Numerická stabilita algoritmů

1.1 Počítání v aritmetice s konečnou přesností

Omezené možnosti konečné aritmetiky

Protože počítače používají k reprezentaci reálného čísla pouze konečný počet bitů, reprezentuje množina všech čísel, které lze zobrazit na počítači, pouze konečnou podmnožinu reálných čísel. To s sebou přináší dvě omezení, kterých si musíme být při počítání v konečné aritmetice vědomi:

- reprezentovaná čísla *nemohou* být libovolně velká či libovolně malá,
- mezi dvěma reprezentovanými čísly je *mezera*.

V IEEE aritmetice s dvojitou přesností je největší zobrazitelné číslo 1.79×10^{308} a nejmenší 2.23×10^{-308}, což představuje dostatečné rozpětí čísel pro většinu výpočtů.

Na druhou stranu, mezery mezi čísly v konečné aritmetice počítače představují problém, který se může projevit v podobě nepřesných výsledků. Například v IEEE aritmetice s dvojnásobnou přesností je reálný interval $[1, 2]$ reprezentován diskrétní podmnožinou

$$1, \quad 1 + 2^{-52}, \quad 1 + 2 \times 2^{-52}, \quad 1 + 3 \times 2^{-52}, \ldots, \quad 2\,.$$

Interval $[2, 4]$ je reprezentován stejnou diskrétní podmnožinou násobenou dvěmi,

$$2, \quad 2 + 2^{-51}, \quad 2 + 2 \times 2^{-51}, \quad 2 + 3 \times 2^{-51}, \ldots, \quad 4\,,$$

a obecně interval $[2^j, 2^{j+1}]$ je reprezentován stejnou množinou přenásobenou 2^j. Jinak řečeno, v IEEE aritmetice s dvojitou přesností není mezera mezi dvěma čísly nikdy větší v relativním smyslu než $2^{-52} \approx 2.22 \times 10^{-16}$.

Čísla v aritmetice s pohyblivou řádovou čárkou

Reálné číslo je v konečné aritmetice počítače reprezentováno jako (racionální) číslo s pohyblivou řádovou čárkou ve tvaru

$$z = \pm \frac{m}{\beta^t} \times \beta^e, \tag{1.1}$$

kde celé číslo $\beta \geq 2$ (obvykle $\beta = 2$) je nazýváno *základem*, celé číslo $t \geq 1$ určuje *přesnost*, celé číslo m pohybující se v rozsahu $0 \leq m \leq \beta^t$ je nazýváno *mantisou* a celočíselný parametr e *exponentem*.

Číslo z lze přepsat v názornějším tvaru

$$z = \pm\beta^e \times \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \ldots + \frac{d_t}{\beta^t}\right) = \pm\beta^e \times (0.d_1 d_2 \ldots d_t)_\beta, \tag{1.2}$$

kde každá číslice d_i leží v intervalu $0 \leq d_i \leq \beta - 1$ a $(0.d_1 d_2 \ldots d_t)_\beta$ představuje číslo zapsané v číselné soustavě se základem β. Pokud bychom nekladli žádná další omezení, bylo by možné jedno číslo reprezentovat v uvažované aritmetice mnoha způsoby, např. $0.1 = 0.01 \times 10^1 = 0.001 \times 10^2$ atd. Proto je výhodné zavést konvenci, že první číslice mantisy je nenulová ($d_1 \neq 0$), je-li z nenulové číslo, tj. klademe omezující podmínku $m \geq \beta^{t-1}$ pro $z \neq 0$. Potom je mantisa určena jednoznačně. Systém dodržující tuto konvenci nazýváme *normalizovaný*. Množina všech čísel s pohyblivou řádovou čárkou $\mathbf{F}$ je plně určena parametry β, t a horní, resp. dolní mezí celočíselného exponentu. Čísla v pohyblivé řádové čárce *nejsou* vzhledem k $\mathbb{R}$ rovnoměrně rozložena.

Zaokrouhlovací jednotka (strojová přesnost)

Přesnost zobrazení reálného čísla v množině $\mathbf{F}$ je charakterizována *zaokrouhlovací jednotkou* (strojovou přesností)

$$\mathbf{u} \equiv \frac{1}{2}\beta^{1-t}.$$

Toto číslo je polovinou vzdálenosti mezi číslem 1 a nejbližším větším číslem z $\mathbf{F}$. Strojová přesnost má následující vlastnost vypovídající o relativní přesnosti, s jakou jsme schopni aproximovat reálné číslo pomocí čísla z množiny $\mathbf{F}$. *Pro každé aproximovatelné reálné číslo x existuje číslo $x' \in \mathbf{F}$ takové, že platí*

$$\frac{|x - x'|}{|x|} \leq \mathbf{u}.$$

Aproximovatelným reálným číslem rozumíme reálné číslo, které leží mezi nejmenším a největším zobrazitelným číslem aritmetiky $\mathbf{F}$. Nechť $\mathrm{fl} : \mathbb{R} \to \mathbf{F}$ je funkce, která reálnému číslu přiřazuje nejbližší číslo z $\mathbf{F}$. Předchozí nerovnost lze také formulovat tak, že pro každé aproximovatelné reálné číslo x existuje ϵ, $|\epsilon| \leq \mathbf{u}$ takové, že

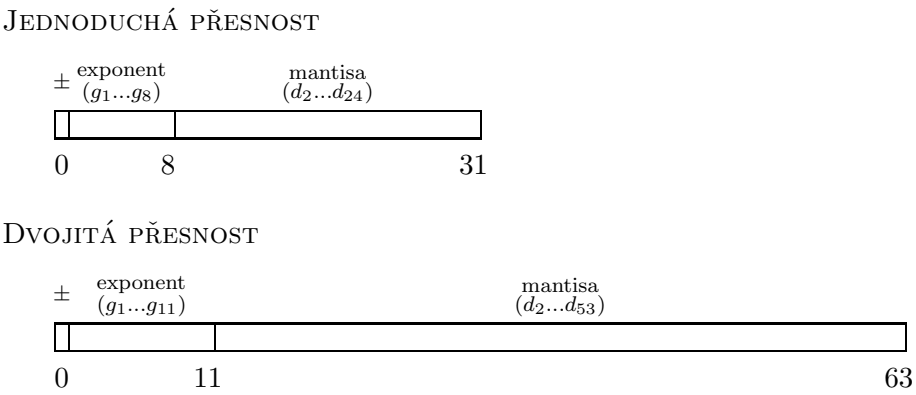
$$\mathrm{fl}(x) = x(1 + \epsilon).$$

V IEEE aritmetice s jednoduchou přesností (single precision - viz. dále) je možné zobrazit čísla v rozsahu $10^{\pm 38}$ se strojovou přesností $\mathbf{u} = 2^{-24} \approx 5.96 \times 10^{-8}$, v aritmetice s dvojitou přesností (double precision) potom čísla v rozsahu $10^{\pm 308}$ se strojovou přesností $\mathbf{u} = 2^{-53} \approx 1.11 \times 10^{-16}$.

IEEE aritmetika

IEEE aritmetika používá $\beta = 2$ a rozlišuje dva základní formáty čísel v pohyblivé řádové čárce: čísla s *jednoduchou* a *dvojitou* přesností (single and double precision). V prvém případě je k uložení čísla použito 32, ve druhém 64 bitů. Uložení jednotlivých parametrů je patrné ze schematu na obrázku 3.1. V případě jednoduché přesnosti je na exponent vyhrazeno 8 bitů, do kterých je možné uložit celé číslo v rozmezí 0 až 255. Čísla 0 a 255 mají speciální význam. Zbylá čísla 1 až 254 určují hodnotu veličiny $e + 126$, tj. hodnota exponentu e se pohybuje v rozmezí $-125 \leq e \leq 128$. Na mantisu je vyhrazeno zbývajících 23 bitů, přičemž se standardně využívá normalizace, cifra $d_1 = 1$ se přitom nezapisuje.

Je-li řetězec mantisy i exponentu nulový (tj. znaménkový bit je 0 nebo 1, všechny ostatní bity jsou nulové) dostaneme reprezentaci čísel ± 0 ($+0$ a -0 mají odlišnou reprezentaci, samozřejmě je zajištěno, že platí $+0 = -0$). Je-li řetězec exponentu roven $(1111\ 1111)_2 = 255$ a mantisa je nulová, pak zobrazené číslo je definováno jako $\pm$`Inf` ($\pm\infty$). Je-li řetězec exponentu roven $(1111\ 1111)_2 = 255$ a řetězec mantisy je nenulový (jeho hodnota je libovolná nenulová), pak je obsah interpretován jako `NaN` (Not a Number).



Obrázek 1.1: Uložení parametrů IEEE standardni aritmetiky pro jednoduchou a dvojitou přesnost.

Standardní model konečné aritmetiky z matematického hlediska

Pro počítání v konečné aritmetice počítače platí určitá pravidla, která jsou důležitá z matematického hlediska. Standardní model konečné aritmetiky předpokládá platnost vztahu

$$\mathrm{fl}(x \circ y) = (x \circ y)(1+\epsilon), \qquad |\epsilon| \leq \mathbf{u},$$

kde $\circ$ zastupuje operace $+, -, *, /$ a $\mathrm{fl}(\cdot)$ označuje, že výpočet byl proveden v konečné aritmetice počítače. Podrobnější informace lze nalézt v knihách [1, 2, 3].

1.2 Přímá a zpětná analýza chyb

Uvažujme nějaký algoritmus, který řeší daný matematický problém. Použijeme-li tento algoritmus k výpočtu v aritmetice s pohyblivou řádovou čárkou, můžeme vlivem zaokrouhlovacích chyb získat nepřesné výsledky. Úkolem numerické analýzy (či analýzy zaokrouhlovacích chyb) je zjistit, jakým způsobem funguje daný algoritmus v prostředí konečné aritmetiky, a jak nepřesné výsledky může algoritmus generovat. Jinak řečeno, ptáme se, zda je algoritmus spolehlivý (zda dává rozumné výsledky) i v případě, kdy počítá s nepřesnými čísly.

Přesnost výpočtu

Nechť $f(x)$ reprezentuje výsledek spočtený přesným algoritmem a $\mathrm{fl}(f(x))$ výsledek spočtený algoritmem v konečné aritmetice počítače, tj. výsledek je ovlivněný zaokrouhlovacími chybami (x

reprezentuje vstupní data). Přesnost výpočtu můžeme měřit v *absolutním* smyslu pomocí $\|f(x) - \mathrm{fl}(f(x))\|$, kde $\|\cdot\|$ je vhodná norma, nebo v *relativním* smyslu pomocí

$$\frac{\|f(x) - \mathrm{fl}(f(x))\|}{\|f(x)\|}. \tag{1.3}$$

V této přednášce budeme k měření přesnosti algoritmu používat relativní chybu (3.3). Je zřejmé, že při počítání v konečné aritmetice je relativní chyba výpočtu v nejlepším případě úměrná strojové přesnosti (lepší relativní přesnosti nelze obecně dosáhnout),

$$\frac{\|f(x) - \mathrm{fl}(f(x))\|}{\|f(x)\|} = \mathcal{O}(\mathbf{u}),$$

kde $\mathcal{O}(\mathbf{u})$ označuje neurčité číslo úměrné strojové přesnosti. V praxi však občas bývá problém f špatně podmíněn, a špatná podmíněnost se poté samozřejmě promítne i do přesnosti spočteného výsledku.

Význam $\mathcal{O}(\mathbf{u})$

Přesněji řečeno, výraz $\mathcal{O}(\mathbf{u})$ označuje neurčité číslo, jehož velikost lze omezit pomocí

$$|\mathcal{O}(\mathbf{u})| \leq K\mathbf{u},$$

kde K je konstanta. Konstanta K je nezávislá na konkrétních datech, ale může být závislá na dimenzi úlohy. Jsou-li např. vstupem algoritmu čtvercová matice A řádu n a vektor b, pak konstanta K nezávisí na A a b, ale může záviset na n. Vliv zaokrouhlovacích chyb se obecně zvyšuje, pokud se zvyšuje dimenze problému. Tento růst související s dimenzí bývá obvykle velmi pomalý a nezpůsobuje vážné problémy. Závislost konstanty K na dimenzi problému je obvykle lineární, kvadratická či kubická v nejhorším případě. Vznikající chyby jsou pro většinu dat mnohem menší (díky krácení - kancelaci - zaokrouhlovacích chyb) než předpovídá odhad omezující chybu v nejhorší možném případě. Pokud je však závislost konstanty K na dimenzi problému exponenciální, např. $K = 2^n$, potom může být algoritmus velmi citlivý na dimenzi problému, a omezení chyb výrazem $2^n\mathbf{u}$ přestává s rostoucím n poskytovat užitečnou výpověď o velikosti chyb (s takovým případem se setkáme v další kapitole).

Analýza chyb

Smyslem analýzy chyb je zjistit, jaké (největší) chyby se algoritmus může při výpočtu v konečné aritmetice dopustit. To, jak algoritmus v konečné aritmetice „chybuje“ lze měřit a analyzovat dvěma základními způsoby.

- **Přímá analýza chyb**. Postupujeme podle algoritmu a snažíme se popsat šíření elementárních zaokrouhlovacích chyb. Na základě tohoto popisu odhadujeme přímo velikost výsledné chyby. Efektivní přímý odhad chyby je však možný jen zřídka (v případě jednoduchých výpočtů jako je např. skalární součin vektorů, násobení matice vektorem, apod.).
- **Zpětná analýza chyb**. Hledáme modifikovaná vstupní data úlohy tak, aby *přibližné* řešení spočtené algoritmem v konečné aritmetice počítače bylo *přesným* řešením té samé úlohy, avšak s modifikovanými vstupními daty. Cílem zpětné analýzy je interpretovat zaokrouhlovací chyby vzniklé při výpočtu v aritmetice s konečnou přesností pomocí změn vstupních dat.

Pojem přímé a zpětné analýzy chyb budeme demonstrovat na příkladu skalárního součinu.

Přímá a zpětná analýza algoritmu na výpočet skalárního součinu

Pokud máme posloupnost operací jako např.

$$s = x^T y, \qquad x, y \in \mathbf{F}^n, \tag{1.4}$$

kde $\mathbf{F}^n$ označuje vektory délky n, jejichž prvky jsou čísla v pohyblivé řádové čárce ($x_i \in \mathbf{F}$, $y_i \in \mathbf{F}$, $i = 1, \ldots, n$), potom budeme výsledek spočtený algoritmem

```
01  s = 0
02  for j = 1 : n do
03        s = s + x_j y_j
04  end
```

v konečné aritmetice počítače zapisovat ve tvaru

$$s = \mathrm{fl}(x^T y).$$

Nechť $s_0 = 0$ a nechť s_j je označuje hodnotu s na konci j-tého cyklu,

$$s_j = \mathrm{fl}(s_{j-1} + x_j y_j).$$

Podle pravidla (3.4) dostáváme

$$s_j = [s_{j-1} + x_j y_j (1 + \varepsilon_j)](1 + \delta_j),$$

kde $|\varepsilon_j| \leq \mathbf{u}$, $|\delta_j| \leq \mathbf{u}$. Indukcí plyne

$$s_n = \sum_{j=1}^{n} x_j y_j (1 + \varepsilon_j) \prod_{i=j}^{n} (1 + \delta_i),$$

přičemž $\delta_1 = 0$ a kde $|\varepsilon_j| \leq \mathbf{u}$, $|\delta_j| \leq \mathbf{u}$. Definujme

$$\mu_j \equiv (1 + \varepsilon_j) \prod_{i=j}^{n} (1 + \delta_i) - 1.$$

Potom platí

$$s_n = \sum_{j=1}^{n} x_j y_j (1 + \mu_j). \tag{1.5}$$

Ve (3.5) jsme vyjádřili vypočtený výsledek jako přesný výsledek skalárního součinu s perturbovanými (modifikovanými) daty. Konkrétněji,

$$s_n = \widehat{x}^T \widehat{y},$$

kde $\widehat{x} = x$ a $\widehat{y}$ má prvky $\widehat{y}_j = y_j(1 + \mu_j)$. Toto je příklad zpětné analýzy chyb,

Vypočtený výsledek = přesný výsledek na perturbovaných datech.

Úkolem zpětné analýzy chyb je nyní *odhadnout* velikosti perturbací $|\mu_j|$ a vyjádřit relativní vzdálenosti přesných a perturbovaných dat.

Dále lze ukázat, že

$$s_n = x^T y + z, \qquad z = \sum_{j=1}^{n} x_j y_j \mu_j. \tag{1.6}$$

V (3.6) jsme vyjádřili vypočtený výsledek jako přesný výsledek plus nějaká chyba, která závisí na počátečních datech a zaokrouhlovacích chybách. Toto je příklad přímé analýzy chyb,

VYPOČTENÝ VÝSLEDEK = PŘESNÝ VÝSLEDEK + CHYBA.

Úkolem přímé analýzy chyb je odhadovat velikost chyby $|z|$. Přímá a zpětná analýza chyb reprezentují dva pohledy, jak interpretovat chyby při výpočtech.

Odhadněme nyní velikosti chyb. Bez důkazu použijeme následující lemma, které lze nalézt např. v knize N. Highama [2].

Lemma. *Nechť* $|\alpha_i| \le \mathbf{u}$, $i = 1, \ldots, n$ *a nechť* $n\,\mathbf{u} < 0.01$. *Potom platí*

$$\prod_{i=1}^{n} (1 + \alpha_i) = 1 + \beta_n,$$

kde $\beta_n \le 1.01\, n\, \mathbf{u}$.

Použitím tohoto lemmatu lze ukázat, že platí

$$|\mu_j| \le 1.01\, n\, \mathbf{u}$$

a

$$|\mathrm{fl}(x^T y) - x^T y| \le 1.01\, n\, \mathbf{u}\, |x|^T |y|.$$

Odtud již lehce plyne, že platí

$$\mathrm{fl}(x^T y) = \widehat{x}^T \widehat{y}, \tag{1.7}$$

kde

$$\frac{\|x - \widehat{x}\|}{\|x\|} = 0 \qquad \text{a} \qquad \frac{\|y - \widehat{y}\|}{\|y\|} \le 1.01\, n\, \mathbf{u}\,. \tag{1.8}$$

Zpětná stabilita algoritmu

Pomocí analýzy zaokrouhlovacích chyb se popisují a odhadují velikosti zaokrouhlovacích vznikajících při výpočtu algoritmu. Zaokrouhlovací chyby přitom můžeme interpretovat různým způsobem, jak jsme viděli na příkladu přímé a zpětné analýzy chyb. Přímá analýza chyb nám obvykle dává odpověď na otázku jak je algoritmus přesný v konečné aritmetice. Na základě zpětné analýzy chyb dospějeme často k poznání, že spočtené řešení úlohy lze interpretovat jako přesné řešení úlohy pro modifikovaná (perturbovaná) počáteční data, která jsou v jistém smyslu blízká původním počátečním datům. Algoritmus s touto vlastností budeme nazývat *zpětně stabilní*.

Definice. *Algoritmus $f(x)$ je* zpětně stabilní*, pokud platí*

$$\mathrm{fl}(f(x)) = f(\widehat{x}) \qquad \textit{pro nějaké } \widehat{x} \textit{ splňující} \qquad \frac{\|x - \widehat{x}\|}{\|x\|} = \mathcal{O}(\mathbf{u}),$$

Jinými slovy, algoritmus je zpětně stabilní, jestliže se chyby výpočtu způsobené zaokrouhlováním v průběhu algoritmu promítnou do malých změn vstupních dat. Na základě vztahů (3.7) a (3.8) vidíme, že výpočet skalárního součinu je zpětně stabilní.

Přesnost zpětně stabilního algoritmu

Algoritmus f můžeme vidět jako popis, jak řešit daný problém p (představme si pod problémem např. systém lineárních rovnic a pod algoritmem, který daný problém řeší, např. Gaussovu eliminaci). Problém p lze přitom vidět jako zobrazení, které vstupním datům z normovaného vektorového prostoru X přiřazuje řešení z normovaného vektorového prostoru Y, $p : X \to Y$. Zobrazení p je obvykle nelineární (dokonce i v lineární algebře), ale bývá alespoň spojité. *Dobře podmíněný* problém je takový, že malé změny vstupních dat x vedou na malé změny v $p(x)$. *Špatně podmíněný* problém je takový, že malé změny vstupních dat x vedou na velké změny v $p(x)$.

Lze ukázat, že pokud je algoritmus f řešící problém p zpětně stabilní a má-li problém $p(x)$ podmíněnost $\kappa(x)$, pak lze o přesnosti algoritmu v konečné aritmetice říci následující

$$\frac{\|\mathrm{fl}(f(x)) - f(x)\|}{\|f(x)\|} \leq \kappa(x)\mathcal{O}(\mathbf{u}). \tag{1.9}$$

Literatura

[1] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[2] N. J. Higham, *Accuracy and stability of numerical algorithms*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., 2002.

[3] L. N. Trefethen and D. Bau, III, *Numerical linear algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.