Semestrální práce TQS

Jakub Dokulil, 215768

2021

Zadání

Jáma konečné hloubky - první tři stavy a jejich hustoty pravděpodobnosti ($E < V_0$).

1. Zdrojový kód programu .py – s komentáři a odpovídajícím porozumění problematice (budu se ptát osobně nebo prostřednictvím microsoft teams).
2. 2 stránkový dokument ve formátu .pdf (včetně zdroje latex nebo word)
   (a) První 1/3 první strany – nástin zadání problému.
   (b) Druhá 1/3 první strany – analytický Hamiltonián a náznak tvaru maticového Hamiltoniánu.
   (c) Třetí 1/3 první strany – co mne na řešení zaujalo (úlohy a přístupu).
   (d) Druhá strana – dva nejvíce ilustrativní a popsané grafy, které byly získány řešením úlohy (které nejlépe znázorňují řešení).
3. Video – v případě časových vývojů, které nejlépe vystihuje řešení (formát .avi).

Pravoúhlá potenciálová jáma

Při řešení byl použit postup podle článku Garcia, R., Zozulya, A., Stickney, J. (2007). *MATLAB codes for teaching quantum physics: Part 1.* URL: `http://arxiv.org/pdf/0704.1622.pdf`, kód byl implementován v programovacím jazyce Julia s použitím balíků LinearAlgebra a Trapz. Taktéž byl výpočet implementován vy jazyce Python s využitím knihovny NumPy.

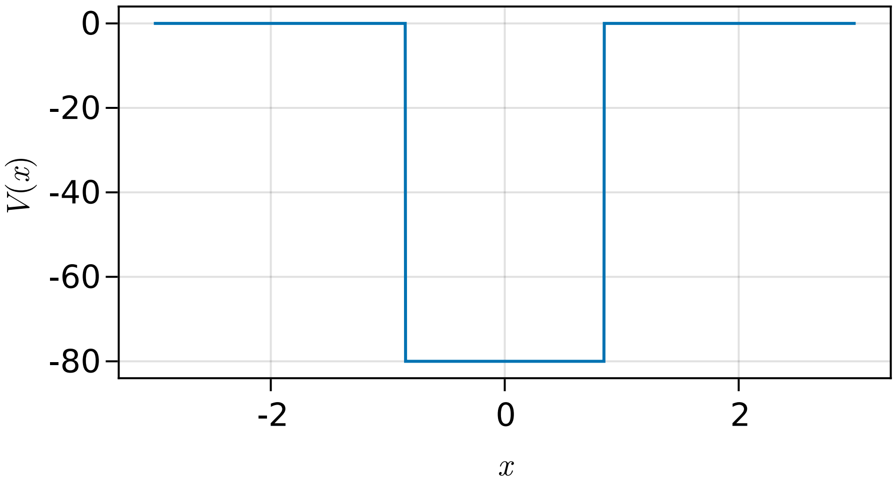
Jáma s pravoúhlým potenciálem.

Obecně lze zapsat stacionární Schrödingerovu rovnici ve tvaru

$$\hat{H}\left|\Psi\right\rangle = E\left|\Psi\right\rangle.$$

Rozepsáním operátoru Hamiltoniálnu $\hat{H}$ dostáváme rovnici.

$$-i\frac{\hbar^2}{2m}\frac{\mathrm{d}^2}{\mathrm{d}x^2}\Psi(x) + V(x) = E\Psi(x)$$



Obrázek 1: Potenciálová jáma.

Pravoúhlý potenciál lze vyjádřit například pomocí Heavisideovy funkce $h(x)$

$$h(x) = \begin{cases} 0 & x < 0 \\ 0{,}5 & x = 0 \\ 1 & x > 1 \end{cases}$$

Potenciál pravoúhlé jámy šířky L a konečné hloubky d lze tedy vyjádřit jako

$$V(x;\ d, L) = -d\left[h\left(x + \frac{L}{2}\right) - h\left(x + \frac{L}{2}\right)\right]$$

Graf potenciálu lze vidět na obrázku 1.

Implementace

Výpečet je prováděn na prostoru se součadnicí x rodělenou na N dílků. Proto je Hamiltonián reprezentován čtvercovou maticí $N \times N$ působící na N-rozměrný vektor $|\Psi\rangle$.

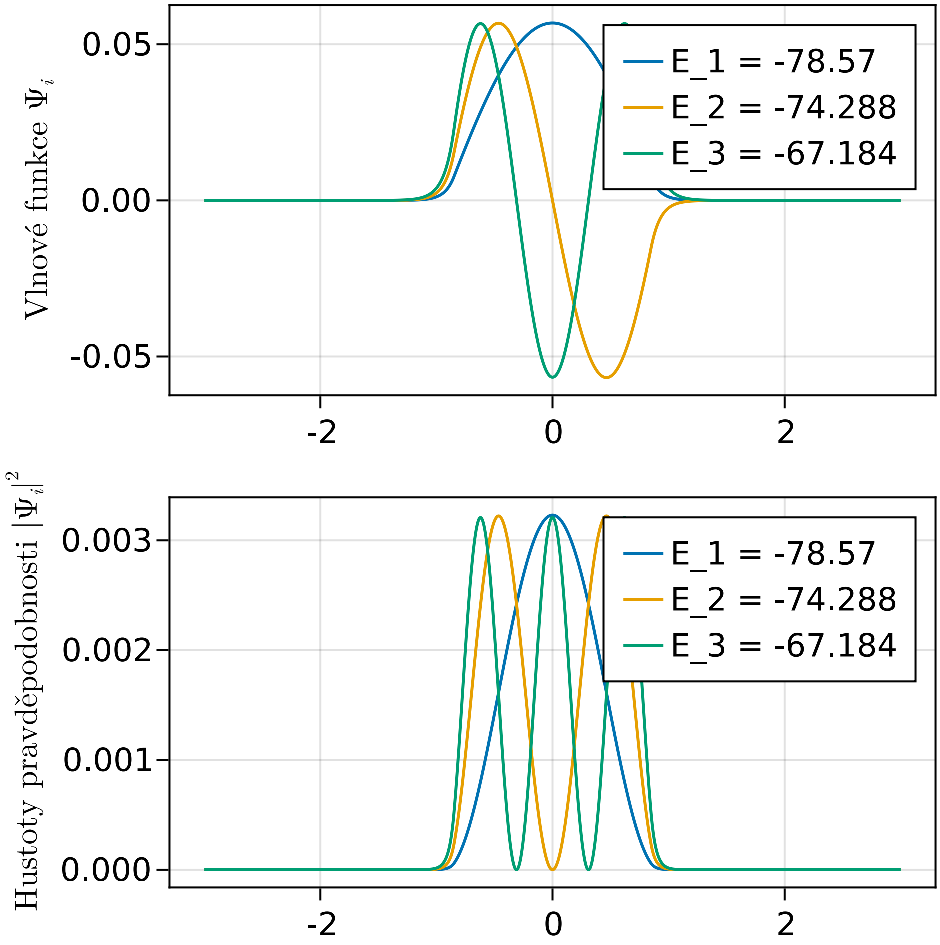
Laplaceův operátor je reprezentován maticí

$$\Delta = \frac{1}{(\mathrm{d}x)^2}\begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & 0 \\ 0 & 1 & -2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & 0 & \dots & 1 & -2 \end{pmatrix}$$

a celý Hamiltonián následně vypadá

$$\hat{H} = \frac{\hbar^2}{2m}\frac{1}{(\mathrm{d}x)^2}\begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & 0 \\ 0 & 1 & -2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & 0 & \dots & 1 & -2 \end{pmatrix} + \mathrm{diag}(V(x))$$

Řešení rovnice se redukuje na hledání vlastních hodnot a hlastních vektorů matice Hamiltoniánu.



Obrázek 2: Výsledek výpočtu pro jámu šířky $d = 1,7$ a hloubky $L = 80$. Vykreslené první tři stavy a příslušné hustoty pravděpodobnosti. Vykresleno pomocí knihovny Makie.

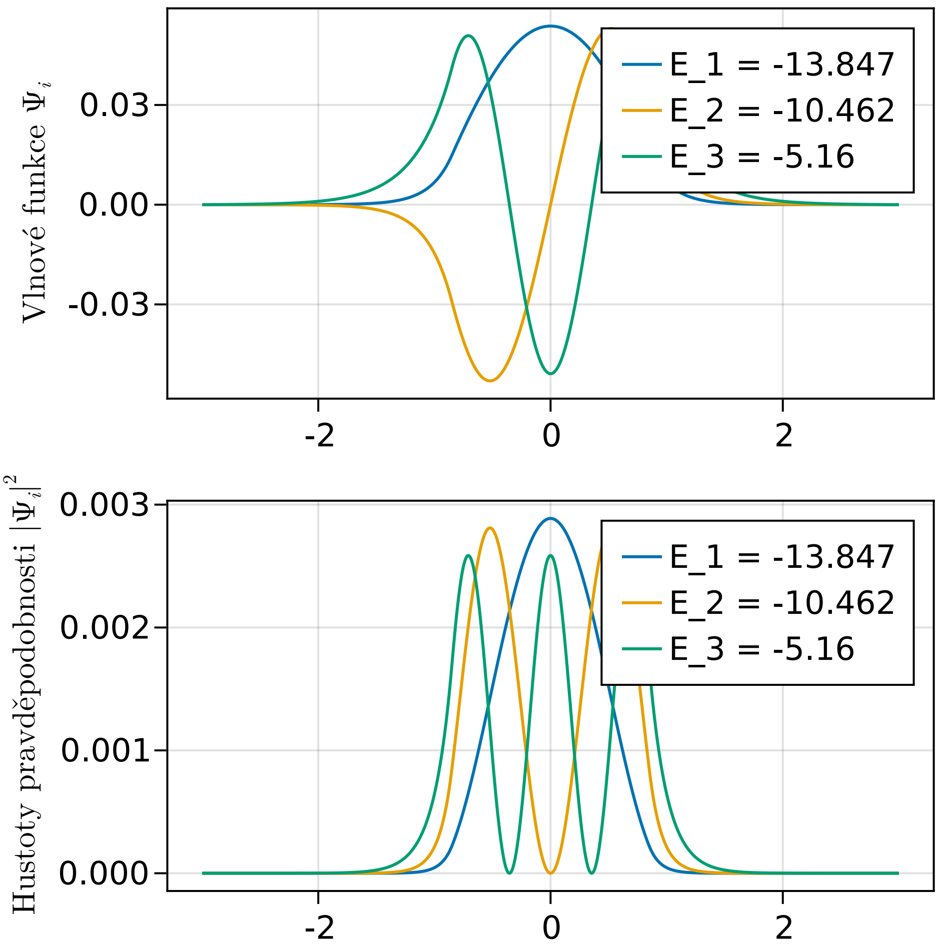
Co mne na řešení zaujalo?

Na řešení mne zaujalo, jak jednoduše proveditelné je. V podstatě jediné o co jde, je sestrojit matici a najít její vlastní čísla a vlastní vektory. Toto může velmi usnadnit řešení komplikovanějších úloh.

Výpočet byl implementován zároveň v Pythonu s užitím knihoven NumPy a v Julii. Pro porovnání byl proveden výpočet stejné jámy s $N = 6000$ v Pythonu a v Julii s využitím 32 a 64 bitových proměnných. Volba N je volena tak, aby výpočet trval lehce přes minutu. Při řešení na notebooku vyšla nejrychleji Julia s 64 bitovou reprezentací. Ukázalo se, že kompilovaná Julie je rychlejší než Pythonovské knihovny a to zhruba dvakrát. To, že 64 bitový reprezentace je rychlejší, je pravděpodobně způsobeno procesorem, který je taktéž 64 bitový.

Výsledky výpočtu

Výsledky Lze visdět na obrázku 2, kde jsou vykresleny první tři stavy. Při provední výpočtu pro mělčí jámu (d = 15) lze vidět prosakování mimo jámu zvláště u vyšších energiových stavů. Výsledek lze vidět na obrázku 3.



Obrázek 3: Výsledek výpočtu pro jámu šířky $d = 1,7$ a hloubky $L = 20$. Vykreslené první tři stavy a příslušné hustoty pravděpodobnosti. Vykresleno pomocí knihovny Makie.