

# Całki po trajektoriach Feynmana

Wstęp do modelowania komputerowego

Adrianna Saribekyan, Ewelina Kowal, Maciej Kucharski

29 stycznia 2022

# Spis treści

1	Wiadomości wstępne	2
1.1	Cel pracy . . . . .	2
1.1.1	Problem . . . . .	2
1.2	Wstęp teoretyczny . . . . .	2
1.2.1	Czasoprzestrzenne rozchodzenie się fal Feynmana - teoria . . . . .	2
1.2.2	Funkcja falowa stanu związanego . . . . .	4
1.3	Całkowanie po trajektoriach siatki . . . . .	4
2	Implementacja algorytmów w Pyhonie	7
2.1	Assesment . . . . .	8
3	Podsumowanie	9
4	Bibliografia	9

# 1 Wiadomości wstępne

## 1.1 Cel pracy

Celem było poznanie teorii Feynmana oraz całkowania po trajektoriach siatki. Należało następnie przedstawić kod określający prawdopodobieństwo stanu podstawowego poprzez całkowanie po trajektoriach Feynmana przy użyciu algorytmu Metropolis.

### 1.1.1 Problem

Przyczepiona do liniowej sprężyny cząstka w podejściu klasycznym, zostaje wprowadzona w prosty ruch harmoniczny z położeniem w przestrzeni w funkcji czasu podanym przez  $x(t) = A \sin(\omega_0 t + \phi)$ . Naszym zadaniem jest wygenerowanie kwantowej funkcji falowej  $\psi(x, t)$  za pomocą funkcji  $x(t)$ .

## 1.2 Wstęp teoretyczny

### 1.2.1 Czasoprzestrzenne rozchodzenie się fal Feynmana - teoria

Szukając bardziej bezpośredniego sformułowania mechaniki klasycznej niż teoria Schroedingera, Feynman poszedł krokami Diraca, który mówił, że zasada najmniejszego działania Hamiltona może być granicą  $\hbar \rightarrow 0$  kwantowej zasady najmniejszego działania.

Zauważywszy, że zasada Hamiltona wyjaśnia drogę cząsteczek przez czasoprzestrzeń, Feynman postulował, że kwantowa funkcja falowa opisująca propagację cząstki swobodnej z punktu w czasoprzestrzeni  $a = (x_a, t_a)$  do  $b = (x_b, t_b)$  może być opisana zależnością:

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

gdzie  $G$ , to funkcja Greena:

$$G(x_b, t_b; x_a, t_a) \equiv \sqrt{\frac{m}{2\pi i(t_b - t_a)}} \exp\left[i \frac{m(x_b - x_a)^2}{2(t_b - t_a)}\right]. \quad (2)$$

Feynman zauważył, że innym sposobem interpretacji równania 1 jest przedstawienie zasady Hamiltona, w której amplituda prawdopodobieństwa (funkcja falowa  $\psi$ ) dla cząstki, aby się znalazła w punkcie B, jest równa sumie wszystkich ścieżek przez czasoprzestrzeń rozpoczynających się w czasie A i kończących się w B (rys. 1).

To podejście uwzględnia statystyczną naturę mechaniki kwantowej - posiadamy różne możliwe drogi cząstki, o różnym prawdopodobieństwie. Tutaj możemy zauważyć różnicę między teorią Schroedingera, a teorią Feynmana. Ta pierwsza rozwiązuje funkcje falowe i uważa ścieżki za klasyczną koncepcję.

Wartości prawdopodobieństwa dla danej drogi można wyznaczyć korzystając z zasady najmniejszego działania Hamiltona.

Najbardziej ogólny ruch cząstki poruszającej się po klasycznej trajektorii  $\bar{x}(t)$  od czasu  $t_a$  do  $t_b$  odbywa się po torze, takim że działanie  $S[\bar{x}(t)]$  jest ekstremum:

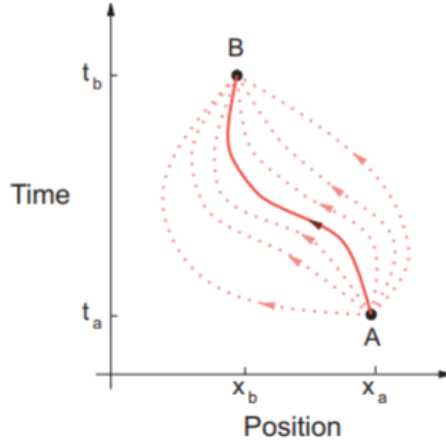
$$\delta S[\bar{x}(t)] = S[\bar{x}(t) + \delta x(t)] - S[\bar{x}(t)] = 0, \quad (3)$$

gdzie droga odbywa się między punktem A i B, czyli  $\delta(x_a) = \delta(x_b) = 0$ .

Jeśli działanie  $S$  przyjmuje się jako całkę krzywoliniową Lagrangianu wzdłuż:

$$S[\bar{x}(t)] = \int_{t_a}^{t_b} L[x(t), \dot{x}(t)] dt, \quad L = T[x, \dot{x}] - V[x], \quad (4)$$

gdzie  $T$  jest energią kinetyczną,  $V$  energią potencjalną, to to sformułowanie mechaniki klasycznej jest równoważne równaniom różniczkowym Newtona.



Rysunek 1: Trajektoria cząstki - wykres czasu od położenia.

Feynman zaobserwował, że klasyczne zachowywanie się swobodnej cząstki ( $V = 0$ )

$$S[b, a] = \frac{m}{2} (\dot{x})^2 (t_b - t_a) = \frac{m}{2} \frac{(x_b - x_a)^2}{t_b - t_a}, \quad (5)$$

jest powiązane z propagatorem cząstki swobodnej (2) przez

$$G(b, a) = \sqrt{\frac{m}{2\pi i(t_b - t_a)}} \exp i \frac{S[b, a]}{\hbar}. \quad (6)$$

Równanie 6 jest ważnym związkiem między mechaniką kwantową, a zasadą najmniejszego działania. Po tym przełomie, Feynman postulował przeformułowanie mechaniki kwantowej, która uwzględniałaby jej aspekty statystyczne, wyrażając  $G(a, b)$  jako:

$$G(b, a) = \sum_{paths} \exp i \frac{S[b, a]}{\hbar} \quad (7)$$

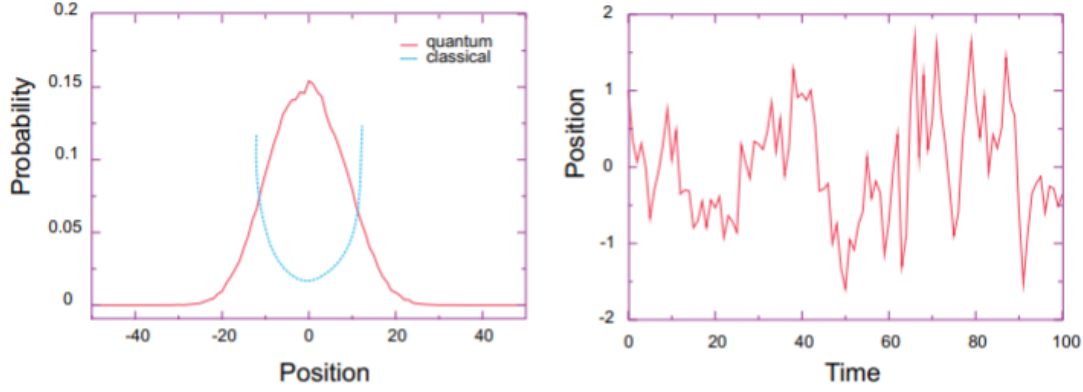
Równanie 7 jest nazywane całkowaniem po trajektoriach (path integral). Tutaj klasyczna akcja  $S$  jest sumowana wzdłuż różnych ścieżek, a wykładnik działania jest sumowany po ścieżkach.

Warto zauważyć, że działanie  $S$  jest bardzo dużą liczbą, więc nawet jeśli wszystkie trajektorie wchodzą w sumę 7, to główne wkłady pochodzą od ścieżek przylegających do klasycznej trajektorii  $\bar{x}$ .

W rzeczywistości, ponieważ  $S$  jest ekstremum dla klasycznej trajektorii, jest stałą do pierwszego rzędu w zmienności ścieżek, a więc pobliskie ścieżki mają fazy które zmieniają się płynnie i stosunkowo wolno.

Natomiast ścieżki dalekie od klasycznej trajektorii są ważone przez szybko oscylujące  $\exp [i \frac{S}{\hbar}]$ , a gdy jest wiele ścieżek, to mają tendencję do anulowania się nawzajem.

W granicy klasycznej  $\hbar \rightarrow 0$ , mamy wkład tylko pojedynczej trajektorii klasycznej, toteż równanie 7 staje się zasadą najmniejszego działania Hamiltona.



Rysunek 2: Przykłas trajektorii użytej w obliczeniach całek po trajektoriach.

### 1.2.2 Funkcja falowa stanu związanego

Przechodząc do dalszych obliczeń, założymy, że Hamiltonian  $\tilde{H}$  spełnia zależność funkcji własnych

$$\tilde{H}\psi_n = E_n\psi_n, \quad (8)$$

ponieważ Hamiltonian  $\tilde{H}$  jest Hermitowski, rozwiązania tworzą kompletny zbiór ortonormalny, gdzie ogólne rozwiązanie jest postaci

$$\psi(x, t) = \sum_{n=0}^{\infty} c_n e^{-iE_n t} \psi_n(x), \quad c_n = \int_{-\infty}^{\infty} dx \psi_n^*(x) \psi(x, t=0), \quad (9)$$

gdzie gdzie wartość współczynników rozszerzalności  $c_n$  wynika z ortonormalności  $\psi_n$ . Podstawiając  $c_n$  mamy

$$\psi(x, t) = \int_{-\infty}^{\infty} dx_0 \sum_n \psi_n^*(x_0) \psi_n(x) e^{-iE_n t} \psi(x_0, t=0) \quad (10)$$

Porównując funkcję 10 z funkcją 2, dostajemy rozwinięcie funkcji własnej dla  $G$ .

$$G(x, t; x_0, t_0 = 0) = \sum_n \psi_n^*(x_0) \psi_n(x) e^{-iE_n t} \quad (11)$$

Szukamy rozwiązania dla funkcji falowej stanu związanego, gdzie wszystkie trajektorie zaczynają się i kończą w punkcie  $x = x_0$ , z założeniem, że  $t_0 = t$ , kontynuując rozwiązanie 11 dla ujemnego czasu urojonego:

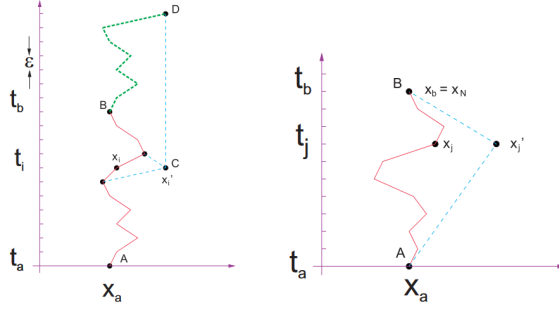
$$\begin{aligned} G(x, -i\tau; x, 0) &= \sum_n |\psi_n(x)|^2 e^{-E_n \tau} = |\psi_0|^2 e^{-E_0 \tau} + |\psi_1|^2 e^{-E_1 \tau} + \dots \\ &\Rightarrow |\psi_0(x)|^2 = \lim_{\tau \rightarrow \infty} e^{E_0 \tau} G(x, -\tau; x, 0), \end{aligned} \quad (12)$$

gdzie granica odpowiada długim czasom urojonym  $\tau$ , po których części  $\psi$  o wyższych energiach ulegają rozpadowi szybciej, pozostawiając tylko stan podstawowy  $\psi_0$ .

Tym sposobem udało nam się dojść do rozwiązania dla funkcji falowej stanu podstawowego bezpośrednio w kategoriach propagatora  $G$  w postaci jawnej (równ. 12).

### 1.3 Całkowanie po trajektoriach siatki

Na początku tworzymy siatkę punktów dyskretnych w czasoprzestrzeni i przedstawiamy trajektorie ruchu cząstki jako serie prostych linii łączących pozycje dla jednego czasu z pozycją dla następnego:



Rys. 3: Wykresy trajektorii w czasoprzestrzeni

Lewa strona: Trajektorja w siatce czasoprzestrzeni która zaczyna się i kończy na  $x = x_a = x_b$ . Działanie jest całką po tej trajektorji, podczas gdy całka trajektorji jest sumą całek po wszystkich trajektorjach. Kropkowana trajektorja BD jest przeniesioną repliką trajektorji AC. Prawa strona: Kreskowana ścieżka łączy czas początkowy z końcowym za pomocą dwóch jednakowych kroków czasowych. Czerwona jednolita trajektorja narysowana jest w  $N$  krokach, każdy z nich o rozmiarze  $\epsilon$ .  $X_j$  to pozycja trajektorji w czasie  $t_j$ .

Dzielimy trajektorje między punktami A i B na  $N$  jednakowych kroków o rozmiarze  $\epsilon$  i oznaczamy je indeksem  $j$ :

$$\epsilon \frac{t_b - t_a}{N} \Rightarrow t_j = t_a + j\epsilon \quad (j = 0, N) \quad (13)$$

Dokładniej byłoby używać  $x(t_j)$  do określenia pozycji trajektorji w czasie  $t_j$ , jednak w praktyce dyskretyzujemy przestrzeń równomiernie oraz posiadamy połączenia między najbliższymi punktami.

Kiedy już stworzyliśmy naszą siatkę, możemy rozpatrzyć całki oraz pochodne na połączeniach między punktami:

$$\frac{dx_j}{dt} \simeq \frac{x_j - x_{j-1}}{t_j - t_{j-1}} = \frac{x_j - x_{j-1}}{\epsilon}, \quad (14)$$

$$S_j \simeq L_j \Delta t \simeq \frac{1}{2} m \frac{(x_j - x_{j-1})^2}{\epsilon} - V(x_j) \epsilon \quad (15)$$

gdzie zakładamy, że Lagranżjan jest stały na każdym połączeniu.

Całki trajektorji na siatce wyznaczamy na podstawie twierdzenia o złożeniu dla propagatorów ( $G$ ):

$$G(b, a) = \int dx_j G(x_b, t_b; x_j, t_j) G(x_j, t_j; x_a, t_a) \quad (t_a < t_j, t_j < t_b) \quad (16)$$

Dla cząstki swobodnej równanie propagatora przybiera postać:

$$G(b, a) = \sqrt{\frac{m}{2\pi i(t_b - t_j)}} \sqrt{\frac{m}{2\pi i(t_j - t_a)}} \int dx_j e^{i(S[b, j] + S[j, a])}$$

$$G(b, a) = \sqrt{\frac{m}{2\pi i(t_b - t_a)}} \int dx_j e^{iS[b, a]} \quad (17)$$

Jak widać dodaliśmy do siebie działania  $S$  ponieważ całki linii łączą się na zasadzie  $S[b, j] + S[j, a] = S[b, a]$ .

Dla trajektorji o  $N$ -punktach między liniami (tak jak na rys.3) równanie (16) wygląda następująco:

$$G(b, a) = \int dx_1 \cdots dx_{N-1} e^{iS[b, a]}, \quad S[b, a] = \sum_{j=1}^N S_j, \quad (18)$$

gdzie  $S_j$  jest wartością działania dla połączenia  $j$ .

W tym momencie całka nad jedną trajektorią pokazaną na rys.3 stała się sumą N elementów, która stanie się sumą nieskończonej ilości elementów w momencie kiedy krok czasu  $\varepsilon$  dąży do zera.

Propagator (7) jest sumą wszystkich połączeń między punktami A i B. Każde połączenie jest wazone poprzez eksponente jej działania wzdłuż trajektorii:

$$G(x, t; x_0, t_0) = dx_1 dx_2 \cdots dx_{N-1} e^{iS[x, x_0]}, \quad (19)$$

$$S[x, x_0] = \sum_{j=1}^{N-1} S[x_{j+1}, x_j] \simeq \sum_{j=1}^{N-1} L(x_j, \dot{x}_j) \varepsilon \quad (20)$$

gdzie  $L(x_j, \dot{x}_j)$  jest średnią wartością Lagranżjanu na połączeniu  $j$  w momencie  $t = j\varepsilon$ .

Możemy zauważyć, że G jest określony dla ujemnego urojonego czasu w wyrażeniu (12) dla funkcji falowej stanu podstawowego. Podobnie rozważamy Lagranżjan dla  $t = -i\tau$

$$L(x, \dot{x}) = T - V(x) = +\frac{1}{2}m\left(\frac{dx}{dt}\right)^2 - V(x), \quad (21)$$

$$\Rightarrow L\left(x, \frac{dx}{-id\tau}\right) = -\frac{1}{2}m\left(\frac{dx}{d\tau}\right)^2 - V(x) \quad (22)$$

gdzie dla ułatwienia obliczeń zakładamy, że potencjał  $V(x)$  nie zależy od prędkości i innych wartości  $x$ .

Po odwróceniu znaku energii kientycznej  $L$  możemy zastosować we wzorze operator Hamiltonianu, opisujący funkcję współrzędnych uogólnionych.

$$H\left(x, \frac{dx}{-id\tau}\right) = \frac{1}{2}m\left(\frac{dx}{d\tau}\right)^2 - V(x) = E \quad (23)$$

$$\Rightarrow L\left(x, \frac{dx}{-id\tau}\right) = -H\left(x, \frac{dx}{d\tau}\right) \quad (24)$$

Przepisując całkę po trajektorii  $t$  z  $L$  jako całkę po  $\tau$  z  $H$  możemy wyrazić funkcję Greena i jej działanie w ujęciu Hamiltonianu. Otrzymujemy następujący wzór gdzie całka krzywoliniowa z  $H$  obejmują całą trajektorię.

$$G(x, -i\tau; x_0, 0) = \int dx_1 dx_2 \cdots dx_{N-1} e^{-\int_0^\tau H(\tau') d\tau'} \quad (25)$$

Całkę z  $H$  wyrażamy jako średnią energię na każdym ogniwnie, gdzie  $E_j = T_j + V_j$ , oraz sumujemy, aby uzyskać łączną energię  $\varepsilon$ :

$$\int H(\tau) d\tau = \sum_j \varepsilon E_j = \varepsilon \varepsilon(x_j) \quad (26)$$

$\varepsilon(x)$  możemy potraktować jako linie prostą, następnie używając pochodnej eulera aby użyć prędkości możemy oszacować potencjał w punkcie środkowym każdego łącza. Podstawiamy równanie (25) do naszego rozwiązania dla funkcji stanu podstawowego (12), gdzie początkowe i końcowe punkty w przestrzeni są takie same. W rezultacie otrzymujemy funkcję podziału, co wykazuje podobieństwo do termodynamiki.

Następnie przekształcamy równienie Schrodingera zależne od czasu na równanie dufyзии ciepła:

$$i \frac{\partial \psi}{\partial(-i\tau)} = \frac{-\nabla^2}{2m} \psi \Rightarrow \frac{\partial \psi}{\partial \tau} = \frac{\nabla^2}{2m} \psi \quad (27)$$

Suma po ścieżkach w kunkcji Greena może być wyrażona przez czynnik Boltzmana, która ściśle wiąże nasze rozważania z termodynamiką.

$$P = e^{-\varepsilon\varepsilon} = e^{-\varepsilon/k_B T} \Rightarrow k_B T = \frac{1}{\varepsilon} = \frac{\hbar}{\varepsilon} \quad (28)$$

W konsekwencji granica  $\varepsilon \rightarrow 0$  która sprawia, że czas jest ciągły, jest granicą „wysokiej temperatury”. Granica  $\tau \rightarrow \infty$  jest wymagana do rzutowania funkcji falowej przy rozwiązywaniu całki. W ten sposób wyraziliśmy funkcje Greena jako całkę po ścieżce, która wymaga zastosowania Hamiltonianu i sumowania po wszystkich ścieżkach.

## 2 Implementacja algorytmów w Pyhonie

Implenetację kodu zaczynamy od zdefiniowania zmiennych *path* oraz *prob*, których będziemy używać w dalszej części przy liczeniu trajektorii oraz prawdopodobieństwa stanu podstawowego.

```

7   N = 100
8   M = 101
9   xsacle = 10.
10  path = np.zeros([M], float)
11  prob = np.zeros([M], float) # Initialize

```

Rys. 4: Funkcje liczące całkę oraz prawdopodobieństwo.

Kolejnym krokiem jest dobranie osi wykresu dla trajektorii.

```

16  def trjaxes(): # axis
17      trax = curve(pos = [vector(-97,-100, 0),vector(100,-100, 0)], color = color.cyan, display = trajec)
18      label(pos = vector(0,-110, 0), text = '0', box = 0, display = trajec)
19      label(pos = vector(60, -110, 0), text = 'x', box = 0, display = trajec)
20
21  wvgraph = display(x = 340, y = 150, width = 500, height = 300, title = 'Ground State')
22  wvplot = curve(x = range(0, 100), display = wvgraph)
23  wwfax = curve(color = color.cyan)
24
25  def wvfaxes(): # axis for probability
26      wwfax = curve(pos=[vector(-600,-155, 0),vector(800,-155, 0)], display = wvgraph, color = color.cyan)
27      curve(pos = [vector(1,-150, 0),vector(0,400, 0)], display = wvgraph, color = color.cyan)
28      label(pos = vector(-80, 450, 0), text = 'Probability', box = 0, display = wvgraph)
29      label(pos = vector(600, -220, 0), text = 'x', box = 0, display = wvgraph)
30      label(pos = vector(0, -220, 0), text = '0', box = 0, display = wvgraph)
31
32  trjaxes()
33  wvfaxes() # plot axes
34

```

Rys. 5: Dobranie odpowiednich osi.

Następnym etapem jest definiowanie odpowiednich funkcji która sumuje całki po trajektorii oraz rysuje trajektorię całki za wcześniej zdefiniowanych osiach *xy* oraz stanu podstawowego.

```

34
35  def energy(path): # HO energy
36      sums = 0.
37      for i in range(0, N-2):
38          sums += (path[i+1] - path[i]) * (path[i+1] - path[i])
39          sums += path[i+1] * path[i+1];
40      return sums
41

```

Rys. 6: Funkcja licząca całkę energii.



```

42 def plotpath(path): # plot trajectory in xy scale
43     for j in range(0, N):
44         trplot.x[j] = 220 * path[j]
45         trplot.y[j] = m2 * j - 100
46
47 def plotwvf(prob):
48     for i in range(0, 100):
49         wvplot.color = color.yellow
50         wvplot.x[i] = 8 * i - 400 # convenient coordinates
51         wvplot.y[i] = 4.0 * prob[i] - 150 # for centered figure
52
53     oldE = energy(path) # find E of path

```

Rys. 7: Funkcje rysujące wykresy odpowiednio trajektorii na osi  $xy$  oraz prawdopodobieństwa stanu podstawowego.

Wreszcie dochodzimy do momentu, w którym stosujemy w kodzie algorytm "Metropolis". Algorytm ten pozwala oszacować całki i rozkłady prawdopodobieństwa dla złożonych systemów, które zazwyczaj są zbyt trudne do modelowania analitycznego, takie jak np układy wielowymiarowe. Następnie przy pomocy pętli *if* wybieramy maksymalne wartości całki oraz stosujemy funkcję, która nie dopuszcza do rysowania wartości ujemnych na wykresie.

```

55 while True: # pick random element
56     rate(10) # slows the painting
57     element = int(N * random.random()) # Metropolis algorithm
58     change = 2.0 * (random.random() - 0.5)
59     path[element] += change # change path
60     newE = energy(path); # find new E
61     if newE > oldE and math.exp(- newE + oldE) <= random.random():
62         path[element] -= change # reject
63         plotpath(path) # plot resulting last trajectory
64         elem = int(path[element] * 16 + 50) # if path = 0, elem = 50

```

Rys. 8: Zastosowanie algorytmu Metropolis.

```

70     if elem < 0:
71         elem = 0, # negative case not allowed
72     if elem > 100:
73         elem = 100 # if exceed max
74     prob[elem] += 1 # increase probability for that x value
75     plotwvf(prob) # plot prob
76     oldE = newE

```

Rys. 9: Funkcja rysująca wykres prawdopodobieństwa rosnący dla wartości  $x$ .

## 2.1 Assesment

- Przy pomocy funkcji, wykreśliłyśmy rzeczywiste ścieżki czasoprzestrzenne używane w symulacji wraz z klasyczną trajektorią.
- W celu uzyskania ciągłego obrazu funkcji falowej zmniejszamy odstępki siatki  $x$ . Następnie zmniejszając  $\epsilon$ , uzyskujemy dokładniejszą wartość funkcji falowej w dowolnych miejscach sieci.

- Szacujemy energię za pomocą poniższej funkcji, gdzie pochodna przestrzenna jest obliczana numerycznie. Fazy można zignorować, ponieważ nie pojawiają się znaki w funkcji falowej stanu podstawowego.

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

- Badamy efekt zwiększając  $\hbar$ , dzięki czemu zapobiegamy większym wahań wokół trajektorii wyjściowej. Zmniejszając wartość wykładnika stałej Boltzmana, określamy czy obliczenia stają się mniej lub bardziej niezawodne pod względem możliwości znalezienia teorii klasycznej.
- Badamy  $\psi$  w zależności od potencjału grawitacyjnego.

### 3 Podsumowanie

Przy pomocy algorytmu Metropolis nauczyliśmy się przedstawiać prawdopodobieństwo stanu podstawowego poprzez całkowanie po trajektoriach Feynmana oraz poznaliśmy ogólną teorię Feynmana. Niestety, kod nie działa w pełni poprawnie, co wynika z zastosowania straszego wzoru, który nie do końca wpisuje się w ówczesne ramy użytkowania języka *Python*.

### 4 Bibliografia

#### Literatura

"A Survey of Computational Physics" - Rubin H. Landau, José Páez and Cristian C. Bordeianu