

Feynman-féle pályaintegrál számítása Metropolis algoritmussal

Nagy Dávid Gergely - II. számítógépes fizika beadandó

2013. január 14.

A pálya-integrál formuláció a kvantummechanika egy megfogalmazása, amely a klasszikus mechanika hatás elvét általánosítja. Feynman interpretációja szerint a klasszikus hatás a kvantum időfejlődés fázisa két fix végpont között, és a teljes kvantummechanika rekonstruálható a következő posztulátumok alapján:

1. Egy esemény valószínűsége a valószínűségi amplitúdó komplex szám négyzetes hosszából kapható meg.
2. A valószínűségi amplitúdót úgy kaphatjuk meg, hogy összeadjuk a konfigurációs tér összes oda vezető útvonalának járulékait.
3. Egy adott útvonal járuléka $e^{i\frac{S}{\hbar}}$ -val arányos, ahol S az adott trajektóriához tartozó hatás, a Lagrange függvény idő szerinti integrálja.

A klasszikus mechanikával való kapcsolat a Hamilton elvhez kötődik, amely szerint a részecske olyan pályán mozog, hogy a hatásnak extrémuma legyen

$$\delta S[x(t)] = 0$$

ahol

$$S(x(t)) = \int_{t_a}^{t_b} dt L(x(t), \dot{x}(t)).$$

E szerint a szabad részecskére vonatkozó hatás

$$S(b, a) = \frac{m(x_b - x_a)^2}{2(t_b - t_a)}.$$

Tegyük fel hogy felírható a b ponthoz tartozó hullámfüggvény a következő módon¹

$$\psi(x_b, t_b) = \int dx_a G(x_b, t_b; x_a, t_a) \psi(x_a, t_a) \quad (1)$$

¹Ez megfeleltethető a Huygens elvnek, ahol a hullámfront minden pontja egy gömbhullámot (a Green függvény) bocsát ki, és ezeknek az interferenciája lesz az új hullámfront.

ahol G a Green függvény vagy propagátor

$$G(x_b, t_b; x_a, t_a) \equiv G(b, a) = \sqrt{\frac{m}{2\pi i(t_b - t_a)}} e^{i\frac{m(x_b - x_a)^2}{2(t_b - t_a)}}. \quad (2)$$

Észrevehetjük, hogy $G(b,a)$ és $S(b,a)$ között összefüggés van

$$G(b, a) = \sqrt{\frac{m}{2\pi i(t_b - t_a)}} e^{\frac{i}{\hbar} S(b,a)}.$$

Feynman ez alapján alkotta meg a kvantummechanika pálya integrálját

$$G(b, a) = \sum_{\forall \text{ pálya}} e^{i\frac{S[b,a]}{\hbar}}$$

ami a $\hbar \rightarrow 0$ határesetben a Hamilton elvet adja vissza. Az exponenciális tag fázisa a klasszikushoz hasonló trajektóriákon viszonylag lassan oszcillál, így ezek a tagok konstruktívan interferálnak, míg a távoli utak kioltják egymást.

Alapállapot hullámfüggvény

Tegyük fel hogy a Hamilton operátor spektruma

$$H|\psi_n\rangle = E_n|\psi_n\rangle.$$

Ekkor G kifejezhető a sajátfüggvényekkel *könyv:(15.48)*

$$G(x, t; x_0, t_0 = 0) = \sum \psi_n^*(x_0)\psi_n(x)e^{-iE_n t}$$

Ha G -t analitikusan kiterjesztjük negatív imaginárius időkre, akkor látható, hogy a gerjesztett állapotok az alapállapothoz képest gyorsan lecsengenek, így az alapállapot hullámfüggvény kifejezhető a következő módon

$$|\psi_0(x)|^2 = \lim_{\tau \rightarrow \infty} e^{E_0 \tau} G(x, -i\tau; x, 0)$$

Ennek megfelelően a Lagrange függvényt $-i\tau$ szerint véve azt kapjuk *könyv:(15.61)*, hogy

$$\mathcal{L}\left(x, \frac{dx}{-id\tau}\right) = -H\left(x, \frac{dx}{d\tau}\right)$$

Így a hatást $\int \mathcal{L} dt$ helyett $\int H d\tau$ -ként adjuk meg, és így a Green függvény is a Hamilton operátor szerint fejezhető ki

$$G(x, -i\tau; x_0, 0) = \int dx_1 \dots dx_{N-1} e^{\int_0^\tau H(\tau') d\tau'}$$

Tehát itt is a Boltzmann-faktorhoz hasonló tag jelenik meg, ami nem meglepő, mivel a Schrödinger egyenletben az időt imagináriusnak véve a diffúziós egyenletet kapjuk.

A fenti integrált egy téridő rács linkjein való szummaként értelmezve megkapjuk az algoritmust arra, hogy hogyan számolhatjuk ki a Green függvényt (és ezáltal az alapállapot hullámfüggvényt is) egy téridő rácson. Az integrálok elvégzéséhez persze végtelen lépés szükséges, mivel az összes pályára összegezni kell. Ezért sztochasztikusan, a nagy járuléku tagokból mintavételezve próbálkozunk.

Metropolis algoritmus

A Metropolis algoritmus egy Markov chain Monte Carlo módszer. Az MCMC algoritmusok segítségével bonyolult (pl sok dimenziós) valószínűségi eloszlásokból lehet sztochasztikusan mintavételezni illetve integrálokat közelíteni. A módszer lényege, hogy egy olyan Markov láncot konstruálunk, amelynek az egyensúlyi eloszlása a mintavételezni kívánt $P(x)$ sűrűségfüggvény és a lánc sok lépés utáni állapotait tekintjük a mintáknak.

1. Kezdjük valamilyen x_0 kezdőállapotból
2. Egy tetszőleges, de szimmetrikus ² eloszlás (*proposal distribution*) segítségével válasszunk egy javasolt új állapotot.
3. Számítsuk ki az elfogadási arányt (*acceptance ratio*) $a = \frac{P(x')}{P(x_t)}$
 - (a) Ha $a \geq 1$ akkor fogadjuk el a javasolt új állapotot, azaz $x_{t+1} = x'$
 - (b) Ha $a < 1$ akkor $1 - a$ valószínűséggel vessük el az állapotot, ellenkező esetben fogadjuk el.
4. Folytassuk a 2. lépéstől.

Az így kapott x_t állapotok az eloszlás szerint magas valószínűségű állapotok között bolyonganak, de a 3.a miatt néha alacsonyabb valószínűségű irányokba is lépnek. A 3. pontban szereplő arány azért előnyös, mert emiatt elég egy $P(x)$ -el arányos mennyiséggel számolnunk, mert a konstans szorzó - pl egy nehezen kiszámolható állapotösszeg - kiesik. A Metropolis algoritmusnak speciális esete a Gibbs mintavételezés, amikor Boltzmann eloszlást használunk.

A módszer hátrányai, hogy

- a minták korreláltak. Ennek csökkentésére lehet azt a módszert alkalmazni, hogy csak minden n -ik mintát tárolják el (thinning) illetve az ugrások hosszát növelni. Utóbbi esetben viszont ha túl nagy az ugrás hossza, akkor minden javaslat vissza lesz utasítva.
- A kezdeti minták erősen függhetnek a kezdeti állapottól, főleg ha az egy alacsony valószínűségű régióban van. Ezen effektus elkerülése végett el szokták dobni az adatok első részét (*burn-in period*).

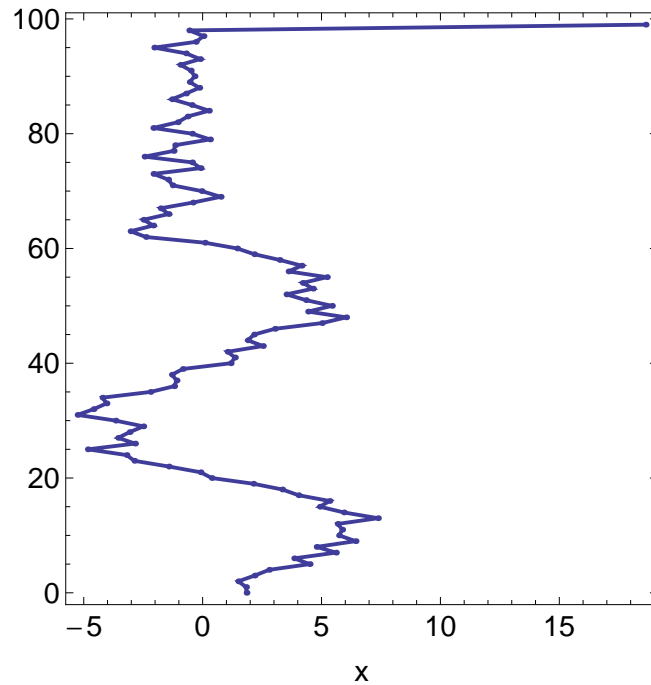
²Az általánosabb Metropolis-Hastings algoritmusnál nincs szükség rá hogy szimmetrikus legyen

Algoritmus a pályaintegrál téridő rácson való implementációjára

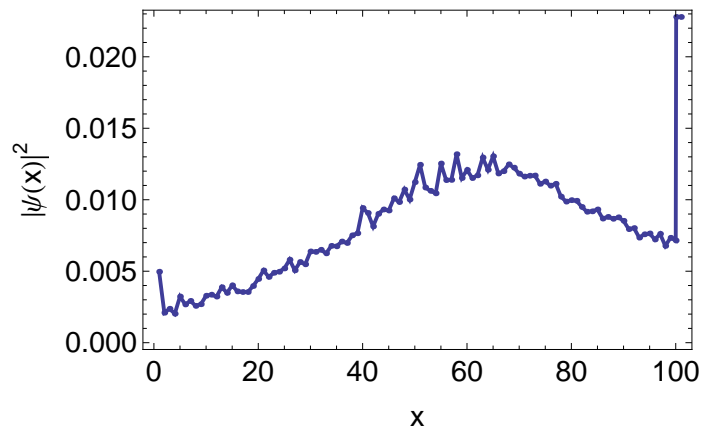
1. Vegyünk fel egy téridő rácsot, ahol az idő tengely N lépés elépésközzel, a hely pedig M pont δ -nként. Figyeljünk arra, hogy az utóbbi intervallum többszöröse legyen a vizsgálandó potenciál karakterisztikus méretének. A hullámfüggvény számításakor a rácspontok közé eső x és t értékeket rendeljük a legközelebbi rácspontokhoz.
2. Minden t_j időpillanathoz rendeljünk egy x_j pontot, az $x_0 = x_N$ határfeltételnek megfelelően. Ezután kössük össze a pontokat hogy megkapjuk a részecske diszkretizált világvonalát.
3. Számoljuk ki a pályához tartozó energiát: $j = 0$ -tól minden összeköttetésre adjuk össze a kinetikus és a potenciális energiát a következő képlet szerint
4. Metropolis algoritmus
 - (a) Egy véletlenszerű t_j -hez tartozó x_j -t változtassunk meg x'_j -re. Ez két összekötő vonalat változtat meg.
 - (b) Súlyozzuk a boltzmann faktoral
 - (c) Minden ciklusban növeljük az aktuális rácsponthoz rendelt, a $|\psi(x_j)|^2$ értékét reprezentáló számot 1-el.
5. Ismételjük meg a szimulációt több különböző seed-del, mivel sok közepesen hosszú szimuláció eredményeinek átlaga jobb mint egy nagyon hosszú szimulációból kapott érték.

A program

A könyvben található programban több eltérést is találtam ahhoz képest amit a fent vázolt algoritmus leír. Ezt alátámasztja az is, hogy a programot sokáig futtatva teljesen rossz eredményeket ad mind a pályákban mind a hullámfüggvényben.



1. ábra. A részecske szimulált $t(x)$ világvonala a könyvhöz csatolt QMC.py program 69k lépésig történő futtatása után.



2. ábra. A részecske hullámfüggvényének abszolútérték négyzete, a könyvhöz csatolt QMC.py program 69k lépésig történő futtatása után.

Ennek okai:

- A végpontok nincsenek rögzítve, a véletlen pont választásánál a véletlen számnak $(0, N)$ helyett $(1, N - 1)$ között kellene lennie.
- Az energia számolásában a potenciál a végpontra nem vonatkozik, így ez egy idő után 'elmászik' a végtelenbe, innen származik a valószínűségrés utolsó pontjában a magas érték.

- Az energia pályán való integrálásánál a következő képlet van megadva a könyvben:

$$\sum_{j=1}^N \frac{m}{2} \left(\frac{x(j) - x(j-1)}{\epsilon} \right)^2 + \sum_{j=1}^N \frac{1}{2} \left(\frac{x(j) + x(j-1)}{2} \right)^2$$

ami az $m = 1$ és $j \rightarrow j + 1$ helyettesítéssel a

$$\sum_{j=0}^{N-1} \frac{1}{2\epsilon^2} (x(j+1) - x(j))^2 + \sum_{j=0}^{N-1} \frac{1}{8} (x(j+1) + x(j))^2$$

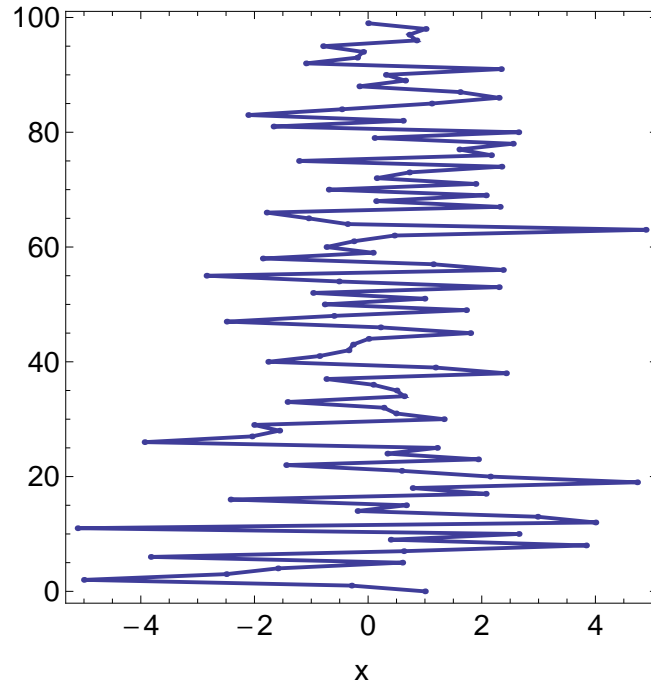
alakra hozható. A programban viszont a

$$\sum_{j=0}^{N-2} (x(j+1) - x(j))^2 + \sum_{j=1}^{N-2} (x(j+1) + x(j))^2$$

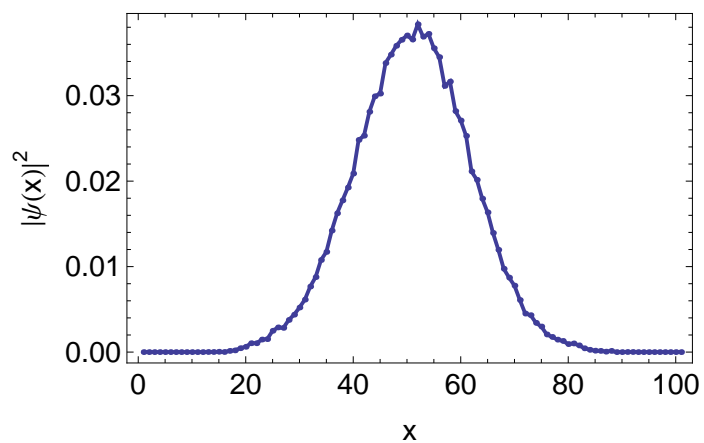
képlet szerepel, amiből hiányoznak a konstansok. A metropolis algoritmusban szereplő boltzmann-faktorból is hiányzik ϵ . Ezek miatt az időlépések hossza nem állítható.

- A $(0 \ 0 \ \dots \ 0)$ kezdeti feltétel miatt a rövid futásai idővel kapott eredményekben a hullámfüggvény $x = 0$ -hoz tartozó értéke nagyon magas. (*burn-in period*)

A javított kóddal kapott eredmények



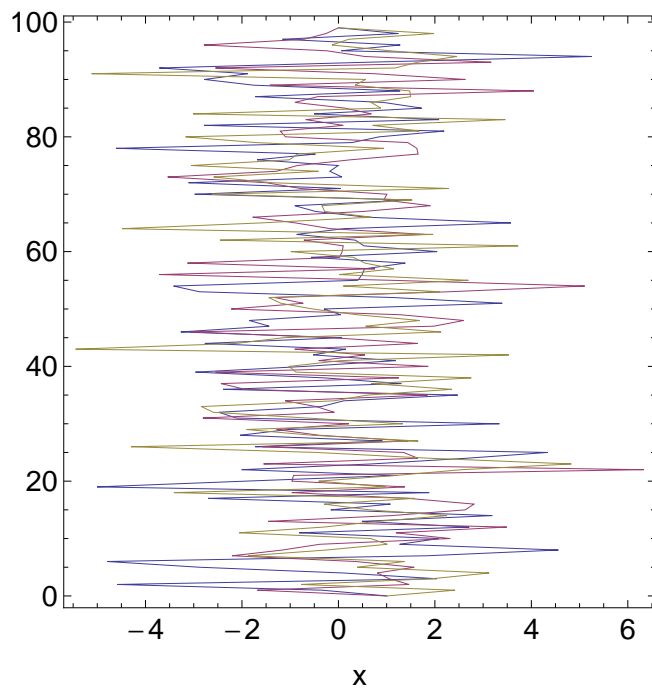
3. ábra. A részecske szimulált $t(x)$ világvonala a javított QMCjav.py program 79k lépésig történő futtatása után.



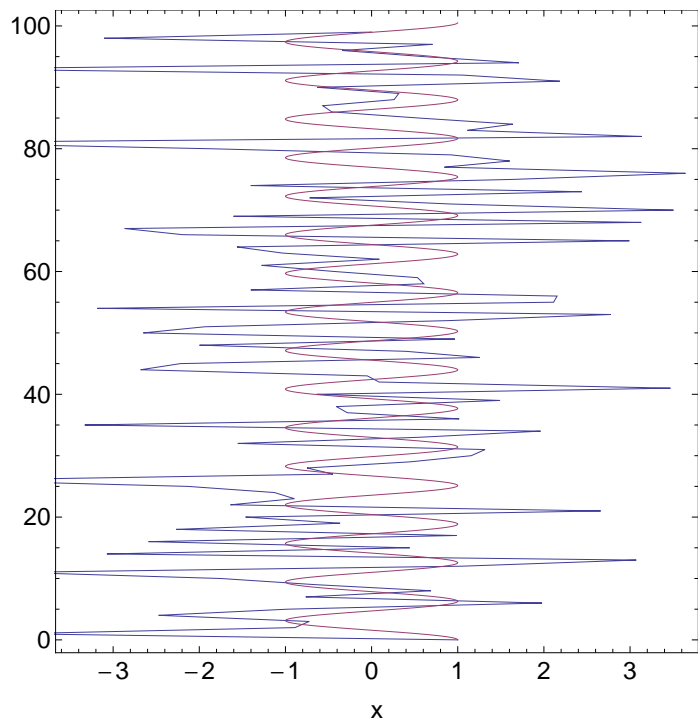
4. ábra. A részecske hullámfüggvényének abszolútérték négyzete, a javított QMC-jav.py program 79k lépésig történő futtatása után.

Feladatok

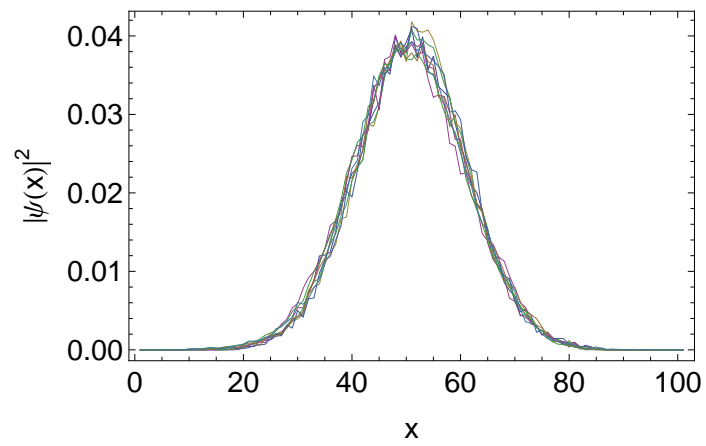
1. feladat. Ábrázoljunk néhány a szimulációval generált pályát a klasszikus pályával együtt.



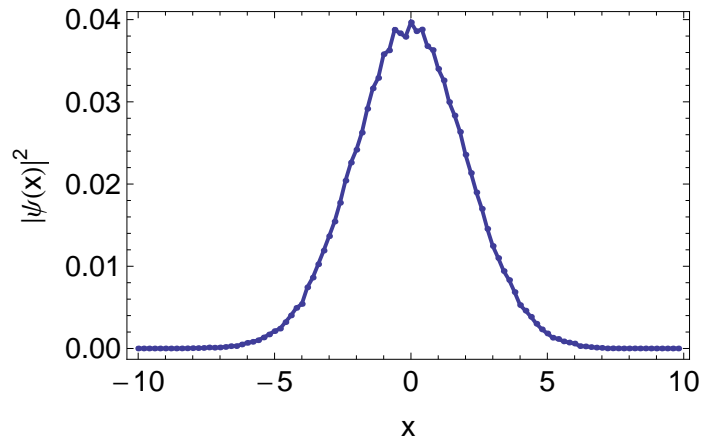
5. ábra.



6. ábra.

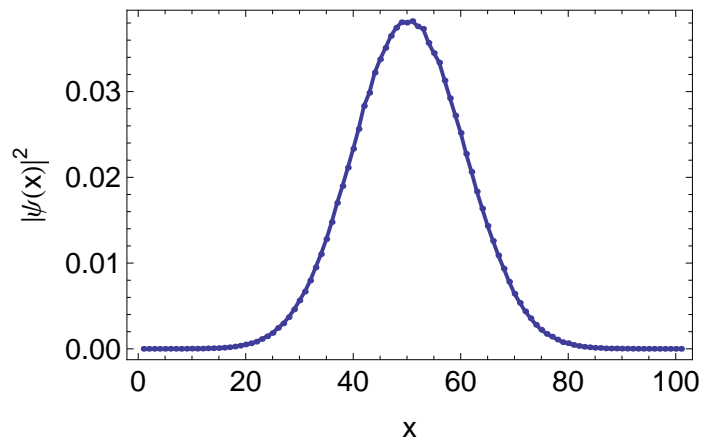


7. ábra. Több, 30k körüli lépésszámú szimuláció hullámfüggvényei.



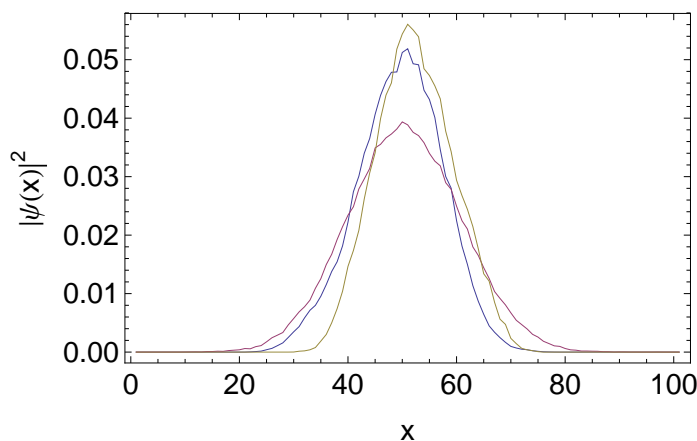
8. ábra. 9 db, 30k körüli lépésszámú szimuláció hullámfüggvényeiből kapott átlag.

2. feladat. Tegyük folytonosabbá a hullámfüggvényt δ csökkentésével, a hullámfüggvény értékének egy adott pontban való pontosabb becslése érdekében használjunk kisebb ϵ időbeli lépésközt, vagy futtassuk a szimulációt hosszabb ideig.

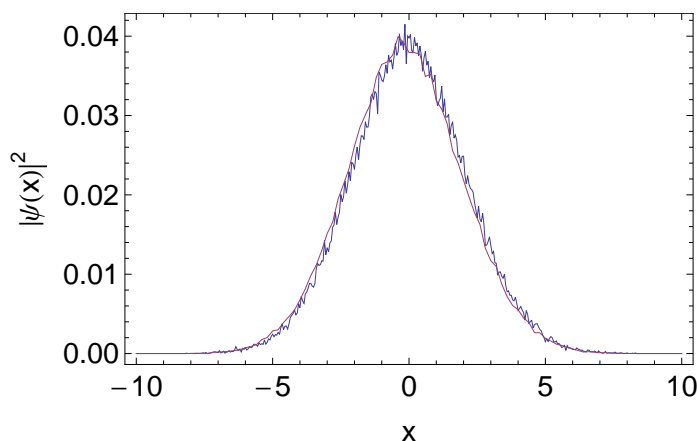


9. ábra. A hullámfüggvény abszolútérték négyzete $1.8 \cdot 10^6$ lépés után. Látható, hogy tényleg pontosabbak lettek az értékek.

Az ϵ csökkentésével viszont inkább romlást sikerült elérni a hullámfüggvény értékeiben, ez véleményem szerint azért van, mert ilyenkor (a $\epsilon = 0.1$) az oszcillátor periódus ideje túl közel kerül a szimulált időtartamhoz, és ilyenkor a könyv levezetése szerint nem az alapállapot hullámfüggvényét látjuk, hanem annak, és gerjesztett állapotoknak valamilyen keverékét. A sűrűség ilyenkor általában nem szimmetrikus és néha két csúccsal is kialakul.



10. ábra. A piros vonal az eredeti, $\epsilon = 1$ melletti hullámfüggvény, a másik kettőnél $\epsilon = 0.1$. Mindegyik kb 150k lépésig futott.



11. ábra. A piros görbe δ , a kék görbe $\frac{\delta}{4}$ térbeli felbontással készült. Ez egyébként csak a hullámfüggvényt befolyásolja, a szimuláció során az x értékek valós számok.

3. feladat. Mivel az alapállapot hullámfüggvényben nincsen előjelváltás, elhagyhatjuk a fázist és ezért feltételezhetjük, hogy $\psi(x) = \sqrt{\psi(x)^2}$. Használjuk ezt arra, hogy megbecsüljük a hullámfüggvényhez tartozó energiát a következő képlet segítségével

$$E = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} = \frac{\omega}{2\langle \psi | \psi \rangle} \int_{-\infty}^{\infty} \psi^*(x) \left(-\frac{d^2}{dx^2} + x^2 \right) \psi(x) dx$$

ahol a deriváltat numerikusan számoljuk.

Mivel az euler módszerrel való deriválásnál az első derivált még viszonylag használható, de a másodiknál már nagyon nagy a hiba, így nem ezzel a módszerrel deriváltam. Ehelyett egy egymillió lépéshosszú szimulációból kapott hullámfüggvény abszolútérték

négyzetére illeszttem egy $a \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ alakú hullámfüggvényt, és ennek gyökét vettem $\psi(x)$ analitikus változatának.

$$\psi_{analytic}(x) = 0.195545 \sqrt{e^{-0.114748(0.113287+x)^2}}$$

Ezután a feladatban megadott integrált már szimbolikusan is el lehetett végezni

$$E = \int_{-\infty}^{\infty} \psi_{analytic}(x) \left(-\frac{d^2}{dx^2} + x^2 \right) \psi_{analytic}(x) dx = 0.431448\omega$$

Természetes egységekben (hossz: $\sqrt{\hbar/m\omega} = 1$, energia $\hbar\omega$) az energia várt értéke

$$E_n = n + \frac{1}{2}$$

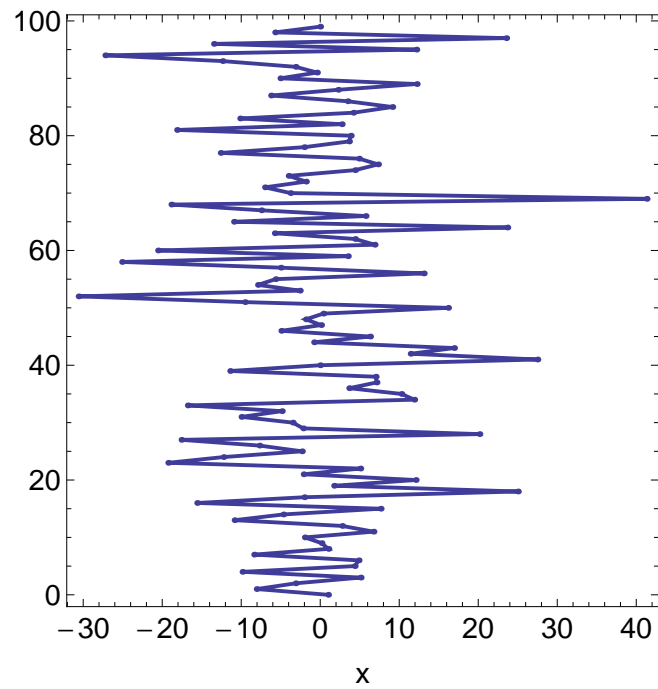
tehát a várt 0.5-höz igen közeli értéket kaptunk.

4. feladat. Vizsgáljuk meg hogy mi történik, ha Planck állandót nagyobbak vesszük. Jobban, vagy kevésbé lesz ezáltal robosztus a szimuláció a klasszikus trajektória megtalálására való képessége?

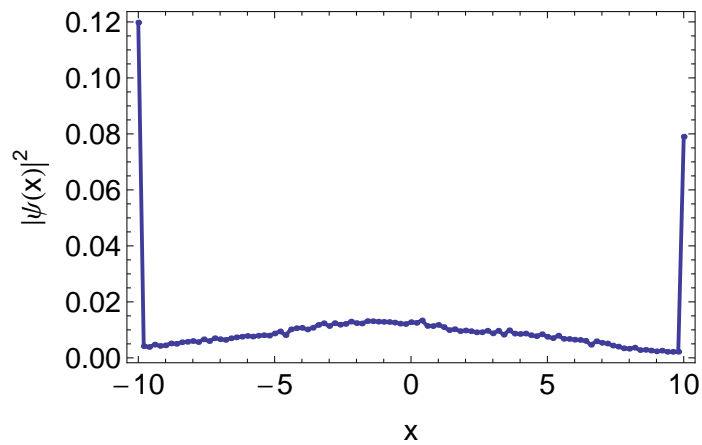
Tekintve hogy

$$e^{-\frac{\epsilon}{\hbar}} = e^{-\frac{\epsilon}{k_B T}}$$

látható, hogy \hbar a Boltzmann-faktorban a hőmérséklettel arányos, így tehát a növelése a véletlenszerű fluktuációkat erősíti, illetve növeli az energiaszintet. Ez annak köszönhető, hogy a Metropolis algoritmusban gyakorlatilag minden felvetett változás elfogadásra kerül, mivel az exponenciális tagnak mindig 1 körüli értéke van. Ez a szimuláció futtatásából is látszik.



12. ábra.



13. ábra.

5. feladat. Vizsgáld meg a gravitációs potenciálban pattogó kvantumos részecske esetét is (*quantum bouncer*).

A pattogó kvantumos részecske szimulált egy pályája és a megtalálási valószínűség a könyv *QMCBouncer.py* programja alapján

