

# Symulacja dynamiki gazu

## Metody Monte Carlo w Fizyce

Julia Potempa (411073)



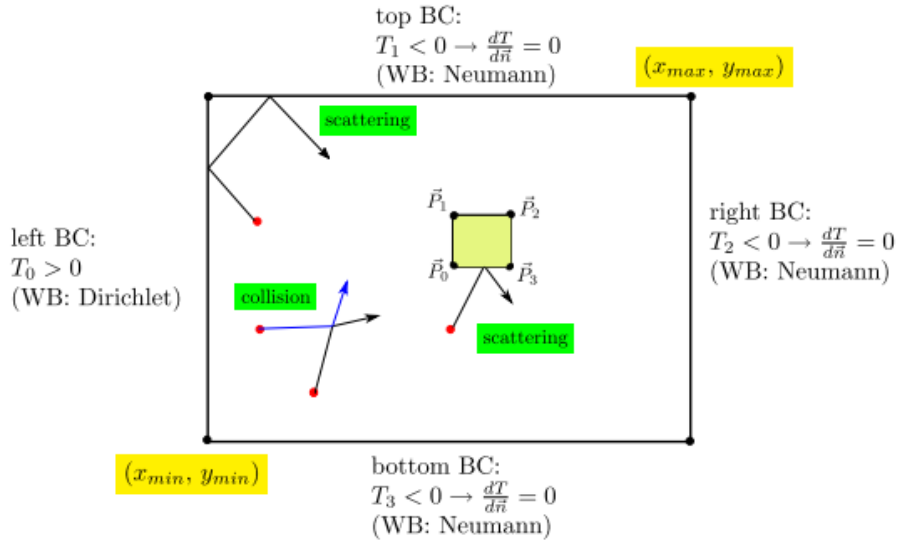
Akademia Górniczo-Hutnicza  
im. Stanisława Staszica w Krakowie

11. czerwca 2025r.

# Cel ćwiczenia

Zasymulowanie dynamiki gazu metodą bezpośrednią Monte Carlo (z ang. *Direct Simulation Monte Carlo* DSMC).

## 1 Wstęp



Rysunek 1: Rysunek przedstawiający układ, w którym symulowane są cząsteczki gazu.

Na rysunku 1 przedstawiony jest układ symulowany w tym zadaniu. Standardowo symuluje się liczbę cząstek rzędu  $10^5 - 10^7$ , więc aby przyspieszyć działanie programu symuluje się zderzenia w lokalnym otoczeniu. To znaczy dzieli się obszar na mniejsze komórki  $\Delta x \times \Delta y$  indeksowane  $(i, j)$  i sprawdza zderzenia w nich i 8 w ich najbliższym otoczeniu. Cząstki, które znajdują się blisko brzegu w przedziale czasu  $(t, t + \Delta t)$  mogą przejść do sąsiednich komórek, gdzie dalej się zderzają.

Najważniejsze elementy algorytmu DSMC:

- **Warunek początkowy**

Na początku tej symulacji definiujemy stan nieustalony układu ze względu na jego charakter transportowy. Chcemy doprowadzić układ do równowagi termodynamicznej, czyli aby rozkład prędkości był opisany rozkładem Maxwella (z uwzględnieniem braku wymiany ciepła z otoczeniem - układ izolowany). Naszym punktem wyjścia jest rozkład Boltzmanna:

$$f_E = \left( \frac{m}{2\pi k_B T} \right)^{\frac{d}{2}} e^{-\frac{E_{\text{kin}}}{k_B T}} \quad d = 1, 2, 3 \text{ (liczba wymiarów)}$$

z warunkiem unormowania

$$\int_0^\infty f_E(E) dE = 1$$

Energia kinetyczna może być wyrażona wzorem:  $E_{\text{kin}} = m(v_x^2 + v_y^2)\frac{1}{2}$ , więc:

$$f_E(E) = \underbrace{\left(\frac{m}{2\pi k_B T}\right)^{\frac{1}{2}} e^{-\frac{mv_x^2}{2k_B T}}}_{f_{E_x}} \cdot \underbrace{\left(\frac{m}{2\pi k_B T}\right)^{\frac{1}{2}} e^{-\frac{mv_y^2}{2k_B T}}}_{f_{E_y}} = f_{E_x} \cdot f_{E_y}$$

a składowe prędkości można losować z rozkładu normalnego:

$$v_x, v_y \sim \sigma_v \cdot N(0, 1) \quad \sigma_v = \sqrt{\frac{k_B T}{m}}$$

Aby uzyskać rozkład Maxwella, czyli taki, który zależy wyłącznie od wartości prędkości, to musimy dokonać transformacji zmiennych na cylindryczne:

$$f_E(E)dE = f_1(v_x, v_y)dv_x dv_y = 2\pi f_2(v)dv = f_v^{2D} dv$$

co daje nam rozkład

$$f_v^{2D} = \frac{mv}{k_B T} e^{-\frac{mv^2}{2k_B T}}, \quad v = \sqrt{v_x^2 + v_y^2}$$

z warunkiem normalizacji

$$\int_0^\infty f_v^{2D} dv = 1$$

### • Krok czasowy

Zakładamy, że w naszej symulacji w czasie  $\Delta t$  cząstka nie może przemieścić się o więcej niż szerokość lub wysokość komórki definiującej jej lokalne otoczenie

$$\Delta t(t) \leq \frac{\min\{\Delta x, \Delta y\}}{v_{\max}(t)}, \quad v_{\max} = \max\{v_1, v_2, \dots, v_{n_{\text{tot}}}\}$$

Cząstki w wyniku zderzenia zmieniają prędkość, więc  $v_{\max}(t)$  i  $\Delta t(t)$  trzeba wyznaczać w każdym kroku.

### • Zderzenia

W pełnej wersji algorytmu DSMC cząstki są rozpraszane w komórkach w sposób losowy - na podstawie ich średniej prędkości kwadratowej oraz przekrojów czynnych na rozpraszanie określa się ile par cząstek ma się rozproszyć, po czym rozprasza się je w układzie środka masy w losowych kierunkach. Zderzenia muszą przebiegać w taki sposób, aby pęd środka masy był zachowany.

W naszej wersji DSMC zderzenia są wykrywane, a rozpraszają się tylko te cząstki, których trajektorie się przecinają. Kierunki zderzeń są losowane w układzie środka masy. Jest to mniej wydajne podejście, ale zwiększa dokładność i rozdzielczość przestrzenną metody.

## 2 Metodyka

Kod symulujący gaz został przygotowany w języku C++ i wykorzystana została do tego klasa DSMC\_2D udostępniona na potrzeby zajęć. Aby uruchomić symulację należało najpierw przygotować plik wejściowy z parametrami początkowymi.

```
1.0 2.0 0.0 1.0 //xmin, xmax, ymin, ymax
50 50 //nx, ny
1.0E-23 //stala Boltzmanna kB
300. -1. -1. -1. -1. // temperatura i warunki brzegowe na odpowiednio lewym, gornym,
    prawym i dolnym brzegu
2 //init_dist, czyli rozkład poczatkowy gazu
1 //n_mix - liczba typow czastek
200000 40.0E-27 1.0E-6 //liczba czastek, masa czastki, promien czastki
4 //wierzcholki bariery wewnatrz
0.2 0.2
0.2 0.4
0.4 0.4
0.4 0.2 //brzegi tej bariery
```

Program wykorzystał funkcjonalność klasy wykorzystującą OpenMP, która pozwala na prowadzenie obliczeń wielordzeniowo. W każdej z symulacji zostało wykorzystane 7 wątków.

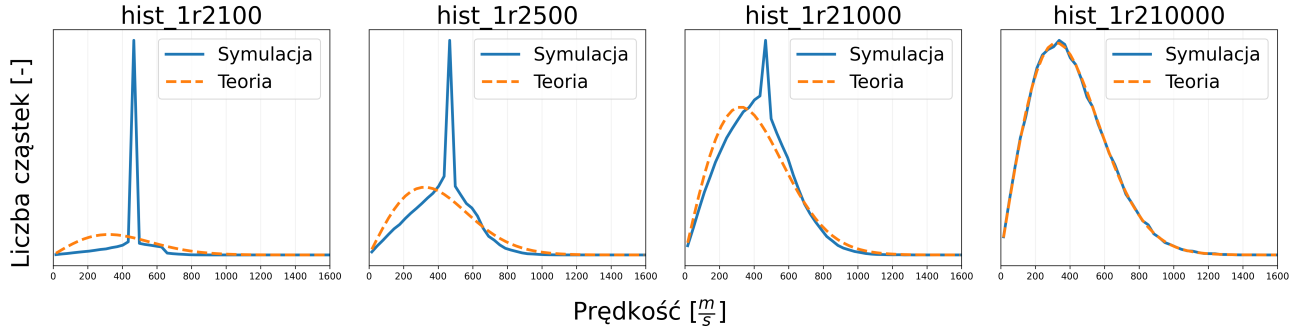
Przyjęte parametry poszczególnych symulacji:

0. Parametry podstawowe dla każdej symulacji:  $x_{min} = 0$ ,  $y_{min} = 0$ ,  $y_{max} = 1$ ,  $x_{max} = 1$ ,  $n_x = n_y = 50$ ,  $k_B = 1 \cdot 10^{-23}$ ,  $temp = 300$ , warunki brzegowe von Neumanna, liczba cząstek  $n_{mix} = 1$ ,  $n_1 = 10^5$ , masa cząsteczki  $m_1 = 40 \cdot 10^{-27}$ kg, promień cząstki  $r_1 = 10^{-6}$ m, **nodes**= 0.
1. **init\_dist**= 1 - cząsteczki mają identyczne energie kinetyczne, dwa różne promienie  $r_1 = 10^{-6}$ m lub  $r_1 = 10^{-5}$ m.
2. **init\_dist**= 3 - wszystkie cząstki umieszczone początkowo w jednej komórce.
3. **init\_dist**= 2 - rozkład Maxwella w całym obszarze, ustawiamy na lewym brzegu temperaturę 1000K.
4. Tak samo jak w 3., ale ustawiamy na prawym brzegu temperaturę 300K.

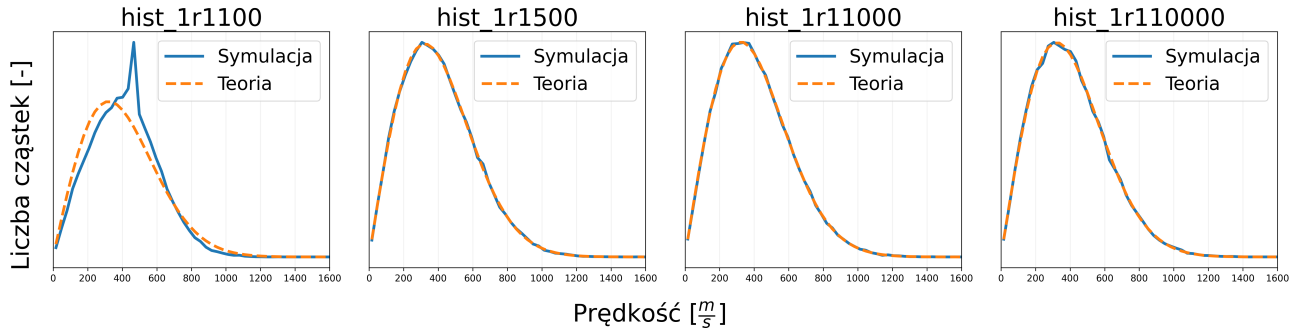
Wykresy otrzymanych wyników zostały przygotowane w języku Python.

## 3 Wyniki

### 3.1 Pierwszy zestaw parametrów



Rysunek 2: Wykresy prezentujące porównanie rozkładu prędkości w kolejnych krokach symulacji z teoretycznym rozkładem Maxwella dla promienia  $r = 10^{-5}\text{m}$ .

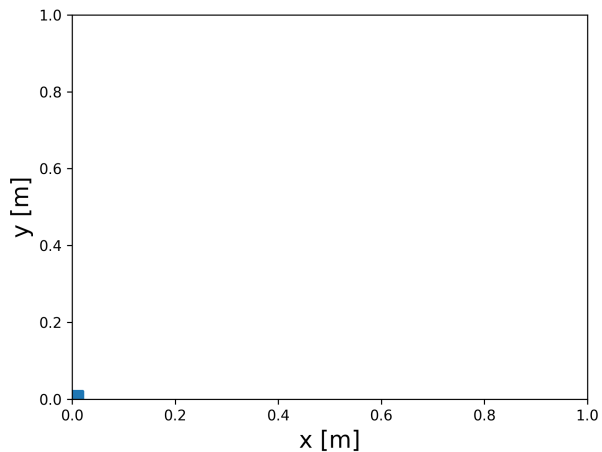


Rysunek 3: Wykresy prezentujące porównanie rozkładu prędkości w kolejnych krokach symulacji z teoretycznym rozkładem Maxwella dla promienia  $r = 10^{-6}\text{m}$ .

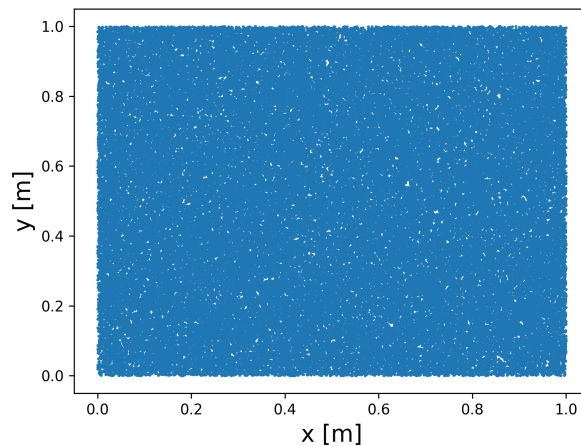
Rozkład prędkości dla cząsteczek o promieniu  $r = 10^{-5}$  dużo wolniej osiągnęły kształt rozkładu zgodny z teoretycznym. Dla początkowych iteracji widzimy wyraźną przewagę cząsteczek o prędkości około  $500 \frac{\text{m}}{\text{s}}$ .

### 3.2 Drugi zestaw parametrów

W drugim zestawie widać, że zgodnie z założeniami wszystkie cząstki gazu początkowo znajdują się w małej komórce w początku układu współrzędnych. Na końcu symulacji cząstki rozpraszają się po całej dostępnej przestrzeni.

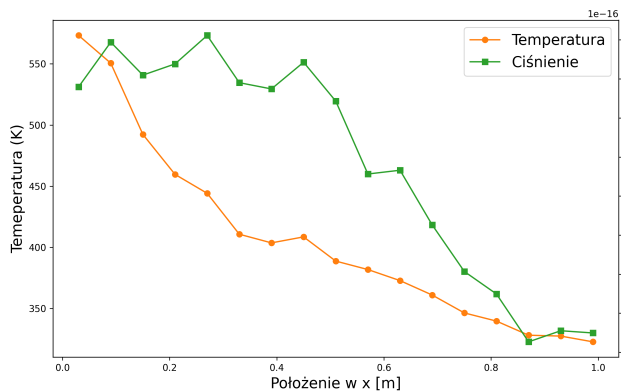


(a) Umiejscowienie cząstek gazu na początku symulacji.

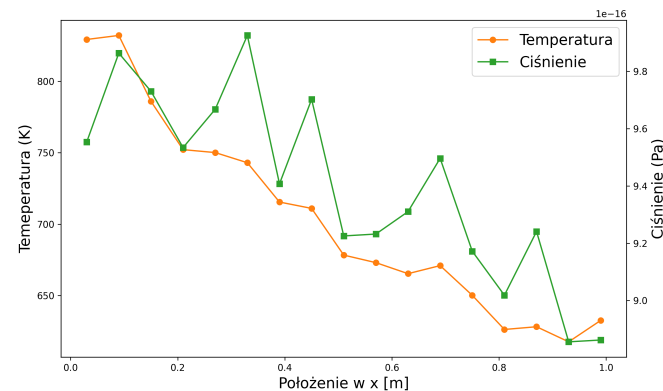


(b) Umiejscowienie cząstek gazu na końcu symulacji.

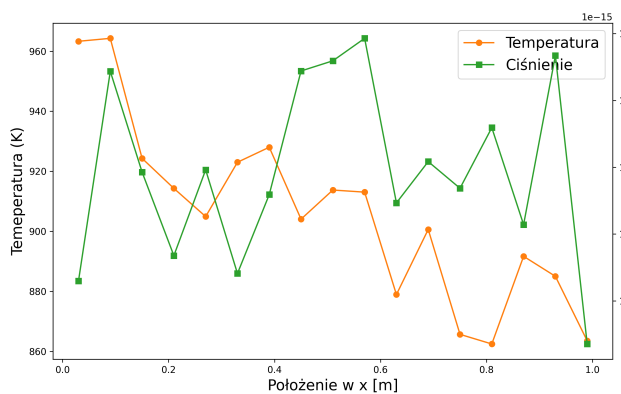
### 3.3 Trzeci zestaw parametrów



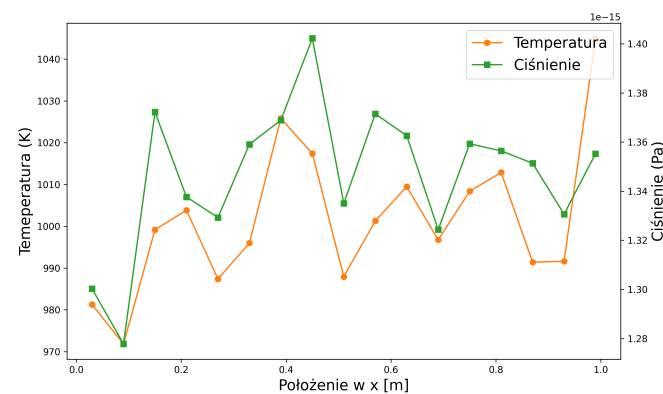
(a) it = 100



(b) it = 500



(c) it = 1000

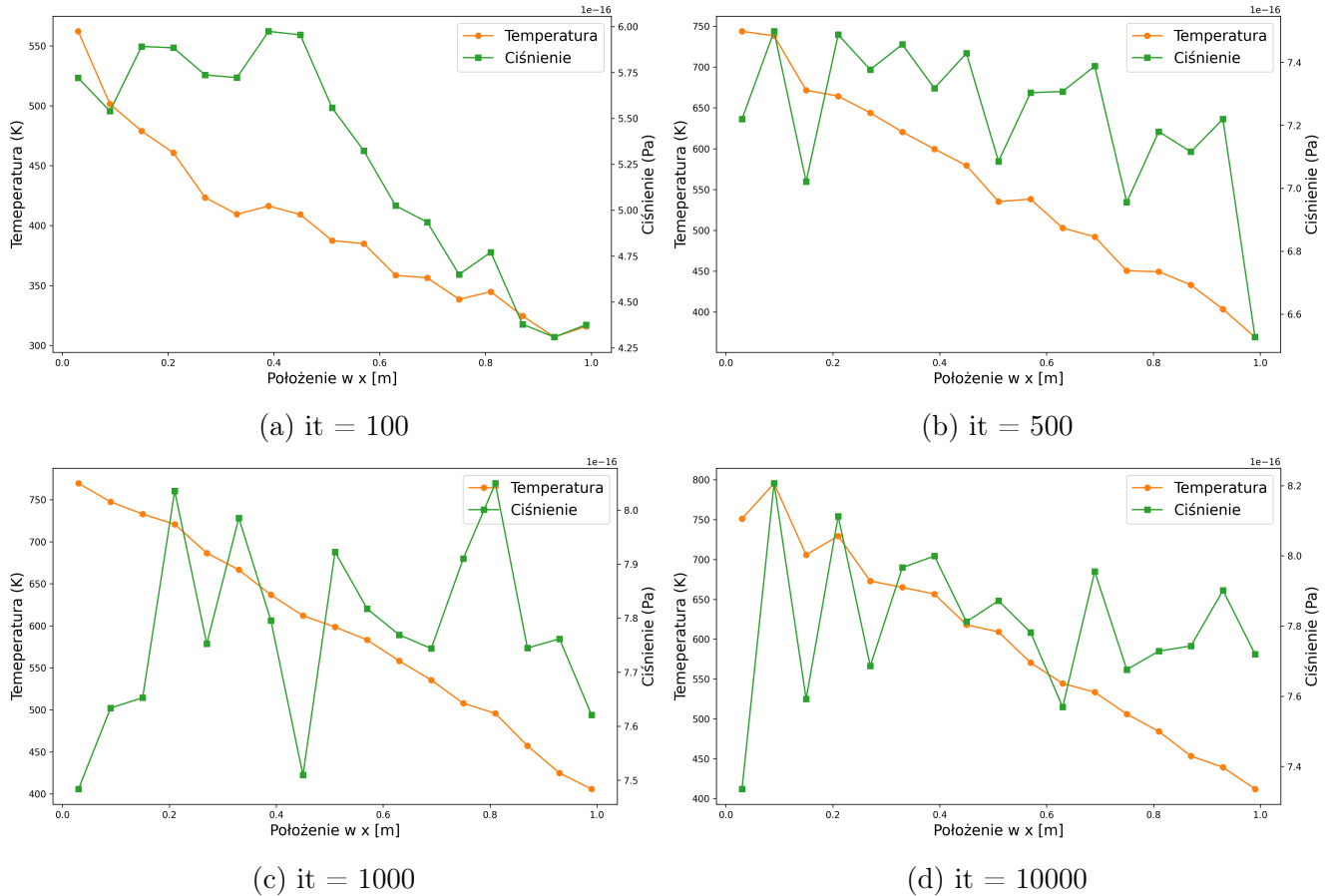


(d) it = 10000

Rysunek 5: Zmiany temperatury i ciśnienia w funkcji położenia x dla różnych iteracji symulacji.

W przypadku, w którym zadajemy temperaturę 1000K na lewym brzegu widzimy, że spada ona miarowo w początkowych iteracjach, a na końcu zupełnie się wyrównuje w przekroju.

### 3.4 Czwarty zestaw parametrów



Rysunek 6: Zmiany temperatury i ciśnienia w funkcji położenia x dla różnych iteracji symulacji.

Tutaj podobnie zgodnie z oczekiwaniami temperatura utrzymuje gradient w każdej iteracji, natomiast zmiany obserwujemy w ciśnieniu, które powoli się stabilizuje na tej samej wartości.

## 4 Wnioski

Metoda DSMC bardzo dobrze symuluje zachowanie się cząsteczek gazu. Zostały sprawdzone zmiany parametrów takich jak rozkład prędkości w gazie, wprowadzenie gradientu temperatury, która się ustabilizowała oraz symulacja grzania jednej ze ścianek układu. We wszystkich przypadkach udało się uzyskać zadowalające wyniki, natomiast wadą tej metody okazał się bardzo długi czas obliczeń dla każdego z układów.