitoly je ukázat, kde všude se při převodu nějakého problému z praxe na úlohu numerickou dopouštíme nepřesností. Dále se seznámíme s veličinami, které používáme při hodnocení získaného přibližného výsledku - absolutní a relativní chybou - a s tím, co se děje při používání zaokrouhlených čísel během výpočtu.

1.1 Zdroje a typy chyb

Pomineme-li jako zdroj chyb člověka dopouštějícího se omylů, můžeme chyby rozdělit na několik základních druhů:

- **chyby matematického modelu** – vznikají nahrazením reálné fyzikální situace matematickým modelem. Může se jednat například o popis nějakého fyzikálního děje pomocí diferenciální rovnice.

- **chyby vstupních dat** – jsou způsobeny nepřesnostmi při měření fyzikálních veličin.

- **chyby numerické metody** – vznikají při náhradě původní matematické úlohy jednodušší úlohou numerickou. Často se jedná o náhradu nekonečného procesu procesem konečným, např. při výpočtu hodnoty některé elementární funkce pomocí součtu několika prvních členů její nekonečné Taylorovy řady nebo při aproximaci určitého integrálu součtem konečného počtu funkčních hodnot. Odhad této chyby je důležitou součástí řešení každé numerické úlohy.

- **chyby zaokrouhlovací** – vznikají tím, že při výpočtech pracujeme s čísly zaokrouhlenými na určitý, relativně nevelký, počet míst. Tyto chyby se při výpočtu mohou kumulovat, nebo naopak navzájem rušit. Při velkém počtu operací je posouzení jejich vlivu velmi náročné.

1.2 Definice chyb

Je-li $\hat{x}$ přesná hodnota nějakého čísla a x její aproximace, jejich rozdíl

$$E(x) = \hat{x} - x$$

nazýváme **absolutní chyba** aproximace. Obvykle se budeme zabývat odhadem této chyby, ale je-li přesná hodnota veličiny velmi malá nebo velmi velká, má větší význam užívat **relativní chybu**

$$RE(x) = \frac{\hat{x} - x}{x},$$

která se též často vyjadřuje v procentech.
Například absolutní chyba 10^6 se může na první pohled zdát velmi velká. Je-li ovšem přesná hodnota veličiny 10^{15}, už se chyba tak závažná nejeví. Tento fakt lze nejlépe vyjádřit pomocí relativní chyby, v tomto případě je $RE = 10^{-9} = 10^{-7}\,\%$.

Přesnou hodnotu chyb zpravidla neznáme. Proto jsou důležité odhady chyb.

Každé nezáporné číslo $ME(x)$, pro které platí

$$|\,\hat{x} - x| \leq ME(x)\ \text{, tj. } \hat{x} \in \langle x - ME(x), x + ME(x)\rangle$$

nazýváme **odhad absolutní chyby** aproximace x nebo **mezní absolutní chyba**.
Každé nezáporné číslo $MR(x)$, pro které platí

$$\frac{|\hat{x} - x|}{|x|} \leq MR(x),\ x \neq 0$$

nazýváme **odhad relativní chyby** nebo **mezní relativní chyba**.

Často užíváme symbolických zápisů

$$\hat{x} = x \pm ME(x),\ \text{resp.}\ \ \hat{x} = x(1 \pm MR(x)).$$

1.3 Zaokrouhlování. Šíření chyb při výpočtu

Je-li x reálné číslo, které má obecně nekonečné dekadické vyjádření, pak číslo $x^{(d)}$, které má d desetinných míst, je **správně zaokrouhlenou hodnotou** čísla x, platí-li

$$|\,x - x^{(d)}| \leq \frac{1}{2}\,10^{-d} \tag{1.1}$$

Tedy např. má-li být $x^{(1)}$ správně zaokrouhlená hodnota čísla x na jedno desetinné místo, nesmí se od x lišit o více než o $\frac{1}{2}\,10^{-1} = 0,05$.
Jestliže číslo x, které chceme zaokrouhlit na d desetinných míst, má právě $d+1$ desetinných míst, z nichž poslední je pětka, často se používá pravidlo (čtenáři snad známé ze základní školy), že pětka po liché číslici se zaokrouhluje nahoru, po sudé dolů. Lze ale také (a některé počítačové programy tak činí) volit vždy zaokrouhlení nahoru nebo vždy zaokrouhlení dolů.

Při numerických výpočtech pracujeme se zaokrouhlenými čísly. Výsledky početních operací s těmito čísly jsou opět zaokrouhlovány a dále se s nimi pracuje. Tím se zaokrouhlovací chyby šíří. Budeme se nyní zabývat tím, co se děje při základních aritmetických operacích.
Nechť x a y jsou aproximace čísel $\hat{x}$ a $\hat{y}$.

Pro **chybu součtu a rozdílu** platí

$$\begin{aligned} |\,E(x \pm y)| &= |\,(\hat{x} \pm \hat{y}) - (x \pm y)| = |\,(\hat{x} - x) \pm (\hat{y} - y)| = \\ &= |\,E(x) \pm E(y)| \leq |E(x)| + |E(y)| \leq ME(x) + ME(y) \end{aligned} \tag{1.2}$$

Odhad chyby součinu a podílu je o něco pracnější. Pro **chybu součinu** platí

$$\begin{aligned} |\,E(x \cdot y)| &= |\,\hat{x}\hat{y} - xy| = |\,E(x) \cdot y + E(y) \cdot x + E(x) \cdot E(y)| \leq \\ &\leq |\,y| \cdot ME(x) + |\,x| \cdot ME(y) + ME(x) \cdot ME(y) \end{aligned} \tag{1.3}$$

Protože součin $ME(x) \cdot ME(y)$ bývá vzhledem k ostatním sčítancům zanedbatelný, dostáváme pro **relativní chybu součinu**

$$|\,RE(xy)| \approx \left| \frac{E(x) \cdot y + E(y) \cdot x}{xy} \right| \leq MR(x) + MR(y) \tag{1.4}$$

Podobně pro **chybu podílu** platí

$$\left| E(\frac{x}{y}) \right| = \left| \frac{x + E(x)}{y + E(y)} - \frac{x}{y} \right| = \left| \frac{E(x) \cdot y - x \cdot E(y)}{y(y + E(y))} \right| \leq \frac{|\,y|ME(x) + |\,x|ME(y)}{|\,y|(|\,y| - ME(y))} \tag{1.5}$$

a je-li $ME(y)$ zanedbatelné vzhledem k y, pak pro **relativní chybu podílu** dostaneme

$$\left| R(\frac{x}{y}) \right| \leq MR(x) + MR(y)$$

Nyní se ještě zmíníme obecně o **chybě při výpočtu funkční hodnoty**. Máme stanovit, jaké chyby se dopustíme při výpočtu hodnoty funkce $f(x_1, x_2, \ldots, x_n)$ v bodě $[\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n]$, jestliže přesné hodnoty $\hat{x}_i$ nahradíme přibližnými hodnotami x_i. Chybu i-té proměnné označíme E_i. Platí

$$f(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) = f(x_1, x_2, \ldots, x_n) + \sum_{i=1}^{n} E_i \frac{\partial f}{\partial x_i} + \frac{1}{2} \Big(\sum_{i=1}^{n} E_i \frac{\partial}{\partial x_i} \Big)^2 f + \cdots$$

kde parciální derivace se berou v bodě $[x_1, x_2, \ldots, x_n]$. Protože obvykle budeme moci předpokládat, že členy obsahující součiny chyb jsou malé ve srovnání s ostatními členy na pravé straně, můžeme psát

$$f(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n) - f(x_1, x_2, \ldots, x_n) \approx \sum_{i=1}^{n} E_i \frac{\partial f}{\partial x_i} \tag{1.6}$$

Všimněme si, že 1.2, 1.3 a 1.5 jsou speciálními případy tohoto vzorce.

Zde je na místě zmínit se o tom, že při odečítání dvou sobě blízkých čísel se může velmi zvětšit relativní chyba. Pokud pak takto získaný výsledek použijeme dále jako dělitele, může dojít i k podstatnému zvětšení absolutní chyby. Tento jev ukážeme na příkladech.

Příklad 1.1 *Nechť $x = 2,78493$ a $y = 2,78469$ jsou aproximace čísel $\hat{x}$ a $\hat{y}$ získané zaokrouhlením těchto čísel na pět desetinných míst. Určete odhady absolutní a relativní chyby rozdílu $x - y$.*

Řešení: Mezní absolutní chyby x a y jsou podle 1.1 $ME(x) = ME(y) = \frac{1}{2}10^{-5}$. Tedy podle 1.2 $|E(x-y)| \leq 10^{-5} = ME(x-y)$.
Mezní relativní chyba x je $MR(x) = \frac{\frac{1}{2}10^{-5}}{2{,}78493} \doteq 1,8 \cdot 10^{-6}$ ($MR(y)$ vyjde skoro stejně), zatímco pro rozdíl může být relativní chyba řádově vyšší, její odhad je roven $\frac{ME(x-y)}{x-y} = \frac{10^{-5}}{0{,}00024} \doteq 4,2 \cdot 10^{-2}$.

Příklad 1.2 *Nechť $z = 1,23456$ je aproximace čísla $\hat{z}$ získaná zaokrouhlením tohoto čísla na pět desetinných míst. Určete odhad chyby podílu $\frac{z}{x-y}$, kde x a y jsou čísla z příkladu 1.1*

Řešení: Z příkladu 1.1 známe odhad chyby jmenovatele. Dále víme, že $ME(z) = \frac{1}{2}10^{-5}$. Pro odhad chyby podílu stačí dosadit do 1.5:

$$\begin{aligned} \left| E\left(\frac{z}{x-y}\right)\right| &\leq \frac{|x-y| \cdot ME(z) + |z| \cdot ME(x-y)}{|x-y|(|x-y| - ME(x-y))} = \\ &= \frac{0,00024 \cdot \frac{1}{2} \cdot 10^{-5} + 1,23456 \cdot 10^{-5}}{0,00024 \cdot (0,00024 - 10^{-5})} \doteq 2,2 \cdot 10^{2} \end{aligned}$$

Tedy, zatímco vstupní hodnoty x, y a z měly chybu řádově v stotisícinách, výsledek může mít chybu řádově ve stovkách!
Proto, je-li to možné, je žádoucí se odečítání blízkých čísel vyvarovat.

1.4 Podmíněnost numerických úloh a numerická stabilita algoritmů

Při numerickém řešení různých úloh musíme zkoumat, jaký vliv na výsledek mají malé změny ve vstupních hodnotách a zaokrouhlování během výpočtu.

Řešení numerických úloh můžeme považovat za postup, kterým přiřazujeme vstupním údajům výstupní data. Je-li toto přiřazení spojité zobrazení, pak říkáme, že numerická úloha je **korektní úloha**, v opačném případě se jedná o úlohu nekorektní.
Pro tyto úlohy má zásadní význam relativní citlivost výsledku na malé změny ve vstupních parametrech úlohy. Korektní **úloha je dobře podmíněná**, jestliže malým relativním změnám vstupních údajů odpovídají malé relativní změny výstupních údajů. Číslo

$$C_p = \frac{\text{relativní chyba výstupních údajů}}{\text{relativní chyba vstupních údajů}}$$

nazýváme **číslo podmíněnosti úlohy**. Pro dobře podmíněné úlohy je číslo C_p blízké číslu 1.
Pokud malé relativní změny na vstupu způsobí velké relativní změny na výstupu, pak mluvíme o **špatně podmíněné úloze**. Řešení špatně podmíněných úloh je nejlépe se vyhnout, protože výsledky jakéhokoli algoritmu jsou velmi nespolehlivé.

Podobně řekneme, že je **algoritmus dobře podmíněný**, je-li málo citlivý na poruchy ve vstupních datech. Kromě nepřesností ve vstupních údajích ovlivňuje výsledek použitého algoritmu i zaokrouhlování čísel během výpočtu. Je-li vliv zaokrouhlovacích chyb na výsledek malý, mluvíme o **numericky stabilním algoritmu**. Algoritmus dobře podmíněný a numericky stabilní se nazývá **stabilní**.

Shrnutí pojmů

Při sestavování numerické úlohy se dopouštíme chyby už tím, že reálnou situaci nahradíme zjednodušeným matematickým modelem. Další chyby mohou vzniknout kvůli nepřesnosti vstupních dat. Podstatným zdrojem chyb je nahrazení původní matematické úlohy úlohou numerickou (konkrétní příklady těchto nahrazení uvidíme v dalších kapitolách). A konečně, nemalý vliv mohou mít chyby, které vzniknou zaokrouhlováním čísel během výpočtu.
Kvalitu výsledku získaného nějakou numerickou metodou můžeme popsat pomocí absolutní chyby - rozdílu přesné a přibližné hodnoty. Někdy je výstižnější relativní chyba - podíl absolutní chyby a vypočtené přibližné hodnoty. Protože však přesnou hodnotu často neznáme a tím pádem absolutní chybu nejsme schopni určit, důležité jsou odhady absolutní a relativní chyby.
Používáme-li při výpočtu zaokrouhlená čísla, chyby se šíří. Zvlášt nebezpečné je odečítání sobě blízkých čísel.

1.5 Otázky a příklady ke cvičení

U následujících výroků rozhodněte, zda se jedná o výrok pravdivý či nepravdivý.

Otázka 1.1 *Relativní chyba závisí na velikosti absolutní chyby.*

Otázka 1.2 *Je-li* $x > 10^6$*, bude relativní chyba* $RE(x)$ *určitě velmi malá.*

Otázka 1.3 *Je-li absolutní chyba* $E(x) < 10^{-6}$*, je určitě i relativní chyba* $RE(x) < 10^{-6}$.

Otázka 1.4 *Jestliže* x *aproximuje přesnou hodnotu* $\hat{x}$ *s chybou* $E(x) = 0,01$*, pak* $y = 2x$ *aproximuje* $\hat{y} = 2\hat{x}$ *s chybou* $E(y) = 0,02$.

Otázka 1.5 *Pokud jsme čísla* x *a* y *získali zaokrouhlením čísel* $\hat{x}$ *a* $\hat{y}$ *na* n *desetinných míst, pak na* n *desetinných míst zaokrouhlená hodnota čísla* $\hat{x} + \hat{y}$ *je rovna* $x + y$*. (*$\hat{x}$ *a* $\hat{y}$ *mohou být libovolná reálná čísla.)*

Otázka 1.6 *Čím větší je relativní chyba výstupních údajů dané úlohy, tím větší je číslo podmíněnosti této úlohy.*

Odpovědi na otázky viz 9.1

2 Exkurze do funkcionální analýzy

Cíl kapitoly

Tato kapitola tvoří teoretický základ pro metody probírané v dalších dvou kapitolách. Protože prostor, který lze této problematice věnovat, je velmi omezený, pokusíme se zde vysvětlit jen nejnutnější pojmy. Pokud by někoho odrazovala přílišná teoretičnost a „vědeckost" této kapitoly a spokojil by se s tím, **že** metody popsané v kapitolách 3 a 4 fungují, aniž by se zajímal o to, **proč** fungují, mohl by snad následující text přeskočit.

2.1 Metrický prostor

Studenti určitě umí vypočítat vzdálenost dvou reálných čísel na číselné ose nebo vzdálenost dvou bodů v rovině či v prostoru. Podobně se dá určovat „vzdálenost" různých jiných objektů. Této zobecněné vzdálenosti se říká metrika.

Definice. Buď X množina (prvků jakéhokoli typu). Řekneme, že na této množině je definována **metrika** d, jestliže každým dvěma prvkům $x, y \in X$ je přiřazeno reálné číslo $d(x, y)$ tak, že

1) $d(x, y) \geq 0 \quad \forall x, y \in X \quad , \quad d(x, y) = 0 \Leftrightarrow x = y$

2) $d(x, y) = d(y, x) \quad \forall x, y \in X$

3) $d(x, z) \leq d(x, y) + d(y, z) \quad \forall x, y, z \in X \quad$ (trojúhelníková nerovnost)

Množinu X s metrikou d pak nazýváme **metrický prostor**.

Příklady metrických prostorů
Asi nejjednodušším příkladem metrického prostoru je množina všech reálných čísel $\mathbb{R}$ s metrikou d definovanou jako $d(x, y) = |x - y|$.
Jako množinu X však nemusíme brát celé $\mathbb{R}$, může to být i jakákoli jeho podmnožina, např. interval nebo množina všech racionálních čísel $\mathbb{Q}$.
Jiným příkladem je množina všech uspořádaných n-tic reálných čísel. Je-li $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ a $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, metriku d můžeme definovat různě. Jako nejpřirozenější se jeví obvyklá vzdálenost dvou bodů:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}, \tag{2.1}$$

existují však i jiné možnosti, např.

$$d(\mathbf{x}, \mathbf{y}) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n| \tag{2.2}$$

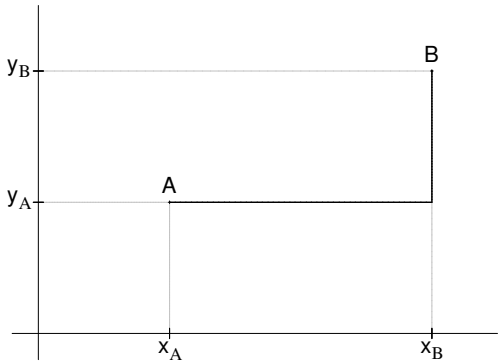
nebo

$$d(\mathbf{x}, \mathbf{y}) = \max\Big(|x_1 - y_1|, |x_2 - y_2|, \ldots, |x_n - y_n|\Big). \tag{2.3}$$

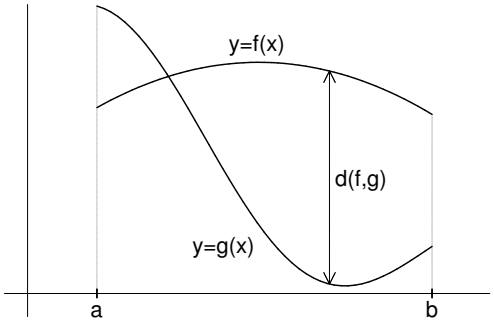
Jako poslední příklad uvedeme množinu všech funkcí definovaných a spojitých na intervalu $< a, b >$, která se označuje jako $C(< a, b >)$. Jsou-li $f, g \in C(< a, b >)$, definujeme

$$d(f, g) = \max_{x \in <a,b>} |f(x) - g(x)|. \tag{2.4}$$

Obrázky 2.1 a 2.2 poslouží k objasnění některých uvedených metrik.



Obrázek 2.1: „Vzdálenost" bodů A, B podle metriky 2.2 je délka silné černé čáry.



Obrázek 2.2: „Vzdálenost" dvou spojitých funkcí v metrice 2.4

2.2 Úplný metrický prostor

Již na střední škole se studenti seznámili s posloupnostmi reálných čísel a (snad) i s jejich limitami.
Připomeňme, že limita posloupnosti reálných čísel $(a_n)_{n=1}^{\infty}$ je, populárně řečeno, takové číslo a, ke kterému se členy posloupnosti pro n jdoucí do nekonečna přibližují. Přesněji:
Reálné číslo a se nazývá limitou posloupnosti $(a_n)_{n=1}^{\infty}$, jestliže ke každému $\varepsilon > 0$ existuje přirozené číslo N tak, že pro všechna $n > N$ platí $|a_n - a| < \varepsilon$.
Neboli: ať zvolíme ε libovolně malé, od jistého indexu N se členy posloupnosti budou od a lišit méně než o ε.
Posloupnosti však můžeme sestavovat i z jiných objektů než z reálných čísel. Stejně tak můžeme u takových posloupností říci, zda mají, nebo nemají limitu. Pro posloupnosti sestavené z prvků obecného metrického prostoru se limita definuje velmi podobně, jen je třeba zobecnit ono „lišení se o méně než ε". To se provede pomocí metriky.

Definice. Buď X metrický prostor s metrikou d a $(x_n)_{n=1}^{\infty}$ posloupnost prvků z X. Řekneme, že $x \in X$ je **limitou** této posloupnosti, píšeme $\lim_{n\to\infty} x_n = x$, jestliže ke každému $\varepsilon > 0$ existuje přirozené číslo N tak, že pro všechna $n > N$ platí $d(x_n, x) < \varepsilon$.

Posloupnost, která má limitu, se nazývá **konvergentní**.

Nyní definujeme další vlastnost posloupností.

Definice. Buď X metrický prostor s metrikou d a $(x_n)_{n=1}^{\infty}$ posloupnost prvků z X. Řekneme, že tato posloupnost je **cauchyovská**, jestliže ke každému $\varepsilon > 0$ existuje přirozené číslo N tak, že pro všechna $n > N$ a každé přirozené číslo k platí $d(x_n, x_{n+k}) < \varepsilon$.

Dá se říci, že cauchyovská posloupnost je taková, jejíž členy se výše popsaným způsobem zahušťují.
Dá se dokázat, že každá konvergentní posloupnost je cauchyovská. Intuitivně by se mohlo zdát, že to musí být i naopak. Existují ale prostory, v nichž najdeme cauchyovské posloup-

nosti, které v daném prostoru limitu nemají. Ukážeme to na následujícím příkladu:
Mějme například množinu všech reálných čísel a v něm posloupnost $a_1 = 3.1, a_2 = 3.14, a_3 = 3.141, a_4 = 3.1415, \ldots$. Tato posloupnost má limitu π a tedy je cauchyovská.
Nyní vezměme tutéž posloupnost, ale v množině všech racionálních čísel $\mathbb{Q}$. Je to posloupnost cauchyovská, ale limitu v $\mathbb{Q}$ nemá (protože $\pi \notin \mathbb{Q}$).

Existují tedy prostory, v nichž „něco schází", neobsahují limity některých posloupností, které se jinak chovají tak, jako by limitu mít měly. Tím se dostáváme k definici úplného prostoru.

Definice. Metrický prostor se nazývá **úplný**, jestliže každá cauchyovská posloupnost v něm má limitu.

Příklady úplných a neúplných prostorů
Množina $\mathbb{R}$ s metrikou $d(x, y) = |x - y|$ je úplný metrický prostor.
Jakýkoli uzavřený interval $\langle a, b\rangle$ s toutéž metrikou je také úplný prostor.
Otevřený interval s toutéž metrikou není úplný. To můžeme ukázat na příkladu intervalu $(0, 1)$ a posloupnosti $x_n = \frac{1}{n}$. Tato posloupnost je cauchyovská a přitom v intervalu $(0, 1)$ nemá limitu $(0 \notin (0, 1))$.
Dá se dokázat, že prostor všech uspořádaných n-tic reálných čísel s kteroukoli z metrik 2.1, 2.2, 2.3 je úplný.

2.3 Pevný bod zobrazení, iterační proces

Definice. Řekneme, že F je zobrazení množiny X do množiny Y, píšeme $F : X \to Y$, jestliže každému prvku $x \in X$ je pomocí F přiřazen právě jeden prvek $y \in Y$, $y = F(x)$.

Budeme se zabývat hlavně zobrazeními množiny do sebe sama, tj. zobrazení $F : X \to X$. Takové zobrazení přiřazuje každému prvku $x \in X$ opět (obecně jiný) prvek z X. Nás bude zajímat, jestli existuje takový prvek x, který se zobrazí sám na sebe, případně jak takový prvek najít.

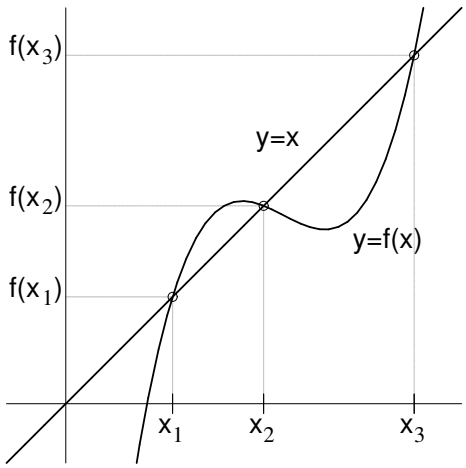
Definice. Prvek $x \in X$ se nazývá **pevný bod** zobrazení $F : X \to X$, jestliže platí

$$F(x) = x.$$

Jestliže za množinu X vezmeme $\mathbb{R}$, pak zobrazení $F : \mathbb{R} \to \mathbb{R}$ je obyčejná funkce jedné proměnné. Na obrázku 7.2 jsou vyznačeny pevné body jisté funkce f. Jsou to body, v nichž se protne graf funkce f s přímkou $y = x$.

Příklad. Funkce $f(x) = x^2$ má právě dva pevné body, a to $x = 0$ a $x = 1$, protože $0^2 = 0$ a $1^2 = 1$.

Hledání pevného bodu zobrazení má v numerické matematice velký význam. Některé úlohy, jejichž zadání zpočátku vypadá úplně jinak, lze převést právě na problém nalezení pevného bodu. Proto se nyní budeme zabývat otázkou, jak ověřit, že nějaké zobrazení pevný bod má a jak jej najít. Dá se dokázat, že jistý druh zobrazení má pevný bod vždy a existuje postup, který nás k němu dovede.



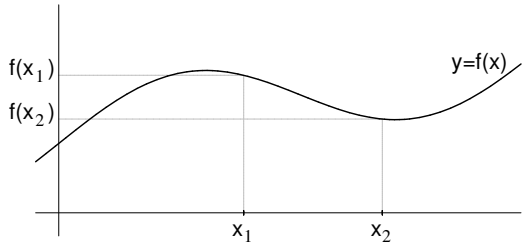
Obrázek 2.3: Pevné body reálné funkce

Definice. Buď X metrický prostor. Řekneme, že zobrazení $F : X \to X$ je **kontraktivní** (**kontrakce**), jestliže existuje $\alpha \in \langle 0, 1)$ tak, že pro každé dva prvky $x, y \in X$ platí

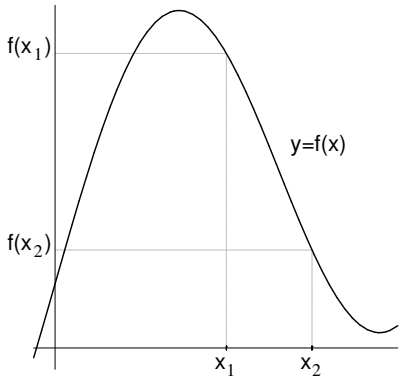
$$d(F(x), F(y)) \leq \alpha\, d(x, y) \tag{2.5}$$

Číslo α nazýváme koeficient kontrakce.

„Kontrakce" česky znamená „stažení". Dá se tedy, byť poněkud nepřesně, říct, že kontraktivní zobrazení je takové, u nějž jsou si obrazy (funkční hodnoty) bližší, než byly vzory.



Obrázek 2.4: Funkce, která je kontraktivní



Obrázek 2.5: Funkce, která není kontraktivní

Věta 2.1 *Buď X úplný metrický prostor a $F : X \to X$ kontraktivní zobrazení. Pak existuje právě jeden pevný bod tohoto zobrazení $\hat{x}$, pro nějž platí*

$$\hat{x} = \lim_{n\to\infty} x_n, \tag{2.6}$$

kde $(x_n)_{n=1}^{\infty}$ je tzv. posloupnost postupných aproximací, která je definována takto: x_0 je libovolný prvek z X a další členy posloupnosti jsou definovány předpisem

$$x_{k+1} = F(x_k), \quad k = 0, 1, \ldots \tag{2.7}$$

Dále pro všechna přirozená čísla n platí:

$$d(\hat{x}, x_n) \leq \frac{\alpha}{1-\alpha}\, d(x_n, x_{n-1}) \tag{2.8}$$

$$d(\hat{x}, x_n) \leq \frac{\alpha^n}{1-\alpha}\, d(x_0, x_1), \tag{2.9}$$

kde α je koeficient kontrakce.

Tato věta nám dává návod, **jak pevný bod** zadaného zobrazení alespoň přibližně **najít**.

Zvolíme $x_0 \in X$. Tomuto bodu se říká **počáteční aproximace**.

Pak počítáme další členy posloupnosti podle předpisu 2.7. Tomuto výpočtu se říká **iterační proces**, k-tý člen posloupnosti, x_k, se nazývá **k-tá aproximace**.

Protože podle 2.6 je pevný bod limitou posloupnosti $(x_n)_{n=1}^{\infty}$, postupné aproximace se k němu budou přibližovat. Kdybychom v iteračním procesu mohli pokračovat donekonečna, dostali bychom se nakonec k pevnému bodu. To ale není možné, a proto se v určitý moment zastavíme a řekneme, že pevný bod $\hat{x}$ je přibližně roven poslednímu vypočtenému členu posloupnosti.

Kdy iterační proces zastavit, rozhodneme podle toho, s jakou přesností chceme mít pevný bod vypočtený. Můžeme k tomu použít např. odhad 2.8, který říká, jak je n-tá aproximace nanejvýš vzdálena od pevného bodu. K tomu ovšem musíme znát hodnotu koeficientu kontrakce α, která může být u některých úloh velmi obtížně zjistitelná. Proto se častěji používají empirická kritéria, jež pro konkrétní úlohy později popíšeme.

2.4 Normovaný vektorový prostor

V prvním semestru se studenti seznámili s vektorovými prostory.

Prvky vektorových prostorů mohou být objekty nejrůznějšího typu. Nemusí to být pouze „vektory" v tom smyslu, jaký si člověk obvykle pod tímto pojmem představí (tj. uspořádané n-tice reálných čísel).

Nejjednodušším příkladem vektorového prostoru je množina všech reálných čísel $\mathbb{R}$ s obvyklými operacemi $+$ a $\cdot$.

Vektorovým prostorem je i množina všech matic typu (m, n) s operacemi $+$ (sčítání matic) a $\cdot$ (násobení matice konstantou).

Vektorový prostor může být tvořen též funkcemi jedné nebo více proměnných s určitou vlastností. V některých oblastech matematiky se často setkáváme např. s prostorem všech

funkcí spojitých na daném intervalu $\langle a, b\rangle$, či s prostorem všech funkcí na intervalu $\langle a, b\rangle$ integrovatelných.

Studenti jistě vědí, co je absolutní hodnota čísla nebo délka vektoru. Tyto veličiny udávají velikost daného čísla, resp. vektoru bez ohledu na jeho znaménko, resp. směr. „Velikost" lze různým způsobem určovat i u jiných objektů. Jakési zobecnění velikosti, které zachovává její přirozené vlastnosti, se nazývá norma.

Definice. Buď V vektorový prostor. Řekneme, že na tomto prostoru je definována **norma**, jestliže každému prvku $v \in V$ je přiřazeno reálné číslo $\|v\|$ (norma v) tak, že

1) $\|v\| \geq 0 \quad \forall v \in V \quad , \quad \|v\| = 0 \Leftrightarrow v = 0$

2) $\|k \cdot v\| = |\,k| \cdot \|v\| \quad \forall v \in V, \forall k \in \mathbb{R}$

3) $\|v_1 + v_2\| \leq \|v_1\| + \|v_2\| \quad \forall v_1, v_2 \in V \quad$ (trojúhelníková nerovnost)

Prostor V pak nazýváme **normovaný vektorový prostor**.

Je známo, že absolutní hodnota rozdílu dvou reálných čísel udává vzdálenost těchto čísel na číselné ose. Podobně si lze normu rozdílu dvou prvků vektorového prostoru $\|u - v\|$ představit jako vzdálenost těchto dvou prvků. To znamená, že na vektorovém prostoru můžeme definovat metriku předpisem

$$d(v_1, v_2) = \|\,v_1 - v_2\|. \tag{2.10}$$

Příklady normovaných vektorových prostorů:
Na množině všech reálných čísel $\mathbb{R}$ lze zavést normu jako $\|x\| = |\,x|\ ,\ \forall x \in \mathbb{R}$.
Na „obvyklém" vektorovém prostoru všech uspořádaných n-tic reálných čísel V_n můžeme zavést normu různým způsobem.
Je-li $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in V_n$, pak jeho norma může být např. definována jako délka tohoto vektoru

$$\|\,\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}\ . \tag{2.11}$$

Tato norma se často značí jako $\|\,\mathbf{v}\|_2$ a nazývá se eukleidovská norma. Existují však i jiné možnosti. V dalším textu se setkáme s normami

$$\|\,\mathbf{v}\|_1 = |\,v_1| + |\,v_2| + \cdots + |\,v_n| \tag{2.12}$$

$$\|\,\mathbf{v}\|_\infty = \max(|\,v_1|, |\,v_2|, \ldots, |\,v_n|) \tag{2.13}$$

U matic lze normu počítat podobně jako u vektorů. V kapitole 3 budeme pracovat s následujícími normami ($\mathbf{A}$ je matice typu (m, n) s prvky $a_{ij}, i = 1, \ldots, m,\ j = 1, \ldots, n$):

$$\|\,\mathbf{A}\|_\infty = \max_{i=1,\ldots,m} \sum_{j=1}^{n} |\,a_{ij}| \qquad \text{řádková norma} \tag{2.14}$$

$$\|\,\mathbf{A}\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{m} |\,a_{ij}| \qquad \text{sloupcová norma} \tag{2.15}$$

Příklad 2.1 *Vypočtěte řádkovou a sloupcovou normu matice*

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

Řešení: Řádková norma matice je maximum ze součtů absolutních hodnot prvků v jednotlivých řádcích.
Součet absolutních hodnot prvků v prvním řádku matice je $|-3|+|2|+|5| = 10$, ve druhém řádku je součet roven 7 a ve třetím 8. Největší z těchto čísel je 10 a proto $\| \mathbf{A}\|_\infty = 10$.
Sloupcová norma je maximum ze součtů absolutních hodnot prvků v jednotlivých sloupcích. Tedy $\| \mathbf{A}\|_1 = \max(7, 7, 11) = 11$.

Čtenář si možná povšiml značné podobnosti norem 2.11, 2.12 a 2.13 s metrikami uvedenými v kapitole 2.1. Skutečně, všechny tyto metriky můžeme dostat z výše uvedených norem pomocí 2.10.

Nabízí se otázka, proč jsme označili řádkovou normu matice 2.14 stejně jako normu vektoru 2.13 a sloupcovou normu matice 2.15 stejně jako normu vektoru 2.12.
Tyto normy skutečně mají mnoho společného. Představíme-li si vektor $\mathbf{v}$ dimenze n jako **sloupec**, můžeme jej považovat za matici o n řádcích a jediném sloupci. Vypočteme-li nyní řádkovou normu této matice, dostaneme právě normu vektoru 2.13, vypočteme-li sloupcovou normu matice, dostaneme normu vektoru 2.12.

Dále platí, a to je pro další úvahy podstatnější, že

$$\begin{aligned} \| \mathbf{Av}\|_\infty &\leq \| \mathbf{A}\|_\infty \cdot \| \mathbf{v}\|_\infty \\ \| \mathbf{Av}\|_1 &\leq \| \mathbf{A}\|_1 \cdot \| \mathbf{v}\|_1 \end{aligned}$$

Můžeme říct, že řádková norma matice je **přidružená** vektorové normě 2.13 a sloupcová norma matice je přidružená vektorové normě 2.12.
(Obecně se maticová norma přidružená vektorové normě definuje docela složitě, o tom zde mluvit nebudeme. Např. maticová norma přidružená eukleidovské normě vektoru se počítá zcela odlišně.)

Shrnutí pojmů

Metrický prostor je množina X, na níž je definována metrika d - funkce s jistými vlastnostmi, která každým dvěma prvkům $x, y \in X$ přiřadí číslo $d(x, y)$, které lze popsat jako „vzdálenost" x od y. V metrickém prostoru můžeme definovat limitu posloupnosti složené z jeho prvků. Má-li posloupnost limitu, řekneme, že je konvergentní. Cauchyovská posloupnost je posloupnost, jejíž prvky se určitým, v předchozím textu přesně popsaným, způsobem zahušťují. Je-li v metrickém prostoru X každá cauchyovská posloupnost konvergentní, mluvíme o prostoru úplném.
Mnoho úloh numerické matematicky se dá převést na hledání pevného bodu nějakého zobrazení. Pevný bod daného zobrazení $F : X \to X$ je takové $x \in X$, které se zobrazí samo na sebe, tj. $F(x) = x$.

Kontraktivní zobrazení je zobrazení, pro které platí $d(F(x), F(y)) \leq \alpha\, d(x, y)$, kde $\alpha \in \langle 0, 1)$. Je-li X s metrikou d úplný metrický prostor, pak každé kontraktivní zobrazení $F : X \to X$ má právě jeden pevný bod. Tento pevný bod je roven limitě posloupnosti $\{x_k\}_{k=0}^{\infty}$, kterou získáme tak, že $x_0 \in X$ zvolíme libovolně a další členy posloupnosti jsou dány vztahem $x_{k+1} = F(x_k), k = 0, 1, 2, \ldots$. Pevný bod přibližně najdeme pomocí tzv. iteračního procesu. Počítáme členy posloupnosti $\{x_k\}_{k=0}^{\infty}$, dokud podle nějakého kriteria nerozhodneme, že už jsme pevný bod s požadovanou přesností našli.
Normovaný prostor je vektorový prostor V, na němž je definována norma $\|\cdot\|$ - funkce s jistými vlastnostmi, která každému prvku $v \in V$ přiřadí číslo $\|v\|$, které lze popsat jako „velikost" v.
Na prostoru všech n-rozměrných vektorů můžeme kromě obvyklé eukleidovské normy definovat normu předpisem $\|\mathbf{v}\|_1 = |v_1| + |v_2| + \cdots + |v_n|$, resp. $\|\mathbf{v}\|_\infty = \max(|v_1|, |v_2|, \ldots, |v_n|)$.
Důležitým příkladem normovaného prostoru je prostor všech matic typu $m \times n$ s řádkovou nebo sloupcovou normou. Řádková norma matice A je maximum ze součtů absolutních hodnot prvků této matice v jednotlivých řádcích, sloupcová maximum ze součtů ve sloupcích.

2.5 Otázky a příklady ke cvičení

U následujících výroků rozhodněte, zda se jedná o výrok pravdivý či nepravdivý.

Otázka 2.1 *Může se stát, že pro dva různé prvky metrického prostoru x a y je $d(x, y) = 0$.*

Otázka 2.2 *Každá posloupnost, která má limitu, je cauchyovská.*

Otázka 2.3 *Každý metrický prostor je úplný.*

Otázka 2.4 *Pevný bod funkce $f(x) = \sin x$ je 0.*

Otázka 2.5 *Každá funkce jedné reálné proměnné má aspoň jeden pevný bod.*

Otázka 2.6 *Je-li $F : X \to X$ kontrakce a $x, y \in X$, pak $d(F(x), F(y)) < d(x, y)$.*

Otázka 2.7 *Iterační proces je postup, který slouží k nalezení pevného bodu.*

Otázka 2.8 *V praxi pomocí iteračního procesu vždy najdeme přesnou hodnotu pevného bodu.*

Otázka 2.9 *Řádková norma čtvercové matice je vždy různá od sloupcové normy.*

Odpovědi na otázky viz 9.2

3 Numerické řešení soustavy lineárních rovnic

Cíl kapitoly

Řešení soustav lineárních rovnic patří mezi nejdůležitější části numerické matematiky. Mnoho praktických úloh nakonec vede k řešení takovýchto soustav, často velmi rozsáhlých. K obrovským soustavám rovnic dospějeme např. při hledání rozložení nějaké fyzikální veličiny v určitém tělese. Problém se, velmi zhruba řečeno, může řešit tak, že hledáme hodnoty této veličiny pouze v konečném počtu bodů (a čím více těchto bodů bude, tím lépe), a to právě jako řešení soustavy lineárních rovnic.
Cílem této kapitoly je seznámit čtenáře s několika metodami používanými pro řešení těchto soustav. Zvláštní pozornost bude věnována Gaussově eliminační metodě. Také probereme dvě iterační metody - Jacobiho a Gauss-Seidelovu. Tyto dvě metody jsou z iteračních metod asi nejjednodušší. Pokud si je studenti osvojí, bude pro ně snazší pochopit jiné dnes v praxi používané iterační metody.

Budeme se zabývat řešením soustavy n lineárních rovnic

$$\begin{array}{ccccccccc} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \cdots & + & a_{1n}\,x_n & = & b_1 \\ a_{21}\,x_1 & + & a_{22}\,x_2 & + & \cdots & + & a_{2n}\,x_n & = & b_2 \\ \vdots & & & & & & & \vdots & \\ a_{n1}\,x_1 & + & a_{n2}\,x_2 & + & \cdots & + & a_{nn}\,x_n & = & b_n \end{array} \tag{3.1}$$

s neznámými $x_1, x_2, \ldots, x_n$.
Připomeňme, že matice $\mathbf{A} = (a_{ij}), i, j = 1, \ldots, n$, se nazývá **matice soustavy** a sloupcový vektor $\mathbf{b} = (b_1, \ldots, b_n)^T$ **vektor pravých stran**.
Soustavu můžeme zapsat maticově ve tvaru

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \tag{3.2}$$

Všude v dalším textu budeme předpokládat, že matice soustavy je regulární, tj. že řešená soustava má právě jedno řešení. (V technických úlohách, kde se problém řešení soustavy lineárních rovnic může vyskytnout, to tak zpravidla bývá.)

V prvním semestru se studenti seznámili s Gaussovou eliminační metodou a s Cramerovým pravidlem. Obě tyto metody patří mezi tzv. **metody přímé**. Druhou skupinou metod řešení soustav lineárních rovnic jsou **metody iterační**.

3.1 Přímé metody

Přímé metody vedou k řešení soustavy po konečném počtu kroků. Takto nalezené řešení by bylo přesné, kdybychom se v průběhu výpočtu nedopouštěli zaokrouhlovacích chyb.
Připomeneme metody, které by studenti měli znát z prvního semestru a uvedeme některé další.

3.1.1 Cramerovo pravidlo

Je-li matice soustavy 3.2 regulární, tj. její determinant je nenulový, pak řešení soustavy lze vypočítat jako

$$x_1 = \frac{D_1}{D} \quad , \quad x_2 = \frac{D_2}{D} \quad , \ \dots \ , \quad x_n = \frac{D_n}{D}$$

kde D je determinant matice soustavy $\mathbf{A}$ a $D_k, k = 1, \dots, n$ jsou determinanty matic, které vzniknou z matice $\mathbf{A}$ nahrazením k-tého sloupce této matice vektorem pravých stran $\mathbf{b}$.

Příklad 3.1 *Pomocí Cramerova pravidla najděte řešení soustavy rovnic*

$$\begin{array}{rcrcl} 2\,x_1 & + & 3\,x_2 & = & 5 \\ -x_1 & + & 2\,x_2 & = & 8 \end{array}$$

Řešení: Determinant matice soustavy je

$$D = \begin{vmatrix} 2 & 3 \\ -1 & 2 \end{vmatrix} = 7$$

a determinanty matic vzniklých nahrazením prvního, resp. druhého sloupce matice soustavy vektorem pravých stran jsou

$$D_1 = \begin{vmatrix} 5 & 3 \\ 8 & 2 \end{vmatrix} = -14, \quad D_2 = \begin{vmatrix} 2 & 5 \\ -1 & 8 \end{vmatrix} = 21.$$

Řešení soustavy je tedy

$$x_1 = \frac{-14}{7} = -2, \quad x_2 = \frac{21}{7} = 3.$$

Cramerovo pravidlo je vhodné pouze pro velmi malé soustavy rovnic, např. pro soustavu dvou rovnic s „ošklivými" koeficienty. Pro větší soustavy by bylo nutné počítat mnoho determinantů vysokého řádu, což je velmi pracné. Proto se pro řešení velkých soustav rovnic tato metoda nepoužívá.

3.1.2 Gaussova eliminační metoda

Základem této metody je úprava soustavy na trojúhelníkový tvar pomocí elementárních úprav.

Přidáme-li v soustavě 3.2 vektor pravých stran $\mathbf{b}$ jako (n+1)-ní sloupec k matici $\mathbf{A}$, můžeme soustavu přepsat ve tvaru

$$\begin{array}{rcrcccrcl} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \cdots & + & a_{1n}\,x_n & = & a_{1\,n+1} \\ a_{21}\,x_1 & + & a_{22}\,x_2 & + & \cdots & + & a_{2n}\,x_n & = & a_{2\,n+1} \\ \vdots & & & & & & & \vdots & \\ a_{n1}\,x_1 & + & a_{n2}\,x_2 & + & \cdots & + & a_{nn}\,x_n & = & a_{n\,n+1} \end{array}$$

Nyní se pomocí přičítání vhodných násobků první rovnice budeme snažit z ostatních rovnic eliminovat x_1. (Je-li $a_{11} = 0$, vyměníme první rovnici s první takovou rovnicí, která na prvním místě nulu nemá.)
Odečteme-li postupně první rovnici, vynásobenou číslem $\frac{a_{i1}}{a_{11}}$, od i-té rovnice, pro $i = 2, 3, \ldots, n$, dostaneme

$$\begin{array}{rcccccccl} a_{11}\,x_1 & + & a_{12}\,x_2 & + & \cdots & + & a_{1n}\,x_n & = & a_{1\,n+1} \\ & & a_{22}^{(1)}\,x_2 & + & \cdots & + & a_{2n}^{(1)}\,x_n & = & a_{2\,n+1}^{(1)} \\ & & \vdots & & & & \vdots & & \\ & & a_{n2}^{(1)}\,x_2 & + & \cdots & + & a_{nn}^{(1)}\,x_n & = & a_{n\,n+1}^{(1)} \end{array}$$

Nové koeficienty jsou vypočteny jako $a_{ij}^{(1)} = a_{ij} - \frac{a_{i1}}{a_{11}}\,a_{1j},\ i = 2, 3, \ldots, n, j = 2, 3, \ldots, n+1$.
Nyní budeme pomocí vhodných násobků druhé rovnice eliminovat x_2 v třetí, čtvrté, ... n-té rovnici. (Opět, je-li $a_{22}^{(1)} = 0$, vyměníme druhou rovnici s první z dalších rovnic, ve které u x_2 nula není.)
Tím dostaneme

$$\begin{array}{rcccccccccl} a_{11}\,x_1 & + & a_{12}\,x_2 & + & a_{13}\,x_3 & + & \cdots & + & a_{1n}\,x_n & = & a_{1\,n+1} \\ & & a_{22}^{(1)}\,x_2 & + & a_{23}^{(1)}\,x_3 & + & \cdots & + & a_{2n}^{(1)}\,x_n & = & a_{2\,n+1}^{(1)} \\ & & & & a_{33}^{(2)}\,x_3 & + & \cdots & + & a_{3n}^{(2)}\,x_n & = & a_{3\,n+1}^{(2)} \\ & & & & \vdots & & & & \vdots & & \\ & & & & a_{n3}^{(2)}\,x_3 & + & \cdots & + & a_{nn}^{(2)}\,x_n & = & a_{n\,n+1}^{(2)} \end{array}$$

kde $a_{ij}^{(2)} = a_{ij}^{(1)} - \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}\,a_{2j}^{(1)},\ i = 3, 4, \ldots, n, j = 3, 4, \ldots, n+1$.
Pokračujeme-li dále stejným způsobem, dostaneme po n-1 krocích soustavu v trojúhelníkovém tvaru

$$\begin{array}{rcccccccccl} a_{11}\,x_1 & + & a_{12}\,x_2 & + & a_{13}\,x_3 & + & \cdots & + & a_{1n}\,x_n & = & a_{1\,n+1} \\ & & a_{22}^{(1)}\,x_2 & + & a_{23}^{(1)}\,x_3 & + & \cdots & + & a_{2n}^{(1)}\,x_n & = & a_{2\,n+1}^{(1)} \\ & & & & a_{33}^{(2)}\,x_3 & + & \cdots & + & a_{3n}^{(2)}\,x_n & = & a_{3\,n+1}^{(2)} \\ & & & & & & & & \vdots & & \\ & & & & & & & & a_{nn}^{(n-1)}\,x_n & = & a_{n\,n+1}^{(n-1)} \end{array} \tag{3.3}$$

Z této soustavy snadno určíme hledané řešení:

$$\begin{aligned} x_n &= \frac{a_{n\,n+1}^{(n-1)}}{a_{nn}^{(n-1)}} \\ x_{n-1} &= \frac{1}{a_{n-1\,n-1}^{(n-2)}}\left(a_{n-1\,n+1}^{(n-2)} - a_{n-1\,n}^{(n-2)}\,x_n\right) \\ &\vdots \\ x_1 &= \frac{1}{a_{11}}\left(a_{1n+1} - a_{12}\,x_2 - a_{13}\,x_3 - \cdots - a_{1n}\,x_n\right) \end{aligned} \tag{3.4}$$

Postup vedoucí k soustavě 3.3 se nazývá **Gaussova eliminace**, výpočet neznámých dle 3.4 **zpětná substituce** nebo též **zpětný chod**. Číslo $a_{kk}^{(k-1)}$ nazýváme hlavní prvek.

Příklad 3.2 *Pomocí Gaussovy eliminace vyřešte soustavu rovnic*

$$\begin{array}{rcrcrcl} 1,67\,x_1 & - & 0,15\,x_2 & + & 2,51\,x_3 & = & -0,84 \\ 2,15\,x_1 & + & 3,02\,x_2 & - & 0,17\,x_3 & = & 2,32 \\ 1,71\,x_1 & - & 2,83\,x_2 & + & 1,45\,x_3 & = & 1,26 \end{array}$$

Řešení: Koeficienty soustavy opíšeme do matice:

$$\left(\begin{array}{cccc} 1,67 & -0,15 & 2,51 & -0,84 \\ 2,15 & 3,02 & -0,17 & 2,32 \\ 1,71 & -2,83 & 1,45 & 1,26 \end{array}\right)$$

Od druhého řádku odečteme první řádek vynásobený $\frac{2,15}{1,67}$ a od třetího vynásobený $\frac{1,71}{1,67}$ (všechny mezivýsledky jsou zaokrouhlovány na pět desetinných míst):

$$\left(\begin{array}{cccc} 1,67 & -0,15 & 2,51 & -0,84 \\ 0 & 3,21311 & -3,40144 & 3,40144 \\ 0 & -2,67641 & -1,12012 & 2,12012 \end{array}\right)$$

Nyní od třetího řádku odečteme druhý vynásobený $\frac{-2,67641}{3,21311}$. Tím dostaneme

$$\left(\begin{array}{cccc} 1,67 & -0,15 & 2,51 & -0,84 \\ 0 & 3,21311 & -3,40144 & 3,40144 \\ 0 & 0 & -3,95339 & 4,95339 \end{array}\right),$$

což už odpovídá soustavě v trojúhelníkovém tvaru

$$\begin{array}{rcrcrcl} 1,67\,x_1 & - & 0,15\,x_2 & + & 2,51\,x_3 & = & -0,84 \\ & & 3,21311\,x_2 & - & 3,40144\,x_3 & = & 3,40144 \\ & & & - & 3,95339\,x_3 & = & 4,95339 \end{array}$$

Řešení této soustavy je

$$\begin{aligned} x_3 &= \frac{4,95339}{-3,95339} \doteq -1,25295 \\ x_2 &= \frac{1}{3,21311}\Big(3,40144 + 3,40144 \cdot (-1,25295)\Big) \doteq -0,26777 \\ x_1 &= \frac{1}{1,67}\Big(-0,84 + 0,15 \cdot (-0,26777) - 2,51 \cdot (-1,25295)\Big) \doteq 1,35613 \end{aligned}$$

Řešení získané Gaussovou eliminační metodou by bylo přesné, kdybychom se v průběhu výpočtu nedopouštěli zaokrouhlovacích chyb. U některých soustav může být bohužel vliv zaokrouhlování na výsledek značný. Algoritmus Gaussovy eliminace se proto někdy modifikuje způsobem popsaným v následující kapitole.

3.1.3 Eliminace s výběrem hlavního prvku

Eliminace s výběrem hlavního prvku je modifikace Gaussovy eliminační metody, která slouží ke zmenšení zaokrouhlovacích chyb.
Je-li absolutní hodnota některého z dělitelů $a_{ii}^{(i-1)}$ malá ve srovnání s absolutní hodnotou prvků $a_{ki}^{(i-1)}$, $k > i$, může hrozit nebezpečí velkých zaokrouhlovacích chyb. Zaokrouhlovací chyba v absolutní hodnotě malého čísla způsobí totiž velkou chybu v jeho převrácené hodnotě a tedy i v číslech, jimiž násobíme řádky při eliminaci.
Abychom se vyhnuli dělení čísly, která jsou malá vzhledem k ostatním veličinám, použijeme postup zvaný **výběr hlavního prvku**:
V prvním kroku eliminace najdeme rovnici, která má u x_1 v absolutní hodnotě největší koeficient. Vyměníme ji s první rovnicí a pak pomocí jejích násobků eliminujeme x_1 z ostatních rovnic. Ve druhém kroku najdeme mezi všemi rovnicemi kromě první tu rovnici, která má v absolutní hodnotě největší koeficient u x_2. Vyměníme ji s druhou rovnicí a pomocí jejích násobků eliminujeme x_2 z dalších rovnic. Obecně v k-tém kroku eliminace najdeme mezi posledními $n - k + 1$ rovnicemi tu, která má největší koeficient u x_k, vyměníme ji s k-tou rovnicí a pak pomocí ní eliminujeme.

Příklad 3.3 *Soustavu z příkladu 3.2 řešte eliminací s výběrem hlavního prvku.*

Řešení: Postupujeme podobně jako v předchozím příkladu. Vybraný hlavní prvek je vždy v rámečku.

$$\begin{pmatrix} 1,67 & -0,15 & 2,51 & -0,84 \\ \boxed{2,15} & 3,02 & -0,17 & 2,32 \\ 1,71 & -2,83 & 1,45 & 1,26 \end{pmatrix} \sim \begin{pmatrix} 2,15 & 3,02 & -0,17 & 2,32 \\ 0 & -2,49577 & 2,64205 & -2.64205 \\ 0 & \boxed{\text{-5,23195}} & 1.58521 & -0,58521 \end{pmatrix} \sim$$

$$\begin{pmatrix} 2,15 & 3,02 & -0,17 & 2,32 \\ 0 & -5,23195 & 1.58521 & -0,58521 \\ 0 & 0 & 1,88586 & -2,36289 \end{pmatrix}$$

Následovala by zpětná substituce.

Právě popsanou metodu bychom mohli nazvat výstižněji eliminační metodou s **částečným výběrem hlavního prvku**.
Úplný výběr hlavního prvku spočívá v tom, že v k-tém kroku volíme za hlavní prvek ten, který je největší v absolutní hodnotě v submatici vytvořené vynecháním prvních $k-1$ řádků a sloupců v upravované matici. Nutnost hledat největší prvek v celé submatici a vyměňovat řádky i sloupce způsobuje větší časovou (a programátorskou) náročnost této metody. Gaussova eliminační metoda s částečným výběrem je proto obvykle efektivnější než metoda s úplným výběrem hlavního prvku.

Na závěr poznamenejme, že Gaussova eliminační metoda, ať už s výběrem hlavního prvku nebo bez, je pro opravdu velké matice časově náročná. Máme-li řešit n rovnic, je u obyčejné eliminace potřeba vykonat přibližně $n^3/3$ aritmetických operací, což pro velké n dokáže zaměstnat i relativně výkonný počítač. Proto se hodí nejlépe pro nepříliš rozsáhlé soustavy. Dnes však existují profesionální programy i pro řešení velkých soustav rovnic s řídkou maticí koeficientů (řídkou maticí se rozumí taková matice, která má v každém řádku jen malý počet nenulových prvků).

3.2 Iterační metody

Iterační metody, na rozdíl od přímých metod, nevedou k přesnému řešení po konečném, předem daném počtu kroků. U iteračních metod zvolíme počáteční aproximaci řešení a určitým postupem ji v každém kroku metody zlepšíme. K řešení se přibližujeme postupně a obecně ho dosáhneme až v limitě. Protože výpočet nelze provádět do nekonečna, po jisté době jej ukončíme. Výsledkem bude přibližné řešení soustavy.

3.2.1 Jacobiho metoda

Nejprve popíšeme, jak se Jacobiho metodou soustavy rovnic řeší a kdy se touto metodou řešit mohou. Na konci kapitoly teoreticky zdůvodníme, proč Jacobiho metoda funguje. (Aby čtenář děsící se jakékoli teorie mohl konec kapitoly přeskočit a nebyl hned zpočátku zastrašen.)
Budeme opět pracovat se soustavou lineárních rovnic

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

Z první rovnice vyjádříme x_1, ze druhé rovnice x_2 atd. Dostaneme

$$\begin{aligned} x_1 &= \frac{1}{a_{11}}\Big(b_1 - a_{12}\,x_2 - a_{13}\,x_3 - \cdots - a_{1n}\,x_n\Big) \\ x_2 &= \frac{1}{a_{22}}\Big(b_2 - a_{21}\,x_1 - a_{23}\,x_3 - \cdots - a_{2n}\,x_n\Big) \\ &\vdots \\ x_n &= \frac{1}{a_{nn}}\Big(b_n - a_{n1}\,x_1 - a_{n2}\,x_2 - \cdots - a_{n\,n-1}\,x_{n-1}\Big) \end{aligned} \tag{3.5}$$

Řešení soustavy budeme hledat následujícím způsobem:
Libovolně zvolíme počáteční aproximaci řešení $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})^T$.
Tato čísla dosadíme do pravé strany 3.5. Tím dostaneme novou aproximaci řešení $\mathbf{x}^{(1)} = (x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)})^T$. Tu opět dosadíme do pravé strany 3.5 atd.
Obecně každou další aproximaci řešení získáme podle předpisu

$$\begin{aligned} x_1^{(r+1)} &= \frac{1}{a_{11}}\Big(b_1 - a_{12}\,x_2^{(r)} - a_{13}\,x_3^{(r)} - \cdots - a_{1n}\,x_n^{(r)}\Big) \\ x_2^{(r+1)} &= \frac{1}{a_{22}}\Big(b_2 - a_{21}\,x_1^{(r)} - a_{23}\,x_3^{(r)} - \cdots - a_{2n}\,x_n^{(r)}\Big) \\ &\vdots \\ x_n^{(r+1)} &= \frac{1}{a_{nn}}\Big(b_n - a_{n1}\,x_1^{(r)} - a_{n2}\,x_2^{(r)} - \cdots - a_{n\,n-1}\,x_{n-1}^{(r)}\Big), \end{aligned} \tag{3.6}$$

Za jistých (dále popsaných podmínek) se tímto postupem budeme přibližovat k přesnému řešení soustavy.

Ve výpočtu pokračujeme, dokud se nedosáhne určité předem dané přesnosti, např. dokud se aproximace řešení neustálí na požadovaném počtu desetinných míst, nebo dokud není překročen předem daný maximální počet kroků.

Jacobiho metodou nemusíme řešení soustavy najít vždy. V některých případech posloupnost postupných aproximací k řešení soustavy nekonverguje. Uvedeme nyní podmínky, které zaručí, že metoda konverguje (tj. najdeme pomocí ní přibližné řešení).

Definice. Matice A se nazývá **řádkově ostře diagonálně dominantní** právě tehdy, když

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}| \quad \text{pro } i = 1, \ldots, n \tag{3.7}$$

(neboli když je v každém řádku matice absolutní hodnota prvku na diagonále větší než součet absolutních hodnot všech ostatních prvků v onom řádku)
a **sloupcově ostře diagonálně dominantní** právě tehdy, když

$$|a_{jj}| > \sum_{i=1, i\neq j}^{n} |a_{ij}| \quad \text{pro } j = 1, \ldots, n \tag{3.8}$$

(neboli když je v každém sloupci matice absolutní hodnota prvku na diagonále větší než součet absolutních hodnot všech ostatních prvků v onom sloupci).

Na konci této kapitoly dokážeme, že:

Je-li matice soustavy 3.2 ostře řádkově nebo sloupcově diagonálně dominantní, Jacobiho metoda konverguje.

Jestliže matice soustavy 3.2 není diagonálně dominantní, Jacobiho metoda konvergovat může a nemusí.
Existuje podmínka pro konvergenci Jacobiho metody nutná a dostatečná (tj. pokud je splněna, metoda konverguje a pokud není splněna, metoda diverguje), jenže je pro velké matice prakticky neověřitelná.
Proto, nejsme-li si jisti konvergencí metody, je vhodné stanovit maximální počet kroků a je-li překročen, výpočet ukončit s tím, že metoda diverguje. Pak je potřeba zvolit jinou metodu nebo soustavu nějak upravit.

Příklad 3.4 *Jacobiho metodou řešte soustavu*

$$\begin{array}{rcrcrcr} 15\,x_1 & - & x_2 & + & 2\,x_3 & = & 30 \\ 2\,x_1 & - & 10\,x_2 & + & x_3 & = & 23 \\ x_1 & + & 3\,x_2 & + & 18\,x_3 & = & -22 \end{array}$$

Řešení: Matice soustavy je diagonálně dominantní, protože platí

$$|15| > |-1| + |2|\ ,\ |-10| > |2| + |1|\ ,\ |18| > |1| + |3|.$$

Proto je konvergence metody zaručena. Vypíšeme iterační vztahy:

$$
\begin{aligned}
x_1^{(r+1)} &= \frac{1}{15}\Big(30 + x_2^{(r)} - 2\,x_3^{(r)}\Big)\\
x_2^{(r+1)} &= -\frac{1}{10}\Big(23 - 2\,x_1^{(r)} - x_3^{(r)}\Big)\\
x_3^{(r+1)} &= \frac{1}{18}\Big(-22 - x_1^{(r)} - 3\,x_2^{(r)}\Big)
\end{aligned}
$$

Jako počáteční aproximaci zvolíme $\mathbf{x} = (0,0,0)^T$. Postupně získávané aproximace řešení budeme zapisovat do tabulky:

r	$x_1^{(r)}$	$x_2^{(r)}$	$x_3^{(r)}$
0	0	0	0
1	2	-2,3	-1,2222
2	2,0096	-2,0222	-0,9500
3	1,9918	-1,9930	-0,9968
4	2,0000	-2,0013	-1,0007

Je vidět, že posloupnost postupných aproximací konverguje k řešení soustavy (2,-2,-1). Kdybychom chtěli získat řešení s přesností $\varepsilon = 0,01$, mohli bychom nyní výpočet zastavit, protože

$$
\begin{aligned}
|\,x_1^{(4)} - x_1^{(3)}| &= |\,2,0000 - 1,9918| < 0,01\\
|\,x_2^{(4)} - x_2^{(3)}| &= |-2,0013 - (-1,9930)| < 0,01\\
|\,x_3^{(4)} - x_3^{(3)}| &= |-1,0007 - (-0,9968)| < 0,01,
\end{aligned}
$$

zatímco kdybychom požadovali přesnost $\varepsilon = 0,001$, museli bychom ve výpočtu pokračovat, protože např. $|\,x_1^{(4)} - x_1^{(3)}| > 0,001$.

Ukázka divergence Jacobiho metody
Kdybychom rovnice z předcházejícího příkladu přepsali v jiném pořadí, např.

$$
\begin{aligned}
x_1 &+ 3\,x_2 &+ 18\,x_3 &= -22\\
15\,x_1 &- x_2 &+ 2\,x_3 &= 30\\
2\,x_1 &- 10\,x_2 &+ x_3 &= 23,
\end{aligned}
$$

příslušné iterační vztahy by vypadaly takto:

$$
\begin{aligned}
x_1^{(r+1)} &= -22 - 3\,x_2^{(r)} - 18\,x_3^{(r)}\\
x_2^{(r+1)} &= -30 + 15\,x_1^{(r)} + 2\,x_3^{(r)}\\
x_3^{(r+1)} &= 23 - 2\,x_1^{(r)} + 10\,x_2^{(r)}.
\end{aligned}
$$

Podmínka konvergence metody není splněna. Podívejme se, jak se budou chovat postupné aproximace řešení:

r	$x_1^{(r)}$	$x_2^{(r)}$	$x_3^{(r)}$
0	0	0	0
1	-22	-30	23
2	-346	-314	-233
3	5114	-5686	-2425

Na první pohled je zřejmé, že k řešení soustavy $(2, -2, -1)$ touto cestou nedojdeme, metoda diverguje.

Jacobiho metoda z teoretického hlediska

Nyní ukážeme, proč Jacobiho metoda funguje a proč konverguje zrovna za výše uvedených podmínek.
Rovnice 3.5 se dají zapsat maticově jako

$$\mathbf{x} = \mathbf{C}_J\,\mathbf{x} + \mathbf{d},$$

kde $\mathbf{C}_J$ je tzv. **iterační matice** Jacobiho metody. Prvky matice $\mathbf{C}_J$ a vektoru $\mathbf{d}$ jsou

$$\begin{aligned} c_{ij} &= -\frac{a_{ij}}{a_{ii}} \quad \text{pro } i \neq j\ , \quad c_{ii} = 0 \\ d_i &= \frac{b_i}{a_{ii}}. \end{aligned}$$

Tím, že jsme původní soustavu rovnic $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ upravili na tento tvar, se úkol najít řešení soustavy rovnic převedl na hledání pevného bodu zobrazení

$$F(\mathbf{x}) = \mathbf{C}_J\,\mathbf{x} + \mathbf{d}, \tag{3.9}$$

protože řešením původní soustavy rovnic je právě takový vektor $\mathbf{x}$, pro nějž platí $F(\mathbf{x}) = \mathbf{x}$.

V kapitole 2 jsme předvedli obecný postup, který vede k nalezení pevného bodu. Je to tzv. metoda postupných aproximací, iterační proces.
Proto řešení hledáme výše popsaným způsobem, tj. zvolíme libovolně počáteční aproximaci $\mathbf{x}^{(0)}$ a další aproximace počítáme jako

$$\mathbf{x}^{(r+1)} = F(\mathbf{x}^{(r)}) = \mathbf{C}_J\,\mathbf{x}^{(r)} + \mathbf{d}. \tag{3.10}$$

Dále jsme v kapitole 2 uvedli, za jakých podmínek je jisté, že pevný bod zobrazení existuje a že metodou postupných aproximací k němu dojdeme. Prozkoumáme nyní, jak vypadají tyto obecné podmínky pro naši konkrétní situaci.
Máme zobrazení $F : V_n \to V_n$, kde V_n je prostor všech uspořádaných n-tic reálných čísel. Na tomto prostoru můžeme zavést metriku předpisem

$$d(\mathbf{x}, \mathbf{y}) = \|\,\mathbf{x} - \mathbf{y}\|,$$

kde $\|\cdot\|$ je některá z norem 2.12, 2.13. Prostor V_n s touto metrikou je úplný. Zjistíme, kdy bude zobrazení F kontraktivní.
Platí

$$\begin{aligned} d(F(\mathbf{x}), F(\mathbf{y})) &= \|\,F(\mathbf{x}) - F(\mathbf{y})\| = \|\,\mathbf{C}_J\,\mathbf{x} + \mathbf{d} - (\mathbf{C}_J\,\mathbf{y} + \mathbf{d})\| = \|\,\mathbf{C}_J\,(\mathbf{x} - \mathbf{y})\| \leq \\ &\leq \|\mathbf{C}_J\| \cdot \|\,\mathbf{x} - \mathbf{y}\| = \|\mathbf{C}_J\| \cdot d(\mathbf{x}, \mathbf{y}), \end{aligned}$$

kde $\|\mathbf{C}_J\|$ je norma matice přidružená použité normě vektoru.

Je-li tedy $\|\mathbf{C}_J\| < 1$, je zobrazení F kontraktivní s koeficientem kontrakce $\alpha = \|\mathbf{C}_J\|$ a je zaručeno, že posloupnost postupných aproximací získaná podle předpisu 3.10 konverguje k pevnému bodu zobrazení 3.9.
(Je-li $\|\mathbf{C}_J\| > 1$, o konvergenci či divergenci iteračního procesu nevíme nic.)

Nyní se podíváme na to, jak podmínka $\|\mathbf{C}_J\| < 1$ souvisí s diagonální dominantností matice soustavy $\mathbf{A}$.
Předpokládejme, že matice $\mathbf{A}$ je ostře řádkově diagonálně dominantní.
Počítáme-li řádkovou normu matice $\mathbf{C}_J$, bereme součty absolutních hodnot prvků v jednotlivých řádcích a z nich pak vybíráme maximum. Součet absolutních hodnot prvků prvního řádku je

$$\left|-\frac{a_{12}}{a_{11}}\right| + \left|-\frac{a_{13}}{a_{11}}\right| + \cdots + \left|-\frac{a_{1n}}{a_{11}}\right| = \frac{|\,a_{12}| + |\,a_{13}| + \cdots + |\,a_{1n}|}{|a_{11}|}.$$

Protože je $\mathbf{A}$ řádkově diagonálně dominantní, musí být

$$|\,a_{11}| > |\,a_{12}| + |\,a_{13}| + \cdots + |\,a_{1n}|$$

a tedy součet absolutních hodnot prvků prvního řádku matice $\mathbf{C}_J$ musí být menší než 1. Úplně stejně se ukáže, že i součty v ostatních řádcích jsou menší než jedna.
Řádková norma matice $\mathbf{C}_J$, coby největší z čísel menších než jedna, bude určitě také menší než jedna. Proto, je-li $\mathbf{A}$ řádkově diagonálně dominantní, je zaručeno, že Jacobiho metoda konverguje.

Zcela analogicky se dá ukázat, že je-li $\mathbf{A}$ ostře sloupcově diagonálně dominantní, je sloupcová norma matice $\mathbf{C}_J$ menší než 1.

V případě, že je $\|\mathbf{C}_J\| < 1$, platí odhady 2.8 a 2.9 z věty 2.1. Zde jsou přepsány speciálně pro naši úlohu:

$$\|\,\mathbf{x}^{(r)} - \mathbf{x}\| \le \frac{\|\mathbf{C}_J\|}{1 - \|\mathbf{C}_J\|}\,\|\,\mathbf{x}^{(r)} - \mathbf{x}^{(r-1)}\| \tag{3.11}$$

$$\|\,\mathbf{x}^{(r)} - \mathbf{x}\| \le \frac{\|\mathbf{C}_J\|^r}{1 - \|\mathbf{C}_J\|}\,\|\,\mathbf{x}^{(0)} - \mathbf{x}^{(1)}\| \tag{3.12}$$

Pomocí odhadu 3.11 můžeme rozhodnout, kdy zastavit iterační proces, chceme-li mít jistotu, že se přibližné řešení od přesného v použité normě neliší víc než o předem dané ε.
Odhad 3.12 může posloužit k určení počtu kroků metody, který bude stačit pro dosažení přesnosti ε.
Protože však pro velké soustavy rovnic je vypočítat normu matice $\mathbf{C}_J$ pracné, pro zastavení výpočtu se spíše používá kriterium

$$\|\,\mathbf{x}^{(r)} - \mathbf{x}^{(r-1)}\| < \varepsilon,$$

i když jeho splněním není zaručeno, že bude i $\|\,\mathbf{x}^{(r)} - \mathbf{x}\| < \varepsilon$.
(Toto kriterium se objevilo již v příkladu 3.4, použita byla norma $\|\cdot\|_\infty$.)

Příklad 3.5 *Odhadněte, o kolik se nanejvýš liší přibližné řešení získané v příkladu 3.4 od přesného řešení v normě* $\|\cdot\|_\infty$.

Řešení: K odhadu chyby použijeme vzorec 3.11. K tomu musíme vypočítat řádkovou normu iterační matice $\mathbf{C}_J$. Nejprve vypíšeme samotnou iterační matici:

$$\mathbf{C}_J = \begin{pmatrix} 0 & \frac{1}{15} & -\frac{2}{15} \\ \frac{2}{10} & 0 & \frac{1}{10} \\ -\frac{1}{18} & -\frac{3}{18} & 0 \end{pmatrix}$$

$\|\mathbf{C}_J\|_\infty = \max\left(\frac{3}{15}, \frac{3}{10}, \frac{4}{18}\right) = \frac{3}{10} = 0,3\,.$
Dále vypočteme normu rozdílu posledních dvou získaných aproximací
$\mathbf{x}^{(3)} = (1,9918\,;\,-1,9930\,;\,-0,9968)$ a $\mathbf{x}^{(4)} = (2,0000\,;\,-2,0013\,;\,-1,0007)$:

$$\|\mathbf{x}^{(4)} - \mathbf{x}^{(3)}\|_\infty = \max(|\,0,0082|\,;\,|-0,0095|\,;\,|-0,0039|) = 0,0095$$

Nyní dosadíme do 3.11

$$\|\mathbf{x}^{(4)} - \mathbf{x}\|_\infty \leq \frac{0,3}{1-0,3} \cdot 0,0095 \doteq 0,0041$$

To znamená, že každá ze složek přibližného řešení $\mathbf{x}^{(4)}$ se od odpovídající složky přesného řešení může lišit nanejvýš o 0,0041.

3.2.2 Gauss-Seidelova metoda

Gauss-Seidelova metoda je velmi podobná metodě Jacobiho. Liší se od ní pouze v tom, že při výpočtu další aproximace řešení použijeme vždy nejnovější přibližné hodnoty $x_1, x_2, \ldots, x_n$, které máme k dispozici.
Podrobněji: $x_1^{(r+1)}$ vypočteme stejně jako u Jacobiho metody a při výpočtu $x_2^{(r+1)}$ je ihned použijeme (zatímco u Jacobiho metody jsme použili staré $x_1^{(r)}$). Při výpočtu $x_3^{(r+1)}$ použijeme nové $x_1^{(r+1)}$ a $x_2^{(r+1)}$ atd.
Obecně iterační vztahy vypadají takto:

$$\begin{aligned}
x_1^{(r+1)} &= \frac{1}{a_{11}}\Big(b_1 - a_{12}\,x_2^{(r)} - a_{13}\,x_3^{(r)} - \cdots - a_{1n}\,x_n^{(r)}\Big) \qquad (3.13)\\
x_2^{(r+1)} &= \frac{1}{a_{22}}\Big(b_2 - a_{21}\,x_1^{(r+1)} - a_{23}\,x_3^{(r)} - \cdots - a_{2n}\,x_n^{(r)}\Big)\\
x_3^{(r+1)} &= \frac{1}{a_{33}}\Big(b_3 - a_{31}\,x_1^{(r+1)} - a_{32}\,x_2^{(r+1)} - \cdots - a_{2n}\,x_n^{(r)}\Big)\\
&\vdots\\
x_n^{(r+1)} &= \frac{1}{a_{nn}}\Big(b_n - a_{n1}\,x_1^{(r+1)} - a_{n2}\,x_2^{(r+1)} - \cdots - a_{n\,n-1}\,x_{n-1}^{(r+1)}\Big),
\end{aligned}$$

Dá se dokázat, že **je-li matice soustavy 3.2 ostře řádkově diagonálně dominantní, Gauss-Seidelova metoda konverguje.**

V jiném kritériu konvergence se objevuje pojem pozitivně definitní matice. Protože není jisté, zda se s ním studenti již setkali, řekneme, co to je.

Definice. Symetrická matice $\mathbf{A}$ řádu n se nazývá **pozitivně definitní**, jestliže pro každý nenulový sloupcový vektor $\mathbf{x} = (x_1, \ldots, x_n)^T$ platí

$$\mathbf{x}^T \mathbf{A}\, \mathbf{x} > 0$$

Příklad. Pozitivně definitní je např. matice

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

protože pro každý vektor $\mathbf{x} = (x_1, x_2)^T \neq (0,0)^T$ platí

$$\mathbf{x}^T \mathbf{A}\, \mathbf{x} = x_1^2 + 4x_1x_2 + 5x_2^2 = (x_1 + 2x_2)^2 + x_2^2 > 0\,,$$

zatímco matice

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

není pozitivně definitní, protože např. pro $\mathbf{x} = (1,0)^T$ platí

$$(1,0)\, \mathbf{B} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = (1,0) \begin{pmatrix} -1 \\ 2 \end{pmatrix} = -1 < 0\,.$$

Platí: **Je-li matice soustavy 3.2 pozitivně definitní, Gauss-Seidelova metoda konverguje.**

Ověření toho, že je daná matice pozitivně definitní, je náročné a pro velké matice prakticky neproveditelné. Naštěstí je u některých úloh z povahy řešeného problému předem jasné, že matice soustavy pozitivně definitní bude.
Pozitivní definitnost je vlastnost prospěšná nejen u Gauss-Seidelovy iterační metody. Dá se např. ukázat, že je-li matice soustavy symetrická pozitivně definitní, Gaussova eliminační metoda je málo citlivá na zaokrouhlovací chyby.

Poznámka. Vynásobíme-li libovolnou regulární čtvercovou matici $\mathbf{A}$ zleva maticí k ní trasponovanou, vzniklá matice $\mathbf{A}^T\mathbf{A}$ bude symetrická a pozitivně definitní.
Proto, vynásobíme-li soustavu rovnic $\mathbf{Ax} = \mathbf{b}$ s regulární maticí $\mathbf{A}$ zleva maticí $\mathbf{A}^T$, dostaneme novou soustavu

$$\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b},$$

jejíž matice je pozitivně definitní a je tedy zaručeno, že Gauss-Seidelova metoda bude pro tuto novou soustavu konvergovat.
V případě takto získaných soustav však Gauss-Seidelova metoda může konvergovat velmi pomalu.

Příklad 3.6 *Gauss-Seidelovou metodou řešte tutéž soustavu jako v příkladu 3.4, t.j.*

$$\begin{aligned} 15\,x_1 &- x_2 &+ 2\,x_3 &= 30 \\ 2\,x_1 &- 10\,x_2 &+ x_3 &= 23 \\ x_1 &+ 3\,x_2 &+ 18\,x_3 &= -22 \end{aligned}$$

Řešení: Již jsme ověřili, že podmínka konvergence je splněna. Vypíšeme iterační vztahy:

$$\begin{aligned} x_1^{(r+1)} &= \frac{1}{15}\Big(30 + x_2^{(r)} - 2\,x_3^{(r)}\Big) \\ x_2^{(r+1)} &= -\frac{1}{10}\Big(23 - 2\,x_1^{(r+1)} - x_3^{(r)}\Big) \\ x_3^{(r+1)} &= \frac{1}{18}\Big(-22 - x_1^{(r+1)} - 3\,x_2^{(r+1)}\Big) \end{aligned}$$

Jako počáteční aproximaci zvolíme opět $\mathbf{x} = (0, 0, 0)^T$.

r	$x_1^{(r)}$	$x_2^{(r)}$	$x_3^{(r)}$
0	0	0	0
1	2	-1,9	-1.0167
2	2,0089	-1,9999	-1,0005
3	2,0001	-2,0000	-1,0000
4	2,0000	-2,0000	-1,0000

Vidíme, že se k řešení soustavy přibližujeme rychleji než pomocí Jacobiho metody.

I obecně se dá říci, že Gauss-Seidelova metoda obvykle konverguje rychleji než metoda Jacobiho. Proto se používá častěji. Další její výhodou oproti Jacobiho metodě je, že pro uložení přibližného řešení v paměti počítače nám stačí jediné pole, jehož složky postupně přepisujeme, zatímco u Jacobiho metody si musíme pamatovat pole dvě: starou a novou aproximaci řešení.

Shrnutí pojmů

Gaussova eliminace a Cramerovo pravidlo vedou přímo k řešení soustavy. Kdybychom se nedopouštěli zaokrouhlovacích chyb, našli bychom pomocí těchto metod přesné řešení.

Základem Gaussovy eliminační metody je úprava matice soustavy na trojúhelníkový tvar. Ten dostaneme pomocí přičítání vhodných násobků vybraných řádků matice k ostatním řádkům.
Vliv zaokrouhlovacích chyb u Gaussovy eliminace může být značný, zvlášť u některých typů matic. Proto se používá tzv. eliminace s výběrem hlavního prvku.
Eliminační metoda je velmi náročná z časového i paměťového hlediska. Nejlépe se hodí pro nepříliš rozsáhlé soustavy s plnou maticí.

U Cramerova pravidla jednotlivé neznámé počítáme jako podíly determinantů. Cramerovo pravidlo je vhodné pouze pro velmi malé soustavy rovnic.

Pomocí iteračních metod obvykle najdeme pouze přibližné řešení soustavy (pokud nenastane dosti nepravděpodobný případ, kdy se v některém kroku trefíme přímo do řešení).

Na začátku zvolíme počáteční aproximaci řešení, a tu pak opakovaným dosazováním do iteračních vztahů, např. 3.6 (u Jacobiho metody) nebo 3.13 (u Gauss-Seidelovy metody), zpřesňujeme. S výpočtem skončíme obvykle tehdy, je-li norma rozdílu po sobě jdoucích aproximací dostatečně malá.
Iterační metody mohou divergovat (řešení pomocí nich nemusíme najít). Zda bude metoda konvergovat, či nikoli, závisí na vlastnostech matice soustavy. U Jacobiho metody zaručí konvergenci řádková nebo sloupcová diagonální dominance , u Gauss-Seidelovy metody řádková diagonální dominance nebo pozitivní definitnost matice.
Iterační metody jsou vhodné pro řešení velkých soustav s řídkou maticí koeficientů. Pro řešení malého počtu rovnic vhodné nejsou, tam lépe poslouží eliminace.

3.3 Otázky a příklady ke cvičení

U následujících výroků rozhodněte, zda se jedná o výrok pravdivý či nepravdivý.

Otázka 3.1 *Cramerovo pravidlo se hodí pro řešení malého počtu rovnic.*

Otázka 3.2 *Řešení získané Gaussovou eliminační metodou může být ovlivněno zaokrouhlovacími chybami.*

Otázka 3.3 *Výběr hlavního prvku slouží k urychlení algoritmu Gaussovy eliminace.*

Otázka 3.4 *Jacobiho metoda vždy konverguje.*

Otázka 3.5 *Konvergence či divergence Gauss-Seidelovy metody závisí pouze na volbě počáteční aproximace* $\mathbf{x}^{(0)}$.

Otázka 3.6 *Konvergence či divergence Gauss-Seidelovy metody závisí na vlastnostech matice řešené soustavy.*

Otázka 3.7 *Jacobiho metoda je vhodná pro soustavu dvou rovnic o dvou neznámých.*

Otázka 3.8 *Jestliže* $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| < \varepsilon$, *kde* $\mathbf{x}^{(k)}$ *a* $\mathbf{x}^{(k-1)}$ *jsou po sobě jdoucí aproximace řešení získané Gauss-Seidelovou metodou, pak je zaručeně i* $\|\mathbf{x}^{(k)} - \mathbf{x}\| < \varepsilon$ *(*$\mathbf{x}$ *je přesné řešení soustavy).*

Otázka 3.9 *Jestliže u Jacobiho metody vyjde* $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)}$, *nalezli jsme přesné řešení a platí* $\mathbf{x} = \mathbf{x}^{(k)}$.

Příklad 3.1 *Pomocí Cramerova pravidla vyřešte soustavu rovnic*

$$\begin{array}{rcrcl} 2,43x & + & 7,21y & = & 1,25 \\ 8,03x & - & 4,20y & = & 5,69 \end{array}$$

Příklad 3.2 *Pomocí Gaussovy eliminační metody vyřešte soustavu rovnic*

$$\begin{array}{rcrcrcr} 9,50x & + & 4.86y & - & 4,56z & = & -8,90 \\ -2,31x & + & 8,91y & + & 0,19z & = & 6,15 \\ 6,07x & + & 7,62y & + & 8,21z & = & -7,92 \end{array}$$

Příklad 3.3 *Pomocí Jacobiho metody řešte soustavu rovnic*

$$\begin{aligned} 20x &+ 3y - 5z &= -12 \\ 2x &+ 25y - 3z &= 13 \\ 4x &- 5y - 32z &= 10 \end{aligned}$$

Ověřte, že jsou splněny podmínky konvergence metody, pak proveďte tři kroky, počáteční aproximaci volte $(0, 0, 0)$.

Příklad 3.4 *Pomocí Gauss-Seidelovy metody najděte řešení soustavy rovnic*

$$\begin{aligned} 28x &- 6y + 2z &= 25 \\ 2x &- 30y + 4z &= -10 \\ x &+ 3y + 36z &= -16 \end{aligned}$$

s přesností $\varepsilon = 0,001$. *Ověřte, že jsou splněny podmínky konvergence metody. Počáteční aproximaci volte* $(0, 0, 0)$.

Příklad 3.5 *Upravte následující soustavu tak, aby byla zaručena konvergence Gauss-Seidelovy metody. Pak udělejte dva kroky metody, vyjděte z* $(0, 0, 0)$.

$$\begin{aligned} 2x &- y &= -3 \\ 3x &+ z &= -6 \\ -2x &+ 2y + 4z &= 2 \end{aligned}$$

Odpovědi na otázky a řešení příkladů viz 9.3

Programovací úlohy

Úlohy označené * jsou obtížnější a nejsou míněny pro běžné cvičení, ale spíše jako námět pro zájemce.

Programovací úloha 1 Napište program, který řeší soustavu (max. 20) lineárních rovnic

a) Gaussovou eliminační metodou

b) Gaussovou eliminační metodou s částečným výběrem hlavního prvku

c)* Gaussovou eliminační metodou s úplným výběrem hlavního prvku

Programovací úloha 2 Napište program, který řeší soustavu (max. 20) rovnic

a) Jacobiho metodou

b) Gauss-Seidelovou metodou

Programovací úloha 3 * Napište program, který řeší velkou soustavu rovnic s řídkou maticí (= v paměti držte pouze nenulové prvky matice) Jacobiho nebo Gauss-Seidelovou metodou.

4 Numerické metody řešení nelineárních rovnic

Cíl kapitoly

V této kapitole se seznámíme s některými metodami pro řešení rovnice $f(x) = 0$ a pro řešení soustavy rovnic $\mathbf{F}(x_1, x_2, \ldots, x_n) = \mathbf{o}$.
Ukážeme, co lze od jednotlivých metod očekávat: zda jsou vždy konvergentní (tj. řešení, existuje-li, pomocí nich vždy najdeme), nebo zda mohou divergovat. U metod, které nemusí konvergovat vždy, ukážeme podmínky, které konvergenci zaručí, a zmíníme se o tom, jak případnou divergenci ošetřit v počítačovém programu. Také budeme zkoumat, a hlavně na příkladech předvádět, rychlost jednotlivých metod.

4.1 Numerické metody řešení jedné nelineární rovnice

Budeme se zabývat řešením nelineární rovnice

$$f(x) = 0, \tag{4.1}$$

tj. hledáním takových bodů $\xi \in \mathbb{R}$, že $f(\xi) = 0$. Takovéto body budeme nazývat **kořeny rovnice** 4.1.

Při hledání kořenů rovnice 4.1 nejprve zjistíme, kolik kořenů rovnice má a najdeme intervaly obsahující právě jeden kořen rovnice. Tato část řešení se nazývá **separace kořenů rovnice.**
Pak budeme pomocí některé z dále popsaných metod hledat přibližnou hodnotu vybraného kořene rovnice.

Při hledání kořenů je užitečná následující věta, jejíž význam je patrný z obrázku 4.1

Věta 4.1 *Je-li funkce f spojitá na intervalu $\langle a, b\rangle$ a platí-li*

$$f(a) \cdot f(b) < 0, \tag{4.2}$$

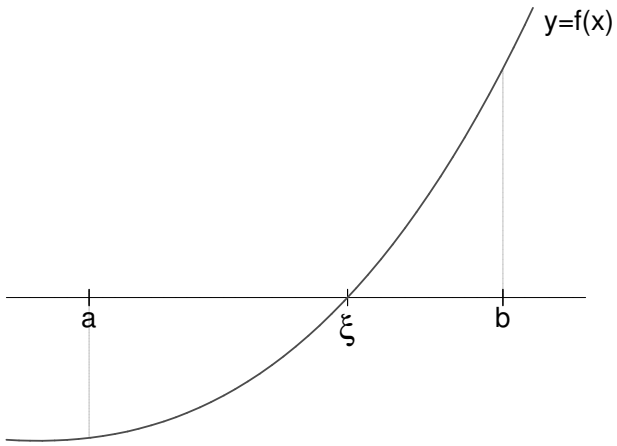
pak v intervalu $\langle a, b\rangle$ leží alespoň jeden kořen rovnice $f(x) = 0$.

Poznámka. Podmínka 4.2 znamená, že znaménka funkčních hodnot v krajních bodech intervalu $\langle a, b\rangle$ jsou opačná.
Kořenů rovnice může být v uvedeném intervalu i více, o jejich počtu věta nic neříká. Na druhou stranu, není-li podmínka 4.2 splněna, neznamená to, že v intervalu $\langle a, b\rangle$ žádný kořen rovnice neleží.

Pro nalezení počtu a polohy kořenů je vhodné prozkoumat vlastnosti funkce f a načrtnout (nebo si pomocí vhodného prostředku nechat načrtnout) její graf.
U některých úloh je možné upravit rovnici 4.1 na tvar

$$f_1(x) = f_2(x),$$

kde f_1 a f_2 jsou funkce, jejichž grafy umíme nakreslit. V bodech, kde se grafy funkcí f_1 a f_2 protnou, se nacházejí kořeny původní rovnice.



Obrázek 4.1: Podmínka $f(a) \cdot f(b) < 0$ zaručí existenci kořene.

Příklad 4.1 *Najděte počet kořenů rovnice*

$$e^x + x^2 - 3 = 0$$

a intervaly, v nichž tyto kořeny leží.

Řešení: Zadanou rovnici můžeme upravit na tvar

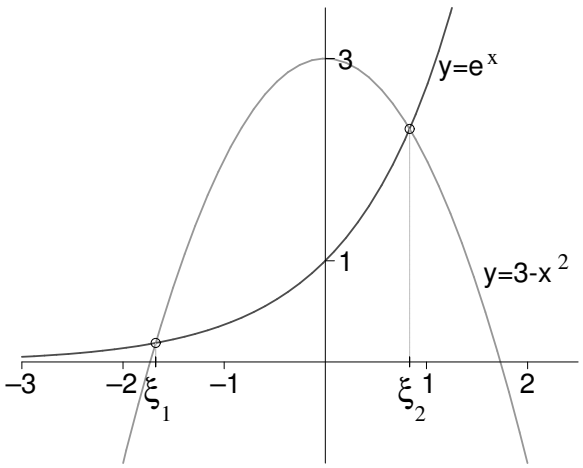
$$e^x = 3 - x^2.$$

Grafy funkcí $f_1(x) = e^x$ a $f_2(x) = 3 - x^2$ umíme načrtnout - viz obrázek 4.2. Z obrázku vidíme, že rovnice má právě dva kořeny ξ_1 a ξ_2, $\xi_1 \in \langle -2, -1 \rangle$, $\xi_2 \in \langle 0, 1 \rangle$.

Nyní postupně probereme metody, které lze použít pro nalezení kořenů rovnice 4.1. **Všude dál v této kapitole budeme předpokládat, že funkce f je na zkoumaném intervalu spojitá.**

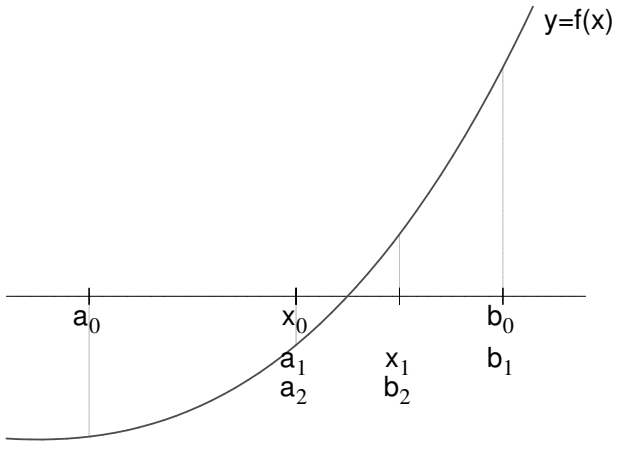
4.1.1 Metoda půlení intervalu

Metoda půlení intervalu je nejjednodušší z metod řešení nelineárních rovnic.
Mějme interval $\langle a, b \rangle$ takový, že $f(a) \cdot f(b) < 0$, tj. leží v něm alespoň jeden kořen rovnice $f(x) = 0$. Tento výchozí interval označíme jako $\langle a_0, b_0 \rangle$. Interval rozpůlíme. Jeho střed je $x_0 = \frac{a_0 + b_0}{2}$. Z intervalů $\langle a_0, x_0 \rangle$, $\langle x_0, b_0 \rangle$ vybereme ten, ve kterém je zaručena existence kořene. Který z nich to je, rozeznáme podle znamének funkčních hodnot v krajních bodech. Je-li $f(a_0) \cdot f(x_0) < 0$, budeme pokračovat s intervalem $\langle a_0, x_0 \rangle$, v opačném případě s intervalem $\langle x_0, b_0 \rangle$. (Platí-li $f(x_0) = 0$, nalezli jsme kořen rovnice a výpočet ukončíme.) Nový interval poloviční délky označíme $\langle a_1, b_1 \rangle$, opět jej rozpůlíme a stejným způsobem pokračujeme.



Obrázek 4.2: K příkladu 4.1 - separace kořenů rovnice

Takto postupně sestrojíme posloupnost intervalů $\langle a_0, b_0\rangle$, $\langle a_1, b_1\rangle$, $\langle a_2, b_2\rangle$, . . . Každý další interval získáme tak, že z předchozího (na základě znamének funkčních hodnot v krajních bodech a uprostřed) vybereme tu jeho polovinu, která obsahuje kořen rovnice - viz obrázek 4.3.



Obrázek 4.3: Metoda půlení intervalu

V půlení pokračujeme tak dlouho, dokud nenarazíme na kořen rovnice, nebo dokud se

interval nezúží na předem danou délku 2ε, neboli dokud pro nějaké k neplatí

$$b_k - a_k < 2\varepsilon$$

Za přibližnou hodnotu kořene pak vezmeme střed posledního nalezeného intervalu

$$x_k = \frac{a_k + b_k}{2}$$

Protože kořen se určitě nachází uvnitř posledního intervalu, může se x_k od přesné hodnoty kořene lišit nanejvýš o polovinu jeho délky, tj. o ε,

$$|\, x_k - \xi| < \varepsilon.$$

Touto metodou kořen rovnice 4.1 nalezneme vždy. Obsahuje-li výchozí interval $\langle a, b\rangle$ více kořenů, najdeme jeden z nich. Nevýhodou metody půlení intervalu je, že konverguje (přibližuje se ke kořeni) dosti pomalu. Proto je vhodné použít ji na zúžení původního intervalu a pak pokračovat jinou, rychlejší metodou.

Příklad 4.2 *Metodou půlení intervalu najděte kladný kořen rovnice z příkladu 4.1*

$$e^x + x^2 - 3 = 0$$

s přesností $\varepsilon = 0,01$.

Řešení: Kladný kořen zadané rovnice leží v intervalu $\langle 0, 1\rangle$. Postupně vypočítávané hodnoty a_k, b_k, x_k budeme zapisovat do tabulky. Je vhodné si také zapisovat znaménka funkčních hodnot funkce $f(x) = e^x + x^2 - 3$ v těchto bodech.

k	a_k	b_k	x_k	$f(a_k)$	$f(b_k)$	$f(x_k)$
0	0	1	0,5	-	+	-
1	0,5	1	0,75	-	+	-
2	0,75	1	0,875	-	+	+
3	0,75	0,875	0,8125	-	+	-
4	0,8125	0,875	0,84375	-	+	+
5	0,8125	0,84375	0,828125	-	+	-
6	0,828125	0,84375	0,8359375			

Nyní můžeme výpočet ukončit, protože $b_6 - a_6 < 2 \cdot 0,01$. Řešení rovnice $e^x + x^2 - 3 = 0$ s přesností $0,01$ je $x_6 \doteq 0,84$.

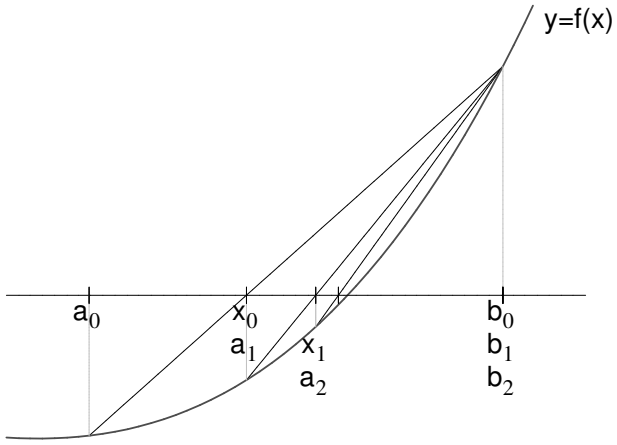
Příklad 4.3 *Kolik dalších kroků metody půlení intervalu by bylo potřeba provést v předchozím příkladu, kdybychom chtěli najít řešení s přesností* $0,001$ *?*

Řešení: V každém kroku se interval zkrátí na polovinu. Vyjdeme-li z intervalu délky l, po k krocích se zúží na $\frac{l}{2^k}$. V našem případě vycházíme z intervalu $\langle a_6, b_6\rangle$ délky $0,015625$. Hledáme tedy k tak, aby platilo $\frac{0,015625}{2^k} < 2 \cdot 0,001$. Odtud $k > \frac{\ln \frac{0,015625}{0,002}}{\ln 2} \doteq 2,97$. Musíme tedy udělat ještě tři kroky.

Je vidět, že počet kroků metody půlení intervalu nutný k nalezení kořene se zadanou přesností vůbec nezávisí na řešené rovnici. Dá se ukázat (podobně jako v řešení příkladu 4.3), že k zpřesnění výsledku o jedno desetinné místo je vždy potřeba udělat 3-4 kroky této metody.

4.1.2 Metoda regula falsi

Princip metody regula falsi je velmi podobný jako u metody půlení intervalu. Opět postupně zužujeme interval obsahující kořen rovnice 4.1. Tentokrát ale dělicím bodem není polovina intervalu, nýbrž průsečík sečny vedené body $[a_k, f(a_k)]$ a $[b_k, f(b_k)]$ s osou x - viz obrázek 4.4.



Obrázek 4.4: Metoda regula falsi

Tento průsečík vypočteme podle vzorce

$$x_k = b_k - \frac{b_k - a_k}{f(b_k) - f(a_k)} f(b_k) \tag{4.3}$$

Z intervalů $\langle a_k, x_k \rangle$, $\langle x_k, b_k \rangle$ pak vybereme ten, v jehož krajních bodech mají funkční hodnoty funkce f opačná znaménka.
Platí-li $f(a_k) \cdot f(x_k) < 0$, položíme $a_{k+1} = a_k, b_{k+1} = x_k$, platí-li $f(b_k) \cdot f(x_k) < 0$, položíme $a_{k+1} = x_k, b_{k+1} = b_k$. V případě, že $f(x_k) = 0$, našli jsem kořen rovnice a výpočet ukončíme.

Ve výpočtu pokračujeme tak dlouho, dokud nenarazíme na kořen, nebo dokud neplatí

$$|\, x_k - x_{k-1}| < \varepsilon,$$

kde $\varepsilon > 0$ je předem dané číslo. Splněním tohoto kriteria ale bohužel není zaručeno, že přesná hodnota kořene ξ se od jeho aproximace x_k liší o méně než ε. Chceme-li se přesvědčit, že $|\, x_k - \xi| < \varepsilon$, můžeme vypočítat $f(x_k + \varepsilon)$ a $f(x_k - \varepsilon)$. Platí-li $f(x_k) \cdot f(x_k + \varepsilon) < 0$, resp. $f(x_k) \cdot f(x_k - \varepsilon) < 0$, je jisté, že kořen ξ leží v intervalu $\langle x_k, x_k + \varepsilon \rangle$, resp. $\langle x_k - \varepsilon, x_k \rangle$, a tedy se od x_k nemůže lišit o více než ε.

Metoda regula falsi je vždy konvergentní (vždy najde kořen). Bývá rychlejší než půlení intervalu, ale existují případy, kdy je pomalejší.

Příklad 4.4 *Metodou regula falsi najděte kladný kořen rovnice z příkladu 4.1*

$$e^x + x^2 - 3 = 0$$

s přesností $\varepsilon = 0,01$.

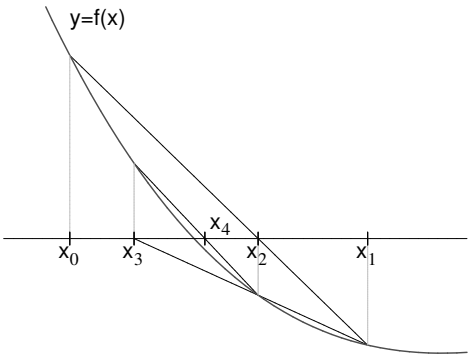
Řešení: Mohli bychom vyjít z intervalu nalezeného metodou půlení v příkladu 4.2, ale pro srovnání obou metod začneme opět s intervalem $\langle 0, 1\rangle$. U metody regula falsi budeme potřebovat i funkční hodnoty v bodech a_k, b_k a x_k, nejen jejich znaménka.

k	a_k	b_k	x_k	$f(a_k)$	$f(b_k)$	$f(x_k)$
0	0	1	0,73576	-2	0,71828	-0,37159
1	0,73576	1	0,82585	-0,37159	0,71828	-0,03414
2	0,82585	1	0,83375	-0,03414	0,71828	-0,00291

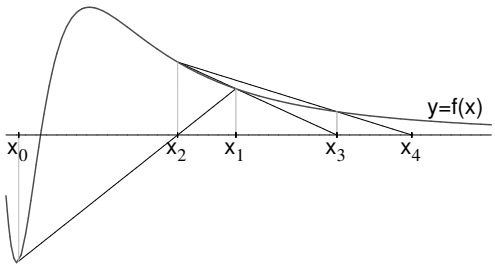
Platí $|x_2 - x_1| < 0,01$, proto výpočet ukončíme. Přibližné řešení rovnice je $x_2 \doteq 0,83$.

4.1.3 Metoda sečen

Metoda sečen je velmi podobná jako metoda regula falsi. Vyjdeme z intervalu $\langle a, b\rangle$ obsahujícího kořen rovnice. Označíme $x_0 = a$ a $x_1 = b$. Vedeme sečnu body $[x_0, f(x_0)]$ a $[x_1, f(x_1)]$ a najdeme její průsečík s osou x. Ten označíme x_2. Na rozdíl od metody regula falsi však nyní nevybíráme interval obsahující kořen, ale vedeme sečnu body $[x_1, f(x_1)]$, $[x_2, f(x_2)]$, její průsečík označíme x_3, pak vedeme sečnu body $[x_2, f(x_2)]$ a $[x_3, f(x_3)]$ atd. - viz obrázek 4.5.



Obrázek 4.5: Metoda sečen



Obrázek 4.6: Metoda sečen může divergovat.

V k-tém kroku metody počítáme aproximaci kořene podle vzorce

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}\, f(x_k), \tag{4.4}$$

kde $x_0 = a, x_1 = b$.

Výpočet ukončíme, když je splněna podmínka

$$|x_k - x_{k-1}| < \varepsilon,$$

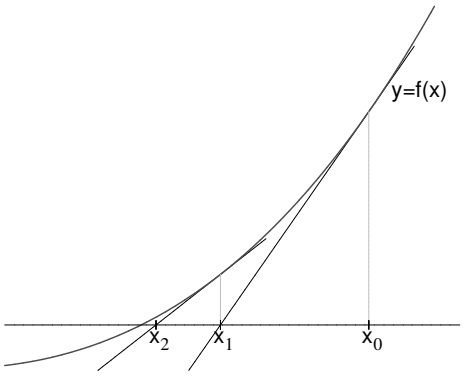
nebo když narazíme přímo na kořen rovnice.
Připomeňme, že daná podmínka nezaručuje, že platí $|x_k - \xi| < \varepsilon$.

Metoda sečen je rychlejší než metoda regula falsi, nemusí ale vždy konvergovat - viz obrázek 4.6.
Protože je obtížné předem zjistit, zda metoda pro danou rovnici konverguje nebo diverguje, je vhodné zadat při výpočtu maximální počet kroků. Je-li tento počet překročen a kořen rovnice jsme nenašli, výpočet ukončíme s tím, že metoda diverguje. Pak je nutno změnit počáteční aproximace nebo zvolit jinou metodu.

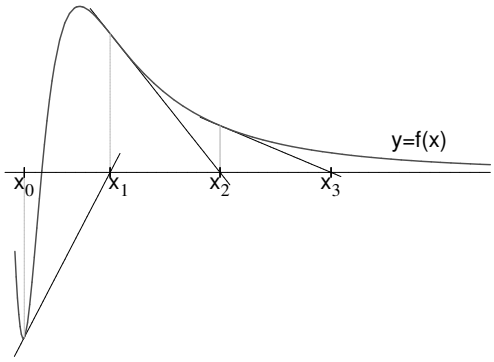
4.1.4 Newtonova metoda (metoda tečen)

Už sám název metody říká, že budeme pracovat s tečnami ke grafu funkce f. Proto všude v této kapitole budeme předpokládat, že funkce f má derivaci.

Newtonovu metodu můžeme popsat graficky takto:
Zvolíme počáteční aproximaci kořene x_0. Bodem $[x_0, f(x_0)]$ vedeme tečnu ke grafu funkce f. Její průsečík s osou x označíme x_1. Pak vedeme tečnu bodem $[x_1, f(x_1)]$, její průsečík s osou x označíme x_2 atd. - viz obrázek 4.7.



Obrázek 4.7: Newtonova metoda



Obrázek 4.8: Newtonova metoda může divergovat

Průsečík tečny v bodě $[x_k, f(x_k)]$ s osou x vypočteme jako

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{4.5}$$

Výpočet provádíme tak dlouho, dokud není splněna podmínka

$$|x_k - x_{k-1}| < \varepsilon$$

Při splnění této podmínky však nemusí platit $|x_k - \xi| < \varepsilon$. Kdybychom si chtěli být opravdu jisti, že se x_k od kořene ξ liší o méně než ε, mohli bychom použít dále uvedený odhad 4.6, případně vypočítat $f(x_k)$ a $f(x_k \pm \varepsilon)$ a použít postup popsaný u metody regula falsi.

Newtonovu metodu lze odvodit i pomocí Taylorova vzorce. Ukážeme nyní jak, protože stejný postup později zobecníme i pro soustavu rovnic.
Předpokládejme, že známe k-tou aproximaci řešení x_k. Pak můžeme psát

$$f(\xi) = f(x_k) + f'(x_k)\,(\xi - x_k) + R,$$

kde R je zbytek v Taylorově vzorci.
Zanedbáme-li tento zbytek a uvědomíme-li si že $f(\xi) = 0$ (protože ξ je kořenem rovnice $f(x) = 0$), můžeme z předchozí rovnice přibližně vyjádřit kořen ξ jako

$$\xi \doteq x_k - \frac{f(x_k)}{f'(x_k)},$$

což je právě x_{k+1} nalezené dříve popsaným způsobem.

Z Taylorova vzorce lze také odvodit odhady chyby k-té aproximace kořene získané Newtonovou metodou. Má-li funkce na intervalu I obsahujícím x_k i kořen ξ druhou derivaci, platí

$$|\,\xi - x_k| \;\leq\; \frac{M_2}{2m_1}\,(x_k - x_{k-1})^2 \tag{4.6}$$

$$|\,\xi - x_k| \;\leq\; \frac{M_2}{2m_1}\,(\xi - x_{k-1})^2\,, \tag{4.7}$$

kde $M_2 = \max|f''(x)|$ a $m_1 = \min|f'(x)|$ pro $x \in I$.

Newtonova metoda je z metod pro řešení nelineárních rovnice nejefektivnější, nemusí však konvergovat - viz obrázek 4.8.
Jestli Newtonova metoda konvergovat bude, nebo nebude, závisí do značné míry také na tom, jak zvolíme počáteční aproximaci x_0. Při pohledu na obrázek 4.7 je zřejmé, že zde byla počáteční aproximace zvolena vhodně. Kdybychom jako x_0 zvolili např. levý krajní bod zobrazeného intervalu, konvergence už by zaručena (ovšem ani vyloučena) nebyla. Tím se dostáváme k podmínkám, při jejichž splnění bude jisté, že Newtonova metoda konverguje.

Věta 4.2 *(Fourierova podmínka)*
Nechť v intervalu $\langle a, b\rangle$ leží jediný kořen rovnice $f(x) = 0$ a nechť $f'(x)$ a $f''(x)$ jsou spojité a nemění znaménko na intervalu $\langle a, b\rangle$. Zvolíme-li za počáteční aproximaci $x_0 \in \langle a, b\rangle$ tak, aby byla splněna podmínka

$$f(x_0) \cdot f''(x_0) > 0, \tag{4.8}$$

Newtonova metoda bude konvergovat.

Připomeňme v souvislosti s předpoklady věty 4.2 některé poznatky z prvního semestru. To, že $f'(x)$ nemění znaménko na intervalu $\langle a, b\rangle$, znamená, že funkce f buď na celém intervalu $\langle a, b\rangle$ roste, nebo na celém intervalu klesá.
To, že znaménko nemění $f''(x)$, znamená, že funkce f je buď na celém intervalu $\langle a, b\rangle$ konvexní (nad tečnou), nebo je na celém intervalu konkávní (pod tečnou).

Podmínka 4.8 znamená, že ze x_0 vybereme bod, v němž má funkční hodnota stejné znaménko jako druhá derivace.

Funkce, jejíž graf je na obrázku 4.7, je na celém zobrazeném intervalu rostoucí a konvexní. To znamená, že její druhá derivace je na tomto intervalu kladná. Proto se jako počáteční aproximace zvolil bod, v němž byla i funkční hodnota kladná.

Čtenář si může zkusit představit další možné situace, např. funkci na celém intervalu rostoucí a konkávní - zde by se jako x_0 zvolil levý krajní bod - a podobně.

Příklad 4.5 *Newtonovou metodou najděte záporný kořen rovnice z příkladu 4.1*

$$e^x + x^2 - 3 = 0$$

s přesností $\varepsilon = 0,01$.

Řešení: Víme, že kořen leží v intervalu $\langle -2, -1\rangle$. Ověříme, že na tomto intervalu jsou splněny předpoklady věty 4.2.
Vypočteme první a druhou derivaci funkce $f(x) = e^x + x^2 - 3$:

$$f'(x) = e^x + 2x \quad , \quad f''(x) = e^x + 2$$

Na celém intervalu $\langle -2, -1\rangle$ je $f'(x) < 0$ a $f''(x) > 0$ (tzn. ani první, ani druhá derivace zde nemění znaménko).
Nyní vybereme počáteční aproximaci x_0 tak, aby byla splněna podmínka 4.8. Protože $f(-2) = e^{-2} + 1 > 0$ a $f(-1) = e^{-1} - 2 < 0$, zvolíme $x_0 = -2$.
Další aproximace řešení budeme počítat pomocí iteračního vztahu

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{e^{x_k} + x_k^2 - 3}{e^{x_k} + 2x_k}$$

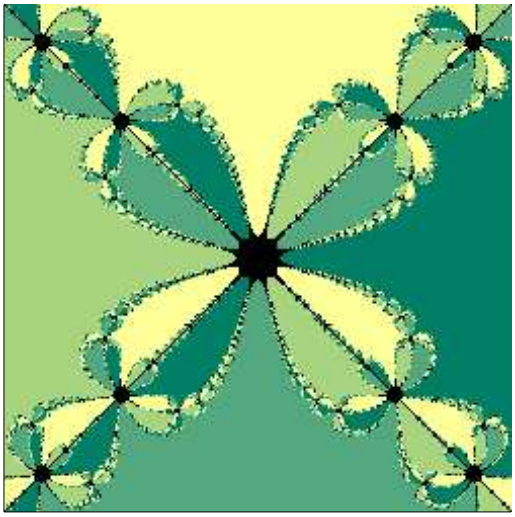
Dostaneme

$$\begin{aligned} x_0 &= -2 \\ x_1 &\doteq -1,70623 \\ x_2 &\doteq -1,67752 \\ x_3 &\doteq -1,67723 \end{aligned}$$

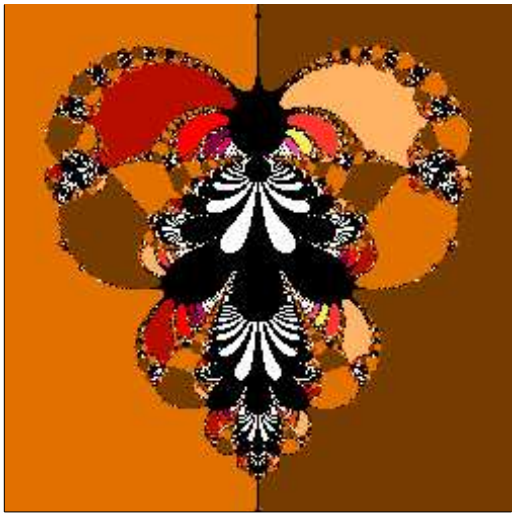
Nyní můžeme výpočet zastavit, protože $|x_3 - x_2| < 0,01$. Všimněme si, že tři kroky by nám stačily i pro dosažení přesnosti $0,001$. Newtonova metoda je obvykle velice rychlá.
Přibližné řešení rovnice je $x_3 \doteq -1,68$.

Nejsme-li schopni ověřit podmínky z věty 4.2, můžeme Newtonovu metodu přesto použít. Pokud tyto podmínky neplatí, Newtonova metoda konvergovat může a nemusí. Proto je při výpočtu vhodné stanovit maximální počet kroků metody a je-li překročen, výpočet ukončit a zvolit jinou počáteční aproximaci, resp. jinou metodu řešení.

Poznámka - Newtonova metoda pro komplexní kořeny



Obrázek 4.9: Fraktál vzniklý řešením rovnice $z^4 = 1$



Obrázek 4.10: Fraktál vzniklý řešením rovnice $z - e^z = 0$

Newtonovou metodou můžeme hledat i komplexní kořeny rovnice $f(z) = 0$. Postupuje se úplně stejně jako při hledání reálných kořenů, jenom je potřeba počítat s komplexními čísly. Zvlášť počáteční aproximaci z_0 je nutno zvolit komplexní, chceme-li dojít ke komplexnímu kořenu.
Podmínky, za kterých Newtonova metoda v komplexním oboru konverguje, jsou uvedeny např. v [**?**].
Zde se zmíníme o jednom zajímavém aspektu Newtonovy metody v komplexním oboru. Řešená rovnice $f(z) = 0$ může mít více kořenů. Na příklad rovnice $z^4 - 1 = 0$ má čtyři kořeny: $1, -1, i$ a $-i$. Který z nich pomocí Newtonovy metody najdeme, záleží na zvolené počáteční aproximaci z_0. Obarvíme-li v komplexní rovině všechny body, z nichž dojdeme k prvnímu kořenu, jednou barvou, všechny body, z nichž dojdeme k druhému kořenu, další barvou atd., dostaneme velmi zajímavý obrázek - fraktál.

4.1.5 Metoda prosté iterace

Metoda prosté iterace pro řešení jedné nelineární rovnice je další aplikací obecné metody postupných aproximací, popsané v kapitole 2.

Rovnici $f(x) = 0$ upravíme na tvar

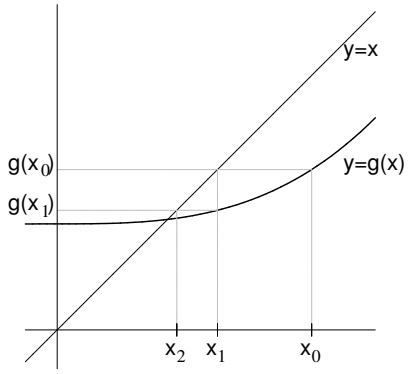
$$x = g(x).$$

Funkce g se nazývá **iterační funkce**. Nyní budeme místo kořene původní rovnice hledat pevný bod funkce $g(x)$. Uděláme to postupem uvedeným v kapitole 2.

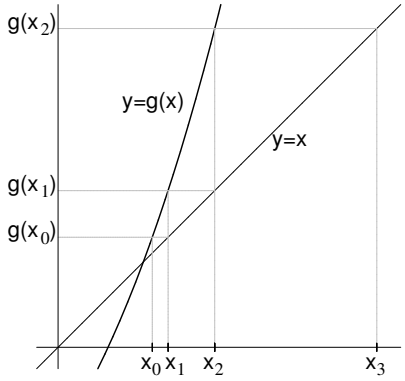
Zvolíme počáteční aproximaci x_0 a další aproximace pevného bodu (neboli řešení původní rovnice) budeme počítat jako

$$x_{k+1} = g(x_k) \tag{4.9}$$

Tímto způsobem můžeme a nemusíme dojít k pevnému bodu funkce g - viz obrázky 4.11 (kde se pevný bod najde) a 4.12 (kde metoda diverguje, i když počáteční aproximace byla pevnému bodu velmi blízko)



Obrázek 4.11: Metoda prosté iterace



Obrázek 4.12: Metoda prosté iterace může divergovat

Nyní řekneme, kdy je zaručeno, že metoda prosté iterace konverguje.
V kapitole 2 jsme se dozvěděli, že metoda postupných aproximací konverguje, je-li zobrazení, jehož pevný bod hledáme, kontraktivní. U funkce jedné proměnné kontraktivita úzce souvisí s rychlostí růstu této funkce - viz obrázky 2.4 a 2.5 v kapitole 2.3. Proto platí

Věta 4.3 *Nechť funkce g zobrazuje interval $\langle a, b\rangle$ do sebe a má na tomto intervalu derivaci. Jestliže existuje číslo $\alpha \in \langle 0, 1)$ tak, že*

$$|\, g'(x)| \leq \alpha \quad \forall x \in \langle a, b\rangle \ , \tag{4.10}$$

pak v intervalu $\langle a, b\rangle$ existuje pevný bod ξ funkce g a posloupnost postupných aproximací získaná předpisem 4.9 k němu konverguje pro libovolnou počáteční aproximaci $x_0 \in \langle a, b\rangle$. Dále platí

$$|\, x_k - \xi| \ \leq \ \frac{\alpha}{1-\alpha}\, |\, x_k - x_{k-1}| \tag{4.11}$$

$$|\, x_k - \xi| \ \leq \ \frac{\alpha^k}{1-\alpha}\, |\, x_1 - x_0| \tag{4.12}$$

Odhad 4.11 lze použít při rozhodování o zastavení iteračního procesu. Protože však ověření podmínky 4.10 a nalezení α může být obtížné, jako kriterium pro zastavení výpočtu se opět spíše používá podmínka

$$|\, x_k - x_{k-1}| < \varepsilon$$

(která opět nezaručuje, že $|\, x_k - \xi| < \varepsilon$).
Také je vhodné stanovit maximální počet kroků a je-li překročen, výpočet ukončit. Pak je potřeba zvolit jinou iterační funkci nebo jinou metodu.

Příklad 4.6 *Metodou prosté iterace najděte záporný kořen rovnice z příkladu 4.1*

$$e^x + x^2 - 3 = 0$$

s přesností $\varepsilon = 0,01$.

Řešení: Víme, že kořen leží v intervalu $\langle -2, -1\rangle$. Budeme hledat vhodnou iterační funkci g.
Jedna možnost, jak ze zadané rovnice vyjádřit x, je

$$x^2 = 3 - e^x \quad \Rightarrow \quad x = \pm\sqrt{3 - e^x}.$$

Protože hledáme záporný kořen, je

$$g(x) = -\sqrt{3 - e^x}$$

Ověříme, je-li splněna podmínka 4.10. K tomu je potřeba funkci g zderivovat. Dostaneme

$$g'(x) = \frac{e^x}{2\sqrt{3 - e^x}}\,.$$

Nyní budeme hledat maximum $|\,g'(x)|$ na intervalu $\langle -2, -1\rangle$. Na tomto intervalu je $|\,g'(x)| = \frac{e^x}{2\sqrt{3-e^x}}$. Derivace této funkce je $\frac{e^x(6-e^x)}{4(3-e^x)^{\frac{3}{2}}}$. To je funkce na intervalu $\langle -2, -1\rangle$ kladná, tedy $|\,g'(x)|$ je na tomto intervalu rostoucí a svého maxima nabývá v pravém krajním bodě tohoto intervalu. Hodnota maxima je $|\,g'(-1)| = \frac{e^{-1}}{2\sqrt{3-e^{-1}}} \leq 0.12 < 1$. To znamená, že podmínka 4.10 je splněna.
Ještě bychom měli ověřit, že funkce g zobrazuje interval $\langle -2, -1\rangle$ do sebe. Protože je na tomto intervalu $g'(x) > 0$, je funkce g rostoucí a stačí ověřit, že hodnoty g v krajních bodech intervalu do tohoto intervalu patří. (Kdyby g nebyla monotonní, museli bychom hledat její maximum a minimum na zkoumaném intervalu, nestačilo by dosadit krajní body.)
Protože $g(-2) \doteq -1,69 \in \langle -2, -1\rangle$ a $g(-1) \doteq -1,62 \in \langle -2, -1\rangle$, funkce g zobrazuje zkoumaný interval do sebe.
Konvergence iteračního procesu je tedy zaručena.
Můžeme zvolit např. $x_0 = -2$.
Další aproximace pak budeme počítat podle předpisu

$$x_{k+1} = g(x_k) = -\sqrt{3 - e^{x_k}}$$

Dostaneme

$$\begin{aligned} x_0 &= -2 \\ x_1 &\doteq -1,69253 \\ x_2 &\doteq -1,67808 \\ x_3 &\doteq -1,67728 \end{aligned}$$

Nyní můžeme výpočet zastavit, protože $|x_3 - x_2| < 0,01$. Iterační metoda v tomto případě konverguje docela rychle, protože hodnota $\alpha = 0,12$ je malá. Obecně platí, že čím je

derivace funkce g v absolutní hodnotě v okolí pevného bodu menší, tím rychleji metoda prosté iterace konverguje.
Přibližné řešení rovnice je $x_3 \doteq -1,68$

Jiná možnost, jak z rovnice vyjádřit x, je

$$x = \ln(3 - x^2) \quad \text{, tj.} \quad g(x) = \ln(3 - x^2).$$

V tomto případě by na intervalu $\langle -2, -1\rangle$ podmínky konvergence splněny nebyly. Podívejme se, jak se budou chovat postupné aproximace, zvolíme-li $x_0 = -1$:

$$\begin{aligned} x_0 &= -1 \\ x_1 &\doteq 0,69315 \\ x_2 &\doteq 0,92408 \\ x_3 &\doteq 0,76364 \\ x_4 &\doteq 0,88247 \\ &\vdots \end{aligned}$$

Nakonec bychom našli kladný kořen rovnice, který již jsme hledali metodou půlení a metodou regula falsi.

Poznámka. Způsobů, jak z rovnice $f(x) = 0$ vyjádřit x, je nekonečně mnoho. Jedna z možností je vydělit rovnici $f(x) = 0$ derivací funkce f, pak rovnici vynásobit -1 a nakonec na obě strany přičíst x. Dostaneme

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

vztah, který by nám měl být povědomý.
Newtonova metoda je tedy speciálním (a obvykle nejvhodnějším) případem metody prosté iterace.

4.2 Numerické metody řešení soustav nelineárních rovnic

Budeme se zabývat řešením soustavy n nelineárních rovnic o n neznámých

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \tag{4.13}$$

kterou můžeme přepsat vektorově jako

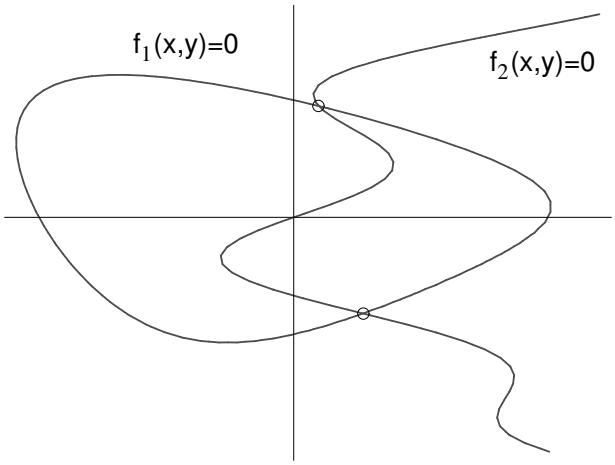
$$\mathbf{F}(\mathbf{x}) = \mathbf{o}, \tag{4.14}$$

kde $\mathbf{F} = (f_1, \ldots, f_n)^T$, $\mathbf{x} = (x_1, \ldots, x_n)^T$ a $\mathbf{o}$ je nulový vektor.

Přesné řešení této soustavy opět budeme značit $\xi = (\xi_1, \ldots, \xi_n)^T$.

Ukážeme zde metodu prosté iterace a Newtonovu metodu. Obě tyto metody vypadají velice podobně jako pro jedinou nelineární rovnici. Ve skutečnosti je ale vícedimenzionální případ mnohem složitější, protože na rozdíl od jediné rovnice je velmi nesnadné získat dobré informace o poloze kořene. Podmínky konvergence obou uvedených metod se také ověřují mnohem obtížněji než u jediné rovnice.

V případě, že řešíme dvě rovnice, hledáme vlastně průsečíky dvou křivek v rovině daných implicitně rovnicemi $f_1(x, y) = 0$ a $f_2(x, y) = 0$ - viz obrázek 4.13



Obrázek 4.13: Grafický význam řešení dvou nelineárních rovnic

4.2.1 Metoda prosté iterace

Soustavu 4.13 upravíme na tvar

$$\begin{aligned} x_1 &= g_1(x_1, x_2, \ldots, x_n) \\ x_2 &= g_2(x_1, x_2, \ldots, x_n) \\ &\vdots \\ x_n &= g_n(x_1, x_2, \ldots, x_n) \end{aligned} \tag{4.15}$$

což můžeme zapsat vektorově jako

$$\mathbf{x} = \mathbf{G}(\mathbf{x}), \tag{4.16}$$

kde $\mathbf{G} = (g_1, \ldots, g_n)^T$

Podobně jako u jedné rovnice zvolíme počáteční aproximaci $\mathbf{x}^{(0)}$ a počítáme posloupnost postupných aproximací z iteračního vztahu

$$\mathbf{x}^{(k+1)} = \mathbf{G}(\mathbf{x}^{(k)}) \tag{4.17}$$

Jsou-li funkce $g_1, \ldots, g_n$ diferencovatelné, lze vyslovit podmínky konvergence pro metodu prosté iterace, podobné těm z věty 4.3.
Protože pracujeme s n funkcemi n proměnných, v roli derivace zde bude vystupovat matice

$$\mathbf{G}' = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ & & \ddots & \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_n} \end{pmatrix}$$

Věta 4.4 *Nechť* $\mathbf{G}$ *zobrazuje uzavřenou oblast* D *do sebe a je v této oblasti diferencovatelná. Jestliže existuje číslo* $\alpha \in \langle 0, 1)$ *tak, že*

$$\| \mathbf{G}' \| \leq \alpha \quad \forall \mathbf{x} \in D\,, \tag{4.18}$$

kde $\| \mathbf{G}' \|$ *je řádková nebo sloupcová norma matice* $\mathbf{G}'$*, pak v oblasti* D *existuje pevný bod* ξ *zobrazení* G *a posloupnost postupných aproximací získaná předpisem 4.17 k němu konverguje pro libovolnou počáteční aproximaci* $\mathbf{x}^{(0)} \in D$*.*

Pro odhad chyby platí podobné vztahy jako 4.11, 4.12 u jedné rovnice.

Pro zastavení výpočtu se používá kriterium

$$\| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \| < \varepsilon,$$

kde $\| \cdot \|$ je některá z norem 2.13, 2.12.

Příklad 4.7 *Metodou prosté iterace najděte kořen soustavy rovnic*

$$\begin{aligned} 3x + x^2 y - 3 &= 0 \\ x^2 - 5y &= 0, \end{aligned}$$

který leží v oblasti $D = \langle 1/2; 1\rangle \times \langle 0; 1/2\rangle$ *s přesností* 0,01.

Řešení: Iteračních funkce mohou být např. $g_1(x, y) = 1 - \frac{x^2 y}{3}$, $g_2(x, y) = \frac{x^2}{5}$. Ověříme, zda jsou splněny podmínky konvergence.
$\mathbf{G} = (g_1, g_2)^T$ zobrazuje D do sebe: Jestliže $x \in \langle 1/2; 1\rangle$ a $y \in \langle 0; 1/2\rangle$, pak $\frac{x^2 y}{3} \in \langle 0; 1/6\rangle$ a tedy $g_1(x, y) \in \langle 5/6; 1\rangle \subseteq \langle 1/2; 1\rangle$. Podobně $g_2(x, y) \in \langle 1/20, 1/5\rangle \subseteq \langle 0; 1/2\rangle$.
Nyní ověříme, zda $\| \mathbf{G}' \|_\infty \leq \alpha < 1$, neboli zda $|\frac{\partial g_1}{\partial x}| + |\frac{\partial g_1}{\partial y}| \leq \alpha$ i $|\frac{\partial g_2}{\partial x}| + |\frac{\partial g_2}{\partial y}| \leq \alpha$.

$$\mathbf{G}' = \begin{pmatrix} -\frac{2xy}{3} & -\frac{x^2}{3} \\ \frac{2x}{5} & 0 \end{pmatrix}$$

Jestliže $x \in \langle 1/2; 1\rangle$ a $y \in \langle 0; 1/2\rangle$, pak $| -\frac{2xy}{3}| + | -\frac{x^2}{3}| \leq 1/3 + 1/3 = 2/3 < 1$ a $|\frac{2x}{5}| \leq 2/5 < 1$. (Tedy $\alpha = 2/3$.) Podmínky konvergence jsou splněny.

Jako počáteční aproximaci můžeme zvolit např. $(x_0, y_0) = (1, 0)$. Další aproximace pak budeme počítat podle vzorců

$$\begin{aligned} x_{k+1} &= g_1(x_k, y_k) = 1 - \frac{x_k^2 y_k}{3} \\ y_{k+1} &= g_2(x_k, y_k) = \frac{x_k^2}{5}. \end{aligned}$$

Postupně dostaneme

$$\begin{array}{llll} x_0 = 1 & & y_0 = 0 \\ x_1 = 1 & & y_1 = 0,2 \\ x_2 = 0,933 & & y_2 = 0,2 \\ x_3 = 0,942 & & y_3 = 0,174 \\ x_4 = 0,948 & & y_4 = 0,177. \end{array}$$

Protože $|x_4 - x_3| < 0,01$ i $|y_4 - y_3| < 0,01$, můžeme výpočet ukončit. Přibližné řešení soustavy je $x \doteq 0,95, y \doteq 0,18$.

Protože ověření podmínek konvergence může být dost problematické, je vhodné předem stanovit maximální počet kroků metody a je-li překročen, výpočet ukončit s tím, že metoda diverguje. Pak je potřeba zvolit jinou počáteční aproximaci, jiné iterační funkce, nebo jinou metodu.

Poznamenejme, že **najít vhodné iterační funkce může být velmi obtížné**.

4.2.2 Newtonova metoda

Předpokládejme, že již máme aproximaci řešení $\mathbf{x^{(k)}}$.
Podobně jako u diferencovatelné funkce jedné proměnné platilo pro x_k blízké ke kořeni ξ

$$f(\xi) \doteq f(x_k) + f'(x_k)(\xi - x_k),$$

platí pro n-tici diferencovatelných funkcí n proměnných $\mathbf{F} = (f_1, \ldots, f_n)^T$

$$\mathbf{F}(\xi) \doteq \mathbf{F}(\mathbf{x}^{(k)}) + \mathbf{F}'(\mathbf{x}^{(k)}) \cdot (\xi - \mathbf{x}^{(k)}),$$

kde

$$\mathbf{F}' = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\ & & \ddots & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \ldots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

a $\cdot$ značí násobení matic.

Uvědomíme-li si, že $\mathbf{F}(\xi) = \mathbf{o}$, můžeme odtud ξ přibližně vyjádřit, čímž získáme jeho další aproximaci $\mathbf{x}^{(k+1)}$. Dostaneme

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \left(\mathbf{F}'(\mathbf{x}^{(k)})\right)^{-1} \cdot \mathbf{F}(\mathbf{x}^{(k)}) \tag{4.19}$$

Při výpočtu další aproximace řešení vzorec 4.19 nepoužíváme. Museli bychom počítat inverzní matici, což je velmi pracné, zvlášť pro matice velkých rozměrů. Místo toho postupujeme následovně:
Vzorec 4.19 přepíšeme na tvar

$$\mathbf{F}'(\mathbf{x}^{(k)}) \cdot (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = -\mathbf{F}(\mathbf{x}^{(k)}) \,.$$

Označíme

$$\delta^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = (\delta_1^{(k)}, \ldots, \delta_n^{(k)})^T \tag{4.20}$$

a vyřešíme soustavu rovnic

$$\mathbf{F}'(\mathbf{x}^{(k)}) \cdot \delta^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)}) \tag{4.21}$$

s neznámými $\delta_1^{(k)}, \ldots, \delta_n^{(k)}$.

Řešíme-li dvě rovnice, hodí se pro řešení soustavy 4.21 Cramerovo pravidlo.
Máme-li velký počet rovnic, použijeme některou z dalších metod popsaných v kapitole 3.

Novou aproximaci řešení pak vypočteme z 4.20 jako

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta^{(k)} \,.$$

Ve výpočtu pokračujeme tak dlouho, dokud není splněna podmínka

$$\| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \| < \varepsilon \quad \text{, neboli} \quad \| \delta^{(k-1)} \| < \varepsilon,$$

nebo dokud není překročen předem stanovený maximální počet kroků (v takovém případě je nutno zvolit jinou počáteční aproximaci).

V každém kroku Newtonovy metody musíme vyřešit soustavu lineárních rovnic. Z toho je vidět, že Newtonova metoda je pracná a časově náročná. Na druhou stranu, začneme-li blízko kořene, konverguje obvykle velmi rychle.

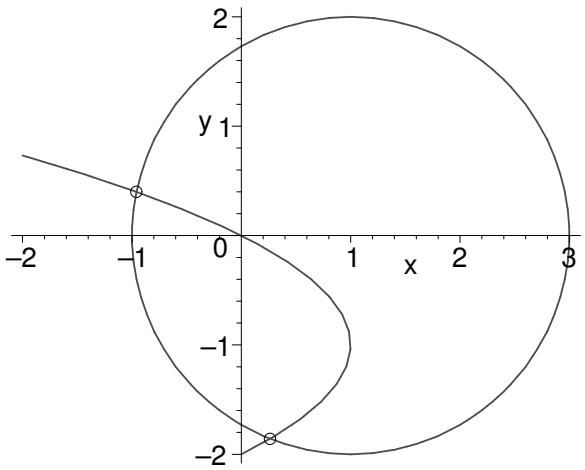
Příklad 4.8 *Newtonovou metodou najděte řešení soustavy rovnic*

$$\begin{aligned} (x-1)^2 + y^2 - 4 &= 0 \\ x + (y+1)^2 - 1 &= 0 \end{aligned}$$

s přesností $\varepsilon = 0,01$.

Řešení: Počet a polohu kořenů můžeme v tomto případě odhadnout graficky. První rovnice je rovnicí kružnice a druhá rovnice je rovnicí paraboly - viz obrázek 4.14.
Vidíme, že soustava má dvě řešení. Budeme hledat např. kořen ležící ve čtvrtém kvadrantu.
Jako počáteční aproximaci můžeme zvolit $\mathbf{x}^{(0)} = (0, -2)$.
Dále musíme vypočítat matici parciálních derivací funkcí

$$f_1(x,y) = (x-1)^2 + y^2 - 4 \quad , \quad f_2(x,y) = x + (y+1)^2 - 1$$



Obrázek 4.14: K příkladu 4.8 - odhad polohy kořenů.

Dostaneme

$$\mathbf{F}' = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 2(x-1) & 2y \\ 1 & 2(y+1) \end{pmatrix}$$

1. krok
Dosadíme bod $\mathbf{x}^{(0)} = (0, -2)$ do matice derivací a do funkcí f_1 a f_2:

$$\mathbf{F}'(0,-2) = \begin{pmatrix} -2 & -4 \\ 1 & -2 \end{pmatrix} \quad , \quad \mathbf{F}(0,-2) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Soustava rovnic pro neznámé δ_1 a δ_2 (horní index, označující krok, pro přehlednost vynecháme, je ale nutno mít na paměti, že v každém kroku budeme počítat jiné δ_1 a δ_2) bude

$$\begin{array}{rcrcr} -2\,\delta_1 & - & 4\,\delta_2 & = & -1 \\ \delta_1 & - & 2\,\delta_2 & = & 0 \end{array}$$

Snadno zjistíme, že řešením této soustavy je $\delta_1 = \frac{1}{4} = 0,25, \delta_2 = \frac{1}{8} = 0,125$.
Odtud $\mathbf{x}^{(1)} = (0 + 0,25\,; -2 + 0,125) = (0,25\,;\, -1,875)$.

2. krok

$$\mathbf{F}'(0,25\,;-1,875) = \begin{pmatrix} -1,5 & -3,75 \\ 1 & -1,75 \end{pmatrix} \quad , \quad \mathbf{F}(0,25\,;-1,875) \doteq \begin{pmatrix} 0,07812 \\ 0,01562 \end{pmatrix}$$

Budeme řešit soustavu

$$\begin{array}{rcrcr} -1,5\,\delta_1 & - & 3,75\,\delta_2 & = & -0,07812 \\ \delta_1 & - & 1,75\,\delta_2 & = & -0,01562 \end{array}$$

Řešení této soustavy můžeme najít pomocí Cramerova pravidla:

$$\delta_1 = \frac{\begin{vmatrix} -0,07812 & -3,75 \\ -0,01562 & -1,75 \end{vmatrix}}{\begin{vmatrix} -1,5 & -3,75 \\ 1 & -1,75 \end{vmatrix}} \doteq 0,01225 \quad , \quad \delta_2 = \frac{\begin{vmatrix} -1,5 & -0,07812 \\ 1 & -0,01562 \end{vmatrix}}{\begin{vmatrix} -1,5 & -3,75 \\ 1 & -1,75 \end{vmatrix}} \doteq 0,01593$$

Odtud $\mathbf{x}^{(2)} = (0,25 + 0,01225\,; -1,875 + 0,01593) = (0,26225\,; -1,85907)$.

3. krok

$$\mathbf{F}'(0,26225\,; -1,85907) \doteq \begin{pmatrix} -1,47549 & -3,71814 \\ 1 & -1,71814 \end{pmatrix}$$

$$\mathbf{F}(0,26225\,; -1,85907) \doteq \begin{pmatrix} 0,00040 \\ 0,00025 \end{pmatrix}$$

Budeme řešit soustavu

$$\begin{aligned} -1,47549\,\delta_1 &- 3,71814\,\delta_2 &&= -0,00040 \\ \delta_1 &- 1,71814\,\delta_2 &&= -0,00025 \end{aligned}$$

Řešení této soustavy pomocí Cramerova pravidla:

$$\delta_1 = \frac{\begin{vmatrix} -0,00040 & -3,71814 \\ -0,00025 & -1,71814 \end{vmatrix}}{\begin{vmatrix} -1,47549 & -3,71814 \\ 1 & -1,71814 \end{vmatrix}} \doteq -0,00004 \ , \ \delta_2 = \frac{\begin{vmatrix} -1,47549 & -0,00040 \\ 1 & -0,00025 \end{vmatrix}}{\begin{vmatrix} -1,47549 & -3,71814 \\ 1 & -1,71814 \end{vmatrix}} \doteq 0,00012$$

Odtud $\mathbf{x}^{(3)} = (0,26221\,; -1,85894)$.

Protože $|\,\delta_1| < 0,01$ i $|\,\delta_2| < 0,01$ (tj. $\|\,\delta\|_\infty < 0,01$), můžeme výpočet ukončit. Přibližné řešení je $(0,26\,; -1,86)$.

Shrnutí pojmů

Řešíme-li jednu nelineární rovnici $f(x) = 0$, musíme napřed zjistit, kolik má kořenů a kde. To nejlépe uvidíme z grafu funkce f - kořeny rovnice jsou body, v nichž graf protíná osu x. Jiná možnost, použitelná jen u některých rovnic, je převést rovnici na tvar $f_1(x) = f_2(x)$ a podívat se, kde se protínají grafy funkcí f_1 a f_2.
Víme-li už, kde zhruba kořen rovnice leží, můžeme jeho polohu upřesnit.
Je-li funkce f spojitá a jsou-li znaménka funkčních hodnot v bodech a, b opačná, je jisté, že v intervalu $\langle a, b\rangle$ leží kořen rovnice $f(x) = 0$.
Z tohoto faktu vychází metoda půlení intervalu a metoda regula falsi. U těchto metod začneme s intervalem $\langle a, b\rangle$ obsahujícím kořen a pak tento interval postupně zužujeme tak, aby další, užší, interval opět obsahoval kořen. U metody půlení intervalu nový interval získáme rozpůlením předchozího a ve výpočtu pokračujeme tak dlouho, dokud se interval dostatečně nezúží. U metody regula falsi je novým krajním bodem intervalu a zároveň novou aproximací kořene průsečík sečny vedené body $[a, f(a)]$, $[b, f(b)]$.

Ve výpočtu pokračujeme, dokud po sobě jdoucí aproximace kořene nejsou dostatečně blízké. Obě metody jsou vždy konvergentní. Jejich nevýhodou je, že jsou dosti pomalé, pro dosažení vysoké přesnosti je potřeba udělat velký počet kroků.
Z hlediska rychlosti je mnohem výhodnější Newtonova metoda. Zde zvolíme počáteční aproximaci x_0 a další aproximaci $x_{k+1}, k = 0, 1, \ldots$ najdeme vždy jako průsečík tečny ke grafu funkce f vedené bodem $[x_k, f(x_k)]$ s osou x. Počítáme tak dlouho, dokud po sobě jdoucí aproximace nejsou dostatečně blízké, nebo můžeme použít teoretický odhad chyby 4.6. Nevýhodou Newtonovy metody je, že nemusí kořen najít vždy, může divergovat. Konvergenci Newtonovy metody zaručí vhodná volba počáteční aproximace x_0 pomocí tzv. Fourierovy podmínky. Nejsme-li schopni nebo ochotni tuto podmínku použít, výpočet pomocí Newtonovy metody přesto stojí za pokus. Jen musíme počítat s možností divergence a omezit počet kroků, který se maximálně provede.
Metoda sečen má podobné vlastnosti jako Newtonova metoda - je vcelku rychlá, ale nemusí konvergovat. Místo průsečíku tečny s osou x zde v každém kroku hledáme bod x_{k+1}, kde se s osou x protne sečna vedená body $[x_{k-1}, f(x_{k-1})]$, $[x_k, f(x_k)]$, kde x_{k-1}, x_k jsou dvě předchozí aproximace kořene.
U metody prosté iterace rovnici napřed převedeme na tvar $x = g(x)$. Zvolíme x_0 a další aproximace počítáme opakovaným dosazováním do funkce g. Metoda prosté iterace nemusí vždy konvergovat a rychlost, s jakou případně najde kořen, závisí na volbě iterační funkce $g(x)$. Dá se říci, že obvykle je vhodnější Newtonova metoda, která je ostatně speciálním případem metody prosté iterace.
Řešení soustavy rovnic $\mathbf{F}(\mathbf{x}) = \mathbf{o}$ je náročnější než řešení jediné rovnice. Jednak samozřejmě kvůli objemu prováděných výpočtů, jednak zde mohou být značné problémy s nalezením přibližné polohy kořenů.
O metodě prosté iterace se dá říci v podstatě totéž, co u jediné rovnice - vše záleží na volbě iteračních funkcí.
Newtonova metoda pro soustavu je obvykle velmi rychlá, vyjdeme-li z bodu dostatečně blízkého ke kořeni. Je ale dosti pracná a nemusí vždy konvergovat. V každém kroku musíme vyřešit soustavu lineárních rovnic $\mathbf{F}'(\mathbf{x}^{(k)}) \cdot \delta^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})$, kde neznámé složky vektoru $\delta^{(k)}$ jsou přírůstky jednotlivých složek vektoru $\mathbf{x}^{(k)}$. Nová aproximace řešení se vypočte jako $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta^{(k)}$. Výpočet se provádí tak dlouho, dokud se všechny složky řešení neustálí na požadovaném počtu desetinných míst. Při výpočtu musíme počítat s možnou divergencí metody a omezit počet kroků, který se bude maximálně provádět.

4.3 Otázky a příklady ke cvičení

U následujících výroků rozhodněte, zda se jedná o výrok pravdivý či nepravdivý.
O funkci f v těchto otázkách vždy předpokládáme, že je spojitá, případně že má spojitou derivaci.

Otázka 4.1 *Jestliže pro funkci f, která je na intervalu $\langle a, b\rangle$ rostoucí, platí $f(a) \cdot f(b) < 0$, pak má rovnice $f(x) = 0$ v intervalu $\langle a, b\rangle$ právě jeden kořen.*

Otázka 4.2 *Jestliže platí $f(a) > 0$ i $f(b) > 0$, pak rovnice $f(x) = 0$ v intervalu $\langle a, b\rangle$ určitě nemá žádný kořen.*

Otázka 4.3 *Jestliže výchozí interval* $\langle a, b\rangle$ *obsahuje právě tři kořeny rovnice* $f(x) = 0$, *pak metodou půlení intervalů vždy najdeme prostřední z nich.*

Otázka 4.4 *Vyjdeme-li z intervalu obsahujícího právě jeden kořen, metodou půlení intervalu tento kořen určitě nalezneme.*

Otázka 4.5 *Zvolíme-li* x_0 *tak, že jeho vzdálenost od kořene* ξ *je nanejvýš* $0,01$, *Newtonovou metodou* ξ *určitě najdeme.*

Otázka 4.6 *Newtonova metoda je obvykle mnohem rychlejší než metoda půlení intervalů.*

Otázka 4.7 *Je-li funkce na intervalu* $\langle a, b\rangle$ *(obsahujícím kořen) rostoucí a konkávní (pod tečnou) a zvolíme-li* $x_0 = a$, *Newtonova metoda určitě bude konvergovat.*

Otázka 4.8 *Jacobiho metoda pro soustavu lineárních rovnic je speciálním případem metody prosté iterace pro soustavu rovnic.*

Otázka 4.9 *Newtonova metoda pro soustavu vždy konverguje.*

Otázka 4.10 *V každém kroku Newtonovy metody pro soustavu musíme vyřešit soustavu lineárních rovnic. Řešením této soustavy je nová aproximace řešení původní nelineární soustavy rovnic.*

Upozornění: Řeší-li se úloha, v níž se vyskytují goniometrické funkce, pomocí kalkulačky, je nutné mít **kalkulačku přepnutou na radiány (RAD)**, nikoli na stupně (DEG).

Příklad 4.1 *Zjistěte, kolik kořenů má rovnice* $\sin x - (x-2)^2 = 0$. *Najděte intervaly délky nejvýše 1, v nichž leží vždy právě jeden kořen. Největší kořen pak najděte metodou půlení intervalu s přesností* 0,1, *nejmenší metodou regula falsi s přesností* 0,01. *Ostatní kořeny hledejte metodou sečen s přesností* 0,001.

Příklad 4.2 *Newtonovou metodou najděte s přesností* 10^{-5} *záporný kořen rovnice* $x^4 + x - 3 = 0$. *Počáteční aproximaci zvolte podle Fourierovy podmínky.*

Příklad 4.3 *Metodou prosté iterace najděte s přesností* 0,01 *všechny kořeny rovnice* $2\ln x - x + 2 = 0$. *Pro každý kořen najděte vhodnou iterační funkci, ověřte, že jsou splněny podmínky konvergence.*
Pak některý z kořenů najděte s toutéž přesností Newtonovou metodou, porovnejte rychlost konvergence.

Příklad 4.4 *Najděte nejmenší kladný kořen rovnice* $\sin 2x = \cos 3x$ *s přesností* 10^{-5}. *Použijte libovolnou z probraných metod.*

Příklad 4.5 *S přesností* 10^{-2} *najděte bod, v němž funkce* $f(x) = x - \mathrm{e}^{x^2}$ *nabývá lokálního maxima. Použijte libovolnou z probraných metod.*

Příklad 4.6 *Newtonovou metodou najděte s přesností* 0,001 *kořen zadané soustavy rovnic. Vyjděte z bodu* $(x_0, y_0) = (1, 0)$.

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

Příklad 4.7 *Soustavu rovnic z předchozího příkladu řešte metodou prosté iterace. S přesností* 0,001 *najděte kořen, který leží v okolí bodu* $(1, 0)$.

Příklad 4.8 *Newtonovou metodou řešte zadanou soustavu rovnic. Proveďte jeden krok. Vyjděte z bodu* $(x_0, y_0, z_0) = (1, 0, 1)$.

$$\begin{aligned} x^2 + y^2 &= z^2 \\ x^2 + y^2 + z^2 &= 1 \\ 6x - 3y + 2z &= 1 \end{aligned}$$

Příklad 4.9 *Pomocí rovnice tečny ke grafu funkce odvoďte vztah pro výpočet další aproximace kořene Newtonovou metodou.*

Příklad 4.10 *Odvoďte vztah pro výpočet aproximace kořene metodou regula falsi.*

Odpovědi na otázky a řešení příkladů viz 9.4

Programovací úlohy

Zda budou funkce $f(x), g(x), f'(x)$ a pod. zadány přímo v programu, nebo se budou zadávat z klávesnice, ponecháme na zkušenosti a odvaze programátora.

Programovací úloha 1 Napište program, který najde kořen rovnice $f(x) = 0$ ležící v intervalu $\langle a, b\rangle$ s přesností ε

a) metodou půlení intervalu

b) metodou regula falsi

Programovací úloha 2 Napište program, který najde kořen rovnice $f(x) = 0$ s přesností ε Newtonovou metodou. Ošetřete i případ divergence metody.

Programovací úloha 3 Napište program, který najde kořen rovnice $f(x) = 0$ s přesností ε metodou prosté iterace. Ošetřete i případ divergence metody.

Programovací úloha 4 Napište program, který najde kořen soustavy rovnic $f_1(x, y) = 0$, $f_2(x, y) = 0$ s přesností ε metodou prosté iterace. Ošetřete i případ divergence metody.

Programovací úloha 5 Napište program, který najde kořen soustavy rovnic $f_1(x, y) = 0$, $f_2(x, y) = 0$ s přesností ε Newtonovou metodou. Ošetřete i případ divergence metody.

5 Aproximace funkcí

Cíl kapitoly

Čtenář se již určitě mnohokrát setkal s různými funkcemi a s výpočtem jejich hodnot. U některých funkcí se funkční hodnota vypočítá snadno, u jiných by to člověk „ručně" nezvládl a musí použít kalkulačku. Některé funkce jsou zadány tak složitým předpisem (viz část o statistice), že jejich hodnoty je jednodušší nalézt v tabulce, než je počítat. Někdy též máme funkci, která není zadána vůbec žádným předpisem, ale známe pouze její hodnoty v určitých bodech, např. získané nějakým měřením.
Naskýtá se otázka, jak zjistit hodnotu takové funkce v netabulkovém bodě, jak vypočítat hodnotu její derivace v určitém bodě nebo jak ji zintegrovat.
Řešením je nahradit zkoumanou funkci funkcí jinou, která se jí jakýmsi způsobem podobá a se kterou se lépe pracuje.
Cílem kapitoly o aproximaci je ukázat několik možností takovéto náhrady.
Nejčastěji „náhradní" funkcí bývá algebraický polynom, protože v tomto případě jsou všechny výše uvedené výpočty skutečně velmi jednoduché.
Požadavky, podle nichž vybíráme onu náhradní funkci, mohou být různé. Zde si blíže všimneme **interpolace**, kde se požaduje, aby aproximující funkce měla s funkcí původní v určitých bodech stejné hodnoty a **metody nejmenších čtverců**, kde má aproximující funkce procházet zadaným bodům v jistém smyslu nejblíže, ale přímo jimi procházet nemusí.

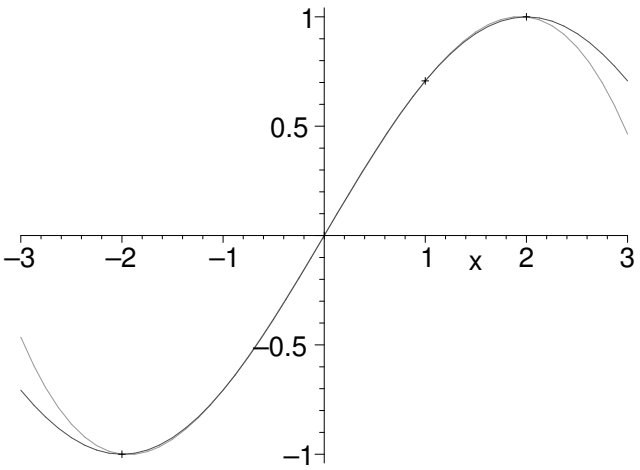
5.1 Interpolace algebraickými polynomy

Při interpolaci zní základní úloha takto: Máme $n+1$ navzájem různých bodů $x_0, x_1, \ldots, x_n$, kterým říkáme uzlové body nebo uzly interpolace a dále funkční hodnoty v těchto bodech $f_0 = f(x_0), f_1 = f(x_1), \ldots, f_n = f(x_n)$. Hledáme polynom $P_n(x)$ stupně nejvýše n takový, že v uzlových bodech nabývá týchž hodnot jako funkce f, tj. $P(x_i) = f_i, \quad i = 0, \ldots, n$.
Poznámka. Někdy se též hledá polynom, který má se zadanou funkcí nejen stejné funkční hodnoty v uzlových bodech, ale i stejné hodnoty derivací až do určitého řádu.

5.1.1 Existence a jednoznačnost interpolačního polynomu

Věta 5.1 *Nechť jsou dány body* $[x_i, f_i], i = 0, \ldots n$. *Pak existuje právě jeden polynom* P_n *stupně nanejvýš* n *takový, že* $P_n(x_i) = f_i, i = 0, \ldots n$.

Důkaz. Existenci interpolačního polynomu dokážeme tím způsobem, že předvedeme postup, kterým jej lze pro libovolné navzájem různé uzlové body zkonstruovat. Tomu bude věnován další odstavec této kapitoly.
To, že interpolační polynom procházející danými body existuje právě jeden, dokážeme sporem. Předpokládejme, že existují dva polynomy stupně nanejvýš n, označme je $P_n(x)$ a $R_n(x)$ takové, že $P_n(x_i) = f_i, i = 0, \ldots n$ i $R_n(x_i) = f_i, i = 0, \ldots n$. Ukážeme, že tyto dva polynomy jsou shodné. Za tím účelem označme $Q_n(x) = P_n(x) - R_n(x)$. Je vidět, že $Q_n(x)$ je opět polynom stupně nejvýše n a navíc $Q_n(x_i) = 0, i = 0, \ldots, n$. Máme tedy



Obrázek 5.1: Funkce a interpolační polynom

polynom stupně nejvýše n, který má $n+1$ kořenů. To je možné jedině tak, že $Q_n(x)$ je identicky roven nule, $Q_n(x) \equiv 0$ a tedy $P_n(x) \equiv R_n(x) \forall x \in \mathbb{R}$

5.1.2 Konstrukce interpolačního polynomu, Lagrangeův interpolační polynom

Interpolační polynom daný body $[x_i, f_i], i = 0, \dots n$ sestavíme pomocí polynomů $l_i(x)$ takových, že

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

Čtenář snadno ověří, že polynom

$$l_0(x) = \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)}$$

má v x_0 hodnotu 1 a v ostatních uzlových bodech hodnotu 0.
Podobně dostaneme i ostatní polynomy $l_i, i = 0, \dots n$:

$$l_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_0 - x_1)(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$

Interpolační polynom $L_n(x)$ nyní dostaneme snadno jako kombinaci $l_i(x)$:

$$\begin{aligned} L_n(x) &= f_0 l_0(x) + f_1 l_1(x) + \dots + f_n l_n(x) = \qquad (5.1) \\ &= f_0 \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)} + f_1 \frac{(x - x_0)(x - x_2) \dots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \dots (x_1 - x_n)} + \dots \\ &\quad \dots + f_n \frac{(x - x_0)(x - x_1) \dots (x - x_{n-1})}{(x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1})} \end{aligned}$$

Interpolační polynom ve tvaru 5.1 se nazývá **Lagrangeův interpolační polynom**.

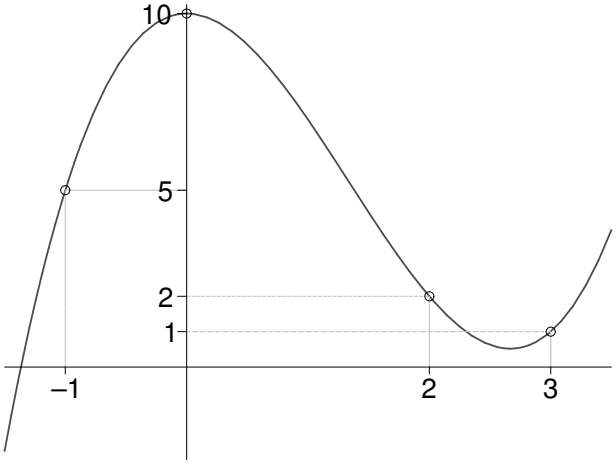
Příklad 5.1 *Najděte Lagrangeův interpolační polynom daný body*

x_i	-1	0	2	3
f_i	5	10	2	1

Řešení: Máme zadány 4 body, interpolační polynom bude tedy stupně nejvýše třetího. Pro jeho konstrukci použijeme vzorec 5.1:

$$\begin{aligned} L_3(x) &= 5\,\frac{(x-0)(x-2)(x-3)}{(-1-0)(-1-2)(-1-3)} + 10\,\frac{(x-(-1))(x-2)(x-3)}{(0-(-1))(0-2)(0-3)} + \\ &\quad +2\,\frac{(x-(-1))(x-0)(x-3)}{(2-(-1))(2-0)(2-3)} + 1\,\frac{(x-(-1))(x-0)(x-2)}{(3-(-1))(3-0)(3-2)} = x^3 - 4x^2 + 10 \end{aligned}$$

Výsledný interpolační polynom je spolu se zadanými body znázorněn na obrázku 5.2.



Obrázek 5.2: K příkladu 5.1: Zadané body a výsledný interpolační polynom

5.1.3 Newtonův interpolační polynom

Interpolační polynom v Lagrangeově tvaru má tu nevýhodu, že chceme-li přidat další uzlový bod, musíme celý polynom přepočítat znovu. Také výpočet hodnoty tohoto polynomu v určitém bodě je dosti pracný. Proto je někdy výhodnější hledat interpolační polynom v jiném tvaru než 5.1. Jako vhodný se ukazuje tvar

$$N_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}) \quad (5.2)$$

Koeficienty $a_0, a_1, \ldots, a_n$ lze získat řešením soustavy rovnic vzniklé rozepsáním podmínek $N_n(x_i) = f(x_i),\ i = 0, 1, \ldots n$, ale přehlednější a méně pracné je vypočítat tyto koeficienty pomocí takzvaných **poměrných diferencí**.

Pro danou funkci f a uzlové body $x_i,\ i = 0, \ldots, n$ nazveme podíly

$$f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i},\ i = 0, 1, \ldots n-1$$

poměrnými diferencemi prvního řádu
Pomocí poměrných diferencí prvního řádu definujeme poměrné diference druhého řádu jako

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i},\ i = 0, 1, \ldots, n-2$$

a obecně **poměrné diference k-tého řádu** pro $k \leq n$ definujeme takto:

$$f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i},\ i = 0, \ldots n-k.$$

Dá se dokázat, že pro koeficienty $a_i,\ i = 0, 1, \ldots, n$ v 5.2 platí

$$\begin{aligned} a_0 &= f(x_0) \\ a_1 &= f[x_0, x_1] \\ a_2 &= f[x_0, x_1, x_2] \\ &\vdots \\ a_n &= f[x_0, x_1, \ldots, x_n] \end{aligned}$$

Dosazením těchto hodnot do 5.2 dostaneme **Newtonův interpolační polynom**

$$\begin{aligned} N_n(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ &\quad \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\ldots(x - x_{n-1}) \end{aligned} \tag{5.3}$$

Poznámka. Newtonův interpolační polynom není vhodné upravovat roznásobováním. Pro rychlé dosazení se používá jiná úprava, kterou předvedeme v následujícím příkladu.

Příklad 5.2 *Aproximujte funkci* $f(x) = \frac{1}{x}$ *Newtonovým interpolačním polynomem v uzlech*

x_i	1	2	2,5	3,2	4

a pak pomocí něj vypočtěte přibližnou hodnotu funkce f *v bodech* $x = 3$ *a* $x = 10$.

Řešení: Abychom mohli sestavit Newtonův interpolační polynom, musíme vypočítat poměrné diference funkce f až do řádu 4. Budeme je postupně, po sloupcích, zapisovat do tabulky. Podtržené hodnoty pak použijeme pro interpolační polynom.

i	x_i	$f(x_i) = \frac{1}{x_i}$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, \ldots, x_{i+3}]$	$f[x_0, \ldots, x_4]$
0	1	<u>1</u>	<u>-0,5</u>	<u>0,2</u>	<u>-0,0625</u>	<u>0,015625</u>
1	2	0,5	-0,2	0,0625	-0,015625	
2	2,5	0,4	-0,125	0,03125		
3	3,2	0,3125	-0,078125			
4	4	0,25				

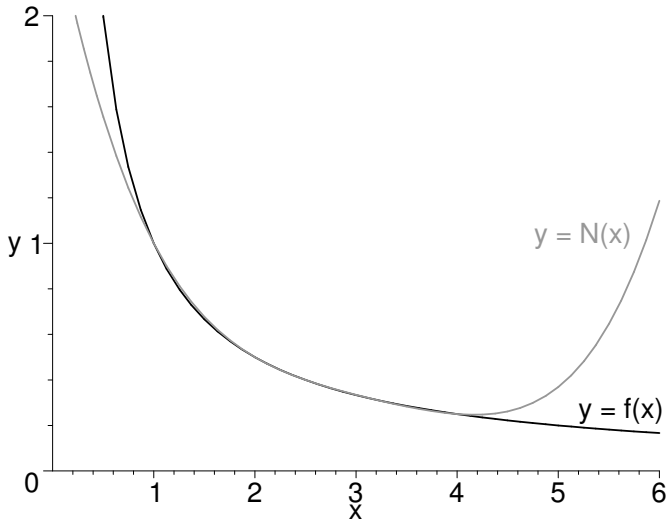
Nyní dosadíme do vzorce 5.3

$$\begin{aligned} N_4(x) &= 1 - 0,5(x-1) + 0,2(x-1)(x-2) - 0,0625(x-1)(x-2)(x-2,5) + \\ &\quad +0,015625(x-1)(x-2)(x-2,5)(x-3,2) \end{aligned}$$

Přibližnou hodnotu funkce f v bodě $x = 3$ vypočteme dosazením do interpolačního polynomu $N_4(x)$. Pro výpočet funkčních hodnot interpolačního polynomu v Newtonově tvaru je vhodné si tento polynom poněkud upravit. Můžeme vytknout $(x-1)$, pak ve zbytku $(x-2)$ a tak dále, až nakonec dostaneme

$$N_4(x) = 1 - (x-1)\Big(-0,5 + (x-2)\Big(0,2 + (x-2,5)(-0,0625 + (x-3,2)0,015625)\Big)\Big)$$

Dosazovat se hodí „zevnitř".
Při použití tohoto tvaru se značně sníží počet výpočetních operací nutných pro získání výsledku. Je-li čtenář obeznámen s Hornerovým schématem, možná najde jistou podobnost s tímto postupem.
V našem případě dostaneme $N_4(3) \doteq 0,334$, zatímco přesná hodnota je $\frac{1}{3} \doteq 0,333$.
Pro $x = 10$ vyjde $P_4(10) = 34.525$, zatímco přesná hodnota je $\frac{1}{10} = 0,1$.
Vidíme, že v bodě, který byl zhruba uprostřed uzlových bodů, je aproximace dobrá, hodnoty interpolačního polynomu a zadané funkce jsou blízké. Naopak v bodě, který leží daleko vně intervalu $\langle 1, 4\rangle$, je aproximace velmi špatná.
Situace je dobře patrná z obrázku 5.3, kde je vykreslen graf funkce f spolu s vypočteným interpolačním polynomem. Můžeme si všimnout, že na intervalu $\langle 1, 4\rangle$ interp. polynom dobře vystihuje chování funkce f, ale mimo tento interval se od sebe hodnoty funkce f a interpolačního polynomu značně liší.



Obrázek 5.3: K příkladu 5.2: Srovnání funkce a interpolačního polynomu

Poznámka. Bod $x = 10$ ležel vně intervalu ohraničeného nejmenším a největším uzlovým bodem. V takovém případě mluvíme o **extrapolaci**. Obecně je extrapolaci vhodné používat

pouze v bodech blízkých nejmenšímu nebo největšímu uzlovému bodu. O tom, čím je způsobena velká odchylka funkce a interpolačního polynomu v bodech vzdálených od uzlových bodů a jakou přesnost lze při interpolaci očekávat, pojednává kapitola 5.1.4.

Newtonův interpolační polynom pro ekvidistantní uzly

Jestliže vzdálenosti mezi sousedními uzlovými body jsou konstantní, tj. platí-li $x_{i+1} - x_i = h$ pro všechna $i = 1, \dots n$, kde $h \in \mathbb{R}$ je konstanta, říkáme, že uzly jsou **ekvidistantní**. Konstantu h nazýváme **krok**.
Všimněme si, že pro takovéto uzly platí

$$x_i = x_0 + ih, \quad i = 0, \dots, n. \tag{5.4}$$

Pro ekvidistantní uzly lze odvodit jiný, jednodušší tvar Newtonova (i Lagrangeova) interpolačního polynomu.

Místo poměrných diferencí budeme používat „obyčejné" diference:
Diference prvního řádu funkce $f(x)$ se definuje jako

$$\Delta f(x) = f(x + h) - f(x), \tag{5.5}$$

a **diference k-tého řádu** jako

$$\Delta^k f(x) = \Delta^{k-1} f(x + h) - \Delta^{k-1} f(x) \tag{5.6}$$

Pro ekvidistantní uzly $x_i, i = 0, \dots, n$, budeme diferenci k-tého řádu v uzlu x_i, $\Delta^k f(x_i)$, značit zkráceně jako $\Delta^k f_i$. Platí

$$\begin{aligned} \Delta f_i &= f(x_i + h) - f(x_i) = f(x_{i+1}) - f(x_i) = f_{i+1} - f_i \\ \Delta^k f_i &= \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i \end{aligned}$$

Poměrné diference lze v případě ekvidistantních uzlů vyjádřit pomocí obyčejných diferencí. Zřejmě platí

$$f[x_i, x_{i+1}] = \frac{\Delta f_i}{h}. \tag{5.7}$$

Pro poměrnou diferenci druhého řádu platí $f[x_i, x_{i+1}, x_{i+2}] = \frac{\frac{\Delta f_{i+1}}{h} - \frac{\Delta f_i}{h}}{2h} = \frac{\Delta^2 f_i}{2h^2}$.
Matematickou indukcí lze dokázat, že k-tá poměrná diference se dá vyjádřit jako

$$f[x_i, \dots, x_{i+k}] = \frac{\Delta^k f_i}{k!\, h^k}. \tag{5.8}$$

Tyto vztahy dosadíme do Newtonova interpolačního polynomu 5.3. Zjednodušit však můžeme i výrazy $(x - x_0) \cdots (x - x_k)$, které se v tomto polynomu vyskytují. K tomu účelu zavedeme místo x novou proměnnou q vztahem

$$q = \frac{x - x_0}{h}\,, \quad \text{neboli} \quad x = x_0 + qh. \tag{5.9}$$

Potom

$$x - x_0 = qh, \quad x - x_1 = x - x_0 - h = (q-1)h, \text{ obecně } x - x_k = (q-k)h \qquad (5.10)$$

Vztahy 5.8 a 5.10 nyní dosadíme do 5.3. Po snadné úpravě (zkrácení h) vyjde vzorec pro **Newtonův interpolační polynom pro ekvidistantní uzly**

$$\begin{aligned} N_n(x) &= f_0 + \frac{q}{1!}\Delta f_0 + \frac{q(q-1)}{2!}\Delta^2 f_0 + \cdots + \frac{q(q-1)\cdots(q-n+1)}{n!}\Delta^n f_0 \qquad (5.11) \\ q &= \frac{x - x_0}{h} \end{aligned}$$

Chceme-li do interpolačního polynomu $N_n(x)$ dosadit za x určité číslo, vypočteme příslušnou hodnotu q, a tu pak do N dosadíme. Není vhodné výraz roznásobovat, pro dosazování je lepší úprava, kterou předvedeme v příkladu 5.3.

Příklad 5.3 *Pomocí Newtonova interpolačního polynomu vypočtěte přibližnou hodnotu Gaussovy funkce*

$$G(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2} \mathrm{d}t$$

v bodě $x = 1{,}17$, *známe-li hodnoty* $G(x)$ *v následujících bodech:*

x	*1*	*1,1*	*1,2*	*1,3*
$G(x)$	*0,8427*	*0,8802*	*0,9103*	*0,9340*

Řešení: Vypočteme potřebné diference, dvakrát podtržená čísla jsou použita v interpolačním polynomu:

i	x_i	G_i	ΔG_i	$\Delta^2 G_i$	$\Delta^3 G_i$
0	1	0,8427	0,0375	-0,0074	0,0010
1	1,1	0,8802	0,0301	-0,0064	
2	1,2	0,9103	0,0237		
3	1,3	0,9340			

Nyní dosadíme do vzorce 5.11:

$$\begin{aligned} N_3(x) &= 0{,}8427 + \frac{q}{1!}\cdot 0{,}0375 - \frac{q(q-1)}{2!}\cdot 0{,}0074 + \frac{q(q-1)(q-2)}{3!}\cdot 0{,}0010 \\ q &= \frac{x-1}{0{,}1} \end{aligned}$$

Pro snížení počtu potřebných početních operací můžeme tento polynom upravit (podobně, jako jsme to udělali v příkladu 5.2):

$$N_3(x) = 0{,}8427 + \frac{q}{1}\left(0{,}0375 + \frac{q-1}{2}\left(-0{,}0074 + \frac{q-2}{3}\cdot 0{,}0010\right)\right)$$

Nyní chceme vypočítat přibližnou hodnotu $G(1{,}17)$. Ta bude přibližně rovna hodnotě interpolačního polynomu pro $x = 1{,}17$, tzn. pro $q = \frac{1{,}17-1}{0{,}1} = 1{,}7$:

$$G(1{,}17) \doteq N_3(1{,}17) = 0{,}8427 + \frac{1{,}7}{1}\left(0{,}0375 + \frac{0{,}7}{2}\left(-0{,}0074 + \frac{-0{,}3}{3}\cdot 0{,}0010\right)\right) \doteq 0{,}9020$$

Přesná hodnota $G(1{,}17)$ je po zaokrouhlení na čtyři desetinná místa také $0{,}9020$.

5.1.4 Odhad chyby

Věta 5.2 *Nechť interval I obsahuje body $x_0, x_1, \ldots, x_n$ a nechť f je (n+1)-krát diferencovatelná funkce na I. Nechť $P_n(x)$ je interpolační polynom n-tého stupně určený hodnotami funkce f v bodech $x_0, \ldots x_n$. Potom pro libovolné $x \in I$ existuje $\xi \in I$ takové, že pro chybu interpolace $E(x)$ platí*

$$E(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n). \tag{5.12}$$

Důkaz není úplně jednoduchý a lze jej nalézt např. v [**?**].

Vzorec 5.12 slouží hlavně jako teoretický základ pro určení chyby u dalších metod, např. u numerické integrace. Jinak je jeho použití poněkud problematické, protože bod ξ je pro každé $x \in I$ jiný a jeho nalezení je prakticky nemožné. Chybu interpolace však můžeme alespoň shora odhadnout:
Označíme-li $M_{n+1} = \max_{t \in I} |f^{(n+1)}(t)|$, platí

$$|E(x)| = |f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |(x - x_0)(x - x_1)\ldots(x - x_n)| \tag{5.13}$$

Najít veličinu M_{n+1} však také nemusí být zrovna jednoduché.

Poznámka. Odhad 5.13 lze použít např. v případě, kdy chceme sestavit tabulku hodnot nějaké funkce $f(x)$ s konstantním krokem mezi hodnotami x a ptáme se, jak tento krok zvolit, aby chyba např. při interpolaci lineárním polynomem nepřevýšila dané ε.

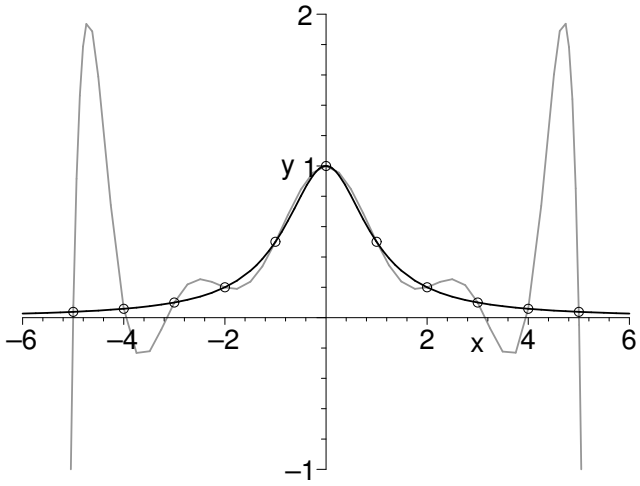
Příklad 5.4 *Odhadněte chybu interpolace z příkladu 5.2 v bodě $x = 3$.*

Poznámka. Tento příklad slouží spíše k ozřejmení jednotlivých veličin ve vzorci 5.13 a jako ukázka, že vzorec „funguje", protože v tomto případě můžeme určit i přesnou hodnotu chyby a nemusíme nic odhadovat.

Řešení: Pro odhad chyby potřebujeme vypočítat pátou derivaci interpolované funkce $f(x) = \frac{1}{x}$ (protože n je v tomto případě 4) a najít maximum její absolutní hodnoty na intervalu $I = \langle 1, 4\rangle$ (I je nejmenší interval obsahující všechny uzlové body a bod, v němž chceme odhadovat chybu).
Vyjde $f^{(5)}(x) = -\dfrac{120}{x^6}$
Je vidět, že $|f^{(5)}(x)| = \frac{120}{x^6}$, což je funkce na I klesající. Svého maxima na tomto intervalu proto dosahuje v bodě $x = 1$ a jeho hodnota je $M_5 = \frac{120}{1^6} = 120$.
Nyní dosadíme do 5.12:
$|E(3)| \leq \frac{120}{5!} |(3-1)(3-2)(3-2,5)(3-3,2)(3-4)| = |2 \cdot 1 \cdot 0,5 \cdot (-0,2) \cdot (-1)| = 0,2$
Odhad chyby je v tomto případě dosti nadsazený, chyba v bodě $x = 3$ je ve skutečnosti mnohem menší než $0,2$, viz řešení příkladu 5.2
To, že teoretický odhad chyby je příliš pesimistický, je poměrně časté i u jiných metod.

V bodech vzdálených uzlovým bodům nabývá výraz $(x - x_0)(x - x_1)\ldots(x - x_n)$, který se vyskytuje v odhadu chyby, velkých hodnot. Proto se interpolační polynom pro výpočet přibližných hodnot funkce v takovýchto bodech nehodí.

Aproximace ale v některých případech nemusí být dobrá ani v bodech relativně blízkých uzlovým bodům. To ilustruje obrázek 5.4, na němž je graf funkce $f(x) = \frac{1}{1+x^2}$ a interpolační polynom daný vyznačenými uzlovými body.



Obrázek 5.4: Nevhodná aproximace interpolačním polynomem

Situace by se příliš nezlepšila, ani kdybychom přidali více uzlových bodů.
Zde je velká odchylka funkce a polynomu taktéž způsobena velkými hodnotami součinu $(x - x_0)(x - x_1) \dots (x - x_n)$, především poblíž konců interpolačního intervalu.

Proto je někdy vhodné nenahrazovat funkci, zvláště chceme-li ji aproximovat na delším intervalu, jedním interpolačním polynomem, ale interval rozdělit na malé části a na každé z nich funkci nahradit polynomem nízkého stupně. To bude námětem následující kapitoly.

5.2 Interpolace pomocí splajnů

Základní myšlenka interpolace pomocí splajnů je obdobná jako u Lagrangeovy interpolace. Máme zadány uzlové body $a = x_0 < x_1 < \cdots < x_n = b$ a funkční hodnoty v nich, které označíme $f_0, f_1, \dots, f_n$. Stejně jako předtím hledáme funkci $S(x)$ takovou, že platí $S(x_i) = f_i,\ i = 0, 1, \dots, n$, ale tentokrát je funkce $S(x)$ po částech polynom (obecně na každém intervalu $\langle x_i, x_{i+1} \rangle$, $i = 0, 1, \dots n-1$, jiný) a splňuje určité požadavky hladkosti (tj. spojitosti derivací).

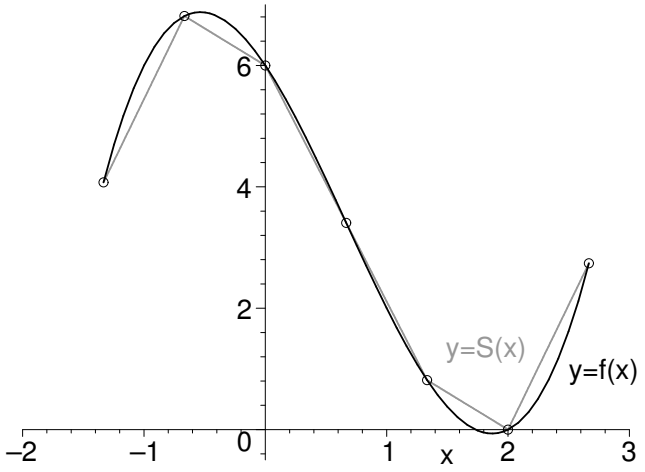
Konkrétně **splajnem řádu** k pro uzly $a = x_0 < x_1 < \cdots < x_n = b$ rozumíme funkci, která je v každém intervalu $\langle x_i, x_{i+1} \rangle$, $i = 0, \dots n-1$, polynom stupně k a která má v celém intervalu $\langle a, b \rangle$ spojité derivace až do řádu $k-1$ včetně.

Poznámka. Slovo „splajn" pochází z anglického „spline", což znamená pružné konstruktérské pravítko. V české literatuře se někdy píše splajn a někdy spline.

Nejjednodušším příkladem je splajn řádu 1, **lineární splajn**. Funkce je na každém subintervalu $\langle x_i, x_{i+1}\rangle$, $i = 0, \ldots n-1$, nahrazena úsečkou, jejíž rovnice je

$$S_i(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}\,(x - x_i), \quad x \in \langle x_i, x_{i+1}\rangle$$

U splajnu 1. řádu požadujeme spojitost derivací do řádu 0 včetně, tj. spojitost samotné funkce $S(x)$. Snadno se přesvědčíme, že hodnoty jednotlivých funkcí $S_i(x)$ v krajních bodech příslušného intervalu $\langle x_i, x_{i+1}\rangle$ jsou rovny $f(x_i)$, resp. $f(x_{i+1})$, čímž je zaručeno, že na sebe tyto funkce v uzlových bodech spojitě navazují (viz obrázek 5.5). Zlepšení aproximace dosáhneme zjemněním intervalů mezi uzlovými body.



Obrázek 5.5: Nahrazení funkce lineárním splajnem

Nejčastěji užívané jsou tzv. **kubické splajny**, kdy k=3.

Definice a konstrukce kubického splajnu

Kubický splajn pro funkci f s uzlovými body $x_0, x_1, \ldots, x_n$ je funkce $S(x)$, která je kubický polynom označený $S_i(x)$ na každém subintervalu $\langle x_i, x_{i+1}\rangle$, $i = 0, 1, \ldots, n-1$, vyhovuje podmínkám

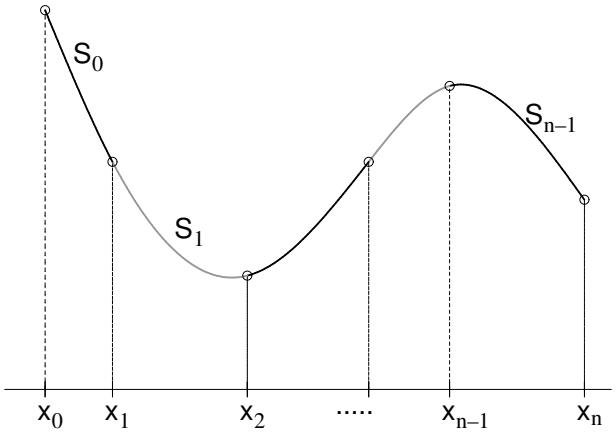
$$S_i(x_i) = f(x_i), \qquad i = 0, \ldots, n \tag{5.14}$$
$$S_i(x_{i+1}) = S_{i+1}(x_{i+1}), \quad i = 0, \ldots, n-2 \tag{5.15}$$
$$S_i'(x_{i+1}) = S_{i+1}'(x_{i+1}), \quad i = 0, \ldots, n-2 \tag{5.16}$$
$$S_i''(x_{i+1}) = S_{i+1}''(x_{i+1}), \quad i = 0, \ldots, n-2 \tag{5.17}$$

a okrajovým podmínkám a), b) nebo c)

a) $S''(x_0) = S''(x_n) = 0$
b) $S''(x_0) = f_0'',\ S''(x_n) = f_n''$
c) $S'(x_0) = f_0',\ S'(x_n) = f_n'$

(f_0'', f_n'', f_0' a f_n' jsou předem zadané konstanty).

Podmínky 5.15 znamenají spojitost funkce S v uzlových bodech, podmínky 5.16 a 5.17 spojitost prvních, resp. druhých derivací.



Obrázek 5.6: Přirozený kubický splajn

Kubický splajn splňující okrajové podmínky a) se nazývá **přirozený kubický splajn**.

Na obrázku 5.7 je znázorněna aproximace funkce $f(x) = \frac{1}{1+x^2}$ pomocí přirozeného kubického splajnu. Můžeme porovnat s obrázkem 5.4, kde byla tatáž funkce nahrazena interpolačním polynomem daným stejnými uzlovými body.

Nyní se budeme zabývat problémem, jak k zadaným uzlovým bodům a hodnotám funkce v nich sestrojit přirozený kubický splajn. (Splajn vyhovující jiným okrajovým podmínkám by se našel podobně.)

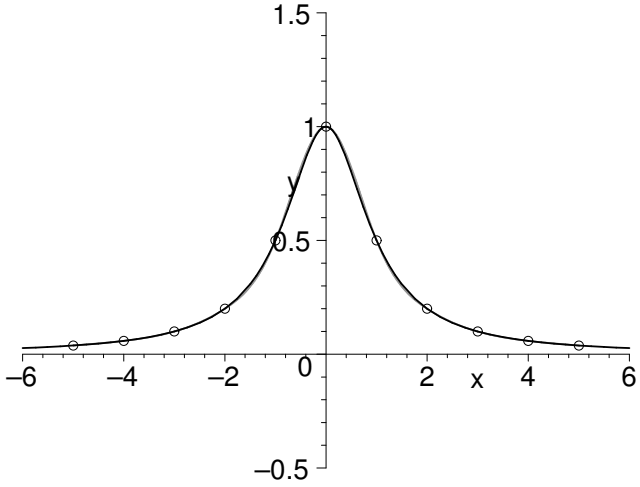
Na jednotlivých intervalech $\langle x_i, x_{i+1}\rangle$, $i = 0, 1, \ldots, n-1$, budeme splajn hledat ve tvaru

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Z podmínek 5.14 dostaneme $a_i = f(x_i), i = 0, 1, \ldots n-1$. Odtud, z podmínek 5.15, 5.16, 5.17 a z okrajových podmínek $S_0''(x_0) = S_{n-1}''(x_n) = 0$ lze po jistém úsilí odvodit soustavu rovnic s neznámými $c_i,\ i = 0, \ldots, n$

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\left(\frac{\Delta f_i}{h_i} - \frac{\Delta f_{i-1}}{h_{i-1}}\right), \quad i = 1, \ldots, n-1 \quad (5.18)$$

$$c_0 = c_n = 0$$



Obrázek 5.7: Nahrazení funkce $f(x) = \frac{1}{1+x^2}$ přirozeným kubickým splajnem.

kde $h_i = x_{i+1} - x_i$ a $\Delta f_i = f(x_{i+1}) - f(x_i),\ i = 0, \ldots, n-1$.
Po rozepsání a dosazení za c_0 a c_n soustava vypadá takto:

$$\begin{array}{rcrcrcl} 2(h_0+h_1)c_1 & + & h_1c_2 & & & = & 3(\frac{\Delta f_1}{h_1} - \frac{\Delta f_0}{h_0}) \\ h_1c_1 & + & 2(h_1+h_2)c_2 & + & h_2c_3 & = & 3(\frac{\Delta f_2}{h_2} - \frac{\Delta f_1}{h_1}) \\ & & & \ddots & & \vdots & \\ & & h_{n-2}c_{n-2} & + & 2(h_{n-2}+h_{n-1})c_{n-1} & = & 3(\frac{\Delta f_{n-1}}{h_{n-1}} - \frac{\Delta f_{n-2}}{h_{n-2}}) \end{array} \tag{5.19}$$

Jedná se o třídiagonální soustavu rovnic a lze ji vyřešit např. pomocí Gaussovy eliminační metody přizpůsobené pro třídiagonální soustavu.

Koeficienty b_i a d_i pak dopočítáme pomocí c_i ze vztahů (také odvozených z podmínek 5.14 – 5.17)

$$b_i = \frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{c_{i+1} + 2c_i}{3} h_i \qquad i = 0, \ldots, n-1 \tag{5.20}$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \qquad i = 0, \ldots, n-1 \tag{5.21}$$

Příklad 5.5 *Funkci $f(x) = \sqrt{x}$ aproximujte přirozeným kubickým splajnem s uzlovými body*

x_i	1	1,69	2,25	2,89	4

a pak pomocí tohoto splajnu vypočtěte přibližně hodnotu $f(2)$.

Řešení: Dopočítáme funkční hodnoty v uzlových bodech a pak vypočteme $h_i, i = 0, 1, 2, 3$, tj. délky jednotlivých intervalů, a $\Delta f_i, i = 0, 1, 2, 3$. Vypočtené hodnoty jsou zapsány v následující tabulce

i	0	1	2	3	4
x_i	1	1,69	2,25	2,89	4
$f(x_i)=\sqrt{x_i}$	1	1,3	1,5	1,7	2
h_i	0,69	0,56	0,64	1,11	
Δf_i	0,3	0,2	0,2	0,3	

Víme, že $c_0 = 0$. Pro neznámé c_1, c_2, c_3 dostaneme podle 5.19 soustavu rovnic

$$\begin{aligned} 2,5c_1 &+ 0,56c_2 &&= -0,232919 \\ 0,56c_1 &+ 2,4c_2 &+ 0,64c_3 &= -0,133929 \\ &0,64c_2 &+ 3,5c_3 &= -0,126689 \end{aligned}$$

Řešením této soustavy je $c_1 = -0,087085,\ c_2 = -0,027155,\ c_3 = -0,031231$.
Koeficienty b_i a $d_i, i = 0,1,2,3$, dopočítáme podle vzorců 5.20 a 5.21. (Při výpočtu b_3 a d_3 použijeme $c_4 = 0$.)
Tedy např. $b_0 = \dfrac{1,3-1}{0,69} - \dfrac{-0,087085+2\cdot 0}{3}\cdot 0,69 \doteq 0,454812$
Ostatní koeficienty by se vypočítaly podobně. Vyjde:

i	0	1	2	3
a_i	1	1,3	1,5	1,7
b_i	0,454812	0,394724	0,330749	0,293381
c_i	0	-0,087085	-0,027155	-0,031231
d_i	-0,042070	0,035672	-0,002123	0,009379

Výsledný přirozený kubický splajn je tedy

$$S(x) = \begin{cases} S_0(x)=1+0{,}454812(x-1)-0{,}042070(x-1)^3 & x\in\langle 1\,;\,1{,}69\rangle \\ S_1(x)=1{,}3+0{,}394724(x-1{,}69)-0{,}087085(x-1{,}69)^2+0{,}035672(x-1{,}69)^3 & x\in\langle 1{,}69\,;\,2{,}25\rangle \\ S_2(x)=1{,}5+0{,}330749(x-2{,}25)-0{,}027155(x-2{,}25)^2-0{,}002123(x-2{,}25)^3 & x\in\langle 2{,}25\,;\,2{,}89\rangle \\ S_3(x)=1{,}7+0{,}293381(x-2{,}89)-0{,}031231(x-2{,}89)^2+0{,}009379(x-2{,}89)^3 & x\in\langle 2{,}89\,;\,4\rangle \end{cases}$$

Přibližnou hodnotu funkce f v bodě $x = 2$ nyní vypočteme jako $S_1(2) \doteq 1,415058$ (protože $2 \in \langle 1,69\,;\,2,25\rangle$). Pro srovnání, přesná hodnota je $\sqrt{2} \doteq 1,414214$.

5.3 Metoda nejmenších čtverců

V předchozích částech této kapitoly jsme požadovali, aby interpolační polynom, resp. splajn, nabýval v uzlových bodech stejných hodnot jako funkce, již se snažíme aproximovat. V případě, že jsou funkční hodnoty získány experimentálně, např. jako výsledky nějakého měření, je interpolace nevhodná. Výsledky jsou totiž zatíženy chybami a interpolační funkce by tyto chyby kopírovala, což je přesně to, čeho se chceme vyvarovat. Kromě toho povaha experimentů nevylučuje možnost několika měření při nezměněné hodnotě x, tj. nemusí být všechny uzlové body navzájem různé. Vzhledem k těmto okolnostem není dobré požadovat, aby aproximační funkce nabývala v uzlových bodech předem daných hodnot. V mnoha případech máme určitou představu o povaze funkce, jejíž hodnoty jsme naměřili, např. může se jednat o lineární nebo kvadratickou závislost. Pak hledáme mezi všemi funkcemi tohoto známého typu takovou, která prochází k zadaným bodům v jistém smyslu nejblíže.

Formulace problému

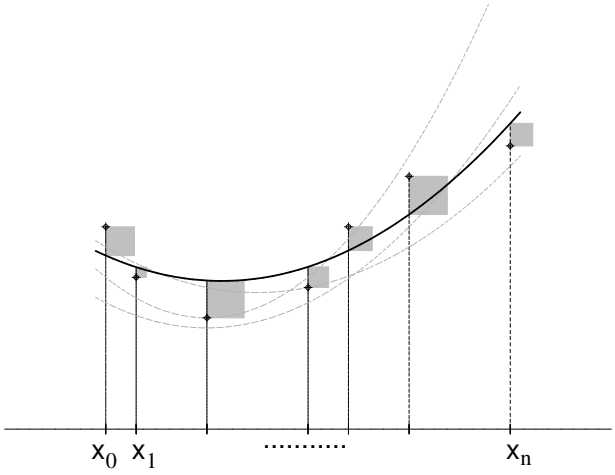
Jsou dány body $x_i,\ i = 0, \dots n$ a funkční hodnoty v nich y_i. Dále jsou dány funkce $\varphi_i, i = 0, \dots, m,\ m < n$. Mezi všemi funkcemi tvaru

$$P_m(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \cdots + c_m\varphi_m(x), \tag{5.22}$$

$c_0, \dots, c_m$ jsou reálná čísla, hledáme takovou, pro niž veličina

$$\rho^2(c_0, \dots c_m) = \sum_{i=0}^{n}(y_i - P_m(x_i))^2,$$

kterou nazýváme **kvadratická odchylka**, nabývá minimální hodnoty.
(Kvadratická odchylka ρ^2 udává součet obsahů čtverců se stranami o délkách $|y_i - P_m(x_i)|$, $i = 0, \dots, n$, viz obrázek 5.8. Odtud pochází název metody.)
Takovou funkci pak nazýváme **nejlepší aproximací** experimentálních dat $y_0, \dots y_n$ v dané třídě funkcí **ve smyslu metody nejmenších čtverců**.



Obrázek 5.8: Metoda nejmenších čtverců - mezi všemi funkcemi známého typu 5.22 (zde jsou to paraboly) hledáme tu, pro kterou je součet obsahů čtverců nejmenší možný.

Nalezení nejlepší aproximace

Protože body $[x_i, y_i]\,,\ i = 0, \dots, n$, a funkce $\varphi_i,\ i = 0, \dots, m$, jsou dány, kvadratická odchylka

$$\rho^2 = \sum_{i=0}^{n}(y_i - c_0\varphi_0(x_i) - c_1\varphi_1(x_i) - \cdots - c_m\varphi_m(x_i))^2$$

závisí pouze na koeficientech $c_0, \ldots, c_m$. Z diferenciálního počtu funkcí více proměnných je známo, že nutnou podmínkou pro to, aby $\rho^2(c_0, \ldots, c_m)$ nabývala minima, je splnění rovnic

$$\frac{\partial}{\partial c_j}(\rho^2) = \frac{\partial}{\partial c_j}\Big[\sum_{i=0}^{n}(y_i - c_0\varphi_0(x_i) - c_1\varphi_1(x_i) - \cdots - c_m\varphi_m(x_i))^2\Big] = 0, \quad j = 0, \ldots, m$$

Zderivováním dostaneme

$$\sum_{i=0}^{n} 2(y_i - c_0\varphi_0(x_i) - c_1\varphi_1(x_i) - \cdots - c_m\varphi_m(x_i))(-\varphi_j(x_i)) = 0, \quad j = 0, \ldots, m.$$

Rovnice vydělíme -2 a rozdělíme na jednotlivé sumy:

$$\sum_{i=0}^{n} y_i\varphi_j(x_i) - \sum_{i=0}^{n} c_0\varphi_0(x_i)\varphi_j(x_i) - \cdots - \sum_{i=0}^{n} c_m\varphi_m(x_i)\varphi_j(x_i) \quad j = 0, \ldots, m.$$

Z každé sumy můžeme vytknout odpovídající koeficient c_k. Snadnou úpravou pak dostaneme tzv. **normální rovnice** pro neznámé $c_0, \ldots, c_m$:

$$c_0 \sum_{i=0}^{n} \varphi_0(x_i)\varphi_j(x_i) + \cdots + c_m \sum_{i=0}^{n} \varphi_m(x_i)\varphi_j(x_i) = \sum_{i=0}^{n} y_i\varphi_j(x_i) \quad j = 0, \ldots, m.$$

Tato soustava rovnic po rozepsání vypadá takto:

$$\begin{array}{llll}
c_0 \sum_{i=0}^{n} \varphi_0^2(x_i) & + c_1 \sum_{i=0}^{n} \varphi_1(x_i)\varphi_0(x_i) & + \cdots + c_m \sum_{i=0}^{n} \varphi_m(x_i)\varphi_0(x_i) = \sum_{i=0}^{n} y_i\varphi_0(x_i) \\
c_0 \sum_{i=0}^{n} \varphi_0(x_i)\varphi_1(x_i) & + c_1 \sum_{i=0}^{n} \varphi_1^2(x_i) & + \cdots + c_m \sum_{i=0}^{n} \varphi_m(x_i)\varphi_1(x_i) = \sum_{i=0}^{n} y_i\varphi_1(x_i) \\
 & & \ddots \quad \vdots & \\
c_0 \sum_{i=0}^{n} \varphi_0(x_i)\varphi_m(x_i) & + c_1 \sum_{i=0}^{n} \varphi_1(x_i)\varphi_m(x_i) & + \cdots + c_m \sum_{i=0}^{n} \varphi_m^2(x_i) = \sum_{i=0}^{n} y_i\varphi_m(x_i)
\end{array} \tag{5.23}$$

Získaná soustava rovnic vypadá možná poněkud hrozivě a nepřehledně, ale uvidíme, že s konkrétními funkcemi φ_i se situace vyjasní.

Aproximace metodou nejmenších čtverců algebraickými polynomy

Velmi častá volba funkcí φ_i je $\varphi_i(x) = x^i,\ i = 0, 1, \ldots, m$, tj.

$$\varphi_0(x) = 1, \varphi_1(x) = x, \ldots, \varphi_m(x) = x^m$$

Aproximující funkce P_m je pak tvaru

$$P_m(x) = c_0 + c_1x + \cdots + c_mx^m$$

a jednotlivé sumy v soustavě normálních rovnic vyjdou

$$\sum_{i=0}^{n}\varphi_0^2(x_i)=\sum_{i=0}^{n}1=\underbrace{1+1+\cdots+1}_{n+1}=n+1\ ,\ \sum_{i=0}^{n}\varphi_1(x_i)\varphi_0(x_i)=\sum_{i=0}^{n}x_i\cdot 1=\sum_{i=0}^{n}x_i\ ,\dots$$

obecně

$$\sum_{i=0}^{n}\varphi_k(x_i)\varphi_l(x_i)=\sum_{i=0}^{n}x_i^k\cdot x_i^l=\sum_{i=0}^{n}x_i^{k+l},\quad k,l=0,\dots,m$$

Soustava normálních rovnic pak vypadá následovně

$$\begin{array}{ccccccccc}
c_0(n+1) & + & c_1\sum_{i=0}^{n}x_i & + & \dots & + & c_m\sum_{i=0}^{n}x_i^m & = & \sum_{i=0}^{n}y_i \\
c_0\sum_{i=0}^{n}x_i & + & c_1\sum_{i=0}^{n}x_i^2 & + & \dots & + & c_m\sum_{i=0}^{n}x_i^{m+1} & = & \sum_{i=0}^{n}x_iy_i \\
 & & & & & & \vdots & & \\
c_0\sum_{i=0}^{n}x_i^m & + & c_1\sum_{i=0}^{n}x_i^{m+1} & + & \dots & + & c_m\sum_{i=0}^{n}x_i^{2m} & = & \sum_{i=0}^{n}x_i^my_i
\end{array}\tag{5.24}$$

Speciálně pro aproximaci přímkou $P_1(x)=c_0+c_1\,x$ dostaneme soustavu

$$\begin{array}{ccccc}
c_0(n+1) & + & c_1\sum_{i=0}^{n}x_i & = & \sum_{i=0}^{n}y_i \\
c_0\sum_{i=0}^{n}x_i & + & c_1\sum_{i=0}^{n}x_i^2 & = & \sum_{i=0}^{n}x_iy_i
\end{array}\tag{5.25}$$

a pro aproximaci parabolou $P_2(x)=c_0+c_1\,x+c_2\,x^2$ soustavu

$$\begin{array}{ccccccccc}
c_0(n+1) & + & c_1\sum_{i=0}^{n}x_i & + & c_2\sum_{i=0}^{n}x_i^2 & = & \sum_{i=0}^{n}y_i \\
c_0\sum_{i=0}^{n}x_i & + & c_1\sum_{i=0}^{n}x_i^2 & + & c_2\sum_{i=0}^{n}x_i^3 & = & \sum_{i=0}^{n}x_iy_i \\
c_0\sum_{i=0}^{n}x_i^2 & + & c_1\sum_{i=0}^{n}x_i^3 & + & c_2\sum_{i=0}^{n}x_i^4 & = & \sum_{i=0}^{n}x_i^2y_i
\end{array}\tag{5.26}$$

Příklad 5.6 *Funkci zadanou následující tabulkou bodů aproximujte metodou nejmenších čtverců pomocí přímky.*

x_i	*0,2*	*0,5*	*0,9*	*1,6*	*2,0*	*2,9*	*3,5*
y_i	*16,58*	*19,30*	*18,12*	*20,94*	*20,90*	*24,66*	*24,50*

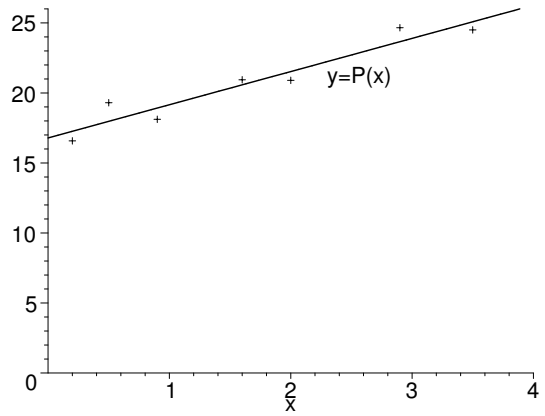
Řešení: Bylo zadáno 7 bodů, proto $n=6$. Koeficienty přímky získáme jako řešení soustavy rovnic 5.25. Pro přehlednost si všechny potřebné hodnoty zapíšeme do tabulky:

i	x_i	y_i	x_i^2	$x_i y_i$
0	0,2	16,58	0,04	3,316
1	0,5	19,30	0,25	9,650
2	0,9	18,12	0,81	16,308
3	1,6	20,94	2,56	33,504
4	2,0	20,90	4,00	41,800
5	2,9	24,66	8,41	71,514
6	3,5	24,50	12,25	85,750
$\sum$	11,6	145,00	28,32	261,842

Nyní můžeme sestavit normální rovnice:

$$\begin{aligned} 7\,c_0 &+ 11,6\,c_1 &&= 145 \\ 11,6\,c_0 &+ 28,32\,c_1 &&= 261,842 \end{aligned}$$

Jejich řešením je $c_0 \doteq 16,788$, $c_1 \doteq 2,370$.
Hledaná přímka je tedy $P_1(x) = 16,788 + 2,370\,x$. Zadané body jsou spolu s touto přímkou zobrazeny na obrázku 5.9.



Obrázek 5.9: K příkladu 5.6: zadané body a nalezená přímka

Poznámka. Pokud naměřené hodnoty vykazují periodické chování, je vhodnější je pomocí metody nejmenších čtverců aproximovat trigonometrickými polynomy. Za funkce φ_i můžeme volit
$\varphi_0(x) = 1$, $\varphi_1(x) = \cos x$, $\varphi_2(x) = \sin x$, $\varphi_3(x) = \cos 2x$, $\varphi_4(x) = \sin 2x$, ...

Shrnutí pojmů

Aproximace funkce spočívá v nahrazení zkoumané funkce f jednodušší funkcí, která nabývá přibližně stejných hodnot jako funkce f a se kterou se snadno pracuje.

U interpolace hledáme funkci, která má s f společné funkční hodnoty v tzv. uzlových bodech $x_0, x_1, \ldots, x_n$. Nejčastěji to bývá interpolační polynom nebo splajn.
Interpolační polynom $P_n(x)$ je algebraický polynom stupně nanejvýš n, pro nějž platí $P(x_i) = f(x_i)$, $i = 0, 1, \ldots, n$. Interpolační polynom pro zadané body existuje vždy právě jeden, ale můžeme jej vyjádřit v různém tvaru.
Existují speciální tvary interpolačních polynomů pro ekvidistatní uzly, tj. uzly takové, že krok mezi všemi dvojicemi sousedních uzlů je konstantní.
Lagrangeův interpolační polynom sestavíme přímo ze zadaných uzlů a funkčních hodnot v nich. Pro konstrukci Newtonova interpolačního polynomu musíme napřed vypočítat poměrné (jedná-li se o neekvidistantní uzly) nebo obyčejné (jedná-li se o ekvidistantní uzly) diference a interpolační polynom pak sestavíme pomocí nich. Výhodou Newtonova interpolačního polynomu oproti Lagrangeovu je, že se do něj snadněji dosazuje a snadněji lze přidat další uzel.
Za příznivých okolností platí v neuzlových bodech $f(x) \doteq P_n(x)$. Použijeme-li však příliš mnoho uzlových bodů, interpolační polynom může (i když nemusí) začít oscilovat. Proto je pro aproximaci funkce na dlouhém intervalu lepší splajn.
Splajn $S(x)$ je také funkce, pro niž platí $S(x_i) = f(x_i)$, $i = 0, 1, \ldots, n$, ale na rozdíl od interpolačního polynomu je to funkce definovaná po částech, je dána jiným předpisem na každém z intervalů $\langle x_i, x_{i+1} \rangle$, $i = 0, 1, \ldots, n-1$. Nejčastěji se používá tzv. přirozený kubický splajn. To je funkce, která je na každém intervalu $\langle x_i, x_{i+1} \rangle$ polynom třetího stupně $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$. Jednotlivé polynomy S_i a S_{i+1} na sebe musí v bodě x_{i+1} (tj. v bodě, kde se jejich definiční obory stýkají) spojitě navazovat až do druhé derivace včetně. Navíc požadujeme platnost okrajových podmínek $S_0''(x_0) = S_{n-1}''(x_n) = 0$.
Při výpočtu splajnu nejprve najdeme koeficienty c_i jako řešení jisté soustavy lineárních rovnic. Koeficienty b_i a d_i pak vypočteme pomocí nich. Pro koeficienty a_i platí $a_i = f_i$.
Metoda nejmenších čtverců se používá především v případě, kdy máme hodnoty $[x_i, y_i]$, $i = 0, 1, \ldots, n$, získané nějakým měřením (tj. zatížené chybami) a máme určitou představu o povaze funkční závislosti y na x. Předpokládáme, že tato funkční závislost je typu $y = c_1\varphi_1(x) + \cdots + c_m\varphi_m(x)$, kde φ_i, $i = 0, \ldots, m$, jsou známé funkce. Mezi všemi funkcemi tohoto známého typu hledáme tu, pro kterou je minimální tzv. kvadratická odchylka. Nalezení této funkce spočívá v nalezení hodnot koeficientů c_i, $i = 0, \ldots, m$. Ty najdeme jako řešení tzv. soustavy normálních rovnic. Pro aproximaci algebraickým polynomem je tvar soustavy známý. Speciálně pro polynom prvního stupně, přímku, je to 5.25 a pro polynom druhého stupně, parabolu, 5.26. Chceme-li použít jiný typ funkcí, dosadíme do obecného tvaru soustavy 5.23.

5.4 Otázky a příklady ke cvičení

U následujících výroků rozhodněte, zda se jedná o výrok pravdivý či nepravdivý.

Otázka 5.1 *Pomocí interpolačního polynomu můžeme vypočítat přibližnou hodnotu interpolované funkce v neuzlovém bodě.*

Otázka 5.2 *Pokud všechny body* $[x_i, f_i], i = 0, \ldots, n$, *leží v jedné přímce, pak grafem interpolačního polynomu daného těmito body je právě tato přímka.*

Otázka 5.3 *Je-li* $L_n(x)$ *Lagrangeův interpolační polynom určený uzly* $[x_i, f_i], i = 0, \ldots n$, *a* $N_n(x)$ *Newtonův interpolační polynom určený týmiž uzly, pak existuje bod* c, *pro který je* $L_n(c) \neq N_n(c)$.

Otázka 5.4 *Grafem lineárního splajnu je lomená čára.*

Otázka 5.5 *Při hledání přirozeného kubického splajnu musíme vyřešit soustavu lineárních rovnic.*

Otázka 5.6 *Kubický splajn může být nespojitá funkce.*

Otázka 5.7 *Graf funkce* $P_m(x)$, *kterou jsme získali metodou nejmenších čtverců z bodů* $[x_i, y_i], i = 0, \ldots, n$, *nikdy neprochází žádným z bodů* $[x_i, y_i]$.

Otázka 5.8 *Jsou-li zadány právě dva body* $[x_0, y_0]$ *a* $[x_1, y_1]$, $x_0 \neq x_1$, *pak přímka získaná metodou nejmenších čtverců pomocí těchto dvou bodů oběma body prochází.*

Příklad 5.1 *Najděte Lagrangeův interpolační polynom daný body*

x_i	-1	0	2
f_i	6	3	9

Polynom upravte. Proveďte zkoušku.

Příklad 5.2 *Najděte Lagrangeův interpolační polynom daný uzly*

x_i	$x_1 - h$	x_1	$x_1 + h$
f_i	f_0	f_1	f_2

Polynom upravte na tvar $L_2(x) = A(x - x_1)^2 + B(x - x_1) + C$.
(Výsledek tohoto příkladu bude použit v příkladech k další kapitole.)

Příklad 5.3 *Najděte Newtonův interpolační polynom daný body z příkladu 1. Pak k zadaným uzlům přidejte ještě bod* $[4, 5]$ *a opět najděte Newtonův interpolační polynom.*

Příklad 5.4 *Vypočtěte funkční hodnoty funkce* $f(x) = \sin x$ *v uzlových bodech* $x_0 = 0$, $x_1 = 0{,}8$, $x_2 = 1{,}6$, $x_3 = 2{,}4$, $x_4 = 3{,}2$. *(Tyto hodnoty jsou v radiánech, nikoli ve stupních.)*

a) Najděte Newtonův interpolační polynom daný těmito uzly a pak pomocí něj vypočtěte přibližně $\sin 1$

b) Přibližnou hodnotu $\sin 1$ *vypočtěte pomocí lineární interpolace ze vhodných dvou uzlů. Hodnoty vypočtené v a), b) porovnejte s přesnou hodnotou. Proč je výsledek b) dost nepřesný?*

Příklad 5.5 *Najděte přirozený kubický splajn daný uzly*

x_i	-3	-1	0	2
f_i	-5	3	4	-100

Vypočtěte hodnoty splajnu v bodech -2, -0,1 a 1.

Příklad 5.6 *Najděte přirozený kubický splajn daný uzly z příkladu 4. Pak pomocí tohoto splajnu vypočtěte přibližně* $\sin 1$.

Příklad 5.7 *Ukažte, že k funkci* $f : y = x + \mathrm{e}^x$ *existuje inverzní funkce* $x = f^{-1}(y)$. *S přesností 0,001 najděte hodnoty funkce* $f^{-1}(y)$ *pro* $y = 0$, $y = 0{,}5$ *a* $y = 1$. *(Použijte k tomu libovolnou z metod probraných v kapitole 4). Pak pomocí interpolace vypočtěte přibližně* $f^{-1}(0{,}3)$ *a* $f^{-1}(0{,}9)$.

Příklad 5.8 *Funkci zadanou následující tabulkou bodů aproximujte metodou nejmenších čtverců pomocí přímky. Načrtněte zadané body a vypočtenou přímku.*

x_i	0	1	2	3	4	5
y_i	-2,654	-0,041	-0,457	0,505	2,751	3,475

Příklad 5.9 *Funkci zadanou následující tabulkou bodů aproximujte metodou nejmenších čtverců pomocí paraboly.*

x_i	1	1,5	2	2,5	3
y_i	0,837	0,192	-0,950	-1,095	1,344

Příklad 5.10 *Funkci zadanou následující tabulkou bodů aproximujte metodou nejmenších čtverců pomocí funkce* $y = c_0 + c_1 \sin x + c_2 \cos x$.

x_i	-5	-4	-3	-2	-1	0	1	2	3	4	5
y_i	1,55	2,85	2,81	0,49	-0,43	-0,92	0,68	2,76	2,96	1,48	-0,80

Odpovědi na otázky a řešení příkladů viz 9.5

Programovací úlohy

U programů na aproximaci funkcí by bylo velmi pěkné mít vždy výstup i v podobě grafu nalezeného interpolačního polynomu a pod. Zda tomu tak skutečně bude, opět ponecháme na schopnostech programátora.

Programovací úloha 1 Napište program, který pro zadané uzly $[x_i, f_i], i = 0, \ldots, n$, vypočítá hodnotu Newtonova interpolačního polynomu v zadaném bodě x.

Programovací úloha 2 Řešte totéž jako v úloze 1, ale pro ekvidistantní uzly.

Programovací úloha 3 Napište program, který pro zadané uzly $[x_i, f_i], i = 0, \ldots, n$, vypočítá hodnotu přirozeného kubického splajnu v zadaném bodě x.

Programovací úloha 4 Napište program, který pro zadané uzly $[x_i, y_i], i = 0, \ldots, n$, vypočítá koeficienty přímky získané metodou nejmenších čtverců a kvadratickou odchylku.

6 Numerické derivování a integrování

Cíl kapitoly

V této kapitole se budeme zabývat otázkou, jak vypočítat derivaci a integrál z funkce, která je zadána pouze tabulkou bodů nebo pro kterou by byl analytický výpočet příliš složitý.
Základní myšlenkou je nahradit funkci interpolačním polynomem, popřípadě jinou aproximací, a derivovat či integrovat aproximující funkci.

6.1 Numerické derivování

Jak již bylo řečeno v úvodu, budeme řešit problém, jak vypočítat hodnotu derivace dané funkce v určitém bodě nikoli analyticky, ale pouze přibližně, a to pomocí známých funkčních hodnot v určitých bodech.
Můžeme k tomu použít interpolační polynom. Hodnotu derivace funkce nahradíme hodnotou derivace interpolačního polynomu. Tedy, je-li $P_n(x)$ interpolační polynom daný funkcí $f(x)$ a uzlovými body $x_0, x_1, \ldots, x_n$, položíme

$$f'(x) \doteq P_n'(x).$$

Podobně pro derivace vyšších řádů (ovšem pouze do řádu n, pro vyšší už ne) můžeme položit

$$f^{(s)}(x) \doteq P^{(s)}(x).$$

Poznamenejme, že v uzlových bodech se hodnoty derivací funkce a interpolačního polynomu nemusejí shodovat. Pro ilustraci může posloužit opět obrázek 5.4, na kterém je dobře vidět, že zatímco funkční hodnoty v uzlových bodech jsou u funkce a interpolačního polynomu stejné, směrnice tečen k těmto dvěma grafům (tj. hodnoty derivací) jsou v uzlových bodech velmi odlišné.

6.1.1 Některé často používané vzorce pro numerické derivování

Uvedeme zde některé jednodušší, často užívané vzorce pro první a druhou derivaci v uzlových bodech. V tomto textu se s nimi ještě setkáme v kapitolách věnovaných numerickému řešení diferenciálních rovnic.

Jako poslední je v každém vzorci uveden chybový člen, který při samotném výpočtu zanedbáváme.
Čím vyšší mocnina kroku h se v něm vyskytuje, tím je chyba menší (a tedy vzorec lepší), neboť h bývá zpravidla malé číslo, $h \ll 1$, a pro taková čísla platí $h > h^2 > h^3 > \cdots$.

Nejjednodušší vzorec pro derivaci prvního řádu dostaneme zderivováním interpolačního polynomu prvního stupně daného uzly x_0 a $x_1 = x_0 + h$.

Má-li funkce f druhou derivaci na intervalu $\langle x_0, x_1\rangle$, pak existují body $\xi_0, \xi_1 \in \langle x_0, x_1\rangle$ tak, že platí

$$f'(x_0) = \frac{f(x_1) - f(x_0)}{h} - \frac{h}{2} f''(\xi_0) \tag{6.1}$$

$$f'(x_1) = \frac{f(x_1) - f(x_0)}{h} - \frac{h}{2} f''(\xi_1). \tag{6.2}$$

Tyto vzorce lze též odvodit pomocí Taylorova rozvoje funkce f.

Derivováním interpolačního polynomu druhého stupně daného uzly $x_0 = x_1 - h, x_1$ a $x_2 = x_1 + h$ dostaneme přesnější vzorce pro první derivaci v těchto uzlových bodech.
Má-li funkce f čtvrtou derivaci na intervalu $\langle x_0, x_2\rangle$, pak existují body $\xi_0, \xi_1, \xi_2 \in \langle x_0, x_2\rangle$ takové, že

$$f'(x_0) = \frac{-3f(x_0) + 4f(x_1) - f(x_2)}{2h} + \frac{h^2}{3} f'''(\xi_0) \tag{6.3}$$

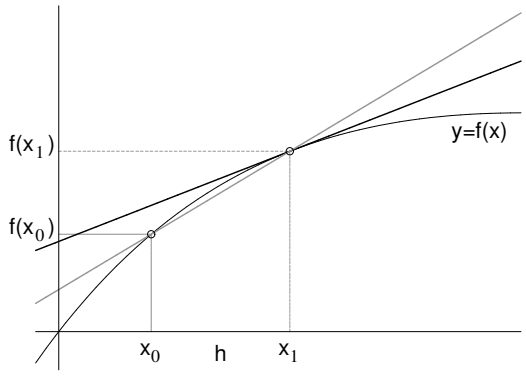
$$f'(x_1) = \frac{f(x_2) - f(x_0)}{2h} - \frac{h^2}{6} f'''(\xi_1) \tag{6.4}$$

$$f'(x_2) = \frac{f(x_0) - 4f(x_1) + 3f(x_2)}{2h} + \frac{h^2}{3} f'''(\xi_2) \tag{6.5}$$

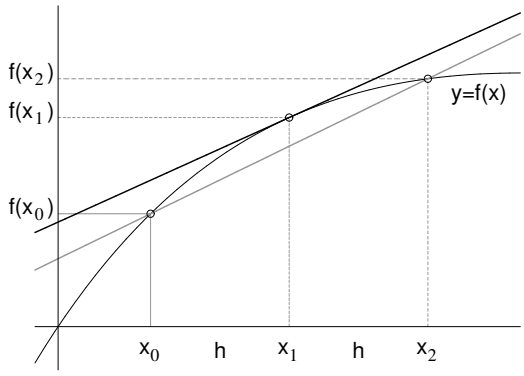
Pomocí druhé derivace téhož interpolačního polynomu dostaneme vzorec pro druhou derivaci funkce f v bodě x_1.
Má-li funkce f pátou derivaci na intervalu $\langle x_0, x_2\rangle$, pak existuje bod $\xi \in \langle x_0, x_2\rangle$ takový, že

$$f''(x_1) = \frac{f(x_0) - 2f(x_1) + f(x_2)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi) \tag{6.6}$$

Na obrázcích 6.1 a 6.2 je zachycen geometrický význam vzorců 6.2 a 6.4. Hodnota derivace funkce f v bodě x_1, tj. směrnice tečny ke grafu funkce v tomto bodě (tečna je na obrázcích nakreslena černě), je přibližně rovna směrnici sečny dané body x_0 a x_1, resp. x_0 a x_2 (tyto sečny jsou na obrázcích nakresleny šedě).



Obrázek 6.1: Ilustrace ke vzorci 6.2



Obrázek 6.2: Ilustrace ke vzorci 6.4

Poznámka o zaokrouhlovací chybě při numerické derivování

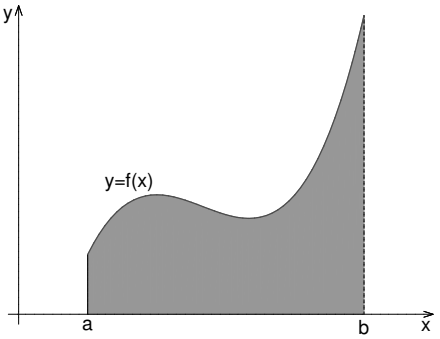
Mohlo by se zdát, že zmenšováním kroku h lze dosáhnout při numerickém derivování libovolné přesnosti. Bohužel se však ukazuje, že při příliš malém h může velmi narůst vliv zaokrouhlovací chyby.
To je vidět už z nejjednoduššího vzorce 6.2. Pro malé h může být $f(x_0) \doteq f(x_1)$ a tedy v čitateli zlomku odčítáme dvě sobě velmi blízká čísla, výsledek pak navíc opět dělíme malým číslem. To jsou operace vzhledem k zaokrouhlovací chybě velmi riskantní, viz kapitolu o chybách.
Naopak, při velkém kroku h nelze očekávat velkou přesnost vzhledem k chybě metody. Proto je potřeba volit kompromis, více o tom v [4].

V případě funkcí, jejichž hodnoty byly získány např. experimentálně a jsou zatíženy nezanedbatelnými chybami, se doporučuje nejprve tyto hodnoty metodou nejmenších čtverců „vyrovnat" a potom teprve funkci derivovat.

6.2 Numerické integrování

Určení primitivní funkce k dané funkci $f(x)$ může být nesnadné, jak si čtenář jistě vzpomene z prvního semestru matematiky, někdy je to zcela nemožné. V případě, že jsou hodnoty funkce f dány tabulkou, pojem primitivní funkce úplně ztrácí smysl. Přesto můžeme chtít z takové funkce integrál vypočítat.

Zde se budeme zabývat výpočtem určitého integrálu $\int_a^b f(x)\mathrm{d}x$. Jak si jistě všichni vzpomenou, pomocí tohoto integrálu se vypočítá obsah plochy pod grafem funkce $f(x)$ na intervalu $\langle a, b\rangle$, viz obrázek 6.3.



Obrázek 6.3: Připomenutí významu určitého integrálu

Numerický výpočet tohoto integrálu se nazývá numerická kvadratura. Jedna z možných cest je nahrazení funkce f na intervalu $\langle a, b\rangle$ interpolačním polynomem. Ten již se pak zintegruje snadno.

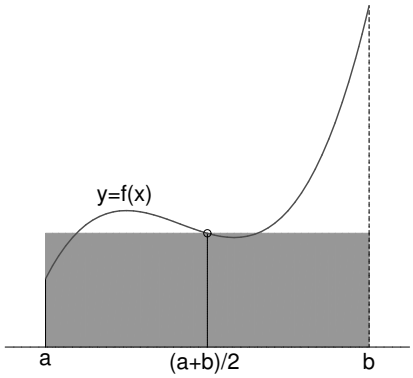
6.2.1 Newton-Cotesovy vzorce

Newton-Cotesovy kvadraturní vzorce (kvadraturní formule) obdržíme integrováním interpolačních polynomů s ekvidistantními uzly. Můžeme je rozdělit do dvou skupin:

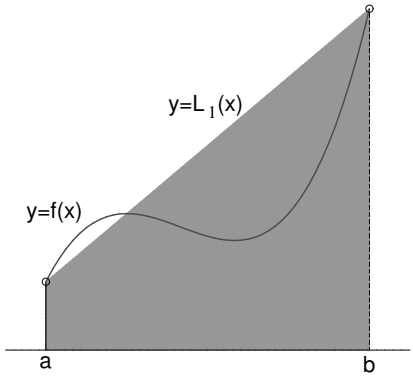
- uzavřené vzorce, kde krajní body intervalu bereme za uzly kvadratury
- otevřené vzorce, kde krajní body nebereme za uzly kvadratury a uzly jsou položeny symetricky podle středu intervalu.

Blíže se zde budeme zabývat uzavřenými formulemi, z otevřených se můžeme zmínit o nejjednodušší z nich, a tou je tzv. **obdélníková metoda**.
Za jediný uzel interpolace bereme střed intervalu $\langle a, b\rangle$, vlastně funkci na tomto intervalu nahradíme konstantou $f(\frac{a+b}{2})$ a integrál je pak přibližně roven obsahu obdélníka, viz obrázek 6.4.

$$\int_a^b f(x)\mathrm{d}x \doteq (b-a)f(\frac{a+b}{2}). \tag{6.7}$$



Obrázek 6.4: Obdélníková metoda



Obrázek 6.5: Lichoběžníková metoda

Z uzavřených vzorců je nejjednodušší **lichoběžníková metoda** (nebo též lichoběžníkové pravidlo).
Funkci $f(x)$ nahradíme na intervalu $\langle a, b\rangle$ lineárním interpolačním polynomem daným uzly a, b (zde zapsaným v Lagrangeově tvaru):

$$L_1(x) = f(a)\,\frac{x-b}{a-b} + f(b)\,\frac{x-a}{b-a}.$$

Integrací tohoto polynomu po použití jednoduchých úprav dostaneme

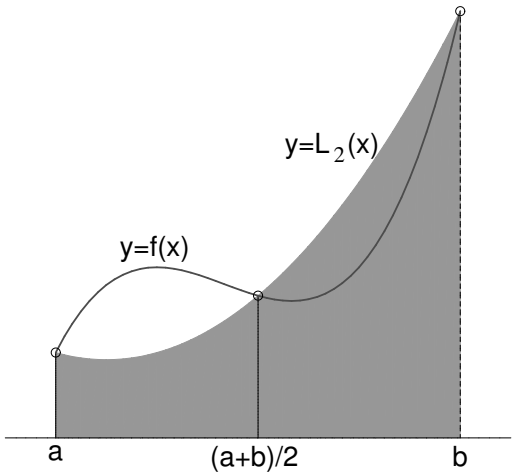
$$\int_a^b f(x)\mathrm{d}x \doteq \int_a^b L_1(x)\mathrm{d}x = \frac{b-a}{2}\,\Big(f(a)+f(b)\Big). \tag{6.8}$$

V tomto případě nahrazujeme obsah podgrafu funkce f obsahem příslušného lichoběžníka, viz obrázek 6.5, odtud název metody.

Poznámka. Vzorec 6.8 můžeme dostat i použitím známého vztahu pro obsah lichoběžníka $S = \frac{1}{2}(A + C)v$, kde A a C jsou délky podstav lichoběžníka a v je jeho výška. Musíme si ovšem uvědomit, že v tomto případě je lichoběžník obrácen, jeho podstavy jsou svisle.

Na integraci interpolačního polynomu druhého stupně, za jehož uzly bereme a, b a střed integračního intervalu, tj. $\frac{a+b}{2}$, je založena tzv. **Simpsonova metoda** (viz obrázek 6.6):

$$\int_a^b f(x)\mathrm{d}x \doteq \frac{b-a}{6}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right). \tag{6.9}$$



Obrázek 6.6: Simpsonova metoda

Podobně bychom mohli integrovat interpolační polynomy vyšších stupňů. Přibližná hodnota integrálu vždy vyjde jako součet určitých násobků funkčních hodnot v uzlech. Obecně je **uzavřený Newton-Cotesův vzorec** tvaru

$$\int_a^b f(x)\mathrm{d}x \doteq (b-a)\sum_{i=0}^{n} H_i f(x_i), \tag{6.10}$$

kde n je stupeň použitého interpolačního polynomu, H_i jsou tzv. Cotesovy koeficienty a x_i jsou uzly, pro něž platí $x_i = a + ih, i = 0, \ldots, n$, ($h = \frac{b-a}{n}$ je krok mezi uzly).
Přehled Cotesových koeficientů až do $n = 8$ lze nalézt např. v [**?**].

Chyba E Newton-Cotesových vzorců se vypočte integrací chyby interpolace 5.12,

$$E = \frac{1}{(n+1)!}\int_a^b f^{(n+1)}(\xi)(x - x_0)\cdots(x - x_n)\mathrm{d}x$$

Zjednodušení tohoto výrazu je dosti obtížné, je ho potřeba provést zvlášť pro n sudé a pro n liché. Podrobnosti lze nalézt v [4].

Pro n sudé platí

$$E = \frac{f^{(n+2)}(\eta)}{(n+2)!} \int_a^b x(x-x_0)\cdots(x-x_n)\,\mathrm{d}x, \tag{6.11}$$

a pro n liché

$$E = \frac{f^{(n+1)}(\eta)}{(n+1)!} \int_a^b (x-x_0)\cdots(x-x_n)\,\mathrm{d}x, \tag{6.12}$$

kde $\eta \in [a,b]$. Integrály v těchto vzorcích lze pro konkrétní n vypočítat (byť je to poněkud pracné).

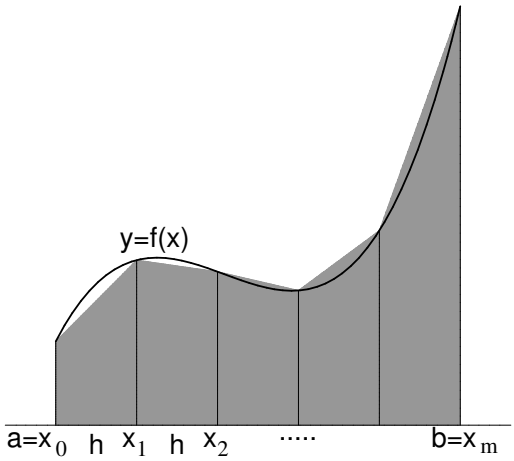
Např. chyba lichoběžníkové metody pomocí vzorce 6.12 vyjde

$$E = -\frac{1}{12}(b-a)^3 f''(\eta). \tag{6.13}$$

V kapitole o interpolaci jsme ukázali, že interpolační polynomy vyšších stupňů mohou oscilovat a nemusejí dobře vystihnout chování interpolované funkce. Také výpočet Cotesových koeficientů je pro velká n složitý. Proto se Newton-Cotesových vzorců vysokých řádů užívá zřídka.

6.2.2 Složené kvadraturní vzorce

Již z obrázků je vidět, že chyba integrace pomocí uvedených Newton-Cotesových vzorců nízkých řádů může být značná. Proto je lepší interval $\langle a,b\rangle$ rozdělit na větší počet stejných dílků a na každém z nich použít vybraný jednoduchý kvadraturní vzorec.



Obrázek 6.7: Složené lichoběžníkové pravidlo

Rozebereme si nyní podrobněji **složené lichoběžníkové pravidlo**.
Interval $\langle a, b\rangle$ rozdělíme na m subintervalů délky $h = \frac{b-a}{m}$ - viz obrázek 6.7. Na každém subintervalu použijeme jednoduché lichoběžníkové pravidlo. Platí

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &= \int_{x_0}^{x_1} f(x)\,\mathrm{d}x + \int_{x_1}^{x_2} f(x)\,\mathrm{d}x + \cdots + \int_{x_{m-1}}^{x_m} f(x)\,\mathrm{d}x \doteq \\ &\doteq \frac{h}{2}\Big(f(x_0)+f(x_1)\Big) + \frac{h}{2}\Big(f(x_1)+f(x_2)\Big) + \cdots + \frac{h}{2}\Big(f(x_{m-1})+f(x_m)\Big)\end{aligned}$$

Celkem tedy

$$\int_a^b f(x)\,\mathrm{d}x \doteq h\left(\frac{1}{2}\,f(x_0) + f(x_1) + \cdots + f(x_{m-1}) + \frac{1}{2}\,f(x_m)\right) = L_m \tag{6.14}$$

Je zřejmé, že čím jemněji interval $\langle a, b\rangle$ nadělíme, tím přesnější bude výsledek.

Chyba integrace na každém dílčím intervalu $\langle x_{i-1}, x_i\rangle$ je podle 6.13 $E_i = -\frac{1}{12}\,h^3 f''(\eta_i)$.
Celková chyba je tedy

$$E = -\frac{h^3}{12}\Big(f''(\eta_1) + f''(\eta_2) + \cdots + f''(\eta_m)\Big).$$

Je-li funkce f'' na intervalu $[a, b]$ spojitá, existuje bod $\eta \in \langle a, b\rangle$ tak, že platí

$$f''(\eta_1) + f''(\eta_2) + \cdots + f''(\eta_m) = m f''(\eta)$$

Dohromady dostaneme pro **chybu složeného lichoběžníkového pravidla**

$$E = -\frac{h^3}{12}\,m f''(\eta) = -\frac{(b-a)^3}{12\,m^3}\,m f''(\eta) = -\frac{(b-a)^3}{12\,m^2}\,f''(\eta). \tag{6.15}$$

Podobně jako u chyby interpolace, je prakticky nemožné určit bod η. Lze-li nalézt $M_2 = \max_{t\in\langle a,b\rangle} |f''(t)|$, můžeme chybu alespoň shora odhadnout. Platí totiž

$$|E| \le \frac{(b-a)^3}{12\,m^2} M_2 \tag{6.16}$$

Tento odhad lze použít též pro určení vhodného počtu dělení m, chceme-li, aby chyba integrace nepřesáhla nějaké zadané ε.

Spíše než odhad chyby se ovšem pro dosažení žádané přesnosti ε používá jiný postup.
Můžeme konstruovat posloupnost $L_1, L_2, L_4, \ldots$ Její výpočet je velmi úsporný, protože všechny funkční hodnoty použité v nějakém L_m se použijí i při výpočtu L_{2m}. Platí

$$L_{2m} = \frac{1}{2}L_m + \frac{b-a}{2m}\Big(f(x_1) + f(x_3) + \cdots + f(x_{2m-1})\Big),$$

kde v závorce je pouze součet funkčních hodnot v nových dělících bodech, které původní dělení zjemňují.
Výpočet zastavíme, jakmile je splněna podmínka $|L_{2m} - L_m| < \varepsilon$.
(Splněním této podmínky ale není zaručeno, že se L_{2m} od přesné hodnoty integrálu liší o méně než ε.)

Zcela analogicky jako složené lichoběžníkové pravidlo můžeme odvodit **složené Simpsonovo pravidlo**.
Interval $\langle a, b\rangle$ rozdělíme na **sudý** počet m dílků délky $h = \frac{b-a}{m}$ a postupně na dvojicích sousedních dílků použijeme jednoduché Simpsonovo pravidlo.
Po úpravě dostaneme

$$\int_a^b f(x)\,\mathrm{d}x \doteq \tag{6.17}$$
$$\doteq \frac{h}{3}\Big(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{m-2}) + 4f(x_{m-1}) + f(x_m)\Big) = S_m$$

Pro **odhad chyby** E se použije vzorec 6.11 a podobné úvahy jako při odvozování chyby složeného lichoběžníkového pravidla. Vyjde

$$E = -\frac{(b-a)^5}{180\,m^4} f^{(4)}(\eta), \quad \eta \in \langle a, b\rangle \tag{6.18}$$

a pro horní odhad chyby

$$|E| \le \frac{(b-a)^5}{180\,m^4} \max_{x\in\langle a,b\rangle} |f^{(4)}(x)|. \tag{6.19}$$

Příklad 6.1 *Vypočtěte přibližnou hodnotu integrálu* $\int_0^2 e^{-x^2}\,dx$ *pomocí složeného lichoběžníkového pravidla pro* $m = 4$. *Odhadněte, jaké chyby se při tomto výpočtu nanejvýš můžeme dopustit.*

Řešení: Dosadíme do vzorce 6.14. Délka kroku h je v tomto případě $\frac{2-0}{4} = 0,5$. Přibližná hodnota integrálu je tedy

$$L_4 = 0,5 \cdot \Big(\frac{1}{2}f(0) + f(0,5) + f(1) + f(1,5) + \frac{1}{2}f(2)\Big) =$$
$$= 0,5 \cdot \Big(\frac{1}{2}\,e^0 + e^{-0,25} + e^{-1} + e^{-2,25} + \frac{1}{2}\,e^{-4}\Big) \doteq 0,8806$$

Odhad chyby dostaneme pomocí vzorce 6.16.
Musíme vypočítat druhou derivaci funkce $f(x) = e^{-x^2}$. Ta vyjde $f''(x) = e^{-x^2}(4x^2 - 2)$. Nyní najdeme maximum její absolutní hodnoty na intervalu $\langle 0, 2\rangle$. Využitím poznatků z prvního semestru matematiky zjistíme, že funkce $f''(x)$ nabývá lokálního minima v bodě $x = 0$ a lokálního maxima v bodech $x = \pm\frac{\sqrt{6}}{2}$. Nás však zajímá maximum absolutní hodnoty na intervalu $\langle 0, 2\rangle$. Vypočteme hodnoty f'' ve všech „podezřelých" bodech:

$$f''(0) = -2 \quad f''(\frac{\sqrt{6}}{2}) \doteq 0,89 \quad f''(2) \doteq 0,26$$

V absolutní hodnotě je z těchto čísel největší -2, tedy $M_2 = |-2| = 2$.
Celkem je tedy absolutní hodnota chyby nanejvýš rovna $\frac{(2-0)^3}{12\cdot 4^2} \cdot 2 = \frac{1}{12} = 0,08\overline{33}$

Příklad 6.2 *Zjistěte, jakou délku kroku je třeba zvolit při výpočtu integrálu* $\int_0^2 e^{-x^2}\,dx$ *(téhož jako v příkladu 6.1) pomocí složeného lichoběžníkového pravidla, chceme-li, aby chyba integrace nebyla větší než* $0,001$.

Řešení: Přehlednější je najít nejprve vhodný počet dělení m, z něj již délku kroku určíme snadno.
Víme, že pro chybu E platí $|E| \leq \frac{(b-a)^3}{12\,m^2} M_2$. V příkladu 6.1 jsme zjistili, že $M_2 = 2$. Najdeme-li m tak, aby výraz na pravé straně předchozí nerovnosti byl menší než $0,001$, bude zaručeno, že i chyba E bude dostatečně malá. Má tedy platit

$$\frac{(2-0)^3}{12\,m^2} \cdot 2 \leq 0,001$$

Odtud snadno dostaneme, že

$$\begin{aligned} m^2 &\geq \frac{8 \cdot 2}{12 \cdot 0,001} \\ m &\geq 36,51 \end{aligned}$$

Zvolíme-li tedy $m = 37$ (nebo jakékoli větší), je zaručeno, že chyba bude menší než 0,001. Hledaná délka kroku může být tedy $\frac{2}{37}$.
Poznamenejme, že takto získaný počet dělení m může být zbytečně velký. V tomto příkladu by ve skutečnosti pro dosažení zadané přesnosti stačilo už $m = 5$ - to ale bez znalosti přesné hodnoty integrálu nejsme schopni rozeznat. S počtem dělení získaným právě předvedeným postupem máme sice možná více práce, ale zato jistotu, že výsledek bude dost přesný.

Poznámka. Kromě Newton-Cotesových kvadraturních vzorců existuje i mnoho dalších. Důležité jsou např. Gaussovy kvadraturní formule. V nich se přibližná hodnota integrálu opět počítá jako lineární kombinace funkčních hodnot,

$$\int_a^b f(x)\,\mathrm{d}x \doteq \sum_{i=0}^{n} H_i f(x_i).$$

Koeficienty $H_i \in \mathbb{R}$ a uzly $x_i \in \langle a, b\rangle$ jsou určeny tak, aby vzorec byl přesný pro integrování polynomů do stupně $2n+1$ včetně.

Poznámka. Numerický výpočet neurčitého integrálu $\int f(x)\,\mathrm{d}x$ spočívá v nalezení funkce $y(x) = \int_{x_0}^{x} f(t)\,\mathrm{d}t$.
Tato úloha je ekvivalentní s nalezením řešení Cauchyovy počáteční úlohy

$$y' = f(x), \quad y(x_0) = 0.$$

Metodám numerického řešení takovýchto úloh bude věnována kapitola 8.

Shrnutí pojmů

Derivaci funkce v určitém bodě můžeme přibližně vypočítat jako hodnotu derivace interpolačního polynomu v tomtéž bodě. Nejčastěji se k tomuto účelu používají interpolační polynomy nízkých stupňů. Použijeme-li lineární polynom, dostaneme pro výpočet první derivace vzorce 6.1 a 6.2, při použití kvadratického polynomu vyjdou přesnější vzorce 6.3–6.5.
Pro přibližný výpočet druhé derivace můžeme použít druhou derivaci interpolačního polynomu, v případě kvadratického vyjde formule 6.6.
U numerického integrování můžeme postupovat obdobně. Integrovanou funkci nahradíme interpolačním polynomem, a ten pak zintegrujeme. Tím dostáváme tzv. Newton-Cotesovy kvadraturní vzorce. Přibližná hodnota integrálu je vyjádřena jako lineární kombinace funkčních hodnot integrované funkce v uzlových bodech. Nejčastěji se k tomu účelu používá interpolační polynom prvního stupně - lichoběžníková metoda - nebo druhého stupně - Simpsonova metoda.
Protože pro interval velké délky by takto získané výsledky byly velmi nepřesné, v praxi se používají složené kvadraturní vzorce. Ty získáme tak, že interval rozdělíme na velký počet malých dílků stejné délky a na každém dílku (u lichoběžníkové metody), resp. na každé dvojici dílků (u Simpsonovy metody), aplikujeme jednoduchý kvadraturní vzorec.
Chybu u numerické integrace lze někdy vypočítat pomocí vzorců 6.16 nebo 6.19, často je však takovýto výpočet příliš obtížný. Proto se v praxi používá spíše postup, při kterém postupně zdvojnásobujeme počet dílků, na který dělíme interval, a zastavíme se, až jsou si výsledky získané s nějakým počtem dílků m a jeho dvojnásobkem $2m$ dostatečně blízké.

6.3 Otázky a příklady ke cvičení

U následujících výroků rozhodněte, zda se jedná o výrok pravdivý či nepravdivý.

Otázka 6.1 *Je-li P_n interpolační polynom daný uzly $[x_i, f_i], i = 0, \ldots, n$, pak v uzlových bodech platí $f'(x_i) = P'_n(x_i)$.*

Otázka 6.2 *Přibližnou hodnotu derivace funkce f v bodě a můžeme určit např. pomocí funkčních hodnot f v bodech $a + 0{,}1$ a $a - 0{,}1$.*

Otázka 6.3 *Čím menší krok h zvolíme při výpočtu přibližné hodnoty derivace pomocí vzorce $f'(x_0) \doteq \frac{f(x_0+h)-f(x_0)}{h}$, tím menší se dopustíme chyby.*

Otázka 6.4 *Pro výpočet integrálu $\int_a^b f(x)\mathrm{d}x$ pomocí lichoběžníkového pravidla musíme nalézt primitivní funkci k funkci f.*

Otázka 6.5 *Složené Simpsonovo pravidlo je obvykle přesnější než složené lichoběžníkové pravidlo (při stejném počtu dělení intervalu).*

Otázka 6.6 *Použijeme-li pro výpočet integrálu $\int_a^b x^2\mathrm{d}x$ Simpsonovo pravidlo, dostaneme přesnou hodnotu tohoto integrálu.*

Otázka 6.7 *Je-li integrovaná funkce f na intervalu $\langle a, b\rangle$ konvexní (nad tečnou), pak přibližná hodnota integrálu $\int_a^b f(x)\mathrm{d}x$ získaná lichoběžníkovou metodou je vždy větší než přesná hodnota tohoto integrálu.*

Příklad 6.1 *Vypočtěte přibližné hodnoty derivace funkce G ve všech uzlových bodech*

a) pomocí vzorce s chybou řádu h (tj. 6.1 nebo 6.2)

b) pomocí vzorců s chybou řádu h^2 (tj. 6.4 ve vnitřních uzlech a 6.3, resp. 6.5, v krajních uzlech)

x	1	1,1	1,2	1,3
$G(x)$	0,8427	0,8802	0,9103	0,9340

Porovnejte vypočtené hodnoty s přesnými hodnotami derivace, víme-li, že $G(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2}\mathrm{d}t$.

Příklad 6.2 *Pomocí výsledku příkladu 2 z kapitoly o aproximaci (interpolační polynom pro funkci f s uzly $x_1 - h, x_1, x_1 + h$) odvoďte vzorce pro numerické derivování 6.3–6.6.*

Příklad 6.3 *Pomocí výsledku příkladu 2 z kapitoly o aproximaci (interpolační polynom pro funkci f s uzly $x_1 - h, x_1, x_1 + h$) odvoďte Simpsonovo pravidlo pro výpočet určitého integrálu.*

Příklad 6.4 *Integrál $\int_0^{\pi/2} \sin x\,\mathrm{d}x$ vypočtěte přibližně (jednoduchou) a)lichoběžníkovou, b)Simpsonovou metodou. Porovnejte s přesnou hodnotou integrálu.*

Příklad 6.5 *Integrál $\int_1^2 \frac{\sin x}{x}\,\mathrm{d}x$ vypočtěte přibližně složeným lichoběžníkovým pravidlem pro a) $m = 4$ b) $m = 8$. Při výpočtu b) využijte výsledek a).*

Příklad 6.6 *Vypočtěte přibližně $G(1, 2)$, je-li $G(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2}\mathrm{d}t$. Použijte složené Simpsonovo pravidlo pro $m = 6$.*

Příklad 6.7 *Integrál $\int_{-1}^1 \frac{1}{1+x^2}\,\mathrm{d}x$ vypočtěte přibližně složeným a) lichoběžníkovým, b) Simpsonovým pravidlem pro $m = 4$. Výsledky porovnejte s přesnou hodnotou integrálu.*

Příklad 6.8 *Vypočtěte přibližně $\int_0^1 f(x)\,\mathrm{d}x$, známe-li tyto hodnoty funkce f :*

x	0	0,25	0,5	0,75	1
$f(x)$	1	0,57	-0,30	-0,07	1,28

Použijte tu z probraných metod, od níž lze očekávat nejvyšší přesnost.

Příklad 6.9 *Určete maximální možnou chybu při výpočtu integrálu $\int_0^1 \sqrt{1 + x^2}\,\mathrm{d}x$ složeným lichoběžníkovým pravidlem s $h = 0,25$.*

Příklad 6.10 *Vypočtěte, na kolik dílků je potřeba rozdělit interval, aby chyba při výpočtu integrálu $\int_{\pi/4}^{\pi/2} \ln(\sin x)\,\mathrm{d}x$ složeným Simpsonovým pravidlem nepřesáhla 10^{-4}.*

Příklad 6.11 *Z jednoduchého Simpsonova pravidla odvoďte složené Simpsonovo pravidlo.*

Příklad 6.12 *Ukažte, že $S_{2m} = \frac{1}{3}(4L_{2m} - L_m)$.*

Odpovědi na otázky a řešení příkladů viz 9.6

Programovací úlohy

Programovací úloha 1 Napište program, který vypočte integrál ze zadané funkce f v zadaných mezích a, b pomocí složeného

a) lichoběžníkového

b) Simpsonova

pravidla se zadaným počtem dělení m.

Programovací úloha 2 Napište program, který vypočte integrál ze zadané funkce f v zadaných mezích a, b pomocí složeného lichoběžníkového pravidla se zadanou přesností ε. Počítejte $L_1, L_2, L_4, L_8, \ldots$, dokud nebude přesnost dosažena.

7 Nepodmíněná optimalizace

Cíl kapitoly

Tato kapitola se zabývá některými numerickými metodami pro nalezení extrému funkce (maximum, minimum) jedné i více proměnných. Této proceduře se obvykle říká optimalizace. Pokud množina, na které extrém hledáme, není omezena žádnými vazbami, mluvíme o nepodmíněné optimalizaci.
Optimalizace má úzký vztah k metodám řešení soustav nelineárních rovnic. Pokud chceme najít přibližně kořen soustavy nelineárních rovnic 4.13, můžeme místo toho hledat minimum funkce

$$\varphi(x_1, x_2, \ldots, x_n) = f_1^2(x_1, x_2, \ldots, x_n) + \ldots + f_n^2(x_1, x_2, \ldots, x_n)$$

a přibližné minimum potom zpřesnit rychle, avšak pouze lokálně konvergentní Newtonovou metodou. Naopak za předpokladu, že funkce $f(x_1, x_2, \ldots, x_n))$ má parciální derivace podle všech svých proměnných, je bod, ve kterém nastává extrém, nutně stacionárním bodem a lze ho hledat jako řešení soustavy rovnic

$$\begin{aligned} \frac{\partial f(x_1, x_2, \ldots, x_n)}{\partial x_1} &= 0 \\ \frac{\partial f(x_1, x_2, \ldots, x_n)}{\partial x_2} &= 0 \\ &\vdots \\ \frac{\partial f(x_1, x_2, \ldots, x_n)}{\partial x_n} &= 0 \end{aligned} \tag{7.1}$$

V dalším se budeme zabývat pouze hledáním minima funkce (minimalizací). Bod, ve kterém funkce nabývá svého minima, budeme značit x^*. Pokud bychom potřebovali najít maximum funkce $f(x)$, stačí hledat bod minima x^* funkce $-f(x)$ a maximum spočítat jako $f(x^*)$.

7.1 Jednorozměrné algoritmy

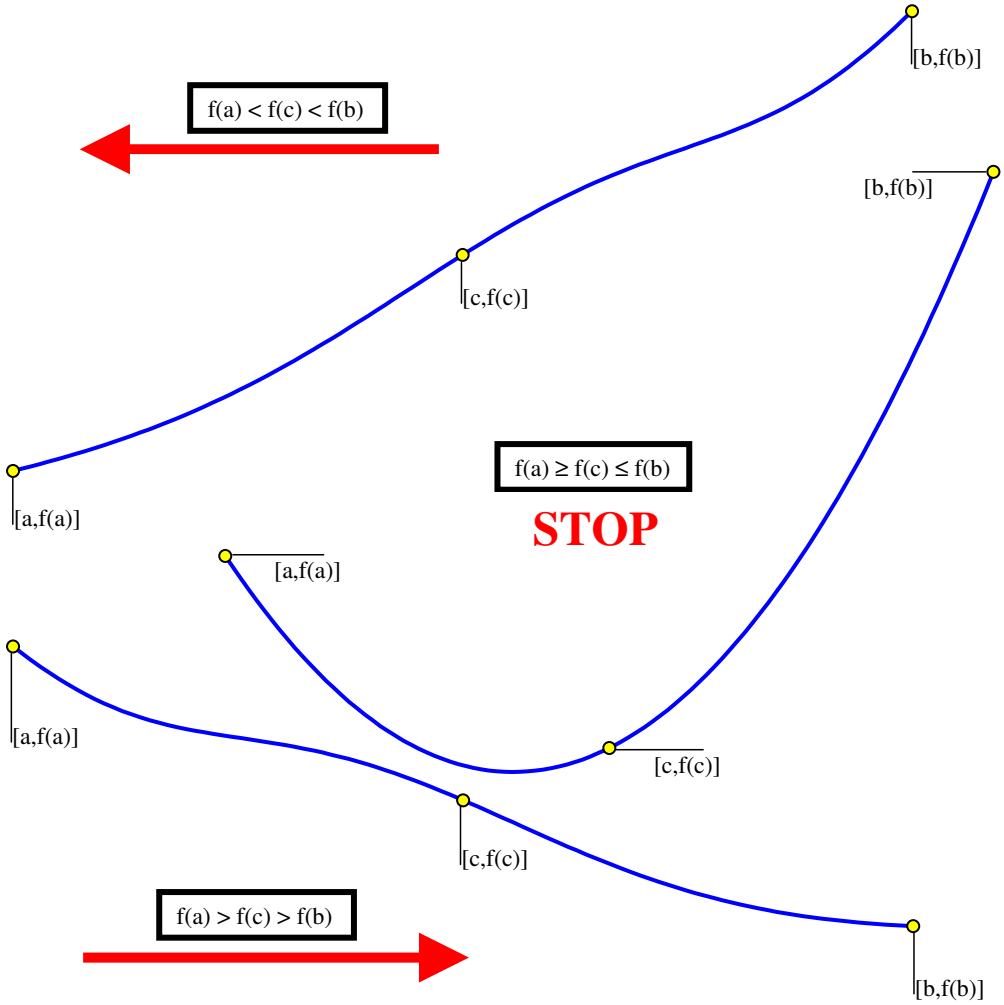
Začneme funkcí jedné proměnné. Pro praktické úlohy mají ovšem zásadní důležitost úlohy s (velmi) mnoha proměnnými. Řada metod (jak uvidíme) ovšem převádí řešení vícerozměrné úlohy na iterační řešení jednorozměrné úlohy. Nejdříve budeme hledat interval, na kterém má funkce jediné minimum.
Po funkci f budeme požadovat, aby byla unimodální na intervalu $[a, b]$. Tato podmínka zaručuje existenci jediného (globálního) minima v $x^* \in [a, b]$. Můžeme ji vyslovit bez jakéhokoliv předpokladu o hladkosti nebo dokonce spojitosti funkce, stačí pouze, aby byla definovaná v každém bodě intervalu $[a, b]$:

Funkce f je unimodální na intervalu $I = [a, b]$, resp. $I = (-\infty, \infty)$, právě když existuje jediné x^* tak, že pro libovolnou dvojici čísel $c, d \in I$, $c < d$, platí

$$c < d \leq x^* \Rightarrow f(c) > f(d); \;\; d > c \geq x^* \Rightarrow f(c) < f(d).$$

To znamená, že f je ostře klesající nalevo od x^* a napravo od x^* je naopak ostře rostoucí.

Existence jediného (globálního) minima není ovšem s unimodálností ekvivalentní. Pokud $[c, d] \subset [a, b]$ a f je unimodální na $[a, b]$, potom f je na $[c, d]$ unimodální, právě když obsahuje bod minima $x^* \in [c, d]$.
Z definice unimodální funkce plyne, že pokud najdu trojici různých bodů, na kterých se funkce nechová ostře monotonně, musí nutně mezi těmito body ležet bod minima x^*. Pro unimodální funkce na celé číselné ose bude proto fungovat následující algoritmus pro hledání počátečního intervalu obsahujícího bod minima, na tomto intervalu je tedy f opět unimodální.



Obrázek 7.1: Hledání intervalu, kde f(x) je unimodální

Vyjdeme z libovolného bodu x_0. Nastavíme krok h a akceleraci $ac \geq 1$. Spočteme funkční hodnoty v bodech $c = x_0, a = x_0 - h, b = x_0 + h$ a jejich porovnáním zjistíme, zda mezi

nimi leží minimum. Pokud ano, proceduru ukončíme, v opačném případě se posuneme o h=ac*h ve směru klesání funkčních hodnot a proceduru zopakujeme:

```
class Interval
{
  public double a,b;
  public Interval(double a, double b)
  {
    this.a=a; this.b=b;
  }
}

double f(double x)
{
  return x*x-x+1;
}

Interval MinInterval(float x0, float h, float ac)
{
  double a,c,b,fa,fc,fb;
  a = x0-h; c = x0; b = x0+h;
  fc = f(c); fa = f(a); fb = f(b);
  while(!(fa>=fc && fc<=fb))
  {
    h *= ac;
    if(fa < fc && fc < fb)
    {
      b = c; fb = fc;
      c = a; fc = fa;
      a -= h; fa = f(a);
    }
    else
    if(fa > fc && fc > fb)
    {
      a = c; fa = fc;
      c = b; fc = fb;
      b += h; fb = f(b);
    }
  }
  return new Interval(a,b);
}
```

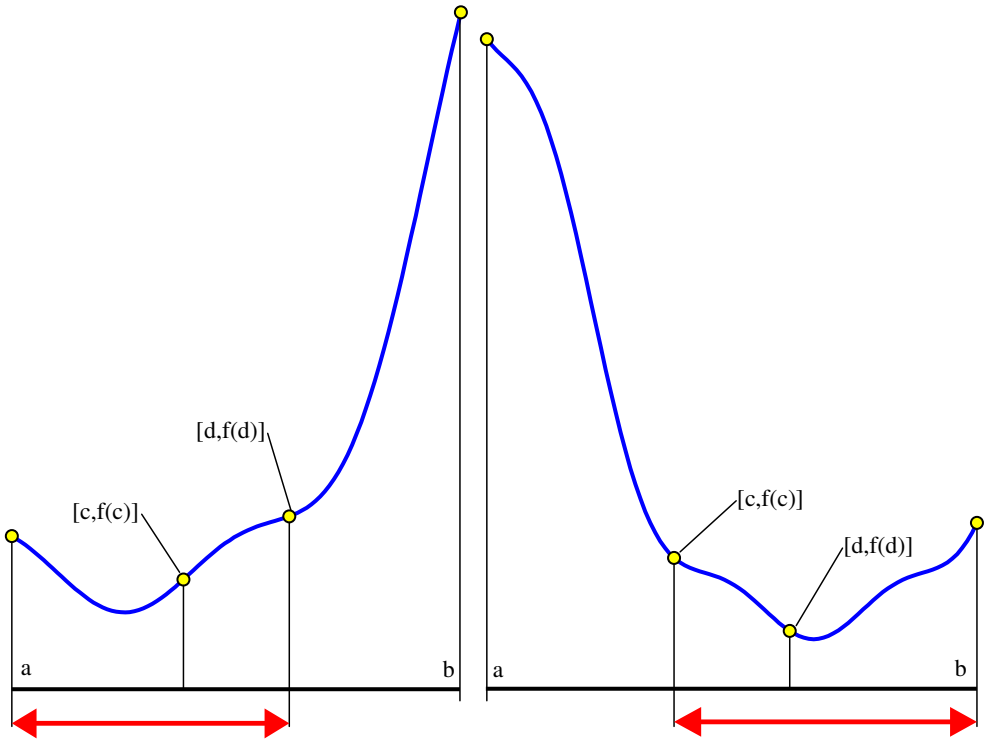
7.1.1 Metody intervalové redukce

Nyní popíšeme několik metod intervalové redukce, které mají blízký vztah k metodě půlení intervalu pro nalezení kořene nelineární rovnice. Vlastnost „f je unimodální na intervalu $[a,b]$" je přitom analogií vlastnosti „$f(a)f(b)<0$".

Předpokládejme, že f je unimodální na intervalu $[a,b]$. Potom lze porovnáním funkčních hodnot ve dvou různých vnitřních bodech $c,d \in [a,b]$ zjistit interval redukované délky, na

kterém je f opět unimodální. Nechť je např. $c < d$, potom [1]

$$f(c) \leq f(d) \quad \Rightarrow \quad x^* \in [a, d],$$
$$f(c) > f(d) \quad \Rightarrow \quad x^* \in [c, b].$$



Obrázek 7.2: Redukce intervalu

Délka nového intervalu nepřevýší $max\{d-a, b-c\}$. Dále se budeme zabývat symetrickým případem, kdy je $d - a = b - c = t(b - a)$. Odtud plyne

$$\begin{aligned} c &= b - t(b-a) = a + (1-t)(b-a), \\ d &= a + t(b-a). \end{aligned} \tag{7.2}$$

Podmínky $c < d$, c, d leží uvnitř $[a, b]$ implikují $0.5 < t < 1$.
Zvolíme-li například hodnotu t velmi blízko hodnotě 0,5, dostaneme metodu redukující délku intervalu téměř na polovinu za cenu výpočtu dvou funkčních hodnot v každém kroku. Existuje však dokonalejší analogie metody půlení intervalu:

Metoda půlení intervalu

[1] Nechť je např. $f(c) \leq f(d)$. Pokud by bylo $d < x^*$, muselo by podle definice unimodálnosti být $f(c) > f(d)$, což by byl spor s předpokladem.

V této metodě rozdělíme původní interval $[a, b]$ na čtyři stejné díly

$$a < c1 < c < c2 < b \qquad (7.3)$$
$$c = (b-a)/2,\ c1 = (c-a)/2,\ c2 = (b-c)/2.$$

Výběr redukovaného intervalu se řídí pravidly

$$\begin{aligned} f(c1) < f(c) &\Rightarrow x^* \in [a, c] \\ f(c2) > fc() &\Rightarrow x^* \in [c, b] \\ f(c1) \geq f(c) \leq f(c2) &\Rightarrow x^* \in [c1, c2] \end{aligned}$$

Nový interval má přesně poloviční délku (odtud název metody) a navíc není potřeba znovu počítat hodnotu funkce v novém středu intervalu (bod c). Každý krok nás stojí výpočet dvou funkčních hodnot.

```
Interval MinBisection(Interval I, double tol)
{
  double a,b,c,c1,c2,h,fc,fc1,fc2;
  a = I.a; b = I.b¨;h = (b-a)/4;
  c = a + 2*h; c1 = a + h; c2 = b - h;
  fc = f(c);fc1 = f(c1); fc2 = f(c2);
  while(b-a>tol)
  {
    h /= 2;
    if(fc1<fc)
    {
      b=c; c=c1; fc=fc1;
    }
    else
    if(fc2>fc)
    {
      a=c1; b=c2;
    }
    else
    {
      a=c; c=c2; fc=fc2;
    }
    c1=a+h; fc1=f(c1);
    c2=b-h; fc2=f(c2);
  }
  return new Interval(a,b);
}
```

Metoda zlatého řezu

Pokusme se určit t ve vztazích 7.2, abychom mohli, podobně jako v metodě půlení intervalu, znovu použít funkční hodnotu z předchozího kroku. Pokud bude $t < \sqrt{0.5} = 0.707106\ldots$, získáme tak efektivnější metodu. Platí

$$d - c = a + t(b-a) - [a + (1-t)(b-a)] = (2t-1)(b-a).$$

Pokud chci znovu použít bod c, musí platit

$$d - c = t^2(b - a) \quad \text{nebo} \quad d - c = t(1 - t)(b - a).$$

První možnost vede na kvadratickou rovnici

$$t^2 - 2t + 1 = 0,$$

jejíž řešení $t = 0.5$ nevyhovuje podmínkám. Druhá možnost odpovídá rovnici

$$t^2 + t - 1 = 0, \tag{7.4}$$

jejíž řešení

$$\tau = \frac{-1 + \sqrt{5}}{2} = 0.618033988749\ldots\,, \tag{7.5}$$

nazývající se **poměr zlatého řezu**, dává této metodě název.

```
Interval MinGoldenSection(Interval I, double tol)
{
  const double t = 0.61803398874989484820458683436564;
  double a,b,c,d,fc,fd;
  a = I.a; b = I.b;
  c = a + (1-t)*(b-a);
  d = a + t*(b-a);
  fc = f(c); fd = f(d);
  while(b-a>tol)
  {
    if(fc<=fd)
    {
      b=d;
      d=c;fd=fc;
      c = a + (1-t)*(b-a);fc = f(c);
    }
    else
    {
      a=c;
      c=d;fc=fd;
      d = a + t*(b-a);fd = f(d);
    }
  }
  return new Interval(a,b);
}
```

Fibonacciova metoda

Je-li dopředu znám počet kroků, je o něco efektivnější tzv. Fibonacciova metoda. Fibonacciova čísla jsou definována rekurentním vztahem

$$F_0 = 1,\ F_1 = 1;\ F_n = F_{n-1} + F_{n-2}. \tag{7.6}$$

Začátek posloupnosti vypadá takto

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 \ldots$$

Označme

$$t_n = \frac{F_n}{F_{n+1}}, \ n = 1, 2, \ldots .$$

Budeme tedy pracovat s posloupností

$$\frac{1}{2}, \frac{2}{3}, \frac{3}{5}, \frac{5}{8}, \frac{8}{13}, \frac{13}{21}, \frac{21}{34}, \frac{34}{55}, \frac{55}{89} \ldots$$

Pro počítání prvních N poměrů Fibonacciových čísel můžeme použít následující třídu, která si uchovává Fibonacciova čísla v členské proměnné F typu pole. Rekurentní vztah 7.6 lze rovněž vyřešit. Funkce generující Fibonacciova čísla vypadá takto

$$F_n = \frac{5+\sqrt{5}}{10}\left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{5-\sqrt{5}}{10}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

```
class Fibonacci
{
  public long[] F;
  int N;
  const double c1=0.27639320225002103035908263312687;
  const double c2=1-c1;
  const double r1=-0.61803398874989484820458683436564;
  const double r2=1-r1;

  public double FN(int i)
  {
    return c1*Math.Pow(r1,i)+c2*Math.Pow(r2,i);
  }

  public Fibonacci(int N)
  {
    this.N = N;
    F=new long[N+1];
    F[0]=1;
    F[1]=2;
    for(int i=2;i<=N;i++)
      F[i]=F[i-1]+F[i-2];
  }

  public double ta(int k)
  {
    return (double)(F[k])/F[k+1];
  }

  public double t(int k)
  {
```

```
    return FN(k)/FN(k+1);
  }
}
```

Platí

$$t_{n-1}t_n = \frac{F_{n-1}}{F_n}\frac{F_n}{F_{n+1}} = \frac{F_{n-1}}{F_{n+1}} = \frac{F_{n+1}-F_n}{F_{n+1}} = 1 - t_n. \tag{7.7}$$

Nechť n je známý počet kroků. Spočítáme $t = t_n$ a uděláme krok intervalové redukce 7.2. V dalším kroku zvolíme $t = t_{n-1}$ a obecně v k-tém kroku zvolíme $t = t_{n-k}$, poslední krok provedeme s hodnotou $t = t_1 = \frac{1}{2}$. Ze vztahu 7.7 plyne, že v k-tém kroku můžeme použít již dříve vypočtenou hodnotu z kroku $k-1$. Jeden krok metody nás stojí výpočet jedné funkční hodnoty, jako u metody zlatého řezu. Poslední krok vyžaduje speciální ošetření, protože body c a d splynou. Vezmeme bod $c + \epsilon$, kde ϵ je dostatečně malé, aby příliš nezměnilo kvalitu poslední redukce ($t_1 = \frac{1}{2}$) a současně ještě tak velké, aby šlo spolehlivě rozlišit mezi $f(c)$ a $f(c + \epsilon)$.

```
Interval MinFibonacci(Interval I, int n)
{
  const double epsilon=1E-10;
  double t=Fibonacci.t(n);
  double a,b,c,d,fc,fd;
  a = I.a; b = I.b;
  c = a + (1-t)*(b-a);
  d = a + t*(b-a);
  fc = f(c); fd = f(d);
  for(int k=1;k<n;k++)
  {
    t = Fibonacci.t(n-k);
    if(fc<=fd)
    {
      b=d;
      d=c;fd=fc;
      c = a + (1-t)*(b-a);fc = f(c);
    }
    else
    {
      a=c;
      c=d;fc=fd;
      d = a + t*(b-a);fd = f(d);
    }
  }
  d=c+epsilon;
  fd=f(d);
  if(fc<=fd)
    b=d;
  else
    a=c;
  return new Interval(a,b);
}
```

Ze vztahu 7.7 plyne

$$t_n t_{n+1} = 1 - t_{n+1} \Rightarrow t_{n+1} = \tfrac{1}{1+t_n},$$
$$t_{n+2} = \tfrac{1}{1+t_{n+1}} = \tfrac{1+t_n}{2+t_n}, \tag{7.8}$$
$$t_{n+2} - t_n = \tfrac{1+t_n}{2+t_n} - \tfrac{1+t_{n-2}}{2+t_{n-2}} = \tfrac{t_n - t_{n-2}}{(2+t_n)(2+t_{n-2})},\ (2+t_n)(2+t_{n-2}) > 0. \tag{7.9}$$

Protože je

$$\begin{aligned} t_3 - t_1 &= \frac{3}{5} - \frac{1}{2} = \frac{1}{10} > 0, \\ t_4 - t_2 &= \frac{5}{8} - \frac{2}{3} = -\frac{1}{24} < 0, \end{aligned}$$

plyne z 7.9 indukcí

$$t_{2n-1} < t_{2n+1}, \tag{7.10}$$
$$t_{2n} > t_{2n+2}. \tag{7.11}$$

Protože je zřejmě posloupnost t_n ohraničená,

$$0 < t_n < 1,$$

existují konečné limity

$$\lim_{n\to\infty} t_{2n-1} = \tau_1,\ \lim_{n\to\infty} t_{2n} = \tau_2.$$

Limitním přechodem v 7.8 dostaneme

$$\tau_i = \frac{1+\tau_i}{2+\tau_i} \Leftrightarrow \tau_i^2 + \tau_i - 1 = 0.$$

Obě dvě limity proto splývají s kladným kořenem kvadratické rovnice 7.4, a tudíž

$$\lim_{n\to\infty} t_n = \tau = \frac{-1+\sqrt{5}}{2} = 0.618033988749\ldots$$

Pro velká n je tedy krok Fibonacciovy metody prakticky shodný s metodou zlatého řezu. Celková redukce délky intervalu je po n krocích podle 7.7

$$t_1 t_2 \ldots t_n = \begin{cases} (1-t_2)(1-t_4)\ldots(1-t_n) & \text{pro n sudé,} \\ (1-t_2)(1-t_4)\ldots(1-t_{n-1})t_n & \text{pro n liché.} \end{cases}$$

Z 7.10 plyne

$$\begin{aligned} t_{2n} > \tau \Rightarrow 1 - t_{2n} &< 1 - \tau = \tau^2, \\ t_{2n-1} &< \tau. \end{aligned}$$

V každém případě je výsledný interval, který obdržíme Fibonacciovou metodou, o něco kratší, než ten který poskytne metoda zlatého řezu:

$$t_1 t_2 \ldots t_n = \frac{1}{F_{n+1}} < \tau^n.$$

Následující tabulka porovnává efektivitu metody zlatého řezu s Fibonacciovou metodou. Rozdíl je sice znatelný, ale z hlediska rychlosti konvergence bezvýznamný. Vyjdeme-li z intervalu délky jedna, budeme potřebovat pro dosažení délky $< 0,01$ u obou metod 10 kroků:

n	$t_1 t_2 \dots t_n$	τ^n	$t_1 t_2 \dots t_n / \tau^n$
1	0,5	0,618033988749895	0,809016994374947
2	0,333333333333333	0,381966011250105	0,872677996249965
3	0,2	0,23606797749979	0,847213595499958
4	0,125	0,145898033750316	0,85676274578121
5	0,0769230769230769	0,0901699437494743	0,853089995673036
6	0,0476190476190476	0,0557280900008412	0,854489138571388
7	0,0294117647058823	0,034441853748633	0,853954172169077
8	0,0181818181818182	0,0212862362522082	0,854158432068141
9	0,0112359550561798	0,0131556174964248	0,854080400196588
10	0,00694444444444444	0,00813061875578336	0,854110204036417
15	0,000626174076393237	0,000733137435857406	0,854101899272031
20	5,64620857094461E-05	6,61069613518961E-05	0,854101966794252

7.1.2 Metody interpolace

Tyto metody nemusí vždy konvergovat. Budeme podobně jako např. u metody sečen potřebovat dobrou počáteční aproximaci minima.

Metoda kvadratické (parabolické) interpolace Jedná se vlastně o analogii metody sečen pro optimalizační úlohu. V okolí bodu x_k nahradíme zadanou funkci kvadratickým interpolačním polynomem P (polynom prvního stupně nemá minimum) a novou hodnotu x_{k+1} získáme jako bod minima polynomu P.

Bude se nám proto hodit obecný vzorec pro minimum kvadratického interpolačního polynomu. Uvažujme proto následující interpolační úlohu:

i	0	1	2
x_i	a	b	c
y_i	$f(a)$	$f(b)$	$f(c)$

Interpolační polynom můžeme psát ve tvaru

$$P = f(b) + c_1(x - b) + c_2(x - b)^2.$$

Podmínky interpolace

$$\begin{aligned} P(b) &= f(b), \\ P(a) &= f(b) + c_1(a - b) + c_2(a - b)^2 = f(a), \\ P(c) &= f(b) + c_1(c - b) + c_2(c - b)^2 = f(c), \end{aligned}$$

generují soustavu dvou lineárních rovnic:

$$\left[\begin{array}{cc|c} a-b & (a-b)^2 & f(a)-f(b) \\ c-b & (c-b)^2 & f(c)-f(b) \end{array}\right]$$

Bod minima x^* je dán vztahem

$$\begin{aligned} \frac{dP}{dx} &= c_1 + 2c_2(x^* - b) = 0 \\ x^* &= b - \frac{1}{2}\frac{c_1}{c_2} \end{aligned}$$

Koeficienty c_1, c_2 najdeme pomocí Cramerova pravidla

$$\begin{aligned} x^* &= b - \frac{1}{2}\frac{\frac{D_1}{D}}{\frac{D_2}{D}} = b - \frac{1}{2}\frac{D_1}{D_2} = \\ &= b - \frac{1}{2}\frac{\left|\begin{array}{cc} f(a)-f(b) & (a-b)^2 \\ f(c)-f(b) & (c-b)^2 \end{array}\right|}{\left|\begin{array}{cc} a-b & f(a)-f(b) \\ c-b & f(c)-f(b) \end{array}\right|} = \\ &= b - \frac{1}{2}\frac{(c-b)^2[f(a)-f(b)] - (a-b)^2[f(c)-f(b)]}{(a-b)[f(c)-f(b)] - (c-b)[f(a)-f(b)]} \end{aligned}$$

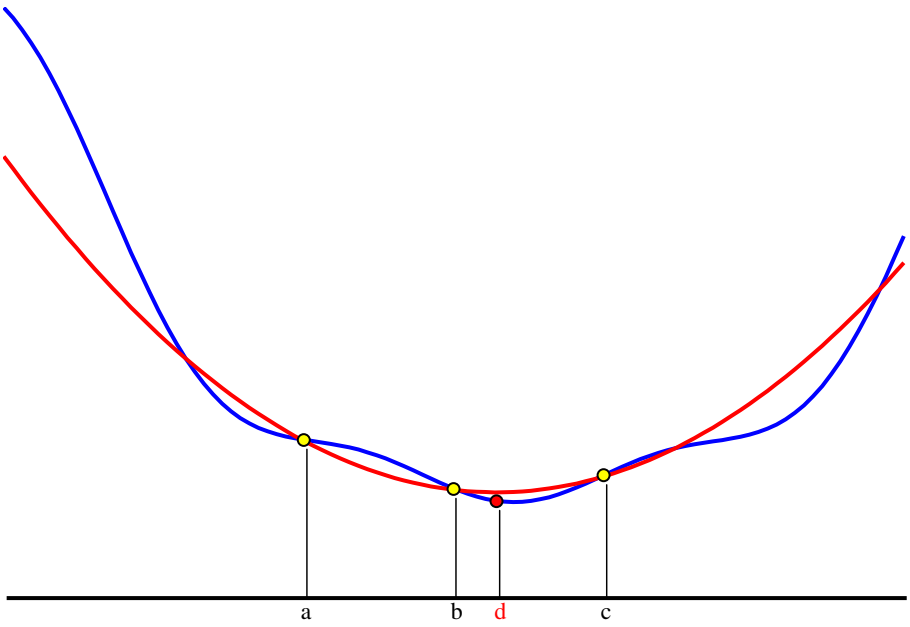
a upravíme

$$\begin{aligned} x^* &= \frac{1}{2}\frac{2(ab-b^2)f(c) + 2(bc-ab)f(b) + 2(b^2-bc)f(a)}{(a-b)f(c) + (c-a)f(b) + (b-c)f(a)} + \\ &+ \frac{1}{2}\frac{(a^2-2ab+b^2)[f(c)-f(b)] - (c^2-2bc+b^2)[f(a)-f(b)]}{(a-b)f(c) + (c-a)f(b) + (b-c)f(a)} = \\ &= \frac{1}{2}\frac{(a^2-b^2)f(c) + (c^2-a^2)f(b) + (b^2-c^2)f(a)}{(a-b)f(c) + (c-a)f(b) + (b-c)f(a)} \end{aligned}$$

```
class ABCD
{
  public double a,b,c,d,fa,fb,fc,fd;
}

double xmin(ABCD u)
{
  double ca,cb,cc,c2a,c2b,c2c;
  ca = u.b-u.c; c2a = u.b*u.b-u.c*u.c;
  cb = u.c-u.a; c2b = u.c*u.c-u.a*u.a;
```

```
  cc = u.a-u.b; c2c = u.a*u.a-u.b*u.b;
  return 0.5*(c2a*u.fa+c2b*u.fb+c2c*u.fc)/(ca*u.fa+cb*u.fb+cc*u.fc);
}

ABCD select(ABCD u)
{
  ABCD v = new ABCD();
  if(u.d<=u.a)
  {
    v.a=u.d; v.b=u.a; v.c =u.b;
    v.fa=u.fd; v.fb=u.fa; v.fc=u.fb;
    return v;
  }
  else
    if(u.d<=u.b)
  {
    v.a=u.a; v.b=u.d; v.c=u.b;
    v.fa=u.fa; v.fb=u.fd; v.fc=u.fb;
    return v;
  }
  else
  {
    v.a=u.b; v.b=u.d; v.c=u.c;
    v.fa=u.fb; v.fb=u.fd; v.fc=u.fc;
    return v;
  }
}

double MinParabola(double x0, double h, double tol)
{
  double fl,fr,d1,d2;
  ABCD u=new ABCD();
  fl = f(x0);
  fr = f(x0+h);
  if(fl<fr)
  {
    u.a = x0-h; u.fa = f(u.a);
    u.b = x0; u.fb = fl;
    u.c = x0+h; u.fc = fr;
  }
  else
  {
    u.a = x0; u.fa = fl;
    u.b = x0+h; u.fb = fr;
    u.c = x0+2*h; u.fc = f(u.c);
  }
  d2 = xmin(u);
  u.d=d2; u.fd=f(d2);
  do
  {
    d1=d2;
    u=select(u);
    d2 = xmin(u);
```

```
      u.d=d2; u.fd=f(d2);
    }
    while(Math.Abs(d1-d2)>tol);
    return d2;
}
```



Obrázek 7.3: Metoda parabolické interpolace

Newtonova Metoda

Místo Lagrangeova interpolačního polynomu můžeme použít Taylorův polynom druhého stupně

$$T = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2.$$

Najdeme bod minima x^{k+1} z rovnice

$$\begin{aligned} T'(x_{k+1}) &= f'(x_k) + f''(x_k)(x_{k+1} - x_k) = 0, \\ x_{k+1} &= x_k - \frac{f'(x_k)}{f''(x_k)}. \end{aligned} \tag{7.12}$$

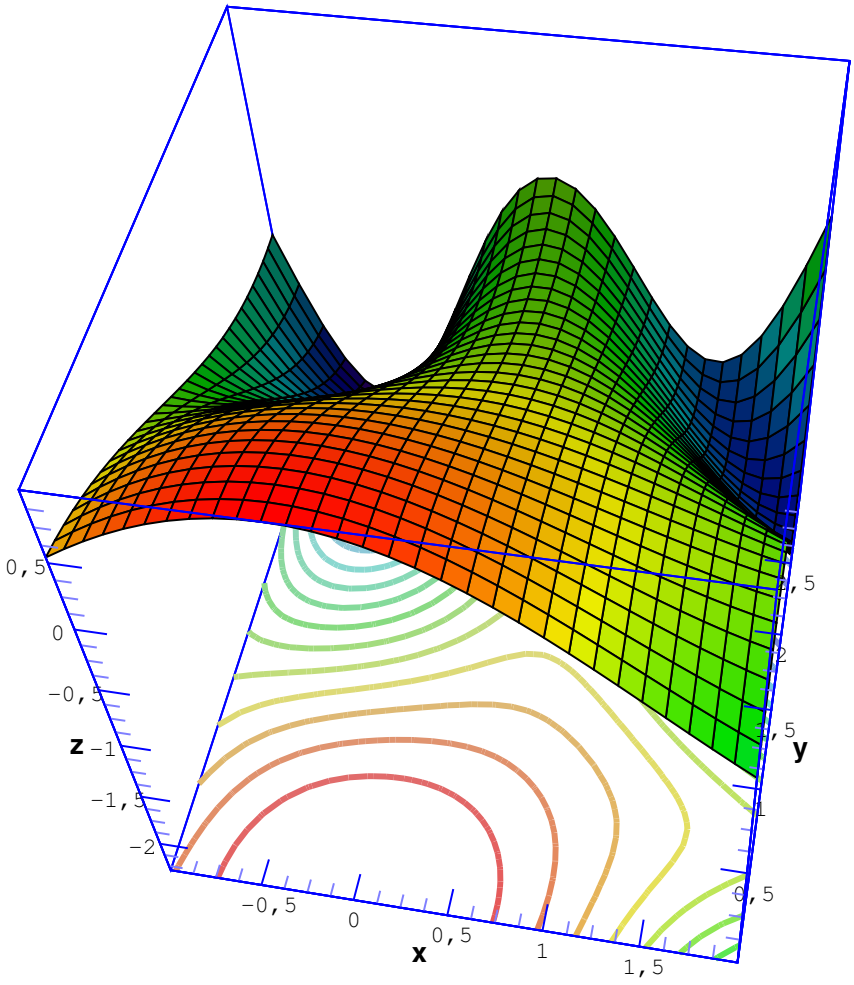
Vztah 7.12 je totožný s Newtonovou iterační formulí pro rovnici určující stacionární bod (rovnice 7.1 v jednorozměrném případě).

7.2 Vícerozměrné algoritmy

V případě funkce n reálných proměnných nebudeme specifikovat přesně podmínky, za kterých fungují dané algoritmy. Rovněž nebudeme hledat globální minimum, spokojíme se zpravidla s aproximací „nejbližšího" lokálního minima. Na obrázku 7.4 vidíme trojrozměrný graf funkce dvou proměnných (spolu s vrstevnicemi):

$$f(x) = sin(xy) + cos(x + y) \quad [x, y] \in [-1, 2] \times [0, 3]. \tag{7.13}$$

Na této funkci budeme předvádět následující algoritmy.



Obrázek 7.4: Funkce $f(x) = sin(xy) + cos(x + y)$, kterou minimalizujeme

7.2.1 Simplexová metoda [Nelder, Mead 1965]

Tento algoritmus nevyžaduje počítání derivací. Popíšeme jej jen v nejjednodušší variantě pro $n = 2$. Vyjdeme z bodu $\mathbf{x}_0$ a sestrojíme rovnostranný trojúhelník ΔABC tak, aby se

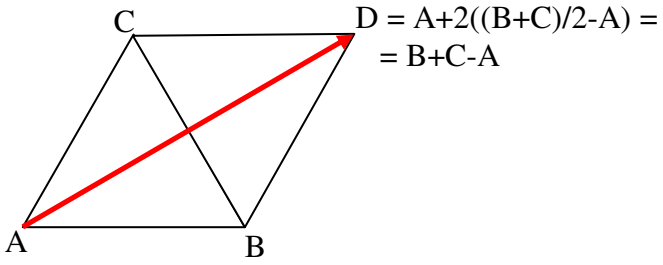
$T = \mathbf{x}_0$ stal jeho těžištěm. Najdeme vrchol P_{max} trojúhelníka ΔABC, v němž funkce $f(x)$ nabývá maximální hodnoty

$$f(P_{max}) \geq f(P) \quad \forall P \in V = \{A, B, C\}.$$

Sestrojíme reflexi tohoto bodu vzhledem ke zbývajícím bodům.

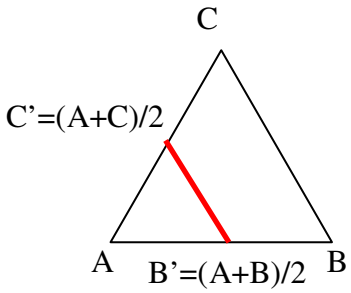
$$D = \frac{2}{|V^*|} \sum_{P \in V^*} P - P_{max}, \quad \text{kde } V^* = V - P_{max}.$$

Za předpokladu, že $P_{max} = A$, vidíme situaci na obrázku 7.5.



Obrázek 7.5: Reflexe

Pokud je $f(D) \leq f(A)$, pokračujeme trojúhelníkem ΔBDC a tento krok nazveme reflexe. V opačném případě, tj. pro $f(D) > f(A)$, provedeme kontrakci: trojúhelník ΔABC nahradíme trojúhelníkem $\Delta AB'C'$ podle obrázku 7.6.

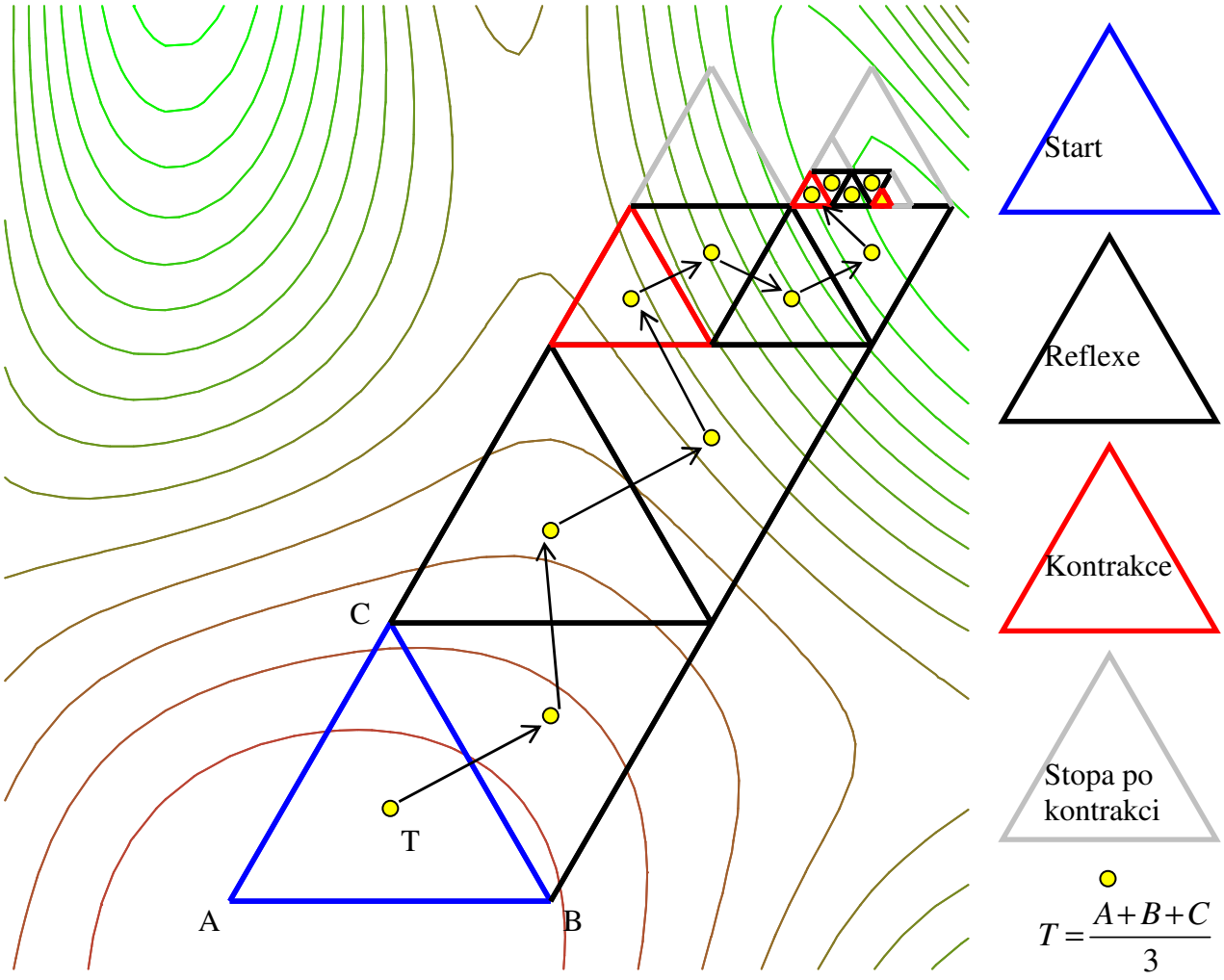


Obrázek 7.6: Kontrakce

Na obrázku 7.7 vidíme několik iterací simplexové metody, za aproximaci minima můžeme považovat těžiště postupně vznikajících trojúhelníků.

7.2.2 Metody redukce dimenze

Následující dvě metody není problém popsat v libovolné dimenzi. Jejich principem je redukce n-rozměrného problému na posloupnost jednorozměrných úloh. Pokud je $\mathbf{x}^i$ vektor i-té aproximace minima, setrojíme následující aproximaci $\mathbf{x}^{i+1}$ řešením (přesným nebo přibližným) jednorozměrné minimalizační úlohy:



Obrázek 7.7: Simplexová metoda pro funkci 7.13

Minimalizujte funkci

$$\varphi(t) = f(\mathbf{x}^i + t\mathbf{s}(\mathbf{x}^i)).$$

Argumentem funkce f je lineární funkce (přímka). Proměnná t je parametr a vektor $\mathbf{s}$ je směrový vektor, který je obecně závislý na $\mathbf{x}^i$. Dobrou představu získáme ve dvou dimenzích. Zúžení f na φ je jakýsi přímočarý řez terénem.
Je-li t^i bod, ve kterém $\varphi(t)$ nabývá svého minima, sestrojíme další aproximaci původního problému takto

$$\mathbf{x}^{i+1} = \mathbf{x}^i + t^i\mathbf{s}(\mathbf{x}^i).$$

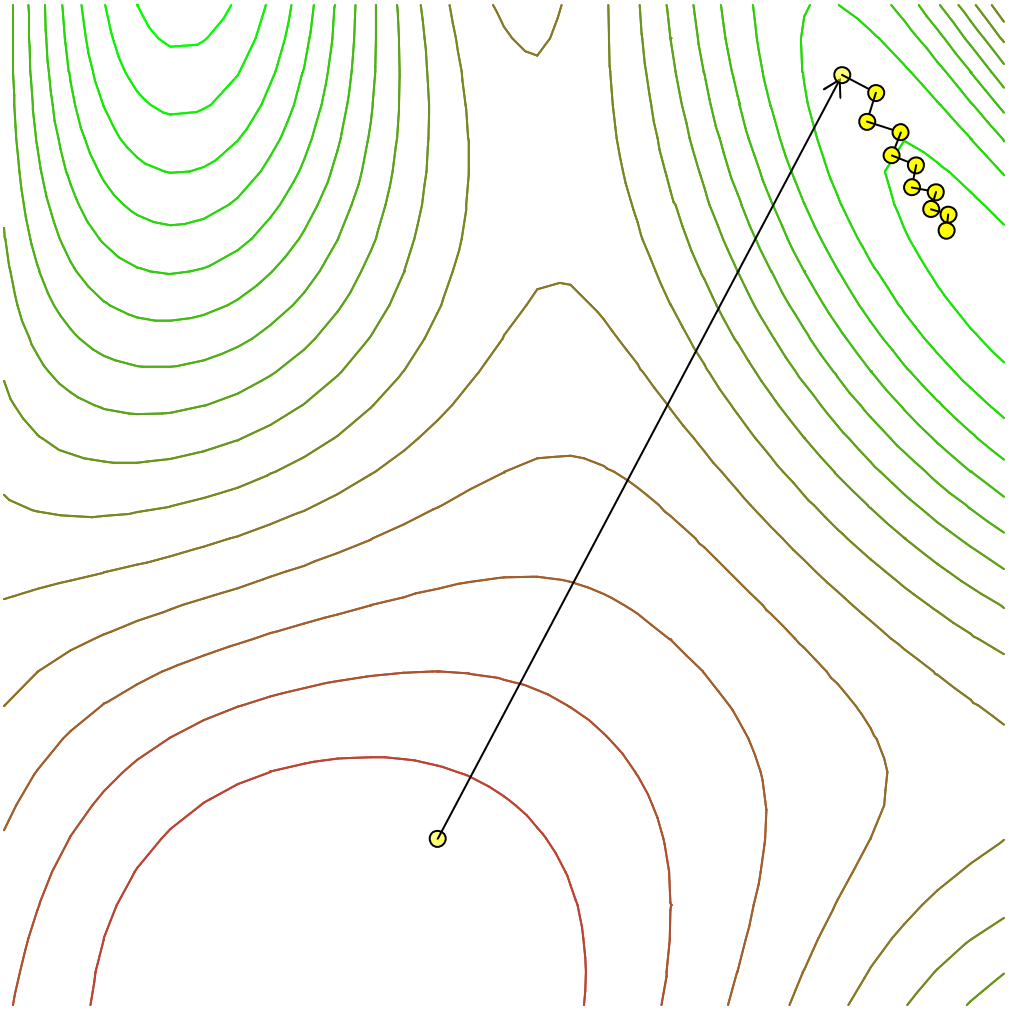
Metoda souřadnicových směrů

Metodě se také může říkat metoda střídavých směrů, protože vektor $\mathbf{s}$ závisí pouze na i:

$$\mathbf{s}(\mathbf{x}^i) = e_{(i \bmod n)+1}.$$

Zde je n dimenze problému, *a mod b* operátor "zbytek po dělení" a e_i jednotkový vektor v směru i-té souřadnice. V případě $n = 3$ střídáme vektory

$$\mathbf{s}^0 = \mathbf{e}_1 = (1,0,0),\ \mathbf{s}^1 = \mathbf{e}_2 = (0,1,0),\ \mathbf{s}^2 = \mathbf{e}_3 = (0,0,1),$$
$$\mathbf{s}^3 = \mathbf{e}_1 = (1,0,0),\ \mathbf{s}^4 = \mathbf{e}_2 = (0,1,0),\ \mathbf{s}^5 = \mathbf{e}_3 = (0,0,1),$$
$$\dots$$



Obrázek 7.8: Metoda největšího spádu pro funkci 7.13

Metoda největšího spádu

V této metodě se volí

$$\mathbf{s}(\mathbf{x}^i) = -\mathbf{grad}(\mathbf{x}^i) = -(f_1'(\mathbf{x}^i), \ldots, f_n'(\mathbf{x}^i)).$$

Přestože zejména metoda nejmenšího spádu vypadá velmi nadějně, ve většině případů vede k velmi pomalé konvergenci. Proto je potřeba ji po několika krocích nahradit za rychle (ale pouze lokálně) konvergentní Newtonovu metodu. Na obrázku 7.8 vidíme několik iterací metody největšího spádu.

7.2.3 Newtonova metoda

Tato metoda není nic jiného, než Newtonova metoda pro řešení soustav nelineárních rovnic aplikovaná na rovnice pro stacionární bod 7.1. Matice soustavy lineárních rovnic, kterou je třeba v každém kroku řešit, je zde generována tzv. Hessovou maticí

$$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ & & \ddots & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

Newtonova metoda má tvar

$$\nabla^2 f \delta^i = -\mathbf{grad}(\mathbf{x}^i) = -(f_1'(\mathbf{x}^i), \ldots, f_n'(\mathbf{x}^i)),$$

kde

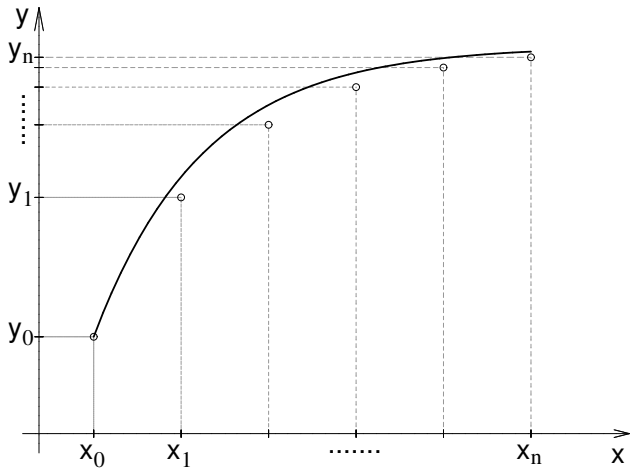
$$\delta^i = \mathbf{x}^{i+1} - \mathbf{x}^i = (\delta_1^i, \ldots, \delta_n^i)^T.$$

8 Numerické řešení diferenciálních rovnic

Cíl kapitoly

Pomocí diferenciálních rovnic jsou popsány nejrůznější fyzikální děje. Ve třetím semestru se studenti seznámili s některými typy rovnic, jejichž řešení lze nalézt analyticky. V praktických problémech se však vyskytují i složitější rovnice. Některé z nich jsou analyticky řešitelné jen obtížně a některé analyticky vyřešit nelze. Proto se k jejich řešení používají metody přibližné, z nichž některé nyní popíšeme.
Nejprve se zaměříme na metody pro řešení jedné diferenciální rovnice prvního řádu se zadanou počáteční podmínkou - počáteční úlohy. Potom ukážeme, jak tyto metody zobecnit pro řešení soustavy diferenciálních rovnic prvního řádu. Předvedeme, že diferenciální rovnice vyšších řádů se zadanými počátečními podmínkami lze snadno převést na soustavy diferenciálních rovnic prvního řádu. V závěru se budeme zabývat řešením okrajových úloh, kde jsou předepsány hodnoty řešení na počátku a na konci zkoumaného intervalu.

Společným znakem všech dále uvedených metod je, že řešení nehledáme jako spojitou funkci, definovanou na celém zkoumaném intervalu $\langle a, b\rangle$, ale hodnoty přibližného řešení počítáme pouze v konečném počtu bodů $a = x_0 < x_1 < \cdots < x_n = b$. Těmto bodům říkáme **uzlové body** nebo **uzly sítě** a množině $\{x_0, x_1, \ldots, x_n\}$ říkáme **síť**. Rozdíl $h_i = x_{i+1} - x_i$ se nazývá **krok** sítě v uzlu x_i.
Přibližné hodnoty řešení v uzlových bodech, vypočtené nějakou numerickou metodou, budeme značit $y_0, y_1, \ldots, y_n$, na rozdíl od hodnot přesného řešení, které budeme značit $y(x_0), y(x_1), \ldots, y(x_n)$.
Na obrázku 8.1 vidíme přesné řešení diferenciální rovnice, které je vykresleno plnou černou čarou a přibližné hodnoty řešení v uzlových bodech, vyznačené kroužky.



Obrázek 8.1: Přesné a přibližné řešení diferenciální rovnice

V příkladu z obrázku 8.1 byla použita **pravidelná (ekvidistantní) síť** - krok h mezi jednotlivými uzly byl konstantní.

Všude v dalším textu, nebude-li výslovně uvedeno jinak, budeme pracovat s pravidelnými sítěmi.

Chceme-li znát přibližnou hodnotu řešení v jiném než uzlovém bodě, můžeme použít některou z interpolačních metod, popsaných v kapitole 5, např. nahradit řešení lomenou čarou procházející vypočtenými body.

8.1 Počáteční úlohy

Nejprve se budeme zabývat řešením obyčejné diferenciální rovnice prvního řádu se zadanou počáteční podmínkou

$$y' = f(x, y) \quad , \quad y(x_0) = y_0 \tag{8.1}$$

Připomeňme podmínky, které zajistí existenci a jednoznačnost řešení úlohy 8.1.

Věta 8.1 *Je-li funkce* $f(x, y)$ *spojitá na obdélníku* $R = \{(x, y); |x - x_0| \leq a, |y - y_0| \leq b\}$, $a > 0, b > 0$, *pak existuje řešení počáteční úlohy 8.1 na intervalu* $\langle x_0 - \alpha, x_0 + \alpha \rangle$, *kde* $\alpha = \min(a, \frac{b}{M})$, $M = \max_R |f(x, y)|$
Je-li dále funkce $\frac{\partial f(x,y)}{\partial y}$ *ohraničená na obdélníku* R, *pak toto řešení je jediné.*

Tato věta však udává pouze postačující podmínky pro existenci jediného řešení. Také v mnoha případech zaručuje existenci a jednoznačnost řešení pouze na velmi malém okolí bodu x_0. Při řešení konkrétního matematického modelu technické úlohy proto existenci a jednoznačnost řešení posuzujeme i na základě informací o řešené úloze, případně fyzikálních vlastností hledaného řešení.
V dalším textu vysvětlíme několik obecných pojmů týkajících se numerických metod řešení diferenciálních rovnic, ale nejprve ukážeme nejjednodušší z těchto metod, aby čtenář získal konkrétní představu, jak numerické řešení diferenciálních rovnic může vypadat.

8.1.1 Eulerova metoda

Mějme dánu počáteční úlohu 8.1 a pravidelnou síť $\{x_0, x_1, \ldots, x_n\}$ s krokem h. Ve všech bodech sítě by podle rovnice 8.1 mělo platit

$$y'(x_i) = f(x_i, y(x_i))$$

Derivaci na levé straně této rovnice můžeme nahradit diferencí podle jednoho ze vzorců 6.2. Dostaneme

$$\frac{y(x_{i+1}) - y(x_i)}{h} \doteq f(x_i, y(x_i))$$

Nahradíme-li $y(x_i)$ přibližnou hodnotou y_i, můžeme odtud vyjádřit přibližnou hodnotu $y(x_{i+1})$ jako

$$y_{i+1} = y_i + hf(x_i, y_i) \tag{8.2}$$

Pomocí tohoto vzorce vypočteme přibližnou hodnotu řešení v dalším uzlovém bodě pomocí hodnoty v uzlu předchozím. Hodnotu řešení v bodě x_0 známe z počáteční podmínky, je rovna y_0.

Příklad 8.1 *Eulerovou metodou s krokem* $h = 0,1$ *řešte počáteční úlohu*

$$y' = x^2 - y \quad , \quad y(0) = 1$$

na intervalu $\langle 0\,;\, 0,5\rangle$.

Řešení: V našem případě je $x_0 = 0$, $y_0 = 1$ a $f(x, y) = x^2 - y$. Přibližné hodnoty řešení v dalších bodech budeme počítat podle vzorce 8.2, konkrétně

$$y_{i+1} = y_i + 0,1 \cdot (x_i^2 - y_i) \quad , \quad i = 0, \ldots, 4$$

Vypočtené hodnoty zapíšeme do tabulky. Pro srovnání jsou v tabulce uvedeny i hodnoty přesného řešení $y = -e^{-x} + x^2 - 2x + 2$ v uzlových bodech. Všechna čísla v tabulce jsou zaokrouhlena na 4 desetinná místa.

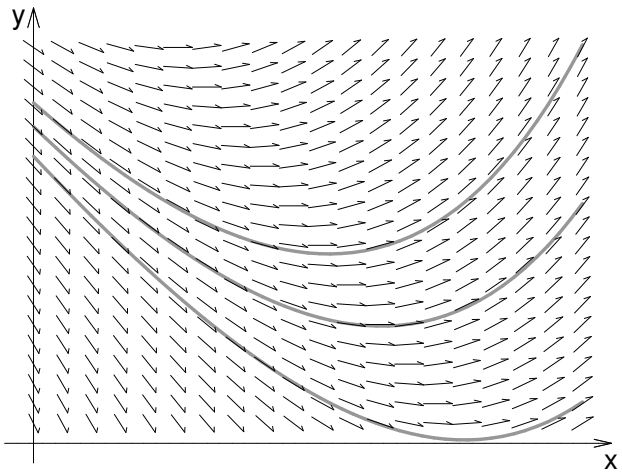
i	0	1	2	3	4	5
x_i	0	0,1	0,2	0,3	0,4	0,5
y_i	1	0,9	0,811	0,7339	0,6695	0,6186
$y(x_i)$	1	0,9052	0,8213	0,7492	0,6897	0,6435

Geometrická interpretace Eulerovy metody

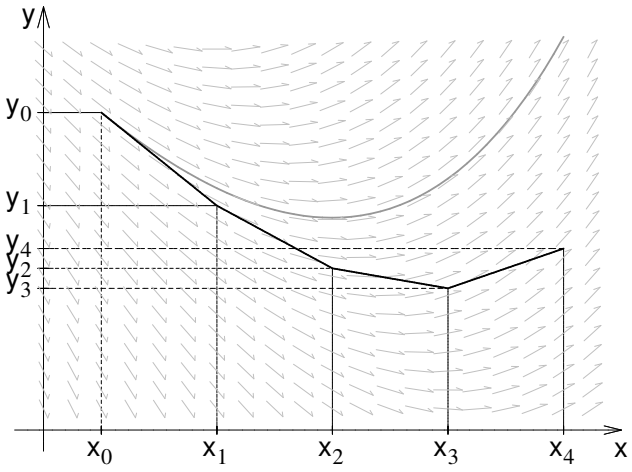
Pro vysvětlení geometrické interpretace Eulerovy metody připomeňme nejprve, že diferenciální rovnicí $y' = f(x, y)$ je dáno tzv. směrové pole. V každém bodě $[x, y]$ roviny (x, y), kterým prochází některé řešení této rovnice, je hodnota $f(x, y)$ rovna směrnici tečny ke grafu tohoto řešení. Proto si směrové pole můžeme, zhruba řečeno, představit tak, že v každém bodě roviny (x, y) stojí šipka, která říká, kterým směrem máme pokračovat, dostaneme-li se do tohoto bodu. Na obrázku 8.2 vidíme směrové pole příslušné jisté diferenciální rovnici a několik řešení téže rovnice.
Při řešení diferenciální rovnice Eulerovou metodou postupujeme vlastně takto:
Vyjdeme z bodu $[x_0, y_0]$ směrem, který udává „šipka" v tomto bodě stojící, to znamená po přímce o rovnici $y = y_0 + f(x_0, y_0)(x - x_0)$, dokud nedojdeme do bodu s x-ovou souřadnicí x_1. Ypsilonová souřadnice tohoto bodu je $y_1 = y_0 + f(x_0, y_0)(x_1 - x_0) = y_0 + hf(x_0, y_0)$. Z bodu $[x_1, y_1]$ pokračujeme ve směru daném směrovým polem v tomto bodě, tj. po přímce $y = y_1 + f(x_1, y_1)(x - x_1)$, dokud nedojdeme do bodu s x-ovou souřadnicí x_2 atd. Situace je znázorněna na obrázku 8.3. Graf přesného řešení vyhovujícího počáteční podmínce $y(x_0) = y_0$, na obrázku nakreslený šedě, aproximujeme lomenou čarou procházející body $[x_0, y_0],\ [x_1, y_1],\ [x_2, y_2] \ldots$

8.1.2 Typy a vlastnosti metod pro řešení počátečních úloh, lokální a globální chyba

Jak jsme viděli na příkladu Eulerovy metody, při numerickém řešení počáteční úlohy 8.1 můžeme vypočítat přibližnou hodnotu řešení v dalším uzlovém bodu pomocí hodnoty řešení v uzlovém bodu předchozím. U některých jiných metod sice postupujeme



Obrázek 8.2: Směrové pole



Obrázek 8.3: Přibližné řešení diferenciální rovnice Eulerovou metodou

poněkud důmyslněji než u metody Eulerovy, ale stále využíváme pouze informace z jediného předchozího kroku. Takovýmto metodám říkáme **metody jednokrokové**.
U jiných metod využíváme informace z několika předchozích kroků. Těmto metodám říkáme **metody vícekrokové**.

Je vcelku zřejmé, že nakolik se přiblížíme k přesnému řešení, závisí na délce kroku h, který použijeme. Základní vlastnost, kterou od použitelné numerické metody požadujeme, je,

aby numerické řešení získané touto metodou pro $h \to 0$ konvergovalo k přesnému řešení dané úlohy.

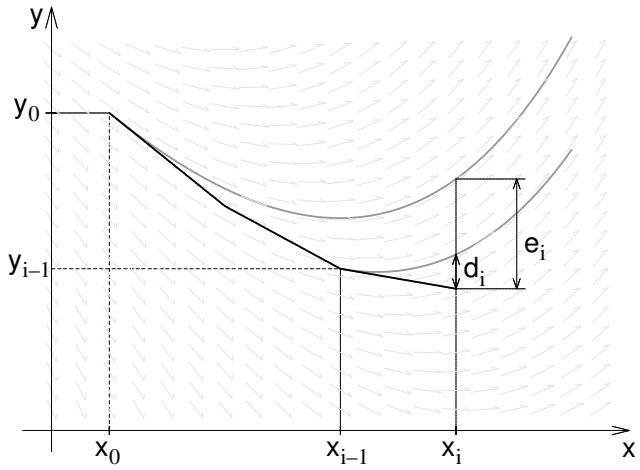
Řekneme, že metoda je **konvergentní**, jestliže pro libovolnou počáteční úlohu 8.1 platí pro každé $x \in \langle a, b \rangle$

$$\lim_{\substack{h\to 0 \\ n\to\infty}} y_n = y(x) \quad \text{, kde } x = x_0 + nh \ .$$

U každé metody je důležitá otázka, jak se přibližné řešení získané touto metodou liší od řešení přesného, neboli jak vypadá **globální diskretizační chyba**

$$e_i = y(x_i) - y_i$$

Pro získání představy o globální diskretizační chybě bývá mnohdy velmi užitečné znát tzv. **lokální diskretizační chybu** dané metody. Je to chyba, které se dopustíme v jednom kroku dané metody za předpokladu, že všechny hodnoty, které jsme při výpočtu použili, byly přesné. Lokální diskretizační chybu v i-tém uzlu budeme značit d_i. Na obrázku 8.4 vidíme globální diskretizační chybu e_i a lokální diskretizační chybu d_i u přibližného řešení získaného Eulerovou metodou. Lokální chyba Eulerovy (i jakékoli jiné jednokrokové) metody v uzlu x_i je rozdíl přibližného řešení a řešení, které splňuje počáteční podmínku $y(x_{i-1}) = y_{i-1}$.



Obrázek 8.4: Globální a lokální chyba

Při numerickém řešení diferenciální rovnice se dopouštíme lokální diskretizační chyby v každém kroku. Globální diskretizační chyba je tedy výsledkem nakupení lokálních chyb, přičemž je třeba brát v úvahu, že každý krok vychází z hodnot, které už jsou zatíženy chybou z předešlého průběhu. Je žádoucí, aby u dané metody nedocházelo ke katastrofální

akumulaci lokálních diskretizačních chyb. O metodách, u kterých se daří udržet chybu malou vzhledem k přesnému řešení, se obvykle mluví jako o **stabilních metodách**.

Pro popis rychlosti konvergence metody používáme pojem **řád metody**. Zhruba řečeno je řád metody přirozené číslo p takové, že pro malá h je lokální diskretizační chyba d_i řádově velikosti h^{p+1}. Přesnější definici lze nalézt např. ve skriptech [**?**]. U jednokrokových metod p-tého řádu lze dokázat, že globální diskretizační chyba je řádově velikosti h^p.

Eulerova metoda je řádu prvního. V dalších dvou kapitolách ukážeme několik jednokrokových metod vyšších řádů.

8.1.3 Modifikace Eulerovy metody

Jak již název napovídá, budeme postupovat podobně jako u Eulerovy metody.
Nejprve vypočteme pomocné hodnoty k_1 a k_2 a pomocí nich pak přibližnou hodnotu řešení v dalším uzlovém bodě.
U první modifikované Eulerovy metody počítáme podle vzorců

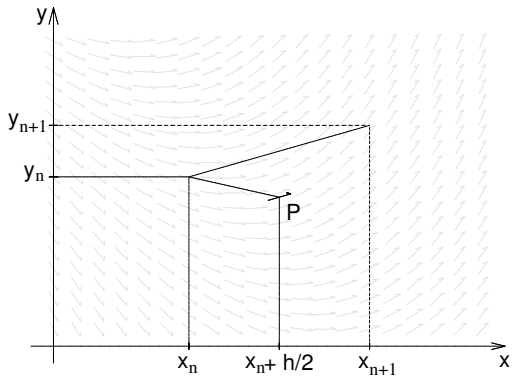
$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,hk_1) \\ y_{n+1} &= y_n + hk_2, \end{aligned} \tag{8.3}$$

u druhé modifikace podle vzorců

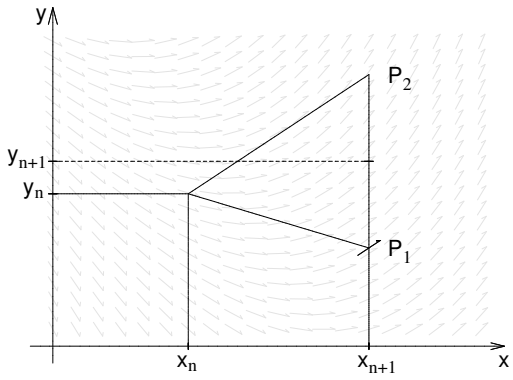
$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + h\,, y_n + hk_1) \\ y_{n+1} &= y_n + \frac{1}{2}\,h(k_1 + k_2). \end{aligned} \tag{8.4}$$

Obě modifikované Eulerovy metody jsou druhého řádu.

Geometricky lze tyto metody interpretovat podobně jako Eulerovu metodu.
Na obrázcích 8.5, resp. 8.6 vidíme jeden krok první, resp. druhé modifikované Eulerovy metody.
U první modifikace nejprve najdeme pomocný bod P, a to tak, že z bodu $[x_n, y_n]$ vyjdeme po přímce se směrnicí $f(x_n, y_n)$, tj. stejně jako u Eulerovy metody, ale dojdeme jen do bodu s x-ovou souřadnicí $x_n + \frac{h}{2}$. Přibližnou hodnotu řešení v bodě x_{n+1} pak získáme tak, že z bodu $[x_n, y_n]$ jdeme po přímce se směrnicí určenou směrovým polem v bodě P, dokud nedojdeme do bodu s x-ovou souřadnicí x_{n+1}.
U druhé modifikace zkonstruujeme dva pomocné body P_1 a P_2. Bod P_1 dostaneme jedním krokem obyčejné Eulerovy metody. Bod P_2 pak získáme tak, že z bodu $[x_n, y_n]$ jdeme po přímce se směrnicí danou směrovým polem v bodě P_1 do bodu s x-ovou souřadnicí x_{n+1}. Nový bod $[x_{n+1}, y_{n+1}]$ pak leží ve středu úsečky P_1P_2.



Obrázek 8.5: První modifikace Eulerovy metody



Obrázek 8.6: Druhá modifikace Eulerovy metody

8.1.4 Rungovy-Kuttovy metody

Rungovy-Kuttovy metody jsou jedna z nejdůležitějších skupin jednokrokových metod. Se dvěma jednoduchými příklady metod Runge-Kutta, první a druhou modifikovanou Eulerovou metodou, jsme se již setkali v předchozí kapitole.

Obecný tvar Rungovy-Kuttovy metody je

$$y_{n+1} = y_n + h(w_1 k_1 + \cdots + w_s k_s), \tag{8.5}$$

kde

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_i &= f(x_n + \alpha_i h\,, y_n + h\sum_{j=1}^{i-1} \beta_{ij} k_j)\,, \quad i = 2, \ldots, s \end{aligned} \tag{8.6}$$

a w_i, α_i a β_{ij} jsou konstanty volené tak, aby metoda měla maximální řád.
(Více o způsobu volby těchto konstant lze nalézt např. v [**?**] nebo [4].)

U první modifikované Eulerovy metody bylo $w_1 = 0, w_2 = 1, \alpha_2 = \frac{1}{2}$ a $\beta_{21} = \frac{1}{2}$, u druhé modifikace $w_1 = w_2 = \frac{1}{2}\,, \alpha_2 = 1$ a $\beta_{21} = 1$.

V praxi se nejvíce používá následující metoda Runge-Kutta 4. řádu. Často, mluví-li se o Rungově-Kuttově metodě, myslí se tím právě tato konkrétní metoda.

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6}\,h(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,hk_1) \\ k_3 &= f(x_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,hk_2) \\ k_4 &= f(x_n + h, y_n + hk_3) \end{aligned} \tag{8.7}$$

Příklad 8.2 *Rungovou-Kuttovou metodou řešte počáteční úlohu*

$$y' = x^2 - y \quad , \quad y(0) = 1$$

s krokem $h = 0,1$ *na intervalu* $\langle 0\,;\, 0,5\rangle$.

Řešení: V každém kroku musíme vypočítat čísla k_1, k_2, k_3 a k_4 a pomocí nich pak přibližnou hodnotu řešení v dalším uzlovém bodě podle vzorce 8.7. Všechny potřebné hodnoty budeme opět zapisovat do tabulky. Ve sloupcích označených x a y jsou souřadnice bodů, v nichž vyčíslujeme funkci $f(x, y) = x^2 - y$ při výpočtu k_i. Pro srovnání vypíšeme i hodnoty přesného řešení $y = -e^{-x} + x^2 - 2x + 2$. Tentokrát jsou čísla zaokrouhlována na 7 desetinných míst.

n	x_n	y_n	$y(x_n)$	x	y	
0	0	1	1	0	1	$k_1 = -1$
				0,05	0,95	$k_2 = -0,9475$
				0,05	0,952625	$k_3 = -0,950125$
				0,1	0,9049875	$k_4 = -0,8949875$
1	0,1	0,9051627	0,9051626	0,1	0,9051627	$k_1 = -0,8951627$
				0,15	0,8604046	$k_2 = -0,8379046$
				0,15	0,8632675	$k_3 = -0,8407675$
				0,2	0,8210860	$k_4 = -0,7810860$
2	0,2	0,8212695	0,8212693	0,2	0,8212695	$k_1 = -0,7812695$
				0,25	0,7822060	$k_2 = -0,7197060$
				0,25	0,7852842	$k_3 = -0,7227842$
				0,3	0,7489911	$k_4 = -0,6589911$
3	0,3	0,7491822	0,7491818	0,3	0,7491822	$k_1 = -0,6591822$
				0,35	0,7162230	$k_2 = -0,5937230$
				0,35	0,7194960	$k_3 = -0,5969960$
				0,4	0,6894826	$k_4 = -0,5294826$
4	0,4	0,6896804	0,6896800	0,4	0,6896804	$k_1 = -0,5296804$
				0,45	0,6631964	$k_2 = -0,4606964$
				0,45	0,6666456	$k_3 = -0,4641456$
				0,5	0,6432659	$k_4 = -0,3932659$
5	0,5	0,6434699	0,6434693			

Výsledky můžeme porovnat s hodnotami přibližného řešení vypočtenými Eulerovou metodou v příkladu 8.1 (kde se řešila tatáž počáteční úloha). Vidíme, že řešení získané metodou Runge-Kutta 4. řádu je podstatně přesnější.

8.1.5 Odhad chyby. Metoda poloviční kroku

Teoretické odhady chyb zde uvedených jednokrokových metod lze nalézt v literatuře. Jejich použití v praxi je však problematické. Proto se používá spíše tzv. **metoda poloviční kroku**, kterou nyní velmi zjednodušeně popíšeme.

Mějme numerickou metodu pro řešení počátečních úloh, která je řádu p. Pro účely této kapitoly změníme poněkud dosud užívané značení. Přesné řešení úlohy budeme stále značit

$y(x)$. Jako $y(x,h)$ označíme přibližnou hodnotu řešení v bodě x, kterou jsme dostali použitím naší numerické metody s krokem h.
Protože metoda je p-tého řádu, pro chybu platí

$$y(x) - y(x,h) \doteq c \cdot h^p,$$

kde c závisí na x, ale nikoli na h, neboli

$$y(x) \doteq y(x,h) + c \cdot h^p. \tag{8.8}$$

Do stejného bodu x můžeme dojít i pomocí poloviční kroku. V tomto případě platí

$$y(x) \doteq y\left(x, \frac{h}{2}\right) + c\left(\frac{h}{2}\right)^p. \tag{8.9}$$

Rovnici 8.9 můžeme vynásobit 2^p a odečíst od rovnice 8.8. Tím se vyloučí člen obsahující neznámou konstantu c a po mírné úpravě dostaneme nové přibližné vyjádření $y(x)$,

$$y(x) \doteq \frac{2^p\, y(x, \frac{h}{2}) - y(x,h)}{2^p - 1}, \tag{8.10}$$

které je přesnější než obě přibližné hodnoty $y(x,h)$ a $y(x, \frac{h}{2})$. Z posledního vztahu můžeme vyjádřit chybu v bodě x pro krok $\frac{h}{2}$

$$y(x) - y\left(x, \frac{h}{2}\right) \doteq \frac{1}{2^p - 1}\left(y\left(x, \frac{h}{2}\right) - y(x,h)\right), \tag{8.11}$$

resp. pro krok h

$$y(x) - y(x,h) \doteq \frac{2^p}{2^p - 1}\left(y\left(x, \frac{h}{2}\right) - y(x,h)\right). \tag{8.12}$$

Odhad chyby 8.12 lze použít pro **řízení délky kroku** h. Vypočteme vždy přibližnou hodnotu řešení v bodě x_i jedním krokem metody s použitím kroku h a dvěma kroky metody s použitím kroku $\frac{h}{2}$. Pak můžeme pomocí těchto dvou hodnot odhadnout chybu. Je-li příliš velká, vrátíme se do předchozího uzlového bodu a pokračujeme s polovičním krokem, je-li chyba vzhledem k našim požadavkům na přesnost příliš malá, pokračujeme dále s větším krokem, např. dvojnásobným. Jako výslednou aproximaci pak můžeme vzít kombinaci obou hodnot vypočtenou podle vzorce 8.10. Tato metoda je dosti pracná, ale účinná a v praxi často používaná.

Příklad 8.3 *Metodou Runge-Kutta čtvrtého řádu najděte hodnotu řešení počáteční úlohy*

$$y' = y\mathrm{e}^x\,, \quad y(0) = 1$$

bodě $x = 0{,}2$ *s přesností* 10^{-7}.

Řešení: Použijeme metodu polovičního kroku. Začneme s krokem $h = 0{,}2$, provedeme jeden krok metodou Runge-Kutta. Vyjde $y(0{,}2; 0{,}2) \doteq 1{,}24782070$.
Nyní dojdeme do bodu 0,2 pomocí dvou kroků metody R-K s krokem $h = 0{,}1$. Vyjde $y(0{,}2; 0{,}1) \doteq 1{,}24782556$.
Podíváme se, je-li chyba dostatečně malá:
$\frac{1}{2^4-1}(y(0{,}2; 0{,}1) - y(0{,}2; 0{,}2)) \doteq 3 \cdot 10^{-7} > 10^{-7}$
S výsledkem se tedy nemůžeme spokojit. Musíme začít znovu od začátku a použít menší krok, $h = 0{,}1$. Vypočteme hodnotu řešení v bodě 0,1 nejprve pomocí jednoho kroku metody s $h = 0{,}1$ a pak pomocí dvou kroků metody s $h = 0{,}05$:
$y(0{,}1; 0{,}1) \doteq 1{,}11090035, \quad y(0{,}1; 0{,}05) \doteq 1{,}11090046$
Odhadneme chybu:
$\frac{1}{2^4-1}(y(0{,}1; 0{,}05) - y(0{,}1; 0{,}1)) \doteq 7 \cdot 10^{-9} < 10^{-7}$.
Zatím je všechno v pořádku, můžeme pokračovat se stejným krokem. Jako přibližnou hodnotu řešení v bodě 0,1 vezmeme kombinaci
$y = \frac{2^4 y(0{,}1;0{,}05) - y(0{,}1;0{,}1)}{2^4-1} \doteq 1.11090047$. (Mohli bychom ale pracovat i s $y(0{,}1; 0{,}05)$.)
Uděláme další krok - tím se dostaneme do bodu 0,2. Pak se do téhož bodu dostaneme dvěma kroky s $h = 0{,}05$: $y(0{,}2; 0{,}1) \doteq 1{,}24782569, \quad y(0{,}2; 0{,}05) \doteq 1{,}24782589$,
$\frac{1}{2^4-1}(y(0{,}2; 0{,}05) - y(0{,}2; 0{,}1)) \doteq 10^{-9} < 10^{-7}$.
Hodnota řešení zadané počáteční úlohy v bodě $x = 0{,}2$ s přesností 10^{-7} je tedy
$y(0{,}2; 0{,}05) \doteq 1{,}2478259$ (případně bychom mohli použít i kombinaci $y(0{,}2; 0{,}05)$ a $y(0{,}2; 0{,}1)$, ta je ještě přesnější).

8.1.6 Vícekrokové metody

U vícekrokových metod počítáme přibližné řešení v dalším uzlovém bodě sítě pomocí několika předchozích uzlů. Protože přitom používáme nejen hodnoty přibližného řešení, ale také hodnoty pravé strany $f(x, y)$ v těchto bodech, budeme kvůli snadnějšímu zápisu používat označení $f_j = f(x_j, y_j)$.
Obecně vypadá **lineární k-kroková metoda** takto:

$$y_{n+1} = a_1\, y_n + a_2\, y_{n-1} + \cdots + a_k\, y_{n-k+1} + h\,(b_0\, f_{n+1} + b_1\, f_n + \cdots + b_k\, f_{n-k+1}) \ , \quad (8.13)$$

kde k je přirozené číslo a alespoň jedna z konstant a_k, b_k je různá od nuly.

Zřejmou nevýhodou k-krokové metody je, že řešení v prvních k uzlových bodech $x_0, \ldots, x_{k-1}$ musíme získat nějakým jiným způsobem. K tomuto účelu se zpravidla používá jednokroková metoda stejného řádu přesnosti, jaký má dále použitá vícekroková metoda.

Je-li $b_0 = 0$, metoda 8.13 se nazývá **explicitní**. V tomto případě můžeme hodnotu v novém uzlovém bodě přímo vypočítat dosazením do vzorce 8.13.

Je-li $b_0 \neq 0$, metoda 8.13 se nazývá **implicitní**. Pak se na pravé straně rovnice 8.13 kromě známých hodnot vyskytuje také $f_{n+1} = f(x_{n+1}, y_{n+1})$, takže y_{n+1} nemůžeme vypočítat přímo, ale v každém kroku musíme řešit rovnici

$$y_{n+1} = h b_0 f(x_{n+1}, y_{n+1}) + g$$

s neznámou y_{n+1}, kde $g = \sum_{j=1}^{k} a_j\, y_{n-j+1} + h \sum_{j=1}^{k} b_j\, f_{n-j+1}$ je známé číslo (v každém kroku jiné).

V případě některých pravých stran f tuto rovnici vyřešíme přesně, obecně je však potřeba tuto rovnici řešit numericky, většinou metodou prosté iterace.
Tato nevýhoda je však vyvážena příznivými vlastnostmi implicitních metod. Tyto metody jsou při daném k přesnější a jsou také stabilnější než explicitní metody.

Příklad 8.4 *Explicitní čtyřkrokovou metodou čtvrtého řádu*

$$y_{n+1} = y_{n-3} + \frac{4}{3}\,h\,(2f_{n-2} - f_{n-1} + 2f_n) \tag{8.14}$$

řešte počáteční úlohu

$$y' = x^2 - y \quad , \quad y(0) = 1$$

s krokem $h = 0,1$ *na intervalu* $\langle 0\,;\,0,7\rangle$.

Řešení: Nejprve musíme nějakým způsobem najít řešení v bodech $x_1 = 0{,}1$, $x_2 = 0{,}2$, $x_3 = 0{,}3$. Použijeme výsledky příkladu 8.2, kde jsme řešili tutéž počáteční úlohu metodou Rungeho-Kutty. Potřebné hodnoty zde znovu vypíšeme, včetně hodnot pravé strany $f(x, y) = x^2 - y$.

$y_1 = 0,9051627 \qquad y_2 = 0,8212695 \qquad y_3 = 0,7491822$
$f_1 = -0,8951627 \quad f_2 = -0,7812695 \quad f_3 = -0,6591822$

V dalších uzlových bodech už budeme postupovat podle vzorce 8.14, tzn.

$$y_4 = y_0 + \frac{4}{3}\,h\,(2f_1 - f_2 + 2f_3)\ , \quad y_5 = y_1 + \frac{4}{3}\,h\,(2f_2 - f_3 + 2f_4) \quad \text{atd.}$$

Vypočtené hodnoty zapíšeme do tabulky. Pro srovnání uvádíme i hodnoty přesného řešení.

n	x_n	y_n	f_n	$y(x_n)$
4	0,4	0,6896773	-0,5296773	0,6896800
5	0,5	0,6434678	-0,3934678	0,6434693
6	0,6	0,6111865	-0,2511865	0,6111884
7	0,7	0,5934142	-0,1034142	0,5934147

Příklad 8.5 *Implicitní tříkrokovou metodou čtvrtého řádu*

$$y_{n+1} = \frac{1}{8}\,(9y_n - y_{n-2}) + \frac{3}{8}\,h(f_{n+1} + 2f_n - f_{n-1}) \tag{8.15}$$

řešte počáteční úlohu

$$y' = x^2 - y \quad , \quad y(0) = 1$$

s krokem $h = 0,1$ *na intervalu* $\langle 0\,;\,0,4\rangle$.

Řešení: Jako výchozí hodnoty y_1, y_2 opět použijeme výsledky získané metodou Runge-Kutta v příkladu 8.2. Řešení v bodě $x_3 = 0,3$ budeme již počítat podle vzorce 8.15.
y_3 získáme jako řešení rovnice $y_3 = \frac{1}{8}\,(9y_2 - y_0) + \frac{3}{8}\,0,1(f(x_3, y_3) + 2f_2 - f_1)$, tj.
$y_3 = \frac{1}{8}\,(9y_2 - y_0) + \frac{3}{8}\,0,1(0,3^2 - y_3 + 2f_2 - f_1)$.
Vyjde $y_3 = 0,7491822$. K dalším výpočtům potřebujeme ještě $f_3 = -0,6591822$.
y_4 získáme jako řešení rovnice $y_4 = \frac{1}{8}\,(9y_3 - y_1) + \frac{3}{8}\,0,1(0,4^2 - y_4 + 2f_3 - f_2)$.
Vyjde $y_4 = 0,6896806$.
V tomto příkladu bylo řešení rovnic s neznámou y_{n+1} velmi jednoduché. Většinou je však potřeba složitější postup, který popíšeme v kapitole 8.1.8.

8.1.7 Vícekrokové metody založené na numerické integraci

Nyní ukážeme, jak odvodit některé konkrétní vícekrokové metody.

Řešenou rovnici $y'(x) = f\left(x, y(x)\right)$ můžeme zintegrovat na intervalu $\langle x_{n+1-s}, x_{n+1}\rangle$. Tím dostaneme

$$y(x_{n+1}) - y(x_{n+1-s}) = \int_{x_{n+1-s}}^{x_{n+1}} f\left(x, y(x)\right) \mathrm{d}x \tag{8.16}$$

Funkci f nahradíme interpolačním polynomem a ten zintegrujeme. Podle toho, jak zvolíme s a uzly interpolace, dostáváme různé metody.

Metoda použitá v příkladu 8.4 byla získána integrací přes interval $\langle x_{n-3}, x_{n+1}\rangle$ a použitím otevřeného Newton-Cotesova vzorce s uzly x_{n-2}, x_{n-1} a x_n.

Častější než použití Newton-Cotesových vzorců je však jiný postup: Funkci f nahradíme interpolačním polynomem s uzly $x_{n+1-k}, \ldots, x_n$, resp. s uzly $x_{n+1-k}, \ldots, x_{n+1}$, a rovnici zintegrujeme přes interval $\langle x_n, x_{n+1}\rangle$ (tzn. s v 8.16 je rovno 1). Tím dostaneme explicitní, resp. implicitní k-krokovou metodu.

Explicitní lineární k-krokové metody odvozené výše popsaným postupem se nazývají **Adams-Bashforthovy**.
Nejjednodušším případem Adams-Bashforthovy metody, kdy $k = 1$, je metoda Eulerova. V tomto případě funkci f nahrazujeme konstantou f_n. Integrací přes interval $\langle x_n, x_{n+1}\rangle$ dostaneme známý vzorec $y_{n+1} = y_n + hf_n$.
Zvolíme-li $k = 2$, budeme místo funkce f integrovat lineární polynom procházející body $[x_{n-1}, f_{n-1}], [x_n, f_n]$. Čtenář si může ověřit, že vyjde $y_{n+1} = y_n + h\left(\frac{3}{2}f_n - \frac{1}{2}f_{n-1}\right)$.
Podobně pro další k dostaneme vždy integrál z interpolačního polynomu jako lineární kombinaci funkčních hodnot $f_i, i = n, n-1, \ldots, n+1-k$. Obecný tvar Adams-Bashforthových metod je proto

$$y_{n+1} = y_n + h(b_1 f_n + b_2 f_{n-1} + \cdots + b_k f_{n+1-k}) \tag{8.17}$$

Přehled koeficientů b_i pro $k = 1, 2, 3, 4$ je v následující tabulce spolu s řádem přesnosti p každé metody.

k	b_1	b_2	b_3	b_4	p
1	1				1
2	3/2	-1/2			2
3	23/12	-16/12	5/12		3
4	55/24	-59/24	37/24	-9/24	4

Pokud za uzel interpolace vezmeme i x_{n+1}, dostaneme **Adams-Moultonovy metody**. Nejjednodušší z nich je tzv. implicitní Eulerova metoda : $y_{n+1} = y_n + hf_{n+1}$.
Obecný tvar Adams-Moultonových metod je

$$y_{n+1} = y_n + h\left(b_0 f_{n+1} + b_1 f_n + \cdots + b_k f_{n+1-k}\right). \tag{8.18}$$

Přehled koeficientů b_i pro $k = 0, 1, 2, 3$ je v následující tabulce, opět i s řádem přesnosti p. Všimněme si, že zde je řád p vyšší než k (na rozdíl od Adams-Bashforthových metod, kde byl stejný jako k).

k	b_0	b_1	b_2	b_3	p
0	1				1
1	1/2	1/2			2
2	5/12	8/12	-1/12		3
3	9/24	19/24	-5/24	1/24	4

Poznámka. Existují i metody založené na numerickém derivování. V tomto případě nahrazujeme derivaci neznámé funkce $y(x)$ na levé straně řešené diferenciální rovnice derivací interpolačního polynomu.

8.1.8 Metody prediktor-korektor

Jak již bylo řečeno, při použití implicitních vícekrokových metod je potřeba v každém kroku vypočítat y_{n+1} jako řešení rovnice

$$y_{n+1} = hb_0 f(x_{n+1}, y_{n+1}) + g, \tag{8.19}$$

kde $g = \sum_{j=1}^{k} a_j\, y_{n-j+1} + h \sum_{j=1}^{k} b_j\, f_{n-j+1}$.
Všimněme si, že rovnice 8.19 je zapsána ve tvaru vhodném pro použití metody prosté iterace, popsané v kapitole 4.1.5. K hledané hodnotě se můžeme postupně přibližovat iteračním procesem

$$y_{n+1}^{(r+1)} = hb_0 f(x_{n+1}, y_{n+1}^{(r)}) + g. \tag{8.20}$$

Dá se dokázat, že jsou-li splněny předpoklady věty 8.1, rovnice 8.19 má jediné řešení a zvolíme-li h dostatečně malé, iterační metoda konverguje.
Zbývá otázka, jak získat dobrou počáteční aproximaci $y_{n+1}^{(0)}$. K tomu se nabízí použití explicitní vícekrokové metody.

Princip metod prediktor-korektor je tedy tento:
V každém kroku nejprve vypočteme počáteční aproximaci $y_{n+1}^{(0)}$ pomocí explicitní vícekrokové metody - **prediktoru** (predikce = předpověď).
Tuto hodnotu zpřesníme použitím implicitní vícekrokové metody - **korektoru** (korekce = oprava), a to dosazením $y_{n+1}^{(0)}$ do 8.20 (s tím, že $r = 0$). Tím dostaneme $y_{n+1}^{(1)}$.
Někdy se korektor použije pouze jednou, někdy se iterace 8.20 opakují tak dlouho, dokud se nedosáhne požadované přesnosti, tj. dokud pro nějaké r neplatí $|\, y_{n+1}^{(r+1)} - y_{n+1}^{(r)}| < \varepsilon$.

Jako dvojici prediktor-korektor volíme vždy explicitní a implicitní metodu téhož řádu.

Jedna z možností je použití metody z příkladu 8.4 jako prediktoru a k tomu metody z příkladu 8.5 jako korektoru, ale používá se i řada jiných metod, viz následující příklad.

Příklad 8.6 *Metodou prediktor-korektor, konkrétně*
prediktor: $y_{n+1}^{(0)} = y_n + \frac{1}{24}\, h\, (55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3})$

korektor: $y_{n+1}^{(r+1)} = y_n + \frac{1}{24}\,h\left(9f(x_{n+1}, y_{n+1}^{(r)}) + 19f_n - 5f_{n-1} + f_{n-2}\right)$,
(Adams-Bashforthova a Adams-Moultonova metoda čtvrtého řádu)
řešte počáteční úlohu $y' = \dfrac{2-y}{x+1}$, $y(1) = 0$ *s krokem* $h = 0{,}1$ *na intervalu* $\langle 0\,;0{,}5\rangle$.
Korektor použijte vždy jednou.

Řešení: Protože při použití prediktoru musíme vždy znát řešení ve čtyřech předchozích uzlových bodech, musíme nejprve vypočítat řešení v bodech $x_1 = 1{,}1, x_2 = 1{,}2, x_3 = 1{,}3$ (hodnotu v $x_0 = 1$ známe z počáteční podmínky). Provedeme to pomocí metody Runge-Kutta čtvrtého řádu. Vyjde:

$$\begin{array}{llll} y_0 = 0 & y_1 = 0,095238 & y_2 = 0,181818 & y_3 = 0,260870 \\ f_0 = 1 & f_1 = 0,907029 & f_2 = 0,826446 & f_3 = 0,756144. \end{array}$$

Dále budeme pokračovat metodou prediktor korektor.

V uzlovém bodě $x_4 = 1,4$:
$y_4^{(0)} = y_3 + \frac{1}{24}\cdot 0{,}1\cdot(55f_3 - 59f_2 + 37f_1 - 9f_0) = 0,333318$
$y_4^{(1)} = y_3 + \frac{1}{24}\cdot 0{,}1\cdot(9f(1,4\,;\,0,333318) + 19f_3 - 5f_2 + f_1) = 0,333334.$
Tedy $y_4 = 0,333334$, hodnota pravé strany f je $f_4 = 0,694444$. Pro srovnání, přesná hodnota řešení je $y(1,4) = 1/3 \doteq 0,333333$.

V uzlovém bodě $x_5 = 1{,}5$:
$y_5^{(0)} = y_4 + \frac{1}{24}\cdot 0{,}1\cdot(55f_4 - 59f_3 + 37f_2 - 9f_1) = 0,399989$
$y_5^{(1)} = y_4 + \frac{1}{24}\cdot 0{,}1\cdot(9f(1,5\,;\,0,399989) + 19f_4 - 5f_3 + f_2) = 0,400002.$
Tedy $y_5 = 0,400002$. Přesná hodnota řešení je $y(1,5) = 0,4$.

Poznámka. Někdy se mezi prediktorem a korektorem používá tzv. modifikátor, jímž hodnotu získanou prediktorem před použitím korektoru ještě zpřesníme. Více o tom např. v [4] nebo [**?**].

8.1.9 Řešení soustav diferenciálních rovnic

Řešení soustavy obyčejných diferenciálních rovnic prvního řádu s počátečními podmínkami

$$\begin{array}{rclrcl} y_1' &=& f_1(x, y_1, y_2, \ldots, y_n) & y_1(x_0) &=& \eta_1 \\ y_2' &=& f_2(x, y_1, y_2, \ldots, y_n) & y_2(x_0) &=& \eta_2 \\ &\vdots& & &\vdots& \\ y_n' &=& f_n(x, y_1, y_2, \ldots, y_n) & y_n(x_0) &=& \eta_n \end{array} \tag{8.21}$$

se hledá velmi podobně jako řešení jediné diferenciální rovnice s počáteční podmínkou. Soustavu 8.21 můžeme přepsat vektorově jako

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}), \quad \mathbf{y}(x_0) = \eta, \tag{8.22}$$

kde $\mathbf{y} = (y_1, \ldots, y_n)^T$, $\mathbf{f} = (f_1, \ldots, f_n)^T$ a $\eta = (\eta_1, \ldots, \eta_n)^T$.

Pro její numerické řešení můžeme použít kteroukoli z dříve popsaných metod, jen je potřeba pracovat s vektory.

Eulerova metoda pro soustavu je tvaru

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\,\mathbf{f}(x_n, \mathbf{y}_n), \tag{8.23}$$

Rungova-Kuttova metoda 4. řádu pro soustavu vypadá následovně:

$$\begin{aligned} \mathbf{y}_{n+1} &= \mathbf{y}_n + \frac{1}{6}\,h(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \\ \mathbf{k}_1 &= \mathbf{f}(x_n, \mathbf{y}_n) \\ \mathbf{k}_2 &= \mathbf{f}(x_n + \frac{1}{2}\,h, \mathbf{y}_n + \frac{1}{2}\,h\,\mathbf{k}_1) \\ \mathbf{k}_3 &= \mathbf{f}(x_n + \frac{1}{2}\,h, \mathbf{y}_n + \frac{1}{2}\,h\,\mathbf{k}_2) \\ \mathbf{k}_4 &= \mathbf{f}(x_n + h, \mathbf{y}_n + h\,\mathbf{k}_3) \end{aligned} \tag{8.24}$$

Řešíme-li soustavu dvou rovnic, je jednodušší označit neznámé funkce jako y a z a funkce na pravé straně jako f a g, abychom se vyhnuli nepříjemné práci s mnoha indexy. Řešená soustava pak je

$$\begin{array}{rclcrcl} y' &=& f(x,y,z) & & y(x_0) &=& y_0 \\ z' &=& g(x,y,z) & & z(x_0) &=& z_0. \end{array} \tag{8.25}$$

Eulerovu metodu pak můžeme zapsat jako

$$\begin{aligned} y_{n+1} &= y_n + hf(x_n, y_n, z_n) \\ z_{n+1} &= z_n + hg(x_n, y_n, z_n), \end{aligned} \tag{8.26}$$

metodu Runge-Kutta 4. řádu jako

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6}\,h(k_1 + 2k_2 + 2k_3 + k_4) \\ z_{n+1} &= z_n + \frac{1}{6}\,h(l_1 + 2l_2 + 2l_3 + l_4), \end{aligned} \tag{8.27}$$

kde

$$\begin{array}{rclcrcl} k_1 &=& f(x_n, y_n, z_n) & & l_1 &=& g(x_n, y_n, z_n) \\ k_2 &=& f(x_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,hk_1, z_n + \frac{1}{2}\,hl_1) & & l_2 &=& g(x_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,hk_1, z_n + \frac{1}{2}\,hl_1) \\ k_3 &=& f(x_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,hk_2, z_n + \frac{1}{2}\,hl_2) & & l_3 &=& g(x_n + \frac{1}{2}\,h, y_n + \frac{1}{2}\,hk_2, z_n + \frac{1}{2}\,hl_2) \\ k_4 &=& f(x_n + h, y_n + hk_3, z_n + hl_3) & & l_4 &=& g(x_n + h, y_n + hk_3, z_n + hl_3) \end{array}$$

Příklad 8.7 *Soustavu diferenciálních rovnic s počátečními podmínkami*

$$\begin{array}{rclcl} y' &=& x - y - z & & y(0) = 1 \\ z' &=& y\,e^z & & z(0) = 0 \end{array}$$

řešte Eulerovou metodou s krokem $h = 0,05$. Proveďte 2 kroky.

Řešení: V tomto případě je $f(x,y,z) = x - y - z$, $g(x,y,z) = y\,e^z$, $y_0 = 1$ a $z_0 = 0$. Přibližné hodnoty řešení v uzlových bodech $x_1 = 0,05$ a $x_2 = 0,1$ vypočteme podle vzorců 8.26:

$$\begin{array}{ll} y_1 = 1 + 0,05\,(0 - 1 - 0) = 0,95 & z_1 = 0 + 0,05 \cdot 1 \cdot e^0 = 0,05 \\ y_2 = 0,95 + 0,05\,(0,05 - 0,95 - 0,05) = 0,9025 & z_2 = 0,05 + 0,05 \cdot 0,95 \cdot e^{0,05} \doteq 0,0999. \end{array}$$

8.1.10 Řešení diferenciálních rovnic vyššího řádu

Obyčejnou diferenciální rovnici n-tého řádu s počátečními podmínkami

$$y^{(n)} = f(x, y, y', \ldots, y^{(n-1)}), \quad y(x_0) = y_0, y'(x_0) = y'_0, \ldots, y^{(n-1)}(x_0) = y_0^{(n-1)} \tag{8.28}$$

můžeme **převést na soustavu diferenciálních rovnic prvního řádu**, a to následujícím způsobem:
Označíme $y_1 = y, y_2 = y', \ldots, y_n = y^{(n-1)}$. Potom zřejmě platí, že $y'_1 = y_2, y'_2 = y_3$ atd. Podle zadané diferenciální rovnice má platit $y^{(n)} = f(x, y, y', \ldots, y^{(n-1)})$, což při našem označení znamená $y'_n = f(x, y_1, y_2, \ldots, y_n)$. Tím jsme získali soustavu n diferenciálních rovnic prvního řádu

$$\begin{array}{rclcrcl} y'_1 &=& y_2 & & y_1(x_0) &=& y_0 \\ y'_2 &=& y_3 & & y_2(x_0) &=& y'_0 \\ & \vdots & & & & \vdots & \\ y'_n &=& f(x, y_1, y_2, \ldots, y_n) & & y_n(x_0) &=& y_0^{(n-1)}, \end{array} \tag{8.29}$$

kterou můžeme řešit kteroukoli z výše popsaných metod. Řešením původní rovnice n-tého řádu je pak první složka řešení soustavy 8.29.

Příklad 8.8 *Diferenciální rovnici druhého řádu $y'' = y \cdot y' - x^2$ s počátečními podmínkami $y(0) = 1, y'(0) = 1$ nejprve převeďte na soustavu dvou rovnic prvního řádu a tu pak řešte metodou Runge-Kutta 4. řádu. Proveďte dva kroky s krokem $h = 0,1$.*

Řešení: Označíme $z = y'$. Soustava rovnic prvního řádu je pak

$$\begin{array}{rcll} y' &=& z & y(0) = 1 \\ z' &=& y \cdot z - x^2 & z(0) = 1 \end{array}$$

Tuto soustavu budeme řešit metodou Runge-Kutta. Všechny potřebné hodnoty jsou zapsány v následující tabulce. Ve sloupcích označených x, y a z jsou souřadnice bodů, v nichž vyčíslujeme hodnoty funkcí $f(x, y, z) = z$ a $g(x, y, z) = y \cdot z - x^2$ při výpočtu k_i a l_i.

n	x_n	y_n	z_n	x	y	z		
0	0	1	1	0	1	1	$k_1 = 1$	$l_1 = 1$
				0,05	1,05	1,05	$k_2 = 1,05$	$l_2 = 1,1$
				0,05	1,10525	1,055	$k_3 = 1,055$	$l_3 = 1,107888$
				0,1	1,1055	1,110789	$k_4 = 1,110789$	$l_4 = 1,217977$
1	0,1	1,105346	1,110563	0,1	1,105346	1,110563	$k_1 = 1,110563$	$l_1 = 1,217556$
				0,15	1,160875	1,171440	$k_2 = 1,171440$	$l_2 = 1,337395$
				0,15	1,163918	1,177432	$k_3 = 1,177432$	$l_3 = 1,347935$
				0,2	1,223090	1,245356	$k_4 = 1,245356$	$l_4 = 1,483182$
2	0,2	1,222908	1,245086					

Přibližné hodnoty řešení původní rovnice druhého řádu v uzlových bodech $x_1 = 0,1$ a $x_2 = 0,2$ tedy jsou $y_1 \doteq 1,105346$ a $y_2 \doteq 1,222908$.

8.2 Okrajové úlohy

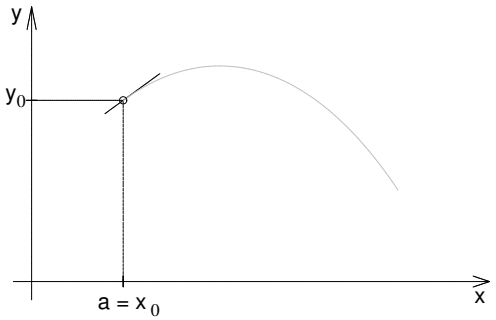
Dosud jsme se zabývali úlohami, u kterých jsme znali hodnotu řešení, případně hodnoty derivací řešení, v počátečním bodě intervalu, který nás zajímal. U okrajových úloh je situace jiná. Jak již název napovídá, budou zadány hodnoty řešení v krajních bodech zkoumaného intervalu.

V této kapitole budeme hledat řešení diferenciální rovnice druhého řádu

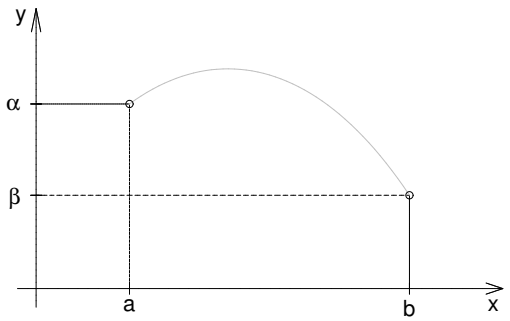
$$y'' = f(x, y, y') \tag{8.30}$$

na intervalu $\langle a, b\rangle$ s okrajovými podmínkami

$$y(a) = \alpha, \quad y(b) = \beta. \tag{8.31}$$



Obrázek 8.7: Počáteční úloha - zadáno je $y(x_0)$ a $y'(x_0)$ (tj. směrnice tečny).



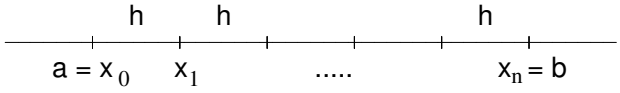
Obrázek 8.8: Okrajová úloha

Okrajové podmínky mohou být i jiného tvaru než 8.31. O tom, jak se postupuje při řešení takovýchto úloh, se zmíníme později.

Teorie existence a jednoznačnosti řešení okrajových úloh je mnohem komplikovanější než u úloh počátečních a zdaleka není tak univerzální. Obtížnější je i numerické řešení těchto úloh.
V dalším textu se seznámíme s metodou konečných diferencí a velmi stručně s metodou střelby a uvedeme podmínky zaručující existenci a jednoznačnost řešení pro některé konkrétní typy rovnic.

8.2.1 Metoda konečných diferencí

Tato metoda se též nazývá **metoda sítí**. Podobně jako u dříve probraných metod budeme hledat přibližné hodnoty řešení pouze v tzv. **uzlových bodech** x_i, $i = 0, 1, \ldots, n$, které získáme tak, že interval $\langle a, b\rangle$ rozdělíme na n stejných dílků délky $h = (b - a)/n$. Uzlové body pak jsou $x_i = a + ih$.



Budeme požadovat platnost rovnice 8.30 ve všech vnitřních uzlech x_i, $i = 1, \ldots, n-1$, tj.

$$y''(x_i) = f(x_i, y(x_i), y'(x_i)), \quad i = 1, \ldots, n-1.$$

Derivace vystupující v této rovnici nahradíme diferencemi (viz kapitola 6.1), např. takto:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f\left(x_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}\right), \quad i = 1, \ldots, n-1 \tag{8.32}$$

Přidáme-li k rovnicím 8.32 okrajové podmínky 8.31, dostaneme tzv. **soustavu diskretizačních rovnic** (obecně nelineárních, záleží na povaze funkce f) s neznámými $y_1, \ldots, y_{n-1}$. Tuto soustavu pak vyřešíme některou metod popsaných v kapitolách 3 a 4.2.
Přesnost výsledku závisí na přesnosti zvolených diferenčních formulí a na metodě užité k řešení vzniklé soustavy rovnic.

Metodu konečných diferencí nyní podrobněji předvedeme na okrajové úloze

$$-y'' + \sigma(x)y = f(x), \quad y(a) = \alpha, \ y(b) = \beta \tag{8.33}$$

Věta 8.2 *Jsou-li funkce $\sigma(x)$ a $f(x)$ spojité na intervalu $\langle a, b\rangle$ a $\sigma(x) \geq 0$ pro $x \in \langle a, b\rangle$, pak okrajová úloha 8.33 má jediné řešení pro jakékoli hodnoty α a β.*

Poznámka. Nejsou-li splněny předpoklady věty 8.2, úloha 8.33 řešení mít může a nemusí. Předvedeme to na jednoduchém příkladu rovnice $y'' + y = 0$ (neboli $\sigma(x) \equiv -1$). Obecné řešení této rovnice je $y = c_1 \sin x + c_2 \cos x$.
Pro okrajové podmínky $y(0) = 0, y(\frac{\pi}{2}) = 1$ má úloha řešení jediné, zatímco předepíšeme-li okrajové podmínky $y(0) = 0, y(\pi) = 0$, úloha bude mít nekonečně mnoho řešení tvaru $y = c_1 \sin x$, kde c_1 je libovolná konstanta, a naopak, předepíšeme-li okrajové podmínky $y(0) = 0, y(\pi) = 1$, úloha nebude mít řešení žádné.

Nyní odvodíme soustavu diskretizačních rovnic pro úlohu 8.33. Označíme $\sigma(x_i) = \sigma_i$, $f(x_i) = f_i$ a druhou derivaci neznámé funkce y nahradíme diferencí podle předpisu 6.6:

$$-\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + \sigma_i y_i = f_i, \quad i = 1, \ldots, n-1$$

Rovnici vynásobíme h^2 a sloučíme členy obsahující y_i. Dostaneme:

$$-y_{i-1} + (2 + h^2\sigma_i)y_i - y_{i+1} = h^2 f_i, \quad i = 1, \ldots, n-1. \tag{8.34}$$

Dosadíme-li za y_0 a y_n z okrajových podmínek α a β, dostaneme soustavu

$$\begin{array}{rcccccl}
(2 + h^2\sigma_1)y_1 & - & y_2 & & & & = h^2 f_1 + \alpha \\
-y_1 & + & (2 + h^2\sigma_2)y_2 & - & y_3 & & = h^2 f_2 \\
& & & \ddots & & & \vdots \\
& - & y_{n-2} & + & (2 + h^2\sigma_{n-1})y_{n-1} & & = h^2 f_{n-1} + \beta
\end{array} \tag{8.35}$$

Je vidět, že matice této soustavy je třídiagonální, symetrická a diagonálně dominantní. Dá se ukázat, že je také pozitivně definitní. Soustavu můžeme řešit např. Gaussovou eliminací přizpůsobenou pro třídiagonální soustavu.

Příklad 8.9 *Metodou konečných diferencí řešte okrajovou úlohu*

$$-y'' + (1 + x^2)y = x, \quad y(0) = 1,\ y(1) = 2$$

s krokem $h = 0,25$.

Řešení: Protože krok je $h = 0,25$, budeme hledat přibližné hodnoty řešení v uzlových bodech $x_1 = 0,25$, $x_2 = 0,5$, $x_3 = 0,75$. V krajních bodech intervalu $x_0 = 0$ a $x_4 = 1$ řešení známe z okrajových podmínek.
Vypočteme potřebné hodnoty σ_i a f_i:

i	0	1	2	3	4
x_i	0	0,25	0,5	0,75	1
$\sigma_i = 1 + x_i^2$	-	1,0625	1,25	1,5625	-
$f_i = x_i$	-	0,25	0,5	0,75	-

Soustava diskretizačních rovnic pak je:

$$\begin{aligned} 2,06640625y_1 &- y_2 &&= 0,015625 + 1 \\ -y1 &+ 2,078125y_2 - y_3 &&= 0,03125 \\ &- y_2 + 2,09765625y_3 &&= 0,046875 + 2 \end{aligned}$$

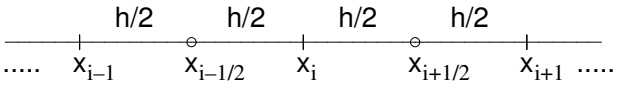
Řešení této soustavy je $y_1 \doteq 1,140$, $y_2 \doteq 1,341$, $y_3 \doteq 1,615$.
Pro srovnání, hodnoty přesného řešení jsou $y(x_1) \doteq 1,138$, $y(x_2) \doteq 1,337$, $y(x_3) \doteq 1,612$.
Kdybychom chtěli dosáhnout větší přesnosti, museli bychom interval rozdělit jemněji.

Nyní se budeme zabývat významným typem okrajových úloh, tzv. **rovnicí v samoadjungovaném tvaru**

$$-(p(x)y')' + q(x)y = f(x), \quad y(a) = \alpha,\ y(b) = \beta. \tag{8.36}$$

Věta 8.3 *Jsou-li funkce* $p(x), p'(x), q(x)$ *a* $f(x)$ *na intervalu* $\langle a, b\rangle$ *spojité a splňují-li na něm podmínky* $p(x) > 0, q(x) \geq 0$, *pak okrajová úloha 8.36 má jediné řešení pro jakékoli hodnoty* α, β.

Při řešení úlohy 8.36 metodou sítí budeme opět hledat řešení v uzlových bodech x_i, ale pro náhradu derivací diferencemi použijeme navíc ještě „poloviční uzly" $x_{i+1/2} = x_i + \frac{h}{2}$.



Podobně jako dříve budeme značit $q_i = q(x_i)$, $f_i = f(x_i)$ a $p_{i+1/2} = p(x_{i+1/2})$.
Vnější derivaci členu $(p(x)y')'$ v i-tém uzlu můžeme nahradit diferencí takto (v podstatě podle vzorce 6.4):

$$(p\,y')'(x_i) \doteq \frac{p_{i+1/2}\,y'(x_{i+1/2}) - p_{i-1/2}\,y'(x_{i-1/2})}{h}$$

Nyní nahradíme diferencemi hodnoty $y'(x_{i+1/2})$ a $y'(x_{i-1/2})$:

$$y'(x_{i+1/2}) \doteq \frac{y_{i+1} - y_i}{h}\ , \quad y'(x_{i-1/2}) \doteq \frac{y_i - y_{i-1}}{h}$$

Dosazením těchto vztahů do rovnice 8.36 dostaneme

$$-\frac{1}{h}\left(p_{i+1/2}\frac{y_{i+1} - y_i}{h} - p_{i-1/2}\frac{y_i - y_{i-1}}{h}\right) + q_i y_i = f_i\ .$$

Vynásobením rovnice h^2 a sloučením členů obsahujících y_i získáme soustavu diskretizačních rovnic pro neznámé $y_1, \ldots, y_{n-1}$

$$-p_{i-1/2}\,y_{i-1} + (p_{i-1/2} + p_{i+1/2} + q_i h^2)y_i - p_{i+1/2}\,y_{i+1} = h^2 f_i\ , \quad i = 1, \ldots, n-1 \quad (8.37)$$

V první a poslední rovnici přitom využijeme hodnoty známé z okrajových podmínek - $y_0 = \alpha$ a $y_n = \beta$.
Soustava v rozepsaném tvaru pak vypadá následovně:

$$\begin{array}{rcccccl}
(p_{1/2}+p_{3/2}+h^2q_1)\,y_1 & - & p_{3/2}\,y_2 & & & = & h^2 f_1 + p_{1/2}\,\alpha \\
-p_{3/2}\,y_1 & + & (p_{3/2}+p_{5/2}+h^2q_2)\,y_2 & - & p_{5/2}\,y_3 & = & h^2 f_2 \\
& & & \ddots & & \vdots & \\
& - & p_{n-3/2}\,y_{n-2} & + & (p_{n-3/2}+p_{n-1/2}+h^2q_{n-1})\,y_{n-1} & = & h^2 f_{n-1} + p_{n-1/2}\,\beta
\end{array} \quad (8.38)$$

Matice této soustavy je (stejně jako u rovnice 8.33) třídiagonální, symetrická, diagonálně dominantní a pozitivně definitní.

Příklad 8.10 *Metodou konečných diferencí řešte okrajovou úlohu*

$$-(x^2y')' + xy = 1, \quad y(1) = 1,\ y(2) = 0,5$$

s krokem $h = 0,2$.

Řešení: Podmínky existence a jednoznačnosti řešení zadané úlohy jsou splněny:
Funkce $p(x) = x^2, p'(x) = 2x, q(x) = x$ a $f(x) = 1$ jsou spojité na intervalu $\langle 1, 2\rangle$, $p(x) > 0$ a $q(x) \geq 0$ na tomto intervalu.
Sestavíme soustavu diskretizačních rovnic pro neznámé hodnoty řešení v uzlových bodech $x_1 = 1,2, x_2 = 1,4, x_3 = 1,6$ a $x_4 = 1,8$.
Potřebné hodnoty funkcí p, q a f můžeme opět vypsat do tabulky:

i	0		1		2		3		4		5
x_i	1		1,2		1,4		1,6		1,8		2
$q_i = x_i$			1,2		1,4		1,6		1,8		
$f_i = 1$			1		1		1		1		
$x_{i+1/2}$		1,1		1,3		1,5		1,7		1,9	
$p_{i+1/2} = x^2_{i+1/2}$		1,21		1,69		2,25		2,89		3,61	

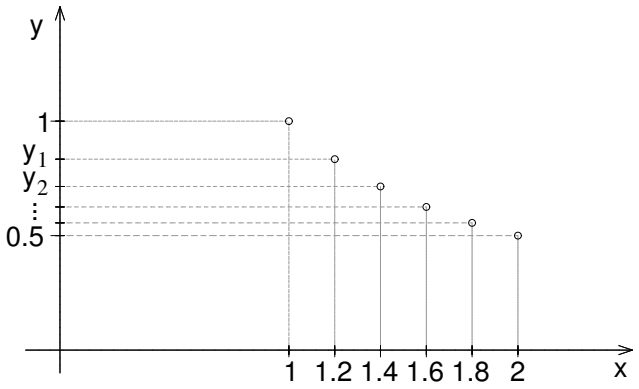
Soustava diskretizačních rovnic pak je:

$$\begin{aligned}
2,948y_1 &- 1,69y_2 &&&&= 0,04 + 1,21 \cdot 1\\
-1,69y_1 &+ 3,996y_2 &- 2,25y_3 &&&= 0,04\\
&- 2,25y_2 &+ 5,204y_3 &- 2,89y_4 &&= 0,04\\
&&- 2,89y_3 &+ 6,572y_4 &&= 0,04 + 3,61 \cdot 0.5
\end{aligned}$$

Řešení této soustavy, zaokrouhlené na čtyři desetinná místa, je v následující tabulce. Pro srovnání uvádíme i hodnoty přesného řešení v uzlových bodech.

i	0	1	2	3	4	5
x_i	1	1,2	1,4	1,6	1,8	2
y_i	1	0,8337	0,7147	0,6253	0,5557	0,5
$y(x_i)$	1	0,8333	0,7143	0,625	0,5556	0,5

Na obrázku 8.9 jsou vypočtené hodnoty znázorněny.



Obrázek 8.9: K příkladu 8.10 - nalezené přibližné řešení.

Poznámka. Každou lineární diferenciální rovnici druhého řádu

$$y'' + f_1(x)y' + f_2(x)y = f_3(x) \tag{8.39}$$

lze vhodnou úpravou převést na samoadjungovaný tvar $-(p(x)y')' + q(x)y = f(x)$, kde

$$p(x) = e^{\int f_1(x)\,\mathrm{d}x},\ q(x) = -f_2(x)e^{\int f_1(x)\,\mathrm{d}x} \text{ a } f(x) = -f_3(x)e^{\int f_1(x)\,\mathrm{d}x}.$$

(Integrační konstantu c v $\int f_1(x)\,\mathrm{d}x$ volíme rovnu nule.)

Příklad 8.11 *Převeďte na samoadjungovaný tvar rovnici*

$$y'' - 2xy' - 2y = x.$$

Řešení: Podle předchozí poznámky bude $p(x) = e^{\int(-2x)\mathrm{d}x} = e^{-x^2}$, $q(x) = -(-2e^{-x^2})$ a $f(x) = -xe^{-x^2}$. Tedy rovnice v samoadjungovaném tvaru je

$$-(e^{-x^2}y')' + 2e^{-x^2}y = -xe^{-x^2}.$$

Snadno se můžeme přesvědčit, že použitím pravidla pro derivaci součinu a následným vydělením rovnice $-e^{-x^2}$ dostaneme původně zadanou rovnici.

Poznámka. Možná čtenáře napadla otázka, proč naopak samoadjungovanou rovnici nerozepíšeme na tvar 8.39, nenahradíme zvlášť druhou a první derivaci neznámé y a neřešíme takto vzniklou soustavu rovnic. To samozřejmě udělat můžeme. Samoadjungovaný tvar ale má své výhody, rozhodně to není jen výmysl „zlých" matematiků. Mnoho úloh technické praxe vyjde jako rovnice v samoadjungovaném tvaru přímo z podstaty řešeného problému a tento typ úloh má svou podobu i u parciálních diferenciálních rovnic, tzn. u funkcí více proměnných. Další výhodou řešení rovnice v samoadjungovaném tvaru jsou výše popsané příznivé vlastnosti matice soustavy diskretizačních rovnic.

Obecnější okrajové podmínky

Zatím jsem se zabývali pouze okrajovými podmínkami tvaru 8.31, tzn. měli jsme zadány přímo hodnoty řešení v krajních bodech intervalu $\langle a, b\rangle$. V okrajových podmínkách se však může vyskytovat také první derivace hledaného řešení. Obecně mohou okrajové podmínky vypadat takto:

$$\begin{aligned} \alpha_1\, y'(a) + \alpha_2\, y(a) &= \alpha_3 \\ \beta_1\, y'(b) + \beta_2\, y(b) &= \beta_3 \end{aligned} \tag{8.40}$$

$\alpha_i, \beta_i, i = 1, 2, 3$, jsou reálná čísla. Některá z nich mohou být nulová - např. pro $\alpha_1 = \beta_1 = 0$ dostaneme již probrané okrajové podmínky 8.31, ale nesmí být současně α_1 i α_2 rovno nule ani současně β_1 i β_2 rovno nule.
Ukážeme, jak si s takovýmito okrajovými podmínkami poradit, řešíme-li okrajovou úlohu metodou sítí. V předchozí kapitole jsme ukázali, jak získáme soustavu diskretizačních rovnic s neznámými $y_1, \ldots, y_{n-1}$. V našem případě ale máme o dvě neznámé více, hodnoty řešení v krajních bodech $a = x_0$ a $b = x_n$, y_0 a y_n, nejsou okrajovými podmínkami přímo zadány. Proto musíme k soustavě diskretizačních rovnic přidat další dvě rovnice. Ty získáme z okrajových podmínek 8.40 nahrazením derivace diferencí. To můžeme provést několika způsoby:

• Derivaci nahradíme nejjednodušším možným způsobem,

$$y'(x_0) \doteq \frac{y_1 - y_0}{h}\ , \quad \text{resp.} \quad y'(x_n) \doteq \frac{y_n - y_{n-1}}{h}\ . \tag{8.41}$$

K diskretizačním rovnicím pak přidáme ještě rovnice

$$\begin{aligned} \alpha_1 \frac{y_1 - y_0}{h} &+ \alpha_2\, y_0 &= \alpha_3 \\ \beta_1 \frac{y_n - y_{n-1}}{h} &+ \beta_2\, y_n &= \beta_3 \end{aligned}$$

Tato metoda je velmi jednoduchá, má ovšem jeden háček. Vzorce 8.41 mají malou přesnost, jejich chyba je řádově h. K aproximaci derivací při sestavování diskretizačních rovnic však

obvykle používáme přesnější formule s chybou řádu h^2. Mohlo by se zdát, že přidáním dvou méně přesných rovnic se toho moc nezkazí, ale ukazuje se, že větší nepřesnost aproximace v krajních bodech ovlivní velikost chyby ve všech bodech x_i.

• Derivaci nahradíme složitějším vzorcem, zato s vyšší přesností (chyba řádu h^2):

$$y'(x_0) \doteq \frac{-3y_0 + 4y_1 - y_2}{2h}\ , \quad \text{resp.} \quad y'(x_n) \doteq \frac{y_{n-2} - 4y_{n-1} + 3y_n}{2h} \tag{8.42}$$

Použitím této metody se vyhneme ztrátě přesnosti, která se objevuje u metody předchozí. Jistou nevýhodou však je, že přidáním příslušných rovnic můžeme přijít o některé příjemné vlastnosti matice soustavy diskretizačních rovnic, např. diagonální dominanci.

• Derivaci můžeme také nahradit centrální diferencí, tj.

$$y'(x_0) \doteq \frac{y_1 - y_{-1}}{2h}\ , \quad \text{resp.} \quad y'(x_n) \doteq \frac{y_{n+1} - y_{n-1}}{2h}\ , \tag{8.43}$$

kde y_{-1} a y_{n+1} jsou hodnoty řešení v tzv. fiktivních uzlech $x_{-1} = a - h$ a $x_{n+1} = b + h$. Vzorce 8.43 mají chybu řádově h^2. Tímto způsobem jsme si ale přidali další dvě neznámé, y_{-1} a y_{n+1}, a musíme proto k soustavě přidat ještě další dvě rovnice. Ty získáme tak, že budeme požadovat platnost rovnice 8.30 i v krajních bodech x_0 a x_n, neboli platnost rovnic 8.32 i pro $i = 0$ a $i = n$.

8.2.2 Metoda střelby

Metoda střelby je další významná metoda pro řešení okrajových úloh. Zde jen nastíníme její princip, neboť na důkladné probrání v obsáhlých osnovách tohoto kursu asi stejně nezbude čas.
Základem metody střelby je převedení okrajové úlohy na úlohu počáteční.
Připomeňme, že řešíme diferenciální rovnici druhého řádu

$$y'' = f(x, y, y'), \quad y(a) = \alpha, \quad y(b) = \beta. \tag{8.44}$$

U počáteční úlohy druhého řádu musíme znát hodnotu řešení v bodě $a = x_0$ a hodnotu derivace řešení v tomto bodě:

$$y'' = f(x, y, y'), \quad y(a) = \alpha, \quad y'(a) = \gamma. \tag{8.45}$$

Kdybychom si za hodnotu derivace v bodě a určité γ zvolili, mohli bychom pomocí některé z metod popsaných v kapitole 8.1 najít přibližné řešení takovéto počáteční úlohy. Jenomže bychom se na konci s největší pravděpodobností netrefili do požadovaného β. U metody střelby je proto základní otázka: Jak zvolit hodnotu derivace v bodě a, tj. pod jakým úhlem zamířit (viz obrázky 8.7 a 8.8), abychom na konci intervalu zasáhli β, neboli aby vyšlo $y(b) = \beta$?
V podstatě se jedná o řešení rovnice

$$y(\gamma, b) = \beta \tag{8.46}$$

s neznámou γ, kde $y(\gamma, b)$ označuje hodnotu řešení počáteční úlohy s počátečními podmínkami $y(a) = \alpha$, $y'(a) = \gamma$ v bodě b.

K řešení takovéto rovnice lze použít např. analogii metody půlení intervalu z kapitoly 4. Podaří-li se nám najít γ_1 a γ_2 takové, že $y(\gamma_1, b) < \beta$ a $y(\gamma_2, b) > \beta$, vypočteme $\gamma_3 = \frac{\gamma_1+\gamma_2}{2}$ a dále pokračujeme s tou dvojicí „gam", pro kterou vychází jedna hodnota řešení v bodě b pod β a druhá nad β.

Shrnutí pojmů

Při numerickém řešení diferenciálních rovnic se nesnažíme hledané řešení vyjádřit ve tvaru funkce, ale hledáme pouze přibližné hodnoty řešení v uzlových bodech.
U počátečních úloh známe hodnotu řešení v bodě x_0 z počáteční podmínky. Přibližné hodnoty řešení v dalších bodech pak počítáme pomocí hodnoty řešení v jednom nebo několika předchozích bodech.
U jednokrokových metod používáme hodnotu řešení v jediném předchozím bodě. Nejjednodušší jednokrokovou metodou je metoda Eulerova. Nejpoužívanější z jednokrokových metod je metoda Runge-Kutta 4. řádu. Výpočet pomocí ní je sice pracný, v každém kroku musíme čtyřikrát vyčíslit funkční hodnotu pravé strany řešené diferenciální rovnice, ale to je vyváženo její vysokou přesností.
K odhadu chyby a případnému řízení délky kroku se u jednokrokových metod často používá metoda polovičního kroku, kdy do stejného bodu dojdeme zvolenou metodou jednak s krokem délky h, jednak s krokem délky $h/2$, a pomocí takto získaných výsledků odhadneme chybu.
U k-krokových metod používáme k výpočtu přibližného řešení v dalším uzlovém bodě k předchozích hodnot. Na počátku, pro výpočet v prvních k uzlech, proto musíme použít vhodnou jednokrokovou metodu a pak teprve pokračovat metodou vícekrokovou.
Vícekrokové metody se obvykle nepoužívají samostatně, ale ve dvojici - tzv. metoda prediktor-korektor. Přibližnou hodnotu řešení nejprve vypočteme pomocí explicitní vícekrokové metody, prediktoru, a pak ji zpřesníme pomocí implicitní metody, korektoru.
Soustavy diferenciálních rovnic prvního řádu se řeší velmi podobně jako jediná rovnice, až na to, že místo jediné funkce f a skalárů y_i pracujeme s vektory (n-ticemi) funkcí a hodnot řešení.
Diferenciální rovnice vyšších řádů s počátečními podmínkami se nejprve převedou na soustavu rovnic prvního řádu, kterou pak řešíme obvyklým způsobem.
Řešení okrajových úloh je od řešení počátečních úloh dosti odlišné. Opět sice hledáme řešení pouze v uzlových bodech, ale nemůžeme postupovat od uzlu k uzlu jako u počátečních úloh, musíme brát v úvahu i podmínku na konci intervalu. U metody sítí požadujeme platnost diferenciální rovnice ve všech vnitřních uzlech. Derivace vyskytující se v rovnici nahradíme diferencemi, přidáme okrajové podmínky, a tím získáme tzv. soustavu diskretizačních rovnic pro neznámé hodnoty řešení v uzlových bodech. V případě lineární diferenciální rovnice se vždy jedná o soustavu lineárních rovnic.
Speciální tvar diskretizační soustavy obdržíme pro rovnici v samoadjungovaném tvaru. Matice vzniklé soustavy lineárních rovnic má z hlediska jejího řešení příznivé vlastnosti. Na samoadjungovaný tvar lze převést každou lineární diferenciální rovnici druhého řádu, mnoho rovnic však v tomto tvaru vyjde „samo od sebe", z podstaty řešeného problému.

8.3 Otázky a příklady ke cvičení

U následujících výroků rozhodněte, zda se jedná o výrok pravdivý či nepravdivý.

Otázka 8.1 *Všechny zde probrané metody slouží pro nalezení přibližných hodnot obecného řešení zkoumané rovnice.*

Otázka 8.2 *Eulerovou metodou najdeme přibližné hodnoty řešení ve všech bodech intervalu* $\langle x_0, x_n \rangle$.

Otázka 8.3 *Globální chyba je rozdíl přesného a přibližného řešení v daném uzlovém bodě.*

Otázka 8.4 *Metody Runge-Kutta patří mezi jednokrokové metody.*

Otázka 8.5 *Chceme-li použít metodu Runge-Kutta, musíme vždy napřed najít obecné řešení zkoumané diferenciální rovnice.*

Otázka 8.6 *U k-krokových metod najdeme pomocí řešení v jediném uzlovém bodě* x_i *přibližné hodnoty řešení v k dalších uzlových bodech současně.*

Otázka 8.7 *Vícekrokové metody nelze použít samostatně, vždy je potřeba řešení v prvních několika uzlech najít pomocí vhodné jednokrokové metody.*

Otázka 8.8 *Metody prediktor-korektor jsou vždy kombinací jedné explicitní a jedné implicitní vícekrokové metody.*

Otázka 8.9 *Každá okrajová úloha má právě jedno řešení.*

Otázka 8.10 *Při řešení okrajové úlohy metodou sítí musíme vždy vyřešit soustavu rovnic.*

Otázka 8.11 *Každou lineární diferenciální rovnici druhého řádu lze převést na samoadjungovaný tvar.*

Příklad 8.1 *Eulerovou metodou najděte řešení počáteční úlohy* $y' = \frac{x}{y}, y(1) = 2$ *na intervalu* $\langle 1, 2 \rangle$ *s krokem* $h = 0{,}2$. *Najděte i přesné řešení této úlohy a vypočtěte globální chybu v každém uzlu.*
Pomocí získaných výsledků pak vypočtěte přibližnou hodnotu řešení v bodě $x = 1{,}3$ - *použijte lineární interpolaci.*

Příklad 8.2 *Řešení úlohy z příkladu 1 najděte se stejným krokem metodou Runge-Kutta 4. řádu. Opět vypočtěte globální chybu v každém uzlu.*

Příklad 8.3 *Eulerovou metodou řešte počáteční úlohu* $y' = x^2 + y^2, y(1) = -1$. *Proveďte jeden krok s* $h = 0{,}05$. *Pak metodou polovičního kroku odhadněte chybu a zpřesněte řešení.*

Příklad 8.4 *Metodou Runge-Kutta 4. řádu řešte počáteční úlohu* $y' = x^2 - y^2, y(1) = 0$. *Proveďte jeden krok s* $h = 0{,}2$. *Pak metodou polovičního kroku odhadněte chybu a zpřesněte řešení.*

Příklad 8.5 *Soustavu diferenciálních rovnic*

$$\begin{aligned} y' &= xy + z \qquad y(0) = 0 \\ z' &= y^2 \qquad\quad z(0) = 2 \end{aligned}$$

řešte metodou Runge-Kutta 4. řádu s krokem $h = 0{,}1$. *Proveďte 2 kroky.*

Příklad 8.6 *Rovnici* $y'' = \dfrac{xy'}{y}$ *s počátečními podmínkami* $y(0) = 2, y'(0) = -1$ *převeďte na soustavu dvou rovnic prvního řádu, a tu pak řešte s krokem* $h = 0{,}1$ *Eulerovou metodou. Najděte přibližné hodnoty řešení v bodech* 0,1 *a* 0,2.

Příklad 8.7 *a) Metodou sítí řešte s krokem* $h = 0{,}25$ *okrajovou úlohu* $-y'' + \dfrac{y}{x^2} = -5x$, $y(1) = 1$, $y(2) = 8$. *Prověřte, že jsou splněny podmínky zaručující existenci jediného řešení zadané úlohy.*
b) Ověřte, že $y = x^3$ *je řešením zadané okrajové úlohy. Kdyby se v řešení a) všude počítalo s přesnými čísly, bez zaokrouhlování, vyšly by hodnoty řešení v uzlových bodech metodou sítí přesně. Proč?*

Příklad 8.8 *Okrajovou úlohu* $y'' + \dfrac{y'}{x} - xy = 1$, $y(0{,}1) = 1$, $y(0{,}6) = 0$ *převeďte na samoadjungovaný tvar a pak ji vyřešte metodou sítí s krokem* $h = 0{,}1$. *Ověřte, že jsou splněny podmínky zaručující existenci jediného řešení zadané úlohy.*

Odpovědi na otázky a řešení příkladů viz 9.7

Programovací úlohy

Zda budou funkce $f(x, y), p(x), q(x)$ apod. zadány přímo v programu, nebo se budou zadávat z klávesnice, ponecháme na zkušenosti a odvaze programátora. Totéž platí pro kreslení grafu nalezeného přibližného řešení.

Programovací úloha 1 Napište program, který najde řešení počáteční úlohy $y' = f(x, y)$, $y(x_0) = y_0$ na zadaném intervalu $\langle x_0, b\rangle$ Eulerovou metodou s krokem h.

Programovací úloha 2 Napište program, který najde řešení počáteční úlohy $y' = f(x, y)$, $y(x_0) = y_0$ na zadaném intervalu $\langle x_0, b\rangle$ metodou Runge-Kutta s krokem h.

Programovací úloha 3 * Napište program, který najde řešení počáteční úlohy $y' = f(x, y)$, $y(x_0) = y_0$ na zadaném intervalu $\langle x_0, b\rangle$ Eulerovou metodou nebo metodou Runge-Kutta s přesností ε. (Použijte metodu polovičního kroku.)

Programovací úloha 4 Napište program, který najde řešení rovnice v samoadjungovaném tvaru, $-(p(x)y')' + q(x)y = f(x)$, s okrajovými podmínkami $y(a) = \alpha$, $y(b) = \beta$ metodou sítí s krokem h.

9 Výsledky

9.1 Výsledky cvičení 1.5

Otázky: 1.1-A, 1.2-N, 1.3-N, 1.4-A, 1.5-N (Např. pro $\hat{x} = 1,23$ a $\hat{y} = 2,34$ a $n = 1$ tvrzení neplatí.), 1.6-A

9.2 Výsledky cvičení 2.5

Otázky: 2.1-N, 2.2-A, 2.3-N, 2.4-A, 2.5-N, 2.6-A, 2.7-A, 2.8-N, 2.9-N.

9.3 Výsledky cvičení 3.3

Otázky: 3.1-A, 3.2 -A, 3.3-N, 3.4-N, 3.5-N, 3.6-A, 3.7-N, 3.8-N, 3.9-A.

Výsledky příkladů

ad 3.1 $x \doteq 0,68, y \doteq -0,06$.

ad 3.2 $x \doteq -1,29, y \doteq 0,36, z \doteq -0,35$.

ad 3.3 Podm. konv. jsou splněny - matice soustavy je ryze řádkově diag. dominantní. $(x^{(1)}, y^{(1)}, z^{(1)}) = (-0,6; 0,52; -0,3125)$, $(x^{(2)}, y^{(2)}, z^{(2)}) \doteq (-0,7561; 0,5305; -0,4688)$, $(x^{(3)}, y^{(3)}, z^{(3)}) \doteq (-0,7967; 0,5242; -0,4899)$,

ad 3.4 Podm. konv. jsou splněny - matice soustavy je ryze řádkově diag. dominantní. $(x^{(1)}, y^{(1)}, z^{(1)}) \doteq (0,8929; 0,3929; -0,5020)$, $(x^{(2)}, y^{(2)}, z^{(2)}) \doteq (1,0129; 0,3339; -0,5004)$, $(x^{(3)}, y^{(3)}, z^{(3)}) \doteq (1,0002; 0,3333; -0,5000)$, $(x^{(4)}, y^{(4)}, z^{(4)}) \doteq (1,0000; 0,3333; -0,5000)$, přesnosti je dosaženo, $(x, y, z) \doteq (1,000; 0,333; -0,500)$.

ad 3.5 Pro zadanou soustavu podm. konv. nejsou splněny. Soustavu můžeme vynásobit maticí A^T. Tím dostaneme soustavu, jejíž matice je symetrická a pozitivně definitní, což zaručuje konvergenci G.-S. metody. Takto vzniklá soustava je

$$\begin{array}{rcrcrcr} 17x & - & 6y & - & 5z & = & -28 \\ -6x & + & 5y & + & 8z & = & 7 \\ -5x & + & 8y & + & 17z & = & 2 \end{array}$$

První dvě iterace: $(x^{(1)}, y^{(1)}, z^{(1)}) \doteq (-1,647; -0,576; -0,096)$, $(x^{(2)}, y^{(2)}, z^{(2)}) \doteq (-1,879; -0,702; -0,105)$. (Jiná možnost úpravy soustavy je pomocí přehazování rovnic a přičítání vhodných násobků jedné rovnice k druhé docílit toho, aby matice soustavy byla diag. dom. Tento postup však vyžaduje značnou dávku štěstí a pokud není vhodná úprava na první pohled patrná, nelze ho doporučit.)

9.4 Výsledky cvičení 4.3

Otázky: 4.1-A, 4.2-N, 4.3-N, 4.4-A, 4.5-N, 4.6-A, 4.7-A, 4.8-A, 4.9-N, 4.10-N.

Výsledky příkladů

ad 4.1 Rovnice má právě 2 kořeny. Větší je v int. $\langle 2,3\rangle$. Půlení: $\langle 2{,}5;3\rangle$, $\langle 2{,}5;2{,}75\rangle$, $\langle 2{,}625;2{,}75\rangle$, přesnosti je dosaženo, $x \doteq 2{,}7$.
Menší je v int. $\langle 1,2\rangle$. Regula falsi: $x_0 = 1{,}148$, $x_1 = 1{,}068$, $x_2 = 1{,}065$, přesnosti je dosaženo, $x \doteq 1{,}06$.

ad 4.2 Kořen leží v $\langle -2,-1\rangle$. $x_0 = -2$, $x_1 = -1,645161$, $x_2 = -1,485724$, $x_3 = -1,453806$, $x_4 = -1,452628$, $x_5 = -1,452627$. $x \doteq -1,45263$.

ad 4.3 Rovnice má dva kořeny. Pro kořen z intervalu $\langle 0,1\rangle$ je vhodná např. iterační fce $g(x) = \mathrm{e}^{\frac{x}{2}-1}$: $x_0 = 1$, $x_1 = 0{,}607$, $x_2 = 0{,}498$, $x_3 = 0{,}472$, $x_4 = 0{,}466$, $x \doteq 0{,}47$.
Pro kořen z $\langle 5,6\rangle$ $g(x) = 2\ln x + 2$: $x_0 = 5$, $x_1 = 5{,}219$, $x_2 = 5{,}305$, $x_3 = 5{,}337$, $x_4 = 5{,}349$, $x_5 = 5{,}354$, $x \doteq 5{,}35$

ad 4.4 $x \doteq 0{,}31416$.

ad 4.5 Hledáme kořen rovnice $f'(x) = 0$. Vyjde $x \doteq 0{,}42$. Ověření, že jde skutečně o lok. maximum, lze provést např. pomocí f''.

ad 4.6 $(x_1,y_1) = (1{,}25;0{,}25)$, $(x_2,y_2) = (1{,}2332;0{,}2126)$, $(x_3,y_3) = (1{,}2333;0{,}2122)$, přesnosti je dosaženo, $(x,y) \doteq (1{,}233;0{,}212)$.

ad 4.7 Vhodné iterační funkce jsou např. $g_1(x,y) = \sqrt{x-y+0{,}5}, g_2(x,y) = \frac{x^2-y}{5x}$. Zvolíme-li $(x_0,y_0) = (1,0)$, s těmito funkcemi bude $(x_1,y_1) = (1{,}2247;0{,}2)$, $(x_2,y_2) = (1{,}2348;0{,}2123)$, $(x_3,y_3) = (1{,}2339;0{,}2128)$, přesnosti je dosaženo.

ad 4.8 $(x_1,y_1,z_1) = (3/4,5/3,3/4)$.

ad 4.9 Návod: Najděte rovnici tečny ke grafu funkce f v bodě $[x_k, f(x_k)]$ a pak průsečík tečny s osou x.

ad 4.10 Návod: Najděte rovnici přímky dané body $[a,f(a)]$ a $[b,f(b)]$ a pak průsečík této přímky s osou x.

9.5 Výsledky cvičení 5.4

Otázky: 5.1-A, 5.2-A, 5.3-N, 5.4-A, 5.5-A, 5.6-N, 5.7-N, 5.8-A.

Výsledky příkladů

ad 5.1 $L_2(x) = 2x^2 - x + 3$. Zkouška: Ověříme, že $L_2(-1) = 6$, $L_2(0) = 3$ a $L_2(2) = 9$.

ad 5.2 $L_2(x) = \frac{f_0-2f_1+f_2}{2h^2}(x-x_1)^2 + \frac{f_2-f_0}{2h}(x-x_1) + f_1$.

ad 5.3 $N_2(x) = 6 - 3(x+1) + 2(x+1)x$. Po přidání dalšího bodu: $N_3(x) = N_2(x) - 0{,}65(x+1)x(x-2)$.

ad 5.4 a) Uzly jsou ekvidistantní. $N_4(x) = 0 + \frac{q}{1!} \cdot 0{,}7174 - \frac{q(q-1)}{2!} \cdot 0{,}4351 - \frac{q(q-1)(q-2)}{3!} \cdot 0{,}1712 + \frac{q(q-1)(q-2)(q-3)}{4!} \cdot 0{,}3678$, $q = \frac{x-0}{0{,}8}$ Pro $x = 1$ je $q = 1{,}25$, $\sin 1 \doteq N_4(1) \doteq 0{,}8417$. b) Použijeme uzly $x_1 = 0{,}8, x_2 = 1{,}6$. Lineární interp. pol. lze vyjádřit např. takto: $N_1(x) = 0{,}7174 + q \cdot 0{,}2822$, $q = \frac{x-0{,}8}{0{,}8}$. $\sin 1 \doteq N_1(1) \doteq 0{,}7879$ (za q se dosadilo 0,25). Přesná hodnota je $\sin 1 \doteq 0{,}8415$. Pro lineární interpolaci byl krok mezi uzly příliš velký.

ad 5.5 Soustava, kterou je nutno vyřešit: $6c_1 + c_2 = -9$; $c_1 + 6c_2 = -159$. Splajn:
$x \in \langle -3, -1\rangle : S_0(x) = -5 + 2(x+3) + 0{,}5(x+3)^3$
$x \in \langle -1, 0\rangle : S_1(x) = 3 + 8(x+1) + 3(x+1)^2 - 10(x+1)^3$
$x \in \langle 0, 2\rangle : S_2(x) = 4 - 16x - 27x^2 + 4{,}5x^3$
$S(-2) = S_0(-2) = -2{,}5$, $S(-0{,}1) = S_1(-0{,}1) = 5{,}34$, $S(1) = S_2(1) = -34{,}5$.

ad 5.6 Soustava, kterou je nutno vyřešit: $3{,}2c_1 + 0{,}8c_2 = -1{,}6318$; $0{,}8c_1 + 3{,}2c_2 + 0{,}8c_3 = -2{,}2737$; $0{,}8c_2 + 3{,}2c_3 = -1{,}5365$.
Splajn: $x \in \langle 0; 0{,}8\rangle : S_0(x) = 0{,}9974x - 0{,}1573x^3$,
$x \in \langle 0{,}8; 1{,}6\rangle : S_1(x) = 0{,}7174 + 0{,}6953(x-0{,}8) - 0{,}3776(x-0{,}8)^2 - 0{,}0631(x-0{,}8)^3$
$x \in \langle 1{,}6; 2{,}4\rangle : S_2(x) = 0{,}9996 - 0{,}0302(x-1{,}6) - 0{,}5292(x-1{,}6)^2 + 0{,}0755(x-1{,}6)^3$
$x \in \langle 2{,}4; 3{,}2\rangle : S_3(x) = 0{,}6755 - 0{,}7318(x-2{,}4) - 0{,}3479(x-2{,}4)^2 + 0{,}1449(x-2{,}4)^3$
$\sin 1 \doteq S_1(1) \doteq 0{,}8408$

ad 5.7 $f'(x) = 1 + \mathrm{e}^x > 0 \Rightarrow f$ je rostoucí $\Rightarrow f$ je prostá $\Rightarrow$ existuje funkce k ní inverzní (neboli ke každému $y \in H(f)$ lze jednoznačně určit x takové, že $f(x) = y$). Hodnoty inverzní funkce pro $y = 0$, $y = 0{,}5$ a $y = 1$ najdeme postupně jako řešení rovnic $x + \mathrm{e}^x = 0$, $x + \mathrm{e}^x = 0{,}5$ a $x + \mathrm{e}^x = 1$. Vyjde $f^{-1}(0) \doteq -0{,}567$, $f^{-1}(0{,}5) \doteq -0{,}266$, $f^{-1}(1) = 0$.
Interpolační polynom (v Newtonově tvaru): $N_2(x) = -0{,}567 + \frac{q}{1!} \cdot 0{,}301 - \frac{q(q-1)}{2!} \cdot 0{,}035$, $q = \frac{x-0}{0{,}5}$. $f^{-1}(0{,}3) \doteq N_2(0{,}3) \doteq -0{,}382$ (za q se dosadí 0,6), $f^{-1}(0{,}9) \doteq N_2(0{,}9) \doteq -0{,}050$ (za q se dosadí 1,8).

ad 5.8 Soustava normálních rovnic: $6c_0 + 15c_1 = 3,579; 15c_0 + 55c_1 = 28,939$. Přímka: $y = -2,259 + 1,142\,x$.

ad 5.9 $y = 7,340 - 8,243\,x + 2,047\,x^2$

ad 5.10 Soustava normálních rovnic obecně:
$c_0(n+1) + c_1 \sum \sin x_i + c_2 \sum \cos x_i = \sum y_i$
$c_0 \sum \sin x_i + c_1 \sum \sin^2 x_i + c_2 \sum \sin x_i \cos x_i = \sum y_i \sin x_i$
$c_0 \sum \cos x_i + c_1 \sum \sin x_i \cos x_i + c_1 \sum \cos^2 x_i = \sum y_i \cos x_i$.
Konkrétně pro zadané body:
$11c_0 - 1,47c_2 = 13,43$; $6,09c_1 = 6,31$; $-1,47c_0 + 4,91c_2 = -10.47$.
Řešení: $y = 0,98 + 1,04 \sin x - 1,84 \cos x$.

9.6 Výsledky cvičení 6.3

Otázky: 6.1-N, 6.2-A, 6.3-N (byla by to pravda, kdybychom se nedopouštěli zaokrouhlovacích chyb), 6.4-N, 6.5-A, 6.6-A, 6.7-A.

Výsledky příkladů

ad 6.1 a) Např. podle 6.1 ve všech kromě posledního uzlu, v něm podle 6.2:
$G'(1) \doteq 0{,}3750$, $G'(1,1) \doteq 0,3010$, $G'(1,2) \doteq 0,2370$, $G'(1,3) \doteq 0,2370$.
b) $G'(1) \doteq 0{,}4120$, $G'(1,1) \doteq 0,3380$, $G'(1,2) \doteq 0,2690$, $G'(1,3) \doteq 0,2050$.
Přesně: $G'(x) = \frac{2}{\sqrt{\pi}}\mathrm{e}^{-x^2}$. Zaokrouhleno na 4 desetinná místa: $G'(1) = 0{,}4151$, $G'(1,1) = 0,3365$, $G'(1,2) = 0,2673$, $G'(1,3) = 0,2082$.

ad 6.2 Návod: $L_2(x)$ zderivujte, do derivace dosaďte jednotlivé uzly. Pro vzorec 6.6 vypočtěte druhou derivaci L_2.

ad 6.3 Návod: Vypočtěte $\int_{x_1-h}^{x_1+h} L_2(x)\,\mathrm{d}x$. Je vhodné použít substituci za $x - x_1$.

ad 6.4 a) $\frac{\pi}{4} \doteq 0,79$ b) $\frac{\pi}{12}(2\sqrt{2}+1) \doteq 1,002$. Přesně: 1.

ad 6.5 a) $L_4 \doteq 0,6586$ b) $L_8 = L_4/2 + 0,125(f(1,125)+f(1,375)+f(1,625)+f(1,875)) \doteq 0,6592$. (Přesně 0,6593)

ad 6.6 $S_6 \doteq 0,9103147$ (přesně 0,9103140).

ad 6.7 $L_4 = 1,55$, $S_4 \doteq 1,567$, přesně $\frac{\pi}{2} \doteq 1,571$.

ad 6.8 $S_4 \doteq 0,31$

ad 6.9 $f''(x) = 1/(1+x^2)^{3/2}$, maximum f'' na intervalu $\langle 0,1\rangle$ je $1 \Rightarrow |E| \le \frac{(1-0)^3}{12\cdot 4^2}\cdot 1 = \frac{1}{192} \doteq 0,005$.

ad 6.10 $f^{(4)}(x) = -2\frac{1+2\cos^2 x}{\sin^4 x}$, $|f^{(4)}(x)| = 2\frac{1+2\cos^2 x}{\sin^4 x}$. To je funkce na intervalu $\langle \pi/4, \pi/2\rangle$ klesající $\Rightarrow$ dosahuje maxima pro $x = \pi/4$, $|f^{(4)}(\pi/4)| = 16$. m najdeme tak, aby $\frac{(\pi/4)^5}{180m^4}\cdot 16 < 10^{-4}$. Vyjde $m > 14,8$, tedy $m = 16$.

9.7 Výsledky cvičení 8.3

Otázky: 8.1-N, 8.2-N, 8.3-A, 8.4-A, 8.5-N, 8.6-N, 8.7-A, 8.8-A, 8.9-N, 8.10-A, 8.11-A.

Výsledky příkladů

ad 8.1 $x_0 = 1, y_0 = 2$; $x_1 = 1{,}2, y_1 = 2{,}1$; $x_2 = 1{,}4, y_2 \doteq 2{,}214$; $x_3 = 1{,}6, y_3 \doteq 2{,}341$; $x_4 = 1{,}8, y_4 \doteq 2{,}477$; $x_5 = 2, y_5 \doteq 2{,}623$. Přesné řešení je $y = \sqrt{x^2+3}$. Chyby: $e_1 \doteq 0{,}007$, $e_2 \doteq 0{,}013$, $e_3 \doteq 0{,}017$, $e_4 \doteq 0{,}021$, $e_5 \doteq 0{,}023$.
Přibližnou hodnotu řešení v „neuzlovém" bodě 1,3 vypočteme pomocí interpolačního polynomu s uzly x_1 a x_2 (protože 1,3 leží v intervalu $\langle x_1, x_2\rangle$). Interpolační polynom: $L_1(x) = 2{,}1\,\frac{x-1{,}4}{-0{,}2} + 2{,}214\,\frac{x-1{,}2}{0{,}2}$, $y(1{,}3) \doteq L(1{,}3) = 2{,}157$.

ad 8.2 $y_0 = 2$, $y_1 \doteq 2{,}1071309$, $y_2 \doteq 2{,}2271059$, $y_3 \doteq 2{,}3579654$, $y_4 \doteq 2{,}4979994$, $y_5 \doteq 2{,}6457516$. Chyby: $e_1 \doteq 10^{-7}$, $e_2, e_3, e_4 \doteq 2\cdot 10^{-7}$, $e_5 \doteq 3\cdot 10^{-7}$.

ad 8.3 S krokem $h = 0{,}05 : y(1{,}05; 0{,}05) = -0{,}9$. S krokem $h = 0{,}025 : y(1{,}05; 0{,}025) \doteq -0{,}9012$. Chyba hodnoty dosažené s $h = 0{,}05$ je přibližně $\frac{2^1}{2^1-1}(y(1{,}05; 0{,}025) - y(1{,}05; 0{,}05)) \doteq -0{,}0024$, chyba pro poloviční krok je přibližně $\frac{1}{2^1-1}(y(1{,}05; 0{,}025) - y(1{,}05; 0{,}05)) \doteq -0{,}0012$. Zpřesněná hodnota řešení v bodě $x = 1{,}05$: $\frac{2^1 y(1{,}05;0{,}025) - y(1{,}05;0{,}05)}{2^1-1} \doteq -0{,}9023$. (Pro srovnání, přesná hodnota, zaokrouhl. na 4 místa, je $-0{,}9022$)

ad 8.4 S krokem $h = 0{,}2 : y(1{,}2; 0{,}2) = 0{,}23913405$. S krokem $h = 0{,}1 : y(1{,}2; 0{,}1) \doteq 0{,}23914827$. Chyba hodnoty dosažené s $h = 0{,}2$ je přibližně $\frac{2^4}{2^4-1}(y(1{,}2; 0{,}1) - y(1{,}2; 0{,}2)) \doteq 2 \cdot 10^{-5}$, chyba pro poloviční krok je přibližně $\frac{1}{2^4-1}(y(1{,}2; 0{,}1) - y(1{,}2; 0{,}2)) \doteq 9 \cdot 10^{-7}$. Zpřesněná hodnota řešení v bodě $x = 1{,}2$: $\frac{2^4 y(1{,}2;0{,}1) - y(1{,}2;0{,}2)}{2^4-1} \doteq 0{,}23914922$. (Pro srovnání, přesná hodnota, zaokrouhl. na 8 míst, je 0,23914919)

ad 8.5 1. krok: $x_1 = 0{,}1$, $y_1 \doteq 0{,}200701$, $z_1 \doteq 2{,}001339$
2. krok: $x_2 = 0{,}2$, $y_2 \doteq 0{,}405919$, $z_2 \doteq 2{,}010853$

ad 8.6 Příslušná soustava rovnic: $y' = z$, $y(0) = 2$; $z' = xz/y$, $z(0) = -1$.
Řešení soustavy: $x_1 = 0{,}1$, $y_1 = 1{,}9$, $z_1 = -1$; $x_2 = 0{,}2$, $y_2 = 1{,}8$, $z_2 \doteq -1{,}005$. Přibližné řešení původní rovnice druhého řádu v bodě 0,1, resp. 0,2, je $y_1 = 1{,}9$, resp. $y_2 = 1{,}8$.

ad 8.7 a) Funkce $\sigma(x) = \frac{1}{x^2}$ a $f(x) = -5x$ jsou na intervalu $\langle 1, 2\rangle$ spojité a $\sigma(x) > 0 \Rightarrow$ okrajová úloha má jediné řešení.
Soustava diskr. rovnic: $2{,}0400y_1 - y_2 = 0{,}6094$; $-y_1 + 2{,}0278y_2 - y_3 = -0{,}4688$; $-y_2 + 2{,}0204y_3 = 7{,}4531$
Přibližné řešení:$x_0 = 1$, $y_0 = 1$; $x_1 = 1{,}25$, $y_1 = 1{,}9531$; $x_2 = 1{,}5$, $y_2 = 3{,}3750$; $x_3 = 1{,}75$, $y_4 = 5{,}3594$; $x_5 = 2$, $y_5 = 8$.
b) Ověření: $L = -(x^3)'' + \frac{x^3}{x^2} = -6x + x = -5x = P$, $y(1) = 1^3 = 1$, $y(2) = 2^3 = 8$.
Řešení metodou sítí vyjde přesně, protože použitý diferenční vzorec $y''(x_i) = \frac{y(x_{i-1}) - 2y(x_i) + y(x_{i+1})}{h^2}$ je přesný pro polynomy stupně třetího - chyba je $-\frac{h^2}{12}y^{(4)}(\xi)$ (viz vzorec 6.6). Pro $y(x) = x^3$ je chyba rovna 0.

ad 8.8 Samoadjungovaný tvar: $-(xy')' + x^2y = -x$. Existence jediného řešení je zaručena, protože $p(x) = x$, $p'(x) = 1$, $q(x) = x^2$ i $f(x) = -x$ jsou na intervalu $\langle 0{,}1; 0{,}6\rangle$ spojité funkce a $p(x) > 0$, $q(x) \geq 0$ na tomto intervalu.
Soustava diskr. rovnic: $0{,}4004y_1 - 0{,}25y_2 = 0{,}148$; $-0{,}25y_1 + 0{,}6009y_2 - 0{,}35y_3 = -0{,}003$; $-0{,}35y_2 + 0{,}8016y_3 - 0{,}45y_4 = -0{,}004$; $-0{,}45y_3 + 1{,}0025y_4 = -0{,}005$
Přibližné řešení soustavy: $x_0 = 0{,}1$, $y_0 = 1$; $x_1 = 0{,}2$, $y_1 = 0{,}5923$; $x_2 = 0{,}3$, $y_2 = 0{,}3566$; $x_3 = 0{,}4$, $y_3 = 0{,}1977$; $x_4 = 0{,}5$, $y_4 = 0{,}0838$; $x_5 = 0{,}6$, $y_5 = 0$.

Reference

[1] Germund Dahlquist, Ake Björck : Numerical Mathematics and Scientific Computation

[2] Horová, I. : Numerické metody. Skriptum PřF MU Brno, 1999

[3] Maroš B., Marošová M. : Numerické metody I. Skriptum VUT FSI, Brno 2003

[4] Ralston, A. : Základy numerické matematiky. Praha, Academia 1978

[5] Vitásek, E. : Numerické metody. Praha, SNTL 1987