

---

# 1. naloga: Numerično reševanje Schrödingerjeve enačbe za en delec

---

POROČILO PRI PREDMETU VIŠJE RAČUNSKE METODE

2024/25

*Avtor:*

Rok MLINAR VAHTAR  
28242004

28. februar 2025

# 1 UVOD

V tej nalogi si bomo pogledali, kako lahko numerično rešujemo Schrödingerjevo enačbo na tri različne načine. Specifično nas bo zanimal časovni razvoj enodimenzionalne valovne funkcije v harmoničnem in anharmoničnem oscilatorju.

$$\partial_t \psi(x, t) = -i\hat{H}\psi(x, t) \qquad \hat{H} = \frac{1}{2}\hat{p}^2 + \frac{1}{2}\hat{x}^2 + \lambda\hat{x}^4$$

## 1.1 KONČNE DIFERENCE

Prva metoda reši problem tako, da prostor-čas diskretizira na mreži kot

$$\psi_{m,n} = \psi(x = hm, t = \tau n),$$

odvode pa prevede v končne difference. Mi bomo uporabili

$$(\partial_t \psi)_{m,n} = \frac{\psi_{m,n+1} - \psi_{m,n}}{\tau} + \mathcal{O}(\tau)$$

in

$$(\partial_x^2 \psi)_{m,n} = \frac{\psi_{m+1,n} - 2\psi_{m,n} + \psi_{m-1,n}}{2h^2} + \mathcal{O}(h^2)$$

oziroma

$$(\partial_x^2 \psi)_{m,n} = \frac{-\psi_{m+2,n} + 16\psi_{m+1,n} - 30\psi_{m,n} + 16\psi_{m-1,n} - \psi_{m-2,n}}{12h^2} + \mathcal{O}(h^4).$$

Te zveze nam dajo sistem enačb, ki določi vrednost  $\psi_{m,n}$  kot linearno kombinacijo različnih  $\psi_{m',n-1}$ , torej le iz preteklosti. Z začetnim pogojem lahko tako konstruiramo vsak naslednji trenutek.

## 1.2 SKOKI S KONČNIM PROPAGATORJEM

Drugačen zapis časovnega razvoja valovne funkcije je

$$\psi(x, t + \tau) = e^{-i\tau\hat{H}}\psi(x, t).$$

Ta predpis lahko diskretiziramo in tako dobimo

$$\psi_{m,n+1} = \left( e^{-i\tau\hat{H}}\psi(x, t)\psi \right)_{m,n} = \sum_{k=0}^K \frac{(-i\tau)^k}{k!} \left( \hat{H}^k \psi \right)_{m,n} + \mathcal{O}(\tau^{K+1}),$$

kjer vrsto končamo pri nekem dovolj velikem  $K$ . Sama vrednost  $K$  je odvisna od problema, natančnosti ki jo želimo in težavnosti izvrednotenja  $\hat{H}^k \psi$ . Ker je izvrednotenje v našem primeru, po tem ko odvod  $p = -i\partial_x$  izvrednotimo z končno diferenco, le množenje matrik, lahko vzamemo velik  $K$ , recimo 10.

### 1.3 IMPLICITNA SHEMA

Obe prejšnji metodi imata isti problem, nista unitarni. To pomeni, da bo rešitev po dolgem času popolnoma izgubila normalizacijo. Tu pa nas reši malo drugačen približek za eksponentno funkcijo. Velja namreč

$$e^{-i\tau\hat{H}} = (1 + i\frac{\tau}{2}\hat{H})^{-1}(1 - i\frac{\tau}{2}\hat{H}) + \mathcal{O}(\tau^3),$$

kjer je  $(1 + i\frac{\tau}{2}\hat{H})^{-1}(1 - i\frac{\tau}{2}\hat{H})$  unitaren. Tako lahko zapišemo shemo

$$\left( (1 + i\frac{\tau}{2}\hat{H})\psi \right)_{m,n+1} = \left( (1 - i\frac{\tau}{2}\hat{H})\psi \right)_{m,n},$$

ki pa zdaj meša več  $\psi_{m,n}$  iz prihodnosti in preteklosti. Zato ne moremo zapisati eksplicitne formule, ampak moramo reševati tri ali večdiagonalni sistem, odvisno od prostorske končne difference, ki jo izberemo. Desna stran enačbe je le nek vektor, ki ga poznamo, saj imamo začetni pogoj, leva pa je le neka matrika pomnožena z neznanim vektorjem. Tak sistem oblike  $A\vec{x} = \vec{b}$  rešujemo preko Gaussovske eliminacije ali kakšnega drugega ekvivalentnega algoritma. Sam sem uporabil program `scipy.linalg.solve_banded`.

## 2 RAZVOJ LASTNIH STANJ HARMONIČNEGA OSCILATORJA V ANHARMONSKEM OSCILATORJU

V prvem delu naloge nas bo zanimalo, kako se obnašajo lastna stanja harmonskega oscilatorja ( $\lambda = 0$ )

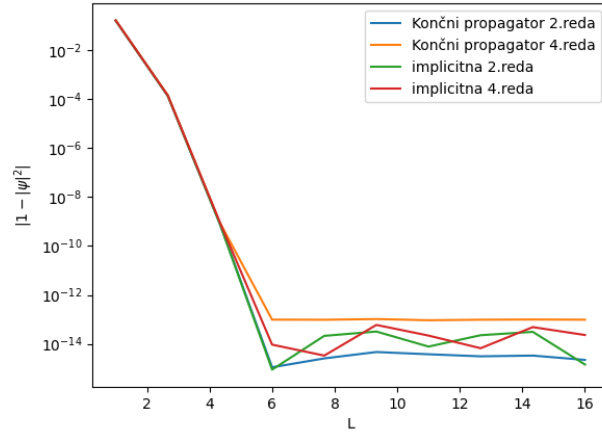
$$\Phi_N(x) = \frac{1}{\pi^{\frac{1}{4}}\sqrt{2^N N!}} H_N(x) e^{-\frac{x^2}{2}},$$

ko jih postavimo v anharmonski oscilator ( $\lambda \neq 0$ ).

### 2.1 PREISKUS METOD

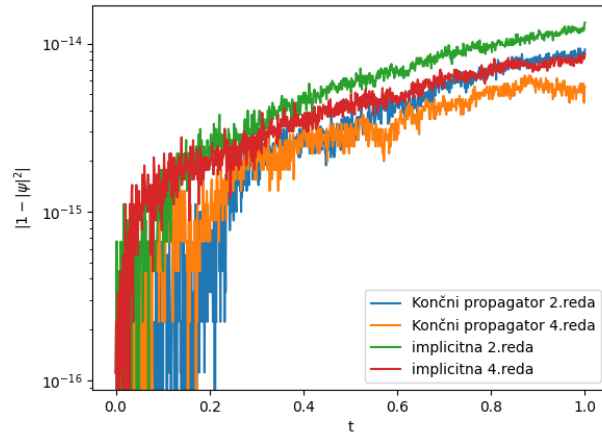
Preden poskusimo odkriti kaj novega, preverimo če naše metode sploh delujejo. Ker so  $\Phi_N(x)$  lastna stanja, vemo da se za  $\lambda = 0$  s časom spreminjajo kot  $\Phi_N(x, t) = \Phi_N(x) \cdot e^{-iE_N t}$ , kar lahko primerjamo z numeričnimi rezultati.

Prva odločitev, ki nas čaka je, kako bomo diskretizirali prostor. Ker ne moremo vzeti intervala  $[-\infty, \infty]$  se moramo zadovoljiti z neko končno mejo. Kakršnakoli smiselna meja mora vsaj ohraniti normalizacijo osnovnega stanja, torej  $\int_{-L}^L \frac{1}{\sqrt{\pi}} e^{-x^2} dx = 1 - \varepsilon$ , kar nam da zvezo  $\text{erf}(L) = \varepsilon$ . Če želimo da je vsaj 99.99% valovne funkcije zaobjete dobimo  $L = \text{erf}^{-1}(0.9999) \approx 2.751$ . Za vsak slučaj to podvojimo in prištejmo 10, saj bomo osnovno stanje kasneje zamikali. Vzemimo torej  $L = 15$ . To izbiro pa lahko tudi preverimo, če pogledamo kako velik mora biti  $L$ , da ne vpliva več na rešitev. Če pogledamo kako se končna normalizacija osnovnega stanja po evoluciji za eno časovno enoto spreminja za različne izbire  $L$  dobimo graf 1. Vidimo, da bi že  $L = 6$  zadoščal, da bomo kasneje lahko nastavili zamik tudi do  $a = 10$ , pa je izbira  $L = 15$  za zdaj dobra, saj nas za zdaj ne stane preveč na času računanja.



Slika 1: Ohranitev normalizacije v odvisnosti od  $L$

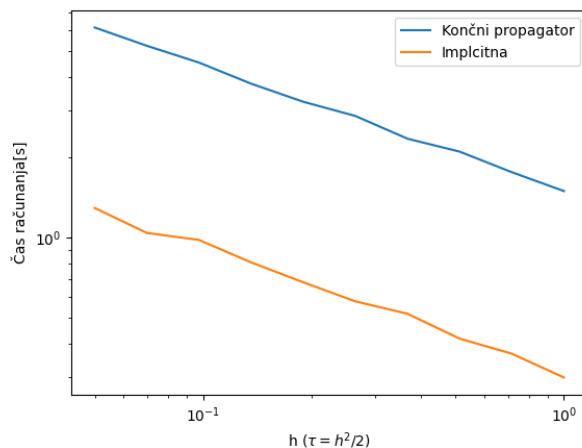
Dobljene rezultate za osnovno stanje sem shranil v priloženih datotekah `animacija1.mp4`, `animacija2.mp4` in `animacija3.mp4`. Vidimo lahko, da čista metoda končnih diferenc ni stabilna, ostali dve pa delujeta zelo dobro. Kako dobro delujeta pa lahko preverimo še bolj podrobno, če preverimo kako ohranjata normalizacijo. Na grafu 2 lahko vidimo, da sta obe metodi približno enako dobri pri ohranjanju lastnega stanja. Metode so ločene na 2. in 4. red glede na to, katero končno diferenco uporabljajo v prostoru.



Slika 2: Ohranitev normalizacije z metodami

Za končno odločitev pa je relevantno tudi, kako hitra je metoda. Iz slike 3 je jasno, da je tu implicitna metoda boljša izbira. Prepričan sem, da bi se dalo tudi metodo z končnim propagatorjem optimizirati,

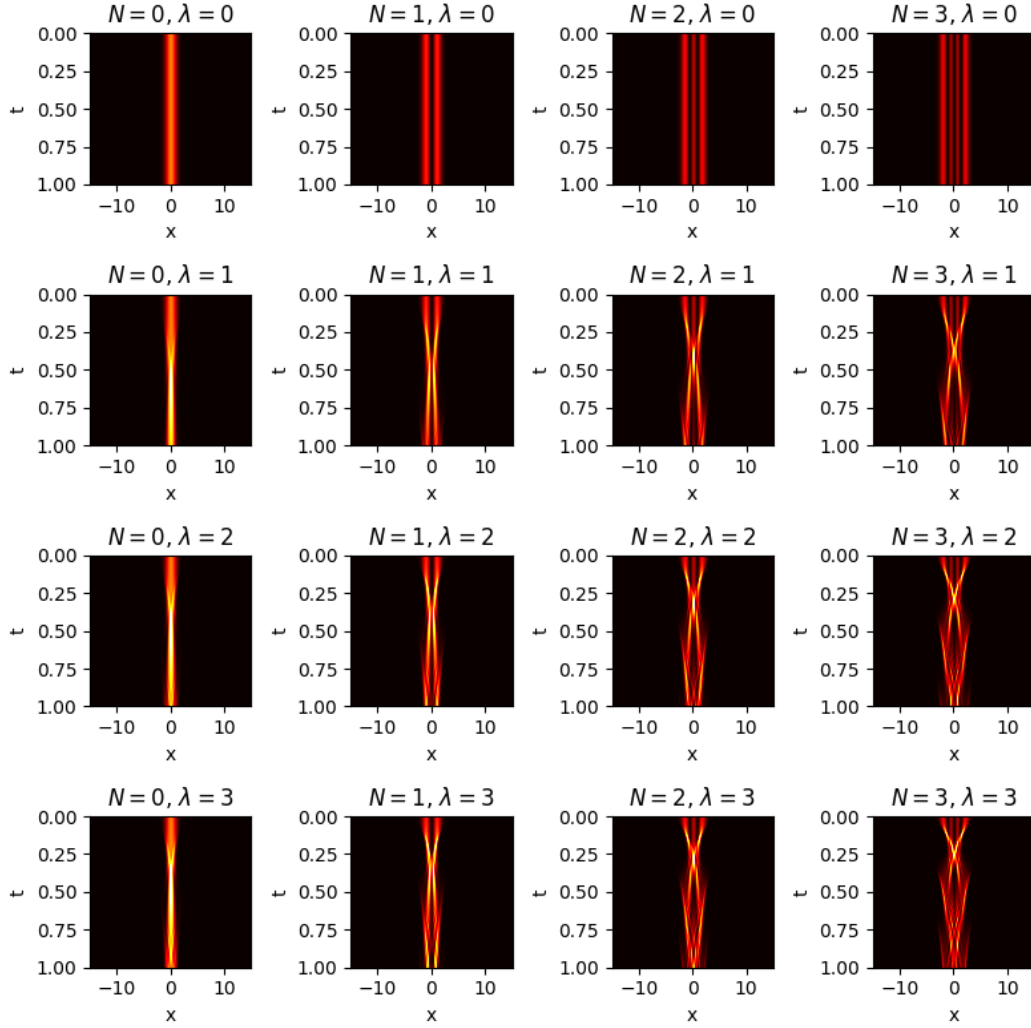
ampak bi to vzelo več časa kot bi ga prihranili, če kar vzamemo implicitno metodo, katere jedro, scipy-jeva metoda `solve_banded`, je že zelo dobro optimizirana za večdiagonalne matrike.



Slika 3: Čas računanja za različni metodi

## 2.2 RAZVOJ LASTNIH STANJ

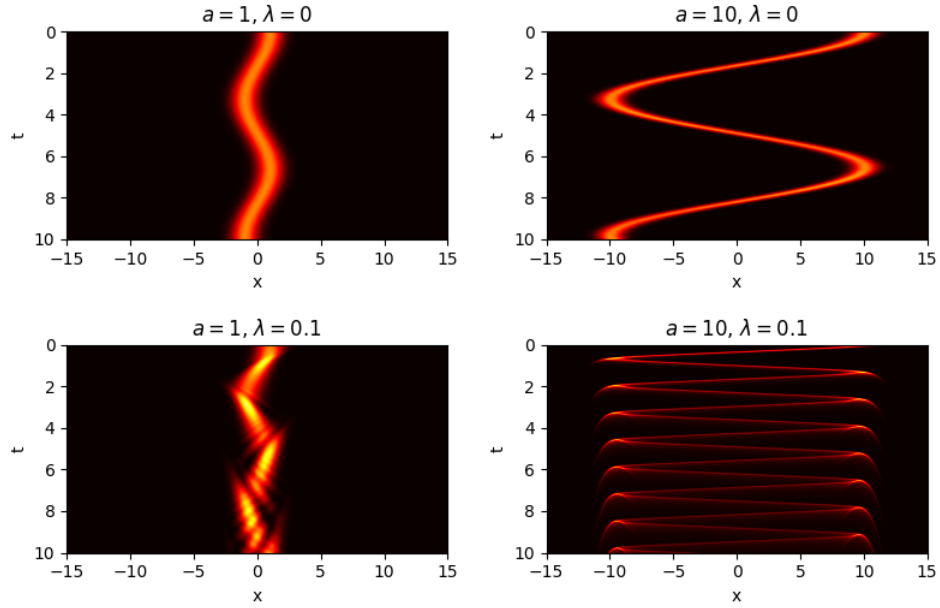
Zdaj, ko smo se prepričali o delovanju metod, se lahko končno lotimo naše dejanske naloge. Kot prvo nas bo zanimalo, kako se različna lastna stanja harmonskega oscilatorja razvijajo v anharmonskem potencialu. Ker je potencial  $x^4$  ostrejši od  $x^2$  bi pričakovali, da bi se stanja bolj zožila, kar tudi vidimo na sliki 4. Osnovno stanje se tu skrči, medtem ko višja bolj opazno zanihajo. Za boljšo predstavbo sem dodal tudi animacije `animacija4.mp4`, `animacija5.mp4` in `animacija6.mp4` z razvojem prvih treh lastnih stanj pri  $\lambda = 1$ .



Slika 4: Časovni razvoji verjetnostnih gostot različnih lastnih stanj pri različnih  $\lambda$

### 3 RAZVOJ KOHERENTNEGA STANJA V ANHARMONSKEM POTENCIALU

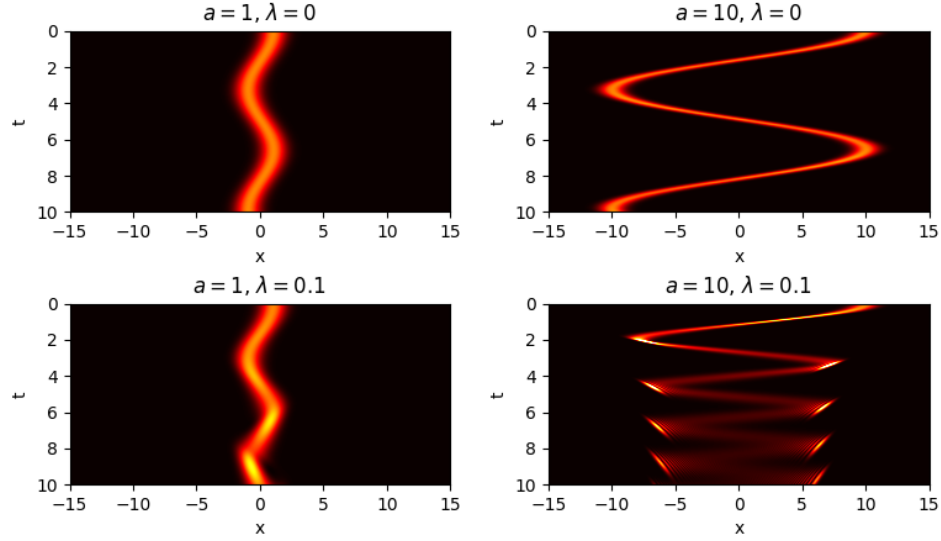
Koherentna stanja harmonskega oscilatorja so le v prostoru zamaknjena osnovna stanja in so znana po tem, da se njihove pričakovane vrednosti obnašajo tako, kot klasičen delec ujet v istem potencialu. V primeru  $\lambda = 0$  lahko zato pričakujemo, da bo stanje nihalo levo in desno, brez da bi spreminjalo obliko, kar tudi vidimo na sliki 5.



Slika 5: Časovni razvoji verjetnostnih gostot različno zamaknjenih koherentnih stanj pri različnih  $\lambda$

V potencialu z neničelnim  $\lambda$  pa opazimo podobno nihanje z nekaj spremembami. Prva opazna sprememba je to, da se valovni paket razvleče in izgubi gaussovsko obliko. Pri večji amplitudi  $a = 10$  pa opazimo tudi to, da anharmonski potencial poveča frekvenco nihanja, kar je razumljivo, saj močnejši potencial potiska paket z večjo sila in počasi manjša amplitudo, saj se rešitev tako kot pri lastnih stanjih koncentrira v območju z manjšim potencialom.

Za konec sem pogledal še primer časovno odvisnega hamiltoniana, za katere moramo metode minimalno dodelati. Vse kar moramo storiti je to, da metodi pri vsakem koraku podamo drugačen potencial  $V$ . Za bolj podroben ogled je na voljo [animacija9.mp4](#) pri  $a = 1$  in  $\lambda = 0.1$ .



Slika 6: Časovni razvoji verjetnostnih gostot različno zamaknjenih koherentnih stanj pri različnih končnih  $\lambda$ , kjer vrednost  $\lambda$  začne pri 0 in linearno raste.

Na sliki 6 lahko še bolj izrazito vidimo vpliv  $\lambda$  na amplitudo.

#### 4 DVODIMENZIONALNI ANHARMONSKI OSCILATOR

Za razliko od prejšnjih dveh nalog imamo zdaj opravka z dvodimenzionalnim problemom, kar pomeni, da prej spisane metode odpadejo, saj so narejene za 1D? To morda ni čisto res. Če diskretiziramo prostor tako kot prej dobimo 2D matriko  $\psi_{kl}(t) = \psi(x = hk, y = hl, t)$  namesto 1D vektorja, ki smo ga imeli prej. To matriko pa še vedno lahko zložimo v vektor tako da stolpce zlagamo enega na drugega. To pa seveda pomeni, da moramo tudi matriko hamiltoniana zapisati drugače. Zapisal bom primer za  $3 \times 3$  diskretizacijo v prostoru z končno diferenco 2. reda, vzorec pa se zelo enostavno sploši na  $N$

$$\partial_x^2 \psi = \frac{1}{2h^2} \begin{bmatrix} -2 & 1 & & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & \\ & & & -2 & 1 \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} \psi_{11} \\ \psi_{12} \\ \psi_{13} \\ \psi_{21} \\ \psi_{22} \\ \psi_{23} \\ \psi_{31} \\ \psi_{32} \\ \psi_{33} \end{bmatrix}$$



$$\partial_y^2 \psi = \frac{1}{2h^2} \begin{bmatrix} -2 & & & 1 & & & \\ & -2 & & & 1 & & \\ & & -2 & & & 1 & \\ 1 & & & -2 & & & 1 \\ & 1 & & & -2 & & \\ & & 1 & & & -2 & \\ & & & 1 & & & -2 \\ & & & & 1 & & \\ & & & & & -2 & \\ & & & & & & -2 \end{bmatrix} \begin{bmatrix} \psi_{11} \\ \psi_{12} \\ \psi_{13} \\ \psi_{21} \\ \psi_{22} \\ \psi_{23} \\ \psi_{31} \\ \psi_{32} \\ \psi_{33} \end{bmatrix}$$

$$V(x, y) \psi = \frac{1}{2h^2} \begin{bmatrix} V_{11} & & & & & & \\ & V_{12} & & & & & \\ & & V_{13} & & & & \\ & & & V_{21} & & & \\ & & & & V_{22} & & \\ & & & & & V_{23} & \\ & & & & & & V_{31} \\ & & & & & & & V_{32} \\ & & & & & & & & V_{33} \end{bmatrix} \begin{bmatrix} \psi_{11} \\ \psi_{12} \\ \psi_{13} \\ \psi_{21} \\ \psi_{22} \\ \psi_{23} \\ \psi_{31} \\ \psi_{32} \\ \psi_{33} \end{bmatrix}$$

Z matriko  $H = -\partial_x^2 - \partial_y^2 + V(x, y)$  lahko zdaj uporabimo iste algoritme kot prej. Jaz sem izbral implicitnega, saj potrebujemo vso stabilnost, ki jo lahko dobimo, brez da bi pretirano zmanjšali korak, za kar je pi mojih izkušnjah implicitna metoda ponavadi najboljša. Glede hitrosti, je problem načeloma še vedno večdiagonalen, tako da bi metoda `numpy.linalg.solve_banded` lahko delovala dobro, če nebi že samo shranjevanje matrike zavzelo nekaj gigabitov prostora za kakršen koli primerno majhen korak v prostoru in času. Zato moramo, če nimamo dostopa do primerljivo velikega računalnika uporabiti redke oz. `scipy.sparse` matrike, ki ne rezervirajo prostora za vse ničle, katerih imamo v naših matrikah na pretek, ampak samo za mesta ki so dejansko zasedena. To pa pomeni tudi, da so nekatere operacije z njimi na ta račun počasnejše.

Na koncu se je izkazalo, da moj računalnik ni dovolj dober, da bi simulacijo izvedel pri zadostno gosti diskretizaciji, brez da bi se posamezna animacija generirala vsaj več kot nekaj ur. Zato rešitev za koherentno stanje pri  $\lambda = 0$  (`animacija_2D_la=0.mp4`), ki bi morala biti le nihajoča translacija, zelo hitro izgubi obliko in se razleze k mejam, ki jih tudi nisem moral postaviti dlje stran. Vseeno pa sem poskusil primerjati rešitve z različnimi  $\lambda$ , v upanju, da je vsaj čisti začetek relativno točen. Drugi priloženi posnetek pri  $\lambda = 0$  in  $a = 1$  (`animacija_2D_la=0.1.mp4`), pa pokaže, da se rešitev zaradi večjega potenciala drži v ožjem območju zvezdaste oblike.

## 5 ZAKLJUČEK

Ta naloga je bila zanimiva, ampak je bila tut kar hud test potrpežljivosti, saj so se grafi in animacije v povprečju računali vsaj 20min. Po drugi strani pa je bilo zanimivo iskati optimizacije, saj se je tudi

najmanjša izboljšava čutila pri tako velikem številu ponovljenih izračunov.