Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

ITERATIVNE METODE

Vježbe 04 - Rješavanje problema svojstvenih vrijednosti

9. prosinca 2021.

Sastavili: Nela Bosner, Zvonimir Bujanović

UVOD. OSNOVE TEORIJE PERTURBACIJA.

Zadatak 1

Učitajte matrice `A` i `B` iz datoteke `pseudoSpektar.mat`.

Za svaki od $\varepsilon \in \{10^{-1}, 10^{-2}, \dots, 10^{-6}\}$ generirajte po 1000 matrica E takvih da je $\|E\| = \varepsilon$, te nacrtajte u kompleksnoj ravnini spektre svih matrica $A + E$ (odnosno $B + E$). Za svaki ε upotrijebite drugu boju. Na kraju nacrtajte i spektar matrice A (odnosno B).

Spektar koje matrice je osjetljiviji na perturbacije?

Rješenje možete vidjeti u datotekama `pseudoSpektar_A/B.fig`. "Zoom-ajte" u područja oko svojstvenih vrijednosti (crni x-evi).

ε	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	0
boja							

Teorem (Ostrowski-Elsner)

Neka su $A, E \in \mathbb{C}^{n \times n}$*.*
Neka je $\Lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$*, te* $\Lambda(A+E) = \{\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n\}$*.*

Tada postoji permutacija π *takva da je*

$$|\lambda_i - \tilde{\lambda}_{\pi(i)}| \leq (2n-1)\left(\|A\| + \|A+E\|\right)^{1-1/n} \cdot \|E\|^{1/n}.$$

Teorem (Bauer-Fike)

Neka je $A \in \mathbb{C}^{n\times n}$ dijagonalizabilna, $A = S\Lambda S^{-1}$.
Neka je $\Lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$.

Tada za svaki $\tilde{\lambda} \in \Lambda(A+E)$ postoji i tako da je

$$|\tilde{\lambda} - \lambda_i| \leq \kappa(S) \cdot \|E\|\,.$$

Osnovni perturbacijski teoremi

Teorem (Weyl, Wielandt-Hoffmann)

Neka su $A, E \in \mathbb{C}^{n\times n}$ *hermitske matrice.*
Neka su $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ *svojstvene vrijednosti od* A*.*
Neka su $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots \geq \tilde{\lambda}_n$ *svojstvene vrijednosti od* $A + E$*.*

Tada vrijedi:

- *Weylov teorem: za svaki* $i = 1, 2, \ldots, n$ *je*

$$|\tilde{\lambda}_i - \lambda_i| \leq \|E\| \,.$$

- *Wielandt-Hoffmann-ov teorem:*

$$\sqrt{\sum_{i=1}^{n} (\tilde{\lambda}_i - \lambda_i)^2} \leq \|E\|_F \,.$$

JACOBIJEV ALGORITAM

Jacobi za 2x2 matricu

Neka $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix}$.

Tražimo $U = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ takvu da je $U^\mathsf{T} \cdot A \cdot U$ dijagonalna:

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} \tilde{a}_{11} & 0 \\ 0 & \tilde{a}_{22} \end{bmatrix}$$

Jacobi za 2x2 matricu

Formule:

$$
\begin{aligned}
\tau &= \frac{a_{22} - a_{11}}{2a_{12}} \\
t &= \frac{sign(\tau)}{|\tau| + \sqrt{\tau^2 + 1}} \quad (t = 1 \text{ ako } \tau = 0) \\
c &= \frac{1}{\sqrt{1 + t^2}} \\
s &= t \cdot c
\end{aligned}
$$

Dobivamo još:

$$
\begin{aligned}
\tilde{a}_{11} &= a_{11} - t \cdot a_{12} \\
\tilde{a}_{22} &= a_{22} + t \cdot a_{12}
\end{aligned}
$$

Klasični Jacobijev algoritam

Algoritam (Klasični Jacobijev algoritam)

while(*true*)
  Pronađi A_{p_i,q_i} *u gornjem trokutu od A maksimalne aps. vrijednosti;*
  Ako je $|A_{p_i,q_i}| < tol$*, izađi iz petlje;*
  Nađi Jacobijevu rotaciju (tj. c, s, t*) za podmatricu* $\begin{bmatrix} A_{p_i,p_i} & A_{p_i,q_i} \\ A_{q_i,p_i} & A_{q_i,q_i} \end{bmatrix}$*;*
  $\tilde{A}_{p_i,p_i} = A_{p_i,p_i} - t \cdot A_{p_i,q_i}$;
  $\tilde{A}_{q_i,q_i} = A_{q_i,q_i} + t \cdot A_{p_i,q_i}$;
  $\tilde{A}_{q_i,p_i} = \tilde{A}_{p_i,q_i} = 0$;
  for $k = 1, 2, \ldots, n, k \neq p_i, q_i$
    $\tilde{A}_{p_i,k} = \tilde{A}_{k,p_i} = c \cdot A_{p_i,k} - s \cdot A_{q_i,k}$;
    $\tilde{A}_{q_i,k} = \tilde{A}_{k,q_i} = s \cdot A_{p_i,k} + c \cdot A_{q_i,k}$;
  end
  $A = \tilde{A}$;
end

Ciklički Jacobijev algoritam

Algoritam (Ciklički Jacobijev algoritam)

while(*true*)
  for $p_i = 1, 2, \ldots, n$
    for $q_i = p_i + 1, \ldots, n$
      Ako je $|A_{p_i,q_i}| < tol$, *continue (ne rotiraj);*
      Nađi Jac. rot.(tj. c, s, t*) za podmatricu* $\begin{bmatrix} A_{p_i,p_i} & A_{p_i,q_i} \\ A_{q_i,p_i} & A_{q_i,q_i} \end{bmatrix}$;
      $\tilde{A}_{p_i,p_i} = A_{p_i,p_i} - t \cdot A_{p_i,q_i}$; $\tilde{A}_{q_i,q_i} = A_{q_i,q_i} + t \cdot A_{p_i,q_i}$;
      $\tilde{A}_{q_i,p_i} = \tilde{A}_{p_i,q_i} = 0$;
      for $k = 1, 2, \ldots, n$, $k \neq p_i, q_i$
        $\tilde{A}_{p_i,k} = \tilde{A}_{k,p_i} = c \cdot A_{p_i,k} - s \cdot A_{q_i,k}$;
        $\tilde{A}_{q_i,k} = \tilde{A}_{k,q_i} = s \cdot A_{p_i,k} + c \cdot A_{q_i,k}$;
      end
  end
  $A = \tilde{A}$;
  Ako for-petlje nisu obavile niti jednu rotaciju, izađi van iz while-a;
end

Zadatak 2

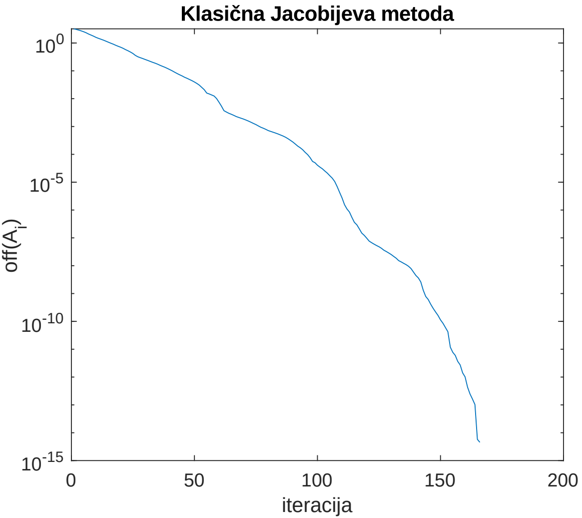
1. Napišite funkciju `[t, c, s]=JacRot(A)` koja za 2×2 matricu A vraća odgovarajuće parametre rotacije koja ju dijagonalizira.

2. Napišite funkcije `[Lam, U] = JacobiKlasicni(A)` i `[Lam, U] = JacobiCiklicki(A)` koje nalaze svojstvene vrijednosti i vektore simetrične matrice A koristeći odgovarajuće pivotne strategije Jacobijevog algoritma. Implementirati pažljivo i efikasno!

3. Testirajte metode iz (b) za tzv. Risovu matricu: $A \in \mathbb{R}^{n \times n}$, uz $A_{ij} = \frac{1}{2(n-i-j+1.5)}$; stavite npr. $n = 10$. Koliko iteracija/*sweepova* treba? Nacrtajte grafove off-normi.

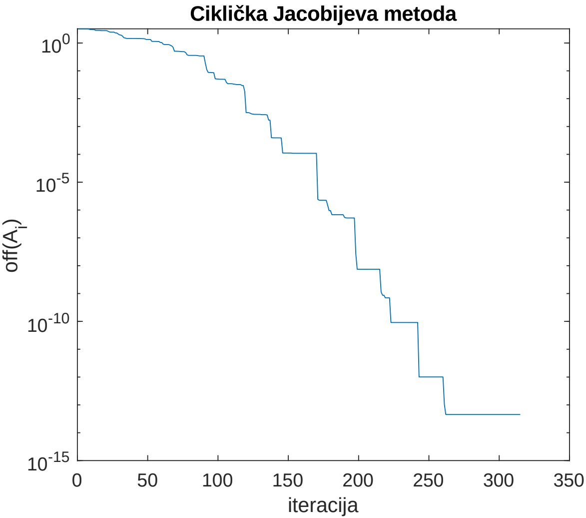
Zadatak 2

Rješenje. Klasični Jacobijev algoritam.



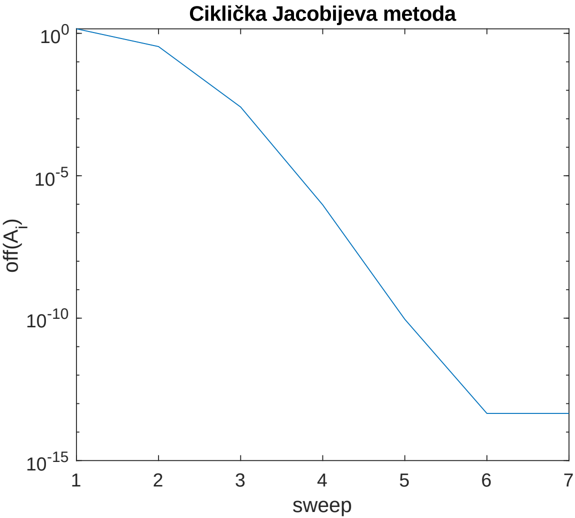
Zadatak 2

Rješenje. Ciklički Jacobijev algoritam.



Zadatak 2

Rješenje. Ciklički Jacobijev algoritam.



QR-ITERACIJE

Zadatak 3

Generirajte slučajnu simetričnu 10×10 matricu A_0.
Neka je $A_i = Q_i R_i$ QR-faktorizacija, te neka je $A_{i+1} = R_i Q_i$.

Što možete reći o matricama A_{100}, A_{1000}?
Ponovite više puta.

Eksplicitni QR-algoritam

Algoritam (Eksplicitni QR-algoritam)

$k = 0$; $A_0 = A$;
while *matrica A nije blizu dijagonalne*
    Odaberi shift σ_k;
    $A_k - \sigma_k I = QR$;
    $A_{k+1} = Q^\top A_k Q$;
end

Eksplicitni QR-algoritam

Problemi:

1. QR na punoj matrici u svakom koraku je spor;
2. Transformacija $A_{k+1} = Q^{\mathsf{T}} A_k Q$ je spora;
3. Kako odabrati shiftove?

Zadatak 4

1. Napišite funkciju `[v, alfa] = house(a)` koja nalazi vektor v i skalar *alfa* takve da je $Ha = alfa \cdot e_1$, te $H = I - 2vv^\top$.

2. Koristeći funkciju `house`, napišite funkciju `[T,Q] = tridijag(A)` koja za simetričnu matricu A pronalazi Q i T takve da je T tridijagonalna, Q ortogonalna, te $T = QAQ^\top$.

Zadatak 4 - Skica rješenja

Algoritam (Svođenje na tridijagonalnu formu)

function *[A, Q] = tridijag(A)*
  $Q = I_n$;
  for *s = 1 : n-2*
    $[v, \alpha] = house(A(s+1:n, s))$;

    $A(s+1, s) = \alpha$; $A(s+2:n, s) = 0$;
    $A(s, s+1) = \alpha$; $A(s, s+2:n) = 0$;

    Updateaj X:=A(s+1:n, s+1:n) formulama
      $u = Xv$; $w = u - (v^\top u)v$;
      $X = X - 2vw^\top - 2wv^\top$

    Updateaj svaki stupac q od $Q(s+1:n, :)$ *formulama*
      $q = q - 2(v^\top q)v$;
  end
end

Zadatak 5

1. Napišite funkciju `[c,s] = givens(x,y)` koja pronalazi rotaciju $\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ takvu da je $\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \star \\ 0 \end{bmatrix}$.

2. Koristeći funkciju `givens`, napišite funkciju `[A,Q] = bulgeChase(A,Q,st,en,shift)` koja provodi *bulge-chase* sa shiftom σ na podmatrici $A(st : en, st : en)$ simetrične tridijagonalne matrice A, te radi odgovarajuće transformacije i na ortogonalnoj matrici Q.

3. Koristeći funkcije `tridijag` i `bulgeChase`, napišite funkciju `[Q,Lam] = QRiter(A)` koja izvodi QR-algoritam s implicitnim shiftovima za simetričnu matricu A.

Zadatak 5 - Skica rješenja

Algoritam (QR-iteracija s implicitnim shiftovima)

while *matrica A nije dijagonalna*

Postavi na nulu sve elemente $A_{i+1,i}$ (kao i $A_{i,i+1}$) takve da je

$$|A_{i+1,i}| < \varepsilon(|A_{i,i}| + |A_{i+1,i+1}|).$$

Pronađi najveće moguće matrice A_2 i A_3 takve da je:

A_3 dijagonalna, A_2 nereducirana tridijagonalna,

$$A = \begin{bmatrix} A_1 & & \\ & A_2 & \\ & & A_3 \end{bmatrix}.$$

Odaberi shift σ za podmatricu A_2.

Provedi bulge chase s tim shiftom na podmatrici A_2.

end

Algoritam (bulge-chase)

```
function [A,Q] = bulgeChase(A,Q,st,en,σ)
    x = A(st,st) − σ;
    y = A(st+1,st);
    for k = st : en-1
        [c,s] = givens(x,y);

        Pomnoži stupce k,k+1 od A zdesna sa [c −s; s c];    (O(1)!)
        Pomnoži retke k,k+1 od A slijeva sa [c s; −s c];    (O(1)!)
        Pomnoži retke k,k+1 od Q slijeva sa [c s; −s c];

        if k < en-1
            x = A(k+1,k);
            y = A(k+2,k);
        end
    end
end
```

Wilkinsonov shift

Shift je ona svojstvena vrijednost podmatrice

$$\begin{bmatrix} A_{n-1,n-1} & A_{n-1,n} \\ A_{n,n-1} & A_{n,n} \end{bmatrix}$$

koja je bliža $A_{n,n}$:

$$d = (A_{n-1,n-1} - A_{n,n})/2$$

$$\sigma = A_{n,n} - \frac{A_{n-1,n}^2}{d + sign(d)\sqrt{d^2 + A_{n,n-1}^2}}.$$

QR-iteracije: Primjer

Neka je polazna matrica

$$A = \begin{bmatrix} 5.0 & 4.0 & 3.0 & 2.0 & 1.0 \\ 4.0 & 5.0 & 4.0 & 3.0 & 2.0 \\ 3.0 & 4.0 & 5.0 & 4.0 & 3.0 \\ 2.0 & 3.0 & 4.0 & 5.0 & 4.0 \\ 1.0 & 2.0 & 3.0 & 4.0 & 5.0 \end{bmatrix}$$

QR-iteracije: Primjer

Nakon svođenja na tridijagonalnu formu $\tilde{A} = Q^T A Q$, QR-iteracije izvodimo na matrici

$$\tilde{A} = \begin{bmatrix} 5.0 & -5.477225 & 0 & 0 & 0 \\ -5.477225 & 13.066666 & 4.419150 & 0 & 0 \\ 0 & 4.419150 & 5.305204 & 0.567750 & 0 \\ 0 & 0 & 0.567750 & 0.968693 & -0.143790 \\ 0 & 0 & 0 & -0.143790 & 0.659436 \end{bmatrix}$$

Vidimo da nema malih poddijagonalnih elemenata, pa je “aktivna” cijela matrica.

Izračunamo shift (pomoću A_{54}, A_{44}, A_{55}):

$$\sigma = 0.602911571890463$$

QR-iteracije: Primjer

Nakon prvog shifta je

$$\tilde{A} = \begin{bmatrix} 15.252996 & -4.411234 & 0 & 0 & 0 \\ -4.411234 & 7.032563 & 0.750092 & 0 & 0 \\ 0 & 0.750092 & 1.383213 & 0.098890 & 0 \\ 0 & 0 & 0.098890 & 0.764488 & 0.060898 \\ 0 & 0 & 0 & 0.060898 & 0.566738 \end{bmatrix}$$

Ponovno nema malih poddijagonalnih elemenata, pa je “aktivna” cijela matrica.

Izračunamo shift (pomoću A_{54}, A_{44}, A_{55}):

$$\sigma = 0.549488944780760$$

Nakon drugog shifta je

$$\tilde{A} = \begin{bmatrix} 17.002479 & -1.436400 & 0 & 0 & 0 \\ -1.436400 & 5.408444 & 0.108532 & 0 & 0 \\ 0 & 0.108532 & 1.275110 & 0.028534 & 0 \\ 0 & 0 & 0.028534 & 0.765532 & -0.000322 \\ 0 & 0 & 0 & -0.000322 & 0.548432 \end{bmatrix}$$

Ponovno nema malih poddijagonalnih elemenata, pa je “aktivna” cijela matrica.

Izračunamo shift (pomoću A_{54}, A_{44}, A_{55}):

$$\sigma = 0.548431622866874$$

QR-iteracije: Primjer

Nakon trećeg shifta je

$$\tilde{A} = \begin{bmatrix} 17.163683 & -0.410303 & 0 & 0 & 0 \\ -0.410303 & 5.250112 & 0.016668 & 0 & 0 \\ 0 & 0.016668 & 1.273699 & 0.008498 & 0 \\ 0 & 0 & 0.008498 & 0.764073 & 3.72 \cdot 10^{-12} \\ 0 & 0 & 0 & 3.72 \cdot 10^{-12} & 0.548431 \end{bmatrix}$$

I dalje nema dovoljno malih poddijagonalnih elemenata, pa je "aktivna" cijela matrica.

Izračunamo shift (pomoæu A_{54}, A_{44}, A_{55}):

$$\sigma = 0.548431620374110$$

QR-iteracije: Primjer

Nakon četvrtog shifta je

$$\tilde{A} = \begin{bmatrix} 17.176674 & -0.115785 & 0 & 0 & 0 \\ -0.115785 & 5.237189 & 0.002577 & 0 & 0 \\ 0 & 0.002577 & 1.273759 & 0.002525 & 0 \\ 0 & 0 & 0.002525 & 0.763944 & -1.3 \cdot 10^{-16} \\ 0 & 0 & 0 & -1.3 \cdot 10^{-16} & 0.548431 \end{bmatrix}$$

Sada je A_{54} dovoljno malen, pa ga postavljamo na 0. “Aktivna” je podmatrica $A(1:4, 1:4)$.

Izračunamo shift (pomoæu A_{43}, A_{33}, A_{44}):

$$\sigma = 0.763932022536664$$

Nakon petog shifta je

$$\tilde{A} = \begin{bmatrix} 17.177714 & -0.031549 & 0 & 0 & 0 \\ -0.031549 & 5.236151 & 0.000293 & 0 & 0 \\ 0 & 0.000293 & 1.273770 & -1.8 \cdot 10^{-13} & 0 \\ 0 & 0 & -1.8 \cdot 10^{-13} & 0.763932 & 0 \\ 0 & 0 & 0 & 0 & 0.548431 \end{bmatrix}$$

A_{43} nije dovoljno malen. “Aktivna” podmatrica ostaje $A(1:4, 1:4)$.

Izračunamo shift (pomoću A_{43}, A_{33}, A_{44}):

$$\sigma = 0.763932022500210$$

Nakon šestog shifta je

$$\tilde{A} = \begin{bmatrix} 17.177791 & -0.008596 & 0 & 0 & 0 \\ -0.008596 & 5.236074 & 0.000033 & 0 & 0 \\ 0 & 0.000033 & 1.273770 & 4.6 \cdot 10^{-17} & 0 \\ 0 & 0 & 4.6 \cdot 10^{-17} & 0.763932 & 0 \\ 0 & 0 & 0 & 0 & 0.548431 \end{bmatrix}$$

Sada je A_{43} dovoljno malen, pa ga postavljamo na 0. “Aktivna” je podmatrica $A(1:3, 1:3)$.

Izračunamo shift (pomoću A_{32}, A_{33}, A_{22}):

$$\sigma = 1.27377076792510 2$$

QR-iteracije: Primjer

Nakon sedmog shifta je

$$\tilde{A} = \begin{bmatrix} 17.177797 & -0.002141 & 0 & 0 & 0 \\ -0.002141 & 5.236068 & -4.5 \cdot 10^{-16} & 0 & 0 \\ 0 & -4.5 \cdot 10^{-16} & 1.273770 & 0 & 0 \\ 0 & 0 & 0 & 0.763932 & 0 \\ 0 & 0 & 0 & 0 & 0.548431 \end{bmatrix}$$

A_{32} je odmah dovoljno malen, pa ga postavljamo na 0. “Aktivna” podmatrica poostaje $A(1:2, 1:2)$.

Izračunamo shift (pomoću A_{21}, A_{22}, A_{11}):

$$\sigma = 5.236067977499796$$

QR-iteracije: Primjer

Konačno, nakon osmog shifta sve poddijagonalne elemente od A možemo postaviti na nulu. Na dijagonali dobivamo svojstvene vrijednosti:

$$\Lambda = \begin{bmatrix} 17.177797611700804 \\ 5.236067977499795 \\ 1.273770767925102 \\ 0.763932022500210 \\ 0.548431620374110 \end{bmatrix}$$

METODA BISEKCIJE

Zadatak 6

1. Napišite funkciju `lam = bisekcija(A,k)` koja pronalazi k-tu po veličini svojstvenu vrijednost simetrične tridijagonalne matrice A koristeći metodu bisekcije.

2. Metodu testirajte na nekoj matrici generiranoj Matlabovom naredbom
   ```
   A = full(gallery('tridiag', X, Y, Z))
   ```
   gdje su $X = Z$ vektori duljine $n - 1$ koji sadrže elemente ispod/iznad glavne dijagonale od A, a Y je vektor duljine n koji sadrži dijagonalne elemente od A.

3. Napišite funkciju `Lam = bisekcijaSve(A)` koja metodom bisekcije pronalazi sve svojstvene vrijednosti simetrične tridijagonalne matrice A.

Zadatak 6

Algoritam (bisekcija)

function *lam* = *bisekcija(A, k)*
    Odaberi $[a_0, b_0]$ *koji sadrži sve svojstvene vrijednosti od A;*
    $i = 0$;
    while $b_i - a_i > tol$
        $mid = (a_i + b_i)/2$;
        x = broj svojstvenih vrijednosti većih od mid;
        if $x \geq k$
            $a_{i+1} = mid$; $b_{i+1} = b_i$;
        else
            $a_{i+1} = a_i$; $b_{i+1} = mid$;
    end
    $lam = (a_i + b_i)/2$;

Zadatak 6

Broj svojstvenih vrijednosti većih od σ je jednak broju promjena predznaka među uzastopnim članovima niza

$$k_0(\sigma), \quad k_1(\sigma), \quad \dots \quad , k_{n-1}(\sigma), \quad k_n(\sigma)$$

(na primjer, ako je niz “2, -3, -4, 1, 2, 6, -5, -3”, onda sadrži 3 promjene predznaka).

$$\begin{aligned} k_0(\sigma) &= 1 \\ k_{j+1}(\sigma) &= (\sigma - A_{j+1,j+1})k_j(\sigma) - A_{j+1,j}^2 \cdot k_{j-1}(\sigma) \end{aligned}$$

METODA POTENCIJA. METODA INVERZNIH ITERACIJA.

Zadatak 7

1. Napišite funkciju `[lam,x] = metodaPotencija(A)` koja pronalazi po modulu najveću svojstvenu vrijednost matrice A koristeći metodu potencija, te pripadni svojstveni vektor.

2. Napišite funkciju `[lam,x] = inverzneIteracije(A, target)` koja pronalazi svojstvenu vrijednost matrice A najbližu zadanom broju `target` koristeći metodu inverznih iteracija, te pripadni svojstveni vektor.

3. Metode testirajte na matricama A i B iz datoteke `'pseudoSpektar.mat'`, te na matricama $A + \mathrm{i}I$, $B + \mathrm{i}I$ ($\mathrm{i} = \sqrt{-1}$). Nacrtajte grafove konvergencije normi reziduala. Usporedite s očekivanom brzinom konvergencije!

Zadatak 7

Algoritam (metoda potencijla)

function *[lam, x] = metodaPotencija(A)*
  Odaberi početni vektor x norme 1*;*
  $\rho = x^{\mathsf{T}} A x;$
  while $\|Ax - \rho x\| > tol$
    $x = Ax / \|Ax\|;$
    $\rho = x^{\mathsf{T}} A x;$
  end
  $lam = \rho;$
end

Treba implementirati tako da se u jednom prolazu kroz petlju samo jednom računa Ax!

Zadatak 7

Algoritam (inverze iteracije)

function *[lam, x] = inverzneIteracije(A, τ)*
  Odaberi početni vektor x norme 1;
  $\rho = x^T Ax$;
  while $\|Ax - \rho x\| > tol$
    Riješi sustav $(A - \tau \cdot I)y = x$;
    $x = y / \|y\|$;
    $\rho = x^T Ax$;
  end
  lam = ρ;
end

Treba implementirati tako da se u jednom prolazu kroz petlju samo jednom računa *Ax*!

Zadatak 7

Brzina konvergencije:

- metoda potencija: neka $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$. Konvergencija je linearna s faktorom

$$k = \frac{|\lambda_2|}{|\lambda_1|}.$$

- inverzne iteracije: neka $|\lambda_1 - \tau| < |\lambda_2 - \tau| \leq |\lambda_3 - \tau| \leq \ldots \leq |\lambda_n - \tau|$. Konvergencija je linearna s faktorom

$$k = \frac{|\lambda_1 - \tau|}{|\lambda_2 - \tau|}.$$

Na idućim slikama, crvena crta je $\|Ax_0 - \rho_0 x_0\| \cdot k^i$, x_0 je početni vektor, $\rho_0 = x_0^\top A x_0$, i je redni broj iteracije.

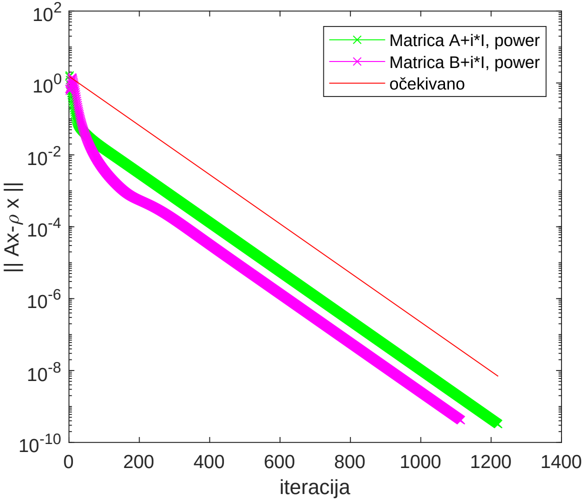
Zadatak 7

Matrice A i B imaju isti spektar.

- Metoda potencija ne konvergira za matrice A i B jer postoje dvije svojstvene vrijednosti λ_1 i λ_2 koje imaju isti (najveći) modul: $\frac{|\lambda_2|}{|\lambda_1|} = 1$.
- Matrice $A + \mathrm{i}I$ i $B + \mathrm{i}I$ imaju jedinstvenu po modulu najveću svojstvenu vrijednost λ_1, pa za njih metoda potencija konvergira: $\frac{|\lambda_2|}{|\lambda_1|} \approx 0.9844$.
- Inverzne iteracije konvergiraju (jako sporo) za matrice A i B jer postoji jedinstvena svojstvena vrijednost λ_1 najbliža $\tau = 1 + 2.5\mathrm{i}$: $\frac{|\lambda_1 - \tau|}{|\lambda_2 - \tau|} \approx 0.9927$.
- Inverzne iteracije konvergiraju (brzo) za matrice $A + \mathrm{i}I$ i $B + \mathrm{i}I$ jer postoji jedinstvena svojstvena vrijednost λ_1 najbliža $\tau = 1 + 2.5\mathrm{i}$: $\frac{|\lambda_1 - \tau|}{|\lambda_2 - \tau|} \approx 0.5990$.

Zadatak 7

Rješenje. Metoda potencija.
Početni vektor = $[1\ 1\ \ \ldots\ \ 1]$. Faktor konvergencije = 0.9844.



Zadatak 7

Rješenje. Inverzne iteracije. target = $1 + 2.5i$
Početni vektor = $[1\ 1\ \ldots\ 1]$. Faktor konvergencije = 0.5990.