Poglavje 2
Sistemi linearnih enačb

Tretji sklop izročkov

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

26. oktober 2020

Analiza zaokrožitvenih napak pri reševanju $Ax = b$

Sistem $Ax = b$ smo rešili numerično in dobili približek $\hat{x}$. Kako natančen je ta približek?

Pišimo $\hat{x} = x + \Delta x$. V resnici velja

$$A(x + \Delta x) = b + \Delta b.$$

Radi bi ocenili velikostni razred Δx v primerjavi z x. V ta namen potrebujemo nekaj osnov vektorskih in matričnih norm.

Vektorska norma je preslikava $\|\cdot\| : \mathbb{C}^n \to \mathbb{R}$, ki zadošča:

1. Pozitivna definitnost: $\|x\| \geqslant 0$ za vsak $x \in \mathbb{C}^n$ in $\|x\| = 0 \Leftrightarrow x = 0$.
2. Homogenost :$\|\alpha x\| = |\alpha| \, \|x\|$ za vsaka $\alpha \in \mathbb{C}$ in $x \in \mathbb{C}^n$
3. Trikotniška neenakost: $\|x + y\| \leqslant \|x\| + \|y\|$ za vsaka $x, y \in \mathbb{C}^n$.

Matrična norma je preslikava $\|\cdot\| : \mathbb{C}^{n \times n} \to \mathbb{R}$, ki zadošča:

1. Pozitivna definitnost: $\|A\| \geqslant 0$ za vsak $A \in \mathbb{C}^{n \times n}$ in $\|A\| = 0 \Leftrightarrow A = 0$.
2. Homogenost :$\|\alpha A\| = |\alpha| \, \|A\|$ za vsaka $\alpha \in \mathbb{C}$ in $A \in \mathbb{C}^{n \times n}$.
3. Trikotniška neenakost: $\|A + B\| \leqslant \|A\| + \|B\|$ za vsaka $A, B \in \mathbb{C}^{n \times n}$.
4. Submultiplikativnost: $\|AB\| \leqslant \|A\| \, \|B\|$ za vsaka $A, B \in \mathbb{C}^{n \times n}$.

Vektorske in matrične norme - primeri

Primer

Nekaj vektorskih norm:

1. $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ *(1-norma).*
2. $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$ *(2-norma).*
3. $\|x\|_\infty = \max_{i=1,\dots,n} |x_i|$ *(∞-norma).*

Vsaka vektorska norma inducira matrično normo, ki jo imenujemo **operatorska norma**:

$$\|A\| := \max_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\| .$$

Analiza zaokrožitvenih napak pri reševanju $Ax = b$

Naj bo $\|\cdot\|$ matrična norma, da velja $\|I\| = 1$. Naj bo A nesingularna matrika,

$$Ax = b \quad \text{in} \quad (A + \Delta A)(x + \Delta x) = b + \Delta b.$$

Preoblikujemo:

$$(A + \Delta A)\Delta x = \Delta b - \Delta A x.$$

Izpostavimo A:

$$A(I + A^{-1}\Delta A)\Delta x = \Delta b - \Delta A x.$$

Predpostavimo še, da je $\left\|A^{-1}\right\| \|\Delta A\| < 1$.

Potem je

$$(I + A^{-1}\Delta A)^{-1} = I - A^{-1}\Delta A + (A^{-1}\Delta A)^2 - (A^{-1}\Delta A)^3 + \cdots$$

obrnljiva matrika in velja

$$\left\|(I + A^{-1}\Delta A)^{-1}\right\| \leqslant \frac{1}{1 - \left\|A^{-1}\Delta A\right\|}.$$

Sledi

$$\Delta x = (I + A^{-1}\Delta A)^{-1} A^{-1} (\Delta b - \Delta A x),$$

$$\|\Delta x\| \leqslant \frac{1}{1 - \|A^{-1}\Delta A\|} \left\|A^{-1}\right\| (\|\Delta b\| + \|\Delta A\| \, \|x\|).$$

Analiza zaokrožitvenih napak pri reševanju $Ax = b$

Ocenimo:

$$\begin{aligned}
\frac{\|\Delta x\|}{\|x\|} &\leqslant \frac{1}{1-\|A^{-1}\Delta A\|}\left\|A^{-1}\right\|\left(\frac{\|\Delta b\|}{\|x\|}+\|\Delta A\|\right)\\
&= \frac{\left\|A^{-1}\right\|\|A\|}{1-\|A^{-1}\|\,\|\Delta A\|}\left(\frac{\|\Delta b\|}{\|A\|\,\|x\|}+\frac{\|\Delta A\|}{\|A\|}\right)\\
&\leqslant \frac{\left\|A^{-1}\right\|\|A\|}{1-\|A^{-1}\|\,\|A\|\,\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta b\|}{\|b\|}+\frac{\|\Delta A\|}{\|A\|}\right).
\end{aligned}$$

Izrazu

$$\kappa(A) = \|A\|\left\|A^{-1}\right\|$$

pravimo **občutljivost matrike** A.

Izrek

Naj bo $\|\cdot\|$ matrična norma, da velja $\|I\| = 1$. Naj bo A nesingularna matrika,

$$Ax = b \quad \text{in} \quad (A+\Delta A)(x+\Delta x) = b+\Delta b.$$

Naj bo še $\left\|A^{-1}\right\|\|\Delta A\| < 1$. Potem velja

$$\frac{\|\Delta x\|}{\|x\|} \leqslant \frac{\kappa(A)}{1-\kappa(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta b\|}{\|b\|}+\frac{\|\Delta A\|}{\|A\|}\right).$$

Primer

Za $A_1 = \begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ *in* $A_2 = \begin{pmatrix}1.00 & 0.99\\ 0.99 & 0.98\end{pmatrix}$ *velja* $\kappa_2(A_1) = 1$ *in* $\kappa_2(A_2) = 3.9\cdot 10^4$.

Iterativne metode za reševanje linearnih sistemov

Doslej smo iskali **točne rešitve** x^* sistema

$$Ax = b.$$

Odslej nas bodo zanimali samo **približki** $\hat{x}$ točnih rešitev x^*.

Naprej si bomo izbrali $\epsilon > 0$ in iskali $\hat{x}$, ki zadošča $\|\hat{x} - x^*\| \leqslant \epsilon$.

Prednosti iterativnih metod pred direktnimi:

- Če je matrika A velika in ima veliko ničel, je bolje uporabiti iterativne metode.
- Ko je rezultat znotraj vnaprej predpisane natančnosti, lahko končamo računanje. Pri direktnih metodah tega vpliva nimamo.

Recimo, da ugibamo, kaj bi lahko bila prava rešitev x sistema $Ax = b$:

$$x^{(0)} \approx x$$

Kako izboljšati $x^{(0)}$? Idealno bi prišteli pravi razliko:

$$x^{(1)} = x^{(0)} + (x^* - x^{(0)}),$$

kar lahko drugače zapišemo kot

$$\begin{aligned} x^{(1)} &= x^{(0)} + (x^* - x^{(0)}) \\ &= x^{(0)} + (A^{-1}b - x^{(0)}) \\ &= x^{(0)} + A^{-1}(b - Ax^{(0)}) \\ &= x^{(0)} + A^{-1}r^{(0)} \end{aligned}$$

Iteracija

Metoda

$$x^{(1)} = x^{(0)} + A^{-1} r^{(0)}$$

pa ni smiselna, ker bi morali izračunati A^{-1}.

Kaj pa, če bi znali aproksimariti A^{-1}?

Recimo, da je približek $Q^{-1} \approx A^{-1}$ poceni za izračunati. Potem izračunamo

$$x^{(1)} = x^{(0)} + Q^{-1} r^{(0)}.$$

Nadaljujemo

$$x^{(k)} = x^{(k-1)} + Q^{-1} r^{(k-1)}$$

Preoblikujmo

$$x^{(k)} = x^{(k-1)} + Q^{-1}(b - Ax^{(k-1)}),$$

v obliko

$$\begin{aligned} Qx^{(k)} &= Qx^{(k-1)} + (b - Ax^{(k-1)}) \\ &= (Q - A)x^{(k-1)} + b. \end{aligned}$$

Ali drugače

$$\boxed{x^{(k)} = Q^{-1}(Q - A)x^{(k-1)} + Q^{-1} b.}$$

Jabobijeva iteracija

Jacobijeva iteracija aproksimira A z $\boxed{Q = \text{diagonala od } A}$.

```
x = x^(0)

Q = D

for k = 1 to k_max
  r = b − Ax

  if ||x^(nov) − x|| / ||x|| ≤ tol, stop
 else
    x = x^(nov)
    x^(nov) = x + Q^(−1) r
end

x = x^(nov)
```

Gauss-Seidlova iteracija

Gauss-Seidlova iteracija aproksimira A z $\boxed{Q = \text{spodnji trikotnik od } A}$.

```
x = x^(0)

Q = D + L

for k = 1 to k_max
  r = b − Ax

  if ||x^(nov) − x|| / ||x|| ≤ tol, stop
 else
     x = x^(nov)
     x^(nov) = x + Q^(−1) r
end

x = x^(nov)
```

Zakaj D in $D + L$?

Še enkrat poglejmo iteracijo

$$x^{(k)} = x^{(k-1)} + Q^{-1} r^{(k-1)},$$

in napako

$$x - x^{(k)} = x - x^{(k-1)} - Q^{-1} r^{(k-1)}.$$

Dobimo

$$e^{(k)} = e^{(k-1)} - Q^{-1} A e^{(k-1)},$$

oz.

$$e^{(k)} = (I - Q^{-1} A) e^{(k-1)},$$

oz.

$$e^{(k)} = (I - Q^{-1} A)^k e^{(0)}.$$

Zakaj D in $D+L$?

Radi bi, da zaporedje

$$e^{(k)} = (I - Q^{-1}A)^k e^{(0)}$$

konvergira. Kdaj zaporedje $a_k = c^k$ konvergira? .....če velja $|c| < 1$

Podobno, naša iteracija konvergira

$$\begin{aligned}\|e^{(k)}\| &= \|(I - Q^{-1}A)^k e^{(0)}\| \\ &\leqslant \|I - Q^{-1}A\|^k \|e^{(0)}\|,\end{aligned}$$

če je $\|I - Q^{-1}A\| < 1$.

Trditev

Iteracija

$$x^{(k)} = x^{(k-1)} + Q^{-1}(b - Ax^{k-1}) = (I - Q^{-1}A)x^{(k-1)} + Q^{-1}b$$

konvergira za poljuben začetni vektor $x^{(0)}$ natanko tedaj, ko ima matrika $I - Q^{-1}A$ vse lastne vrednosti po absolutni vrednosti manjše od 1.

Zakaj Jacobi in Gauss-Seidel delujeta?

Jacobi, Gauss-Seidel (zadostni) konvergenčni izrek

Če je A diagonalno dominantna, potem Jacobijeva in Gauss-Seidlova metoda konvergirata za kateri koli začetni približek $x^{(0)}$.

Definicija: Diagonalna dominanca

Matrika je diagonalno dominantna, če je

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|$$

za vsak i.

Jacobijev algoritem

Algoritem zgoraj uporabi naslednjo matrično reprezentacijo:

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

Diagonala nima nič skupnega z $L + U$, zato lahko zapišemo

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

Torej vsak preskok (iz $k-1$ na k) potrebuje $\mathcal{O}(n)$ operacij za vsak element novega vektorja.

Če je za vsak vrstico i, $a_{ij} = 0$ za vse razen m vrednosti j, vsak preskok potrebuje $\mathcal{O}(mn)$ operacij.

Gauss-Seidlov algoritem

Algoritem zgoraj uporabi naslednjo matrično reprezentacijo:

$$x^{(k)} = -(D+L)^{-1}Ux^{(k-1)} + (D+L)^{-1}b$$

Po komponentah

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

Torej vsak preskok (iz $k-1$ na k) potrebuje $\mathcal{O}(n)$ operacij na vsak element novega vektorja.

Če je za vsak vrstico i, $a_{ij} = 0$ za vse razen m vrednost j, vsak preskok potrebuje $\mathcal{O}(mn)$ operacij.

Razlika v primerjavi z Jacobijevo metodo je ta, da so popravki shranjeni v novem vektorju. Tako pridobimo precej prihranka v spominu.