

### 3. naloga: Lastne vrednosti in lastni vektorji

Gregor Žunič, 28191076

Enodimenzionalni linearni harmonski oscilator (delec mase  $m$  s kinetično energijo  $T(p) = p^2/2m$  v kvadratičnem potencialu  $V(q) = m\omega^2 q^2/2$ ) opišemo z brezdimenzijsko Hamiltonovo funkcijo

$$H_0 = \frac{1}{2} (p^2 + q^2) ,$$

tako da energijo merimo v enotah  $\hbar\omega$ , gibalne količine v enotah  $(\hbar m\omega)^{1/2}$  in dolžine v enotah  $(\hbar/m\omega)^{1/2}$ . Lastna stanja  $|n\rangle$  nemotenega Hamiltonovega operatorja  $H_0$  poznamo iz osnovnega tečaja kvantne mehanike [Strnad III]: v koordinatni reprezentaciji so lastne valovne funkcije

$$|n\rangle = (2^n n! \sqrt{\pi})^{-1/2} e^{-q^2/2} \mathcal{H}_n(q) ,$$

kjer so  $\mathcal{H}_n$  Hermitovi polinomi. Lastne funkcije zadoščajo stacionarni Schrödingerjevi enačbi

$$H_0 |n^0\rangle = E_n^0 |n^0\rangle$$

z nedegeneriranimi lastnimi energijami  $E_n^0 = n + 1/2$  za  $n = 0, 1, 2, \dots$ . Matrika  $\langle i | H_0 | j \rangle$  z  $i, j = 0, 1, 2, \dots, N-1$  je očitno diagonalna, z vrednostmi  $\delta_{ij}(i + 1/2)$  po diagonalni. Nemoteni Hamiltonki dodamo anharmonski člen

$$H = H_0 + \lambda q^4 .$$

Kako se zaradi te motnje spremenijo lastne energije? Iščemo torej matrične elemente  $\langle i | H | j \rangle$  motenega Hamiltonovega operatorja v bazi *nemotenih* valovnih funkcij  $|n^0\rangle$ , kar vemo iz perturbacijske teorije v najnižjem redu. Pri računu si pomagamo s pričakovano vrednostjo prehodnega matričnega elementa za splošno koordinato

$$q_{ij} = \langle i | q | j \rangle = \frac{1}{2} \sqrt{i+j+1} \delta_{|i-j|,1} ,$$

ki, mimogrede, uteleša izbirno pravilo za električni dipolni prehod med nivoji harmonskega oscilatorja. V praktičnem računu moramo seveda matriki  $q_{ij}$  in  $\langle i | H | j \rangle$  omejiti na neko končno razsežnost  $N$ .

*Naloga:* Z diagonalizacijo poišči nekaj najnižjih lastnih vrednosti in lastnih valovnih funkcij za moteno Hamiltonko  $H = H_0 + \lambda q^4$  ob vrednostih parametra  $0 \leq \lambda \leq 1$ . Rešujemo torej matrični problem lastnih vrednosti

$$H |n\rangle = E_n |n\rangle .$$

Nove (popravljenе) valovne funkcije  $|n\rangle$  so seveda linearna kombinacija starih (nemotenih) valovnih funkcij  $|n^0\rangle$ . Matrike velikosti do  $N = 3$  ali  $N = 4$  lahko za silo diagonaliziramo peš; za diagonalizacijo pri večjih  $N$  uporabi enega ali več numeričnih postopkov, na primer rutine `tred2` in `tqli` iz zbirke Numerical Recipes ali iz kakega drugega vira (npr Python). Vsaj enega izmed postopkov izvedi 'ročno' (sprogramiraj, uporabi izvirno kodo). Preveri, da v limiti  $\lambda \rightarrow 0$  velja  $E_n \rightarrow E_n^0$  (če ne velja, je verjetno nekaj narobe s programom). Razišči, kako so rezultati odvisni od razsežnosti  $N$  matrik  $H_0$  oziroma  $q^4$ . Kakšna je konvergenca lastnih vrednosti pri velikih  $N$ ?

Namesto da računamo matrične elemente  $q_{ij} = \langle i | q | j \rangle$  in perturbacijsko matriko razumemo kot  $[q_{ij}]^4$ , bi lahko računali tudi matrične elemente kvadrata koordinate

$$q_{ij}^{(2)} = \langle i | q^2 | j \rangle$$

in motnjo razumeli kot kvadrat ustrezne matrike,

$$\lambda q^4 \rightarrow \lambda \left[ q_{ij}^{(2)} \right]^2 ,$$

ali pa bi računali matrične elemente četrte potence koordinate

$$q_{ij}^{(4)} = \langle i | q^4 | j \rangle$$

in kar to matriko razumeli kot motnjo,

$$\lambda q^4 \rightarrow \lambda \left[ q_{ij}^{(4)} \right] .$$

Kakšne so razlike med naštetimi tremi načini izračuna lastnih vrednosti in funkcij? Pri računu poleg enačbe za  $\langle i | q | j \rangle$  uporabi še enačbi

$$\langle i | q^2 | j \rangle = \frac{1}{2} \left[ \sqrt{j(j-1)} \delta_{i,j-2} + (2j+1) \delta_{i,j} + \sqrt{(j+1)(j+2)} \delta_{i,j+2} \right]$$

ter

$$\begin{aligned} \langle i | q^4 | j \rangle = \frac{1}{2^4} \sqrt{\frac{2^i i!}{2^j j!}} \left[ \right. & \delta_{i,j+4} + 4(2j+3) \delta_{i,j+2} + 12(2j^2+2j+1) \delta_{i,j} \\ & \left. + 16j(2j^2-3j+1) \delta_{i,j-2} + 16j(j^3-6j^2+11j-6) \delta_{i,j-4} \right] , \end{aligned}$$

ki ju ni težko izpeljati iz rekurzijskih zvez za Hermitove polinome.

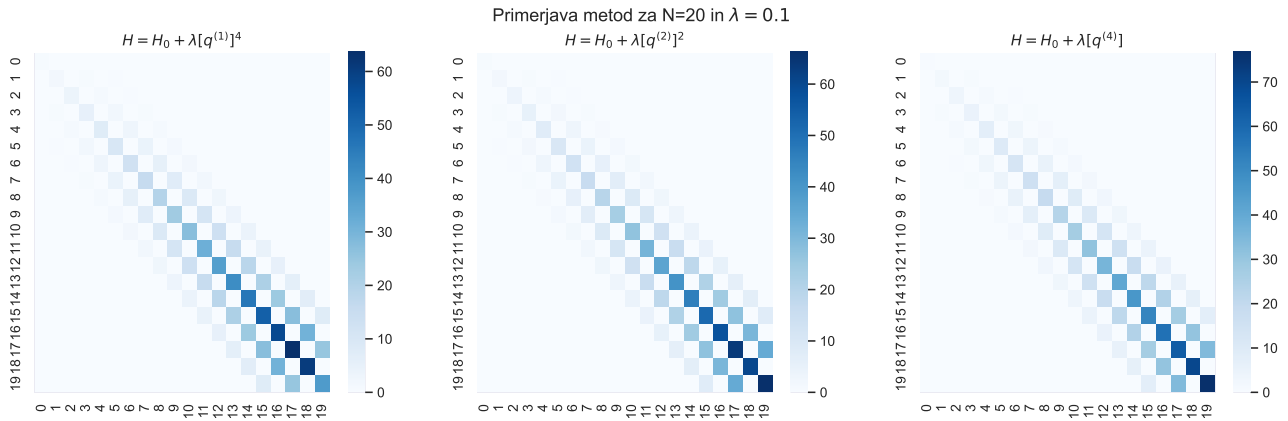
*Dodatna naloga:* Poišči še nekaj najnižjih lastnih energij in lastnih funkcij za problem v potencialu z dvema minimumoma

$$H = \frac{p^2}{2} - 2q^2 + \frac{q^4}{10} .$$

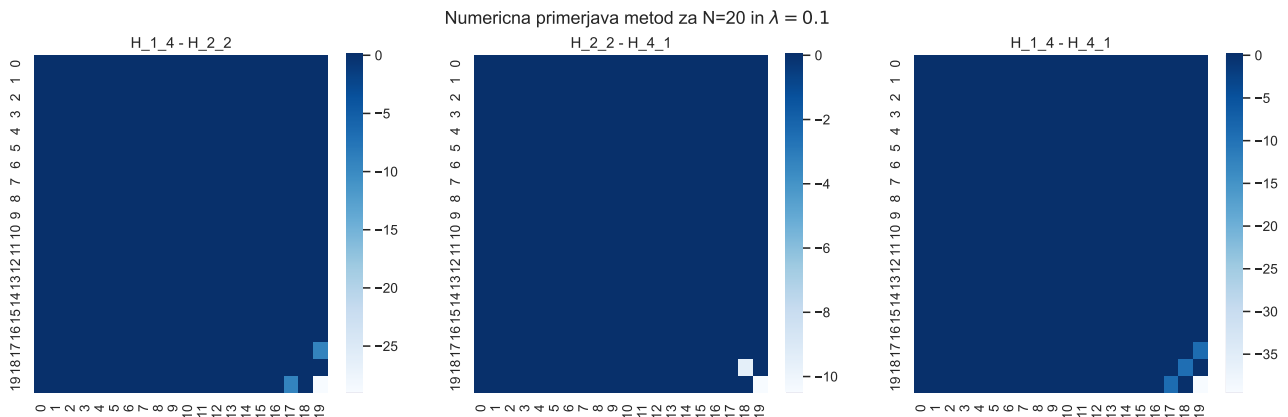
# 1 Reševanje

Kar je treba najprej storiti, je seveda implementacija vseh generacijskih funkcij za matrike. Trivialno vendar je pomembno je, če implementiramo v Pythonu, da matrike na potenciramo na navaden Pythonista način (`**4`), vendar uporabimo funkcijo `linalg.matrix_power` (zgolj komentar na napako, ki mi je vzela dobro uro!).

Matriko  $H$  sestavimo tako, da seštejemo  $H_{ii} = \text{diag}(1/2+i) + \lambda q_{ij}$ , kjer je  $q_{ij}$  matrika, ki jo izračunamo s pomočjo formul v prejšnjem poglavju.



Slika 1: Prikaz matrik za različne načine izračuna matrike  $H$ .

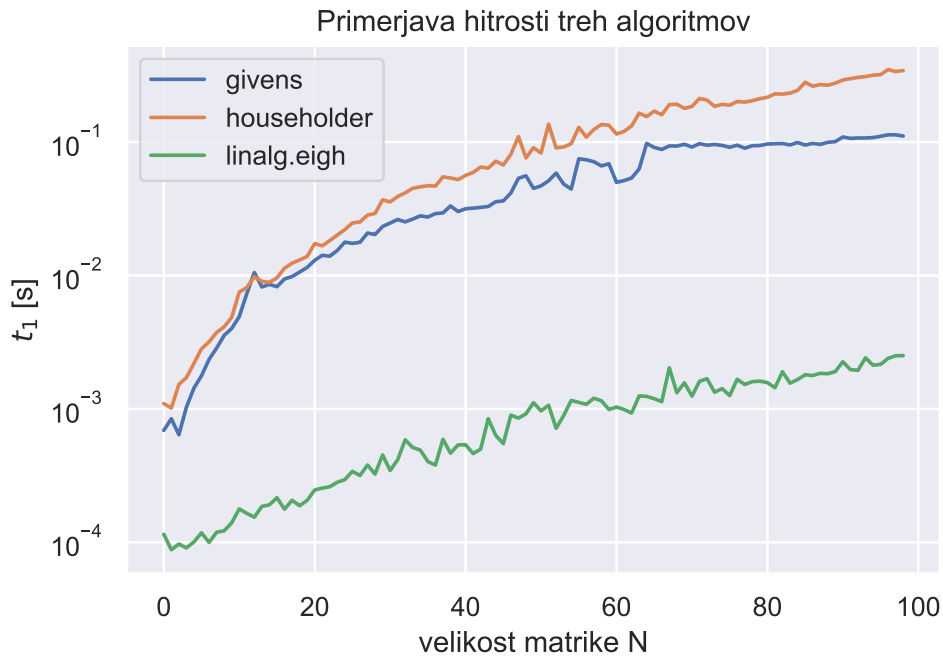


Slika 2: Prikaz razlike med matrikami za različne načine izračuna matrike  $H$ .

Ko generiramo matrike za vse tri možnosti, je dobro vprašanje katera metoda je zares najboljša za diagonalizacijo. Izkaže se (kot je vidno na grafih), da so vse vrednosti zelo blizu, razen desno spodaj (pri velikih vrednostih matrike). Kar lahko storimo je, da zgeneriramo matriko, ki je malce večja kot naša priljubljena dimenzija, ki jo hočemo diagonalizirati oziroma dobiti lastne vektorje. Po tem postopku se zares znebimo razlik med različnimi generacijskimi funkcijami, ker so matrike po vseh treh postopkih enake. Časovna zahtevnost generacij matrik (ki je v tem primeru popolnoma zanemarljiva, zaradi kasnejših korakov) je najbolj učinkovita pri primeru  $[q^{(4)}]$ , ker nimamo nobenih množenj matrik (množenje matrik ima zahtevnost  $n^2$ , kar za zelo velike matrike lahko pomeni težave). Od zdaj naprej bom zaradi natančnosti generiral matrike, ki bodo večje od željene natančnosti, izbira metoda tako potem ni pomembna.

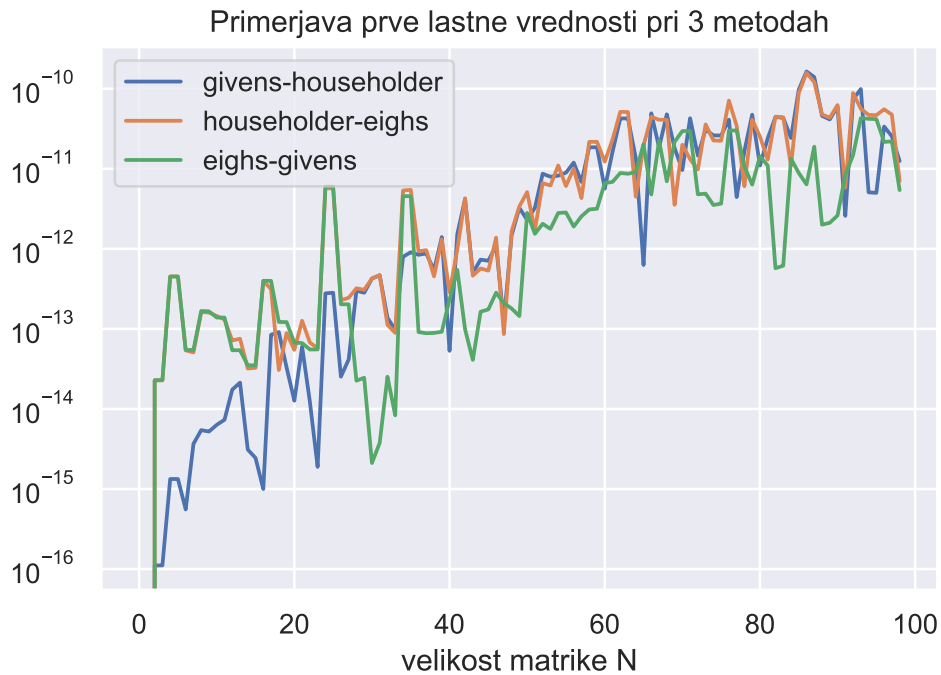
## 1.1 Primerjava metod dekompozicije

Sedaj lahko implementiramo tudi svoje algoritme za diagonalizacijo matrik oziroma izračun lastnih vektorjev. Pričakovano, se izkaže, da je vgrajena funkcija najbolj časovno učinkovita, ker je implementirana v C.



Slika 3: Hitrost izračuna lastnih vrednosti/vektorjev z 3 različnimi metodami.

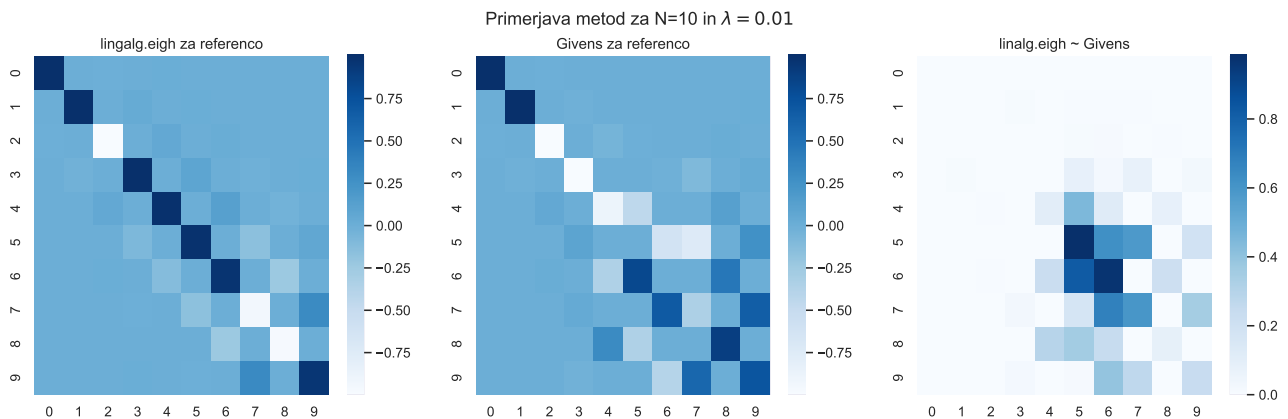
Kot je vidno na grafu, je vgrajena metoda superiorna na celotnem območju velikosti matrik  $N$ . Kar je skupno vsem algoritmom je, da časovna zahtevnost narašča z  $O(n^3)$ . Pri moji implementaciji, je Givensova metoda kar precej hitrejša (ordinata je v logaritemski skali). Sklepam, da taka razlika nastane zaradi večjega števila posameznih korakov, ki jih storim pri Householderjevi metodi. Katera metoda je boljša, se verjetno pokaže na specialnih primerih matrik, ki pa v mojem primeru ne nastopajo. Zato lahko zaključim, da je za primer iskanja lastnih vektorjev in vrednosti pri tej nalogi precej boljša Givensov algoritem.



Slika 4: Napaka prve lastne vrednosti.

Lastne vrednosti, ki jih metode izračunajo so enake. Na grafu je primerjava med lastnimi vrednostmi s posameznimi metodami. Opazimo lahko, da so zelo blizu ter to ni več napaka metod, vendar čisto numerična napaka pri računanju (kar pomeni, da bi z več koraki lahko dobili še večjo natančnost).

Zanimiva primerjava je tudi med lastnimi vektorji, ki jih dobimo z posameznimi metodami. Iz smisla implementacije je malo zakomplicirana primerjava, ker je vrstni red vektorjev drugačen (seveda vsak algoritem lahko v principu poda drugačen vrstni red lastnih vrednosti). Kar moramo storiti, je da pogrupiramo skupaj pravilne lastne vrednosti in vektorje.

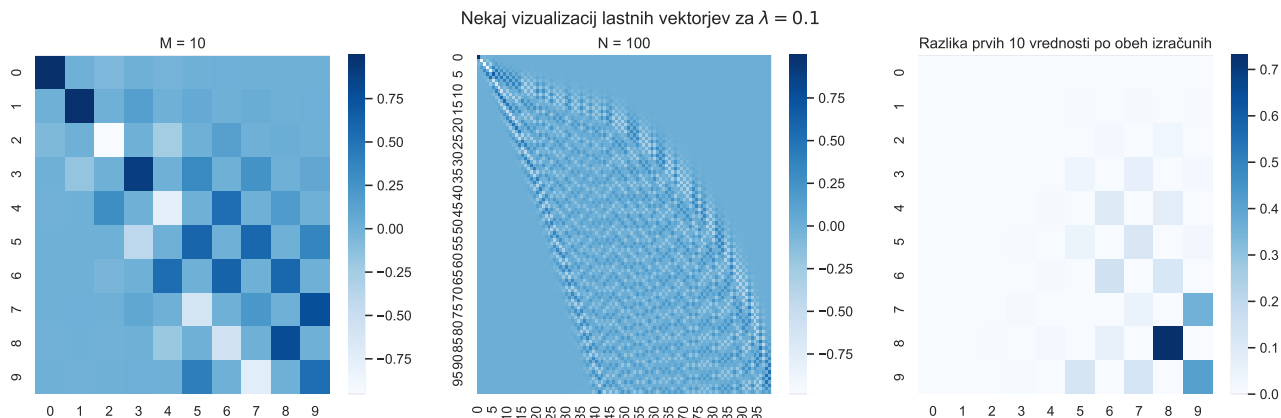


Slika 5: Primerjava matrik lastnih vektorjev (prehodnih matrik).

Opazimo lahko, da čeprav so lastne vrednosti zelo podobne (v tem primeru praktično enake do zelo velika natančnosti), se lastni vektorji precej razlikujejo. Mera, ki sem jo narisal je zgolj, absolutna razlika med absolutnima vrednostma dveh enako-ležečih celic.

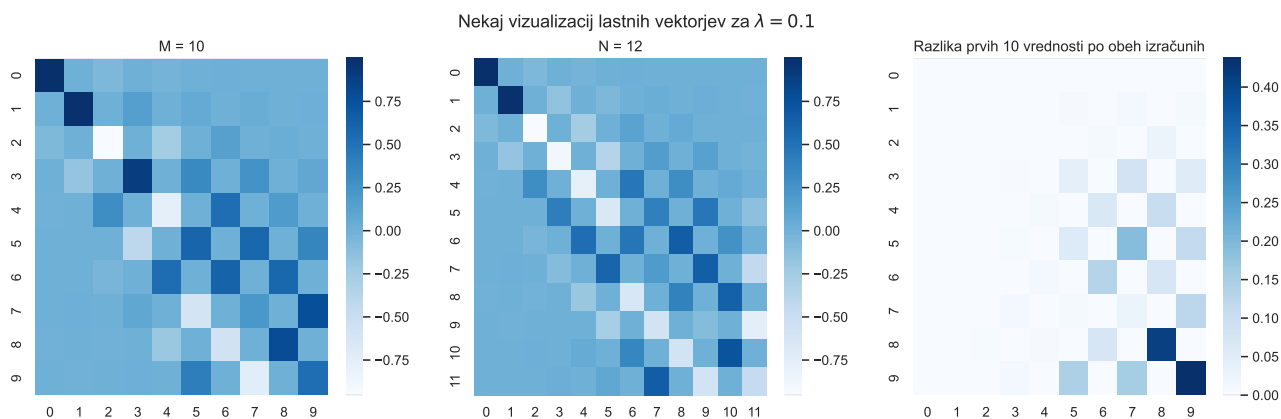
## 1.2 Lastni vektorji

Sedaj lahko narišem še nekaj lastnih vektorjev za nekaj parametrov.

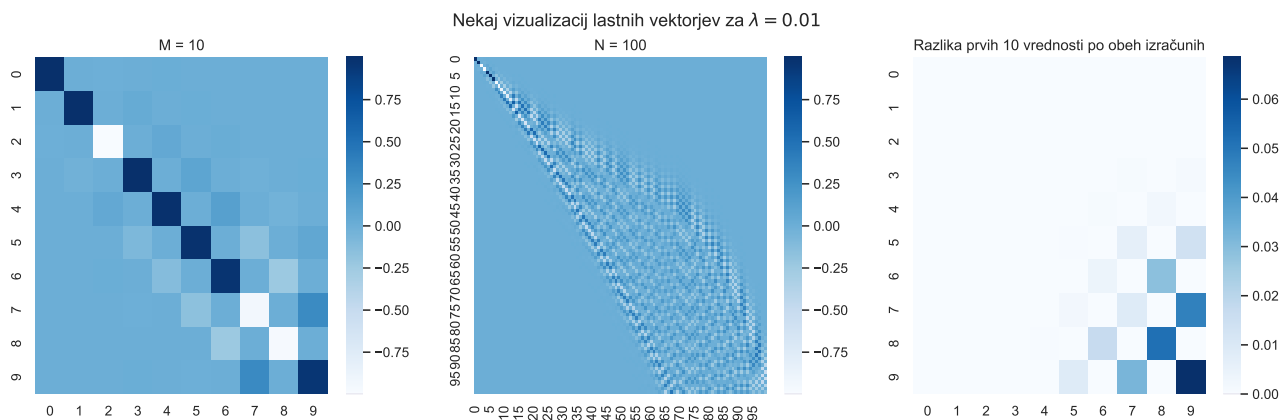


Slika 6: Lastni vektorji za  $N = 100$ ,  $\lambda = 0.1$ .

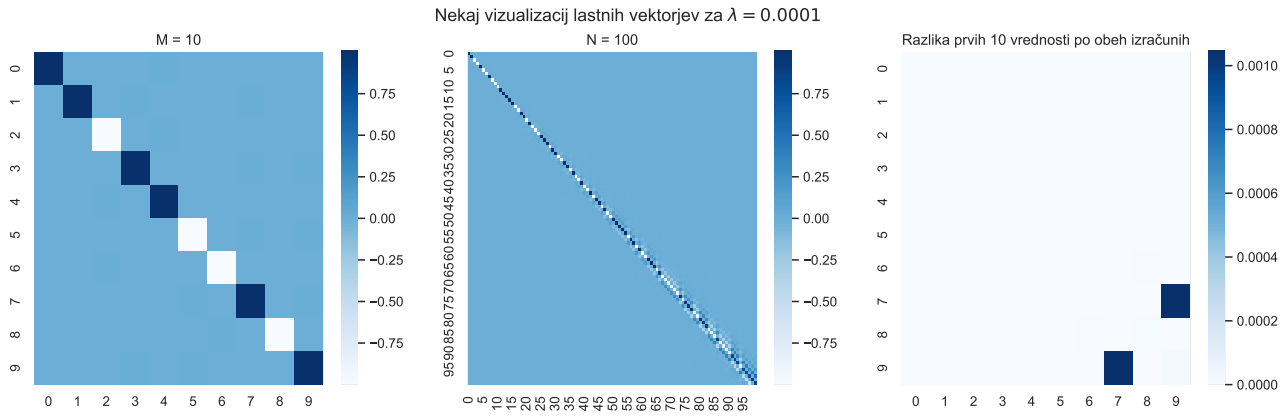
Lastni vektorji pri velikih vrednostih  $N$  postanejo precej kaotični, vendar se pri majhnih vrednostih  $\lambda$  približujejo diagonalni matriki.



Slika 7: Lastni vektorji za  $N = 12$ ,  $\lambda = 0.1$ .



Slika 8: Lastni vektorji za  $N = 100$ ,  $\lambda = 0.01$ .

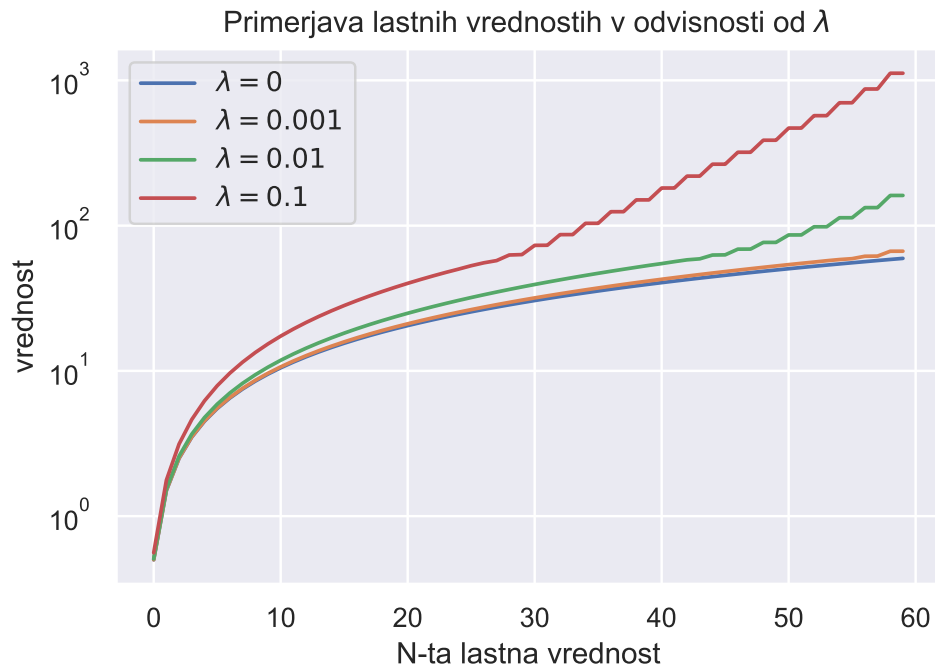


Slika 9: Lastni vektorji za  $N = 100$ ,  $\lambda = 0.0001$ .

Iz grafov zgoraj lahko opazimo, da lastni vektorji zares konvergirajo proti  $E_n^0$ , kar je seveda v skladu s teorijo. To, da so nekatere vrednosti po diagonali negativne, nima pravzaprav nobenega fizikalnega ozadja, vendar je samo posledica algoritma. Kar je važno je, da so vektorji med seboj ortogonalni.

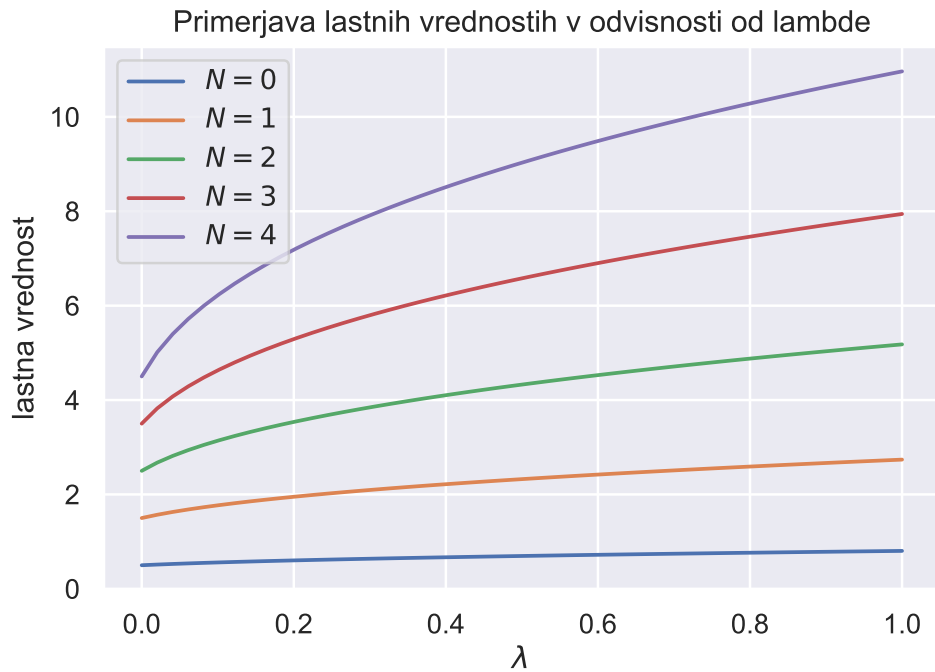
Iz grafov tudi lahko opazimo razliko med matriko izračunano direktno z majhno vrednostjo  $N = M$ , ki jo primerjamo z matriko, ki jo izračunamo z veliko vrednostjo  $N$  in skrčimo na  $M$ . Bolj pravilna matrika je seveda velika matrika, ki jo skrčimo ker bi morali v teoriji vzeti neskončno veliko matriko ter jo nato diagonalizirati. Ta analiza je zanimiva, ker nam pokaže, da tudi veliki členi vplivajo na vrednost lastne vektorje pri majhnih  $n$ .

Poleg lastnih vektorjev je zanimiv tudi graf lastnih vrednosti v odvisnosti od  $\lambda$ . Opazimo lahko, da je rast precej eksponentna.



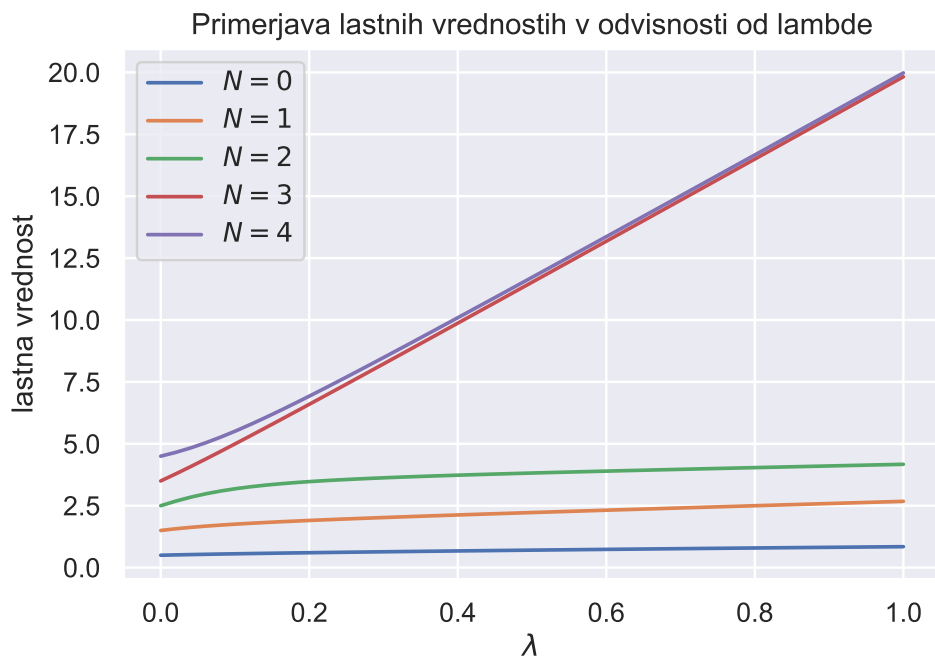
Slika 10: Lastni vrednosti za različne parametre  $\lambda$ .

Bolj zanimiv je obraten graf z odvisnostjo od  $\lambda$  in  $n$ -ta lastna vrednost kot parameter, da zares preverimo konvergenco pri  $\lambda \rightarrow 0$ .



Slika 11: Različne lastne vrednosti v odvisnosti od  $\lambda$ . Velikost začetne matrike je  $M = 50$ .

Iz grafa lahko opazimo, da gre za neke vrste korensko naraščanje lastnih vrednosti v odvisnosti od  $\lambda$ . Začetne vrednosti tudi zelo lepo sledijo modelu pri  $\lambda = 0$  (0.5, 1.5, 2.5, ...). Da dobimo lep graf in pravilno obnašanje moramo vzeti tudi dovolj veliko število začetne matrike  $M$ . Če na primer vzamemo  $M = n$ , nam vrednosti precej divergirajo, kot je vidno na grafu spodaj.



Slika 12: Različne lastne vrednosti v odvisnosti od  $\lambda$ . Velikost začetne matrike je  $M = 5$ . Vrednosti so napačne in precej divergirajo.



### 1.3 Komentar

Lastni vektorji imajo precej kaotične oblike ter lepo konvergirajo proti osnovni verziji (ko nimamo nobene motnje). Rezultati lastnih vrednosti pri mojih metodah se ujemajo z vgrajeno `Scipy` funkcijo, kar nakazuje, da sem pravilno implementiral metode QR dekompozicij. Dodatne naloge tokrat nisem naredil :/