

03 - LASTNE VREDNOSTI IN LASTNI VEKTORJI

Matematično-fizikalni praktikum, oktober 2023
Luka Skeledžija, 28201079

1. Uvod

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 so

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

kjer so \mathcal{H}_n Hermitovi polinomi. Lastne funkcije zadoščajo stacionarni Schrodingerjevi 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 diagonalna, z vrednostmi $\delta_{ij}(i + 1/2)$ po diagonalni. Nemoteni Hamiltonki dodamo anharmonski člen

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

Matriko s tem potencialom v nadaljevanju navajamo s H . Tedaj matriko zapišemo kot

$$\langle i | H | j \rangle = \langle i | H_0 | j \rangle + \lambda \langle i | q^4 | j \rangle$$

Za izračun drugega člena si lahko pomagamo z naslednjimi zvezami

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

Sedaj rešujemo matrični sistem

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

katerega bomo reševali v bazi lastnih funkcij harmonskega oscilatorja $|i\rangle$:

$$|n\rangle = \sum_{i=0}^N c_i |i\rangle$$

Metodo lahko nato prenesemo tudi na drugačne potenciale. V dodatku je analiziran še kvantni harmonski oscilator z dvema minimumoma, katerega H zapišemo kot

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

Podobno kot prej lahko matriko zapišemo kot

$$\langle i|H|j\rangle = \langle i|H_0|j\rangle - \frac{5}{2}\langle i|q^2|j\rangle + \frac{1}{10}\langle i|q^4|j\rangle$$

Matriko s tem potencialom v nadaljevanju navajamo s H_2 .

2. Naloga

Z diagonalizacijo poišči nekaj najnižjih lastnih vrednosti in lastnih valovnih funkcij za moteno Hamiltonko $H = H_0 + \lambda q^4$ ob vrednostih parametra $0 \leq \lambda \leq 1$. 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 $E_n \rightarrow E_n^0$. 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, $\lambda q^4 \rightarrow \lambda [q_{ij}^{(2)}]^2$, 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, $\lambda q^4 \rightarrow \lambda [q_{ij}^{(4)}]$. Kakšne so razlike med naštetimi tremi načini izračuna lastnih vrednosti in funkcij?

Dodatna naloga: Poišči še nekaj najnižjih lastnih energij in lastnih funkcij za problem v potencialu z dvema minimumoma $H = \frac{p^2}{2} - 2q^2 + \frac{q^4}{10}$.

3. Princip reševanja

Reševanja naloge se lotimo v dveh delih / z dvema ključnima elementoma. Najprej s pomočjo perturbacijske teorije generiramo matrike, ki bodo nastopale v matrični enačbi. Potrebujemo torej generator ustreznih matrik. V drugem delu pa moramo ta sistem še rešiti. Za to bomo potrebovali program, ki zna dane matrike kar se da učinkovito diagonalizirati in tako rešiti matrično enačbo.

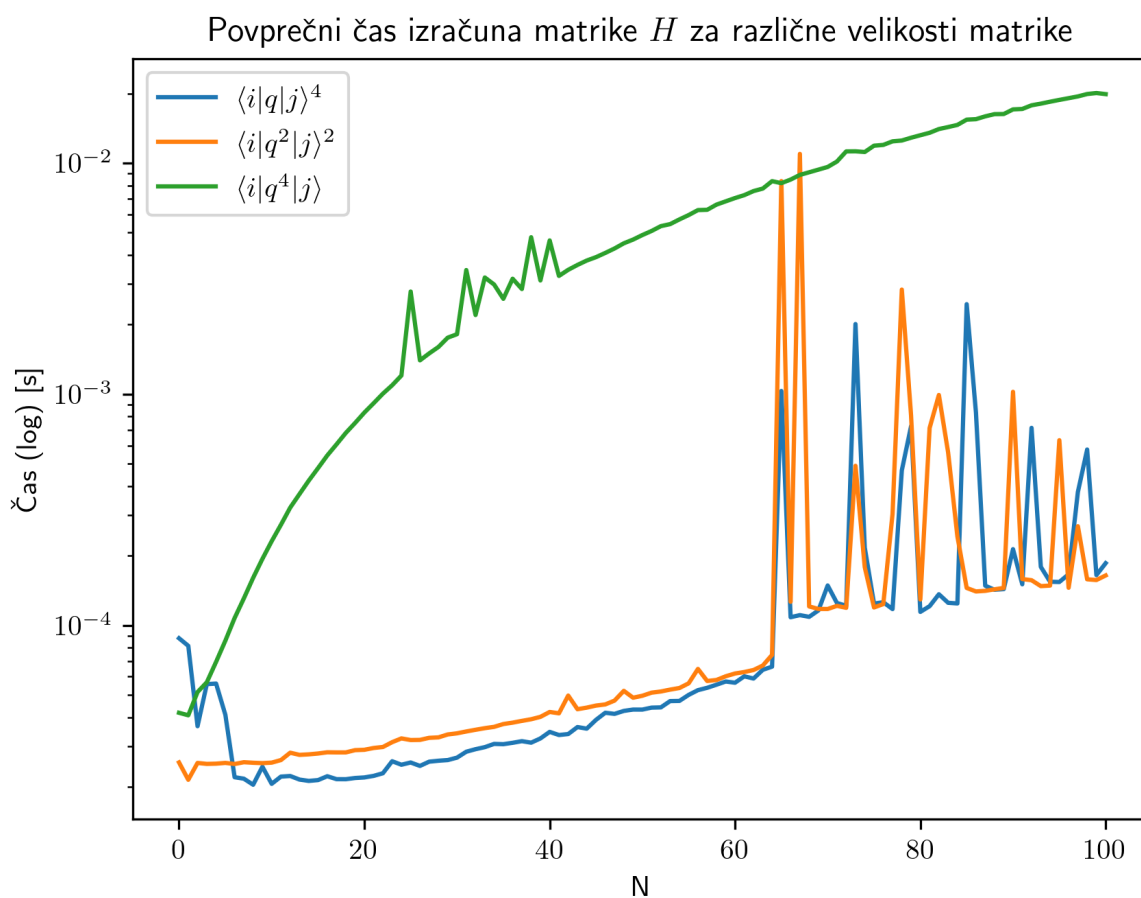
Sledijo štiri poglavja:

1. S pomočjo perturbacijske teorije v 1. redu poiščemo in generiramo nekaj skupnih matričnih Hamiltonovih operatorjev velikosti N . Analiziramo učinkovitost

- uporabljenih metod za različne interpretacije motnje in se odločimo za najučinkovitejšo.
- Primerjamo različne metode numerične diagonalizacije in se odločimo za najučinkovitejšo.
 - Z numerično diagonalizacijo rešimo matrično enačbo in primerjamo rezultate s harmonskim oscilatorjem.
 - Dodatek: Analiziramo še oscilator v potencialu z dvema minimumoma.

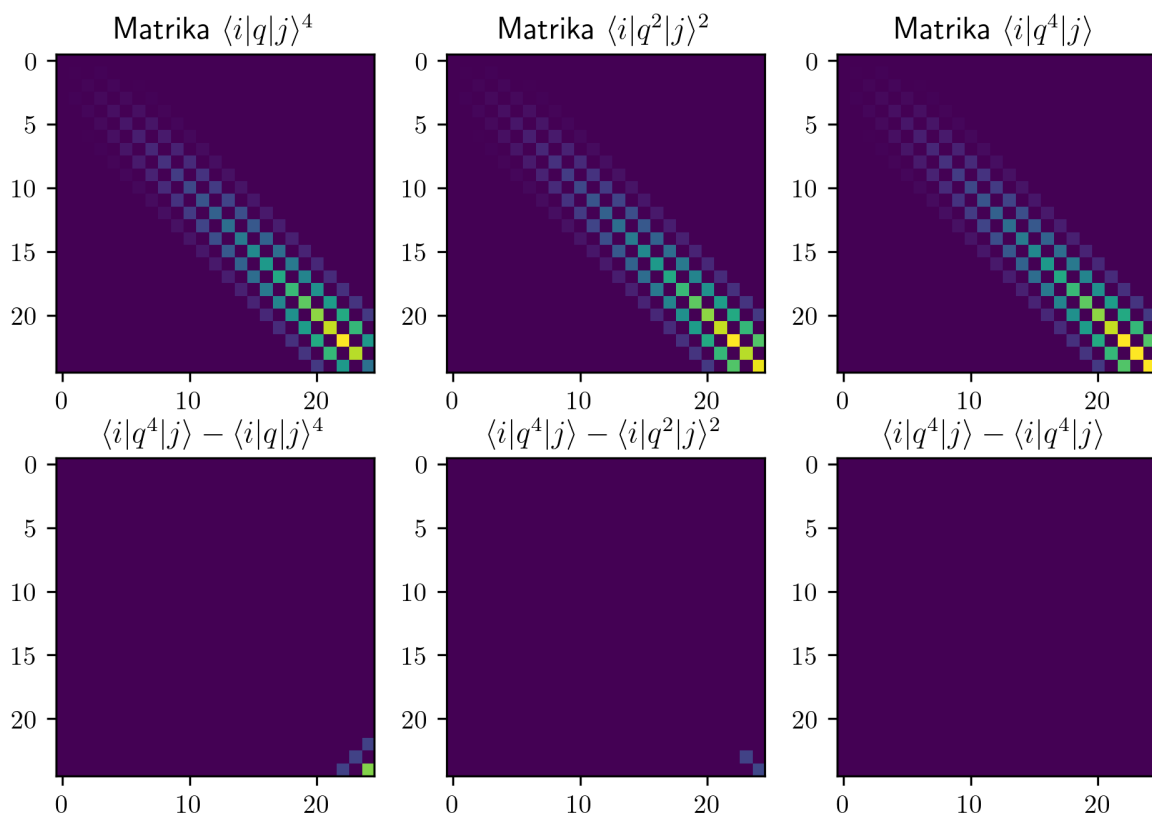
4. Hamiltonov matrični operator za an-harmonski oscilator

Za implementacijo sledimo formulam navedenim v uvodu. Perturbacijski prispevek anharmonskega potenciala si lahko, kot že prej navedeno, interpretiramo na 4 različne načine. Načine primerjamo po hitrosti izračuna.



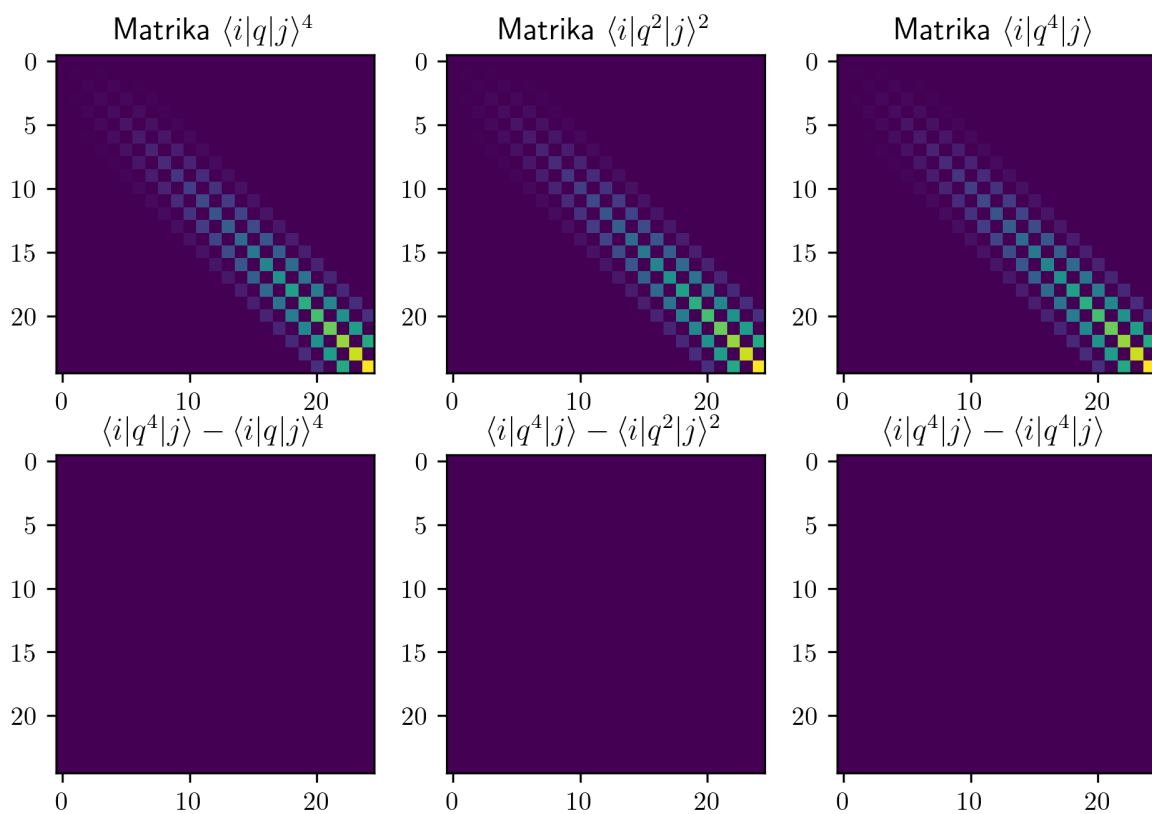
Ugotovimo, da je hitrost izračunavanja največja v prvih dveh primerih, saj uporabljamo knjižnico `numpy`. Za izračun fakultet v zadnjem primeru (zelen graf) uporabljamo dvakrat gnezdeno `for` zanko implementirano v Pythonu. Vendar zadnji princip izračuna ni brez prednosti. Kot opazimo v nadaljevanju, moramo za natančen izračun celotne perturbacijske matrike velikosti 25×25 v prvih dveh primerih, kjer matrike potenciramo, začeti z matrikami, ki so večje kot 25×25 .

Matrike velikosti 25×25



Primerjava 1: Vizualizacija in primerjava vrednosti v matrikah ($N=25$) za različne interpretacije operatorja potenciala.

$\langle i|q^4|j\rangle : 25 \times 25$, ostale: 28×28 in odrezane



Primerjava 2: Če pri matričnem množenju uporabimo večje matrike ($N=28$) in jih nato odrežemo, rezultat konvergira k vrednostim $\langle i|q^4|j\rangle$.

V nadaljevanju se odločimo za uporabo formule $\langle i|q|j\rangle^4$ z upoštevanjem dejstva, da bomo vedno uporabili za 3 dimenzije večje matrike in po potenciranju odvečne dimenzije odrezali. Kljub dodatnim dimenzijam je izračun še vedno za skoraj 1 red velikosti hitrejši.

5. Numerična diagonalizacija matrik

Za reševanje matrične enačbe sedaj potrebujemo učinkovit postopek za diagonalizacijo matrik. V primerjavo bomo vzeli diagonalizacijo po Jacobijevi metodi v lastni implementaciji (Python/numpy in C++), numpy metodo `numpy.linalg.eig` in metodo `Eigen::SelfAdjointEigenSolver` iz paketa Eigen za C++.

Metode bomo testirali s pomočjo generiranih naključnih matrik velikosti 2×2 do 100×100 . Primerjali bomo hitrost in natančnost diagonalizacije (s pomočjo kvadrata ne-diagonalne Frobeniusove norme $P_M = \|M\|_F^2 - \sum_{i=1}^n M_{ii}^2$).

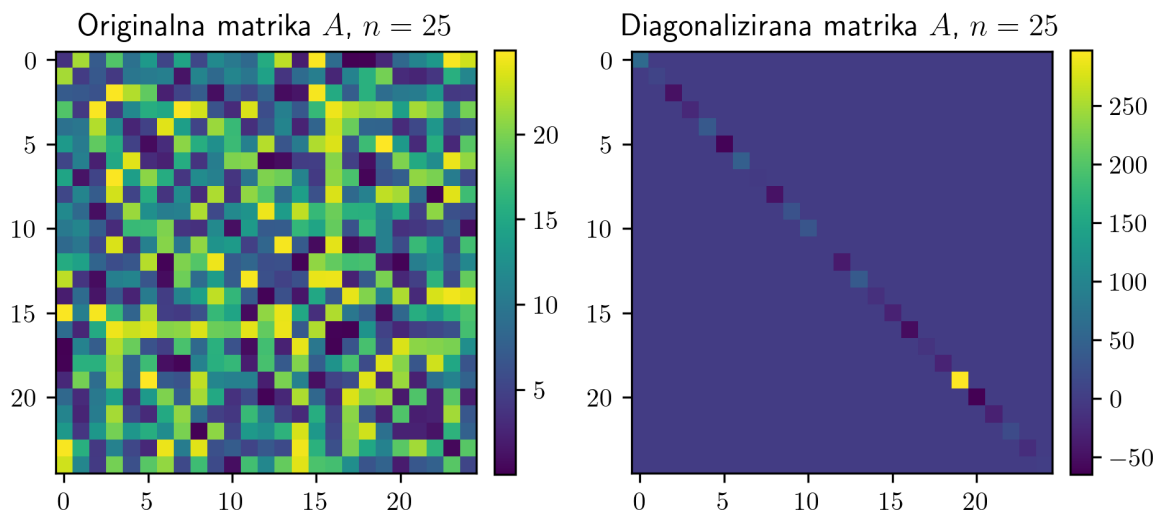
Jacobijev algoritem

Za primerjavo z metodami v programskih paketih bomo "na roke" implementirali [Jacobijev algoritem za izračun lastnih vrednosti in lastnih vektorjev](#). To storimo tako v "enostavnem" programskem jeziku `Python` (z uporabo knjižnice `numpy`, saj je to najbolj realen praktičen primer) in v "hitrem" programskem jeziku `C++`.

Gre za iterativno metodo, ki jo lahko skiciramo v naslednjih korakih:

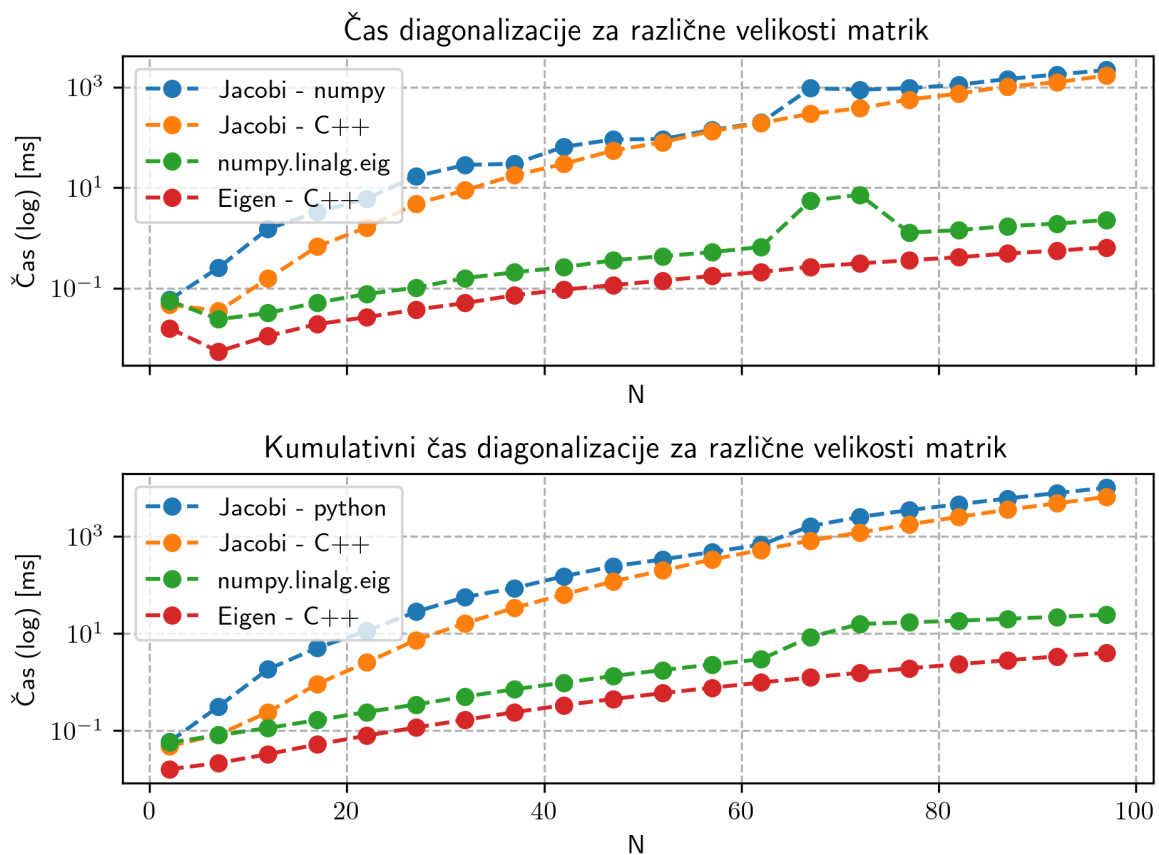
1. Imamo simetrično matriko S in Givensovo rotacijsko matriko $G = G(i, j, \theta)$. Tedaj je $S' = G^\top S G$ simetrična in podobna S .
2. (Zaradi optimizacije) si izberemo po absolutni vrednosti največji člen S_{ij} , ki ga imenujemo `pivot`. Sestavimo tako rotacijsko matriko G , da je $S'_{ij} = S'_{ji} = 0$. To ponovimo $N = n(n-1)/2$ -krat (n je velikost matrike in N število ne-diagonalnih členov). Če na matriko apliciramo N Givensovih/Jacobijevih rotacij, takšno skupinsko operacijo definiramo kot `sweep`.
3. Ponovimo poljubno mnogo operacij `sweep`, da dosežemo željeno numerično natančnost (P). Natančnost merimo kot $P = \|S'\|_F^2 - \sum_{i=1}^n S_{ii}^2$.

Implementacijo testiramo in ugotovimo, da ustrezno diagonalizira matriko.

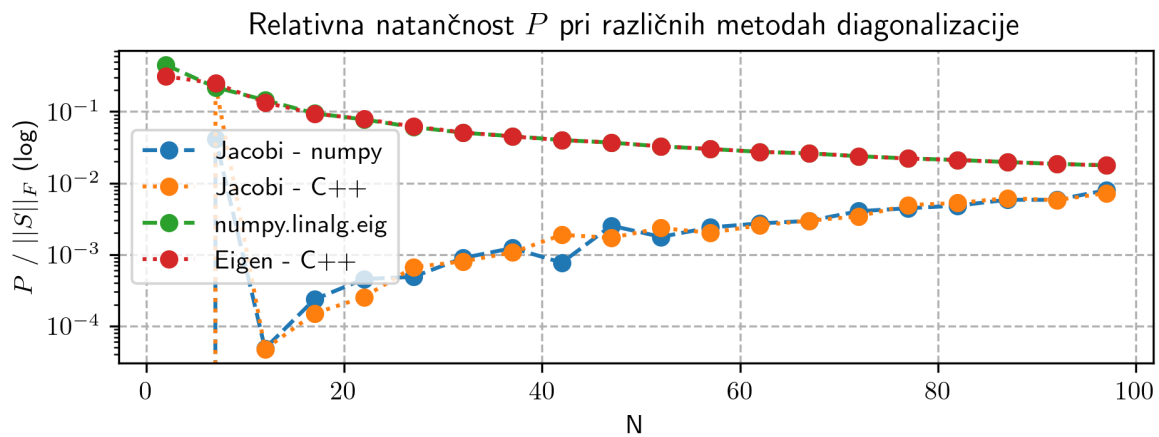
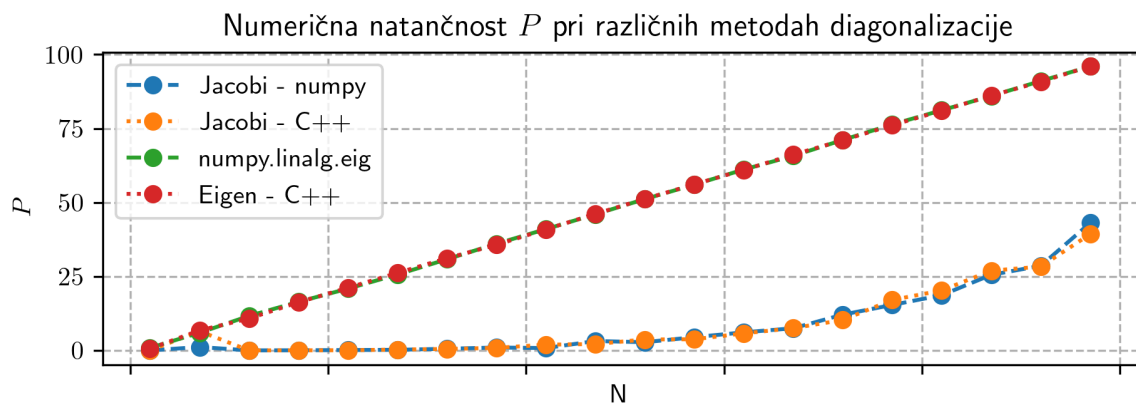


Testna matrika $n = 25$: Testno matriko velikosti N generiramo tako, da po diagonali nanesimo različne lastne vrednosti in nato simetrično po zgornji in spodnji trikotni dodamo naključna števila približno istega velikostnega razreda. Ne-diagonalna norma $P_A = 1.13e-08$

Različne implementacije algoritmov za diagonalizacijo lahko sedaj primerjamo po hitrosti izvedbe za vsako peto matriko v naboru 100 matrik. Implementacijo v C++ smo v strojni jezik prevedli s komando in parametri `g++ -O3 -std=c++11 -I ./Eigen eigenDiag.cpp -o testEigen`.



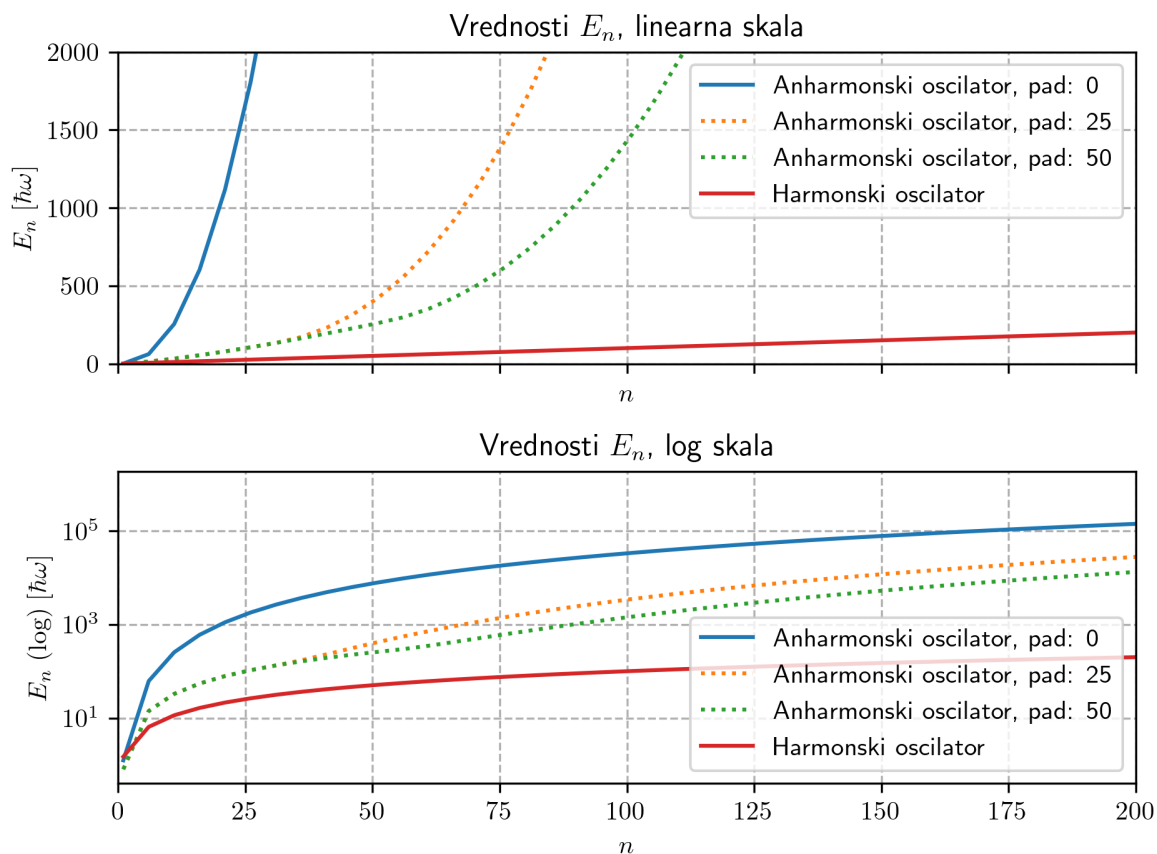
Iz grafa razberemo, da implementacija v jeziku C++ ni bistveno hitrejša (pomnožena kvečjemu za neko konstanto) s povečevanjem velikosti matrik. Pri primerjavi hitrosti smo za Jacobijevo metodo targetirali P sorazmeren z N , kar je privzeto obnašanje ostalih dveh implementacij iz programskih paketov.



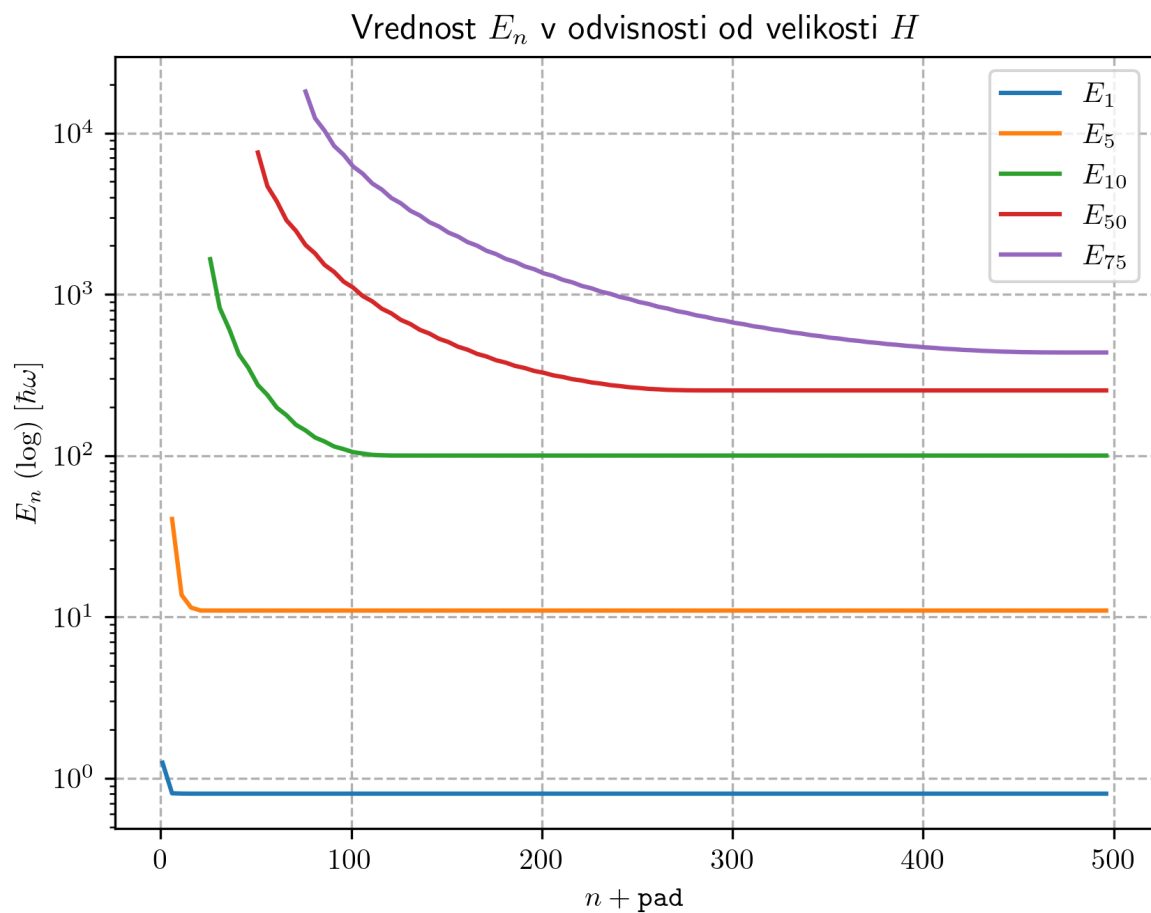
Zaradi učinkovitosti, preprostosti uporabe in dobre integracije s Pythonom se v nadaljevanju odločimo za uporabo funkcije `numpy.linalg.eig`.

6. Anharmonski oscilator

Sedaj z diagonalizacijo rešimo perturbiran Hamiltonjan za anharmonski oscilator. Zavedamo se, da ima Hamiltonjan dejansko neskončno lastnih funkcij/vektorjev $|n\rangle$ in lastnih vrednosti E_n . Torej lahko v naši simulaciji obravnavam samo približek z (dovolj) velikim n .

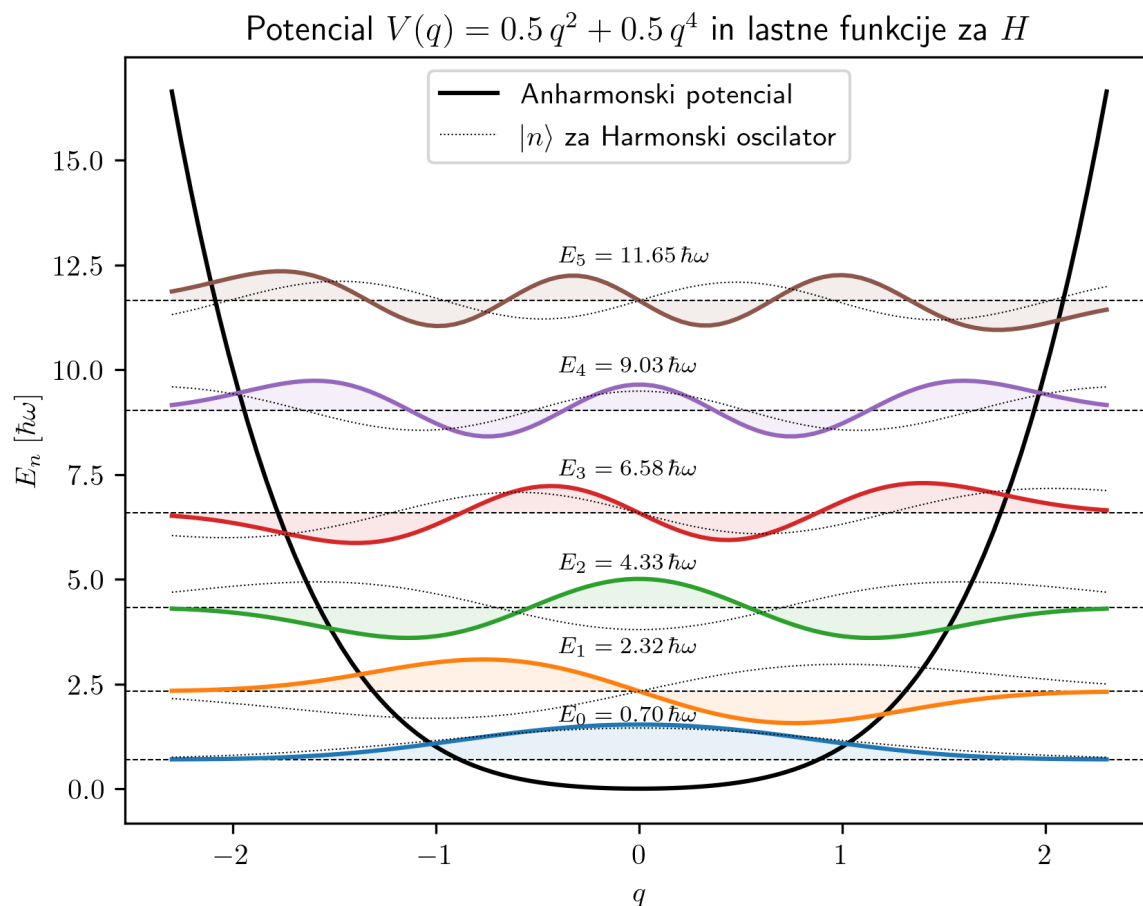


Ugotovimo, da lastne vrednosti E_n pri anharmonskem oscilatorju naraščajo z nelinearno odvisnostjo od N . Opazimo, da se energijski noviji tudi malo spreminjajo, če vzamemo n -ti člen iz Hamiltonjana velikosti $(n + \text{pad}) \times (n + \text{pad})$. Preverimo še kaj se dogaja s posameznim členom E_n , ko gre $\text{pad} \rightarrow \infty$.



Ugotovimo, da se vrednost E_n numerično sčasoma ustali. Torej lahko izrišemo energijske nivoje, če le vzamemo dovolj veliko vrednost pad . Iz zgornjega grafa dobimo

zelo grobo oceno pad $\approx 10 \cdot n$. Izrišimo še energijske nivoje:



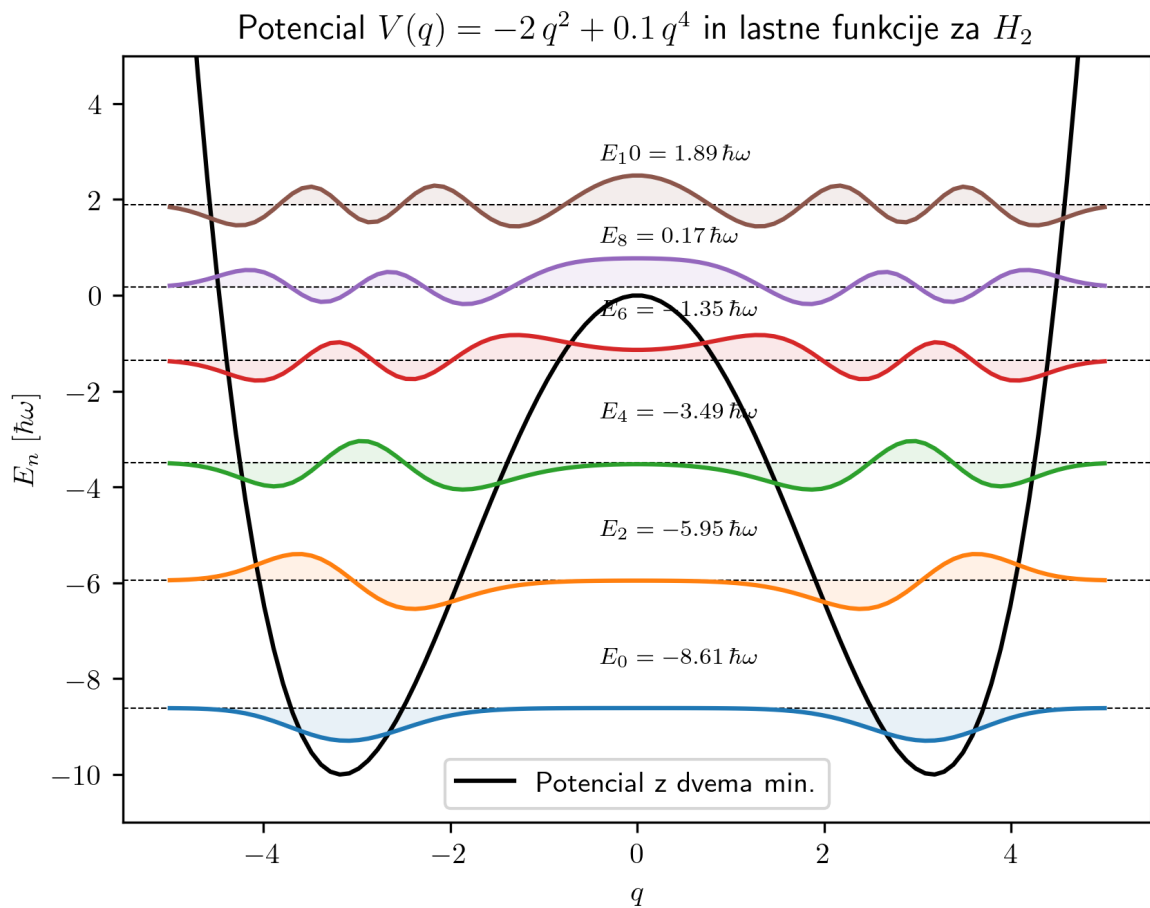
Iz grafa ugotovimo, da so si lastne funkcije za H (barvne črte), ki so linearna kombinacija lastnih funkcij za harmonski oscilator (črtkana), med sabo precej podobne. Podobnost je še bolj očitna, če namesto verjetnostnih gostot izrišemo kar pričakovane vrednosti. V naslednjem koraku bomo hkrati preučili še limitni primer, ko gre $\lambda \rightarrow 0$.



Animacija 1: Anharmonski oscilator v limiti $\lambda \rightarrow 0$. Prikazujemo potencial (črna), vrednosti lastne energije (črtkane vodoravne), verjetnostno gostoto za lastne funkcije H (barvne polne črte) in (referenčne) verjetnostno gostoto harmonskega oscilatorja (črtkano). Anharmonski oscilator limitira h harmonskemu.

7. Dodatek: Oscilator z dvema minimumoma

Analizo ponovimo zdaj še z malo drugačnim hamiltonjanom $H_2 = \frac{p^2}{2} - 2q^2 + \frac{q^4}{10}$. Torej v potencialu z dvema minimumoma.



Opazimo, da je delec z nižjimi energijami lahko lokaliziran predvsem v minimumih potencialov (kvantno tuneliranje?). Delec z višjo energijo pa lahko najdemo na celotni širini.

8. Zaključek

Tekom analize smo s pomočjo kvantne perturbacijske teorije poiskali skupno matrično Hamiltonovo funkcijo za anharmonski potencial λq^4 in potencial z dvema minimumoma. Testirali smo različne numerične načine za diagonalizacijo matrik tako v C++ kot v Pythonu. Z numerično diagonalizacijo smo rešili matrični sistem za anharmonski oscilator in oscilator z dvema minimumoma. Določili smo energijske nivoje ter lastne funkcije. Preverili smo korespondenčno načelo za limito $\lambda \rightarrow 0$ in ugotovili, da anharmonski oscilator v tej limiti (torej ko ugasne anharmonski) člen postane nazaj harmonski.