Predavanja 5

Peti sklop izročkov

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

8. november 2021

Občutljivost sistema $Ax = b$

Vprašanje. Denimo, da smo sistem $Ax = b$ z neko metodo (npr. LU razcepom) rešili numerično in dobili približek $\hat{x}$. Kako dober je ta približek?

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

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

Če vključimo še numerično napako v A, velja

$$(A + \Delta A)(x + \Delta x) = b + \Delta b. \tag{1}$$

Radi bi ocenili velikostni razred Δx v primerjavi z x, tj. $\frac{\|\Delta x\|}{\|x\|}$, kjer je $\|\cdot\|$ neka vektorska norma.

Izberimo vektorsko normo $\|\cdot\|$ in njej pripadajočo matrično normo, tj. $\|A\| = \max_{\|x\|=1} \|Ax\|$. Definirajmo občutljivost oz. pogojenostno število obrnljive matrike A v normi $\|\cdot\|$:

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

Izrek

1. *Privzemimo, da je* $\Delta A = 0$. *Potem velja:*

$$\frac{\|\Delta x\|}{\|x\|} \leqslant \kappa(A) \frac{\|\Delta b\|}{\|b\|}.$$

2. *Naj bo* $\Delta A \neq 0$. *Potem velja:*

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

pri čemer smo potrebovali predpostavki, da za identično matriko I *velja* $\|I\| = 1$ *in matrika* A *zadošča* $\left\|A^{-1}\right\| \|\Delta A\| < 1$.

Dokaz prvega dela izreka. Preoblikujmo (1) v

$$A\Delta x = \Delta b.$$

Po predpostavki je A obrnljiva, zato velja:

$$\Delta x = A^{-1}\Delta b.$$

Torej velja ocena

$$\|\Delta x\| \leqslant \left\|A^{-1}\right\| \|\Delta b\|. \tag{2}$$

Delimo (2) z $\|x\|$ in dobimo

$$\frac{\|\Delta x\|}{\|x\|} \leqslant \frac{\left\|A^{-1}\right\| \|\Delta b\|}{\|x\|}. \tag{3}$$

Ker je $Ax = b$, sledi $\|b\| = \|Ax\| \leqslant \|A\|\|x\|$. Uporabimo to v (3) in dobimo

$$\frac{\|\Delta x\|}{\|x\|} \leqslant \frac{\|A\| \left\|A^{-1}\right\| \|\Delta b\|}{\|x\|} = \kappa(A)\frac{\|\Delta b\|}{\|b\|}. \tag{4}$$

Dokaz drugega dela izreka. Neobvezen, za radovedne. Preoblikujmo (1) v

$$(A + \Delta A)\Delta x = \Delta b - \Delta A x. \tag{5}$$

Po predpostavki je A obrnljiva, zato lahko izpostavimo A:

$$A(I + A^{-1}\Delta A)\Delta x = \Delta b - \Delta A x. \tag{6}$$

Spomnimo se formule za vsoto geometrijske vrste:

$$(1 + q)^{-1} = \frac{1}{1+q} = 1 - q + q^2 - q^3 + \ldots, \qquad |q| < 1. \tag{7}$$

Sedaj lahko oponašamo (7) za matrike in dobimo

$$(I + A^{-1}\Delta A)^{-1} = I - A^{-1}\Delta A + (A^{-1}\Delta A)^2 - (A^{-1}\Delta A)^3 + \ldots \tag{8}$$

Iz (8) sledi (z nekaj utemeljevanja - mi lahko privzamemo)

$$\left\|(I + A^{-1}\Delta A)^{-1}\right\| \leqslant \frac{1}{1 - \|A^{-1}\Delta A\|}. \tag{9}$$

Pomnožimo (6) z leve z A^{-1} in nato z $(I + A^{-1}\Delta A)^{-1}$

$$\Delta x = (I + A^{-1}\Delta A)^{-1} A^{-1}(\Delta b - \Delta A x), \tag{10}$$

Upoštevamo še (9) v (10) in dobimo

$$\|\Delta x\| \leqslant \frac{1}{1 - \|A^{-1}\Delta A\|} \left\|A^{-1}\right\| (\|\Delta b\| + \|\Delta A\| \, \|x\|). \qquad (11)$$

Delimo (11) z $\|x\|$ in preoblikujemo:

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

Primer

Če se spomnimo primera računanja prečišča dveh premic iz prvih predavanj, lahko vidimo, da gre za vprašanje občutljivosti matrik

$$A_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \textit{in} \quad A_2 = \begin{pmatrix} 1.00 & 0.99 \\ 0.99 & 0.98 \end{pmatrix},$$

za kateri velja $\kappa_2(A_1) = 1$ *in* $\kappa_2(A_2) = 3.9 \cdot 10^4$, *kjer* κ_2 *označuje občutljivost v spektralni normi.*

Iterativne metode za reševanje $Ax = b$

Doslej smo iskali točno rešitev x^* sistema

$$Ax = b. \tag{12}$$

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

Naprej si bomo izbrali $\epsilon > 0$ in iskali $\hat{x}$, ki zadošča pogoju

$$\|\hat{x} - x^*\| \leqslant \epsilon.$$

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 sistema (12)

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

Kako izboljšati $x^{(0)}$?

Idealno bi prišteli pravi razliko:

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

kar lahko drugače zapišemo kot

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

Toda ta metoda ni smiselna, saj 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 z $k = 2, 3, \ldots$

$$x^{(k)} = x^{(k-1)} + Q^{-1} \underbrace{(b - Ax^{(k-1)})}_{r^{(k-1)}}. \tag{13}$$

Algoritem iterativnih metod

A je dana $n \times n$ matrika, ki jo aproksimiramo z matriko Q, in $n \times 1$ vektor b.

Izberi zacetni priblizek $x = x^{(0)}$, toleranco dovoljene relativne napake tol in maksimalno stevilo k_{max} korakov iteracije.

$x^{(nov)} = \infty$

for $k = 1$ to k_{max}

  $r = b - Ax$

  if $\frac{\|x^{(nov)} - x\|}{\|x\|} \leqslant tol$, stop

  else

    $x = x^{(nov)}$

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

end

$x = x^{(nov)}$

Jacobijeva iteracija

Aproksimiramo $A = [a_{ij}]_{i,j}$ z diagonalno matriko

$$D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix}.$$

Če pišemo

$$A = S + D + Z, \tag{14}$$

kjer je S strogo spodnjetrikotna matrika in Z strogo zgornjetrikotna matrika, potem (13) postane

$$\begin{aligned} x^{(k)} &= x^{(k-1)} + D^{-1}(b - Sx^{(k-1)} - Dx^{(k-1)} - Zx^{(k-1)}) \\ &= x^{(k-1)} + D^{-1}b - D^{-1}Sx^{(k-1)} - x^{(k-1)} - D^{-1}Zx^{(k-1)} \\ &= D^{-1}(b - Sx^{(k-1)} - Zx^{(k-1)}). \end{aligned} \tag{15}$$

Pišimo

$$x^{(j)} = \begin{pmatrix} x_1^{(j)} & x_2^{(j)} & \dots & x_n^{(j)} \end{pmatrix}^T$$

Po komponentah (15) pomeni

$$\boxed{x_i^{(k)} = \frac{b_i}{a_{ii}} - \sum_{j=1, j\neq i}^{n} \left(\frac{a_{ij}}{a_{ii}}\right) x_j^{(k-1)}}. \qquad (16)$$

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

Če so v vsaki vrstici vsi razen največ m koeficientov a_{ij} neničelnih, potem za vsak korak iteracije potrebujemo $O(mn)$ operacij.

Algoritem in primeri:

https://zalara.github.io/jacobi.m

https://zalara.github.io/testjacobi.m

https://zalara.github.io/testjacobi2.m

Gauss-Seidlova iteracija

Naj bo $A = [a_{ij}]_{i,j} = S + D + Z$ kot v (14). A aproksimiramo s spodnjetrikotno matriko

$$S + D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \cdots & a_{n,n-1} & a_{nn} \end{pmatrix}.$$

Potem (13) postane

$$\begin{aligned} x^{(k)} &= x^{(k-1)} + (D+S)^{-1}(b - (S+D)x^{(k-1)} - Zx^{(k-1)}) \\ &= x^{(k-1)} + (D+S)^{-1}b - x^{(k-1)} - (D+S)^{-1}Zx^{(k-1)} \\ &= (D+S)^{-1}(b - Zx^{(k-1)}), \end{aligned} \tag{17}$$

Algoritem in primeri:

https://zalara.github.io/gaussseidel.m

https://zalara.github.io/testgaussseidel.m

https://zalara.github.io/testgaussseidel2.m

Pomnožimo (17) z leve z $D + S$ in dobimo

$$(D + S)x^{(k)} = b - Zx^{(k-1)}. \tag{18}$$

Odštejemo $Sx^{(k)}$ od obeh strani (18) in dobimo

$$Dx^{(k)} = b - Zx^{(k-1)} - Sx^{(k)}$$

oz.

$$x^{(k)} = D^{-1}(b - Zx^{(k-1)} - Sx^{(k)}). \tag{19}$$

Po komponentah (19) pomeni

$$\boxed{x_i^{(k)} = \frac{b_i}{a_{ii}} - \sum_{j=1, j<i}^{n} \left(\frac{a_{ij}}{a_{ii}}\right) x_j^{(k)} - \sum_{j=1, j>i}^{n} \left(\frac{a_{ij}}{a_{ii}}\right) x_j^{(k-1)}}. \tag{20}$$

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

Če so v vsaki vrstici vsi razen največ m koeficientov a_{ij} neničelnih, potem za vsak korak iteracije potrebujemo $O(mn)$ operacij.

Razlika v primerjavi z Jacobijevo metodo je ta, da so popravki shranjeni v obstoječem vektorju in ne potrebujemo še enega dodatnega vektorja. Tako pridobimo precej prihranka v spominu.

Konvergenca iterativnih metod

Brez dokaza navedimo trditev, ki zagotavlja, kdaj bo Q dobra izbira za aproksimacijo matrike A.

Trditev

Iteracija

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

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

Dokaz. Pišimo $e^{(k)} = x^* - x^{(k)}$ za napako k-tega približka. Iz

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

sledi

$$x^* - x^{(k)} = x^* - x^{(k-1)} - Q^{-1} r^{(k-1)}. \tag{21}$$

(21) prepišemo v

$$e^{(k)} = e^{(k-1)} - Q^{-1} A e^{(k-1)} = (I - Q^{-1} A) e^{(k-1)} \tag{22}$$

Rekurzivno uporabljamo (22) in dobimo

$$e^{(k)} = (I - Q^{-1} A)^2 e^{(k-2)} = (I - Q^{-1} A)^3 e^{(k-3)} = \ldots = (I - Q^{-1} A)^k e^{(0)}.$$

Radi bi, da zaporedje

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

konvergira.

Kdaj zaporedje $a_k = c^k$ konvergira? Odgovor: natanko za $|c| < 1$.

Podobno, naša iteracija konvergira

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

natanko tedaj, ko je $\|I - Q^{-1} A\| < 1$. To pa je res ravno v primeru, ko so vse lastne vrednosti matrike $I - Q^{-1} A$ manjše od 1. □

Kdaj Jacobi in Gauss-Seidel gotovo delujeta?

Matrika je diagonalno dominantna, če za vsak i velja $|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|$.

Izrek

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

Dokaz za primer Jacobijeva iteracije. Kot doslej pišimo $A = S + D + Z$. Velja

$$T_J = I - Q_J^{-1} A = I - D^{-1}(S + D + Z) = D^{-1}(S + Z).$$

Naj bo $v = \begin{pmatrix} v_1 & \dots & v_n \end{pmatrix}^T$ lastni vektor za T_J. i-ta komponenta vektorja $T_J v$ je enaka

$$\frac{1}{a_{ii}} \left(v_1 a_{i1} + \ldots + v_{i-1} a_{i,i-1} + v_{i+1} a_{i,i+1} + \ldots + v_n a_{in} \right).$$

Sledi

$$\| T_J v \|_\infty \leqslant \frac{1}{|a_{ii}|} \cdot \| v \|_\infty \cdot \sum_{j=1, j\neq i}^{n} |a_{ij}| < \| v \|_\infty.$$

Torej so vse lastna vrednosti T_J manjše od 1. □

SOR iteracija

Gauss-Seidlovo iteracijo za reševanje linearnih sistemov $Ax = b$ se da pospešiti z uporabo t.i. *SOR metod*. Korak iteracije je oblike

$$x^{(k)} = T_w x^{(k-1)} + c_w,$$

kjer je $w > 0$ in

$$T_w = (D + wS)^{-1}[(1 - w)D - wZ],$$
$$c_w = w(D + wS)^{-1}b.$$

Algoritem in primeri:

https://zalara.github.io/sor.m

https://zalara.github.io/testsor.m

https://zalara.github.io/testsor2.m