

Modelowanie Komputerowe

Projekt - Feynman Path Integrals

Gabriela Białoskórska, Mikołaj Knysak, Ignacy Tekieli
Fizyka Techniczna

Styczeń 2021

Spis treści

1	Wprowadzenie	2
1.1	Cel pracy	2
1.2	Wstęp teoretyczny	2
1.2.1	Czasoprzestrzenna propagacja Feynmana	2
1.2.2	Stan splątany funkcji falowej	5
2	Materiały i Metody	6
2.1	Wiadomości wstępne	6
2.1.1	Metoda Monte Carlo	6
2.1.2	Algorytm Metropolisa	6
2.1.3	Biblioteki	7
2.2	Tworzenie projektu	7
3	Wyniki	10
4	Podsumowanie	11
5	Bibliografia	11

1 Wprowadzenie

1.1 Cel pracy

Klasyczna cząstka, przymocowana do sprężyny liniowej, podlega prostemu ruchowi harmonicznemu z położeniem w przestrzeni w funkcji czasu, wyrażonym przez $x(t) = A \sin(\omega_0 t + \phi)$. Celem naszego projektu było wykorzystanie wspomnianej, klasycznej trajektorii czasoprzestrzennej $x(t)$ do wygenerowania kwantowej funkcji falowej $\psi(x, t)$, dla cząstki związanej z potencjałem oscylatora harmonicznego.

1.2 Wstęp teoretyczny

1.2.1 Czasoprzestrzenna propagacja Feynmana

Feynman poszukiwał takiego sformułowania mechaniki kwantowej, które uwidaczniałoby bardziej bezpośrednio powiązanie z mechaniką klasyczną niż czyniła to Teoria Schrodingera oraz podkreślało statystyczną naturę mechaniki kwantowej. W tym celu zainspirował się on sugestią Diraca: zasada najmniejszego działania Hamiltona, którą można wykorzystać do wyprowadzenia mechaniki klasycznej, może być granicą $\hbar \rightarrow 0$ kwantowej zasady najmniejszego działania. Widząc, że zasada Hamiltona dotyczy ścieżek cząsteczek w czasoprzestrzeni, Feynman stwierdził, że kwantową funkcję falową, oposującą propagację cząstki swobodnej z punktu czasoprzestrzeni $a = (x_a, t_a)$ do punktu $b = (x_b, t_b)$, można wyrazić jako:

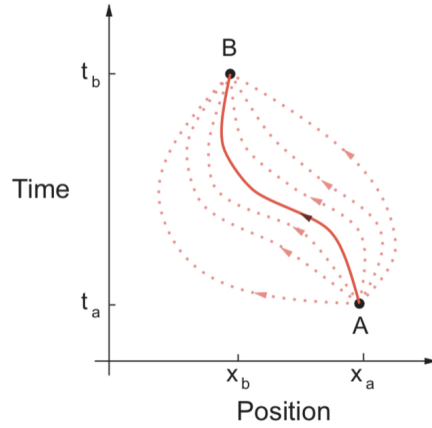
$$\psi(x_b, t_b) = \int dx_a G(x_b, t_b; x_a, t_a) \psi(x_a, t_a) \quad (1)$$

gdzie G - funkcja Greena lub *propagator*:

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

Równanie (1) jest formą zasady Huygensa, w której każdy punkt ośrodka, do którego dotarło czoło fali $\psi(x_a, t_a)$ można uważać za źródło nowej fali kulistej $G(b; a)$, która rozchodzi się do przodu w przestrzeni i czasie. Nowe czoło fali $\psi(x_b, t_b)$ powstaje poprzez sumowanie oraz interferencję ze wszystkimi innymi falami.

Feynman zauważył, że inną formą interpretacji równania (1) może być zasada Hamiltona, w której amplituda prawdopodobieństwa (funkcja falowa ψ) dla cząstki znajdującej się w punkcie B, jest równa sumie wszystkich ścieżek w czasoprzestrzeni, pochodzących z czasu A i kończących na B (Rysunek 1).



Rysunek 1: Metoda całkowa po ścieżce Feynmana w ujęciu mechaniki kwantowej; zbiór ścieżek łączy początkowy punkt A czasoprzestrzeni z końcowym punktem B. Linia ciągła to trajektoria, po której porusza się klasyczna cząstka, natomiast linie przerywane to dodatkowe, prawdopodobne ścieżki cząstek kwantowych. Klasyczna cząstka zdaje się "wiedzieć" uprzednio, że podążając po klasycznej trajektorii minimalizuje akcję S.

Pogląd ten obejmuje statystyczną naturę mechaniki kwantowej, poprzez ukazywanie różnych prawdopodobieństw podróży po innych ścieżkach. Wszystkie one są możliwe, jednak niektóre bardziej prawdopodobne. Wartości prawdopodobieństw ścieżek pochodzą z *klasycznej zasady najmniejszego działania Hamiltona*:

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

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

ze ścieżkami ograniczonymi do przechodzenia przez punkty końcowe:

$$\delta(x_a) = \delta(x_b) = 0$$

Powyższe sformułowanie mechaniki klasycznej, opierające się na rachunku wariacyjnym, jest równoważne równaniom różniczkowym Newtona (jeśli działanie S jest traktowane jako całka po linii prostej Lagrangianu wzdłuż ścieżki):

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

gdzie T - energia kinetyczna, V - energia potencjalna, $\dot{x} = \frac{\partial x}{\partial t}$, a nawiasy kwadratowe wskazują funkcjonal funkcji $x(t)$ oraz $\dot{x}(t)$

Feynman zauważył, że klasyczne działanie dla cząstki swobodnej ($V = 0$),

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

jest powiązane z propagatorem cząstki swobodnej (2) w następujący sposób:

$$G(b, a) = \sqrt{\frac{m}{2\pi i(t_b - t_a)}} e^{iS[b, a]/\hbar} \quad (6)$$

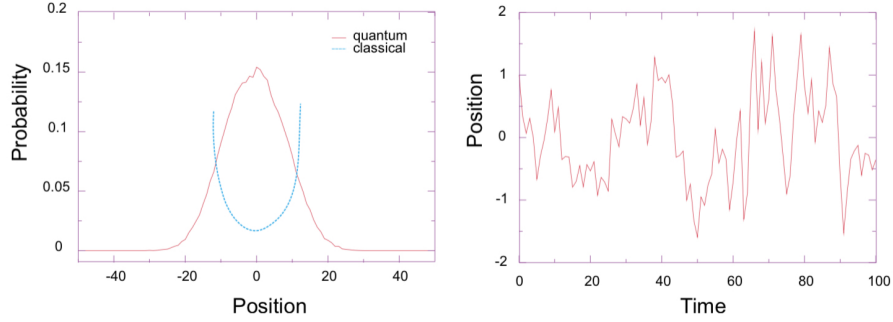
Powyższe równanie było poszukiwanym związkiem pomiędzy mechaniką kwantową a zasadą Hamiltona. W późniejszym czasie Feynman postulował przeformułowanie mechaniki kwantowej, które zawierałoby jej statystyczne aspekty, wyrażając $G(b, a)$ jako sumę ważoną wszystkich *ścieżek* łączących a z b .

$$G(b, a) = \sum_{paths} e^{iS[b, a]/\hbar} \quad (7)$$

(path integral - całka po trajektorii)

W powyższym równaniu klasyczne działanie S (4) jest rozpatrywane wzdłuż różnych ścieżek (Rysunek 1.), a jego wykładnik jest sumowany wzdłuż ścieżek. Suma (7) nazywana jest *całką po trajektorii*, ponieważ sumuje się po kacjach $S[b, a]$, z których każda jest całką. Istotnym powiązaniem pomiędzy mechaniką klasyczną i kwantową jest świadomość, że w jednostkach $\hbar \simeq 10^{-34}$ Js działanie jest bardzo dużą liczbą ($S/\hbar \geq 10^{20}$), zatem nawet jeśli wszystkie ścieżki wchodzi w skład sumy (7), główne składniki pochodzą od tych sąsiadujących z trajektorią klasyczną \bar{x} .

W rzeczywistości, ponieważ S jest ekstremum dla trajektorii klasycznej, jest także stałą pierwszego rzędu w wariacji ścieżek, w związku z czym pobliskie ścieżki posiadają fazy, które zmieniają się płynnie i stosunkowo wolno. W przeciwieństwie do nich, ścieżki dalekie od trajektorii klasycznej są ważone poprzez szybkie oscylacje $exp(iS/\hbar)$, a gdy jest ich wiele, mają one tendencję do wzajemnego znoszenia się. **W klasycznej granicy $\hbar \rightarrow 0$ tylko pojedyncza trajektoria klasyczna ma w tym swój udział i (7) staje się wówczas zasadą najmniejszego działania Hamiltona.** Rysunek 2. przedstawia przykład trajektorii stosowanej w obliczeniach całkowych.



Rysunek 2: *Po lewej*: Rozkład prawdopodobieństwa dla stanu podstawowego oscylatora harmonicznego, określony za pomocą obliczenia całki po trajektorii (klasyczny wynik ma maksima w dwóch punktach zwrotnych). *Po prawej*: Ścieżka kwantowa czasoprzestrzeni, wynikająca z zastosowania algorytmu Metropolis

1.2.2 Stan splątany funkcji falowej

Do naszych obliczeń musimy zastosować jeszcze jedno wyrażenie funkcji Gre-ena, które przybliżymy w poniższym podrozdziale. Załóżmy, że Hamiltonian \hat{H} obsługuje spektrum funkcji własnych

$$\hat{H}\psi_n = E_n\psi_n$$

każda oznaczona indeksem n . Ponieważ H jest Hermitowskie, rozwiązania tworzą kompletny zbiór ortonormalny, w którym rozwiązanie ogólne:

$$\psi(x, t) = \sum_{n=0}^{\infty} c_n e^{-iE_n t} \psi_n(x), \quad c_n = \int_{-\infty}^{+\infty} dx \psi_n^*(x) \psi(x, t=0) \quad (8)$$

gdzie wartość współczynników rozszerzalności c_n wynika z ortonormalności ψ_n . Jeśli podstawimy wspomniane c_n z powrotem do rozwinięcia funkcji falowej (8), otrzymamy tożsamość:

$$\psi(x, t) = \int_{-\infty}^{+\infty} dx_0 \sum_n \psi_n^*(x_0) \psi_n(x) e^{-iE_n t} \psi(x_0, t=0) \quad (9)$$

Porównując powyższe równanie z (11), otrzymujemy rozszerzenie funkcji własnej dla G :

$$G(x, t; x_0, t_0 = 0) = \sum_n \psi_n^*(x_0) \psi_n(x) e^{-iE_n t} \quad (10)$$

Odnosi się to do funkcji falowej stanu splątanego poprzez wymóg, aby wszystkie ścieżki zaczynały się i kończyły w pozycji przestrzennej $x_0 = x$, przyjmując

$t_0 = 0$ oraz przez dokonanie analitycznej kontynuacji (10) aż do ujemnego czasu urojonego (dopuszczalnego dla funkcji analitycznych):

$$G(x, -i\tau; x, 0) = \sum_n |\psi_n(x)|^2 e^{-E_n \tau} = |\psi_0|^2 e^{-E_0 \tau} + |\psi_1|^2 e^{-E_1 \tau} + \dots \quad (11)$$

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

Granica odpowiada w tym przypadku długim, urojonym czasom τ , po których elementy ψ o wyższych energiach rozpadają się szybciej, pozostawiając tylko stan podstawowy ψ_0 .

Równanie (11) zapewnia rozwiązanie funkcji falowej stanu splątanego w postaci zamkniętej, bezpośrednio w kategoriach propagatora G . Już wcześniej (rozdział 1.2.1) ukazaliśmy niektóre wyniki obliczeń (Rysunek 2.). Choć zaczęliśmy od rozkładu prawdopodobieństwa, który osiąga szczyty w pobliżu klasycznych punktów zwrotnych na krawędziach studni, jednak po dostatecznie dużej liczbie iteracji otrzymaliśmy rozkład przypominający oczekiwany Gaussian. Po prawej stronie widoczna jest trajektoria, która została wygenerowana poprzez statystyczną zmienność klasycznej trajektorii $x(t) = A \sin(\omega_0 t + \phi)$

2 Materiały i Metody

2.1 Wiadomości wstępne

2.1.1 Metoda Monte Carlo

Projekt wykonaliśmy zgodnie z **Metodą Monte Carlo**. Jest ona stosowana do matematycznego modelowania procesów zbyt złożonych (obliczania całek, łańcuchów procesów statystycznych) aby można było przewidzieć ich wyniki za pomocą klasycznego podejścia analitycznego. Istotną rolę odgrywa w niej losowanie (wybór przypadkowy) wielkości charakteryzujących proces, przy czym losowanie dokonywane jest zgodnie z rozkładem, który musi być znany.

2.1.2 Algorytm Metropolisa

Dokładnie **Algorytm Metropolisa-Hastingsa** - metoda statystyczna typu MCMC (próbkiowania Monte Carlo łańcuchami Markowa), pozwalająca stochastycznie oszacowywać całki (takie jak estymatory) i rozkłady prawdopodobieństwa dla złożonych systemów, które są zbyt trudne do modelowania analitycznego, np. układów wielowymiarowych. Narzędzie to jest wykorzystywane między innymi we wnioskowaniu bayesowskim i modelowaniu systemów fizycznych. Procedurę tę opisał po raz pierwszy publicznie zespół Metropolisa w 1953 r., a rozwinął ją Hastings w 1970 r.

Systemy wielowymiarowe obarczone są zjawiskiem przekleństwa wymiarowości, polegającego na tym, że wraz ze wzrostem liczby wymiarów problemu, liczba obserwacji potrzebnych do oszacowania ich cech wykładniczo rośnie. Metody Monte Carlo wykorzystujące łańcuchy Markowa są w dużej mierze odporne na ten problem, ponieważ nie wymagają rozwiązania analitycznego, ani nie przeszukują całej przestrzeni problemu, lecz posługują się iterowanym próbkowaniem stochastycznym, które z każdą kolejną iteracją coraz dokładniej skupiają się na centralnych obszarach rozkładów.

Wprowadzenie metod MCMC łącznie z rosnącą dostępnością komputerów umożliwiło przełamanie wielu ograniczeń stojących przed analizą złożonych problemów w naukach empirycznych, opartych wcześniej głównie o uproszczone metody ortodoksyjnej statystyki.

2.1.3 Biblioteki

Kod umieszczony w książce, na podstawie której powstawał niniejszy projekt ("*Computational Physics: Problem Solving with Computers*"), oparty był na bibliotece `visual`. Obecnie nie jest już ona obsługiwana w nowszych wersjach języka Python, wobec czego podjęliśmy próby uruchomienia kodu na stronie `glowscript.org`. Nie przyniosło to jednak efektu, gdyż pojawił się error: *ReferenceError: Can't find variable: zeros*. Wobec powyższych przeszkód zmuszeni byliśmy całkowicie zmienić strukturę programu zamieszczonego w literaturze w taki sposób, aby współgrała ona z inną biblioteką. W tym celu posłużyliśmy się biblioteką `Matplotlib`.

`Matplotlib` to biblioteka służąca do tworzenia wykresów dla języka programowania Python i jego rozszerzenia numerycznego NumPy. Zawiera ona API "pylab" zaprojektowane w taki sposób aby było jak najbardziej zbliżone do MATLABa, przez co jest przystępna do nauki i codziennego użytku. `Matplotlib` został napisany i utrzymany głównie przez Johna Huntera. Jest dostępny na licencji przypominającej licencję BSD.

2.2 Tworzenie projektu

Zgodnie z przytoczonymi powyżej wiadomościami wstępnymi, kod w naszym projekcie powstał na podstawie programu `QMC.py`, zamieszczonego w książce. Określa on prawdopodobieństwo stanu podstawowego poprzez integrację ścieżki Feynmana, przy użyciu algorytmu Metropolis do symulacji wariacji na temat klasycznej trajektorii. Na jego podstawie napisaliśmy podobny kod, jednakże przy wykorzystaniu biblioteki `Matplotlib`. Dokładny opis działania programu krok-po-kroku znajduje się w przytoczonej literaturze (rozdział 15.8.2). Poniżej załączamy naszą interpretację algorytmu.

```

# QMC.py: Quantum MonteCarlo , Feynman path integration
import math
import matplotlib.pyplot as plt
import numpy as np
import random as rand

from drawnow import drawnow

N, M = 100, 101
xscale = 10
path, prob = np.zeros([M], float), np.zeros([M], float)
path_x, path_y = [0 for _ in range(N)], [0 for _ in range(N)]
prob_x, prob_y = [0 for _ in range(N)], [0 for _ in range(N)]

fig_tr, ax_tr = plt.subplots()
ax_tr.set_title("Spacetime Trajectories")
ax_tr.set_xlabel("X")
ax_tr.set_ylabel("O")

fig_wv, ax_wv = plt.subplots()
ax_wv.set_title("Probability")
ax_wv.set_xlabel("X")
ax_wv.set_ylabel("Probability")

def refresh_tr():
    ax_tr.plot(path_x, path_y)

def refresh_wv():
    ax_wv.plot(prob_x, prob_y)

def plotpath(path):
    for i in range(N):
        path_x[i] = 20 * path[i]
        path_y[i] = 2 * i - 100

def plotwvf(prob):
    for i in range(100):
        prob_x[i] = 8 * i - 400
        prob_y[i] = 4 * prob[i] - 150

def energy(path):
    sum_ = 0
    for i in range(N - 2):
        sum_ += (path[i + 1] - path[i]) * (path[i + 1] - path[i])
        sum_ += path[i + 1] ** 2

```



```

    return sum_

def main():
    old_E = energy(path)

    while True:
        element = int(N * rand.random())
        change = 2 * (rand.random() - .5)
        path[element] += change
        new_E = energy(path)

        if new_E > old_E and math.exp(-new_E + old_E) <= rand.random():
            path[element] -= change
            plotpath(path)

        elem = int(path[element] * 16 + 50)

        if elem < 0:
            elem = 0
        elif elem > 100:
            elem = 100

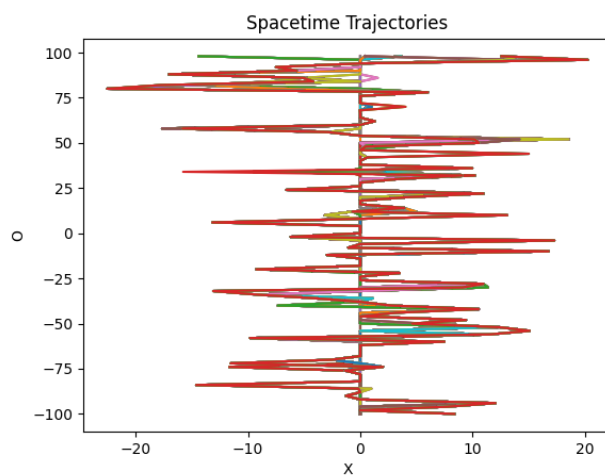
        prob[elem] += 1
        plotwvf(prob)
        old_E = new_E
        drawnow(refresh_tr)
        drawnow(refresh_wv)

if __name__ == '__main__':
    main()

```

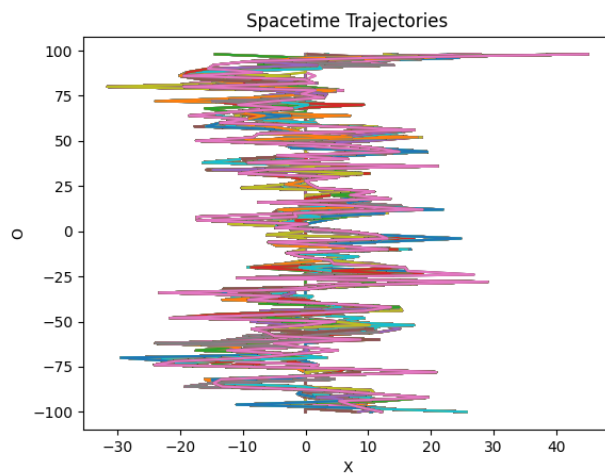
3 Wyniki

Otrzymany przez nas wykres wygląda następująco:



Rysunek 3: Trajektorie w czasoprzestrzeni po uruchomieniu programu

Jego wygląd zmienia się w czasie, stąd po upływie krótkiej chwili otrzymujemy:



Rysunek 4: Trajektorie w czasoprzestrzeni po chwili od uruchomienia programu

4 Podsumowanie

Podsumowując, w niniejszym projekcie przedstawiliśmy funkcję Greena jako całkę po trajektorii, która wymaga integracji Hamiltonianu wzdłuż ścieżek i sumowania po wszystkich ścieżkach, zgodnie z równaniem:

$$\begin{aligned} \lim_{\tau \rightarrow \infty} \frac{G(x, -i\tau, x_0 = x, 0)}{\int dx G(x, -i\tau, x_0 = x, 0)} &= \frac{\int dx_1 \dots dx_{N-1} \exp[-\int_0^\tau H d\tau']}{\int dx dx_1 \dots dx_{N-1} \exp[-\int_0^\tau H d\tau']} \\ \implies |\psi_0(x)|^2 &= \frac{1}{Z} \lim_{\tau \rightarrow \infty} \int dx_1 \dots dx_{N-1} e^{-\epsilon \mathcal{E}} \\ Z &= \lim_{\tau \rightarrow \infty} \int dx dx_1 \dots dx_{N-1} e^{-\epsilon \mathcal{E}} \end{aligned} \quad (12)$$

Powyższą całkę po trajektorii rozważamy jako sumę wszystkich trajektorii w sieci czasoprzestrzennej. Każda próbna ścieżka występuje z prawdopodobieństwem opartym na jej działaniu; używamy algorytmu Metropolis do uwzględnienia fluktuacji statystycznych w połączeniach, tak jakby znajdowały się one w równowadze termicznej. Im więcej iteracji wykona algorytm, tym więcej czasu posiada otrzymana funkcja falowa aby zrównoważyć się ze stanem podstawowym. Otrzymane przez nas wyniki są zatem zgodne z założeniami projektu.

5 Bibliografia

Literatura wykorzystana do napisania programu oraz wstępu:

- *"Computational Physics: Problem Solving with Computers"* - Rubin H. Landau.

Materiały wykorzystane w rozdziale 2.1:

- https://pl.wikipedia.org/wiki/Metoda_Monte_Carlo
- https://pl.wikipedia.org/wiki/Algorytm_Metropolis-Hastingsa

Wykorzystana biblioteka

- <https://matplotlib.org>

Strona umożliwiająca pisanie kodu w bibliotece visual

- www.glowscript.org