

Lastne vrednosti in lastni vektorji

Rok Mlinar Vahtar - 28211126

November 2023

1 Uvod

V tej nalogi se bomo spoznali z numerično diagonalizacijo velikih matrik na primeru anharmonskega oscilatorja. Kot prvo se spomnimo na to kaj je kvantni harmonski oscilator. Njegov hamiltonian se glasi

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

kjer energijo merimo v enotah $\hbar\omega$, gibalne količine v enotah $(\hbar m\omega)^{1/2}$ in dolžine v enotah $(\hbar/m\omega)^{1/2}$. Zanj poznamo tudi stacionarne rešitve

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

kjer so \mathcal{H}_n Hermitovi polinomi.

Anharmonični oscilator definiramo tako, da k hamiltonianu H_0 dodamo še en člen

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

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 $0 \leq \lambda \leq 1$. 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 kaknega drugega vira (npr Python). Vsaj enega izmed postopkov izvedi 'ročno' (sprogramiraj, uporabi izvorno 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 ?

3 Razvoj anharmonskega oscilatorja po bazi

Če želimo diagonalizirati hamiltonijan anharmonskega oscilatorja, ga moramo najprej sploh zapisati v matrični obliki. Za to moramo izbrati bazo v prostoru vseh funkcij in najbolj intuitivno smiselno je, da izberemo lastne funkcije harmonskega oscilatorja. Tako velja

$$|\psi\rangle = \sum_{n=1}^{\infty} c_n |n\rangle \quad \text{in} \quad E = \langle\psi|H_0|\psi\rangle + \lambda\langle\psi|q^4|\psi\rangle = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} c_i c_j (\langle i|H_0|j\rangle + \lambda\langle i|q^4|j\rangle)$$

kjer so $|n\rangle$, $|i\rangle$ in $|j\rangle$ lastne funkcije harmonskega oscilatorja. Matriko hamiltoniana lahko tako zapišemo kot:

$$H_{ij} = \langle i|H_0|j\rangle + \lambda\langle i|q^4|j\rangle, \text{ kjer že poznamo} \quad H_0^{i,j} = (i + \frac{1}{2})\delta_{i,j}$$

ta je načeloma neskončna, ampak jo bomo v približku vzeli kot končno matriko velikosti N , saj je kot neskončne nikoli nebi mogli diagonalizirati.

3.1 Matrični elementi $\langle i|q^k|j\rangle$

Za izračun matričnih elementov za $k = 1, 2, 4$ bosta pomembni dve zvezi

$$\langle i|j\rangle = \int_{-\infty}^{\infty} (2^i i! \sqrt{\pi})^{-1/2} (2^j j! \sqrt{\pi})^{-1/2} e^{-q^2} \mathcal{H}_i(q) \mathcal{H}_j(q) dq = \delta_{i,j}$$

$$x\mathcal{H}_i = \frac{1}{2}\mathcal{H}_{i+1} + i\mathcal{H}_{i-1}$$

Izpeljavo bom predstavil na primeru $\langle i|q|j\rangle$, polna izpeljava za ostale ja je priložena v rokopisu.

$$\begin{aligned} \langle i|q|j\rangle &= \int_{-\infty}^{\infty} (2^i i! \sqrt{\pi})^{-1/2} (2^j j! \sqrt{\pi})^{-1/2} e^{-q^2} \mathcal{H}_i(q) q \mathcal{H}_j(q) dq = \\ &= \int_{-\infty}^{\infty} (2^i i! \sqrt{\pi})^{-1/2} (2^j j! \sqrt{\pi})^{-1/2} e^{-q^2} \mathcal{H}_i(q) (\frac{1}{2}\mathcal{H}_{j+1} + i\mathcal{H}_{j-1}) dq = \\ &= \sqrt{\frac{2^{j+1}(j+1)! \sqrt{\pi}}{2^j j! \sqrt{\pi}}} \frac{1}{2} \langle i|q|j+1\rangle + \sqrt{\frac{2^{j-1}(j-1)! \sqrt{\pi}}{2^j j! \sqrt{\pi}}} j \langle i|q|j-1\rangle = \\ &= \delta_{i,j+1} \sqrt{\frac{i}{2}} + \delta_{i,j-1} \sqrt{\frac{j}{2}} \end{aligned}$$

Na podoben način, ki je v prilogi, bi lahko izpeljali tudi:

$$\begin{aligned} \langle i|q^2|j\rangle &= \frac{1}{2}(\delta_{i,j+2} \sqrt{(j+2)(j+1)} + \delta_{i,j} (2j+1) + \delta_{i,j-2} \sqrt{j(j-1)}) \\ \langle i|q^3|j\rangle &= \frac{1}{2^{\frac{3}{2}}}(\delta_{i,j+3} \sqrt{(j+3)(j+2)(j+1)} + \delta_{i,j+1} 3(j+1)^{\frac{3}{2}} + \delta_{i,j-1} 3j^{\frac{3}{2}} + \delta_{i,j-3} \sqrt{j(j-1)(j-2)}) \\ \langle i|q^4|j\rangle &= \frac{1}{4}(\delta_{i,j+4} \sqrt{(j+4)(j+3)(j+2)(j+1)} + \delta_{i,j+2} (3(j+2)^{\frac{3}{2}} \sqrt{j+1} + j \sqrt{(j+1)(j+2)}) + \\ &+ \delta_{i,j} (3(j^2 + (j+1)^2) + \delta_{i,j-2} (3(j-1)^{\frac{3}{2}} \sqrt{j} + (j+1) \sqrt{j(j-1)}) + \delta_{i,j-4} \sqrt{j(j-1)(j-2)((j-3))}) \end{aligned}$$

Motnjo navadnega harmoničnega oscilatorja, ki jo predstavlja anharmonski člen, si lahko zdaj apriksimiramo na več načinov. Te so $\langle i|q|j\rangle^4$, $\langle i|q^2|j\rangle^2$, $\langle i|q^3|j\rangle^{4/3}$. Navsezadje pa lahko vzamemo točno vrednost $\langle i|q^4|j\rangle$.

4 Numerična diagonalizacija matrik

Ker je matrika hamiltoniana, ki ga želimo diagonalizirati simetrična, se lahko odločimo za algoritem, za diagonalizacijo simetričnih matrik. Najbolj pogosto uporabljen izmed teh je QR iteracija, ki sem jo implementiral z Householderjevimi zrcaljenji.

4.1 Moja implementacija QR iteracije

4.1.1 Householderjeva zrcaljenja

Householdereva zrcaljenja so ortogonalne linearne transformacije, ki prvi vektor poljubne matrike spremenijo v vektor, usmerjen v smer prvega baznega vektorja. Z večkratno uporabo različnih transformacij vedno manjših dimenzij, lahko v matriki eliminiramo vse člene pod diagonalo. Dobili smo tako imenovani QR razcep, kjer je Q ortogonalna matrika in R zgornje trikotna.

$$Q_n \dots Q_2 Q_1 A = R \longrightarrow Q = (Q_n \dots Q_2 Q_1)^T = (Q_n \dots Q_2 Q_1)^{-1} \quad A = QR \quad (1)$$

Posamezno matriko Q_i , ki deluje na matriki $Q_{i-1} \dots Q_2 Q_1 A$ naslednje oblike:

$$\begin{bmatrix} * & * & * & \dots \\ 0 & * & * & \\ 0 & 0 & * & \\ \vdots & & \ddots & \ddots \\ & & & \begin{bmatrix} | & | & \dots \\ \vec{c}_1 & \vec{c}_2 & \dots \\ | & | & \end{bmatrix} \end{bmatrix}$$

dobimo z naslednjim predpisom, kjer $\vec{u} = \vec{c}_1 + \text{sign}(c_{11}) \|\vec{c}_1\| \vec{e}_1$

$$Q = \begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ \vdots & & \ddots & \ddots \\ & & & \begin{bmatrix} \mathbf{I} - 2 \frac{\vec{c}_1 \vec{c}_1^T}{\|\vec{c}_1\|^2} \end{bmatrix} \end{bmatrix}$$

Če enak postopek začnemo eno dimenzijo nižje in isti Q_i uporabimo tako na vrsticah kot stolpcih, v praksi $A_i = Q_i A_{i-1} Q_i$, lahko tako simetrično matriko pretvorimo v tridiagonalno. Na tem mestu bi opozoril tudi na to, da če v algoritmu pride do deljenja z 0, to pomeni, da so ničle, ki jih želimo doseči, že v matriki in lahko en korak preskočimo.

4.1.2 QR iteracija

Metoda QR iteracije je izjemno enostavna. Simetrično matriko A diagonaliziramo tako, da ponavljamo naslednjo operacijo:

$$A_0 = A \quad A_k = Q_k R_k \longrightarrow A_{k+1} = R_k Q_k \quad (2)$$

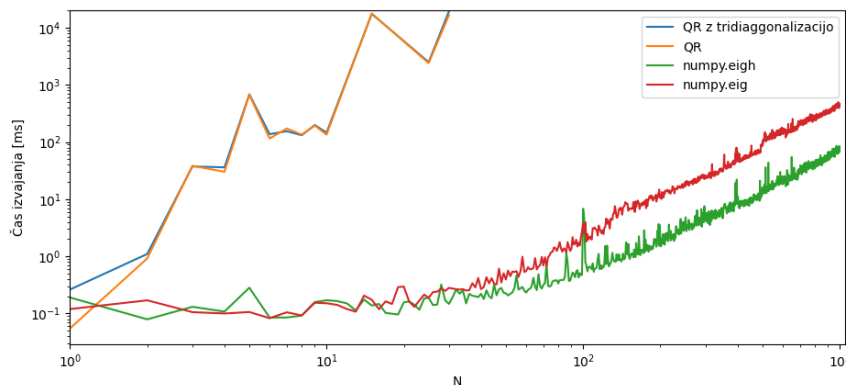
Po vsaki iteraciji preverimo, če je mera za velikost vseh nediagonalnih členov padla pod željeno vrednost. Pogosto, kot se izkaže še posebej za velike matrike, lahko postopeč pospešimo, če matriko najprej prevedemo na tridiagonalno.

4.2 Primerjava različnih algoritmov

Prej omenjene algoritme sem primerjal na matrikah različnih velikosti, z elementi iz $[-10, 10]$. To pomeni, da moramo meritve, ki sledijo, razumeti kot zgornjo mejo za čas trajanja, saj so v fiziki matrike pogosto zelo redke.

- QR Householder brez tridiagonalizacije: Moja implementacija QR iteracije kjer QR razcep izvajamo na začetni matriki.
- QR Householder z tridiagonalizacijo: Moja implementacija QR iteracije kjer QR razcep izvajamo na tridiagonalni matriki, ki jo dobimo preko Householderjevih zrcaljenj.
- `numpy.eigh`: Algoritem, ki matriko prevede na tridiagonalno in nato poišče lastne vektore preko metod deli in vlada.
- `numpy.eig`: Algoritem, ki pretvori matriko na zgornjo Hessenbergovo obliko in nato izvaja QR iteracijo.

Velikost matrike	Algoritem	čas izvajanja[ms]
2	QR Householder z tridiagonalizacijo	25
2	QR Householder brez tridiagonalizacije	10
2	<code>numpy.eigh</code>	1
2	<code>numpy.eig</code>	0.1
10	QR Householder z tridiagonalizacijo	412
10	QR Householder brez tridiagonalizacije	401
10	<code>numpy.eigh</code>	1
10	<code>numpy.eig</code>	0.2
50	QR Householder z tridiagonalizacijo	45600
50	QR Householder brez tridiagonalizacije	327000
50	<code>numpy.eigh</code>	13
50	<code>numpy.eig</code>	0.5
1000	<code>numpy.eigh</code>	73
1000	<code>numpy.eig</code>	344
10000	<code>numpy.eigh</code>	45000
10000	<code>numpy.eig</code>	183000



Slika 1: Povprečen čas izvajanja za različne velikosti N

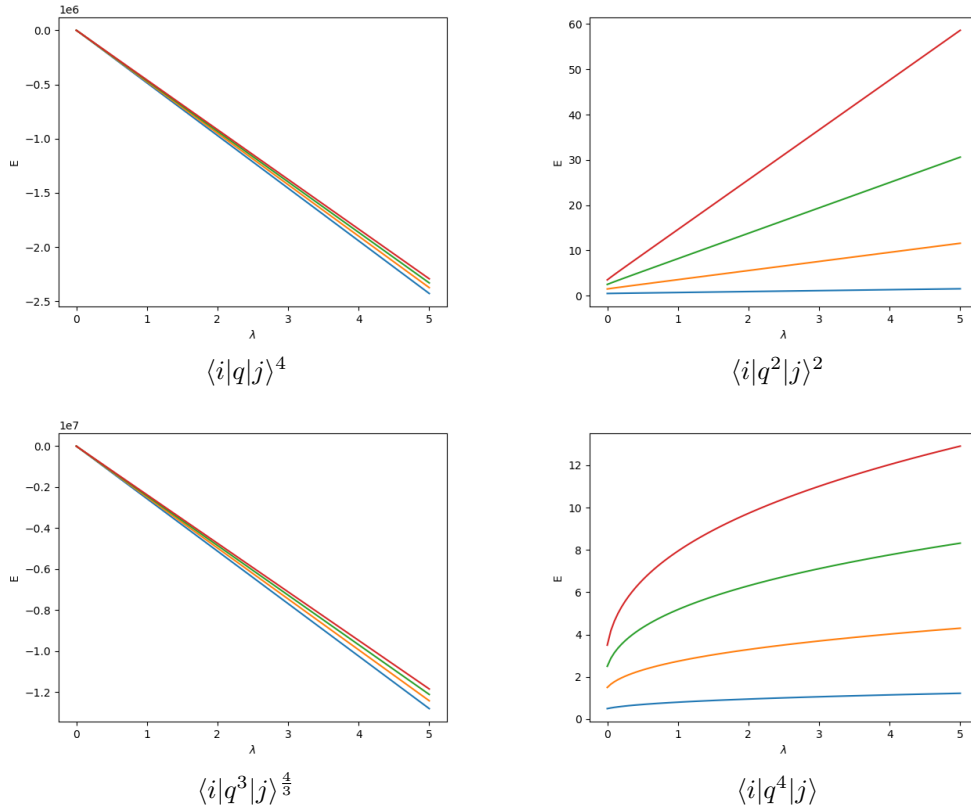
Iz tabele lahko vidimo, da sta metodi, ki izkoristita simetrijo, specifično QR Householder z tridiagonalizacijo in `numpy.eigh` počasnejši na majhnih matrikah, ampak v zameno drastično hitrejši na velikih. V nadaljnje bo za vse račune zato uporabljena funkcija `numpy.eigh`.

Vrednosti na grafu so zaradi dolgega časa risanja izmerjene kot povprečje za majhno število naključnih matrik, zato absolutne številke niso tako relevantne kot splošen trend.

5 Diagonalizacija hamiltoniana

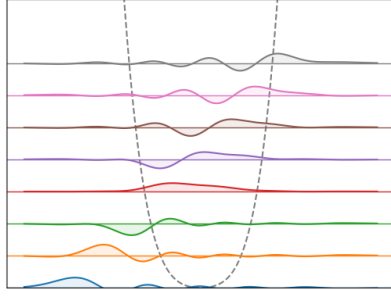
5.1 Spektri

Zdaj, ko smo opremljeni z matriko našega hamiltoniana in algoritmom za diagonalizacijo, lahko končno najdemo lastna stanja anharmoničnega oscilatorja. Kot prvo, si pogledimo spektre pri različnih vrednostih λ .

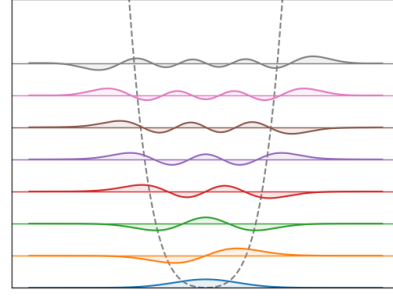


Prvi štirje energijski nivoji v odvisnosti od λ za vse štiri približke

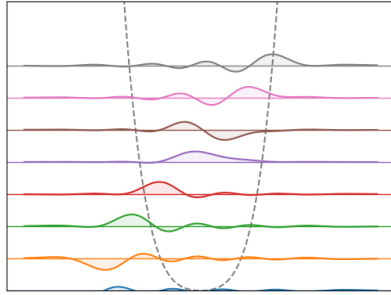
Ker vemo, da je $\langle i|q^4|j\rangle$ najbolj točen približek, lahko ostale tri primerjamo z njim. Vsi približki se, kot bi pričakovali, strinjajo v točki $\lambda = 0$, po tem pa oba liha približka $\langle i|q^3|j\rangle^{\frac{4}{3}}$ in $\langle i|q|j\rangle^4$ eksplodirata v primerjavi z sodima, kar nam pove, da najverjetneje nista dobra. $\langle i|q^2|j\rangle^2$ je veliko boljši, ampak računanje dvakrat polnejše matrike $\langle i|q^4|j\rangle$ ni dovolj zamudno, da bi se izplačal. Za razlago, zakaj liha približka tako "divergirata", pa si oglejmo, kako izgledajo same valovne funkcije.



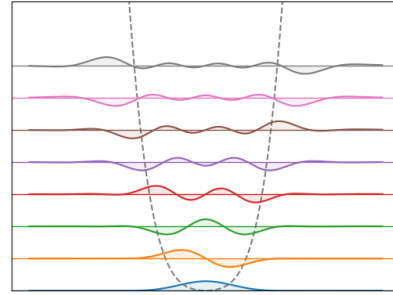
$$\langle i|q|j\rangle^4$$



$$\langle i|q^2|j\rangle^2$$



$$\langle i|q^3|j\rangle^{\frac{4}{3}}$$



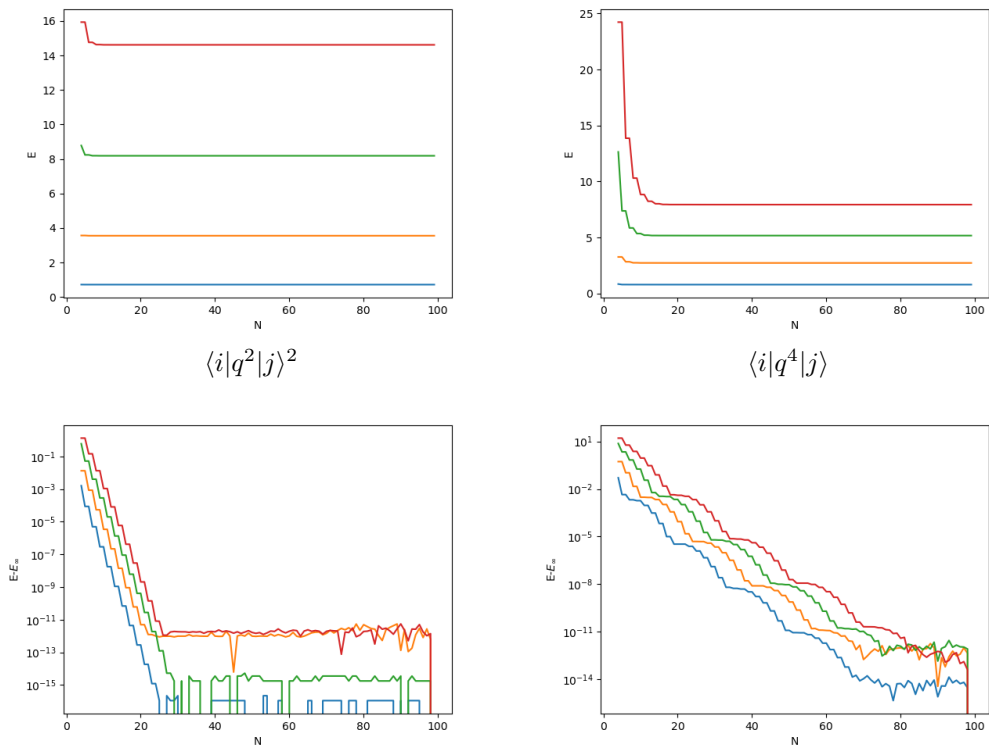
$$\langle i|q^4|j\rangle$$

Prvih osem lastnih funkcij (od spodaj navzgor) za vse štiri približke

Vidimo, da v lihih približkih valovne funkcije poskušajo pobegniti na levo, kar je smiselno če pomislimo, kdaj bi bila približka $\langle i|q|j\rangle$ in $\langle i|q^3|j\rangle$ točna. To bi bilo za nek lihi potencial $V \propto q, q^3$, ta potencial pa bi v primeru q imel minimum nekje levo od ničle, q^3 pa bi desno od ničle imel potencialni hrib, nato pa prepad v $-\infty$. Jasno je, da bosta zato za večje vrednosti λ oba približka hitro dosegla energije manjše od 0. Od zdaj naprej se lahko zato osredotočimo na soda približka.

5.2 Odvisnost približka od velikosti N

Zanimivo bi bilo tudi pogledati, kako veliko matriko H moramo vzeti, da dobimo dober približek.



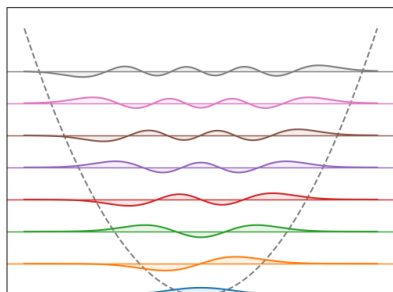
Odstopanje od končne vrednosti za $\langle i|q^2|j \rangle^2$ Odstopanje od končne vrednosti za $\langle i|q^4|j \rangle$

Odvisnost spektrov sodim približkov od velikosti hamiltoniana (Zaporedne lastne vrednosti si sledijo od spodaj navzgor)

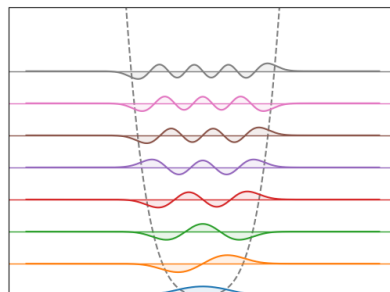
Iz grafov lahko vidimo, da se kvadratični približek zelo hitro stabilizira na končni vrednosti in nato niha okrog nje. Kvartični približek pa se na z večanjem N eksponentno približuje neki točni rešitvi. Dobra novica je tudi to, da za dobro natančnost, zadošča izbira relativno majhnega $N \approx 100$, kar pomeni, da problem lahko z dobrim algoritmom rešimo v roku nekaj deset milisekund.

5.3 Valovne funkcije pri različnih λ

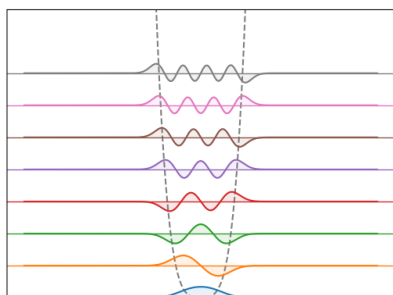
Če zdaj ovržemo še kvadratični približek, si za konec lahko ogledamo rešitve valovne funkcije anharmoničnega oscilatorja za različne vrednosti λ .



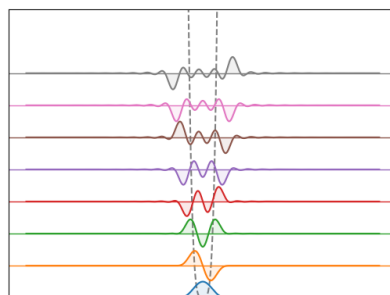
$\lambda = 0$



$\lambda = 1$



$\lambda = 10$



$\lambda = 1000$

Prvih osem lastnih funkcij (zaporedno si sledijo od spodaj navzgor) za različne λ

Kot bi pričakovali, postanejo funkcije z večjimi λ bolj ozke. Opomnil bi, da kljub temu, da na četrti sliki izgleda, kot da funkcije segajo močno čez rob potenciala, se moramo zavedati, da se razlike med lastnimi energijami nelinearno večajo z λ , kar pomeni, da niso narisane na "pravi" višini zaradi grafičnih omejitev.

6 Dodatna naloga

Poišči še nekaj najnižjih lastnih energij in lastnih funkcij za problem v potencialu z dvema minimumoma

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

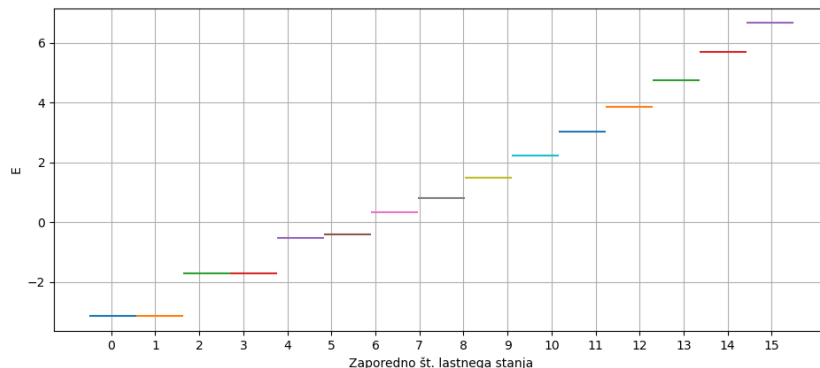
Kot prvo bomo potencial prepisali v podobno obliko kot prej.

$$H = \frac{1}{2}H_0 - \frac{3}{2}q^2 + \frac{1}{10}q^4$$

Po istem postopku kot prej, lahko zdaj zapišemo matriko hamiltoniana kot

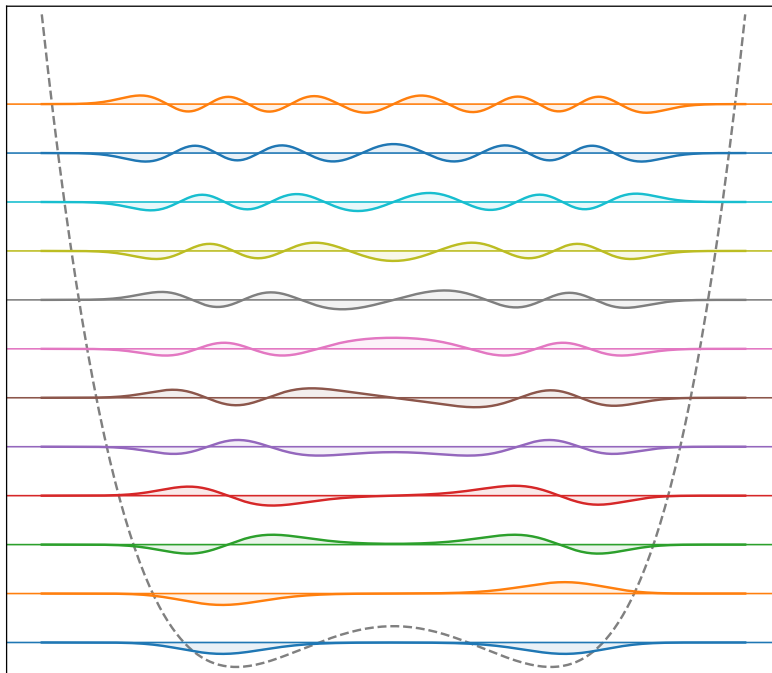
$$H_{ij} = \frac{1}{2}(i + \frac{1}{2})\delta_{i,j} - \frac{3}{2}\langle i|q^2|j\rangle + \frac{1}{10}\langle i|q^4|j\rangle$$

Zdaj pa lahko enostavno izračunamo tako spekter kot lastne funkcije.



Slika 2: Spekter za potencial z dvema minimumoma

Vidimo, da sta prvi dve stanji degenerirani, nato pa se stanja počasi razcepljajo. To lahko vidimo tudi na grafih valovnih funkcij, kjer so degenerirane funkcije le simetrične/antisimetrične variante rešitve uje v posameznem minimumu.



Slika 3: Lastne funkcije (zaporedno si sledijo od spodaj navzgor) za potencial z dvema minimumoma

7 Zaključek

V nalogi smo se spoznali z numeričnim reševanjem problemov lastnih vrednosti in spotoma našli rešitev anharmoničnega oscilatorja. Glavna stvar, ki sem jo pridobil, je veliko spoštovanja do ljudi, ki so spisali knjižnice LAPACK in podobne, ker si nisem predstavljal, da bodo vgrajene metode tako drastično hitrejšje. Za konec pa bi rad še pripomnil, da je izpeljava za različne $\langle i|q^a|j\rangle$ in posledično inkluzija $\langle i|q^3|j\rangle^{\frac{4}{3}}$, ki se vam je morda zdela nepotrebna, saj so izpeljane formule podane na drugi strani navodil, nekaj kar sem dodal le zato, ker sem porabil veliko preveč časa in truda za izpeljavo, preden sem ugotovil, da druga stran navodil sploh obstaja.