

10. naloga: Diferenčne metode za parcialne diferencialne enačbe

Gregor Žunič

Enorazsežna nestacionarna Schrödingerjeva enačba

$$\left(i\hbar\frac{\partial}{\partial t} - H\right)\psi(x, t) = 0$$

je osnovno orodje za nerelativistični opis časovnega razvoja kvantnih stanj v različnih potencialih. Tu obravnavamo samo od časa neodvisne hamiltonske operatorje

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x).$$

Z menjavo spremenljivk $H/\hbar \mapsto H$, $x\sqrt{m/\hbar} \mapsto x$ in $V(x\sqrt{m/\hbar})/\hbar \mapsto V(x)$, efektivno postavimo $\hbar = m = 1$,

$$H = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + V(x). \quad (1)$$

Razvoj stanja $\psi(x, t)$ v stanje $\psi(x, t + \Delta t)$ opišemo s približkom

$$\psi(x, t + \Delta t) = e^{-iH\Delta t}\psi(x, t) \approx \frac{1 - \frac{i}{2}H\Delta t}{1 + \frac{i}{2}H\Delta t}\psi(x, t), \quad (2)$$

ki je unitaren in je reda $\mathcal{O}(\Delta t^3)$. Območje $a \leq x \leq b$ diskretiziramo na krajevno mrežo $x_j = a + j\Delta x$ pri $0 \leq j < N$, $\Delta x = (b - a)/(N - 1)$, časovni razvoj pa spremljamo ob časih $t_n = n\Delta t$. Vrednosti valovne funkcije in potenciala v mrežnih točkah ob času t_n označimo $\psi(x_j, t_n) = \psi_j^n$ oziroma $V(x_j) = V_j$. Krajevni odvod izrazimo z diferenco

$$\Psi''(x) \approx \frac{\psi(x + \Delta x, t) - 2\psi(x, t) + \psi(x - \Delta x, t)}{\Delta x^2} = \frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{\Delta x^2}.$$

Ko te približke vstavimo v enačbo (2) in razpišemo Hamiltonov operator po enačbi (1), dobimo sistem enačb

$$\psi_j^{n+1} - i\frac{\Delta t}{4\Delta x^2} [\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}] + i\frac{\Delta t}{2} V_j \psi_j^{n+1} = \psi_j^n + i\frac{\Delta t}{4\Delta x^2} [\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n] - i\frac{\Delta t}{2} V_j \psi_j^n,$$

v notranjih točkah mreže, medtem ko na robu ($j \leq 0$ in $j \geq N$) postavimo $\psi_j^n = 0$. Vrednosti valovne funkcije v točkah x_j uredimo v vektor

$$\Psi^n = (\psi_1^n, \dots, \psi_{N-1}^n)^T$$

in sistem prepišemo v matrično obliko

$$\mathbf{A}\Psi^{n+1} = \mathbf{A}^*\Psi^n, \quad \mathbf{A} = \begin{pmatrix} d_1 & a & & & & \\ a & d_2 & a & & & \\ & a & d_3 & a & & \\ & & \ddots & \ddots & \ddots & \\ & & & a & d_{N-2} & a \\ & & & & a & d_{N-1} \end{pmatrix},$$

kjer je

$$b = i \frac{\Delta t}{2\Delta x^2}, \quad a = -\frac{b}{2}, \quad d_j = 1 + b + i \frac{\Delta t}{2} V_j.$$

Dobili smo torej matrični sistem, ki ga moramo rešiti v vsakem časovnem koraku, da iz stanja Ψ^n dobimo stanje Ψ^{n+1} . Matrika A in vektor Ψ imata kompleksne elemente, zato račun najlažje opraviš v kompleksni aritmetiki¹.

Naloga: Spremljaj časovni razvoj začetnega stanja

$$\Psi(x, 0) = \sqrt{\frac{\alpha}{\sqrt{\pi}}} e^{-\alpha^2(x-\lambda)^2/2}$$

v harmonskem potencialu $V(x) = \frac{1}{2}kx^2$, kjer je v naravnih enotah $\alpha = k^{1/4}$, $\omega = \sqrt{k}$. Analitična rešitev je koherentno stanje

$$\psi(x, t) = \sqrt{\frac{\alpha}{\sqrt{\pi}}} \exp \left[-\frac{1}{2} (\xi - \xi_\lambda \cos \omega t)^2 - i \left(\frac{\omega t}{2} + \xi \xi_\lambda \sin \omega t - \frac{1}{4} \xi_\lambda^2 \sin 2\omega t \right) \right],$$

kjer je $\xi = \alpha x$, $\xi_\lambda = \alpha \lambda$. Postavi parametre na $\omega = 0.2$, $\lambda = 10$. Krajevno mrežo vpni v interval $[a, b] = [-40, 40]$ z $N = 300$ aktivnimi točkami. Nihajni čas je $T = 2\pi/\omega$ – primerno prilagodi časovni korak Δt in stanje opazuj deset period.

Opazuj še razvoj gaussovskega valovnega paketa

$$\psi(x, 0) = (2\pi\sigma_0^2)^{-1/4} e^{ik_0(x-\lambda)} e^{-(x-\lambda)^2/(2\sigma_0^2)}$$

v prostoru brez potenciala. Postavi $\sigma_0 = 1/20$, $k_0 = 50\pi$, $\lambda = 0.25$ in območje $[a, b] = [-0.5, 1.5]$ ter $\Delta t = 2\Delta x^2$. Časovni razvoj spremljaj, dokler težišče paketa ne pride do $x \approx 0.75$. Analitična rešitev je

$$\psi(x, t) = \frac{(2\pi\sigma_0^2)^{-1/4}}{\sqrt{1 + it/(2\sigma_0^2)}} \exp \left[\frac{-(x - \lambda)^2/(2\sigma_0^2) + ik_0(x - \lambda) - ik_0^2 t/2}{1 + it/(2\sigma_0^2)} \right]$$

Dodatna naloga: Z uporabljenim približkom za drugi odvod reda $\mathcal{O}(\Delta x^2)$ dobimo tridiagonalno matriko. Z diferencami višjih redov dobimo večdiagonalno (pasovno) matriko, a dosežemo tudi večjo krajevno natančnost. Diference višjih redov lahko hitro izračunaš na primer v Mathematici s funkcijo

```
FD[m_,n_,s_] := CoefficientList[Normal[Series[x^s Log[x]^m, {x, 1, n}]/h^m], x];
```

kjer je m red difference (odvoda), n število intervalov širine $h = \Delta x$, ki jih diferenca upošteva, in s število intervalov med točko, kjer diferenco računamo, in skrajno levo točko diferenčne sheme. Zgornjo tritočkovno sheme za drugo diferenco dobimo kot `FD[2, 2, 1]`, saj se razpenja čez $n=2$ intervala, sredinska točka pa je v točki z indeksom $s=1$.

Tudi korakanje v času je mogoče izboljšati z uporabo Padéjeve aproksimacije za eksponentno funkcijo, glej [1] in/ali predavanja.

Literatura

- [1] W. van Dijk, F. M. Toyama, Phys. Rev. E **75**, 036707 (2007).

¹`#include <complex.h>` v c, `#include <complex>` v c++, `from cmath import *` za kompleksne funkcije v Pythonu (sama kompleksna aritmetika pa je vgrajena).

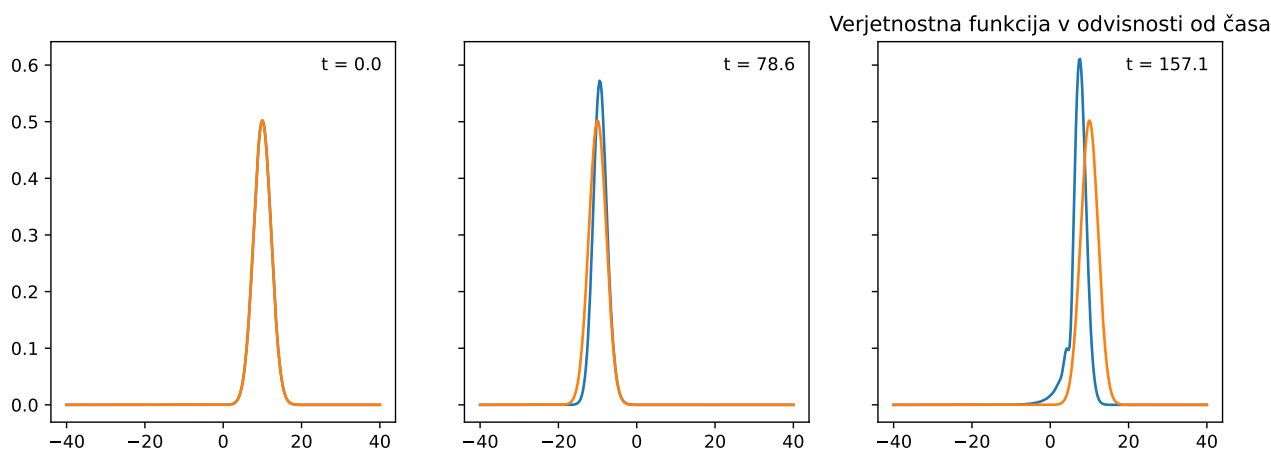
1 Reševanje

Naloga je dokaj direktna. Napisati moramo tridiagonalno matriko, poiskati inverz in jo iterativno množiti z valovno funkcijo, ki pride. Pri implementaciji je smiselno namesto uporabe funkcije `solve_banded` najprej poiskati produkt inverza in konjugirani matrike, ker je ta matrika neodvisna od časa. To pomeni, da se s tem znebimo nepotrebnega iskanja inverza ter je koda precej hitrejša.

Zanimiv opazek je, da python potrebuje relativno veliko časa, da izračuna analitično rešitev (17 sekund za milijon iteracij, kar je relativno veliko). Lahko bi paraleliziral operacijo, vendar ni dovolj pomembno.

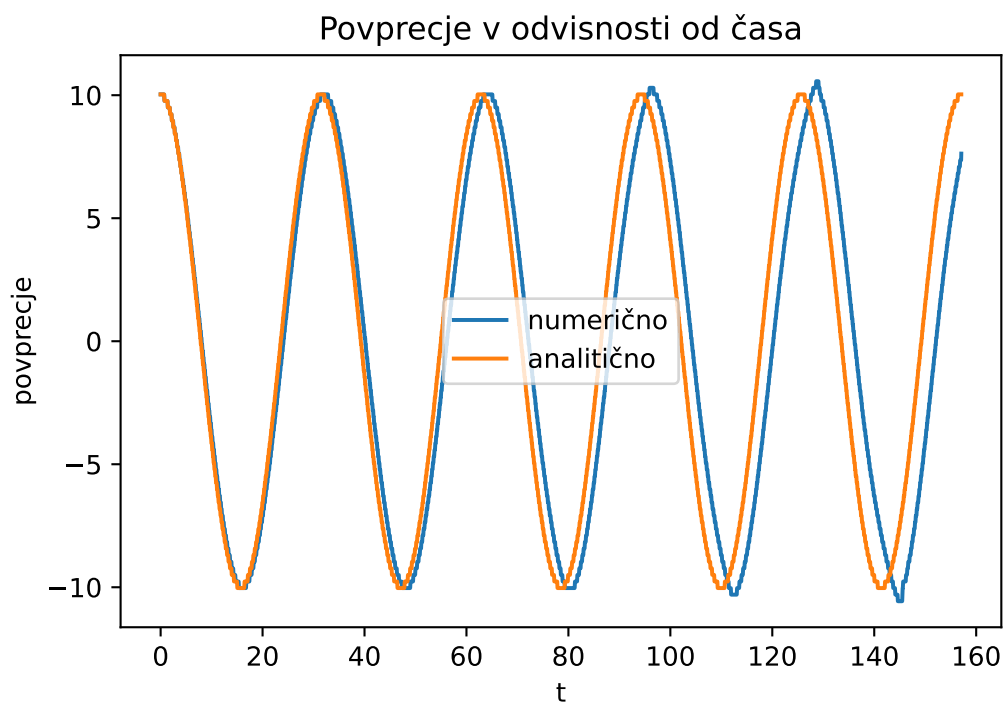
1.1 GVP v harmonskem potencialu

Parametri so enaki, kot jih predpisuje navodilo. Smiselni so, ker se lepo vidi premikanje in razvoj Gausovke v odvisnosti od časa. Kar opazimo je, da se GVP premika levo in desno, njegova višina in širina pa ostaja enaka (vsaj v teoriji, oziroma obnašanje analitične funkcije). Tako zares pridemo do koherentnih stanj, kar nam tudi numerika potrdi.



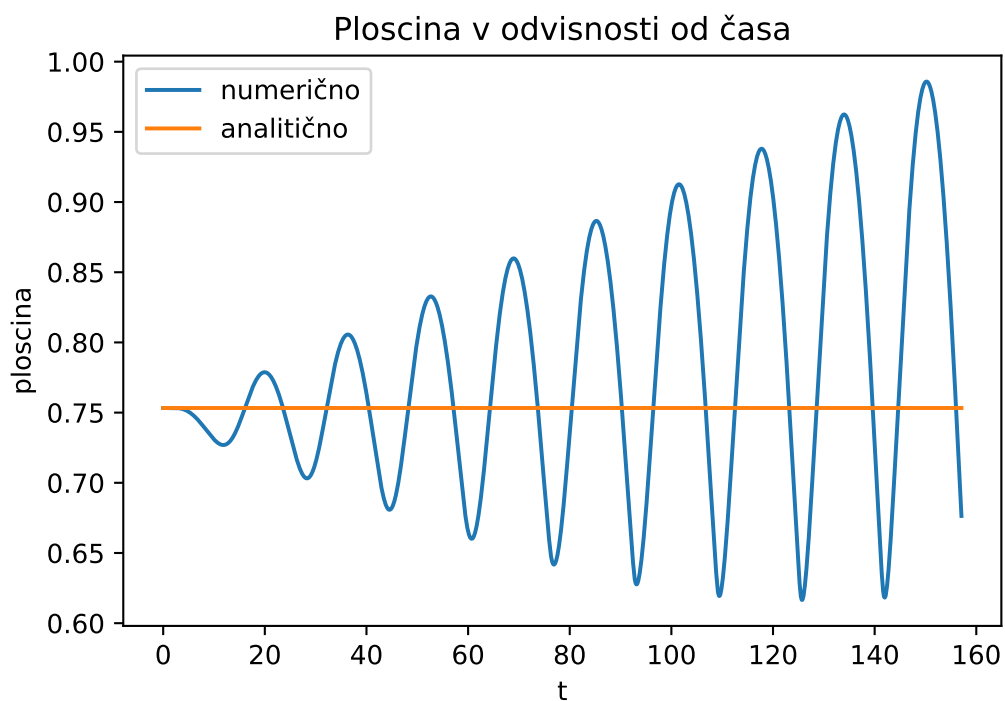
Slika 1: Verjetnostna funkcija pri nekaj časih. Bolj nazorna je animacija (file z imenom `gausovka_razvoj.gif`)

Zanimivo vprašanje, ki se porodi je kako se premika povprečje s časom. Mogla bi se premikati enako za analitično in numerično. Vendar numerična malce zaostaja. Prvi nihaj je dokaj konstanten, kar nakazuje na to, da je bi to napako lahko odpravil z manjšanjem Δt in Δx .



Slika 2: Spreminjanje povprečne lege. Numerično malce zaostaja, in tudi amplituda se večja.

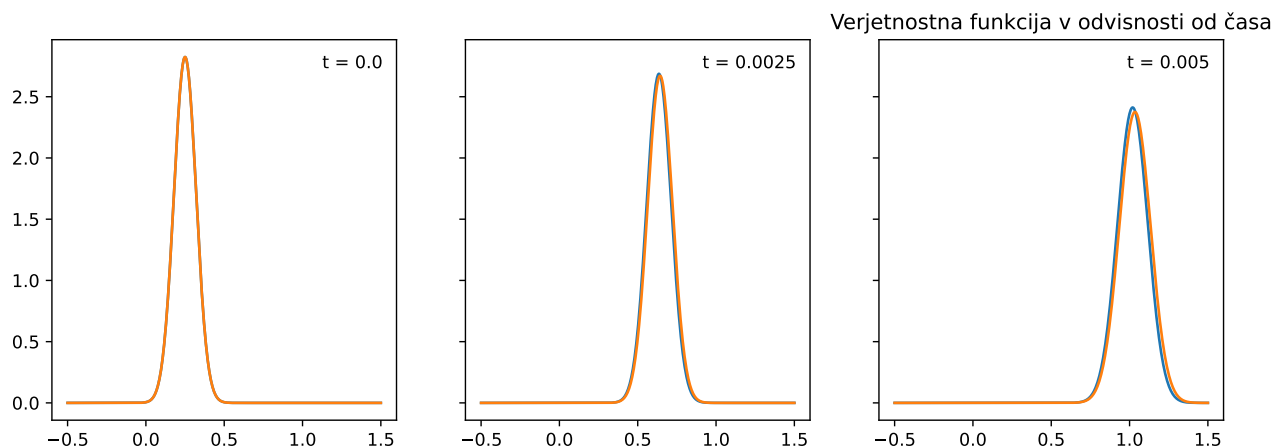
Kaj pa se dogaja z ploščino? Tudi ta bi morala biti v teoriji konstantna, vendar se precej zlomi. Ker je metoda simplektična ploščina ne divergira, vendar samo z večjo amplitudo kroži okrog središča.



Slika 3: Spreminjanje povprečne lege. Numerično malce zaostaja, in tudi amplituda se večja.

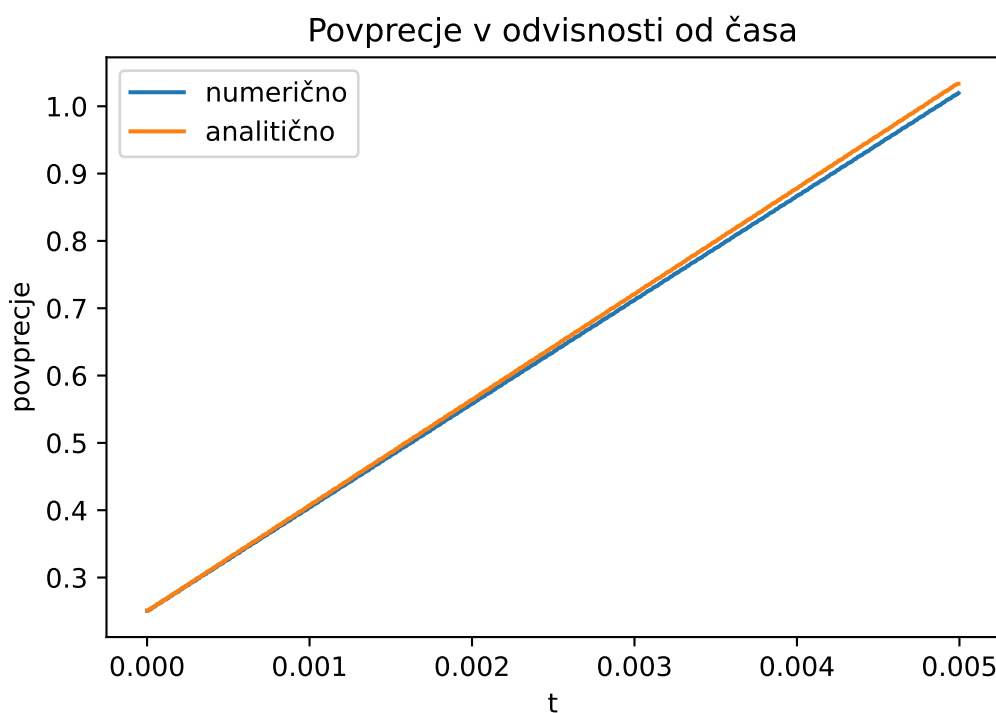
1.2 Prost GVP

Kar bi se moralo zgoditi je, da se vrh linearno premika v desno (kot klasična limita hitrosti delca), Gausovka pa se širi (oziroma verjetnost delca, da ga zaznamo pri nekem območju okrog vrha se manjša). Pri časih večjih kot 0 iz teorije vemo, da se GVP zlomi (oziroma ni več GVP).



Slika 4: Verjetnostna funkcija pri nekaj časih. File `potencial_0.gif` pokaže animacijo

Kako pa se premika povprečna vrednost? Mogla bi se premikati enako za analitično in numerično. Vendar numerična malce zaostaja. Pri natančnosti, sem malce pretiraval v negativno smer (premalo natančno), da se zares vidi obnašanje.



Slika 5: Spreminjanje povprečne lege. Numerična se širi počasneje, kot analitična.

Povprečje malce zaostaja za analitično rešitvijo, vendar jo lahko poljubno približamo analitični rešitvi. Kar moramo paziti je, da se numerična funkcija na robu v bistvu odbije in popolnoma zlomi, kar se

zgodí, ker nimamo pravega robnega pogoja (oziroma računamo zgolj do roba, tako da se tam obnaša, kot da je stena). To nam lepo prikaže animacija `potencial_0_zlom.gif`.

1.3 Komentar

Vidimo, da metoda deluje, vendar ni najboljša. Če nas zanima zgolj približno obnašanje, je metoda OK do natančnosti, vendar je numerično kar precej zahtevna, če se gremo zares natančno sliko. To lahko seveda izboljšamo z višjimi redi diferenc, za katere pa mi je zmanjkalo časa...