

Numerická stabilita algoritmů

Numerická stabilita algoritmů

Pravidla v konečné aritmetice

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

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

kde $\circ$ zastupuje operace $+$, $-$, $*$, $/$ a $\text{fl}(\cdot)$ označuje, že výpočet byl proveden v konečné aritmetice počítače.

Numerická stabilita algoritmů

Přímá a zpětná analýza chyb

- Uvažujme **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** 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.

Numerická stabilita algoritmů

Přesnost výpočtu, absolutní a relativní chyba

- Nechť $f(x)$ reprezentuje výsledek spočtený přesným algoritmem a $\text{fl}(f(x))$ výsledek spočtený algoritmem v konečné aritmetice (x reprezentuje vstupní data).
- Přesnost výpočtu můžeme měřit v **absolutním** smyslu pomocí $\|f(x) - \text{fl}(f(x))\|$, kde $\|\cdot\|$ je vhodná norma, nebo v **relativním** smyslu pomocí

$$\frac{\|f(x) - \text{fl}(f(x))\|}{\|f(x)\|}.$$

- V této přednášce budeme k měření přesnosti algoritmu používat **relativní chybu**.
- Relativní chyba výpočtu **v nejlepším případě** úměrná strojové přesnosti

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

Numerická stabilita algoritmů

Význam $\mathcal{O}(\mathbf{u})$

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

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

- K nezávisí na datech, může záviset na dimenzi úlohy.
- Závislost K na dimenzi problému je obvykle lineární, kvadratická či kubická v nejhorším případě.
- Pokud je závislost K na n například exponenciální (např. $K = 2^n$) $\rightarrow$ **numerická nestabilita** s rostoucím n .

Numerická stabilita algoritmů

Analýza chyb

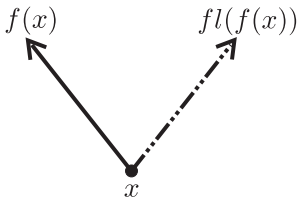
- **Smyslem analýzy chyb** je zjistit, jaké (největší) chyby se algoritmus může při výpočtu v konečné aritmetice dopustit.
- To, jak algoritmus v konečné aritmetice chybuje lze měřit a **analyzovat dvěma** základními **způsoby**.
 - **Přímá analýza chyb**. Postupujeme podle algoritmu a snažíme se popsat šíření elementárních zaokrouhlovacích chyb.
Efektivní přímý odhad chyb je však **možný jen zřídka**.
 - **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.

Cíl: interpretovat zaokrouhlovací chyby vzniklé při výpočtu pomocí změn vstupních dat.

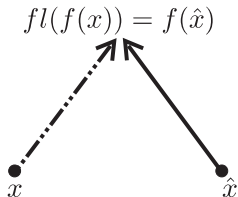
Numerická stabilita algoritmů

Analýza chyb

Přímá analýza
chyb



Zpětná analýza
chyb



Příklad

Přímá a zpětná analýza chyb při výpočtu skalárního součinu

Pokud máme posloupnost operací jako např.

$$s = x^T y, \quad x, y \in \mathbf{F}^n,$$

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

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

Analýza chyb při výpočtu skalárního součinu

Zpětná analýza chyb I

- $s_0 = 0$, nechť s_j je hodnota s na konci j -tého cyklu,

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

- Podle pravidla o chybě vzniklé při jedné operaci 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

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

Analýza chyb při výpočtu skalárního součinu

Zpětná analýza chyb II

- Potom platí

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

- Vypočtený výsledek jsme vyjádřili jako přesný výsledek skalárního součinu s perturbovanými (modifikovanými) daty,

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

kde $\hat{x} = x$ a $\hat{y}$ má prvky $\hat{y}_j = y_j(1 + \mu_j)$.

- Toto je příklad zpětné analýzy chyb,

Vypočtený výsledek = přesný výsledek na perturbovaných datech.

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

Analýza chyb při výpočtu skalárního součinu

Přímá analýza chyb

- Dále lze ukázat, že

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

- Přímá analýza chyb: vyjádřili jsme

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.

Odhady chyb při výpočtu skalárního součinu

- Nechť $|\alpha_i| \leq \mathbf{u}$, $i = 1, \dots, n$ a necht' $n \mathbf{u} < 0.01$. Potom platí

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

kde $\beta_n \leq 1.01 n \mathbf{u}$ ([Lemma \(Higham 2002\)](#)).

- Použitím tohoto lemmatu lze ukázat, že platí

$$\begin{aligned} |\mu_j| &\leq 1.01 n \mathbf{u} \\ |\text{fl}(x^T y) - x^T y| &\leq 1.01 n \mathbf{u} |x|^T |y|. \end{aligned}$$

- Odtud již lehce plyne, že platí

$$\text{fl}(x^T y) = \hat{x}^T \hat{y},$$

kde

$$\frac{\|x - \hat{x}\|}{\|x\|} = 0 \quad \text{a} \quad \frac{\|y - \hat{y}\|}{\|y\|} \leq 1.01 n \mathbf{u}.$$

Zpětná stabilita algoritmu

- **Přímá analýza** chyb nám obvykle dává odpověď na otázku **jak přesný** je algoritmus 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 blízka původním počátečním datům**.
- Definice: Algoritmus $f(x)$ je **zpětně stabilní**, pokud platí

$$\text{fl}(f(x)) = f(\hat{x}) \quad \text{pro nějaké } \hat{x} \text{ splňující } \frac{\|x - \hat{x}\|}{\|x\|} = \mathcal{O}(\mathbf{u}),$$

- Z předchozího, výpočet skalárního součinu je zpětně stabilní.

Přesnost zpětně stabilního algoritmu

- **Algoritmus** $f \rightarrow$ popis, jak řešit daný **problém** p .
- **Problém** p lze přitom vidět **jako zobrazení**, které vstupním datům z normovaného vektorového prostoru X přiřazuje řešení z normovaného vektorového prostoru Y , $p : X \rightarrow Y$.
- p je obvykle **nelineární**, ale bývá alespoň spojitě.
- **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

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