



AKADEMIA GÓRNICZO-HUTNICZA

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ

PODSTAWY FIZYKI TEORETYCZNEJ - LABORATORIUM

Zastosowanie metod dynamiki molekularnej dla gazu rzeczywistego

Autor:
Karol KOZICKI

15 sierpnia 2016

Spis treści

1	Podstawy teoretyczne	2
1.1	Gaz rzeczywisty	2
1.2	Potencjał Lennarda-Jonesa	2
2	Przygotowanie do obliczeń	2
2.1	Wybrane oprogramowanie	2
2.2	Warunki początkowe, stałe	2
3	Wyniki	3
3.1	Periodyczne warunki brzegowe	3
3.2	Zderzenie dwóch cząstek	4
3.3	Obrazy cząstek	6
3.4	Promień odcięcia - tablice Verleta	7
3.5	Symulacja dla wielu cząstek	9
3.5.1	Temperatura gazu	9
3.5.2	Ciśnienie gazu	10
3.5.3	Równanie stanu gazu	11
4	Aplikacja	11
4.1	Kod źródłowy	11
4.2	Opis kodu źródłowego	11
4.3	Dalszy rozwój	12
5	Podsumowanie	12

Streszczenie

Celem pracy jest przeprowadzenie symulacji dynamiki molekularnej gazu rzeczywistego oddziałującego poprzez oddziaływanie Lennarda-Jonesa. W pracy przedstawione są kolejno kroki niezbędne do uzyskania oprogramowania będącego w stanie symulować ruch cząstek a następnie przedstawione są wyniki tych symulacji. Określone zostały także makroskopowe parametry jakimi są temperatura oraz ciśnienie układu.

1 Podstawy teoretyczne

1.1 Gaz rzeczywisty

Gaz rzeczywisty jest to gaz który nie zachowuje się zgodnie z prawami którymi rządzi się gaz doskonały. W praktyce są to wszystkie gazy w przyrodzie bowiem model gazu doskonałego jest jedynie przybliżeniem lepiej lub gorzej działającym w zależności od parametrów opisujących układ.

1.2 Potencjał Lennarda-Jonesa

Potencjał Lennarda-Jonesa jest to jeden z rodzajów potencjałów oddziaływania międzyatomowego. Jest to bardzo dobre przybliżenie dla gazów szlachetnych ale także sprawdza się dla innych atomów. Potencjał opisuje wzór:

$$U(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \quad (1)$$

gdzie:

r_{ij} - jest odległością pomiędzy i-tym oraz j-tym atomem

σ - parametr odległości zależny od pierwiastka

ϵ - parametr energii zależny od pierwiastka

Siłę działającą pomiędzy atomami możemy obliczyć z wzoru:

$$F = -\nabla U \quad (2)$$

Powyższy potencjał w podanej formie został wykorzystany w implementacji.

2 Przygotowanie do obliczeń

2.1 Wybrane oprogramowanie

Wybrany język programowania to *Python3* [3] który jest najlepszym wyborem z racji na swoją prostotę w tworzeniu kodu oraz jego przejrzystość. Aby wykonywane obliczenia były znacznie szybsze wykorzystana została biblioteka *NumPy* wraz z biblioteką *SciPy* do uzyskiwania rozkładów oraz stałych fizycznych. Do graficznego przedstawienia wyników obliczeń numerycznych został wykorzystany program *Gnuplot* [4], umożliwiający wykonywanie złożonych, w pełni dostosowanych wykresów w stosunkowo krótkim czasie.

2.2 Warunki początkowe, stałe

Przyjęte wartości będące stałymi przez cały okres symulacji wynoszą:

- Minimalny promień odcięcia (wy tłumaczone w dalszej części) - $r_c = 0.25(0.3345 * 10^{-9})$
- Promień odcięcia (wy tłumaczone w dalszej części) - $r_c = 10r_c$
- Wymiar pudła obliczeniowego - $dim(x) = dim(y) = dim(z) = 10^{-8}m$
- Potencjał Lennarda-Jonesa - $\epsilon = 125.7k_B$
- Potencjał Lennarda-Jonesa - $\sigma = 0.3345 * 10^{-9}$
- Masa cząstki $m = \frac{39.95 * 10^{-3}}{N_A}$

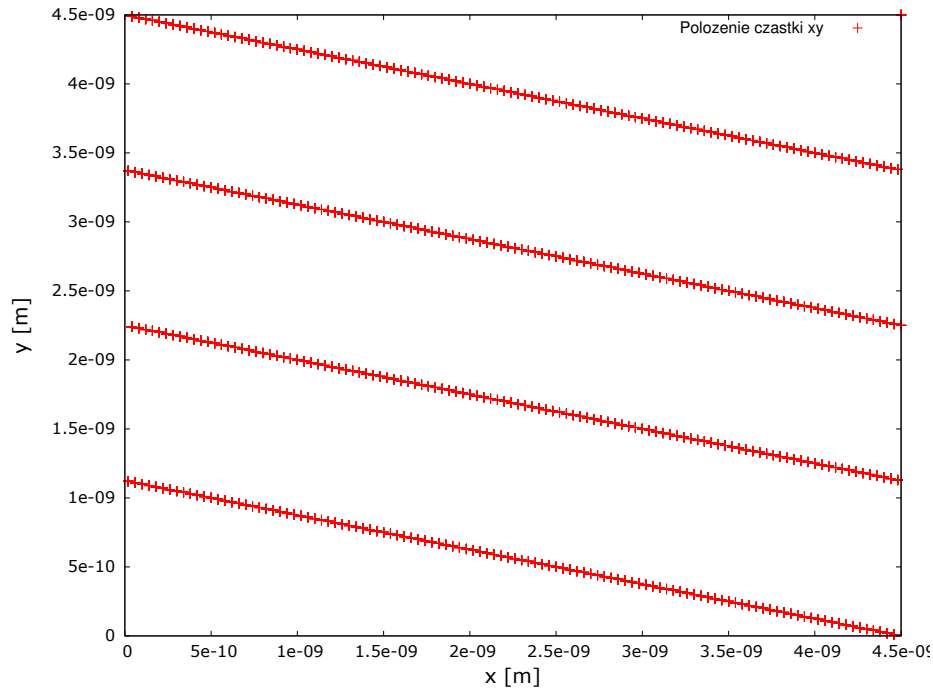
3 Wyniki

3.1 Periodyczne warunki brzegowe

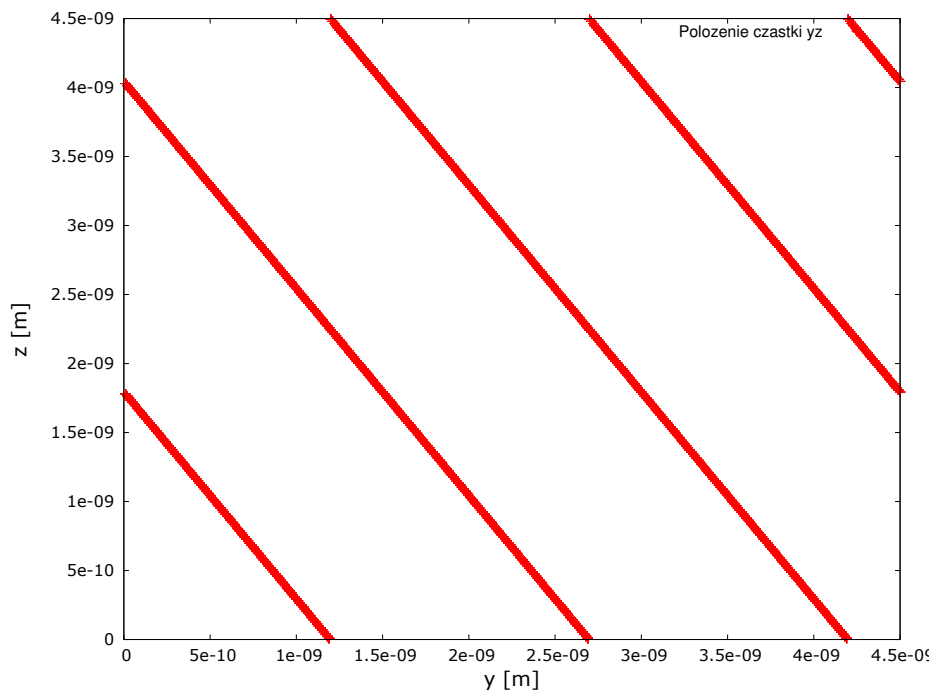
Początkowo zaimplementowane zostały periodyczne warunki brzegowe. Implementacja polega na sprawdzaniu przy każdej iteracji czy pozycja cząstki nie wychodzi poza pudło symulacyjne o określonych z góry wymiarach. Jeśli któraś współrzędna cząstki przekracza odpowiadający wymiar, pozycja jest korygowana poprzez przeniesienie cząstki tak aby znajdowała się ponownie wewnątrz pudła.

Aby zweryfikować poprawność implementacji przeprowadzona została symulacja na jednej cząstce wewnątrz pudła z nadaną prędkością początkową (każda ze składowych prędkości ma inną wartość).

Poniżej przedstawione są wyniki symulacji w formie wykresów:



Rysunek 1: Pozycja cząstki wewnątrz pudła symulacyjnego w rzucie na płaszczyznę xy



Rysunek 2: Pozycja cząstki wewnątrz pudła symulacyjnego w rzucie na płaszczyznę yz

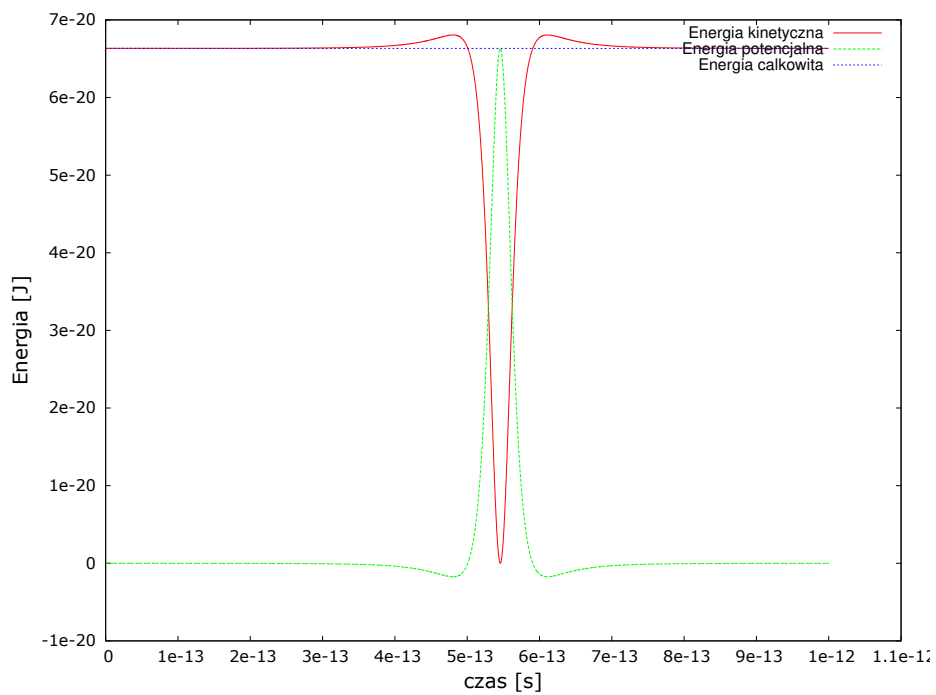
Z powyższych obrazków wnioskujemy, że implementacja periodycznych warunków brzegowych jest poprawna. Pozycja cząstki jest prawidłowo korygowana w każdym z wymiarów.

3.2 Zderzenie dwóch cząstek

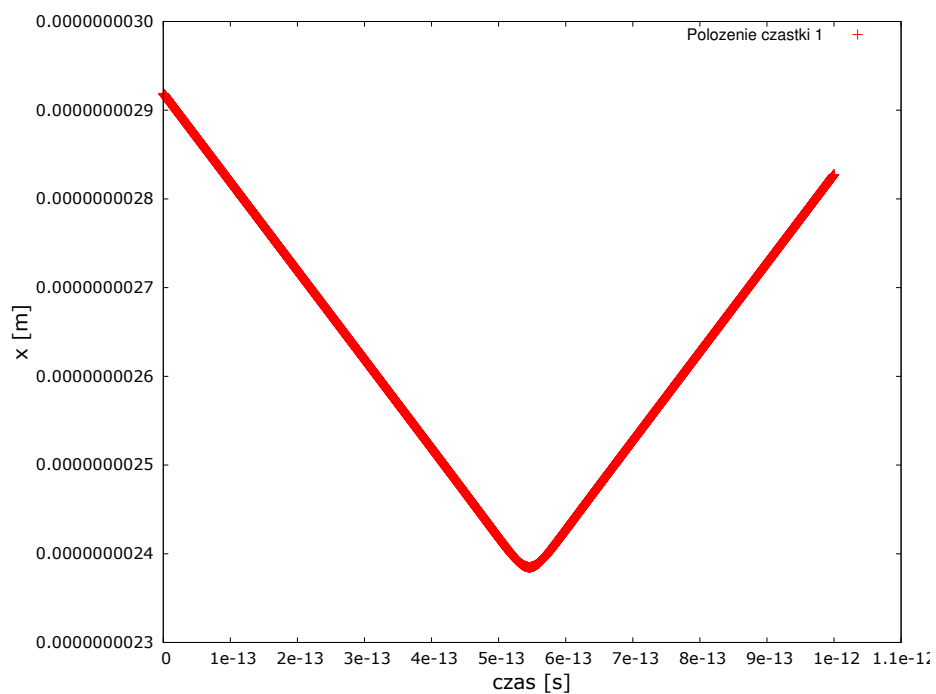
Kolejnym krokiem była implementacja oddziaływania cząstek. Zgodnie z wytycznymi zaimplementowano oddziaływanie Lennarda Jonesa, jednak ogólność implementacji pozwala przeprowadzić symulację dla dowolnego typu oddziaływania, należy jedynie wprowadzić odpowiednie stałe oraz relację energii od odległości.

Aby zweryfikować poprawność implementacji przeprowadzona została symulacja dwóch cząstek początkowo znajdujących się w dużej odległości pozwalającej zaniedbać początkową wartość energii potencjalnej. Prędkości cząstek zostały tak dobrane aby w relatywnie krótkim czasie nastąpiła kolizja tych cząstek.

Poniżej zaprezentowane są wyniki symulacji w formie wykresów:



Rysunek 3: Zależność energii kinetycznej, potencjalnej oraz całkowitej od czasu trwania symulacji



Rysunek 4: Zależność położenia (współrzędnej x) jednej z cząstek od czasu trwania symulacji

Z racji nie występowania żadnych zewnętrznych sił/potencjałów zdolnych zmieniać całkowitą energię układu, spodziewamy się stałości tego parametru. Symulacja zgadza się ze zdrowym rozsądkiem, mianowicie początkowo cząstka miała jedynie energię kinetyczną pochodzącą z niezerowej wartości początkowej prędkości każdej z cząstek. Gdy cząstki znajdują się w dostatecznie bliskiej odległości (jednak nie w bardzo bliskiej) następuje ich przyciąganie - jest to poprawnie opisany proces przez symulację. Widzimy narastanie energii kinetycznej oraz malenie energii potencjalnej. Następnie cząstki zbliżają się na odległość tak bliską, że zaczynają się odpychać. Proces ten poprawnie przedstawia symulacja, widzimy ostry spadek energii kinetycznej na koszt energii potencjalnej - energia ruchu zostaje zmagazynowana w energii potencjalnej oddziaływania. Następnie cząstki zostają odepchnięte i ostatecznie poruszają się tak jak początkowo tylko w przeciwnym kierunku. W całym procesie odbicia energia całkowita jest zachowana co jest poprawnym zachowaniem. Oczywiście wartość ta fluktuuje jednak jest to fluktuacja spowodowana niezerowym krokiem czasowym.

3.3 Obrazy cząstek

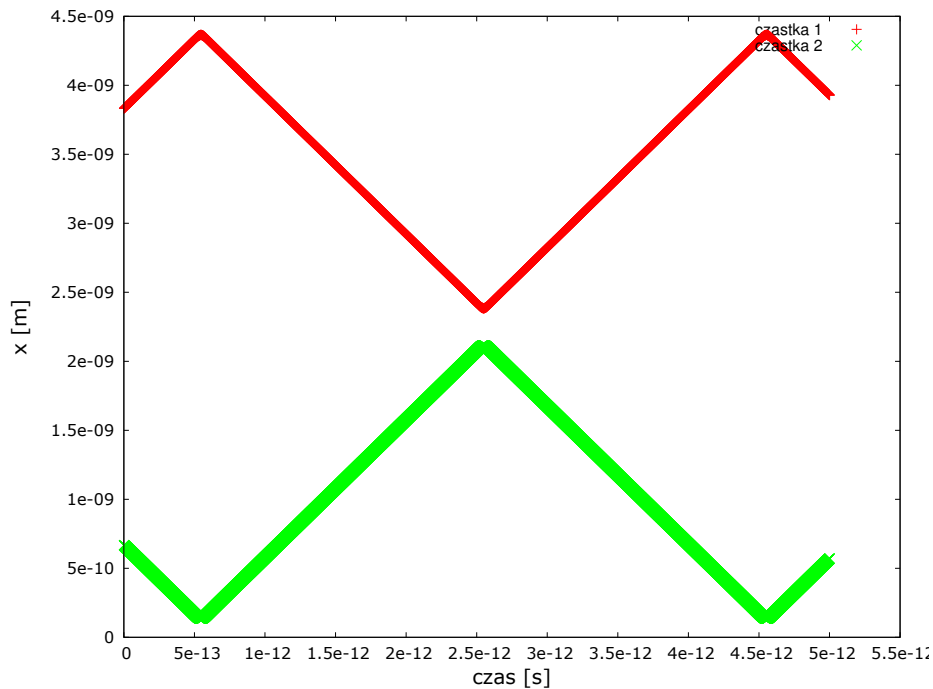
Obrazy cząstki o położeniu r_i są to wszystkie cząstki (wirtualne) w położeniach $r_i + R_n$, gdzie R_n jest długością wymiaru pudła w n-tym wymiarze.

Z powodu wprowadzenia periodycznych warunków brzegowych w obliczeniach musimy brać pod uwagę także obrazy cząstek. Jest to niezbędny zabieg aby obliczenia były poprawne, ponieważ od odległości pomiędzy cząsteczkami zależy całkowita energia potencjalna.

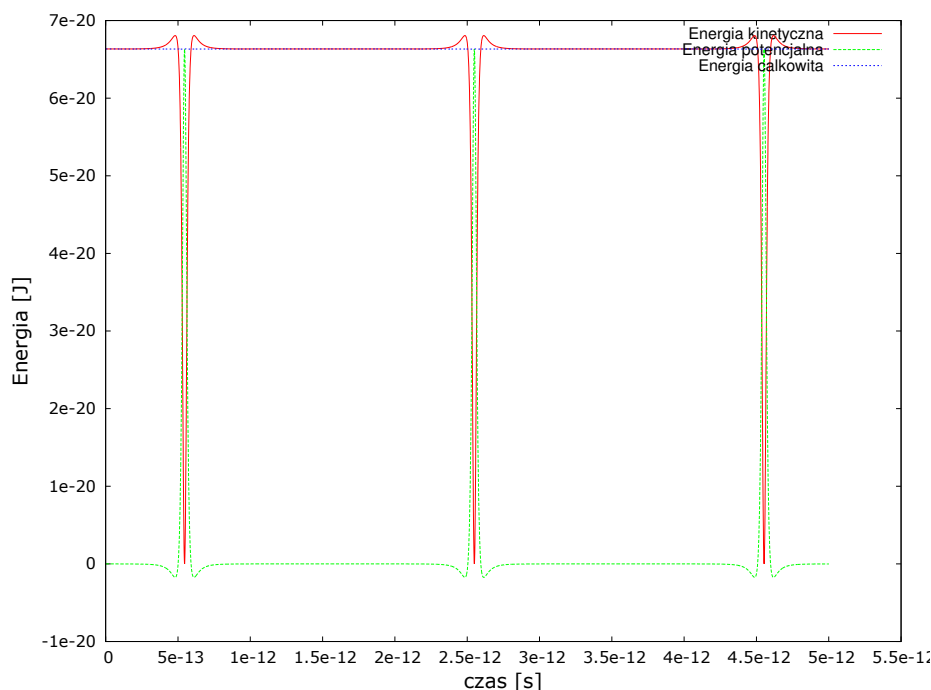
Przy obliczeniach pod uwagę bierzemy najmniejszą z odległości pomiędzy i-tą cząstką a j-tą cząstką (lub jej obrazami). Konwencja ta nazywana jest konwencją minimalnego obrazu.

Aplikacja zawiera niezbędną implementację obrazów cząstek. Aby sprawdzić poprawność implementacji zasymulowano sytuację dwóch cząstek poruszających się naprzeciw siebie. Początkowa odległość została dobrana tak aby odbicie następowało dokładnie na granicy pudła obliczeniowego.

Poniżej zaprezentowane są wyniki symulacji w formie wykresów:



Rysunek 5: Położenie (współrzędna równoległa do prędkości) cząstek odbijających się od siebie na granicy pudła



Rysunek 6: Energia cząstek odbijających się od siebie na granicy pudła

Analizując powyższe wykresy dochodzimy do wniosku, że implementacja obrazów cząstek jest prawidłowa. Przemawiają za tym dwa czynniki. Po pierwsze na wykresie 5 zaobserwować możemy odbicie cząstek od siebie na granicy pudła obliczeniowego. Po drugie na wykresie 6 widzimy stałość energii całkowitej w funkcji czasu.

3.4 Promień odcięcia - tablice Verleta

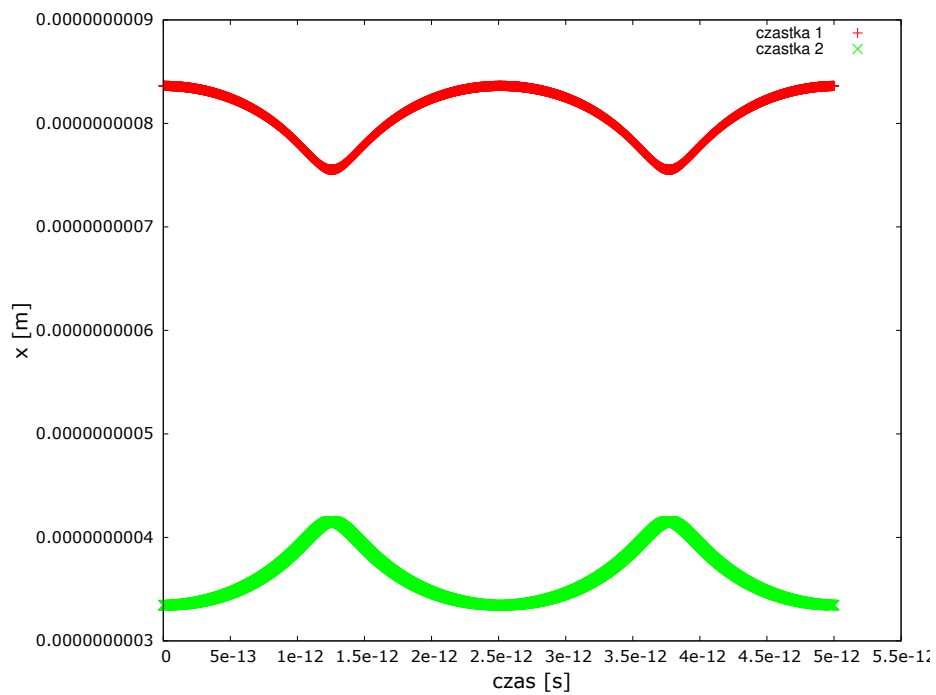
Stosując powyższe uproszczenia w każdym kroku czasowym musimy wykonać (dla N cząstek) $\frac{1}{2}N(N-1)$ obliczeń. W celu zredukowania złożoności obliczeniowej ograniczamy obliczenia do cząstek znajdujących się w bliskich odległościach. Maksymalna odległość w której cząstki oddziałują nazywamy promieniem odcięcia a położenia cząstek znajdujących się w promieniu odcięcia tworzą tablicę Verleta.

Dodatkowo obrano małą liczbę będącą minimalną odległością w jakiej cząstki jeszcze oddziałują. Liczba ta jest na tyle mała, że przy badanych temperaturach cząstki nie są w stanie zliżyć się na tyle aby osiągnąć tak małą odległość. Natomiast należy mieć to na uwadze, że w przypadku symulowania gazu w bardzo wysokich temperaturach należy odpowiednio zmniejszyć parametr minimalnego promienia odcięcia.

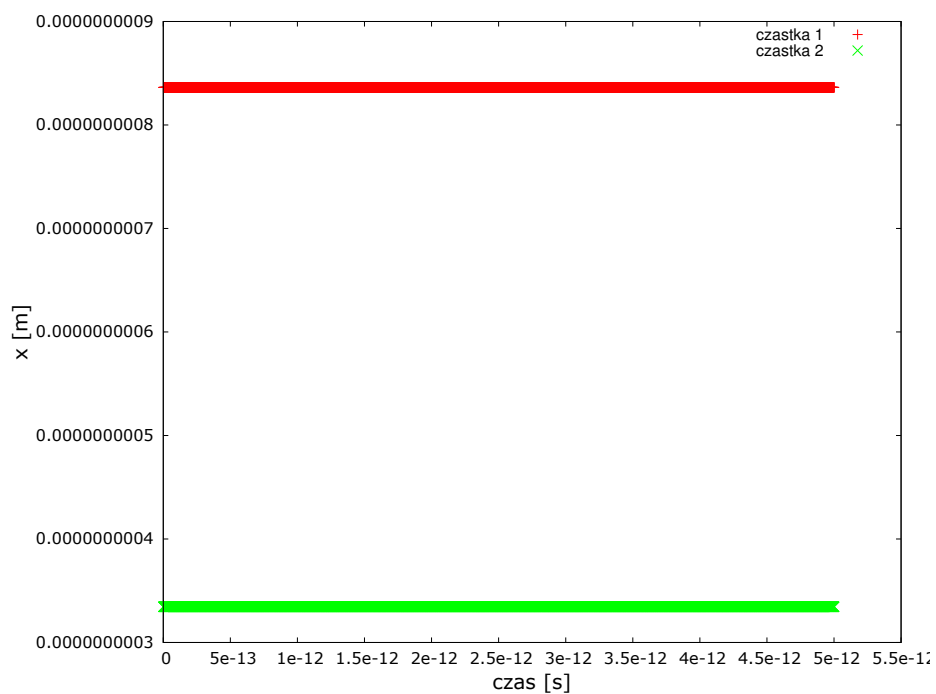
Celem implementacji parametru minimalnego promienia odcięcia jest to żeby w przypadku gdyby jedna cząstka niespodziewanie otrzymała bardzo dużą prędkość kosztem innych (co jest możliwe) to aby zbliżenie się na bardzo dużą odległość nie spowodowało zaistnienia nieskończonej siły która natychmiastowo powodując błąd zatrzymałaby pracę symulacji.

W aplikacji zaimplementowano promień odcięcia. Zasympulowano dwa przypadki. Jeden w przypadku gdy cząstki były w zasięgu oddziaływania a druga poza. W czasie $t = 0$ cząstki nie poruszają się względem siebie.

Poniżej zaprezentowane są wyniki symulacji w formie wykresów:



Rysunek 7: Położenie cząstek w odległości promienia odcięcia



Rysunek 8: Położenie cząstek w odległości większej niż promienia odcięcia

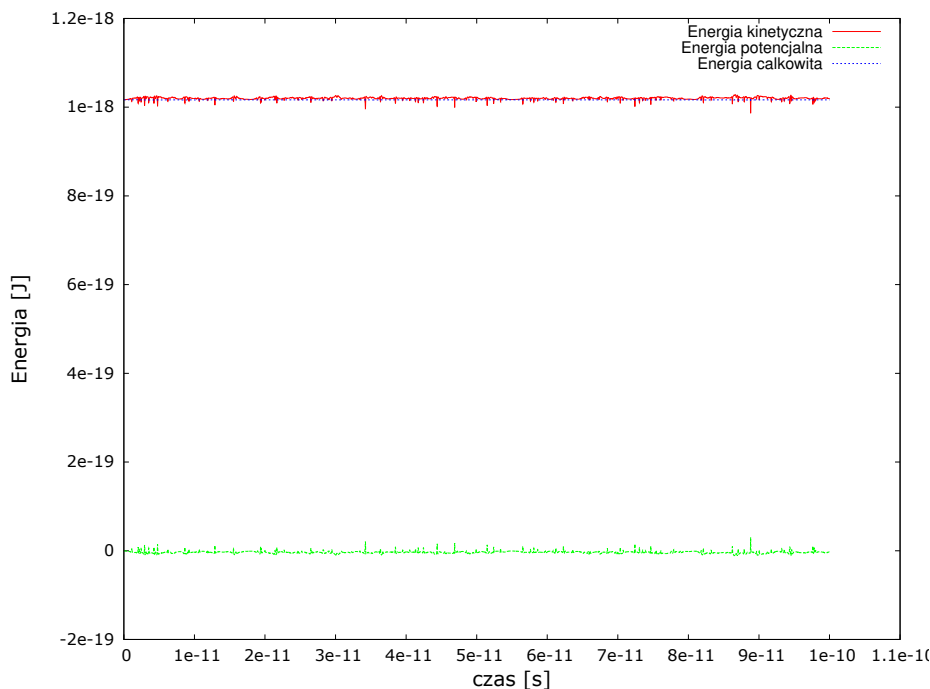
W powyższych wykresów wywnioskować możemy, że promień odcięcia został prawidłowo zaimplementowany. W pierwszym przypadku odległość pomiędzy cząstkami nie przekraczała promienia odcięcia natomiast w drugim przypadku przekraczała.

3.5 Symulacja dla wielu cząstek

Symulacja gazu rzeczywistego została przeprowadzona dla 100 cząstek w trójwymiarowym pudle obliczeniowym o wymiarach $10^{-8}m \times 10^{-8}m \times 10^{-8}m$. Został obrany krok czasowy $10^{-14}s$ natomiast całkowity czas symulacji $10^{-10}s$.

Cząstki układane są w pudle obliczeniowym równomiernie co pewien skok w każdym z wymiarów. Początkowa odległość pomiędzy cząstkami jest tak dobrana aby cząstki znajdowały się poza promieniem odcięcia każdej z cząstek. Dzięki temu na samym początku symulacji energia potencjalna wynosi 0 a z biegiem czasu wartość ta powinna oscylować wokół zera. Prędkość początkowa dla każdej z cząstki losowana jest z rozkładu Maxwella o parametrach takich aby temperatura $500K$ opisywała rozkład.

Poniżej przedstawiony został wykres poszczególnych składowych energii w funkcji czasu.



Rysunek 9: Energia 100 cząstek gazu znajdujących się w pudle obliczeniowym

Symulacja 100 cząstek została przeprowadzona bez żadnych problemów. Energia całkowita układu pozostaje stała. Każdy wzrost energii potencjalnej jest rekompensowany przez zmniejszenie energii kinetycznej i vice versa. Algorytm zachowuje się stabilnie dla obranego czasu trwania symulacji. Czas symulacji wynosił około 370 sekund.

Sprawdzone zostały także parametry makroskopowe układu aby upewnić się, że model oraz symulacja jest poprawna.

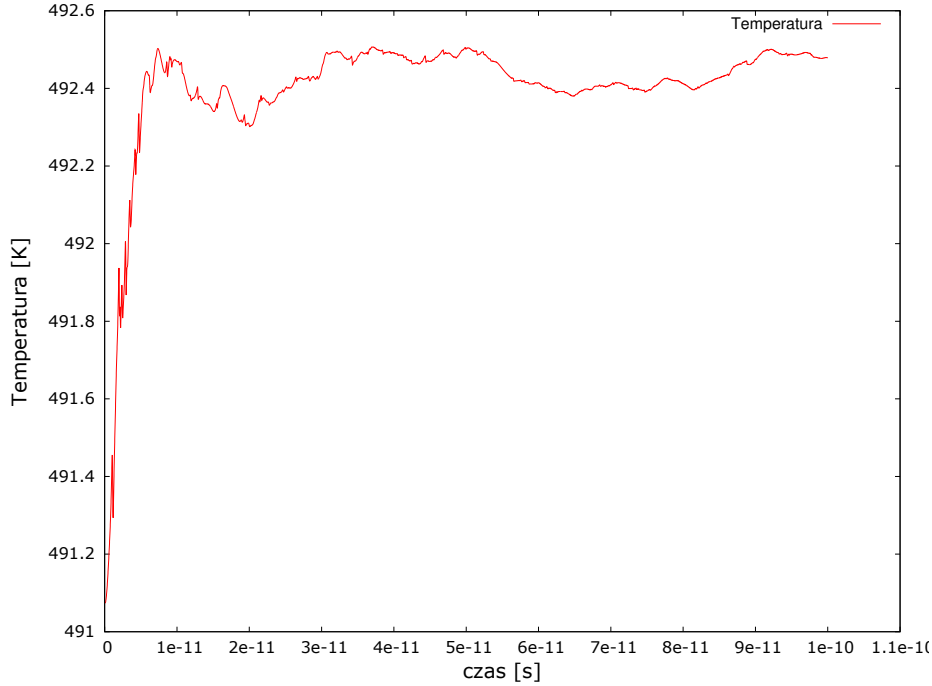
3.5.1 Temperatura gazu

Aby wyznaczyć temperaturę za pomocą metod dynamiki molekularnej należy na początku obliczyć średnią energię kinetyczną cząstek. Następnie korzystając z twierdzenia o ekwipartycji energii można wyprowadzić wzór na bezpośrednią zależność temperatury od prędkości poszczególnych cząstek:

$$T = \frac{2}{3k_B} \frac{1}{MN} \sum_{n=1}^N \sum_{m=1}^M \frac{m}{2} v_i^2(t_n) \quad (3)$$

W przeprowadzonej symulacji temperatura badanego rzeczywistego gazu wynosiła $T = 492.48K$. Wartość ta jest zbliżona do założonej wartości $500K$. Należy pamiętać, że jest to jedynie wartość średnia i przy wielokrotnym uruchamianiu symulacji średnia temperatura gazu ze wszystkich symulacji dążyłaby do temperatury $500K$.

Zbadane zostało także czy temperatura zmienia się wraz z czasem jej trwania.



Rysunek 10: Temperatura gazu w funkcji czasu trwania symulacji

Początkowo temperatura gwałtownie rośnie czego autor nie jest w stanie racjonalnie wytłumaczyć. Aby zbadać ten efekt należałoby sprawdzić czy nie jest to przypadkiem spowodowane niezerowym krokiem czasowym bądź złym początkowym ustawieniem cząstek biorących udział w symulacji. Natomiast po niedługim czasie symulacja ulega ustabilizowaniu i temperatura z dokładnością do niedużych fluktuacji pozostaje stała co jest poprawne gdyż mamy do czynienia z odizolowanym układem.

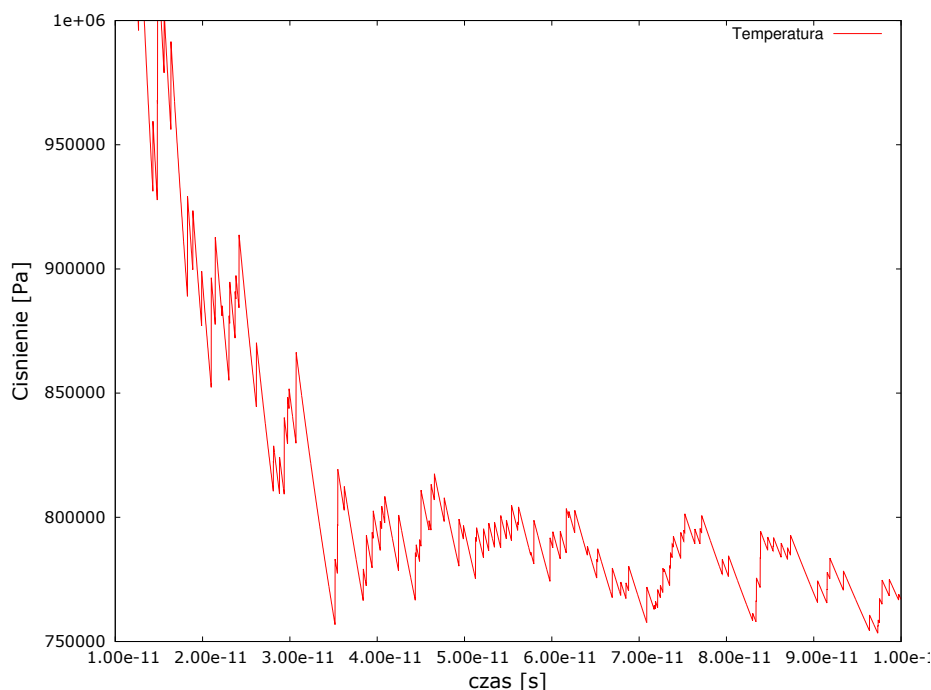
3.5.2 Ciśnienie gazu

Do obliczenia ciśnienia gazu wykorzystany został sposób bazujący na mechanicznej definicji ciśnienia. Mianowicie ciśnienie jest to średni przekaz pędu na jednostkę powierzchni i jednostkę czasu na wybranej powierzchni ΔA :

$$p = m \frac{\Delta v_n^{sr}}{\Delta t \Delta A} \quad (4)$$

gdzie: Δv_n^{sr} jest uśrednioną po czasie zmianą prędkości prostopadłej do rozważanej powierzchni.

W przeprowadzonej symulacji ciśnienie badanego rzeczywistego gazu wynosiło $p = 766877.44 \text{ Pa}$. Zbadana została także zależność ciśnienia od czasu trwania symulacji.



Rysunek 11: Ciśnienie gazu w funkcji czasu trwania symulacji

Należy podkreślić, że początkowe wartości ciśnienia są nieprawdziwe. Dopiero po odpowiednim czasie możemy mówić o statystycznym przepływie cząstek przez odpowiednią ścianę pudła obliczeniowego. Po odpowiednim czasie ciśnienie się stabilizuje i następują jedynie fluktuacje wynikające z tak małej ilości symulowanych cząstek oraz z niezerowego kroku czasowego.

3.5.3 Równanie stanu gazu

Do prostego sprawdzenia czy otrzymane wyniki nie są przekłamane sprawdzona została wartość ciśnienia jaka panowała by w pudle obliczeniowym przy zadanej temperaturze gdyby do gazu stosowało się równanie Clapeyrona (gaz doskonały) - co przy zadanej temperaturze jest dopuszczalnym uproszczeniem.

Zgodnie z równaniem Clapeyrona:

$$pV = NkT \quad (5)$$

Obliczając ciśnienie przy temperaturze $T = 492.48K$ otrzymujemy ciśnienie równe $p = 679969 Pa$ co jest wartością bardzo zliżoną do wartości otrzymanej z symulacji. Pozwala to stwierdzić jakoby model oraz sposób przeprowadzania symulacji był prawidłowy.

4 Aplikacja

4.1 Kod źródłowy

Oprogramowanie dostępne jest w publicznym repozytorium GitHub pod adresem[6]:

<https://github.com/KaRi94/MolecularDynamicsSimulator>

Aplikacja dostępna jest pod licencją MIT[5].

4.2 Opis kodu źródłowego

Kod źródłowy posiada dokumentację metod oraz użytych klas w formie docstringów.

4.3 Dalszy rozwój

Oprogramowanie wymaga jeszcze dużo pracy aby mogło zostać użyte w naukowych celach. Autor sugeruje następujący dalszy rozwój aplikacji:

- Wielowątkowość w obliczeniach - w przypadku wielowątkowych komputerów kilkukrotny wzrost szybkości obliczeń
- Zmiana implementacji metod odpowiedzialnych za obliczanie makroskopowych właściwości oraz energii potencjalnej tak aby uwzględniały możliwość współistnienia w jednym pudle obliczeniowym gazów o różnych cząsteczkach/atomach.
- Zmodularyzowanie klasy odpowiedzialnej za typ cząstki oraz rodzaj oddziaływania tak aby ich zmiana nie wiązała się z ingerencją w kod źródłowy. Użytkownik mógłby załadować swoje typy cząstek/atomów.
- Poprawa złożoności obliczeniowej (autor na czas dzisiejszy wie, że można obliczyć coś z lepszą złożonością wymaga to jednak zmian implementacyjnych)
- Implementacja poprawki długozasięgowej

5 Podsumowanie

Podsumowując, problem badania gazu rzeczywistego za pomocą metod dynamiki molekularnej jest problemem niezwykle rozbudowanym. Obecna szybkość komputerów oraz superkomputerów nie jest wystarczająca abyśmy mogli symulować rzeczywisty gaz w którym mamy ok. 10^{23} cząstek. Najprawdopodobniej jeszcze przez wiele lat nie będziemy w stanie symulować ruch tak ogromnej ilości cząstek. Dlatego wykorzystane metody dynamiki molekularnej będą aktualne a ich ulepszanie oraz wiedza które aspekty można pominąć aby przyspieszyć proces symulacji będzie niezbędna do symulowania gazu rzeczywistego. Zbudowany model dobrze symuluje wiele aspektów cząstek poruszających się w zadanej objętości. Należy jednak podkreślić, że obecna aplikacja jest wolna i aby móc wykorzystywać ją do naukowych celów należałoby przebudować fragmenty kodu tak aby zyskać na złożoności obliczeniowej.

Literatura

- [1] https://pl.wikipedia.org/wiki/Algorytm_Verleta
- [2] http://www.ftj.agh.edu.pl/~adamowski/wyklady_mofit_1/r9.pdf
- [3] <https://www.python.org/>
- [4] <http://www.gnuplot.info/>
- [5] https://en.wikipedia.org/wiki/MIT_License
- [6] <https://github.com/KaRi94/MolecularDynamicsSimulator>