Matematično fizikalni seminar

5.Naloga

Ana Marija Šolar

Vpisna številka: 28191078

UVOD

QR metoda:

S to metodo poiščemo lastne vrednosti simetrične matrike $A \in \mathbb{R}^{n\times n}$ s ponavljajočimi QR razcepi. Začnemo z izbrano matriko $A^{(0)} := A$, nato naredimo QR razcep, kateri je v k-ti iteraciji oblike: $A^{(k)} = Q^{(k)}R^{(k)}$, kjer je $Q^{(k)}$ ortogonalna matrika $(Q^{(k)T}Q^{(k)} = I)$, $R^{(k)}$ zgornje trikotna matrika. Naslednji korak je $A^{(k+1)} = R^{(k)}Q^{(k)}$, ker je A simetrična, ostane $A^{(k)}$ simetrična skozi vse iteracije. Ta postopek se ponavlja dokler $A^{(k)}$ ni skoraj diagonalna: $A^{(k)} \approx \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Lastne vrednosti matrike A so $\lambda_i \approx A_{ii}^{(k)}$.

Householderjeva metoda:

S to metodo pa našo matriko preoblikujemo v simetrično tridiagonalno matriko, katero nato obdelamo s QR metodo. Imamo matriko $A \in \mathbb{R}^{n\times n}$, izberemo en stolpec $\mathbf{x} = A_{(k+1):,k}$, kjer je $k = 0, 1, \ldots, n-3$, s katerim ustvarimo vektor:

$$\mathbf{e}_1 = \begin{bmatrix} \|\mathbf{x}\| \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$ in določimo Householderjev vektor: $$\mathbf{u} = \frac{\mathbf{x} - \mathbf{e}_1}{\|\mathbf{x} - \mathbf{e}_1\|}.$$

Nato izračunamo Householderjevo matriko: $H_k := I - 2\mathbf{u}\mathbf{u}^T$, s katero zgradimo novo matriko: $H^{(k)} = \begin{bmatrix} I_k & 0 & 0 \\ 0 & H_k & 0 \\ 0 & 0 & I \end{bmatrix}$. Nato posodobimo našo začetno matrika $A^{(k+1)} = H^{(k)} A^{(k)} H^{(k)}$, kjer je $H^{(k)}$ ortogonalna matrika, se simetričnost ohrani $(A^{(k+1)})^T = A^{(k+1)}$. Po vseh korakih nam ostane tridiagonalna matrika $T = A^{(n-2)}$, k je podobna začetni matriki, tj. ima enake lastne vrednosti.

NALOGA

Določi lastne vektorje in lastne vrednosti simetričnih matrik in primerjaj rezultate z vgrajenimi funkcijami. Uporabi iterativno QR metodo in razišči, koliko iteracij je potrebno za določeno natančnost. Kako se spremeni hitrost konvergence, če matriko prej preoblikujemo na tridiagonalno matriko?

REZULTATI

Matrika A1:

$$\begin{bmatrix} 1000. & 0.1 & 0.01 & 0.001 & 0.0001 & 0.00001 & 0.000001 \\ 0.1 & 1000. & 0.1 & 0.01 & 0.001 & 0.0001 & 0.00001 \\ 0.01 & 0.1 & 1000. & 0.1 & 0.01 & 0.001 & 0.0001 \\ 0.001 & 0.01 & 0.1 & 1000. & 0.1 & 0.01 & 0.001 \\ 0.0001 & 0.001 & 0.01 & 0.1 & 1000 & 0.1 & 0.01 \\ 0.00001 & 0.0001 & 0.001 & 0.01 & 0.1 & 1000. & 0.1 \\ 0.000001 & 0.00001 & 0.0001 & 0.001 & 0.01 & 0.1 & 1000. \end{bmatrix}$$

Z zgornjima metodama in eno vgrajeno metodo sem dobila naslednje rezultate za prvo matriko, kateri sem določila maksimum iteracij 1000000 in toleranco 1e-8.

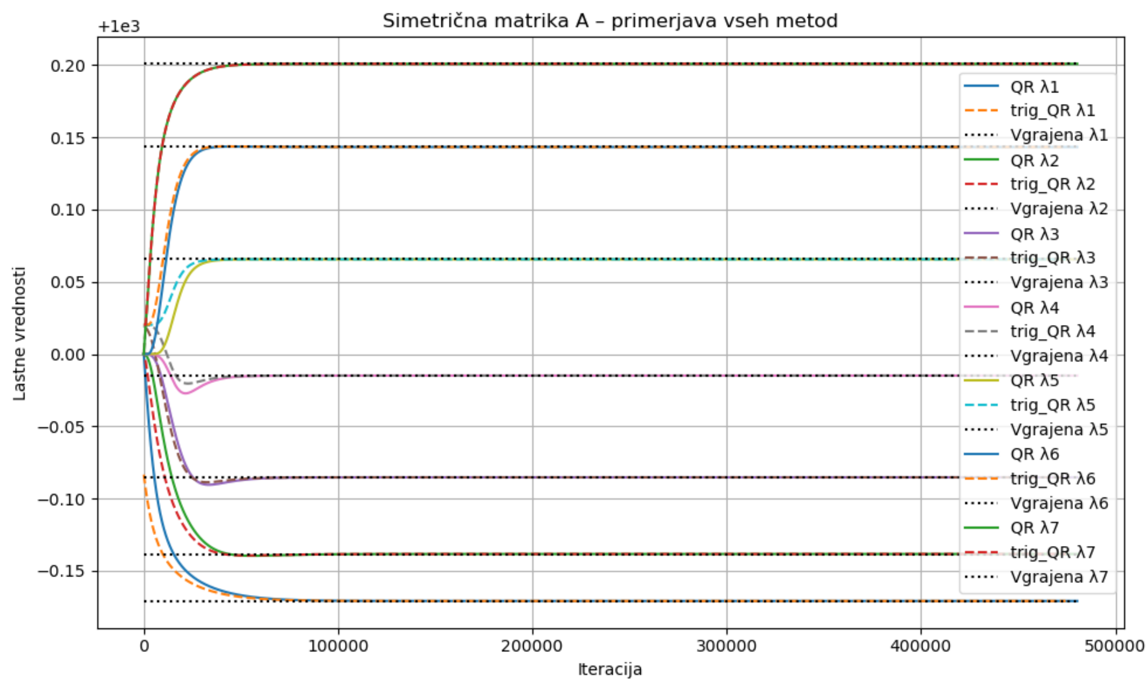
Lastne vrednosti:

Vgrajena Metoda	999.82908751	999.86159725	999.91465698	999.98506851	1000.065552	1000.14323324	1000.20080452
QR Metoda	999.82908751	999.86159726	999.91465698	999.98506851	1000.065552	1000.14323324	1000.20080452
QR s tridiago.	999.82908751	999.86159725	999.91465698	999.98506851	1000.065552	1000.14323324	1000.20080452

Primerjava metod:

	Čas[ms]	*Iteracije*
QR Metoda	39302	480422
QR s tridiagonalizacijo	42337	474507

Graf konvergenc:



Matrika A2:

$$\begin{bmatrix} 10^6 & 0.1 & 0.01 & 0.001 & 0.0001 & 0.00001 & 0.000001 \\ 0.1 & 10^6 & 0.1 & 0.01 & 0.001 & 0.0001 & 0.00001 \\ 0.01 & 0.1 & 10^6 & 0.1 & 0.01 & 0.001 & 0.0001 \\ 0.001 & 0.01 & 0.1 & 10^6 & 0.1 & 0.01 & 0.001 \\ 0.0001 & 0.001 & 0.01 & 0.1 & 10^6 & 0.1 & 0.01 \\ 0.00001 & 0.0001 & 0.001 & 0.01 & 0.1 & 10^6 & 0.1 \\ 0.000001 & 0.00001 & 0.0001 & 0.001 & 0.01 & 0.1 & 10^6 \end{bmatrix}$$

Pri tej matriki sem imela kar veliko težav s rezultati, saj je v vseh primerih bil dosežen maksimum iteracij ne glede na koliko tolerance sem imela. Na koncu sem se odločila za maksimum iteracij: 1e7 in maksimum toleranca: 1e-1.

Lastne vrednosti

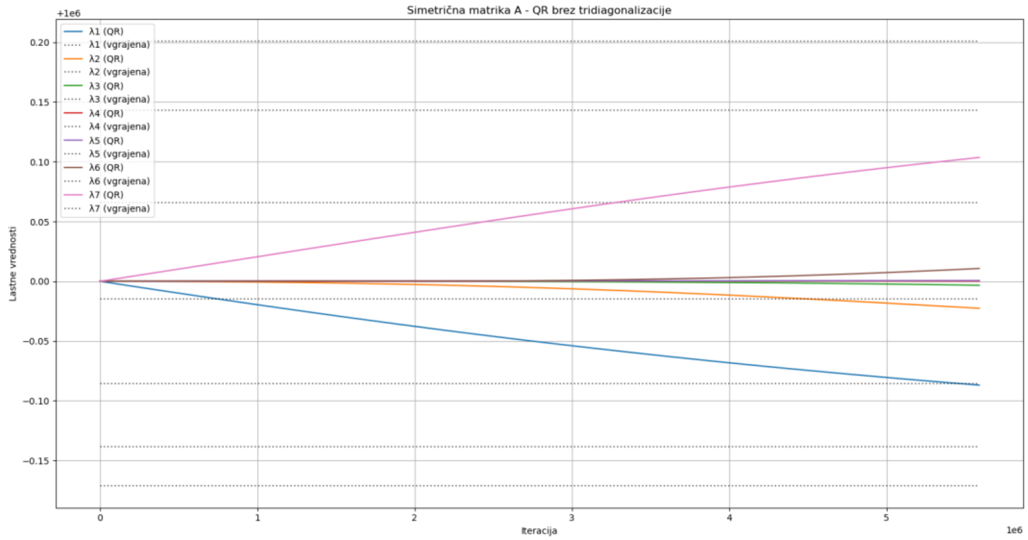
Vgrajena Metoda	999999.82908751	999999.86159725	999999.91465698	999999.98506851	1000000.065552	1000000.14323324	1000000.20080452
QR Metoda	999999.91311488	9999999.97740825	9999999.99664278	10000000.00014123	10000000.00062264	10000000.01070506	10000000.10362857
QR s tridiago. Metoda	9999999.86919841	9999999.93786023	9999999.99098803	10000000.01471178	10000000.02135243	10000000.0419538	10000000.12367863

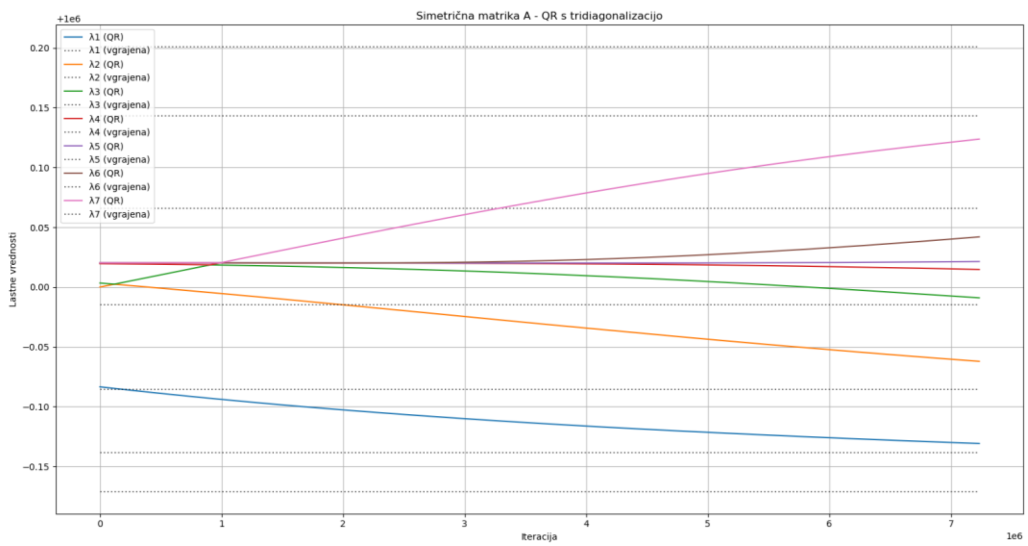
```
--- Simetrična matrika A ---
Vgrajene lastne vrednosti: [ 999999.82908751  999999.86159725  999999.91465698  999999.98506851
 1000000.065552   1000000.14323324 1000000.20080452]
Čas vgrajene metode: 0.000 ms
QR brez tridiagonalizacije (5589148 iteracij): [ 999999.91311488  999999.97740825  999999.99664278 1000000.00014123
 1000000.00062264 1000000.01070506 1000000.10362857]
Čas QR brez tridiagonalizacije: 541990.550 ms
QR s tridiagonalizacijo (7232105 iteracij): [ 999999.86919841  999999.93786023  999999.99098803 1000000.01471178
 1000000.02135243 1000000.0419538  1000000.12367863]
Čas QR s tridiagonalizacijo: 711151.108 ms
```

Primerjava metod:

	Čas[ms]	*Iteracije*
QR Metoda	541990	5589148
QR s tridiagonalizacijo	711151	7232105

Pri grafih sem imela veliko težav s shranjevanjem saj je bil program velikokrat neodziven. Uspelo mi je posamezna grafa.f.





Preko grafov vidimo, da moje metode ne dosežejo pravilne lastne vrednosti matrike, katere smo dobili preko vgrajene metode v nekem normalnem času. Poleg tega problema vidimo pri iteracijah, da je bila tridiagonalna metoda počasnejša. Ne vem od kej ta problem izvira.