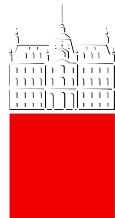


Univerza v Ljubljani
Fakulteta za *matematiko in fiziko*



Višje računske metode

Zaključna naloga:
2D anharmonski oscilator

Avtor: Jan Gavranovič, 28202029

Predavatelj: prof. dr. Tomaž Prosen
Asistent: Žiga Krajnik

24. junij 2021

1 Uvod

Pri nalogi nas zanima časovni razvoj valovne funkcije $\psi(x, y, t)$, ki ga dobimo z reševanjem časovno odvisne dvodimensionalne Schrödingerjeve enačbe

$$i\hbar \frac{\partial}{\partial t} \psi(x, y, t) = H\psi(x, y, t), \quad H = -\frac{\hbar^2}{2m} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) + V(x, y, t). \quad (1)$$

Problem sem reševal v brezdimenzijskih enotah $\hbar = m = 1$. Za potencial sem izbral časovno neodvisni dvodimensionalni anharmonski oscilator

$$V(x, y) = \frac{1}{2}x^2 + \frac{1}{2}y^2 + \lambda x^2y^2, \quad (2)$$

kjer je parameter λ majhna motnja. Za začetno stanje sem vzel lastno funkcijo harmonskega oscilatorja, ki je v eni dimenziji

$$\phi_N(x) = \frac{1}{\pi^{1/4}\sqrt{2^N N!}} H_N(x) \exp(-x^2/2) \quad (3)$$

v dveh dimenzijah pa $\phi_N(x, y) = \phi_N(x)\phi_N(y)$, kjer so H_N Hermitovi polinomi. Za območje sem vzel mrežo s stranicami $x \in [-L, L]$ in $y \in [-L, L]$. Na robovih velja $\psi(x = \pm L, y = \pm L, t) = 0$. Začetno stanje lahko premaknemo za a_x ali a_y , da dobimo koherentno stanje

$$\psi(x, y, t = 0) = \phi_N(x - a_x)\phi_N(y - a_y). \quad (4)$$

1.1 Končne diference in zapis operatorja ∇^2

Uporabimo ekvidistantno mrežo z naslednjimi mrežnimi točkami

$$(x_i, y_j, t_n) = (i\Delta x, j\Delta y, n\Delta t), \quad (5)$$

kjer so $\Delta x = (x_N - x_0)/N$ in $\Delta y = (y_N - y_0)/N$, kar nam da $\Delta x = \Delta y = h$. Valovna funkcija je predstavljena kot matrika dimenzijskega $N \times N$ in jo lahko zapišemo kot vektor dolžine N^2 kot

$$\begin{bmatrix} \psi_{11} & \psi_{12} & \dots & \psi_{1N} \\ \psi_{21} & \psi_{22} & \dots & \psi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{N1} & \psi_{N2} & \dots & \psi_{NN} \end{bmatrix} \rightarrow [\psi_{11} \ \psi_{21} \ \dots \ \psi_{1N} \ \psi_{21} \ \psi_{22} \ \dots \ \psi_{2N} \ \dots \ \psi_{NN}]^\top. \quad (6)$$

Enako naredimo s potencialom, ki pa ni vektor ampak je diagonalna matrika dimenzijskega $N^2 \times N^2$

$$\begin{bmatrix} V_{11} & V_{12} & \dots & V_{1N} \\ V_{21} & V_{22} & \dots & V_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ V_{N1} & V_{N2} & \dots & V_{NN} \end{bmatrix} \rightarrow \begin{bmatrix} V_{11} & & & \\ & V_{12} & & \\ & & \ddots & \\ & & & V_{NN} \end{bmatrix}. \quad (7)$$

Končne simetrične diference drugega reda za $\partial^2/\partial x^2$ in $\partial^2/\partial y^2$ so

$$\frac{\partial^2 \psi}{\partial x^2} \approx \frac{\psi_{i+1j} - 2\psi_{ij} + \psi_{i-1j}}{\Delta x^2} + \mathcal{O}(\Delta x^2) \quad \text{in} \quad \frac{\partial^2 \psi}{\partial y^2} \approx \frac{\psi_{ij+1} - 2\psi_{ij} + \psi_{ij-1}}{\Delta y^2} + \mathcal{O}(\Delta y^2). \quad (8)$$

Če se problema lotimo matrično z zapisom 6 in 7, potem zapišemo zgornjo diskretizacijo na operatorski način. Za diference drugega reda definiramo tridiagonalno $N \times N$ matriko

$$D^{(2)} = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \\ & & & 1 & -2 \end{bmatrix}. \quad (9)$$

Približka za odvod 8 zapišemo kot

$$D_x^{(2)} = \mathbb{1} \otimes D^{(2)} \quad \text{in} \quad D_y^{(2)} = D^{(2)} \otimes \mathbb{1}, \quad (10)$$

ki ju lahko preverimo z množenjem s ψ iz desne. Celoten operator pa je

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \rightarrow \mathbb{1} \otimes D^{(2)} + D^{(2)} \otimes \mathbb{1} = D^{(2)} \oplus D^{(2)}. \quad (11)$$

Zapišimo še petdiagonalno matriko za četrти red

$$D^{(4)} = \frac{1}{12h^2} \begin{bmatrix} -30 & 16 & -1 & & \\ 16 & -30 & 16 & -1 & \\ -1 & 16 & -30 & 16 & -1 \\ & -1 & 16 & -30 & 16 & -1 \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 16 & -30 \end{bmatrix}. \quad (12)$$

1.2 Časovni razvoj in split-step

Časovni razvoj začetnega stanja zapišemo kot

$$\psi(\mathbf{r}, \Delta t) = e^{-iH\Delta t} \psi(\mathbf{r}, 0) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} (iH\Delta t)^n \psi(\mathbf{r}, 0) = U(\Delta t) \psi(\mathbf{r}, 0), \quad (13)$$

kjer je $U(\Delta t) = e^{-iH\Delta t}$ unitarni operator časovnega razvoja. Naš Hamiltonian je sestavljen iz dveh delov $H = A + B$. Lastnost eksponentov $e^{A+B} = e^A e^B$ velja samo, če argumenta komutirata, torej $[A, B] = 0$. Če to ne drži, uporabimo za zapis eksponentov Baker–Campbell–Hausdorffovo formulo

$$e^A e^B = \exp \left[A + B + \frac{1}{2}[A, B] + \frac{1}{12}([A, [A, B]] + [B, [B, A]]) + \dots \right]. \quad (14)$$

Simetrični split-step razcep drugega reda je tako

$$e^{A+B} \approx e^{A/2} e^B e^{A/2} \quad (15)$$

ozziroma

$$\psi(\mathbf{r}, t + \Delta t) \approx e^{-iH_0\Delta t/2} e^{-iV\Delta t} e^{-iH_0\Delta t/2} \psi(\mathbf{r}, t), \quad (16)$$

kjer je $H_0 = -\frac{1}{2}\nabla^2$ operator kinetične energije in V potencial.

2 Numerične metode

Predstavil bom nekaj metod, ki sem jih uporabil pri reševanju časovno odvisne 2D Schrödingerjeve enačbe. Metode so treh vrst: eksplisitne, implicitne in spektralne. Razlikujejo se še v tem ali uporabljam split-step razcep ali pa direktno časovno diskretizacijo. Problem matričnega zapisa z operatorji D je sama velikost matrik, ki raste z drugo potenco števila mrežnih točk N . Ta problem deloma rešijo redke (*sparse*) matrike. Pri reševanju sem zato za delo z matrikami uporabil `scipy.sparse`. Metode bom med seboj primerjal z naslednjimi kriteriji:

- Hitrost računanja, koliko iteracij (časovnih korakov) na sekundo (it/s) doseže posamezen algoritem.
- Ohranitev norme valovne funkcije $\langle \psi | \psi \rangle$.
- Ohranitev skupne energije $\langle \psi | H | \psi \rangle$.
- Ohranitev začetne oblike valovne funkcije.

Primerjavo sem naredil za potencial brez motnje λ , začetno valovno funkcijo sem izmaknil za $a_x = 1$, računal pri $L = 5$ in diskretizaciji velikosti 128×128 . Metode niso stabilne za vse vrednosti časovnega koraka Δt , zato sem s pomočjo bisekcije določil najmanjši Δt za katerega da algoritem še dovolj natančen rezultat. Za kriterij sem postavil, da norma valovne funkcije ne sme preseči vrednosti 1.025 v kateremu koli času. Bisekcijo sem naredil med 1000 in 20000 časovnimi koraki, ponavljal pa sem jo 15 korakov. Sam kriterij ni preveč pomemben, saj je za nestabilno rešitev značilno, da pobegne v neskončnost.

2.1 Eulerjeva in Crank-Nicolsonova metoda

Najenostavnejša shema je Eulerjeva, kjer razvijemo eksponent do prvega reda

$$\psi(\mathbf{r}, t + \Delta t) = (1 - i\Delta t H) \psi(\mathbf{r}, t), \quad (17)$$

kjer za H vzamemo $-\frac{1}{2}\mathbf{D} \oplus \mathbf{D} + V$. Metoda ima pomanjkljivost, da tako zapisan operator časovnega razvoja ni več unitaren, kar jo naredi nestabilno in zato potrebuje eksponentno majhne Δt , ter tudi renormalizacijo valovne funkcije na vsakem koraku.

Slabosti Eulerjeve metode popravi Crank–Nicolsonova aproksimacija

$$\psi(\mathbf{r}, t + \Delta t) = \frac{2\mathbb{1} - iH\Delta t}{2\mathbb{1} + iH\Delta t} \psi(\mathbf{r}, t), \quad (18)$$

ki je unitarna in stabilna. Metoda zahteva izračun inverza, kar je za večje matrike zelo potratno. Zapišemo jo lahko tudi v diferenčni obliki:

$$\begin{aligned} 2\psi_{ij}^{n+1} + i\Delta t \left[-\frac{1}{2h^2} (\psi_{i+1j}^{n+1} + \psi_{ij-1}^{n+1} + \psi_{ij+1}^{n+1} + \psi_{ij-1}^{n+1} - 4\psi_{ij}^{n+1}) + V_{ij}\psi_{ij}^{n+1} \right] = \\ 2\psi_{ij}^n - i\Delta t \left[-\frac{1}{2h^2} (\psi_{i+1j}^n + \psi_{ij-1}^n + \psi_{ij+1}^n + \psi_{ij-1}^n - 4\psi_{ij}^n) + V_{ij}\psi_{ij}^n \right], \end{aligned} \quad (19)$$

kar je linearen sistem enačb katerega reševanje ima enako visoko časovno zahtevnost ($\mathcal{O}(N^3)$ z Gaussovo eliminacijo). Metoda je zaradi tega v dveh dimenzijah praktično neuporabna, razen za trivialno majhne diskretizacije pozicijske mreže.

2.2 Alternating-direction implicitna (ADI) metoda

Ideja te metode je podobna tisti iz 19, le da razdelimo časovni korak na dva podintervala velikosti $\Delta t/2$. Na prvem intervalu uporabimo eksplicitno shemo v eni smeri ter implicitno v drugi, na drugem časovnem intervalu pa ravno obratno (slika 1). Zapišemo Schrödingerjevo enačbo s končnimi diferencami za prvi interval

$$a \left[\psi_{ij}^{n+\frac{1}{2}} - \psi_{ij}^n \right] = b \left[\psi_{i+1j}^{n+\frac{1}{2}} - 2\psi_{ij}^{n+\frac{1}{2}} + \psi_{ij-1}^{n+\frac{1}{2}} + \psi_{ij+1}^n - 2\psi_{ij}^n + \psi_{ij-1}^n \right] + V_{ij}\psi_{ij}^n, \quad (20)$$

kjer je $a = \frac{2i}{\Delta t}$ in $b = -\frac{1}{2h^2}$. Pospravimo člene z $n + \frac{1}{2}$ na levo in dobimo

$$(a + 2b)\psi_{ij}^{n+\frac{1}{2}} - b\psi_{i+1j}^{n+\frac{1}{2}} - b\psi_{ij-1}^{n+\frac{1}{2}} = b [\psi_{ij+1}^n - 2\psi_{ij}^n + \psi_{ij-1}^n] + V_{ij}\psi_{ij}^n, \quad (21)$$

kar je tridiagonalni sistem enačb pri fiksnem j . Za drugi podinterval dobimo

$$(a + 2b)\psi_{ij}^{n+1} - b\psi_{ij+1}^{n+1} - b\psi_{ij-1}^{n+1} = b \left[\psi_{i+1j}^{n+\frac{1}{2}} - 2\psi_{ij}^{n+\frac{1}{2}} + \psi_{ij-1}^{n+\frac{1}{2}} \right] + V_{ij}\psi_{ij}^{n+\frac{1}{2}}, \quad (22)$$

kar je spet tridiagonalen sistem tokrat pri fiksnem i , kjer $\psi_{ij}^{n+\frac{1}{2}}$ dobimo iz prejšnjega koraka.

Celotno metodo lahko zapišemo na operatorski način z matrikami. Schrödingerjeva enačba je

$$a \left[\psi^{n+\frac{1}{2}} - \psi^n \right] = b \left[D_x \psi^{n+\frac{1}{2}} + D_y \psi^n \right] + V \psi^n , \quad (23)$$

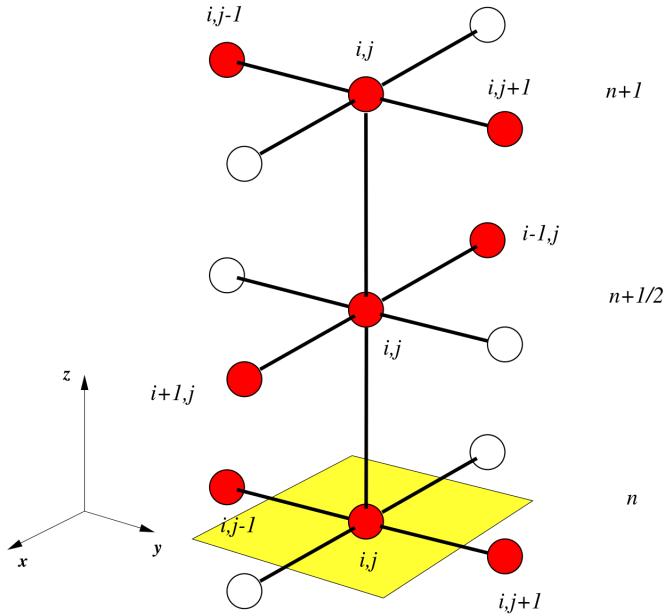
kar preuredimo v

$$[a \mathbb{1} - b D_y] \psi^{n+\frac{1}{2}} = [a \mathbb{1} + b D_x + V] \psi^n . \quad (24)$$

V naslednjem podintervalu dobimo enako kot prej

$$[a \mathbb{1} - b D_y] \psi^{n+1} = [a \mathbb{1} + b D_x + V] \psi^{n+\frac{1}{2}} , \quad (25)$$

kjer $\psi^{n+\frac{1}{2}}$ dobimo iz prvega podintervala.

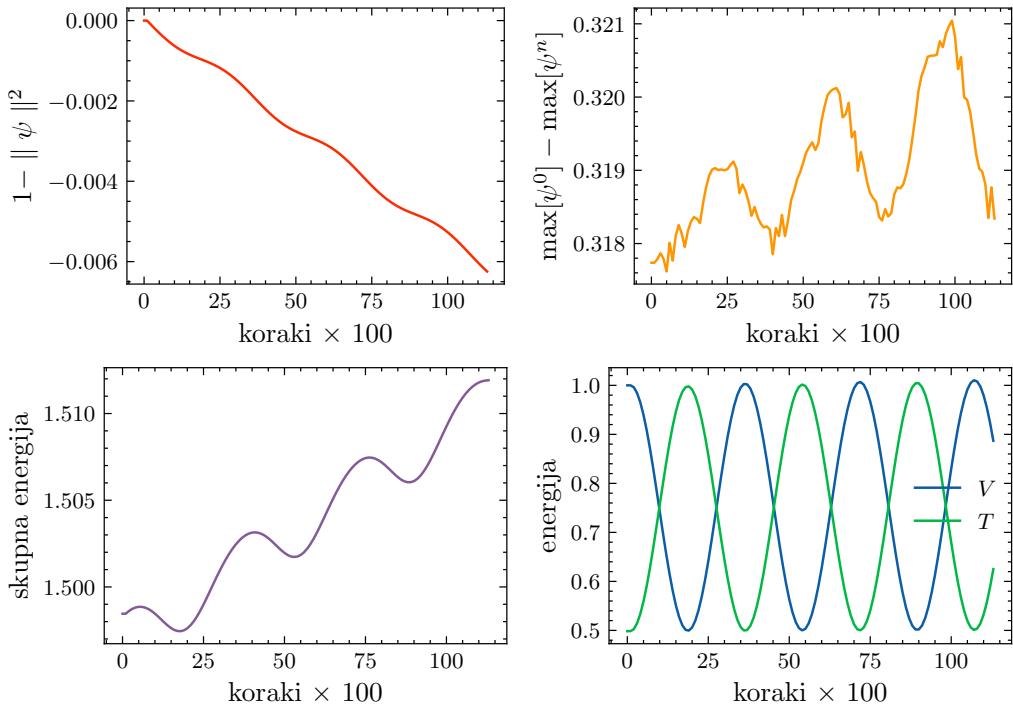


Slika 1: Shematski prikaz iterativnega postopka iz enačb 21 in 22 [2].

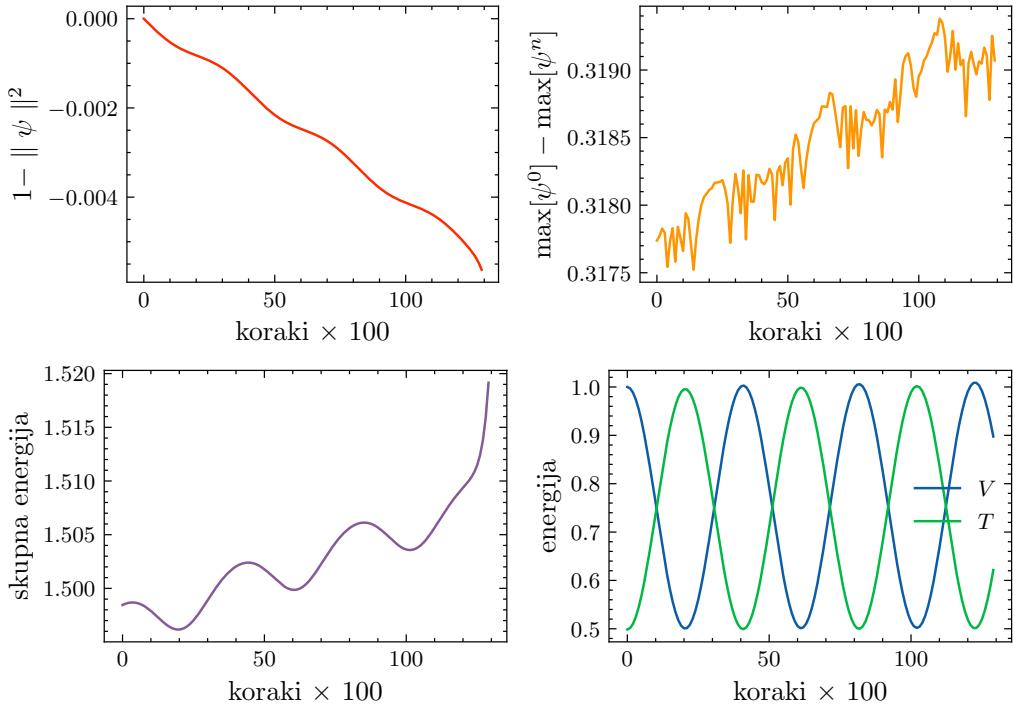
Sistem enačb torej rešimo za vsak i pri fiksniem j ter za vsak j pri fiksniem i . Za reševanje tridiagonalnega sistema sem v 21 in 22 uporabil `scipy.linalg.banded`. Zgornja grafa na sliki 2 prikazujeta odstopanje norme ter razliko največje vrednosti (vrha) verjetnostih gostot prve in trenutne valovne funkcije. Na spodnjih dveh slikah 2 pa je prikazana ohranitev energije.

Na sliki 3 je prikaz enakih količin, le da so rezultati dobljeni z matričnima zvezama 24 in 25. Sistema sedaj nista več tridiagonalna. Opazimo, da je matrika na levi strani pri iteraciji konstanta, zato sem jo že na začetku faktoriziral s `scipy.sparse.linalg.factorized`, kar prihrani nekaj časa pri reševanju sistema.

Oba pristopa potrebuje približno enako majhen Δt . Iz slik vidimo, da metoda ne ohranja dobro norme niti skupne energije. Slabosti lahko popravimo s stalno renormalizacijo po korakih in še z zmanjšanjem Δt , kar pa vzame dodaten računski čas.



Slika 2: Metoda z računanjem rešitev dveh tridiagonalnih sistemov pri $\Delta t = 8.25 \times 10^{-4}$.



Slika 3: Metoda zapisana v matrični obliki z uporabo $D^{(4)}$ pri $\Delta t = 8.89 \times 10^{-4}$.

2.3 Eksplisitne diferenčne metode

Najbolj direktna je metoda pri kateri diskretiziramo $\partial/\partial t$ s prvimi differencami in ∇^2 z drugimi:

$$\frac{i}{2\Delta t} [\psi^{n+1} - \psi^{n-1}] = \left[-\frac{1}{2h^2} (\mathbf{D} \oplus \mathbf{D}) + V \right] \psi^n , \quad (26)$$

kar nam da zvezo za naslednji ψ

$$\psi^{n+1} = \psi^{n-1} + i \frac{\Delta t}{h^2} (\mathbf{D} \oplus \mathbf{D}) \psi^n - 2i\Delta t V \psi^n , \quad (27)$$

kjer moramo poznati ψ^{n-1} v prejšnjem koraku.

Split-step diferenčno metodo lahko dobimo z razliko operatorjev časovnega razvoja za korak naprej in za korak nazaj:

$$\left. \begin{aligned} \psi^{n+1} &= e^{-iH\Delta t} \psi^n \\ \psi^{n-1} &= e^{iH\Delta t} \psi^n \end{aligned} \right\} - , \quad (28)$$

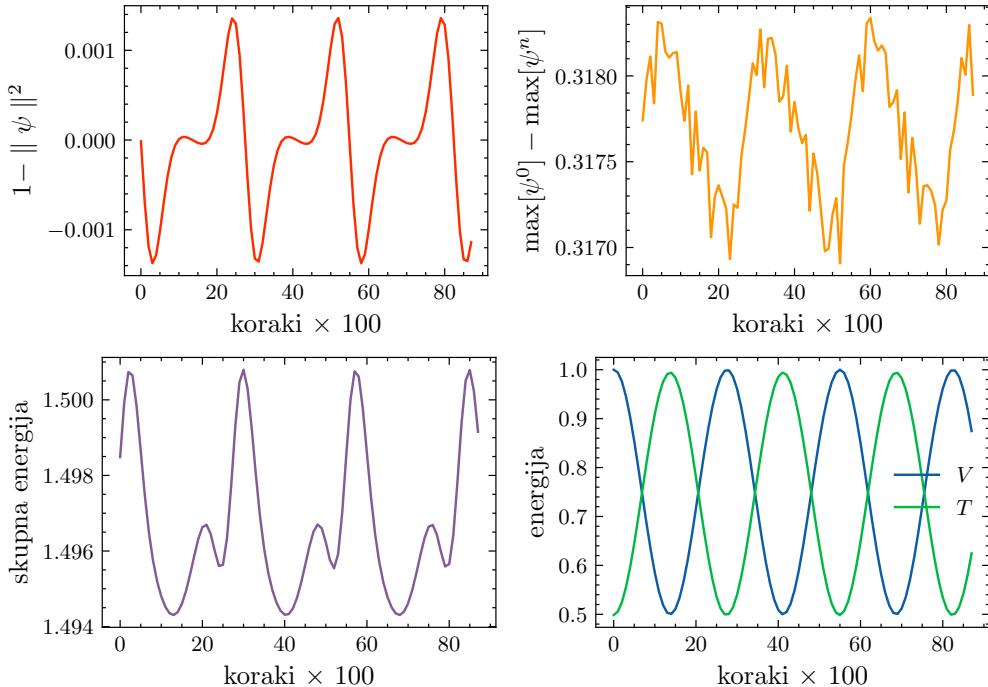
kar nam da z razvojem eksponenta

$$\psi^{n+1} = \psi^{n-1} - 2iH\Delta t \psi^n . \quad (29)$$

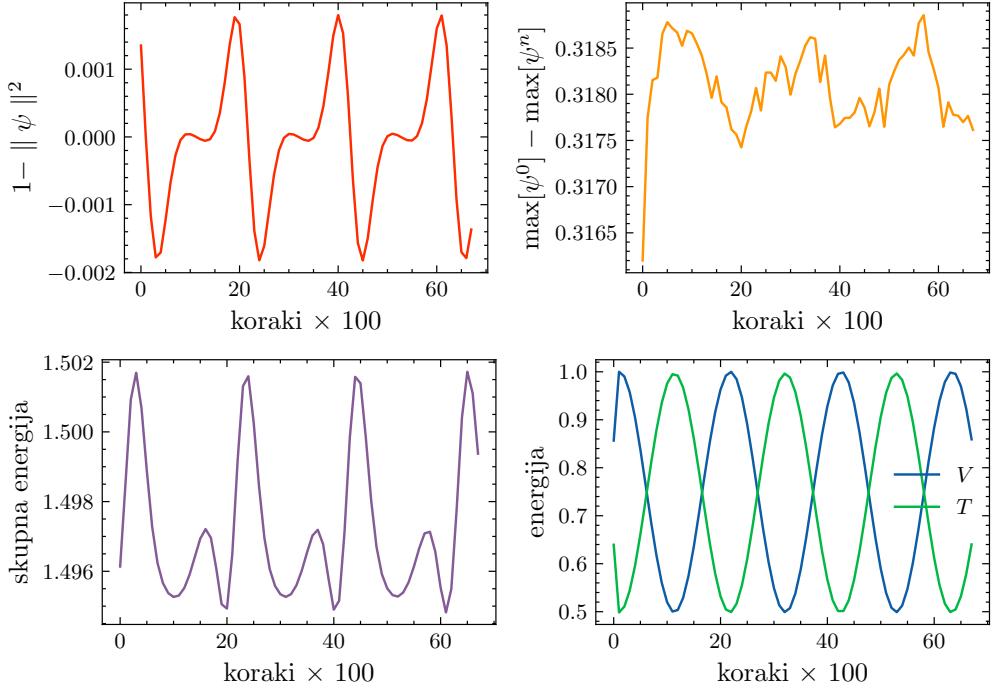
Uporabimo končne difference in dobimo predpis

$$\psi_{ij}^{n+1} = \psi_{ij}^{n-1} - 2i\Delta t \left[-\frac{1}{2h^2} (\psi_{i+1j}^n + \psi_{i-1j}^n + \psi_{ij+1}^n + \psi_{ij-1}^n - 4\psi_{ij}^n) + V_{ij} \psi_{ij}^n \right] . \quad (30)$$

V matrični metodi 27 sem uporabil $\mathbf{D}^{(4)}$ ter zapisal vse z redkimi matrikami. Za pohitritev postopka 30 pa sem si pomagal z `numba.jit`, ki je je just-in-time compiler za Python in Numpy. Na slikah 4 in 5 je enak prikaz kot pri prejšnji metodi. Vidimo, da norma, maksimalna vrednost in energija nihajo okrog neke ravnovesne vrednosti, kar je veliko boljše, kot pa da se večajo ali manjšajo.



Slika 4: Metoda zapisana v matrični obliki pri $\Delta t = 1.14 \times 10^{-3}$.



Slika 5: Iteracijska shema 30 pri $\Delta t = 1.51 \times 10^{-3}$, kar ustreza 6602 časovnima korakoma. Za 6601 časovni korak shema že popolnoma divergira.

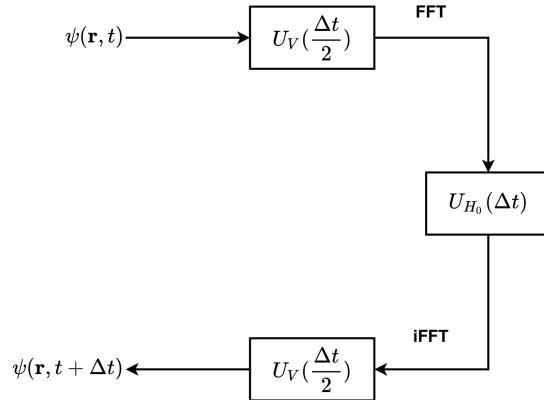
2.4 Spektralna split-step Fourierova metoda (SSFM)

Pri tej metodi uporabimo Fourierovo transformacijo za pretvorbo valovne funkcije v recipročni prostor v katerem velja $\nabla^2 \rightarrow \mathbf{p}^2$. Split-step aproksimacijo 16 izračunamo s FFT, kjer s Fourierovimi transformacijami \mathcal{F} pretvarjammo iz koordinatnega prostora v recipročni prostor ter z inverznimi Fourierovimi transformacijami \mathcal{F}^{-1} iz recipročnega v koordinatni prostor:

$$\psi(\mathbf{r}, t + \Delta t) = \mathcal{F}^{-1} \left[e^{-i\mathbf{p}^2 \Delta t / 2} \mathcal{F} \left[e^{-iV \Delta t} \mathcal{F}^{-1} \left[e^{-i\mathbf{p}^2 \Delta t / 2} \mathcal{F} [\psi(\mathbf{r}, t)] \right] \right] \right] \quad (31)$$

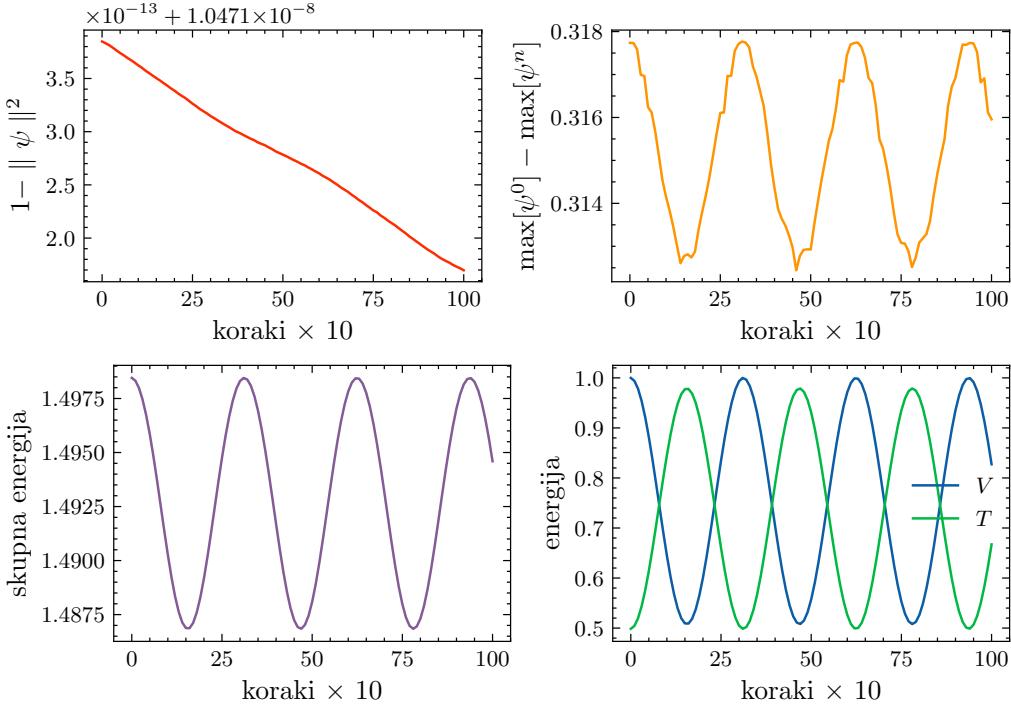
ali pa zapisano v obrnjenem vrstnem redu z manj transformacijami (slika 6):

$$\psi(\mathbf{r}, t + \Delta t) = e^{-iV \Delta t / 2} \mathcal{F}^{-1} \left[e^{-i\mathbf{p}^2 \Delta t} \mathcal{F} \left[e^{-iV \Delta t / 2} \psi(\mathbf{r}, t) \right] \right]. \quad (32)$$



Slika 6: Shematski prikaz Fourierove split-step metode.

Za izvedbo Fourierove transformacije sem uporabil `scipy.fft.fftn` in `scipy.fft.ifftn`. Uporabil sem enačbo 32, saj potrebuje manj klicev FFT in je zato hitrejša. Rezultate prikazuje slika 7. Najprej opazimo, da je metoda stabilna za vse Δt in da deluje že pri velikih vrednosti časovnega koraka v primerjavi s prejšnjimi metodami. Vidimo, da ohranja tudi obliko valovne funkcije in skupno energijo (nihanje okrog ravnovesne vrednosti je precej majhno). Metoda pa ne ohranja norme valovne funkcije. Iz grafa vidimo, da se norma sicer zmanjšuje zelo počasi.



Slika 7: Split-step Fourierova metoda pri $\Delta t = 1.00 \times 10^{-3}$.

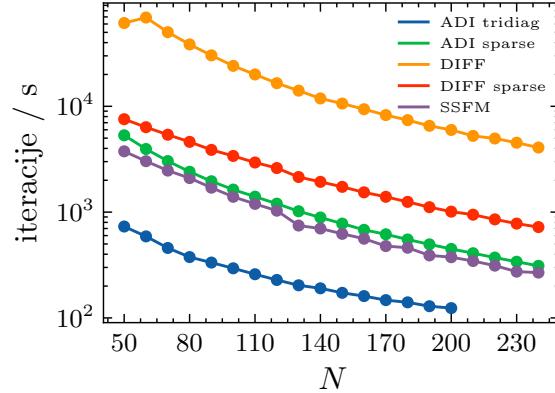
3 Primerjava metod

V spodnji tabeli so prikazani optimalni časovni koraki za vsako od metod:

metoda	# iteracij	Δt
matrična ADI	12117	8.25×10^{-4}
tridiagonalna ADI	11250	8.89×10^{-4}
diference	6602	1.51×10^{-3}
matrične diference	8745	8.14×10^{-3}
split-step Fourier	< 1000	$< 1 \times 10^{-3}$

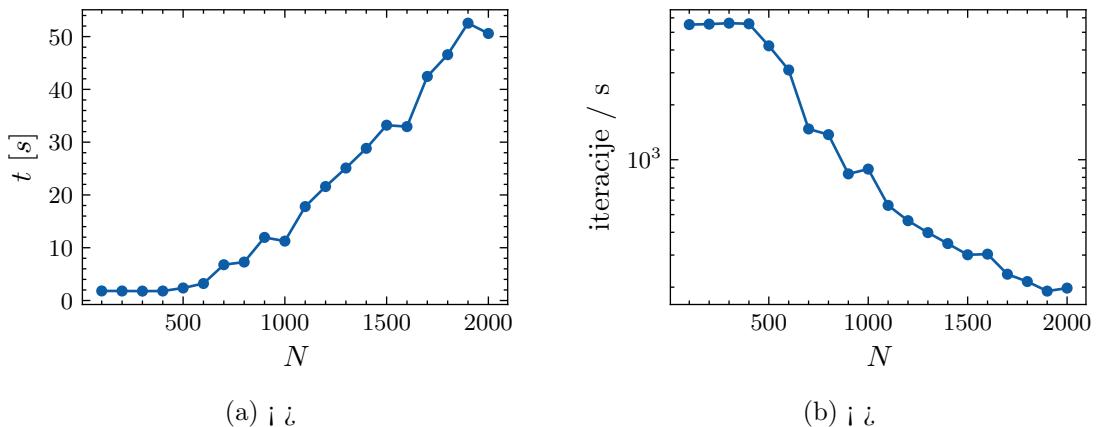
Tabela 1: Dobre vrednosti velikosti časovnega koraka Δt za posamezne metode. Velikost mreže je bila 128×128 , mreža je bila omejena z $L = 5$, parameter λ je bil postavljen na 0, končni čas je bil 10, začetno stanje je bilo izmakneno za 1 v x -smeri, bisekcija je potekala med 1000 in 20000 časovnimi koraki in se je ustavila po 15-ih korakih ali pa po dveh enakih zaporednih vrednostih.

Poglejmo si še hitrosti posameznih metod pri fiksнем časovnem koraku in spreminjajočem se številu mrežnih točk N . Za Δt sem vzel 2×10^{-5} , tako da nobeden od algoritmov ne divergira. Končni čas in postavitev problema je bila enaka kot prej. Na sliki 8 so prikazane iteracije na sekundo (koliko časovnih korakov naredi metoda v danem času) za vsako od metod. Iz grafa vidimo, da so najhitrejše diferenčne metode, najpočasnejše implicitne ter, da je spektralna metoda nekje vmes.



Slika 8: Časovni koraki na sekundo za posamezno metodo.

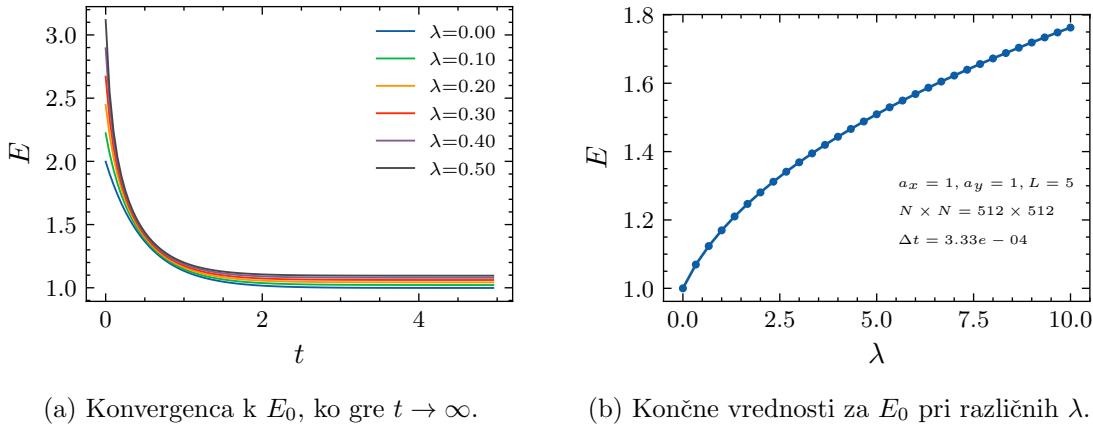
Če še enkrat pogledamo enačbo za split-step Fourierovo metodo 32 opazimo, da v njen nastopajo polne matrike, vektorji, eksponenti in FFT. Računaje vseh teh količin lahko precej pospešimo z uporabo grafične kartice namesto procesorja. To sem storil s pomočjo knjižnice PyTorch ter njenih metod `torch.fft.fft2` in `torch.fft.ifft2`, ki uporabljuje cuFFT. Edina težava je, da metoda ne ohranja norme najboljše, kar vidimo na sliki 7. Majhna izguba norme pri iteraciji se pozna veliko bolj na enojni natačnosti GPU-ja, zato sem valovno funkcijo dodatno renormaliziroval na nekaj korakov. Grafa na sliki 9 prikazujeta hitrosti algoritma izvedenega na grafični kartici, vidimo da izračun rešitev za diskretizacije nad 1000×1000 ne predstavlja bistvenega problema.



Slika 9: $\Delta t = 1 \times 10^{-4}$, število iteracij je 10^4 , renormalizacija na vsak korak.

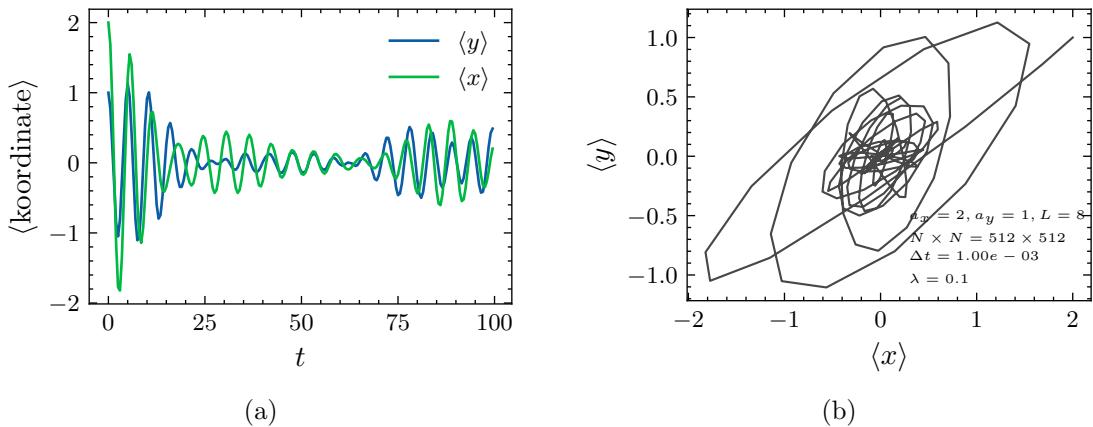
4 Anharmonski oscilator: $\lambda \neq 0$

Iz prejšnjih grafov energij ter iz animacije `ani1.mp4` vidimo, da se (koherentno) stanje izmakneno za nek (a_x, a_y) pri $\lambda = 0$ obnaša semi-klasično, torej oscilira na enak način kot klasični delec v harmonskem potencialu. Poglejmo si še kaj se zgodi, ko povečujemo λ . Energijo osnovnega stanja E_0 sistema sem dobil s propagacijo v imaginarnem času v limiti $t \rightarrow \infty$. To sem storil z zamenjavo $\Delta t \rightarrow -i\Delta\tau$ v enačbi 32. Slika 10a prikazuje konvergenco energije k E_0 , ko propagiramo začetno stanje. Na sliki 10b so prikazani končni rezultati za energijo osnovnega stanja pri različnih vrednostih λ . Za $\lambda = 0$ dobimo vrednost 1, kar je brezdimenzijska E_0 dvodimenzionalnega harmonskega oscilatorja. Za večje λ pa se energija povečuje.

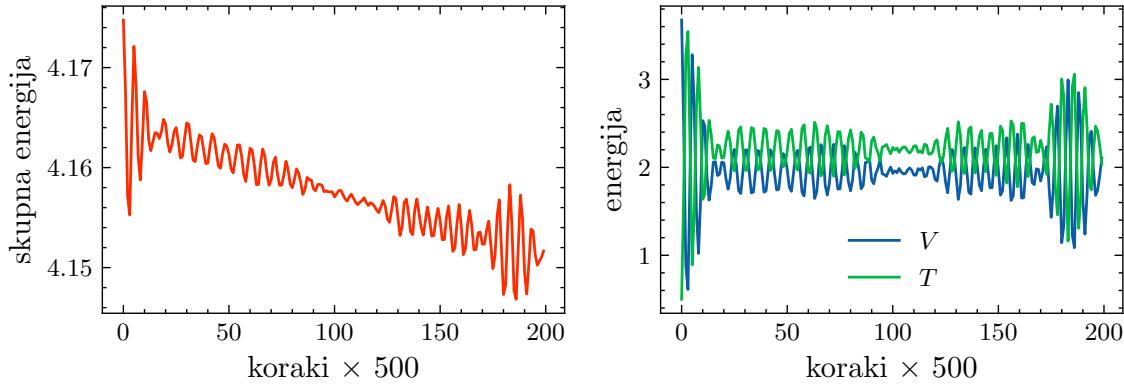


Slika 10: Izračun energije osnovnega stanja.

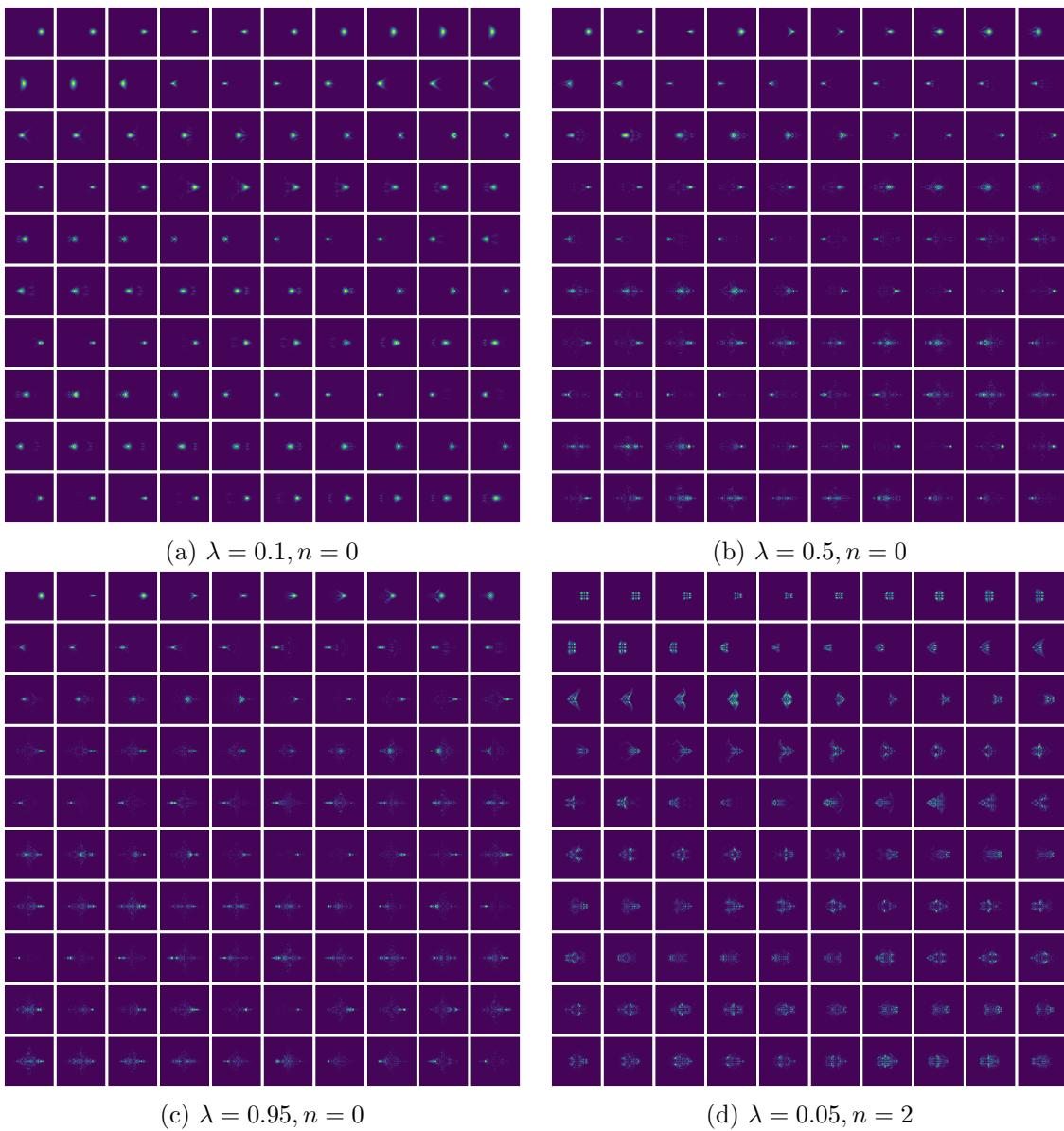
Primer dinamike za $\lambda = 0.1$ je prikazan na sliki 11 ter na animacijah `ani3.mp4` in `ani4.mp4`. Energija za ta primer pa je prikazana na sliki 12. Animacija `ani2.mp4` prikazuje, kaj se zgodi s stanjem, ki ni izmakneno. Ne-izmakneno stanje se ne premika, zaradi dodatnega člena $\lambda \neq 0$ pa amplituda stanja oscilira. Za vsa izmaknjena stanja je značilno, da se po dolgem času razlezejo in končajo na sredini, kjer je potencial najšibkejši. Evolucijo $\|\psi\|^2$ pri različnih vrednostih λ prikazujejo slike 13. Na zadnji sliki je še prikaz začetnega stanja 3 pri katerem vzamemo $n = 2$.



Slika 11: Spreminjanje pričakovanih vrednosti koordinate v času.



Slika 12: Spreminjanje energije stanja pri $\lambda = 0.1$.



Slika 13: Razvoj $||\psi||^2$ v času pri parametrih $N \times N = 1024 \times 1024$, $L = 15$, $a_x = 5$, $a_y = 0$ število časovnih korakov: 50000, končni čas: 20 in različnih λ .

Literatura

- [1] https://en.wikipedia.org/wiki/Crank%20%93Nicolson_method
- [2] https://en.wikipedia.org/wiki/Alternating-direction_implicit_method
- [3] Two-Particle Schrödinger Equation Animations of Wavepacket–Wavepacket Scattering (revised) <https://arxiv.org/abs/physics/9909042v1>
- [4] An introduction to the Split Step Fourier Method using MATLAB https://www.researchgate.net/publication/281441538_An_introduction_to_the_Split_Step_Fourier_Method_using_MATLAB
- [5] GPU-accelerated solutions of the nonlinear Schrödinger equation <https://arxiv.org/abs/2010.15069>
- [6] Massively parallel split-step Fourier techniques for simulating quantum systems on graphics processing units <https://core.ac.uk/download/pdf/276535222.pdf>