Kako se lotiš: Uvod v numerične metode

Patrik Žnidaršič

Prevedeno 7. junij 2024

1 Predstavljivost števil

Množica predstavljivih števil $P(b, t, L, U)$ vsebuje vsa števila oblike

$$\pm 0, c_1 c_2 \dots c_t \cdot b^e,$$

kjer je $L \leq e \leq U$ in $c_i \in \{0, \dots, b-1\}$. Dovoljena so tudi števila 0, ∞, $-\infty$ ter nan. Števila so lahko normalizirana ali denormalizirana; pri prvem tipu velja $c_1 \neq 0$. Denormalizirana števila so dovoljena le, če se tega števila ne mora zapisati v normalizirani obliki. Običajen `float` je razreda $P(2, 24, -125, 128)$, `double` pa $P(2, 53, -1021, 1023)$.

Najbližje predstavljivo število številu x v neki množici zapišemo s $\mathrm{fl}(x)$. Napako operacij lahko izračunamo kot ABSOLUTNO NAPAKO $d_a = \hat{x} - x$ in RELATIVNO NAPAKO $d_r = d_a/x$. Pri ocenjevanju lahko uporabimo dejstvo

$$\frac{|\mathrm{fl}\, x - x|}{|x|} \leq u = \frac{1}{2} b^{1-t}.$$

Standard IEEE zagotavlja omejeno napako pri osnovnih operacijah,

- $\mathrm{fl}\, x \oplus y = (x \oplus y)(1+\delta)$ za $|\delta| \leq u$ in osnovne operacije $+, -, \cdot, /$,
- $\mathrm{fl}\, \sqrt{x} = \sqrt{x}(1+\delta)$ za $|\delta| \leq u$.

2 Nelinearne enačbe

Če iščemo ničlo funkcije $f(x)$ na nekem intervalu, lahko uporabimo bisekcijo ali katero od iteracijskih metod. V splošnem je problem za večkratne ničle bolj občutljiv, za m-kratno ničlo v splošnem velja

$$|x - \alpha| \leq \left(\frac{m!\, \varepsilon}{\left| f^{(m)}(\alpha) \right|} \right)^{1/m},$$

če je $|f(x)| \leq \varepsilon$.

Pri navadni iteraciji iščemo fiksno točko dane funkcije g pri nekem začetnem približku, s postopkom $x_{r+1} = g(x_r)$. Da bo postopek konvergiral, mora biti g v okolici fiksne točke skrčitev. Dovolj je že, da je $|g'(\alpha)| < 1$ in g zvezno odvedljiva. Če vemo, da je g Lipschitzova s konstanto $L < 1$, lahko ocenimo

$$|x_r - \alpha| \leq L^r \, |x_0 - \alpha|$$

oziroma

$$|x_{r+1} - \alpha| \leq \frac{L}{1-L} \, |x_{r+1} - x_r| \,,$$

kjer je α fiksna točka. Interval konvergence lahko torej določamo s pomočjo odvoda (izračunamo, kje je $|g'| < 1$), vendar nam to ne da vedno celotnega intervala. Drug način je, da direktno dokažemo, kakšna je limita zaporedja $(x_r)_r$. Vizualno si lahko pomagamo s pajčevinastim diagramom.

Pri konvergenci poznamo tudi RED. To je število p, za katerega velja

$$\lim_{r\to\infty} \frac{|x_{r+1} - \alpha|}{|x_r - \alpha|^p} = C$$

za neko konstanto $C > 0$. Zadosten pogoj za red p je, da velja $g'(\alpha) = \cdots = g^{(p-1)}(\alpha) = 0$ ter $g^{(p)} \neq 0$. Višji kot je red, boljša je metoda.

Če iščemo ničlo funkcije f in želimo zagotoviti red 2 (za enostavno ničlo), lahko uporabimo TANGENTNO METODO

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Če je f vsaj dvakrat odvedljiva in α ničla kratnosti m, potem za g iz tangentne metode velja

$$\lim_{x\to\alpha} g'(x) = 1 - \frac{1}{m}.$$

Podobna je SEKANTNA METODA, kjer imamo predpis

$$x_{r+1} = x_r - \frac{f(x_r)(x_r - x_{r-1})}{f(x_r) - f(x_{r-1})}.$$

To pa sedaj ni navadna iteracija.

Če imamo dano neko iteracijo

$$x_r = F(x_{r-1}, x_{r-2}, \ldots, x_{r-k}),$$

in vemo, da postopek konvergira (npr. če dokažemo, da je zaporedje omejeno in monotono), potem lahko limito izračunamo s pomočjo trika

$$\lim_{r\to\infty} x_r = \lim_{r\to\infty} x_{r+i},$$

torej je limita $\alpha = F(\alpha, \alpha, \ldots, \alpha)$.

3 Matrične norme

Definicija. Preslikava $\|\cdot\| : \mathbb{C}^{n\times n} \to \mathbb{R}$ je MATRIČNA NORMA, če velja

- $\|A\| \geq 0$ in $\|A\| = 0 \Leftrightarrow A = 0$,
- $\|\alpha A\| = |\alpha| \, \|A\|$,
- $\|A + B\| \leq \|A\| + \|B\|$,
- $\|AB\| \leq \|A\| \, \|B\|$.

Vsota absolutnih vrednosti vseh elementov matrike $N_1(A)$ in maksimum absolutnih vrednosti vseh elementov matrike $N_\infty(A)$ žal nista normi, 2-norma vektorizirane matrike pa dejansko je;

$$\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2}.$$

Velja $\|A\|_F^2 = \mathrm{sl}(A^H A)$. Poleg tega je matrična norma tudi $\|A\| = nN_\infty(A)$, kjer je n dimenzija matrike. Najbolj uporabne pa so operatorske norme; če je $\|\cdot\|_v$ vektorska norma, je pripadajoča matrična norma

$$\|A\|_m = \max_{x\neq 0} \frac{\|Ax\|_v}{\|x\|_v}.$$

Pri tem je operatorska norma $\|A\|_1$ enaka največji 1-normi stolpca matrike A, $\|A\|_\infty$ pa največji 1-normi vrstice. Najbolj uporabna in najtežje izračunljiva je 2-norma, ki je enaka največji singularni vrednosti matrike (t.j. korenu lastne vrednosti matrike $A^H A$). Za poljubno matrično normo in poljubno lastno vrednost λ matrike A velja $|\lambda| \leq \|A\|$.

Računanje druge norme je snov predmeta NLA, lahko pa jo ocenimo z enim od

$$\begin{gathered}
\frac{1}{\sqrt{n}} \|A\|_F \leq \|A\|_2 \leq \|A\|_F \\
\frac{1}{\sqrt{n}} \|A\|_1 \leq \|A\|_2 \leq \sqrt{n} \, \|A\|_1 \\
\frac{1}{\sqrt{n}} \|A\|_\infty \leq \|A\|_2 \leq \sqrt{n} \, \|A\|_\infty \\
\|A\|_2 \leq \sqrt{\|A\|_1 \, \|A\|_\infty} \\
N_\infty(A) \leq \|A\|_2 \leq nN_\infty(A)
\end{gathered}$$

Za Frobeniusovo in drugo normo dodatno velja $\|A\|_F = \left\|A^H\right\|_F$ oziroma $\|A\|_2 = \left\|A^H\right\|_2$.

4 LU razcep

Če imamo dano nesingularno matriko A, jo lahko razcepimo v produkt $A = LU$, kjer je L spodnje diagonalna z enicami na diagonali, in U zgornje diagonalna matrika. Pri

postopku definiramo matrike $L_k = I - l_k e_k^T$ z inverzom $L_k^{-1} = I + l_k e_k^T$, kjer je vektor l_k definiran kot

$$l_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ w_{k+1}/w_k \\ \vdots \\ w_n/w_k \end{bmatrix}.$$

Pri tem je w vektor, ki mu L_k uniči vse komponente pod k-to. Matrika $L_{n-1} \dots L_2 L_1 A = U$ je zgornje trikotne oblike. Potem je L kar $L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1}$. Dobimo ga lahko kar tako, da zapišemo vektorje l_k kot stolpce v matriko. Ker vemo, da ima L na diagonali enice, jih lahko izpustimo, in zapišemo celoten razcep kar v eno matriko. Potem lahko sistem $Ax = b$ rešimo v dveh korakih z enostavno premo oz. obratno substitucijo $Ly = b$, $Ux = y$. Enoličen $A = LU$ razcep obstaja natanko tedaj, ko so vse vodilne podmatrike A nesingularne.

Bolj zapomnljivo lahko postopek opišemo na sledeč način. Ko smo na k-tem koraku, razdelimo matriko A na šest delov;

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix}.$$

Pri tem je A_{11} matrike dimenzij $(k-1)\times(k-1)$, A_{12} in A_{22} pa sta stolpca. Korake štejemo od 1 naprej. Elemente podmatrik A_{11}, A_{12}, A_{13} in A_{21} samo prepišemo. Elemente podmatrike A_{22} delimo z diagonalnim elementom a_{kk} (najnižjim v matriki A_{12}), za elemente matrike A_{23} pa naredimo naslednje: Če želimo transformirati element a_{ij}, vzamemo elementa x in y, ki sta direktno levo in direktno nad elementom a_{ij} v blokih $\tilde{A}_{22}$ oziroma A_{13} v novi (tisti, ki je sedaj na pol generirana) matriki, in popravimo $\tilde{a}_{ij} \leftarrow a_{ij} - xy$. Na koncu dobimo LU razcep v kompresirani obliki.

Ves ta opis je deloval za LU razcep brez pivotiranja. Če izvajamo delno pivotiranje, permutiramo neobravnavane vrstice tako, da na mesto pivota pride po absolutni vrednosti čim večji element. Tako dobimo razcep $PA = LU$, kjer je P permutacijska matrika. Pri ročnem postopku to naredimo tako, da menjamo vrstice v spodnjem delu razdeljene matrike, preden izvedemo korak. Permutacijsko matriko P potem dobimo tako, da začnemo z identiteto, in na njej izvajamo enake menjave vrstic kot v postopku. Podobno teče LU razcep s kompletnim pivotiranjem, kjer si dovolimo tudi menjavo stolpcev. Dobimo razcep $PAQ = LU$, kjer matriki P in Q izračunamo posebej; pri obeh začnemo z identiteto, med postopkom pa na P izvajamo le menjave vrstic, na Q pa le menjave stolpcev.

5 Razcep Choleskega

Matrika A je simetrična pozitivno definitna natanko tedaj, ko zanjo obstaja razcep Choleskega $A = VV^T$, kjer je V spodnje trikotna matrika s pozitivnimi diagonalnimi elementi. Razcep Choleskega poiščemo z enostavnim postopkom; za vsak k med 1 in n nastavimo

$$v_{kk} = \sqrt{a_{kk} - \sum_{i=1}^{k-1} v_{ki}^2},$$

nato pa še za $j = k+1, \ldots, n$

$$v_{jk} = \frac{1}{v_{kk}} \left(a_{jk} - \sum_{i=1}^{k-1} v_{ki} v_{ji} \right).$$

V praksi to izgleda tako, da od diagonalnega elementa odštejemo kvadrat vsakega elementa levo od njega, in dobljeno korenimo; nato pa elementom pod njem odštejemo skalarni produkt k-te in j-te vrstice (do k-tega stolpca), in razliko delimo z v_{kk}.

6 Sistemi nelinearnih enačb

Reševanja sistema nelinearnih enačb se lahko lotimo z navadno iteracijo. Če je $G = (G_1, \ldots, G_n)$ iteracijska funkcija, poznamo dva načina iteracije:

- Jacobijeva iteracija: $x^{[r+1]} = G(x^{[r]})$
- Seidlova iteracija: $x^{[r+1]} = (G_1(x^{[r]}), G_2(x_1^{[r+1]}, x_2^{[r]}, \ldots, x_n^{[r]}), \ldots, G_n(x_1^{[r+1]}, \ldots, x_{n-1}^{[r+1]}, x_n^{[r]}))$

Jacobijeva iteracija konvergira na množici, kjer je G skrčitev, za Seidlovo metodo pa velja podobno za preslikavo H, ki jo dobimo tako, da v poznejše argumente G vstavljamo prejšnje slike. Zadosten pogoj je, da je največja absolutna lastna vrednost matrike DG v poljubni točki omejena z nekim $m < 1$, ali pa da je $\|JG(x)\| < 1$ v poljubni matrični normi.

Podobno kot v drugem poglavju imamo tu Newtonovo metodo

$$G(x) = x - (DF(x))^{-1} F(x),$$

kjer pa v praksi namesto računanja inverza rešimo linearen sistem.

7 Predoločeni sistemi

Rešitev predoločenega sistema lahko dobimo z reševanjem normalnega sistema

$$A^T A x = A^T b,$$

kjer je matrika A^TA simetrična pozitivno definitna, torej lahko sistem rešimo z razcepom Choleskega.

Če iščemo minimum $\|Ax-b\|_2$ pod pogojem $Cx=d$, kjer je prvi sistem predoločen in drugi nedoločen, rešitev dobimo kot $\tilde{x}$ v

$$\begin{bmatrix} A^TA & C^T \\ C & 0 \end{bmatrix} \cdot \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} A^Tb \\ d \end{bmatrix}.$$

7.1 QR razcep

Za matriko $A \in \mathbb{R}^{m\times n}$ polnega ranga obstaja enoličen razcep $A=QR$, kjer je Q ortogonalna in R zgornje trikotna s pozitivnimi diagonalnimi elementi. Če najdemo Q, lahko hitro pridemo do $R=Q^TA$. En način iskanja Q je Gram-Schmittova ortogonalizacija

Algorithm 1 QR razcep s klasičnim Gram-Schmittovim postopkom

for $k=1,\dots,n$ **do**
  $q_k = a_k$
  for $i=1,\dots,k-1$ **do**
    $r_{ik} = q_i^T a_k$
    $q_k = q_k - r_{ik}q_i$
  end for
  $r_{kk} = \|q_k\|_2$
  $q_k = \frac{1}{r_{kk}} q_k$
end for

Če namesto $r_{ik}=q_i^Ta_k$ uporabimo $r_{ik}=q_i^Tq_k$, dobimo teoretično ekvivalentni postopek (modificiran Gram-Schmitt), ki je v praksi bolj stabilen. Za boljšo numerično stabilnost lahko pri reševanju (predoločenega) sistema $Ax=b$ izračunamo QR razcep matrike $[Ab]$

$$\begin{bmatrix} A & b \end{bmatrix} = \begin{bmatrix} Q & q_{n+1} \end{bmatrix} \cdot \begin{bmatrix} R & z \\ 0 & \rho \end{bmatrix},$$

potem bo rešitev enaka $x=R^{-1}z$.

Drug način iskanja QR razcepa so Givensove rotacije, kjer vsak poddiagonalni element matrike A eliminiramo s pomočjo

$$A([i,k],i:n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \cdot A([i,k],i:n)$$

za $r=\sqrt{a_{ii}^2+a_{ki}^2}$ in $c=a_{ii}/r$ ter $s=a_{ki}/r$. Če nas zanimata tudi Q in b, ju računamo kar zraven v postopku, kjer moramo matriko Q na koncu še transponirati;

$$Q([i,k],:) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \cdot Q([i,k],:),$$

$$b([i,k]) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \cdot b([i,k]).$$

Najbolj ekonomična so verjetno Householderjeva zrcaljenja. Matrika

$$P = I - \frac{2}{w^T w} w w^T$$

predstavlja zrcaljenje čez hiperravnino z normalo w. Če želimo vektorju a odstraniti vse komponente razen prve, lahko vzamemo

$$w = \begin{bmatrix} a_1 + \operatorname{sgn} a_1 \cdot \|a\|_2 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}.$$

Potem je matrika Pa oblike

$$Pa = \begin{bmatrix} -\operatorname{sgn} a_1 \cdot \|a\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Na vsakem koraku slikamo zadnjih nekaj vrstic matrike A in vektorja b z (novim) P. Pri množenju pazimo na asociativnost; ne računamo matrike ww^T.

8 Lastne vrednosti

Na vajah smo lastne vrednosti računali s potenčno metodo, torej z iteracijo

$$\begin{aligned} \tilde{x}_k &= A x_{k-1} \\ x_k &= \frac{\tilde{x}_k}{\|\tilde{x}_k\|} \\ \lambda &= x_k^T A x_k \end{aligned}$$

Potenčna metoda nam vedno da največjo lastno vrednost. Če želimo poiskati drugo največjo lastno vrednost, lahko poiščemo unitarno matriko U (npr. Householderjevo zrcaljenje), da velja

$$B = U^H A U = \begin{bmatrix} \lambda_1 & \alpha^T \\ 0 & C \end{bmatrix}$$

in nadaljujemo s potenčno metodo na C.

Drug način, ki smo ga omenjali, je QR iteracija

Algorithm 2 QR iteracija z enojnim premikom

Reduciraj A na zgornje Hessenbergovo obliko $A_0 = Q^T AQ$
for $k = 0, 1, \ldots$ **do**
  Izberi premik σ_k
  Naredi QR razcep $A_k - \sigma_k I = Q_k R_k$
  $A_{k+1} = R_k Q_k + \sigma_k I$
end for

9 Polinomska interpolacija

Funkcije običajno interpoliramo v Lagrangeovi bazi

$$l_{i,n}(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

za neke fiksne x_i. Potem je interpolacija oblike

$$p(x) = \sum_{k=0}^{n} f(x_k) l_{k,n}(x).$$

Za $\omega(x) = (x - x_0) \ldots (x - x_n)$ lahko zapišemo polinom tudi v Baricentrični obliki

$$p(x) = \frac{\sum_{k=0}^{n} \frac{w_k}{x - x_k} f(x_k)}{\sum_{k=0}^{n} \frac{w_k}{x - x_k}},$$

kjer je $w_k = \frac{1}{\omega'(x_k)}$.

Druga možnost je Newtonova oblika, ki uporablja deljene diference

$$p(x) = [x_0]_f + [x_0, x_1]_f (x - x_0) + \ldots + [x_0, \ldots, x_n]_f (x - x_0) \ldots (x - x_{n-1}).$$

Te izračunamo rekurzivno po formuli

$$[x_0, \ldots, x_k]_f = \begin{cases} \frac{f^{(k)}(x_0)}{k!} & x_0 = x_1 = \cdots = x_k \\ \frac{[x_1, \ldots, x_k]_f - [x_0, \ldots, x_{k-1}]_f}{x_k - x_0} & \text{sicer} \end{cases}$$

Napaka interpolacije je oblike

$$f(x) - p(x) = [x_0, \ldots, x_n, x]_f (x - x_0) \ldots (x - x_n).$$

10 Numerično integriranje

Numerično integriranje je goljufija. Pravzaprav izračunamo vrednosti f v nekih specifično izbranih točkah, in jih seštejemo z nekimi koeficienti, odvisnimi od izbire točk.

Pri izpeljavi formule si želimo, da bo točna za polinome čim višje stopnje; iz tega tudi izračunamo koeficiente. Pri tem je najlažje integrirati polinome $1, (x-a), (x-a)^2, \ldots$. Na koncu določimo še napako, ki je vedno oblike

$$C f^{(r)}(\xi),$$

kjer C in r določimo tako, da v formulo vstavimo polinom eno (ali dve) stopnjo višje kot prej.

Še posebej goljufiva so Gaussova integracijska pravila, kjer izračunamo točke x_i tako, da bo formula točna za polinome stopnje $\leq 2n+1$. Na standardni polinomski bazi uporabimo Gram-Schmittovo ortogonalizacijo, s čimer dobimo polinome φ_i. Za vozle lahko vzamemo ničle polinoma φ_{n+1}, koeficiente pa določimo kot prej. Seveda nam v praksi tega ni treba delati; mi samo integriramo standardno bazo polinomov do stopnje najvišje natančnosti, in kot običajno zapišemo kvadraturne formule, ter rešimo sistem na koeficiente in na x_i.

Na listu imej standardne kvadraturne formule.