



ELTE

EÖTVÖS LORÁND
UNIVERSITY

Feynman-féle pályaintegrálok numerikus vizsgálata

Írta:

Fébert Fanni Adina (IVHEKT),
Klettner Lilla Barbara (KTJ29M),
Purzsa Aletta (C2LR62)

Tartalomjegyzék

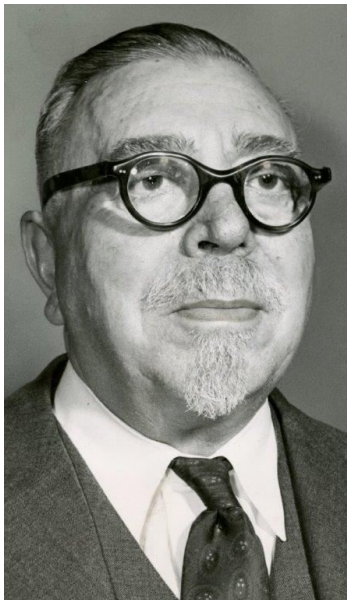
1. Bevezetés	2
1.1. Az pályaintegrálokról általában	2
1.2. A Feynman-féle megközelítés	3
2. Numerikus vizsgálat	6
2.1. A numerikus vizsgálat célja	6
2.2. A kód működéséről	6
2.3. Potenciálok vizsgálata	9
2.3.1. Harmonikus oszcillátor	9
2.3.2. Pöschl-Teller potenciál	9
2.3.3. Morse-potenciál	10
2.3.4. Lépcsőpotenciál	11

1. Bevezetés

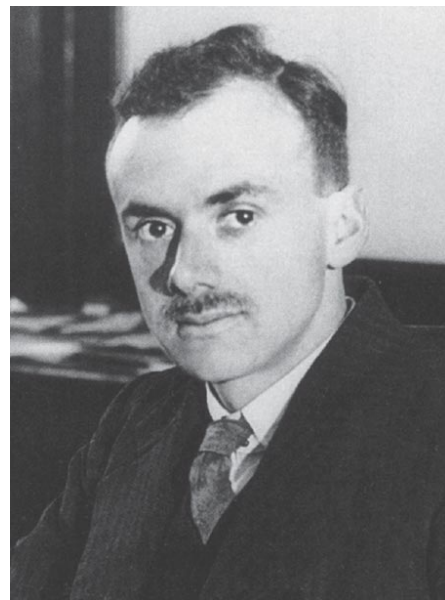
1.1. Az pályaintegrálokról általában

A kvantummechanikában az út- vagy pályaintegrál-formalizmus egy olyan megfogalmazási mód, mely a klasszikus mechanika működési elvét általánosítja [6], és döntő fontosságúnak bizonyult az elméleti fizika későbbi fejlődésének szempontjából. Jelentőségének oka, hogy általa a Lorentz-kovariancia ¹ sokkal könnyebben megvalósítható, mint a szintén a klasszikus mechanikát alapul vevő kanonikus kvantálással.

Az útintegrál megfogalmazásának alapötlete Norbert Wienertől (1a) származik [4], aki a Brown-mozgás és különböző diffúziós problémák vizsgálata során bevezette az úgynevezett Wiener-integrált. Elképzelését Paul Dirac (1b) 1933-ban kiterjesztette a kvantummechanikában való használatra is, azonban a teljes módszer kifejlesztése Richard Feynman (2) nevéhez fűződik.



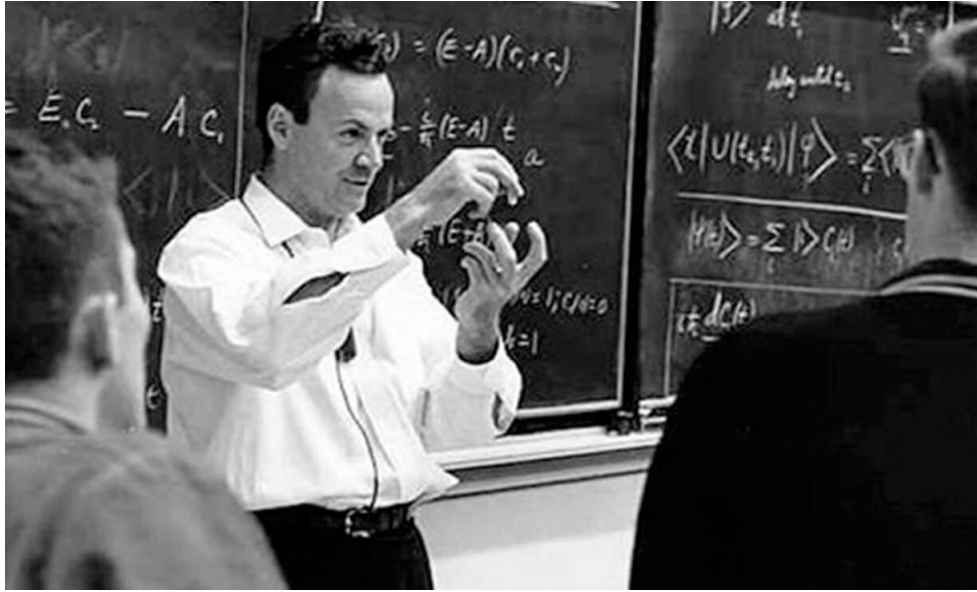
(a) Norbert Wiener



(b) Paul Dirac

1. ábra. Wiener és Dirac portréja [8]

¹A megfigyelés vagy megfigyelési szimmetria ekvivalenciája. A speciális relativitás-elméleten alapul, és lényegében azt jelenti, hogy „a fizika törvényei ugyanazok maradnak minden megfigyelő számára, akik egymáshoz képest mozognak a tehetetlenségi kereten belül”. [3]



2. ábra. Richard Feynman [8]

1.2. A Feynman-féle megközelítés

Feynman a pályaintegrál-formalizmus néhány előzményét már a doktori munkája során kidolgozta, 1948-ra pedig a teljes módszer kifejlesztésével elkészült.

A pályaintegrál-formalizmus a kvantumamplitúdó kiszámításához a klasszikus formulát - mely szerint egyetlen, egyedi klasszikus pálya egy rendszerre vonatkozik - kvantummechanikailag lehetséges trajektóriák végtelen összegével vagy funkcionális integrállal helyettesíti. [4]

Dirac munkája nem adott pontos előírást a pályák összegének kiszámítására, illetve nem mutatta be azt sem, hogy a Schrödinger-egyenletet vagy a kanonikus kommutációs összefüggéseket vissza lehet-e állítani ebből a szabályból. Feynman ezen hiányosságok pótlását tűzte ki célul.

Módszere analóg a hullámoptikával. [6] Így sokáig azt hitték, hogy a kvantummechanikának egy Schrödingertől teljesen független felépítését adja, azonban ez a feltevés nem bizonyult igaznak. Feynman azt javasolta, hogy a kvantummechanika az alábbi *posztulátumok* segítségével kerüljön „felépítésre” [4]:

1. Egy esemény valószínűségét a valószínűségi amplitúdó - mely egy komplex szám - négyzetes modulusa adja meg.
2. A valószínűségi amplitúdót úgy kapjuk meg, hogy összeadjuk a konfigurációs térben lévő összes útvonal „hozzájárulását”.
3. Egy adott pálya „hozzájárulása” $e^{i\frac{S}{\hbar}}$ -val arányos, ahol S a hatásintegrál.

Egy adott folyamat teljes valószínűségi amplitúdójának meghatározásakor integráljuk a 3. posztulátum amplitúdóját a kezdeti- és a végállapotok között a rendszer összes lehetséges útvonalának

terére (beleértve a klasszikus meggondolások alapján abszurdnak mondható eseteket is).

Vegyünk egy részecske útját a téridőben. Az (x_0, t_0) kezdőpont és egy lehetséges jövőbeli (x', t') végpont között a szabad részecske útját leíró hullámfüggvény [2]:

$$\Phi(x', t') = \int dx_0 G(x', t'; x_0, t_0) \Phi(x_0, t_0), \quad (1)$$

ahol G a megfelelő Green-függvény:

$$G(x', t'; x_0, t_0) \equiv \sqrt{\frac{m}{2\pi i(t' - t_0)}} \exp\left(i \frac{m(x' - x_0)^2}{2(t' - t_0)}\right) \quad (2)$$

Feynman a részecske (Φ) valószínűségi amplitúdóját az (x', t') pontban egyenlővé tette az összes olyan úttal a téridőn keresztül, mely t_0 időpillanatban kezdődik és t' időpillanatban végződik. Ez szavakkal kifejezve éppen azt jelenti, hogy minden út lehetséges, csak eltérő valószínűséggel. Az egyes utak bejárások valószínűségei a legkisebb hatás elve alapján alakulnak.

Legyen $x(t)$ a trajektória. A klasszikus eset alapján t_0 és t' időpillanat között az $S[x(t)]$ hatás értéke extrémum [2]:

$$\delta S[x(t)] = S[x(t) + \delta x(t)] - S[x(t)] = 0 \quad (3)$$

Ha az út a kezdőponttól a végpontba tart, akkor:

$$\delta(x_0) = \delta(x') = 0 \quad (4)$$

Amennyiben S egy L funkcionál vonalintegrálja a pálya mentén, a klasszikus mechanikai formalizmus ekvivalens lesz a newtoni differenciálegyenletekkel.

$$\delta S[x(t)] = \int_{t_0}^{t'} dt L[x(t), \dot{x}(t)], L = T[x, \dot{x}] - V[x], \quad (5)$$

ahol T a kinetikus energia, V pedig a potenciális energia. [2]

Ebből Feynman arra a következtetett, hogy a klasszikus hatás szabad részecskére $V = 0$ esetén:

$$S[0,'] = \frac{m}{2} (\dot{x})^2 (t' - t_0) = \frac{m(x' - x_0)^2}{2(t' - t_0)}, \quad (6)$$

mely alapján felírható az alábbi Green-függvény:

$$G(t', 0) = \sqrt{\frac{m}{2\pi i(t' - t_0)}} e^{iS[t', 0]/\hbar} \sum_{\text{paths}} e^{iS[b, a]/\hbar} (\text{pathintegral}) \quad (7)$$

Ez kapcsolatot teremt a klasszikus mechanika és a kvantummechanika között.

Az itt szereplő szummára gyakran útintegrálként hivatkozunk, hiszen ennek minden eleme egy hatásintegrál. Bár minden útnak van valamekkora valószínűsége (szerepel a szummában), a klasszikus $x(t)$ trajektória körüli utak lesznek a legjelentősebbek, a klasszikus esetben ($\hbar \rightarrow 0$) pedig csak egy

trajektória lesz, mely egyenlő lesz a legkisebb hatással.

A fentieket kicsit másképp is megfogalmazhatjuk a következő módon [5] :
Nézzük egy részecske útját két téridőpont között: (x', t') és (x_0, t_0) . Ekkor a valószínűségi átmenet amplitúdója a hullámfüggvény ezen pontjai között:

$$U(x', t'; x_0, t_0) = \langle \phi(x', t') | \phi(x_0, t_0) \rangle \quad (8)$$

Ha a Hamilton függvény nem expliciten időfüggő, akkor az első időpontot vehetjük $t_0 = 0$ -nak, így csak az eltelt időt kell figyelembe vennünk: $t = t - t_0$. Hogy ezt illusztráljuk, a 8 egyenletet gyakran szokás $U(x', t; x_0)$ formában felírni. A fenti propagátor teljes egészében leírja egy rendszer időbeli változását. Szokás még az U helyett a K jelölést használni és “kernel”-nek vagy “Feynman kernel”-nek nevezni. A pályaintegrál egy explicit módja ezen propagátor előállításának.

Vegyünk egy részecske lehetséges $x(t)$ trajektóriáit egy időfüggetlen $V(x)$ potenciálon a fix (x', t') és (x_0, t_0) végpontokkal. Ilyen trajektóriákból végtelen mennyiségű lehet, mindegyik klasszikus hatása $S[x(t)]$. Feynman azt állítja, hogy egy adott trajektória hozzájárulása a propagátorhoz $i \frac{S[x(t)]}{\hbar}$. Tehát minden lehetséges pálya azonos amplitúdóval, de a klasszikus hatásukhoz kötődő fázissal járul hozzá a propagátorhoz. Az összes lehetséges trajektóriát összegezve megkapjuk a propagátort.

$$U(x', t; x_0) = A(t) ; \quad \sum_{\text{paths}} \exp \left[S[x(t)] \frac{i}{\hbar} \right] \quad (9)$$

Ez az egyenlet a pályaintegrál-formalizmus alapja. [5]

2. Numerikus vizsgálat

2.1. A numerikus vizsgálat célja

Numerikusan nem lehet az összes pályát szimulálni, de mivel a hatásokhoz tartozó valószínűségek exponenciálisan csengenek le, ezért jó eredményt kapunk akkor is, ha a hatásminimumhoz tartozó pálya kis környezetében vizsgálódunk. Ehhez az úgynevezett Metropolis-Hastings algoritmust fogjuk használni, melynek lényege, hogy egy eloszlásból mintavételezünk, és az algoritmus véletlenszerű valószínűséggel fogadja el a változást. Célunk, hogy véges sok iterálás után közelítőleg visszakapjuk a részecske valószínűség-sűrűségét a hely függvényében.

2.2. A kód működéséről

Az alábbi C++ kód a pályaintegrál-formalizmus alapján szimulálja az adott potenciálhoz tartozó valószínűségi sűrűséget a hely függvényében. A pályának térben és időben diszkrét rácsponthoz feleltetünk meg, majd véletlenszerűen választjuk ki, hogy a pálya melyik elemét variáljuk. Az új pályára kiszámítjuk az energiát, és aszerint fogadjuk- vagy utasítjuk el a változást. Mindegyik rácsponthoz feleltetünk meg egy valószínűséget, és ahányszor a szimuláció elfogad egy adott rácsponthoz, az ahhoz tartozó valószínűséget megnöveljük eggyel. Ezt ismétljük meg, jelen esetben hárommilliószor. Az egyszerűség kedvéért a program során \hbar értékét 1-nek vettük, így a végeredményt is ennek megfelelő egységekben kaptuk.

```
1 #include<iostream>
2 #include<array>
3 #include<vector>
4 #include<numeric>
5 #include<cmath>
6 #include<algorithm>
7 #include<random>
8 #include<fstream>
9 #include<string>
10 using namespace std;
11
12 double V(double x, double m = 1){return 0.5*m*x*x;}
13
14 double E(std::vector<double> x, double m=1, double epsilon=1)
15 {
16     int N = x.size();
17     double sums = 0;
18     for(int i=1; i < N; i++)
19     {
20         sums += (m/2)*((x[i]-x[i-1])/epsilon)*((x[i]-x[i-1])/epsilon) +
21         V((x[i]+x[i-1])/2, m);
22     }
23     return sums;
24 }
25 double generate_random_number(double x1, double x2)
26 {
```

```

27     std::random_device rd;
28     std::mt19937 gen(rd());
29     std::uniform_real_distribution<> dis(x1, x2);
30     return dis(gen);
31 }
32 int main() {
33     int steps = 300000;
34     int N = 1000;
35     double m = 1;
36     double epsilon = 1;
37     std::vector<double> path(N, 0.0);
38     std::vector<double> prob(N, 0.0);
39     double oldE = E(path, m, epsilon);
40     int counter = 0;
41
42     while (counter < steps)
43     {
44         double r = generate_random_number(0,1);
45         int element = (int)N*r;
46         double w = generate_random_number(0,1);
47         double change = generate_random_number(-1,1);
48         path[element] += change;
49         double newE = E(path, m, epsilon);
50
51         if (newE > oldE and exp(oldE*epsilon-newE*epsilon) <= w)
52         {
53             path[element] -= change;
54         }
55         int elem = int(path[element]*16 + 50);
56
57         if (elem < 0){elem = 0;}
58         if (elem > N){elem = N;}
59         prob[elem] += 1;
60         oldE = newE;
61         counter = counter + 1;
62     }
63
64     fstream file;
65     file.open("vector_file.txt", ios_base::out);
66
67     for(int i = 0; i < prob.size(); i++)
68     {
69         file<<prob[i]<<endl;
70     }
71     file.close();
72     return 0;
73 }

```

A könyvtárak importálása után a 12-31 sorok a szükséges függvényeket definiálják, először definiáljuk a potenciált (jelen esetben harmonikus oszcillátor), utána pedig ennek használatával hozunk létre egy függvényt, ami meghatározza az energiát egy adott pályára. Az **m** változó a

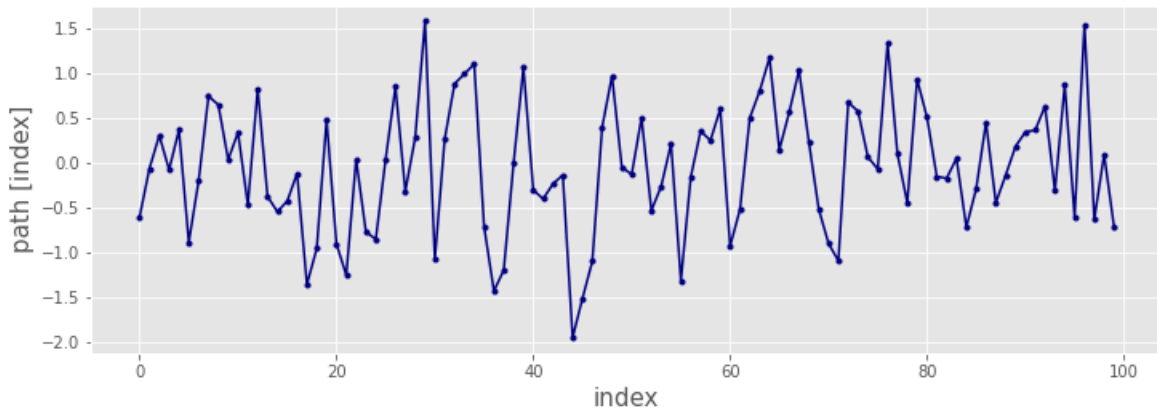
tömeget, míg az **epsilon** az egyes rácspontok között eltelt időegységet jelöli. Nagy szükségünk van továbbá egy véletlenszám generáló függvényre is, mely egy adott intervallumon belül, egyenletes eloszlással választ valós számokat.

A **main** függvényben először definiáljuk a szükséges változókat. A **steps** az iterálások, az **N** pedig a rácspontok számát jelöli. Az egyszerűség kedvéért a tömeget, és az időintervallumok távolságát 1-nek vettük. Két vektort fogunk használni, a **path** tartalmazza az aktuális pálya elemeit, a **prob** jelöli az egyes rácspontokhoz tartozó valószínűségeket. Az energiaértékeket a **oldE** és **newE** változóknak fogjuk megfeleltetni. Az előbbi az aktuális, míg az utóbbi a program által „javasolt” energiát fogja tartalmazni.

A **while** ciklus tartalmazza mindazt, amit ismételni szeretnénk **N**-szer. Először is az **r** véletlen szám segítségével választjuk ki, hogy a pálya melyik indexű elemét szeretnénk megvariálni, ezt az indexet eltároljuk az **element** változóban, majd a hozzá tartozó pályaelemet a **change** értékével megvariáljuk, majd az így kapott pályára is kiszámoljuk az energiát. Ezt a változtatást csak akkor fogadjuk el, ha az így kapott energiaérték kisebb, mint az eredeti, illetve még a „jó” energiák közül sem az összeset fogadjuk el, hanem a **w** változó segítségével mintavételezünk belőle. Ez a Metropolis-Hastings algoritmus lényege. Az 51-54 sorok tartalmazzák azt, hogy amikor a variált pálya nem felel meg a feltételeknek, akkor visszaírjuk az eredeti értéket.

Az 55. sor annak felel meg, hogy a pálya értékeit megfeleltetjük **N** darab rácspontnak egy lineáris transzformációval, így kapjuk az **elem** változót. Mivel index jellegű mennyiségről van szó, ezért a túlszordulást megakadályozzuk az 57-58 sorokban, ezután pedig az adott pályaelemhez tartozó valószínűséget megnöveljük eggyel. Végezetül frissítjük az energiaértéket a következő iterációhoz.

A 64-73 sorok tartalmazzák a **prob** vektor fájlba történő kiíratását.



3. ábra. A **path** vektor elemei **N**=100 és **steps**=3000000 esetén harmonikus oszcillátorra

2.3. Potenciálok vizsgálata

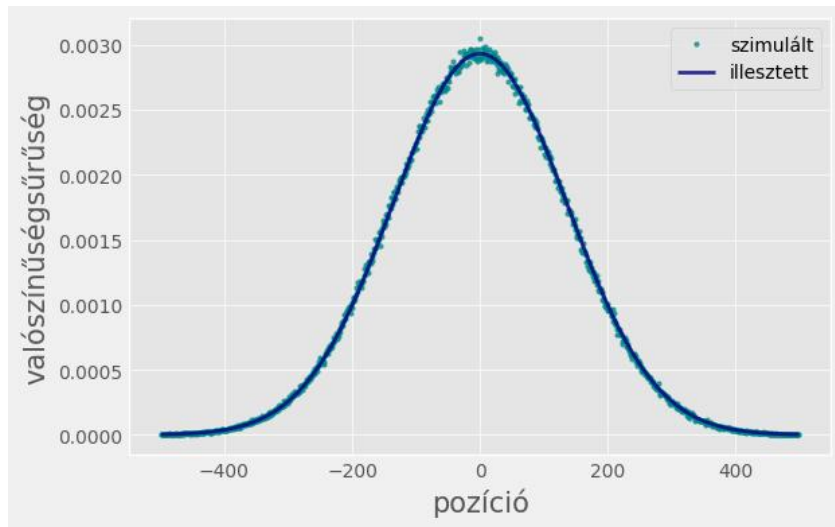
2.3.1. Harmonikus oszcillátor

A kód működésének tesztelését a harmonikus oszcillátorral végeztük, mivel ennek jól definiált az analitikus megoldása, így könnyedén leellenőrizhető a programunk helyessége. A kapott adatsorra illesztettük a potenciálhoz tartozó alapállapotú valószínűségi sűrűség-függvényt, mely a következő alakú:

$$\rho(x) = Ae^{-x^2/c^2} \quad (10)$$

ahol A és c az illesztendő konstansok, x a helykoordináta.

Így az ellenőrzés mellett a numerikus módszerünk pontosságára is tudunk egy becslést adni az illesztés hibájából.



4. ábra. Harmonikus oszcillátor alapállapotához tartozó valószínűségi sűrűsége a hely függvényében

Az illesztett paraméterek relatív hibái:

$$\delta_A = 0.00072026$$

$$\delta_c = 0.00083171$$

Ezek alapján azt mondhatjuk, hogy a szimulációnkkal a harmonikus oszcillátor alapállapotú valószínűségi sűrűség-függvényét körülbelül 0,83% pontossággal tudtuk meghatározni.

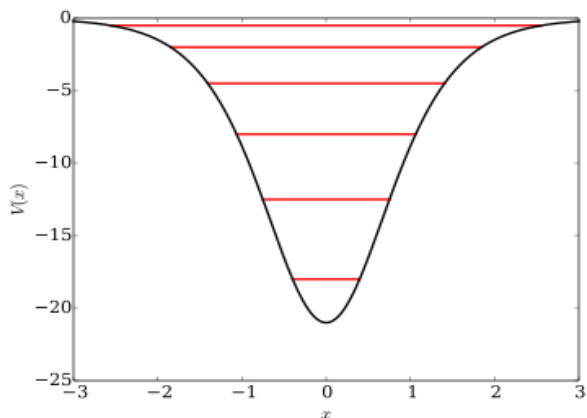
2.3.2. Pöschl-Teller potenciál

Hasonló görbét kapunk, ha a harmonikus oszcillátor helyett a Pöschl-Teller potenciált vizsgáljuk:

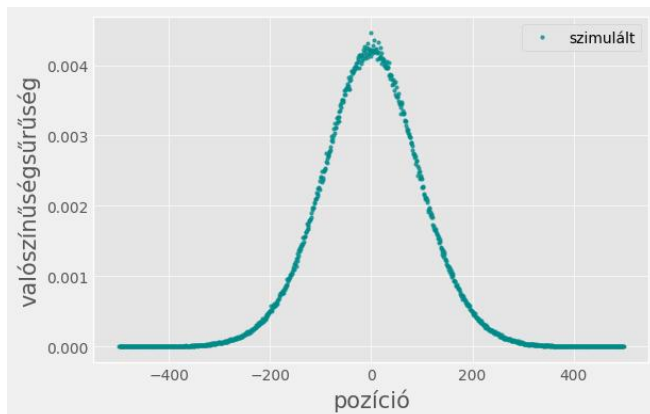
$$V(x) = -\frac{\lambda(\lambda + 1)}{2} \text{sech}^2(x) \quad (11)$$

ahol $\lambda = 1, 2, 3, \dots$

Az 5b ábra alapján látszik, hogy olyasmi adatsort kaptunk, mint a harmonikus oszcillátornál, ennek oka, hogy a két potenciál alakilag nagyon hasonlít egymásra a vizsgált tartományban.



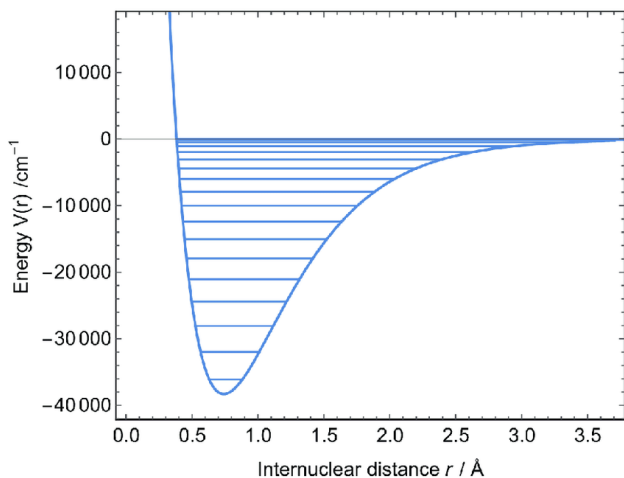
(a)



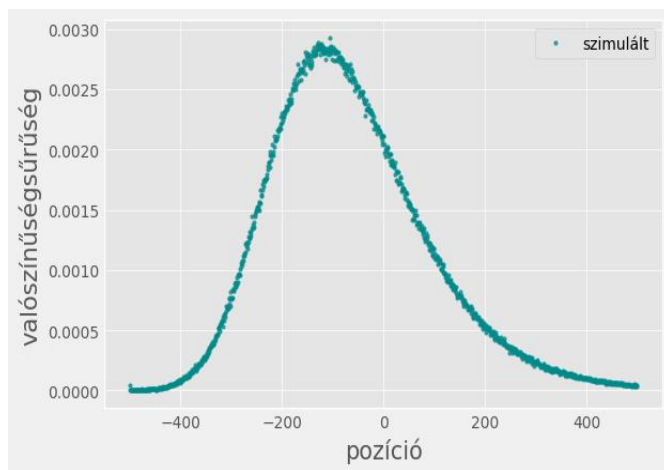
(b)

5. ábra. A Pöschl-Teller potenciál alakja (balra) [9], illetve a szimuláció által kapott valószínűsűrűség-függvény $\lambda = 7$ esetén (jobbra). A vízszintes piros vonalak a különböző λ értékekhez tartoznak.

2.3.3. Morse-potenciál



(a)



(b)

6. ábra. A Morse-potenciál alakja (balra) ismeretlen paraméterek mellett, illetve a szimulációval kapott valószínűsűrűség-függvényünk (jobbra). A bal oldali ábrán a vízszintes vonalak a sajátenergia-értékeket jelölik, melyek a harmonikus potenciállal ellentétben, nem egyenletesen helyezkednek el.

A Morse-potenciál molekulák kötési energiájának kiszámításában használatos. A valóságban a két atom között lévő kölcsönhatás nem szimmetrikus, hanem a molekulák anharmonikus oszcillá-

torként viselkednek, melyet jól leír a Morse-potenciál, melynek háromdimenziós alakja:

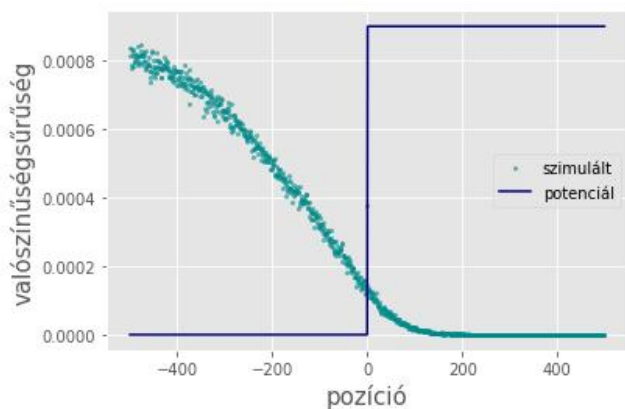
$$V(r) = D_e(1 - e^{-a(r-r_e)})^2 \quad (12)$$

ahol D_e , a és r_e a molekulát jellemző paraméterek. A programunkkal szimuláltuk $D_e = 2$, $a = 1.25$ és $r_e = 1.25$ értékekkel a részecske valószínűségi sűrűség-függvényét. A kapott eredmény a 6b ábrán látható.

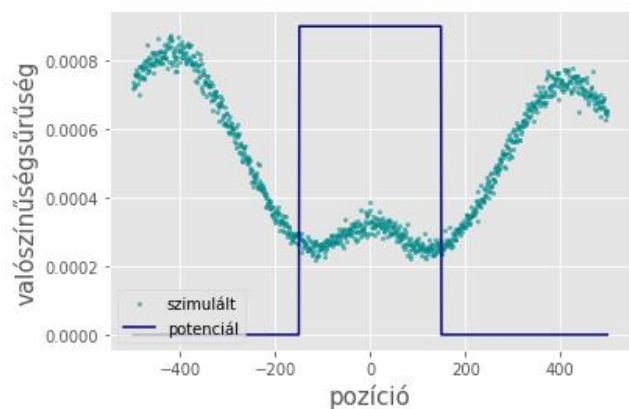
Ebben az esetben az figyelhető meg, hogy a valószínűségi sűrűség-függvény a potenciálhoz hasonlóan nem szimmetrikus, de úgy tűnik, hogy egy (számunkra ismeretlen) analitikus függvényhez konvergál.

2.3.4. Lépcsőpotenciál

A lépcsőpotenciálnál három esetet vizsgáltunk. Az első esetben a teljes pozitív félegyenesen egy konstans potenciál van, míg a másik esetben csak a nulla környezetében egy intervallumon van nem nulla értéke a potenciálnak. A harmadik eset a második inverze, tehát a középső intervallumon nulla a potenciál, a többi helyen pedig konstans. A kapott eredmények a 7 és 8 ábrákon láthatóak.



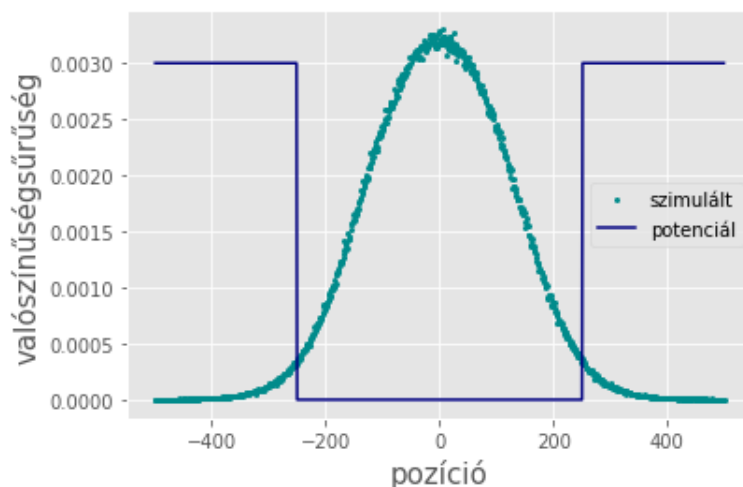
(a)



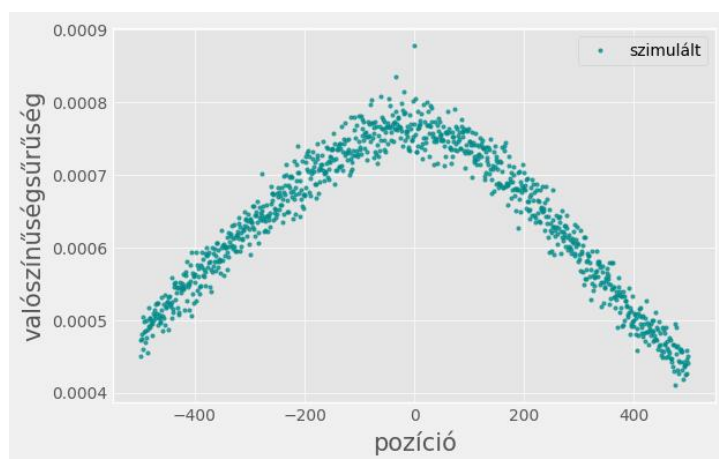
(b)

7. ábra. Különböző lépcsőpotenciálok esetén szimulált valószínűségi sűrűség-függvények. Mindkét ábrán megfigyelhető az alagúteffektus, azaz a részecske valamekkora valószínűséggel áthatolhat a potenciálgáton, ami a klasszikus formalizmus szerint lehetetlen. A jobb oldali ábrán ezen kívül jól megfigyelhető a hullámforma alak, amit az analitikus megoldásból is várnánk.

Ami szembevető különbség az előző, sima potenciálokhoz képest, hogy itt az értékek jóval nagyobb szórást mutatnak a lépcsőn kívül. Ennek oka, hogy itt szabad részecskére végezzük a szimulációt, melynek a valószínűségi sűrűség-függvénye a numerikus kiértékelés miatt nem fog véges sok iterálás után egy analitikus görbébe konvergálni. Összehasonlításképpen a 9 ábrán látható a szabad részecskére vonatkozó eredmény.



8. ábra. Véges potenciálgödör. Az előzőekhez hasonlóan itt is visszakaptuk a szimulációval az alagúteffektust.



9. ábra. Szabad részecske valószínűsűrűség-függvénye a szimuláció alapján

Hivatkozások

- [1] Govind Jha, Raghav *On the Numerical Simulations of Feynman's Path Integrals using Markov Chain Monte-Carlo with Metropolis-Hastings Algorithm*. Kolkata. 2011.
- [2] Landau, R. H., Páez, M. J. & Bordeianu, C. C. *Computational Physics - Problem Solving with Computers*. WILEY-VCH Verlag GmbH & Co. KGaA. 2007.
- [3] "*Lorentz covariance*". Wikipedia, Wikimedia Foundation. Last edited on 7 October 2022, at 17:39 (UTC). Available at: https://en.wikipedia.org/wiki/Lorentz_covariance
- [4] "*Path integral formulation*". Wikipedia, Wikimedia Foundation. Last edited on 5 December 2022, at 12:58 (UTC). Available at: https://en.wikipedia.org/wiki/Path_integral_formulation

- [5] Perepelitsa, D. V. *Path Integrals in Quantum Mechanics*. 70 Amherst Ave. Cambridge, MA 02142
- [6] Vanó L., Tajkov Z., *Kvantummechanika A - Jegyzet Katz Sándor előadása alapján*. Eötvös Loránd Tudományegyetem. 2015.
- [7] Westbroek, M. J. E., King, P. R., Vvedensky, D. D., Dürr, S. *User's guide to Monte Carlo methods for evaluating path integrals*. Imperial College London, London SW7 2BP, United Kingdom & The Blackett Laboratory, Imperial College London, London SW7 2AZ, United Kingdom & Germany and Jülich Supercomputing Centre, Forschungszentrum Jülich, D-52425 Jülich, Germany. 2018.
- [8] A képek forrásai: <https://media.wnyc.org>, <https://physicsworld.com> és <https://jamesclear.com>
- [9] A referencia-ábrák forrása: <https://www.researchgate.net>