
1.1 Neposreden časovni razvoj 1D Schrödingerjeve enačbe

MIHA SRDINŠEK

I. UVOD

Na predavanjih nam je profesor predstavil tri metode s katerimi naj poskušamo najti rešitev Shrödingerjeve enačbe. Ukvarjali smo se s časovnim razvojem enodimenzionalne valovne funkcije, če se nahaja v nekem potencialu. V tej predstavitvi bom predvsem predstavil moje delo in ne teorije, ki jo je že predstavil profesor in jo lahko najdeš na spletni strani ali v mapi VR. Program sem pisal v jeziku *Python*.

II. ALGORITMI

- Najprej sem se lotil metode končnih diferenc (DIF). Program sem zapisal tako, da sprejme kot argumente velikost časovnega in krajevnega koraka, ter velikost območja in čas do katerega naj se sistem razvije. Dolžino časovnega koraka sem najprej nastavljal na roko, a hitro sem ugotovil, da moram paziti na pravilo

$$\frac{\tau}{h^2} < \frac{1}{2}, \quad (1)$$

če sta τ in h časovni in krajevni korak. Nato sem določil mrežo na kateri je kraj vodoravna os in čas vertikalna. To mrežo sem predstavil kot matriko. Začel sem z eno vrstico in potem dodajal nove. Na prvo vrstico sem nastavlil začetni pogoj pri čemer sem robne pogoje poenostavil tako, da sem od celotnega začetnega pogoja odštel vrednost na robni točki začetnega pogoja. Potem sem zapisal zanko in rešil enačbo

$$\psi_{m,n+1} = \psi_{m,n} + i\tau \left\{ \frac{1}{2h^2} (\psi_{m+1,n} + \psi_{m-1,n} - 2\psi_{m,n}) - V_m \psi_{m,n} \right\} \quad (2)$$

za vsak košček. Ker sem v začetnih primerih dobival zelo slabe rezultate, ki so ves čas divergirali, sem v vsakem koraku, vrstico še normiral. To se je na koncu mislim da izkazalo za nepotrebno oz. škodljivo, ker sem verjetno slabo normiral.

- Potem sem se lotil druge metode (SkokK). Lotil sem se je tako, da sem operator \hat{H} zapisal v matrični obliki. Pri čemer sem ga zapisal z metodo končnih diferenc. Uporabljal pa sem

samo trotočkovni razvoj odvoda. Pri tem sem zapisal \hat{H} kot matriko, ki ima v zgornji in spodnji vrstici same ničle. Definiral sem funkcijo, ki je do reda K aproksimiral eksponent operatorja in med eksponenciacijo so se naravno upoštevali robni pogoji, saj je \hat{H}^0 identiteta, kar je pomenilo da je končna matrika robni točki zgolj prekopirala. Pri tem sem dolžino časovnega koraka določil s pomočjo zahteve, ki smo jo zapisali na predavanjih

$$\tau < \frac{2\pi}{\|\hat{H}\|_\infty}, \quad (3)$$

pri čemer sem lenobno za normo vzel kar neskončno normo, ki bi morala biti po moji oceni približno enako velika ali manjša kot operatorska. Na koncu se je to ujemalo z napovedjo profesorja, da velja

$$\|\hat{H}\| \sim h^{-2}, \quad (4)$$

saj se je τ določen po takem postopku razlikoval od mojega za faktor 3. A načeloma bi moral v vsakem koraku poračunati τ . Kar moj postopek rešuje je visok K , ki potem relativizira izbor τ . Od tu se je postopek odvil identično, kot prej, le da sem novo vrstico, ob novem času, napovedoval z množenjem matrike s prejšnjo vrstico

$$\psi_{m,n+1} = (e^{-i\tau\hat{H}})\psi_{m,n}. \quad (5)$$

Tudi tu sem poskušal z normiranjem, a se je izkazalo za nepotrebno. Večinoma sem računal z razvojem do 10 reda in metoda se je izkazala za izredno hitro in tudi precej natančno.

- Tretjo (Implic) metodo sem začel podobno. Določil sem \hat{H} z metodo končnih diferenc in poračunal matriko

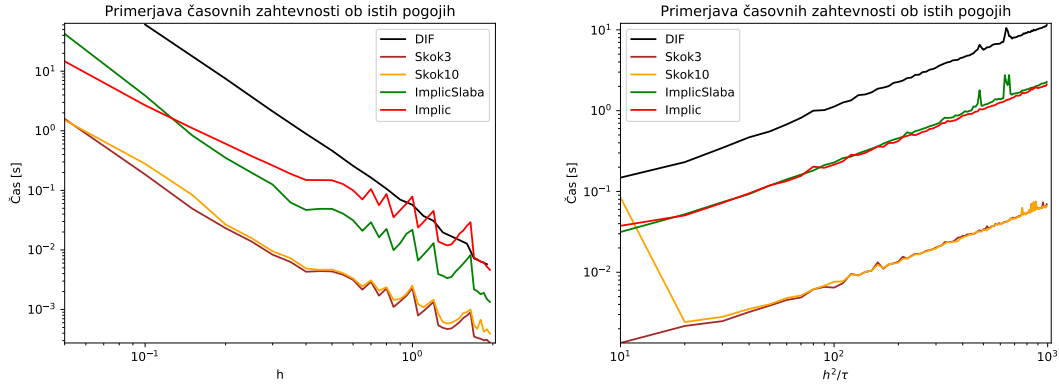
$$\left(1 + i\frac{\tau}{2}\hat{H}\right) \quad (6)$$

in njeno konjugirano prijateljico. Od tu naprej sem uporabil že napisani algoritem, ki reši matrični linearni sistem enačb. Šele kasneje sem zbral voljo in metodo močno pohitril s tem, da sem uporabil rutino prilagojeno tridiagonalnemu sistemu. Metode bi se lahko lotil tudi z razredčenimi matrikami, a sem ocenil, da s tem postopka nebi več bistveno pohitril.

III. PRIMERJAVA METOD

I. Časovna zahtevnost

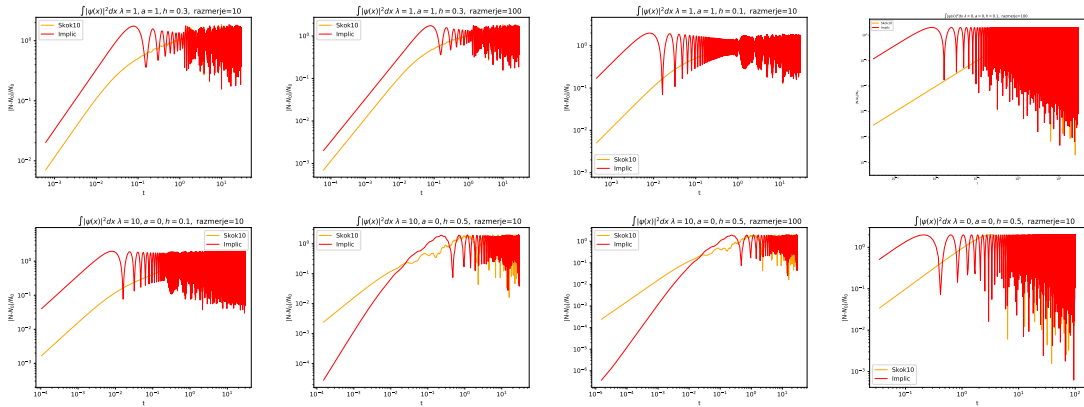
Kar se tiče časovne učinkovitosti, me je presenetila metoda skokov s končnim propagatorjem, ki se je izkazala za izredno hitro. V resnici je to pričakovano, saj smo morali zgolj v nekem koraku skonstruirati matriko, od takrat naprej pa je celotni algoritem zgolj množenje matrik. Kot pričakovano opazimo, da je prilagojena implicitna metoda dosti itrejša od neprilagojene. Opazimo tudi, da K ne vpliva na časovno učinkovitost metode, tako da ga lahko izberemo precej velikega.



Slika 1: DIF predstavlja metodo končnih diferenc, SkokK metodo skokov s končnim propagatorjem pri izbranem $K \in 3, 10$, Implic pa implicitno metodo.

II. Unitarnost

Naslednja stvar, ki nas pri metodah v kvantni mehaniki lahko močno zanima, je ohranjanje števila delcev. Ohranitev števila delcev v kvantni mehaniki pa sledi iz unitarnosti hamiltonjana. To prva metoda in nižji redi druge metode močno kršita, Zato za tiste metode tega niti nisem raziskoval, lahko pa si pogledate **posnetek1**. Na posnetku se recimo vidi kako močno je stabilnost metode in unitarnost odvisna od razmerja τ/h^2 .



Slika 2: Skok10 predstavlja metodo skokov s končnim propagatorjem pri $K = 10$, Implic pa implicitno metodo.

Za preostali dve metodi, pa sem se kanček resneje lotil preučevanja unitarnosti, kar si lahko ogledate na primerjavah zgoraj (slika 2). V splošnem sem ugotovil, da je druga metoda mogoče kanček boljša v tem oziru, a načeloma mislim, da iz rezultatov sledi, da ta obe dobri pri ohranjanju števila.

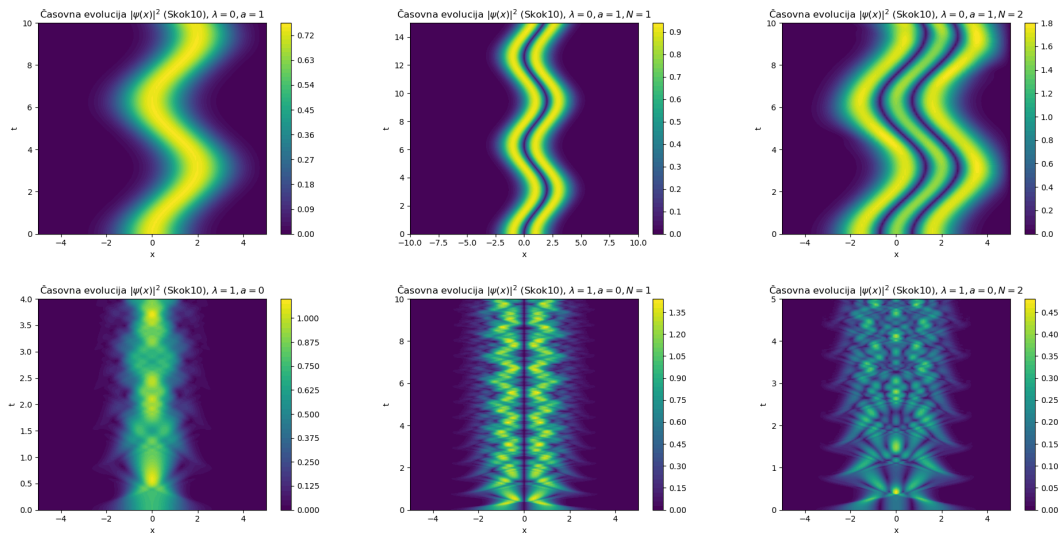
III. Stabilnost

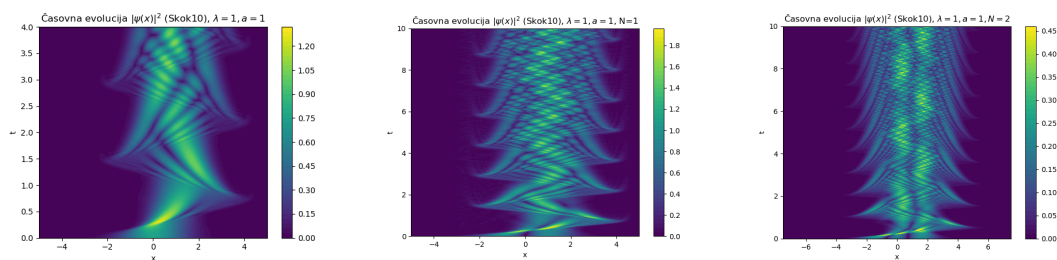
Glede Stabilnosti metod se nisem spomnil učinkovitega dvodimenzionalnega načina primerjanja metod. Ena izmed potencialno zanimivih opazljivk bi bila kinetična energija, ampak že rezultati na sliki 2 se mi zdijo dobro merilo stabilnosti. Zato sem pripravil nekaj posnetkov, ki so se mi zdeli zanimivi. Prvi, ki sem ga že omenil (**posnetek1**), nam prikazuje kako na metodo končnih diferenc vpliva razmerje med τ in h^2 . To bi se dalo kvantificirati z recimo opazovanjem gibanja pričakovana vrednosti. Ko bi začela pričakovana vrednost močneje odstopati od klasične napovedi, bi tako rekli, da smo dosegli nestabilnost. Na tem posnetku sem uporabil metodo končnih diferenc z normiranjem na vsakem koraku, zato je tudi "regilarna" rešitev nepravilna. Ampak opazke glede stabilnosti ostanejo iste.

Lep prikaz tega kaj pomeni, da je metoda nestabilna, vidimo na **posnetek2**, **posnetek3** in **posnetek**. Sploh na **posnetek3** in **posnetek** vidimo, da lahko metoda odpove samo na nekem območju in preostanek območja, še vedno dobro napoveduje. Na tem posnetku tudi postane jasno, zakaj je normiranje, sploh slabo normiranje, slabše od nenormiranja. Sredinsko območje je namreč prikazovalo pravo rešitev, med tem ko so robovi divergirali. Z normiranjem smo samo spravili napake iz robov v notranjost, namesto da bi počakali, da napaka sama pripotuje do njih. Na **posnetek** zaradi neokusno nizke natančnosti vidimo tudi razlike med metodama *Implic* in *Skok10*.

IV. REZULTATI

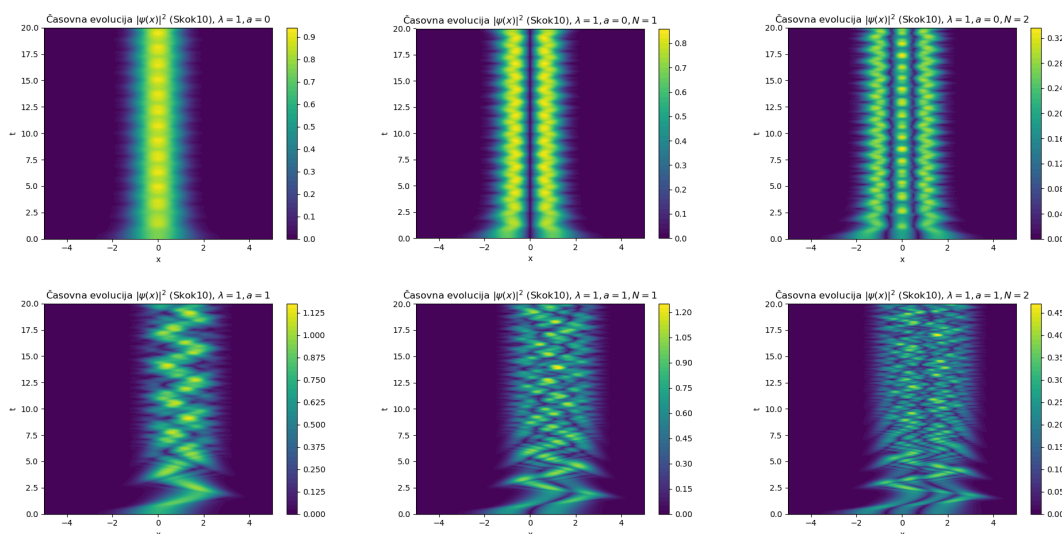
Nato sem še izrisal nekaj rezultatov. Tak lep grafični prikaz, kako povečanje potenciala ali pa zamik potenciala spremeni rešitev, vidimo na **posnetek4** in **posnetek5**. Če oba učinka združimo pa dobimo **posnetek6** in **posnetek9**. Ker morda gledanje videoposnetkov ni najbolj časovno učinkovito sem v to poročilo prilepil tudi nekaj slik 3.





Slika 3: Časovna evolucija lastnih stanj osnovnega potenciala v različnih potencialih.

Potem pa sem puskušal vklop novega potenciala izvesti nežneje. Kar namreč vidimo na zgorjih rezultatih je nasilen prekop iz osnovnega potenciala v hudo pertrubacijo. Moja ideja je torej, da bi ta potencial počasi vkapljal. Iz nič na zeleno vrednost. To pomeni, da sem prilagodil tretjo metodo. Tretjo metodo sem si izbral, ker se mi je zdela najprimernejša za tak podvig, saj je hitra, natančna in ima prostor za spreminjanje potenciala. Če bi potencial spreminjal na vsakem koraku druge metode, bi moral v vsakem koraku izvajati eksponenciacijo, kar bi verjetno hudo upočasnilo metodo. Zdaj pa sem uvedel časovno odvisnost potenciala tako, da sem ga v vsakem koraku na novo poračunal. Sklopitveno konstanto sem linearno prižigal v času. Pri tem sem funkcijo razdelil na dve zanki. Prvo zanko, ki počasi prižge polni potencial, in drugo zanko, ki potem pri polnem potencialu računa po sterem načinu. To je funkcija *ImplicVkllop*. Tako dobimo dosti lepšo dinamiko, vprašljivo pa je katera je bolj fizikalna. Dinamiko v tem primeru vidimo na slikah 4 spodaj.

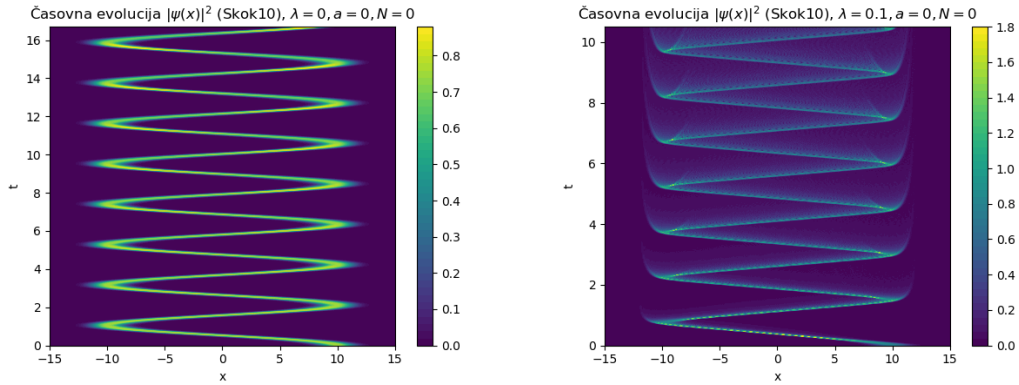


Slika 4: Časovna evolucija lastnih stanj osnovnega potenciala ob počasnem vklopu motnje. Potencial se linearno vkaplja prvo tretjino časovnega intervala

Naredil sem tudi videoposnetke te dinamike, ki so shranjeni v mapi. To so posnetki z naslovom *prizig*....

Na koncu sem opazil, da nam je prosen predlagal, da pogledamo še dinamiko za $\psi(x)' = \psi(x - 10)$. To se je dejansko izkazalo za zanimivo. Predvsem glede učinkovitosti metod, saj pri predvsem (!!) prevelikem **krajevem** koraku, metoda ne deluje dobro. Tako da je potrebno vzeti majhen krajevni korak in ne pretiravati s časovnim, da metoda še ostane učinkovita. Metoda je recimo pri krajevem koraku 0.1, območju velikem 30 in $\lambda = 0$ že odpovedala. Moral

sem korak zmanjšati na $h = 0.05$, da sem dobil lep rezultat, in potem se še dodatno manjšal korak in probal izboljšati tudi razmerje med časovnim in krajevnim, da sem dobil lep rezultat. Kljub temu rezultati še vedno niso popolni, kar se ne vidi na sliki, se pa vidi na **posnetek7**. Na tem posnetku je naslov napisan narobe. Gre za $N = 0, a = 10$, ostalo pa je pravilno. Če nato vklopimo močnejši potencial ponavadi metoda odpove, ker vrednosti hitro divergirajo. Potrebno je počasi in natančno spreminjati natančnost in stvar lahko hitro postane časovno prezahtevna. Zato sem izrisal še rezultat pri šibki motnji $\lambda = 0.1$ na **posnetek8**. Oba rezultata sta prikazana tudi spodaj na sliki 5.



Slika 5: Časovna evolucija pri velikem izmaku.