# Predavanja 13

Trinajsti sklop izročkov

Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

10. januar 2022

# Hitrost konvergence potenčne metode

Naj bo $A$ matrika z lastnimi vrednostmi $|\lambda_1| > |\lambda_2| \geqslant |\lambda_3| \geqslant \ldots \geqslant |\lambda_n|$ in $n$ lastnimi vektorji $x_1, \ldots, x_n$.

Iz linearne algebre sledi naslednja trditev:

## Trditev

$$A = U \cdot \underbrace{diag(\lambda_1, \lambda_2, \ldots, \lambda_n)}_{D} \cdot U^T,$$

*kjer je $U$ matrika s stolpci $x_1, \ldots, x_n$. Matrika $U$ je ortogonalna, tj. $UU^T = U^T U = I_n$.*

Iz zgornje trditve sledi, da je

$$\begin{aligned} A^k &= (UDU^T)^k = UD\underbrace{U^T U}_{I_n} D \underbrace{U^T U}_{I_n} \ldots UDU^T = UD^k U^T \\ &= U \cdot diag(\lambda_1^k, \lambda_2^k, \ldots, \lambda_n^k) \cdot U^T \\ &= \lambda_1^k \cdot U \cdot diag\Big(1, (\frac{\lambda_2}{\lambda_1})^k, \ldots, (\frac{\lambda_n}{\lambda_1})^k\Big) \cdot U^T \end{aligned}$$

Hitrost konvergence. Iz računa $A^k$ vidimo, da na hitrost konvergence potenčne metode vpliva velikost $\left|\frac{\lambda_2}{\lambda_1}\right|$. Vemo, da je $\left|\frac{\lambda_2}{\lambda_1}\right| < 1$. Za čim hitrejšo konvergenco pa želimo, da je to čim bližje 0.

Googlova matrika. V primeru Googlove matrike za potrebe PageRank algoritma obstaja naslednja trditev.

## Trditev (Haveliwala, Kamvar, $\approx$ 2000)

*Za Googlovo matriko $\widetilde{H} = \alpha H + (1-\alpha)\frac{1}{N}S$ (oznake kot na prejšnjih izročkih) velja, da je $\lambda_2 = \alpha$.*

Članek o zgornji trditvi.
`https://nlp.stanford.edu/pubs/secondeigenvalue.pdf`

Konvergenca potenčne metode v tem primeru je torej reda $\alpha^k$.

# Inverzna iteracija

Denimo, da želimo poiskati lastni vektor, ki pripada neki lastni vrednosti, ki ni dominantna. V primeru Googlove matrike s prejšnje strani vemo, da je $\alpha$ lastna vrednost, ki ni dominantna.

## Trditev

*Naj bodo $(\lambda_i, x_i)$, $i = 1, \ldots, n$, lastni pari matrike $A$, tj. $Ax_i = \lambda_i x_i$. Velja naslednje:*

1. *Lastni pari matrike $A - \lambda I$ so $(\lambda_i - \lambda, x_i)$, $i = 1, \ldots, n$.*
2. *Če $\lambda$ ni enak nobeni lastni vrednosti $\lambda_i$, $i = 1, \ldots, n$, potem so lastni pari matrike $(A - \lambda I)^{-1}$ enaki $\left(\frac{1}{\lambda_i - \lambda}, x_i\right)$, $i = 1, \ldots, n$.*

## Dokaz.

Točka (1) sledi iz računa:

$$(A - \lambda I)x_i = \lambda_i x_i - \lambda x_i = (\lambda_i - \lambda)x_i. \tag{1}$$

Točka (2) pa sledi z množenjem (1) z leve z $(A - \lambda I)^{-1}$:

$$x_i = (\lambda_i - \lambda)(A - \lambda I)^{-1} x_i. \tag{2}$$

Enakost (2) delimo še z $\lambda_i - \lambda$ in dobimo $\frac{1}{\lambda_i - \lambda} x_i = (A - \lambda I)^{-1} x_i$. $\square$

Naj bo torej $\alpha$ neka nedominantna lastna vrednost. Naj bo $\lambda \approx \alpha$, vendar ne enako $\alpha$. (Numerično bo neenakost skoraj vedno res.)

Potem je po prejšnji trditvi dominantna lastna vrednost matrike $(A - \lambda I)^{-1}$ enaka $\frac{1}{\alpha - \lambda}$.

Lastni vektor matrike $A$, ki pripada $\alpha$, pa je enak lastnemu vektorju matrike $(A - \lambda I)^{-1}$, ki pripada $\frac{1}{\alpha - \lambda}$. Poiščemo pa ga z uporabo potenčne metode za $(A - \lambda I)^{-1}$.

Ker pa je računanje inverza matrike računsko zahtevna operacija, namesto tega na vsakem koraku rešimo linearni sistem v spremenljivki $u$:

$$(A - \lambda I)u = v \qquad (\Leftrightarrow (A - \lambda I)^{-1} v = u).$$

# Inverzna iteracija

```
x^(0) ∈ ℝ^n zacetni vektor
λ premik
ls = max_i |x_i^(0)|
x^(0) = (1/ls) x^(0)

for k = 1 : N
    solve (A − λI) x^(k) = A x^(k−1)
    ln = max_i |x_i^(k)|
    x^(k) = (1/ln) x^(k)
end

x = x^(N)
```

$x^{(0)} \in \mathbb{R}^n$ zacetni vektor

$\lambda$ premik

$ls = \max_i |x_i^{(0)}|$

$x^{(0)} = \frac{1}{ls} x^{(0)}$

for $k = 1 : N$

  solve $(A - \lambda I)x^{(k)} = Ax^{(k-1)}$

  $ln = \max_i |x_i^{(k)}|$

  $x^{(k)} = \frac{1}{ln} x^{(k)}$

end

$x = x^{(N)}$

# Schurova forma

Cilj. Poiskali bi radi vse lastne vrednosti matrike $A$. Naj bo $A \in \mathbb{R}^{n \times n}$ matrika s samimi realnimi lastnimi vrednostmi.

Schurova forma. Izrek iz linearne algebre, ki ima za razliko od izrek o obstoju Jordanove forme matrike precej enostavnejši dokaz, pove, da se da vsako matriko preoblikovati v zgornje trikotno.

### Izrek

*Naj bo $A$ kot zgoraj. Obstaja ortogonalna matrika $U$, da je $UAU^T$ zgornje trikotna matrika, ki ji pravimo Schurova forma matrike $A$.*

Lastne vrednosti zgornje trikotne matrike so ravno diagonalni elementi. Ker pa imata matriki $A$ in $UAU^T$ po naslednji trditvi enake lastne vrednosti, lahko lastne vrednosti matrike $A$ preberemo iz njene Schurove forme.

### Trditev

*Naj bo $S$ obrnljiva matrika. Potem so lastne vrednosti matrik $A$ in $SAS^{-1}$ enake.*

## Dokaz.

Lastne vrednosti matrike so ničle karakterističnega polinoma:

$$\begin{aligned}\det(SAS^{-1} - \lambda I) &= \det(S(A - \lambda I)S^{-1}) \\ &= \det(S^{-1}S(A - \lambda I)) \\ &= \det(A - \lambda I).\end{aligned}$$

Pri tem smo v prvi enakosti uporabili $I = SS^{-1}$ in izpostavili $S$ z leve ter $S^{-1}$ z desne, v drugi neenakosti pa komutativnost determinante, tj. $\det(XY) = \det(YX)$. □

Ker je $U^{-1}$ za ortogonalno matriko enako $U^T$, imata po tej trditvi $UAU^T$ in $A$ enake lastne vrednosti.

# QR iteracija

*QR* razcep. Iz kratkega razdelka o reševanju predoločenih sistemov po metodi najmanjših kvadratov se spomnimo *QR* razcepa matrike $A$. Za vsako matriko $A$ obstaja ortogonalna matrika $Q$ in zgornje trikotna matrika $R$, da velja $A = QR$. En način za izračun *QR* razcepa je Gram-Schmidtov algoritem iz linearne algebre. V Matlabu pa do *QR* razcepa pridemo z ukazom $[Q, R] = qr(A)$.

Če je $A = QR$, potem je

$$Q^T A Q = Q^T (QR) Q = Q^T Q R Q = RQ.$$

Torej sta po trditvi iz prejšnje strani matriki *QR* in *RQ* podobni ter imata enake lastne vrednosti. To je temelj za postopek iskanja lastnih vrednosti matrike, ki se imenuje *QR* iteracija.

```
A^(0) = A

for k = 1 : N
    [Q, R] = qr(A^(k-1))
    A^(k) = RQ
end
Priblizki last. vred. so diagonalni elementi A^(N).
```

Za dejstvo, da so približki lastnih vrednosti diagonalni elementi $A$ ne zadošča, da imata $QR$ in $RQ$ iste lastne vrednosti. Ta sklep sledi iz naslednje trditve:

### Trditev

*Matrike $A^{(N)}$ iz $QR$ iteracije konvergirajo proti Schurovi formi matrike $A$.*

Hitrost konvergence. Kot pri potenčni metodi, kjer je bila hitrost konvergence odvisna od razmerja $\left|\frac{\lambda_2}{\lambda_1}\right|$, se da tu izpeljati, da je hitrost konvergence $QR$ iteracije odvisna od razmerij $\left|\frac{\lambda_{i+1}}{\lambda_i}\right|$, $i = 1, \ldots, n-1$, kjer je $|\lambda_1| \geqslant |\lambda_2| \geqslant \ldots \geqslant |\lambda_n|$. Če je torej kakšno razmerje $\left|\frac{\lambda_{i+1}}{\lambda_i}\right|$ blizu 1, potem je konvergenca počasna.

Število potrebnih operacij. Za izračun $QR$ razcepa matrike potrebujemo $\frac{4}{3}n^3$ računskih operacij. Čez palec za konvergenco $QR$ iteracije potrebujemo $n$ ponovitev. Skupaj torej $\mathcal{O}(n^4)$ operacij. Potenca 4 pa je numerično slaba za uporabo na velikih matrikah, tako da je za praktično uporabo potrebno narediti izboljšave.

# Pospešitve QR iteracije

Prva izboljšava. Da zmanjšamo računsko zahtevnost $QR$ iteracije, je smiselno $A$ najprej preoblikovati v zgornjo Hessenbergovo obliko (ZHO), tj. poleg zgornjega trikotnika je lahko $A$ neničelna še v pri poddiagonali:

$$A = \begin{bmatrix} * & * & \cdots & \cdots & * \\ * & * & & & * \\ 0 & * & \ddots & & * \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * & * \end{bmatrix}.$$

Pomembna trditev za smiselnost te prevedbe je naslednja:

### Trditev

*ZHO se ohranja med QR iteracijo, tj. če ima $A = QR$ to obliko, jo ima tudi $RQ$.*

Za prevedbo matrike v ZHO potrebujemo $\mathcal{O}(n^3)$ operacij (z uporabo Householderjevih zrcaljenj; podrobnosti izpustimo).

Za izračun $QR$ razcepa zgornje Hessenbergove matrike pa potrebujemo samo $\mathcal{O}(n^2)$ operacij (z uporabo Givensovih rotacij; podrobnosti izpustimo).

# Pospešitve QR iteracije

Druga izboljšava. Da pospešimo konvergenco, je smiselno narediti premike kot pri potenčni metodi.

```
A^(0) = A

for k = 1:N
    izberi premik σ_k
    [Q, R] = qr(A^(k-1) - σ_k I)
    A^(k) = RQ + σ_k I
end
Priblizki last. vred. so diagonalni elementi A^(N).
```

S premikom lastne vrednosti postanejo $\lambda_1 - \sigma_k, \ldots, \lambda_n - \sigma_k$. Razmerja $|\frac{\lambda_{i+1}}{\lambda_i}|$ postanejo $|\frac{\lambda_j - \sigma_k}{\lambda_i - \sigma_k}|$, ki so lahko bistveno ugodnejša, tj. bolj oddaljena od 1, če smo le izbrali premik $\sigma_k$ smiselno (npr. v bližini ene od lastnih vrednosti).

# Izbira premika

Enojni premik. Za premike izbiramo spodnji desni vogal matrike $A$. Po dovolj iteracijah namreč spodnji desni vogal postane lastna vrednost matrike $A$.

```
A^(0) = A

for k = 1:N
    izberi premik a_nn^(k-1)
    [Q, R] = qr(A^(k-1) - a_nn^(k-1) I)
    A^(k) = RQ + a_nn^(k-1) I
end
Priblizki last. vred. so diagonalni elementi A^(N).
```

# Izbira premika

Dvojni premik. Za premike izbiramo spodnji desni $2 \times 2$ vogal matrike $A$. Izračunamo obe lastni vrednosti spodnjega vogala in naredimo dva premika $\sigma_1 I$, nato pa še $\sigma_2 I$:

$$\begin{aligned} A - \sigma_1 I &= QR \\ A' &= RQ + \sigma_1 I \\ A' - \sigma_2 I &= Q'R' \\ A'' &= R'Q' + \sigma_2 I. \end{aligned}$$

Zgornji postopek pa lahko naredimo še enostavneje:

$$\begin{aligned} QQ'R'R &= Q(A' - \sigma_2 I)R = QQ^T(A - \sigma_2 I)QR = (A - \sigma_2 I)QR \\ &= (A - \sigma_2 I)(A - \sigma_1 I) \\ &= A^2 - (\sigma_2 + \sigma_1)A + \sigma_1\sigma_2 I =: N \end{aligned}$$

# Algoritem dvojnega premika

$A^{(0)} = A$

**for** $k = 1 : N$

$\qquad \sigma_1 + \sigma_2 = a_{n-1,n-1}^{(k-1)} + a_{n,n}^{(k-1)}$

$\qquad \sigma_1 \sigma_2 = a_{n-1,n-1}^{(k-1)} a_{n,n}^{(k-1)} - a_{n,n-1}^{(k-1)} a_{n-1,n}^{(k-1)}$

$\qquad N = (A^{k-1})^2 - (\sigma_2 + \sigma_1) A^{(k-1)} + \sigma_1 \sigma_2 I$

$\qquad [Q_k, R_k] = qr(N)$

$\qquad A^{(k)} = Q_k^T A^{(k-1)} Q_k$

**end**

Priblizki last. vred. so diagonalni elementi $A^{(N)}$.

# Parcialne diferencialne enačbe

Parcialne diferencialne enačbe (PDE) so enačbe, v katerih iščemo funkcijo $u$, odvisno od več neodvisnih spremenljivk, v enačbi pa nastopajo odvodi po neodvisnih spremenljivkah.

## Primer

*Rešujemo PDE*

$$u_{xx}(x,t) = u_t(x,t), \qquad (x,t) \in [0,1] \times [0,T],$$

*pri robnih pogojih*

$$\begin{aligned} u(x,0) &= g(x), \\ u(0,t) &= a(t), \\ u(1,t) &= b(t), \end{aligned}$$

*kjer so $g, a, b$ dane funkcije, $u_{xx}$ predstavlja drugi odvod $u$ po $x$, $u_t$ pa prvi odvod po $t$.*

Obstajata dve metodi za reševanje PDE: metoda končnih diferenc in metoda končnih elementov.

# Metoda končnih diferenc

Območje $[0, 1] \times [0, T]$ razdelimo na mrežo $n \times m$ točk. Torej $[0, 1]$ razdelimo na $n + 1$ ekvidistantnih točk $0 = x_0, x_1, \ldots, x_n = 1$, $[0, T]$ pa na $m + 1$ ekvidistantnih točk $0 = t_0, t_1, \ldots, t_m = T$. Označimo vrednost približka rešitve $u(x_i, t_j)$ z $u_{i,j}$.

Uporabimo približke prvih in drugih odvodov:

$$
\begin{aligned}
f'(x) &\approx \frac{1}{h}(f(h+h) - f(x)),\\
f'(x) &\approx \frac{1}{h}(f(x) - f(x-h)),\\
f''(x) &\approx \frac{1}{h^2}(f(x+h) - 2f(x) + f(x-h).
\end{aligned}
$$

Če upoštevamo to v enačbi iz primera, dobimo

$$
\frac{1}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) = \frac{1}{k}(u_{i,j+1} - u_{i,j}), \tag{3}
$$

kjer je $h = x_{i+1} - x_i$, $k = t_{j+1} - t_j$.

Zaradi robnih pogojev iz primera vse vrednosti $u_{i,0}$, $i = 0, \ldots, n$, in $u_{0,j}$, $u_{1,j}$, $j = 0, \ldots, m$, poznamo. Ostale vrednosti pa lahko eksplicitno izračunamo tako, da iz enačbe (3) izrazimo $u_{i,j+1}$, nato pa rešitve izračunavamo glede na naraščajočo drugo koordinato.

Pravkar smo izpeljali eksplicitno metodo za reševanje PDE iz primera.

Če bi na desni strani (3) uporabili aproksimacijo $\frac{1}{k}(u_{i,j} - u_{i,j-1})$, potem bi dobili implicitno metodo za reševanje PDE iz primera.

Implicitna metoda za stabilnost izračuna ne potrebuje omejitev na $h$ in $k$.

# Metoda končnih elementov

Pri metodi končnih elementov naj bo $\Omega$ območje, kjer rešujemo PDE. Iskana funkcija $u$ je zvezna funkcija na tem območju. Torej rešitev iščemo med zveznimi funkcijami. Ker prostor zveznih funkcij ni končno dimenzionalen, ne moremo izbrati končne baze prostora. Lahko pa izberemo neko podmnožico, ki jo imenujemo bazne funkcije $\phi_1, \ldots, \phi_k$.

Naša rešitev $u$ je potem linearna kombinacija baznih funkcij $\phi_1, \ldots, \phi_k$. Torej je $u = \sum_{j=1}^{k} c_j \phi_j$.

Če je $\partial\Omega$ rob območja $\Omega$, robni pogoj pa $u|_{\partial\Omega} = f$, potem iščemo koeficiente $c_1, \ldots, c_k$, da je izraz

$$\Big\| \sum_{j=1}^{k} c_j \phi_j - f \Big\| \tag{4}$$

minimalen.
Prostor zveznih funkcij na robu $\partial\Omega$ lahko spremenimo v vektorski prostor s skalarnim produktom tako, da definiramo normo

$$\langle g, h \rangle = \int_{\partial\Omega} fg \, d\partial\Omega.$$

Izraz (4) pa bo minimalen, ko bo veljalo

$$\sum_{j=1}^{k} c_j \phi_j - f \perp \phi_i, \qquad i = 1, \ldots, k.$$

Torej mora biti

$$\langle \sum_{j=1}^{k} c_j \phi_j - f, \phi_i \rangle = 0, \qquad i = 1, \ldots, k.$$

Torej

$$\sum_{j=1}^{k} c_j \langle \phi_j, \phi_i \rangle - \langle f, \phi_i \rangle = 0, \qquad i = 1, \ldots, k.$$

To pa je linearni sistem

$$Ac = b,$$

kjer je

$$A_{j,i} = \langle \phi_j, \phi_i \rangle, \qquad b_i = \langle f, \phi_i \rangle,$$

in ga rešimo z LU razcepom.