NUMERICKÉ METÓDY LINEÁRNEJ ALGEBRY

02. Podmienenosť problému a stabilita algoritmu

Ing. Marek Macák, PhD.

Konzultácie: podľa potreby/dohody online

19. Februar 2024

Podmienenosť problému

- Problém, ktorý treba vyriešiť, možno modelovať ako vektorovú funkciu f : X $\rightarrow$ Y z normovaného vektorového priestoru vstupných dát X do normovaného vektorového priestoru riešení Y. Obvykle je f nelineárna funkcia, ale je často aspoň spojitá.
- Skúmame chovanie f v danom dátovom vektore $x \in X$ vzhľadom na malé zmeny (perturbácie) bodu x. Keď urobím malú zmenu v x, ako sa táto perturbácia prenesie na riešenie $f(x)$ v Y?
  Dobre podmienený problém má tú vlastnosť, že všetky malé perturbácie bodu x vedú iba k malým zmenám $f(x)$.
  Zle podmienený problém: niektoré malé perturbácie bodu x vedú k veľkým zmenám $f(x)$.

Absolútne a relatívne číslo podmienenosti

- Nech δx je malá perturbácia x a $\delta f = f(x + \delta x) - f(x)$.
- Absolútne číslo podmienenosti $\bar{\kappa} = \bar{\kappa}(x)$ problému f v bode x je definované ako:

$$\bar{\kappa} = \sup_{\delta x} \frac{||\delta f||}{||\delta x||} \quad \mathit{pri} \quad \delta x \to 0. \tag{1}$$

- Relatívne číslo podmienenosti $\kappa = \kappa(x)$ problému f v bode x je definované ako:

$$\kappa = \sup_{\delta x} \left(\frac{||\delta f||}{||f(x)||} \Big/ \frac{||\delta x|}{||x||} \right) \quad \mathit{pri} \quad \delta x \to 0. \tag{2}$$

Absolútne a relatívne číslo podmienenosti

- Ak je f diferencovateľná, definujme Jakobián $J(x)$ funkcie f v bode x ako maticu, ktorej prvok (i, j) je parciálna derivácia $\delta f_i / \delta x_j$ vypočítaná v bode x. Potom:

$$\bar{\kappa} = ||J(x)||, \tag{3}$$

$$\kappa = \frac{||J(x)|}{||f||/||x||}. \tag{4}$$

  kde maticová norma $||J(x)||$ je indukovaná vektorovými normami na X a Y.

- $\kappa(rel.)$ je vhodnejšie ako $\bar{\kappa}(abs.)$, pretože počítanie v pohyblivej rádovej čiarke vnáša do výpočtu relatívne chyby (nie absolútne).
  Dobre podmienená úloha: $\kappa = 1, 10, 10^2$. Zle podmienená úloha: $\kappa = 10^6, 10^{12}$.

Problém versus algoritmus

- Zopakujme, že matematický problém je zobrazenie $f : X \to Y$ z vektorového priestoru dát X do vektorového priestoru riešení Y.
- Algoritmus na riešenie f možno chápať ako iné zobrazenie $\widetilde{f} : X \to Y$ medzi tými istými vektorovými priestormi.
- Spresnenie definície algoritmu: Nech je pevne daný problém f , počitač s floating-point aritmetikou podľa IEEE FPS (1985), algoritmus pre riešenie problému f a implementácia tohto algoritmu vo forme programu na danom počítači. Zoberme vstupné dáta (vektor) $x \in X$, vložme ich do počítača ako vstup pre program (t.j. v tom momente ich zaokrúhlime: $fl(x_i) = x_i(1 + \epsilon_i), |\epsilon_i| \leq \mu$). Potom vykonajme program v počítači. Výsledkom je množina vypočítaných floating-point čísiel z Y, ktorú označíme $\widetilde{f}(x)$.

Presnosť algoritmu

- Na výstup algoritmu $f(x)$ vplývajú minimálne zaokrúhľovacie chyby počas behu programu, ktorý implementuje daný algoritmus. Dokonca na dvoch rôznych počítačoch a na rovnakých vstupných dátach môže ten istý algoritmus (program) dať rôzne výsledky.
- Dobrý algoritmus pre problém f by mal dať 'presný' výsledok.
  Absolútna chyba výpočtu: $||\widetilde{f}(x) - f(x)||$ by mala byť malá pre každé $x \in X$.
  Relatívna chyba výpočtu: $\frac{||\widetilde{f}(x)-f(x)||}{||f(x)||}$ by mala byť malá pre každé $x \in X$,
  t.j. $= O(\mu)$ ('rádovo' μ), kde μ je jednotka zaokrúhľovania.
- Pre zle podmienené problémy je táto požiadavka príliš silná. Už samotné zaokrúhlenie vstupných dát môže podstatne zmeniť výsledok.

Zmysel symbolu $O(\mu)$

- Označenie
$$\phi(t) = O(\psi(t)) \tag{5}$$
znamená, že existuje konštanta $C > 0$ tak, že pre t dostatočne blízko ku stanovenej limite (napr. $t \to 0$ alebo $t \to \infty$) platí: $\phi(t) \leq C(\psi(t))$.
- Takže v NLA tvrdenie: $||\widetilde{x}|| = O(\mu)$ znamená:
$||\widetilde{x}||$ reprezentuje normu vypočítaného výsledku pri použití algoritmu $\widetilde{f}$ na problém f, pričom výsledok závisí od vstupných dát $x \in X$ a od μ.

Stabilita algoritmu

- Namiesto požiadavky na vysokú relatívnu presnosť výpočtu nastupuje požiadavka na stabilitu algoritmu.
- Algoritmus je stabilný, ak pre každý vstup $x \in X$ platí

$$\frac{||\widetilde{f}(x) - f(\widetilde{x})||}{||f(\widetilde{x})||} = O(\mu) \tag{6}$$

pre nejake $\widetilde{x}$ relatívne blizke k x:

$$\frac{||\widetilde{x} - x||}{||x||} = O(\mu) \tag{7}$$

- Slovne: Stabilný algoritmus dáva takmer správny výsledok pre takmer správne vstupné dáta.

Spätná stabilita algoritmu

- Mnohé algoritmy v NLA spĺňajú podmienku, ktorá je silnejšia a jednoduchšia ako stabilita.
- Algoritmus $\widetilde{f}$ pre problém f je spätne stabilný ('backward stable'), ak pre každý vstup $x \in X$ platí

$$\widetilde{f}(x) = f(\widetilde{x}) \tag{8}$$

pre nejake $\widetilde{x}$ relatívne blizke k x:

$$\frac{||\widetilde{x} - x||}{||x||} = O(\mu) \tag{9}$$

- Slovne: Spätne stabilný algoritmus dáva presne správny výsledok pre takmer správne vstupné dáta.

Stabilita algoritmu - analýza chyb

- Zmyslom analýzy chýb je zistiť, akéj (najväčšej) chyby sa algoritmus môže pri výpočte v konečnej aritmetike dopustiť.
- To, ako algoritmus v konečnej aritmetike chybuje možno merať a analyzovať dvoma základnými spôsobmi:
    - **Priama analýza chýb**. Postupujeme podľa algoritmu a snažíme sa popísať šírenie elementárnych zaokrúhľovacích chýb. Efektívny priamy odhad chyby je však možný len zriedka.
    - **Spätná analýza chýb**. Hľadáme modifikované vstupné dáta úlohy tak, aby približné riešenie spočítané algoritmom v konečnej aritmetike počítača bolo presným riešením tej istej úlohy, avšak s modifikovanými vstupnými dátami.

Cieľ: interpretovať zaokrúhľovacie chyby vzniknuté pri výpočte pomocou zmien vstupných dát.

Stabilita algoritmu - analýza chyb

- Príklad: Ak máme postupnosť operácií ako napr.

$$s = x^T y, \quad x, y \in \mathcal{F}^n \tag{10}$$

kde $\mathcal{F}^n$ označuje vektory dĺžky n, ktorých prvky sú čísla v pohyblivej rádovej čiarke ($x_i \in F_t, y_i \in F_t, i = 1, ..., n$), potom budeme výsledok spočítaný algoritmom

```
s=0
for j = 1, 2, ..., n
   s = s + x_j y_j
   end
```

v konečnej aritmetike počítača zapisovať v tvare

$$s = fl(x^T y). \tag{11}$$

Presnosť versus spätná stabilita

- Je spätne stabilný algoritmus aj presný? Závisí to od čísla podmienenosti $\kappa(x)$ problému.
- Veta: Nech je na riešenie problému $f : X \to Y$ s číslom podmienenosti $\kappa(x)$ použitý spätne stabilný algoritmus $\widetilde{f} : X \to Y$, ktorý je implementovaný na počítači spĺňajúcom IEEE Floating Point Standard (1985). Potom relatívna chyba výpočtu je:
$$\frac{||\widetilde{x} - x||}{||x||} = O(\kappa(x)\mu) \tag{12}$$
- Takže ak je $\kappa(x)$ malé, potom výsledok výpočtu bude presný v relatívnom zmysle. Ak je $\kappa(x)$ veľké, potom relatívna presnosť výsledku môže byť nízka (t.j. nemáme zaručenú vysokú relatívnu presnosť).