FO Projekt: Brownian Motion oraz Levy Flight Symulatory

Tomasz Pawlak, Wojciech Grunwald

26 czerwca 2023

1 Wprowadzenie

Symulatory *Ruchów Browna* oraz *Lotu Levy'ego* są stworzone do wyświetlania dynamicznej trajektorii ruchu cząstki wykorzystując proces stochastyczny modelujący ruch w różnych środowiskach

2 Algorytm symulacji

Algorytmy symulacji składają się z następujących etapów:

2.1 Brownian Motion

Algorytm Brownian Motion zastosowany w projekcie składa się z następujących etapów:

1. Inicjalizacja pozycji cząstek na mapie 700x700 px.
2. Generacja liczb losowych determinujących kierunki cząstek (kąt jako współrzędna biegunowa).
3. Odświeżanie pozycji cząstki w iteracji liczonej poprzez estymacje ruchem jednostajnym oraz obsługa odbić cząstek od ścian oraz od cząstek.
4. Rysowanie trajektorii między ostatnimi pozycjami wybranej cząstki.
5. Powtórzenie kroków 2, 3 i 4 w kolejnych iteracjach

2.1.1 Wartości obliczane w iteracjach

1. Wall Momentum P - średnia ostatnich 32 iteracji wartości ciśnienia wywieranego na ściany Kanwy. Pojedyńcze ciśnienie wywierane obliczane jest ze wzoru

$$P = \frac{2N}{3S} \cdot \left(\frac{m}{2\pi k_B T} \right)^{\frac{3}{2}}, \tag{1}$$

gdzie:

- P to ciśnienie,
- S to pole powierzchni naczynia,

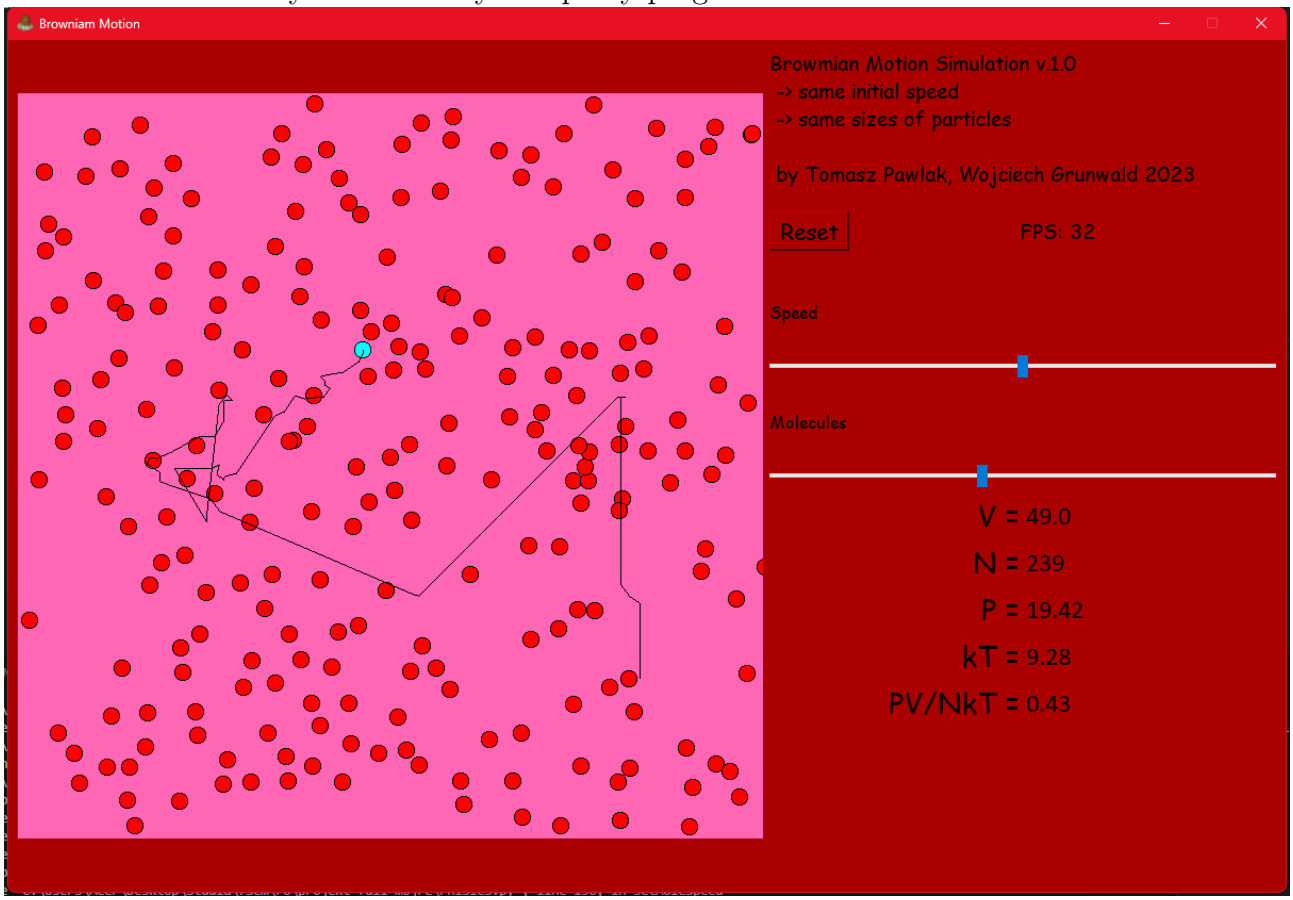
- m to masa cząstki,
- k_B to stała Boltzmanna,
- T to temperatura,
- N to liczba cząstek w naczyniu.

2. Energia kT - energia cząstki liczona ze wzoru

$$kT = mV^2, \tag{2}$$

2.1.2 Manipulacja symulacją

Rysunek 1: Przykład pracy programu Brownian Motion



Programem możemy manipulować parametrami **Speed** oraz **Molecules**. Pierwszy odpowiada za nadpisanie wartości składowych ***Vx, Vy*** każdej z cząstek wartością ułamka:

$$\frac{Vnowe}{Vstare} \cdot Vmole \tag{3}$$

Co jest wymuszone przez fizykę odbić, która powoduje, że zadana prędkość nie jest stała, lecz zależy od fizyki odbić zgodnie z regułami odbić sprężystych.
Ustawianie parametru **Molecules** powoduje generację nowych molekuł na planszy, przy okazji wykluczając nakładanie się nowych cząstek na istniejące.

2.1.3 Obsługa odbić Molekuła-Ściana

Fizyka odbicia od ścian jest realizowana poprzez algorytm:

1. Wykryj przekroczenie granic planszy i ustal, w której osi następuje przekroczenie.
2. Oblicz drogę do krawędzi mapy.
3. Pomniejsz drogę do przebycia o odległość do krawędzi.
4. Pomnóż wartość prękości Vx lub Vy przez -1 zgodnie z przypadkiem odbicia.
5. Oblicz odpowiednią zmianę kąta w układzie kartezjańskim poprzez:

$$\phi := \pi - \phi \tag{4}$$

lub

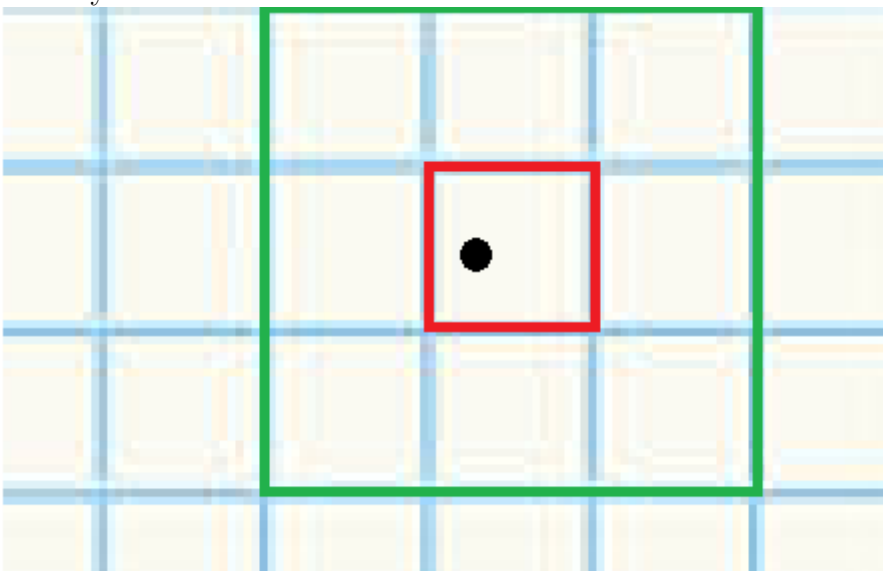
$$\phi := 2\pi - \phi \tag{5}$$

6. Oblicz nowe położenie na nowej (krótszej) drodze obliczonej w pkt. 3

2.1.4 Obsługa odbić Molekuła-Molekuła

Otoczeniem cząstki *Mole* nazywamy obszar zdefiniowany przez stałe **GRID_WIDTH** oraz **GRID_HEIGHT** i jest kwadratowym obszarem obliczanym jako prostokątny podobszar Kanwy porównywalny do kartki w zeszycie.
Domyślnie w programie obszar "kratki" jest wymiaru 70x70px, czyli jest w skali 1/10 w stosunku do Kanwy.
Cząstki $Mole_A$ oraz $Mole_B$ są w otoczeniu wtedy i tylko wtedy, gdy wartość bezwzględna różnicy indeksów w obu osiach siatki ma wartość 0 lub 1:

Rysunek 2: Otoczenie molekuły. Kolorem czerwonym zaznaczono pole siatki, w której znajduje się molekuła, a zielonym otoczenie.



Realizacja odbić Molekuła-Molekuła była najtrudniejszym elementem projektu ze względu na złożoność matematyczną. Jest ona realizowana przez algorytm:

Jeśli $Mole_A$ występuje w otoczeniu $Mole_B$:

1. Oblicz odległość cząstek w danej iteracji oraz składowe n_x oraz n_y tej odległości.
2. Oblicz przeskalowane wektory nachylenia składowych prędkości tan_1, tan_2 dla obu cząstek.
3. Oblicz przeskalowane wartości wektorów normalnych $norm_1$, $norm_2$.
4. Oblicz moment bezwładności cząstek na podstawie pkt. 2, 3 i masy cząstek (zderzenie sprężyste).
5. Oblicz i uaktualnij wartości prękości i położenia cząstek.

2.2 Levy Flight

Algorytm Levy Flight zastosowany w projekcie składa się z następujących etapów:

2.2.1 Teoria

Levy flight jest ruchem typu *random walk*, w których długości kroków mają rozkład stabilny (tym się różni od *Levy walk*), czyli rozkład prawdopodobieństwa z długim ogonem (czyli nieograniczonym eksponencjalnie). W naszym przypadku (2D) kroki następują w przypadkowych izotropowych kierunkach.

Lot Lévy'ego prowadzi do dyfuzji z anomaliami, co oznacza, że średniokwadratowe przemieszczenie od punktu początkowego (MSD) rośnie szybciej niż liniowo z czasem t, podczas gdy ruchy Browna wykazują dyfuzją normalną, w której wzrost jest liniowy.

Algorytm Levy Flight zaimplementowany w projekcie składa się z następujących etapów:

1. Inicjalizacja pozycji startu cząstki w środku masy ekranu (350,350).
2. Generacja liczb losowych zgodnych z rozkładem Levy'ego determinujących kolejną iterację.
3. Odświeżenie pozycji cząstki oraz rysowanie trajektorii między ostatnimi pozycjami cząstki.
4. Powtórzenie kroków 2 i 3 w kolejnych iteracjach.

2.2.2 Wartości obliczane w iteracjach

Levy Flight jest stochastycznym procesem, który można opisać za pomocą równania:

$$\text{step} = \text{stepSize} \cdot \left(\text{randn}(2) \cdot |\text{normal}(0,1)|^{-1/\alpha}\right) \tag{6}$$

gdzie:

- *stepSize* - parametr kontroluje skalę skoków określając, jak daleko cząstka może się przemieścić w każdym kroku iteracji
- α - wskaźnik, który kontroluje rozkład skoków cząstek w symulacji. Określa on charakter skoków, czyli jak często występują skoki o dużej amplitudzie w porównaniu do skoków o małej amplitudzie.

2.2.3 Manipulacja symulacją

Programem możemy manipulować parametrami **stepSize** oraz α.

Rysunek 3: Przykład pracy programu Levy Flight

3 Implementacja

Programy zostały zaimplementowane w języku Python wraz z wykorzystaniem poniższych bibliotek:

- NumPy: Generowanie liczb losowych oraz przeprowadzenie kalkulacji matematycznych.
- Qt: Interfejs graficzny, obsługa asynchronicznych sygnałów sterowania symulacją

3.1 Skład repozytorium

- **Drawer.py** - klasa służąca do inicjowania wizualizacji Kanwy, czyszczenia jej z iteracji na iterację oraz generowania kształtów takich jak linie oraz koła.
- **Phisics.py** - klasa Physics służąca do definiowania środowiska w aplikacji Brownian Motion zawierająca pola takie jak prędkość globalna, prędkości składowe, ciśnienie, volumin, energia układu. Tu znajdują się metody odpowiedzialne za obsługę zmiany liczby cząstek, zmianę prędkości globalnej cząstek, wykonywanie obliczeń celem ustalenia położenia, obliczania współczynnika klatek na sekundę symulacji. Pola klasy Mole odpowiadają za obliczenia dla cząstki elementarnej. Klasa zawiera metodę obliczająca trajektorię każdej z cząstek oraz przy okazji tego obliczania parametrów równania Clapeyrona dla symulacji. Najważniejszą metodą tej klasy jest bounce służące do obsługi odbić Molekuła-Molekuła.
- **Brownian.py** - jest obsługą okna aplikacji Brownian, łączenia asynchronicznych sygnałów modulacji pasków, obsługi resetu symulacji oraz wykonywania nanoszenia wartości obliczonych w Physics na Kanwę

- **LevyPhisics.py** - odpowiednik klasy Physics w aplikacji Levy Flight. Realizuje algorytm opisany w punkcie 2.2. oraz przekazuje wyniki do klasy z pliku LevyApp.py
- **LevyApp.py** - realizuje obsługę komponentów Qt, rysowanie oraz dokonuje interpolacji punktów ścieżki algorytmu LevyTrace, aby zapobiec rysowaniu się trajektorii poza ekranem. W odróżnieniu od Brownian Motion tutaj nie uwzględniamy "ścian" Kanwy, zatem ten zabieg jest konieczny celem zapewnienia czytelności, co widać na Rysunku 3.

4 Uruchamianie

Aby uruchomić programy należy wykonać poniższe kroki:

1. Instalacja Pythona w wersji 3.11.3 i wymaganych bibliotek (PyQtCore, NumPy).
2. Otwarcie terminala oraz nawigacja do głównego folderu projektu.
3. Uruchomienie programu komendą: `python ./Brownian.py` lub `python ./LevyApp.py`
4. Aplikacje uruchamiają się samoczynnie, możemy modyfikować parametry "na żywo" oraz resetować symulację
5. Obserwujemy generację wizualną programu w stylistycznym, przyjemnym dla oka, czerwonym motywie

5 Bugfix w wersji 1.1

1. Naprawiono błędy występujące podczas szybkiego zmieniania wartości Speed paskiem wartości.
2. Dodano marginesy Lotu Levy'ego.
3. Przesunięto zakres Mole Count z 10-100 do 50-500.