

Instytut Fizyki Uniwersytetu Jagiellońskiego

Jakub Pawłowicz

Symulacja polimerów Lennarda-Jonesa

Praca magisterska wykonana w Zakładzie Fizyki Statystycznej pod kierunkiem profesora doktora habilitowanego Lecha Longi w roku akademickim 2003/04

Kraków 2004

Pracę pragnę zadedykować mojej mamie.

Streszczenie

Tematem pracy jest stworzenie pakietu programów służących do badania dynamiki układu polimerów umieszczonych w łaźni o skończonej temperaturze, poddanych działaniu stochastycznej siły zewnętrznej. Modelem polimeru jest trójwymiarowy łańcuch atomów oddziaływujących poprzez potencjał harmoniczny oraz Lennarda-Jonesa. Potencjał harmoniczny działa pomiędzy sąsiednimi atomami w łańcuchu i jest odpowiednikiem wiązania chemicznego pomiędzy atomami polimeru. Natomiast potencjał Lennarda-Jonesa działa pomiędzy wszystkimi atomami wchodzącymi w skład łańcucha i jest odpowiedzialny za oddziaływania daleko zasięgowe. użytą metodą badawczą są symulacje komputerowe, z użyciem algorytmów dynamiki molekularnej. Zewnętrzna siła stochastyczna odpowiada za oddziaływanie polimeru z otoczeniem.

W pracy zostanie omówiona strona fizyczna procesów związanych z dynamiką polimerów oraz próba odtworzenia dynamiki za pomocą symulacji komputerowych. W szczególności omówię podstawowe algorytmy dynamiki molekularnej oraz zastosowanie generatora liczb pseudolosowych w układzie o skończonej temperaturze. Stworzone programy posłużą mi do przeanalizowania zjawiska zwijania, oraz rozwijania się polimerów.

Otrzymane wyniki są w dużej mierze zgodne z przewidywaniami teoretycznymi oraz z wynikami otrzymanymi w pracach doświadczalnych [18]. Na podstawie otrzymanych danych omówię możliwe sposoby uzyskania dokładniejszych wyników, poprzez wykorzystanie symulacji komputerowych. Na zakończenie omówię stworzony pakiet programów, służący do badania dynamiki układów polimerowych, oddziaływujących potencjałem harmonicznym oraz potencjałem Lennarda-Jonesa.

Spis treści.

1. Wstęp.	4
2. Dynamika polimeru Lennarda-Jonesa.	6
2.1 Model symulacyjny.	6
3. Numeryczne metody rozwiązywania równań ruchu.	9
3.1 Algorytmy dynamiki molekularnej.	9
3.1.1 Algorytm Verleta.	10
3.1.2 Algorytm Leap Frog.	10
3.1.3 Algorytm prędkościowy Verleta.	11
3.1.4 Algorytmy wielokrokowe.	11
3.1.5 Metody typu predictor-corrector.	12
3.2 Wybór algorytmu do symulacji dynamiki molekularnej.	12
3.3 Równanie Langevina.	12
4. Generatory liczb pseudolosowych.	14
4.1 Generatory liniowe.	14
4.2 Generatory nieliniowe.	14
4.3 Generowanie ciągu liczb o rozkładzie normalnym.	15
4.4 Test użytego generatora liczb losowych.	15
5. Wyniki.	17
5.1 Stan podstawowy.	17
5.2 Zwijanie się polimeru Lennarda-Jonesa.	17
5.3 Rozwijanie się polimeru Lennarda-Jonesa.	18
5.4 Promień polimeru.	18
5.5 Energia polimeru.	21
6. Dyskusja.	23
7. Podsumowanie.	25
A. Omówienie pakietu programów do symulacji dynamiki molekularnej.	26
A.1 Program do symulacji polimerów.	26
A.2 Program do wizualizacji danych.	27
A.3 Program do analizy danych.	28
A.4 Program do generowania łańcuchów polimerowych.	28
B. Jednostki zredukowane.	30

1. Wstęp.

Metody symulacji komputerowych stały się w ostatnich latach prężnie rozwijającą się gałęzią współczesnej nauki. Stało się tak z co najmniej dwóch równorzędnych powodów. Po pierwsze wiele modeli teoretycznych zawiera bardzo skomplikowaną matematykę, co powoduje, że rzadko istnieją ich rozwiązania analityczne, a jeśli istnieją to tylko dla prostych przypadków. Z pomocą przychodzą tu symulacje komputerowe, dzięki którym możemy pozbyć się problemu nadmiernej złożoności obliczeniowej modeli teoretycznych. Często jednak dzieje się to dużym kosztem: konieczności dyskretyzacji procesu, stworzenia nowych algorytmów zdolnych rozwiązać problem czy też czasochłonności obliczeń. Drugą dziedziną nauki, w której symulacje komputerowe znalazły szerokie zastosowanie, jest weryfikowanie modeli teoretycznych użytych do opisu eksperymentów. Zastosowanie eksperymentu komputerowego pozwala przetestować model teoretyczny i uzyskać dla niego dokładne wyniki, które następnie mogą być sprawdzone drogą eksperymentalną.

Symulacje komputerowe można podzielić na dwie grupy: wykorzystujące metody stochastyczne (Monte-Carlo) lub deterministyczne (dynamika molekularna). Algorytmy pierwszego rodzaju działają w oparciu o wyznaczanie prawdopodobieństw przejścia układu ze stanu aktualnego do stanów sąsiednich. Szansa przejścia układu do innego stanu jest proporcjonalna do tego prawdopodobieństwa. Metody dynamiki molekularnej opierają się na wyznaczeniu trajektorii ruchu poprzez numeryczne rozwiązywanie równań ruchu.

Niniejsza praca jest poświęcona metodzie symulacji dynamiki molekularnej łańcuchów atomów oddziaływujących potencjałem harmonicznym i Lennarda-Jonesa, w których oddziaływanie z otoczeniem uwzględnia się przy pomocy sił Langevina. Potencjał harmoniczny, działający pomiędzy sąsiednimi atomami w łańcuchu, jest odpowiednikiem wiązania chemicznego pomiędzy atomami polimeru. Jest on odpowiedzialny za stabilizację układu w wysokich temperaturach. Z kolei potencjał Lennarda-Jonesa, jako że jest zdefiniowany pomiędzy wszystkimi atomami łańcucha, jest odpowiedzialny za występowanie oddziaływań daleko zasięgowych. W oparciu o tak zdefiniowany model zostaną zbadane dynamiczne własności trójwymiarowych łańcuchów atomów, które mogą być modelami prostych polimerów lub białek.

Aby lepiej zrozumieć analizowany proces, przyjrzyjmy się równaniom opisujących ruch pojedynczego atomu w łańcuchu. Wektorowe równanie ruchu dla takiego atomu, można zapisać jako funkcję oddziaływań z sąsiednimi atomami łańcucha, w postaci:

$$M\ddot{\vec{x}}_i = \vec{F}_L + \vec{F}_R, \quad (1)$$

gdzie M oznacza masę, a \vec{x}_i to odległość i -tej cząstki od początku układu współrzędnych. \vec{F}_L i \vec{F}_R to odpowiednio siła pochodząca od oddziaływania z atomami po lewej i prawej stronie w łańcuchu. Siły oddziaływujące na cząstkę są wyrażone jako suma sił pochodzących od potencjału harmonicznego i potencjału Lennarda-Jonesa:

$$\vec{F}_L = \vec{F}_H(\vec{x}_j - \vec{x}_{j-1}) + \sum_j^{j < i} \vec{F}_{LJ}(\vec{x}_i - \vec{x}_j),$$

$$\vec{F}_R = \vec{F}_H(\vec{x}_{j+1} - \vec{x}_j) + \sum_j^{j>i} \vec{F}_{LJ}(\vec{x}_j - \vec{x}_i),$$

gdzie \vec{F}_H oznacza siłę pochodzącą od potencjału harmonicznego, daną wzorem:

$$U_H(r) = K(r - r_0)^2, \quad (2)$$

$$\vec{F}_H(r) = -\vec{\nabla} U_H(r) = -2K(r - r_0) \cdot \hat{r}.$$

Współczynnik r_0 jest punktem w którym potencjał harmoniczny posiada minimum, K stałą sprężystości, a \hat{r} wersorem jednostkowym w kierunku działania siły. Dla członu \vec{F}_{LJ} , będącego siłą pochodzącą od potencjału Lennarda-Jonesa, zachodzą następujące związki:

$$U_{LJ}(r) = \varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^6 \right], \quad (3)$$

$$\vec{F}_{LJ}(r) = -\vec{\nabla} U_{LJ}(r) = \frac{12\varepsilon}{r} \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \cdot \hat{r}.$$

Tak, że potencjał U_{LJ} ma minimum w punkcie $r = \sigma$ wynoszące $-\varepsilon$.

Równanie (1) nie uwzględnia oddziaływania łańcucha z otoczeniem. Aby dodać oddziaływanie z otoczeniem, zaproponowano dopisanie do prawej stronie równania członów tłumiącego i stochastycznego. Równanie (1) przechodzi wtedy w:

$$M\ddot{\vec{x}}_i = \vec{F}_L + \vec{F}_R - M\gamma\dot{\vec{x}}_i + \vec{\xi}_i, \quad (4)$$

gdzie γ oznacza współczynnik tłumienia, a $\vec{\xi}_i$ losową siłę wymuszającą zdefiniowaną poprzez twierdzenie fluktuacyjno-dysypacyjne. Dodane wyrazy reprezentują oddziaływanie łańcucha z rozpuszczalnikiem oraz pozostałe, nie uwzględnione polimery. Związek pomiędzy γ a $\vec{\xi}_i$ zostanie dokładniej omówiony w rozdziale 2. Równanie (4) jest zwane **równaniem Langevina**.

Aby w pełni zdefiniować model polimeru użyty w symulacjach, konieczne jest podanie wyrażeń charakteryzujących współczynnik tłumienia oraz losową siłę wymuszającą. Omówieniu modelu symulacyjnego zostanie poświęcony następny rozdział.

2. Dynamika polimeru Lennarda-Jonesa.

Ten rozdział zostanie poświęcony opisowi metody symulacji polimerów oddziaływujących przez potencjał harmoniczny oraz Lennarda-Jonesa. Omawiany proces nie posiada rozwiązania analitycznego, więc możliwość jego analizy istnieje jedynie poprzez symulacje komputerowe. Proces zwijania oraz rozwijania się łańcucha polimerowego zachodzi głównie pod wpływem sił Lennarda-Jonesa i zewnętrznej siły losowej, co omówię w dalszej części pracy.

2.1 Model symulacyjny.

Symulacje zostaną przeprowadzone według teorii zaproponowanej we wstępie. Symulacjom zostaną poddane łańcuchy o długości N cząstek, każda o masie m , poddanych ruchowi opisanemu przez równanie Langevina (4). Potencjał harmoniczny (2), zdefiniowany dla sąsiednich atomów w łańcuchu, odpowiada za stabilizację układu w wyższych temperaturach i jest odpowiednikiem wiązania chemicznego występującego pomiędzy atomami polimeru. Potencjał Lennarda-Jonesa (3) odpowiada za oddziaływania daleko zasięgowe i działa pomiędzy wszystkimi atomami łańcucha. Oddziaływanie układu z otoczeniem jest wprowadzone poprzez człon stochastyczny i człon tłumiący. Amplituda szumu stochastycznego musi być ściśle ustalona, tak aby układ znajdował się w temperaturze T , zadanej jako parametr symulacji. Zależność amplitudy szumu od temperatury można uzyskać z twierdzenia fluktuacyjno-dysypacyjnego [8]. Według tego twierdzenia, jeśli ruch układu opisany jest następującym równaniem wektorowym:

$$M\ddot{\vec{x}}(t) = \vec{F}(\vec{x}(t)) - M \int_{t_0}^{t_1} \gamma(t-t') \dot{\vec{x}}(t-t') dt' + \vec{\xi}(t), \quad (5)$$

gdzie \vec{F} jest siłą deterministyczną, a $\vec{\xi}(t)$ siłą losową, to, aby układ znajdował się w stanie równowagi termodynamicznej o temperaturze T , siła losowa $\vec{\xi}(t)$ musi spełniać następujące warunki:

$$\langle \xi_{i_\alpha}(t) \xi_{i'_\beta}(t') \rangle = 2Mk_B T \delta_{ii'} \delta_{\alpha\beta} \gamma(t-t'), \quad (6)$$

$$\langle \xi_{i_\alpha}(t) \rangle = 0, \quad (7)$$

gdzie indeksy α i β przebiegają po współrzędnych wektora $\vec{\xi}(t)$. W przyjętym w symulacjach modelu zakładam, że wpływ szumu na układ polega na działaniu pojedynczych, niezależnych sił losowych na każdy atom wchodzący w skład łańcucha. Jeśli czas takiego uderzenia jest w przybliżeniu równy długości kroku czasowego symulacji, to siła losowa, działająca na każdy atom wchodzący w skład układu, podczas pojedynczego kroku czasowego jest stała. Pominięte zostają również efekty związane z pamięcią. Przy tych założeniach szum w równaniu musi być nieskorelowany, co implikuje jego następującą własność:

$$\gamma(t-t') = \gamma\delta(t-t').$$

Korzystając z zależności (7) można dowieść, że siła losowa $\xi(t)$ musi spełniać następujące warunki:

$$\begin{aligned}\text{var}(\xi_{i_\alpha}(t)) &= \frac{2Mk_B\gamma T}{\Delta t}, \\ \xi_{i_\alpha}(t) &= \eta_i(t)\sqrt{\frac{2M\gamma k_B T}{\Delta t}},\end{aligned}\tag{8}$$

gdzie $\eta_i(t)$ jest zmienną losową o rozkładzie normalnym $N(0,1)$. Warunek (8) jest spełniony, ponieważ:

$$\text{var}(\eta_i(t)) = 1.$$

Warunek (8) znajduje ważne zastosowanie w symulacjach układów, na które działa zewnętrzna siła stochastyczna, gdyż można przez niego kontrolować temperaturę układu.

W celu dokładnego zdefiniowania modelu wybrałem następujące parametry opisujące zachowanie się układu:

opis stałej	symbol stałej	wartość stałej
minimum potencjału Lennarda-Jonesa	σ	1.0
głębokość studni potencjału Lennarda-Jonesa	ε	1.0
minimum potencjału Lennarda-Jonesa	r_0	1.0
stała sprężystości	K	16.67
masa atomów w łańcuchu	M	1.0
stała tłumienia	γ	$3\sqrt{2}$

Tablica 1. Parametry symulacji

Wartość stałej tłumienia K wybrałem tak, aby potencjał harmoniczny odtwarzał potencjał Lennarda-Jonesa na małych odległościach od punktu równowagi. Pozostałe parametry symulacji przyjąłem tak jak w pracy [1]. Współczynnik tłumienia wyznaczyłem z następującej zależności [1]:

$$\gamma = \frac{\omega_0}{4} = 3\sqrt{\frac{2\varepsilon}{Ma^2}} = 3\sqrt{2},$$

gdzie a jest odległością atomów w stanie równowagi, wynoszącą 1.0, a ω_0 częstością najmniejszych drgań w układzie. Czas jest podany w jednostkach częstotliwości:

$$\tau = \frac{2\pi}{\omega_0} t.$$

Wartości energii, temperatury oraz czasu są podane w jednostkach zredukowanych, których pełne definicje zamieszczone są w dodatku B. Opis programu do symulacji znajduje się w dodatku A.

3. Numeryczne metody rozwiązywania równań ruchu.

Wielkością charakteryzującą ruch układu w polu potencjału, jest lagranżjan [3]. Zwykle, wobec zapostulowania zależności lagranżjanu opisującego układ jedynie od położenia x , prędkości \dot{x} oraz czasu t , równania, które należy rozwiązać aby poznać ewolucję czasową układu, są równaniami różniczkowymi drugiego rzędu ze względu na położenie $x(t)$ [7]. W związku z tym, przy numerycznym całkowaniu takich równań, najczęściej stosowane są algorytmy stworzone specjalnie do rozwiązywania równań różniczkowych drugiego rzędu. Metody te znane są pod nazwą algorytmów dynamiki molekularnej. Zasada ich działania opiera się na wyznaczaniu trajektorii układu krok po kroku. Korzystając z parametrów trajektorii wyznaczonych w kroku obecnym, i w krokach poprzednich, wyznaczane są parametry następnego kroku.

3.1 Algorytmy dynamiki molekularnej.

Najprostszym podejściem do rozwiązywania problemu wyznaczenia trajektorii układu w następnym kroku czasowym, jest rozwinięcie położenia i prędkości w szereg Taylora wokół aktualnego położenia:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + O(\Delta t^2), \quad (9)$$

$$v(t + \Delta t) = v(t) + a(t)\Delta t + O(\Delta t^2), \quad (10)$$

gdzie przyśpieszenie $a(t)$ wyznaczamy z potencjału $U(x)$, w polu którego porusza się układ:

$$a(t) = \ddot{x}(t) = -\frac{1}{m} \frac{dU(x)}{dx}. \quad (11)$$

Metoda ta nosi nazwę metody Eulera i jest najprostszą metodą pozwalającą numerycznie rozwiązać równanie różniczkowe drugiego rzędu. Metoda Eulera nie jest używana w obliczeniach numerycznych z powodu niestabilności otrzymanego przy jej użyciu rozwiązania.

Dla niektórych przypadków można ominąć ten problem dobierając mniejszy krok czasowy Δt . Jako że otrzymana trajektoria zależy liniowo od Δt , to w granicy $\Delta t \rightarrow 0$ trajektoria układu powinna odtwarzać prawdziwą trajektorię. Jednak zmniejszając krok czasowy, czas potrzebny do znalezienia całkowitej trajektorii układu znacznie się wydłuża, co powoduje, że niezbędne staje się wybranie pewnego optymalnego kroku Δt . Efektem dokonania takiego wyboru jest wygenerowanie nieprawdziwej trajektorii. Dodatkowym powodem błędnie wyznaczonej trajektorii jest skończona precyzja obliczeń numerycznych wykonywanych na komputerze. Koniecznością jest więc stworzenie takiego algorytmu, który przy dużym kroku czasowym Δt , zachowuje się stabilnie i jak najlepiej odtwarza trajektorię układu. Wiele pożytecznych informacji na temat algorytmów dynamiki molekularnej można znaleźć w książkach [3, 4, 9], oraz w pracy [7]. W następnych punktach omówię cechy kilku wybranych algorytmów, które zaimplementowałem w programie do symulacji.

3.1.1 Algorytm Verleta.

Algorytm ten bazuje na rozwinięciu szeregu Taylora w dwóch chwilach czasowych:

$$x(t - \Delta t) = x(t) - v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2,$$

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2.$$

Dodając te równania stronami otrzymujemy:

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + a(t)\Delta t^2. \quad (12)$$

Widać stąd, że w tej metodzie do wygenerowania trajektorii $x(t)$ nie jest potrzebna znajomość prędkości $v(t)$, a jedynie położenia w dwóch poprzednich chwilach oraz przyspieszenia obliczonego z równania (11). Znajomość prędkości konieczna jest jednak do wyznaczenia np. energii kinetycznej. Można ją wyznaczyć z następującej zależności:

$$v(t) = \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t}. \quad (13)$$

Jak wcześniej wspomniano, prędkością i stabilnością obliczeń można sterować, dobierając odpowiednio krok czasowy Δt . Gdy w metodzie Verleta zmniejsza się Δt , okazuje się że przyrosty położenia $x(t + \Delta t) - x(t - \Delta t)$ stają się coraz mniejsze. Co za tym idzie, obliczając prędkość dzieli się dwie małe liczby przez siebie. W efekcie, przy obliczaniu prędkości, istotną rolę zaczynają grać błędy przypadkowe, wynikające np. z przeprowadzanych w pamięci komputera zaokrągleń. Z tego powodu metoda ta jest często niestabilna numerycznie.

3.1.2 Algorytm Leap Frog.

W celu polepszenia właściwości algorytmu Verleta zaproponowano następujące usprawnienie [14]:

$$x(t + \Delta t) = x(t) + v(t + \frac{1}{2}\Delta t)\Delta t, \quad (14)$$

$$v(t + \frac{1}{2}\Delta t) = v(t - \frac{1}{2}\Delta t) + a(t)\Delta t. \quad (15)$$

Aby obliczyć prędkość najczęściej stosuje się zależność:

$$v(t) = \frac{1}{2} \left[v(t + \frac{1}{2}\Delta t) + v(t - \frac{1}{2}\Delta t) \right]. \quad (16)$$

Metoda ta jest dużo stabilniejsza od metody Verleta, ale wymaga dwukrotnie większego nakładu obliczeń. Wadą jest z pewnością dość niewygodny i nie intuicyjny sposób wyznaczania prędkości. Tą niedogodność likwiduje kolejny algorytm, tzw. prędkościowy algorytm Verleta.

3.1.3 Algorytm prędkościowy Verleta.

Metoda pozbawiona wad poprzednich algorytmów została podana w pracy [15]. Obliczenia są wykonywane w następującej kolejności:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2, \quad (17)$$

$$v(t + \frac{1}{2}\Delta t) = v(t) + \frac{1}{2}a(t)\Delta t, \quad (18)$$

$$v(t + \Delta t) = v(t + \frac{1}{2}\Delta t) + \frac{1}{2}a(t + \Delta t)\Delta t. \quad (19)$$

W tej metodzie prędkość w kolejnym kroku czasowym oblicza się w dwóch etapach. W pierwszym, prędkość wyznacza się w oparciu o aktualne przyśpieszenie, a w drugim dodaje się poprawkę ze względu na przyśpieszenie w kolejnym kroku. W porównaniu z poprzednimi metodami, istotną zaletą tego algorytmu jest jej stabilność (inaczej niż w metodzie Verleta), oraz intuicyjny sposób wyznaczania prędkości. Koszt obliczeniowy tej metody jest zaledwie o połowę większy niż w algorytmie Leap Frog, a otrzymywane dzięki niej rozwiązania posiadają znacznie lepsze własności, niż otrzymywane z wcześniej omawianych metod.

3.1.4 Algorytmy wielokrokowe.

Często pożądaną cechą algorytmów dynamiki molekularnej jest jak najdokładniejsze odtworzenie trajektorii układu. Z myślą o takich zastosowaniach stworzono specjalną grupę algorytmów wielokrokowych. Ideą tych metod jest wykorzystanie parametrów w wielu poprzednich krokach do wyznaczenia położenia układu w następnym kroku czasowym. Przykładem algorytmu wielokrokowego jest algorytm Beemana [13]:

$$v(t + \Delta t) = v(t) + \left[\frac{1}{3}a(t + \Delta t) + \frac{5}{6}a(t) - \frac{1}{6}a(t - \Delta t) \right] \Delta t. \quad (20)$$

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \left[\frac{2}{3}a(t) - \frac{1}{6}a(t - \Delta t) \right] \Delta t^2, \quad (21)$$

Niewątpliwą zaletą tego algorytmu jest dokładniejsze odtworzenie trajektorii układu niż w poprzednich metodach. Niestety dzieje się to kosztem zapamiętania przyśpieszenia w dwóch poprzednich krokach, co w przypadku dużych układów znacznie spowalnia proces całkowania równań ruchu.

3.1.5 Metody typu predictor-corrector¹.

Drugą grupą metod stworzonych specjalnie z myślą o jak najdokładniejszym odtworzeniu trajektorii, są metody typu predictor-corrector. Idea działania tych metod opiera się na następującym schemacie [3]:

$$x^p(t + \Delta t) = x(t - \Delta t) + 2v(t)\Delta t, \quad (22)$$

$$v(t + \Delta t) = v(t) + [a(t + \Delta t) + a(t)]\Delta t, \quad (23)$$

$$x(t + \Delta t) = x(t) + [v(t + \Delta t) + v(t)]\Delta t. \quad (24)$$

W pierwszej kolejności obliczana jest przybliżona wartość położenia $x^p(t+\Delta t)$, która w dalszej kolejności służy do wyznaczenia przyspieszenia $a(t+\Delta t)$ (11) oraz prędkości $v(t+\Delta t)$. Znając prędkość w następnym kroku czasowym można wyznaczyć poprawioną wartość położenia $x(t+\Delta t)$. Korzystając z poprawionej wartości położenia ponownie można wyznaczyć przyspieszenie, prędkość, oraz poprawione położenie $x(t+\Delta t)$. Wykonując kilkakrotnie taką iterację otrzymujemy coraz lepsze przybliżenia położenia i prędkości w następnym kroku czasowym, jednak kosztem czasu potrzebnego do ich wyznaczenia.

3.2 Wybór algorytmu do symulacji dynamiki molekularnej.

W badanym problemie głównymi kryteriami wyboru algorytmu dynamiki molekularnej, były kwestie stabilności oraz prędkości obliczeń zapewniana przez algorytm. W celu wybrania najlepszej metody przeprowadziłem symulacje z użyciem każdego z omawianych algorytmów pod kątem postawionych wymagań. Okazało się, że dla badanego problemu metody Eulera oraz Verleta są niestabilne, więc w końcowej wersji programu nie znalazły się ich implementacje. Pozostałe metody są stabilne dla badanego procesu, jednak znacznie różnią się w kwestii prędkości obliczeń. W programie do symulacji zdecydowałem się użyć wyjątkowo stabilnej i jednocześnie jednej z najszybszych z omawianych metod, czyli algorytmu prędkościowego Verleta. Krok czasowy symulacji został przyjęty jako równy 0.01 jednostki czasowej. Dodatkowo w programie do symulacji zostały zaimplementowane algorytmy Leap-Frog oraz Beemana. O braku implementacji algorytmu predictor-corrector zadecydowała zbyt wolna prędkość obliczeń, co, w przypadku symulowania dużych układów, powodowało znaczne wydłużenie procesu symulacji, nie przynosząc wymiernych korzyści.

3.3 Równanie Langevina.

Wszystkie wyżej wymienione metody odnoszą się do rozwiązywania jednowymiarowych równań ruchu Newtona, jednak w prosty sposób stosują się one do rozwiązywania wektorowych równań ruchu, takich jak równanie Langevina (4). Dzieje się tak, ponieważ równanie Langevina jest równaniem różniczkowym liniowym ze względu na położenia, więc można je potraktować jako układ równań skalarnych.

¹ przewidująco korygujące

Istnienie członu losowego $\vec{\xi}_i$ w równaniu Langevina (4) implikuje potrzebę wykorzystania generatora liczb losowych. Taki generator powinien posiadać możliwie jak najlepsze własności statystyczne, tak, aby zagwarantować poprawną ewolucję układu. W następnym rozdziale przedstawię kilka historycznych i współcześnie stosowanych metod generowania liczb pseudolosowych, oraz przedstawię metodę wykorzystaną w programie do symulacji.

4. Generatory liczb pseudolosowych.

Zastosowanie liczb pseudolosowych pozwala na opisanie wielu zjawisk fizycznych, takich jak np. ruchy Browna. W symulacjach komputerowych znajdują one szerokie zastosowanie jako czynnik, który pozwala wprowadzić do układu parametry, których nie chcemy, lub nie możemy, wprowadzić ściśle. Koniecznością stało się stworzenie specjalnych algorytmów generowania liczb pseudolosowych. Wśród istniejących algorytmów wyróżnię dwie główne grupy, które postaram się opisać w następnych rozdziałach.

4.1 Generatory liniowe.

Z historycznego punktu widzenia, pierwszym generatorem liczb pseudolosowych jest algorytm zaproponowany przez J. von Neumanna w 1949 roku [16]. Metoda ta bazuje na następującej zależności, spełnionej dla liczb o parzystej długości d w systemie o podstawie P :

$$x_{n+1} = \left[x_n^2 P^{\frac{-d}{2}} \right] - P^d \left[x_n^2 P^{\frac{-3d}{2}} \right],$$

gdzie $[x]$ oznacza część całkowitą z liczby x .

Metoda ta nie jest zwykle stosowana ze względu na fakt otrzymywania od pewnego miejsca ciągu liczb, który jest krótkookresowy. Obecnie powszechnie stosuje się metody generowania liczb pseudolosowych, oparte na następującej zależności:

$$x_n = \left(b + \sum_{j=1}^k a_j x_{n-j} \right) (\text{mod } M), \quad (25)$$

gdzie a_i , b i x_i są liczbami całkowitymi z przedziału $[0, M-1]$. Okres tego typu generatorów wynosi $M-1$.

4.2 Generatory nieliniowe.

Ta grupa algorytmów powstała podczas prac, mających na celu poprawienie właściwości statystycznych generatorów liczb pseudolosowych. Pomysł opierał się na założeniu aby zależność pomiędzy kolejnymi wyrazami ciągu nie była liniowa. Jednym z przedstawicieli tej grupy algorytmów jest metoda, dana następującym wzorem:

$$x_n = (a(n + n_0) + b)^{-1} (\text{mod } M), \quad (26)$$

gdzie $n \in \{0, 1, 2, \dots\}$, a $x^{-1} \text{ mod } M$ oznacza taką liczbę, że:

$$x \cdot x^{-1} = 1 (\text{mod } M).$$

Okres powyższego generatora wynosi M dla każdego $a \in \{0,1,2,\dots,M\}$. Jak się okazuje, grupa algorytmów nieliniowych ma bardzo interesującą, z punktu widzenia obliczeń komputerowych, cechę. Mianowicie każda kolejna liczba losowa (26) nie zależy od poprzednio wygenerowanych liczb, więc metody należące do tej grupy idealnie nadają się do obliczeń równoległych.

4.3 Generowanie ciągu liczb o rozkładzie normalnym.

W modelowaniu wielu procesów fizycznych często zachodzi potrzeba wygenerowania ciągu liczb pseudolosowych, podlegających rozkładowi Gaussa (Maxwella). Istnieją dwie proste metody pozwalające wygenerować liczby o rozkładzie normalnym $N(0, 1)$:

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \quad (27)$$

Pierwsza z nich opiera się na centralnym twierdzeniu granicznym [19]. Załóżmy, że zmienne $\xi_1, \xi_2, \dots, \xi_1$ są niezależne i mają rozkład jednostajny na przedziale $[-1, 1]$. Wtedy zmienna:

$$\eta = \left(\sum_{i=1}^{12} \xi_i \right) - 6, \quad (28)$$

ma rozkład normalny $N(0,1)$. Ten sposób posłużył mi do generowania szumu stochastycznego w równaniu Langevina (4). Istnieje jeszcze drugi prosty sposób generowania ciągu liczb o rozkładzie normalnym $N(0,1)$. Podobnie, jeśli założymy, że zmienne losowe ξ_1, ξ_2 o rozkładzie jednostajnym na przedziale $[-1, 1]$ są niezależne, to wtedy zmienne:

$$\eta_1 = \sqrt{-2 \ln \xi_1} \cdot \cos(2\pi \xi_2), \quad (29)$$

$$\eta_2 = \sqrt{-2 \ln \xi_1} \cdot \sin(2\pi \xi_2), \quad (30)$$

są zmiennymi niezależnymi o rozkładzie normalnym $N(0, 1)$.

4.4 Test użytego generatora liczb losowych.

Wybierając generator liczb pseudolosowych zachodzi potrzeba jego przetestowania. Jedną z metod, którą można zastosować w tym celu, jest sprawdzenie współczynników korelacji pomiędzy kolejnymi elementami ciągu liczb losowych [7]. Obliczenie przy pomocy generatora wielowymiarowej całki metodą Monte-Carlo, może również być potraktowane jako dobry test dla generatora. Przykładem prostszego testu, jest sprawdzenie dla ciągu zmiennych losowych x_1, x_2, \dots, x_n , wartości średniej $\langle x \rangle$, wariancji $\text{var}(x)$ oraz wyższych momentów centralnych $m_k(x)$.

W symulacjach, w celu wprowadzenia losowej siły wymuszającej, użyłem generatora stworzonego przez Marsaglię [17]. Generator ten został szeroko przebadany pod kątem właściwości statystycznych i jest obecnie jednym z najlepszych generatorów liczb pseudolosowych. Okres tego generatora wynosi 2^{144} . Aby w prosty sposób sprawdzić czy użyty generator rzeczywiście generuje liczby losowe o rozkładzie jednostajnym na przedziale $[-1, 1]$, przeprowadziłem test, polegający na wygenerowaniu 10^5 , 10^6 i 10^7 liczb pseudolosowych i sprawdzeniu, czy odtwarzają one żądane wartości średniej oraz wariancji. Wartość średnia oraz wariancja dane są następującymi wzorami:

$$\langle x \rangle = \frac{1}{n} \sum_{i=1}^n x_i, \quad (31)$$

$$\text{var}(x) = \frac{1}{n} \sum_{i=1}^n (\langle x \rangle - x_i)^2. \quad (32)$$

gdzie x_i jest ciągiem liczb losowych. W wyniku testów otrzymałem następujące wartości:

Ilość liczb losowych	wartość średnia zmiennej losowej	wariancja zmiennej losowej
10^5	0.0005	1.0010
10^6	-0.0004	0.9990
10^7	-0.0002	1.0002
wartość teoretyczna	0.0000	1.0000

Tablica 2. Test generatora liczb pseudolosowych

Otrzymane wartości średniej oraz wariancji z dużą dokładnością zgadzają się z wartościami teoretycznymi, więc nie jest to powód aby odrzucić ten generator.

W badanym problemie zachodzi potrzeba stworzenia algorytmu generującego zmienne losowe o rozkładzie normalnym $N(0, 1)$. W celu wyznaczenia takiej zmiennej została zastosowana zależność (28), w której zmienne losowe ξ_i o rozkładzie normalnym na przedziale $N[-1, 1]$, zostały wygenerowane przy użyciu generatora Marsaglii. Kolejnym testem poprawności działania użytego generatora, było wyznaczenie gęstości prawdopodobieństwa dla 10^7 liczb losowych o rozkładzie normalnym $N(0,1)$. Przeprowadzony test przedstawia wykres (rys. 1). Z parametrów dopasowania krzywej Gaussa można wyciągnąć wniosek, że tak skonstruowana zmienna istotnie zachowuje się jak zmienna o rozkładzie normalnym $N(0,1)$, danym przez zależność (27).

Przeprowadzone testy nie podejmują tematu sprawdzenia korelacji generowanych liczb, jednak użyty generator był wcześniej wielokrotnie testowany pod tym kątem [7], z bardzo dobrym rezultatem.

Numeryczne algorytmy rozwiązywania równań ruchu oraz metody generowania liczb losowych, posłużą mi do przeanalizowania procesów zwijania i rozwijania się łańcuchów oddziaływujących potencjałem harmonicznym oraz Lennarda-Jonesa, poprzez symulacje komputerowe. Następny rozdział poświęcę omówieniu otrzymanych w symulacjach wyników.

5. Wyniki.

W niniejszym rozdziale przedstawię wyniki symulacji trójwymiarowego polimeru Lennarda-Jonesa. Przeprowadzone symulacje składały się z dwóch etapów. W pierwszym etapie został wygenerowany łańcuch atomów, który był poddawany symulacji w zakresie niskich temperatur ($T \sim 1$). Podczas symulacji został zaobserwowany proces zwijania się polimeru do stanu podstawowego. W drugiej części eksperymentu poddawano układy znajdujące się w stanie podstawowym symulacji w zakresie wysokich temperatur ($T \gg 1$). Podczas analizy otrzymanych trajektorii układów, został zaobserwowany proces rozwijania się polimeru.

5.1 Stan podstawowy.

Stanem podstawowym układu, zawierającego łańcuch atomów oddziaływujących przez potencjały (2) i (3), będę nazywał stan, w którym łańcuch będzie zwinięty w „kłębek” (rys. 2, rys. 3d, 3e). W takim stanie atomy tworzące łańcuch w przybliżeniu jednorodnie wypełniają sferycznie ograniczony obszar. Dla przypadku, w którym każdy atom oddziałuje z wszystkimi pozostałymi atomami łańcucha jedynie poprzez potencjał harmoniczny, istnieje możliwość analitycznego wyznaczenia stanu podstawowego [20]. Okazuje się że w takim przypadku, w stanie podstawowym atomy są równomiernie rozmieszczone na sferze. W badanym problemie nie zaobserwowałem takiego stanu., prawdopodobnie dlatego, że stan zwinięty jest stabilizowany przez oddziaływania Lennarda-Jonesa a cały polimer przez oddziaływanie harmoniczne.

5.2 Zwijanie się polimeru Lennarda-Jonesa.

Proces zwijania się polimeru Lennarda-Jonesa zachodzi głównie pod wpływem oddziaływań daleko zasięgowych. Przykładowy proces zwijania się łańcucha $N=100$ atomów, został przedstawiony na rysunku nr 3. Jak można zauważyć, proces ten zachodzi począwszy od końców łańcucha (rys. 3b), które posiadają jeden stopień swobody więcej, niż atomy znajdujące się w środku łańcucha. Podczas zwijania się łańcucha, pomiędzy końcami tworzą się stany pośrednie (rys. 3c). Ich czas „życia” nie jest jednak zbyt długi, gdyż w krótkim czasie zostają przyłączone do któregoś z końców łańcucha, lub rozpadają się pod wpływem zewnętrznej siły działającej na układ. Po pewnym czasie układ przechodzi do stanu zwiniętego (rys. 3d) i następnie zaczyna oscylować wokół położenia równowagi (rys. 3e).

Gdy w takiej sytuacji zmniejszyć temperaturę układu do $T = 0$, energia całkowita ustali się w pewnym minimum lokalnym. Dzieje się tak, gdyż tylko dla bardzo małych układów ($N < 10$), istnieje dobrze zdefiniowane globalne minimum energii. Dla większych układów występuje niejednorodne rozłożenie atomów wewnątrz sfery, co najprawdopodobniej jest spowodowane faktem, że sąsiadujące atomy w łańcuchu są ze sobą związane przez potencjał harmoniczny, co istotnie zmniejsza liczbę możliwych konformacji polimeru. Aby zaobserwować efekt korzystniejszego rozłożenia atomów w stanie podstawowym dla krótszych łańcuchów, zbadałem zależność stosunku ilości atomów na jednostkę objętości w funkcji długości łańcucha (rys. 4). W tym celu przeprowadziłem symulacje dla łańcuchów o długości od 50 do 500 atomów. Ilość atomów N na jednostkę objętości dla danej długości łańcucha (gęstość), dana wzorem:

$$\rho = \frac{N}{\frac{4}{3}\pi R^3}, \quad (33)$$

została uśredniona po trajektoriach dla różnych czasów symulacji. Promień R został wyznaczony przy użyciu metody (35) omówionej w dalszej części rozdziału. Z wykresu widać, że dla krótkich łańcuchów (50-100 atomów), średnia gęstość jest większa niż dla długich łańcuchów (200-500 atomów), co istotnie potwierdza postawioną wcześniej tezę. Dodatkowo można zauważyć, że dla długich łańcuchów średnia gęstość ustala się na mniej więcej stałym poziomie.

5.3 Rozwijanie się polimeru Lennarda-Jonesa.

Proces rozwijania się polimeru zachodzi pod wpływem zwiększającej się temperatury układu, która powoduje, że wzrasta amplituda zewnętrznej siły stochastycznej. Schemat procesu rozwijania się łańcucha $N=100$ cząstek, został przedstawiony na wykresie (rys. 5). Dla niskich temperatur (rys. 5a), widoczna jest jedynie zmiana kształtu układu, który nadal jest zwinięty w obszarze o podobnym rozmiarze jak dla układu w stanie podstawowym. W przedziale wyższych temperatur, ułożenie łańcucha nie jest ściśle określone, gdyż podczas ewolucji czasowej układ przyjmuje częściowo rozwiniętą formę pośrednią (rys. 5b-5e). Dopiero dla temperatur wyższych od pewnej temperatury granicznej układ preferuje stan rozwinięty (rys. 5f, 5g). Istnienie takich trzech obszarów, potwierdzę poprzez analizę własności dynamicznych polimeru, oraz zbadam ich zachowanie w funkcji długości łańcucha.

5.4 Promień polimeru.

Podczas rozwijania się łańcucha można zaobserwować zmiany parametrów fizycznych opisujących układ. Jednym z takich parametrów jest promień polimeru. Z analizy teoretycznej wiadomo, że dla błędzenia przypadkowego średnia odległość od punktu początkowego jest proporcjonalna do pierwiastka kwadratowego z liczby kroków:

$$\langle r \rangle \sim \sqrt{N}.$$

Podobnie, dla idealnego łańcucha atomów, średnia odległość dwóch końców łańcucha od siebie, jest proporcjonalna do pierwiastka z jego długości [2]:

$$\langle r \rangle = \langle \vec{r}_1 - \vec{r}_N \rangle = \sqrt{N}a, \quad (34)$$

gdzie a jest równowagową odległością pomiędzy atomami w łańcuchu, N jest ilością atomów w łańcuchu, a uśrednianie odbywa się po możliwych realizacjach polimeru. Warunek ten jest spełniony dla dużych czasów uśredniania w granicy wysokich temperatur. Taki sposób liczenia promienia, jako połowy wartości (34), został przyjęty w programie do symulacji polimerów Lennarda-Jonesa.

Powyższy sposób wydaje się być dobry. Ma on jednak pewną wadę, która skłoniła mnie do znalezienia innego sposobu wyznaczania promienia polimeru. W granicy niskich temperatur różnica odległości pomiędzy końcowymi węzłami łańcucha mało zmienia się w czasie, gdyż atomy utkwily w minimach potencjału a zewnętrzna siła losowa (proporcjonalna do temperatury) nie jest w stanie ich z tych minimów wybić. Wyobraźmy sobie teraz sytuację, w której, na skutek przejścia układu z wyższej temperatury, dwa końcowe węzły łańcucha są w zwiniętym polimerze położone na odległość jednego minimum potencjału. W takiej sytuacji obliczanie średniego promienia polimeru ze wzoru (34) traci sens, więc chciałbym zaproponować dwa inne sposoby wyznaczania promienia polimeru.

Pierwszy z nich polega na wyznaczeniu maksymalnej odległości pomiędzy dwoma węzłami łańcucha. Połowę takiej odległości można przyjąć jako maksymalny promień polimeru. Więc dla łańcucha o długości N , należy wyznaczyć maksymalną odległość pomiędzy jego węzłami, uśrednioną po trajektorii:

$$r_{MAX} = \frac{1}{2} \left\langle \max \left(\left| \vec{r}_i - \vec{r}_j \right| \right) \right\rangle, \quad (35)$$

gdzie $i \in \{0,1,2,\dots,N\}$, $j \in \{0,1,2,\dots,N\}$ a funkcja $\max(x_i)$ przyjmuje wartość największej liczby x_i . Tak zdefiniowany promień polimeru, uśredniony po możliwych konfiguracjach, powinien się zachowywać podobnie jak promień obliczony ze wzoru (34).

Drugi sposób opiera się na wyznaczeniu promienia „rozmycia” polimeru² [2]. Jest on zdefiniowany następująco:

$$r_D^2 = \left\langle \frac{1}{N} \sum_{i=1}^N (\vec{r}_{CM} - \vec{r}_i)^2 \right\rangle, \quad (36)$$

gdzie N jest liczbą cząstek, a uśrednianie podobnie odbywa się po możliwych konfiguracjach łańcucha. Z kolei \vec{r}_{CM} oznacza wektor środka masy:

$$\vec{r}_{CM} = \frac{1}{N} \sum_{i=1}^N \vec{r}_i.$$

Niewątpliwą zaletą wyznaczenia promienia „rozmycia” polimeru jest możliwość jego zdefiniowania dla różnych konformacji łańcucha, oraz zmierzenia jego wartości drogą eksperymentalną [18].

Na wykresach (rys. 6) została przedstawiona zależność promienia polimeru, obliczonego z zależności (34), (35) i (36), w funkcji temperatury, dla układu zawierającego 100 atomów. Wartości promieni zostały wyznaczone w trakcie symulacji, trwających $8 \cdot 10^5$ kroków czasowych, poprzez uśrednienie po realizacjach. Symulacje były przeprowadzane w stałej temperaturze, i za każdym razem zaczynały się od stanu rozwiniętego i dopiero w ich trakcie układ dochodził do odpowiedniego stanu końcowego. Wszystkie wartości promienia zostały znormalizowane do stałej, związanej ze wzorem (34), wynoszącej $0.5\sqrt{N}$. Przyjęcie takiej stałej normalizacyjnej również dla wartości obliczonych ze wzorów (35) i (36), wynika z faktu, że jedynie dla promienia wyznaczonego ze wzoru (34), można porównać otrzymane

² radius of gyration

w symulacjach wyniki z wartością teoretyczną. Jest to spowodowane tym, że nie są znane wartości oczekiwane zależności (35) i (36). Na rysunkach linią przerywaną została zaznaczona wartość 1.0.

Z wykresów widać, że wartość średniego promienia polimeru (rys. 6a) nie dąży do wartości 1.0 dla dużych temperatur. Jako że stała normalizacyjna została wyznaczona dla przypadku idealnego łańcucha, w którym nie działają siły daleko zasięgowe, trudno więc oczekiwać, że taka sytuacja istotnie będzie miała miejsce. Wykresy przedstawiające zależność maksymalnego promienia polimeru od temperatury (rys. 6b) oraz promienia „rozmycia” polimeru od temperatury (rys. 6c) zostały zaprezentowane w celu ich porównania z wartością średniego promienia polimeru (rys. 6a). Istotnie okazuje się, że dla badanego układu wartości te różnią się jedynie poprzez normalizację. Dodatkowo do danych eksperymentalnych zostały dopasowane funkcje sigmoidalne Boltzmana. Z parametrów ich dopasowania można wyznaczyć minimalny i maksymalny promień polimeru (parametry R_1 i R_2), które z dobrym przybliżeniem odpowiadają otrzymanym w symulacjach rezultatom. Drugim ważnym parametrem jest temperatura T_0 , w której łańcuch rozwija się najszybciej, która jednak niezbyt dobrze oddaje tę własność, co najprawdopodobniej jest spowodowane faktem arbitralnego przyjęcia funkcji dopasowania.

Analogiczne wykresy zostały zaprezentowane dla symulacji przeprowadzonej dla układu 500 atomów (rys. 7) oraz układu 1000 atomów (rys. 8), przeprowadzonych w stałej temperaturze. Punktem startowym symulacji był, inaczej niż w symulacjach układu 100 atomów, stan zwinięty. Wynikało to z dużej czasochłonności procesu przejścia układu 500 lub 1000 cząstek ze stanu rozwiniętego do stanu zwiniętego, co zaowocowało by dużą rozbieżnością wyników oraz dużym nakładem obliczeniowym takiego procesu. Liczba kroków czasowych wynosiła odpowiednio $3 \cdot 10^5$ dla układu 500 atomów i $1.5 \cdot 10^5$ dla układu 1000 atomów.

Podobnie jak dla układu 100 cząstek, również dla układu 500 cząstek wartość średniego promienia polimeru (rys. 7a) nie dąży w granicy wysokich temperatur do wartości 1.0. Dodatkowo można zauważyć, że wartości średniego promienia polimeru (rys. 7a, 8a) mają większe odchylenie od dopasowanej funkcji, niż w układzie 100 cząstek (rys. 6a). Jest to spowodowane tym, że większe układy potrzebują dłuższego czasu aby ustabilizować się w danej temperaturze, a przeprowadzone symulacje dla 500 i 1000 atomów były krótsze niż dla układu 100 atomów. Podobny efekt można zaobserwować dla parametrów R_1 i R_2 , które dla układów 500 i 1000 atomów znacznie różnią się od wyników otrzymanych w symulacjach. Fakt, że wartość średniego promienia (rys. 8a) dla układu 1000 atomów dąży asymptotycznie do wartości 1.0, może być spowodowana zbyt krótkim czasem przeprowadzania symulacji, a nie, jak mogło by się wydawać, zgodnością z twierdzeniem (34).

Jak się okazuje, wartości maksymalnego promienia (35) oraz promienia „rozmycia” polimeru (36) przedstawione na wykresach (rys. 7b-c, 8b-c), lepiej odtwarzają zachowanie się promienia polimeru dla krótkich czasów symulacji, jednak kosztem niemożności wyznaczenia ich prawidłowej normalizacji. Dla długich czasów symulacji wszystkie metody powinny prowadzić do takich samych wyników, co istotnie można zaobserwować dla układu 100 cząstek (rys. 6a-c).

5.5 Energia polimeru.

Analizując układ cząstek oddziaływujących przez pewien potencjał, w prosty sposób można zdefiniować energię układu. Energii kinetyczną można wyznaczyć z prędkości cząstek wchodzących w skład układu:

$$E_k = \frac{1}{N} \sum_{i=1}^N m_i \vec{V}_i^2. \quad (37)$$

Jako że w układzie wprowadzony jest zewnętrzny szum stochastyczny, to energia kinetyczna jest miarą średniej temperatury układu. Z zasady ekwipartycji energii wiadomo, że średnia energia kinetyczna układu jest proporcjonalna do temperatury:

$$\langle E_k \rangle = \frac{3}{2} N k_B T. \quad (38)$$

Na wykresach (rys 9) przedstawione są zależność energii kinetycznej na cząstkę od temperatury, dla układów 100, 500 i 1000 atomów. Wartości średniej energii kinetycznej układają się na prostej, czyli są liniowo zależne od temperatury. Parametry regresji liniowej pokazują, że współczynnik kierunkowy takiej prostej jest w przybliżeniu równy 1.5, co odpowiada temperaturze wprowadzonej przez szum stochastyczny, spełniającej twierdzenie o ekwipartycji energii (38).

Jeśli energię potencjalną układu cząstek zdefiniujemy następująco:

$$E_p = \sum_{i=1}^N \left[U_H(\vec{r}_{i+1} - \vec{r}_i) + U_H(\vec{r}_i - \vec{r}_{i-1}) + \sum_{j=1}^{j \neq i} (U_{LJ}(\vec{r}_i - \vec{r}_j)) \right],$$

gdzie potencjały U_H oraz U_{LJ} są zdefiniowane odpowiednio wzorami (2) i (3), to wtedy energia całkowita układu, wyrazi się jako suma energii kinetycznej i potencjalnej:

$$E = E_k + E_p. \quad (39)$$

Średnia energia całkowita na cząstkę w funkcji temperatury, dla układów 100, 500 i 1000 atomów, jest przedstawiona na wykresie (rys. 10). Na wykresach wyraźnie widać, że energia całkowita zależy liniowo od temperatury, przy czym inaczej dla niskich i wysokich temperatur. Temperaturę, w której zachodzi zmiana zachowania się energii, można powiązać ze stanem, w którym polimer rozwija się najszybciej. W dużej mierze wartości tej temperatury zgadzają z danymi z wykresów (rys. 6, 7, 8).

Poprzez fluktuacje energii całkowitej można wyrazić ciepło właściwe układu. Najprościej, z punktu widzenia symulacji komputerowych, jest zdefiniować ciepło właściwe poprzez wariancję energii całkowitej:

$$C_V = \frac{\text{var}(E)}{k_B T^2}. \quad (40)$$

Zależność ciepła właściwego na cząstkę od temperatury układu obrazuje wykres (rys. 11). Dla niskich temperatur można się spodziewać, że poprzez ciepło właściwe zaobserwujemy fakt

zmiany wartości promienia polimeru, w formie charakterystycznego piku, którego położenie ściśle wiąże się z temperaturą, w której polimer rozwija się najszybciej. Istotnie dla układu 100 atomów taki pik istnieje w temperaturze $T \cong 2$ (rys. 11a), która z dobrym przybliżeniem odpowiada zmianie w energii całkowitej (rys. 10). Dla pozostałych układów powstawanie piku nie zostało zaobserwowane, czego przyczyną był zbyt krótki czas symulacji. Dodatkowo dla niskich temperatur ($T \sim 1$) ciepło właściwe znacznie wzrasta co prawdopodobnie jest spowodowane fluktuacjami energii pochodzących z różniczkowania równań ruchu [7], których przyczynek może być większy od fluktuacji energii pochodzącej z zewnętrznej siły stochastycznej. Niestety poprzez symulacje komputerowe trudno jest odtworzyć ciepło właściwe w przedziale niskich temperatur ($T < 1$), gdyż wzór (40) jest silnie zależny od błędów wynikających z zaokrągleń. Dla wysokich temperatur ciepło właściwe dąży do klasycznej granicy równej 3.0, która jest wynikiem sumowania się przyczynków do ciepła właściwego pochodzących od energii potencjalnej i kinetycznej. Granica klasyczna została zaznaczona na wykresie linią przerywaną.

6. Dyskusja.

Szczególnie ciekawym zjawiskiem zaobserwowanym podczas symulacji, jest proces rozwijania się polimeru ze stanu podstawowego pod wpływem zwiększającej się temperatury (rys. 5). Proces rozwijania się polimeru można scharakteryzować poprzez przedstawienie, zależności promienia polimeru od temperatury (rys. 6, 7, 8). Za jego pomocą można wyznaczyć trzy charakterystyczne przedziały temperaturowe w których polimer znajduje się w różnych stanach. Pierwszy z nich odpowiada sytuacji, gdy temperatura, wprowadzona przez zewnętrzny szum, jest za niska, aby polimer mógł się rozwinąć. Drugi przedział to stan przejściowy pomiędzy stanem zwiniętym a rozwiniętym. Przy czym w niższych temperaturach, w tym przedziale temperatury, łańcuch preferuje stan zwinięty a w wyższych rozwinięty. Trzeci obszar odpowiada sytuacji gdy łańcuch jest w pełni rozwinięty. Takie obszary istnieją dla każdego z trzech testowanych układów, są tylko przesunięte względem siebie. Chciałbym w tym miejscu zaproponować interpretację faktu istnienia charakterystycznych temperatur, w których przebiega proces rozwijania się polimeru.

Podobnie jak dla procesu zwijania się polimeru, kluczową rolę w procesie rozwijania się, odgrywają końce łańcucha. Dla uproszczenia założmy, że znajdują się one na powierzchni zwiniętego łańcucha. W takiej sytuacji można sprawdzić, ilu najbliższych sąsiadów³ posiada pojedynczy atom. Analogicznie można postąpić dla atomów znajdujących się wewnątrz zwiniętego polimeru

Energia kinetyczna, potrzebna do pokonania energii związania tych atomów, jest dla każdego atomu równa wysokości bariery energetycznej pomiędzy minimum potencjału a punktem dalekim od minimum. Korzystając z twierdzenia o ekwipartycji energii (38), można oszacować jaka temperatura odpowiada tej energii kinetycznej.

Położenie atomu	Średnia liczba najbliższych sąsiadów	Energia kinetyczna potrzebna do wybicia ich z minimum potencjału	Odpowiadająca tej energii temperatura
powierzchnia	3	3.0	2.0
wnętrze	12	12.0	8.0

3. Temperatury charakterystyczne procesu rozwijania się polimeru Lennarda-Jonesa

Graniczną temperaturą „wiązącą” atomy na powierzchni, jest temperatura w której polimer zaczyna się rozwijać. Dlatego pierwszą temperaturę z tabeli 3, można powiązać z początkiem procesu rozwijania się polimeru. Stan rozwinięty odpowiada sytuacji, w której jak najmniej atomów jest związanych przez oddziaływanie Lennarda-Jonesa. Taka sytuacja będzie miała miejsce, gdy atomy wewnątrz zwiniętego polimeru będą miały tak dużą energię, że będą mogły się uwolnić od wiążących je oddziaływań. Stąd wynika, że druga temperatura z tabeli 3, wiąże się ze stanem, w którym polimer jest w pełni rozwinięty.

W porównaniu z danymi otrzymanymi w symulacjach, wydaje się, że niższa temperatura słabo zależy od długości łańcucha i dość dobrze zgadza się dla wszystkich symulowanych konfiguracji. Natomiast wyższa temperatura silnie zależy od liczby atomów w łańcuchu, gdyż dla polimeru składającego się ze 100 atomów, stan rozwinięty jest uzyskiwany dla temperatury $T \cong 4.5$, a dla łańcucha 1000 atomów dla $T \cong 8.0$. Przyczyną otrzymania tak

³ przez miano najbliższych sąsiadów będę oznaczał wszystkie te atomy, które leżą w odległości $r \cong 1.0$, równej równowagowej odległości pomiędzy atomami

różnych wyników, może być fakt, że do przejścia do stanu rozwiniętego nie jest konieczne uwolnienie się wszystkich atomów od oddziaływań Lennarda-Jonesa (rys. 5c-e). Dodatkowo należy zauważyć, że w wyższych temperaturach liczba najbliższych sąsiadów powinna być mniejsza na skutek stopniowego rozwijania się polimeru. Tak więc druga temperatura może być znacznie przeszacowana.

7. Podsumowanie.

Przy użyciu technik symulacji komputerowych zostało zbadane zjawisko zwijania się łańcuchów polimerowych do stanu podstawowego. Okazało się, że stanem podstawowym takiego łańcucha, jest układ, w którym atomy jednorodnie wypełniają w przybliżeniu sferyczny obszar. Stan taki tworzy się w wyniku wprowadzenia do układu oddziaływań daleko zasięgowych.

Zmiany temperatury układu, zawierającego polimer w stanie podstawowym, powodują jego stopniowe rozwijanie się. Taki proces został przebadany pod kątem zachowania się parametrów dynamicznych, oraz termodynamicznych układu. Otrzymane wyniki są jakościowo zgodne z otrzymanymi w pracy [18]. Okazuje się, że można w przybliżeniu oszacować przedział temperatur, w którym będzie zachodził proces rozwijania się łańcucha. Dodatkowo okazuje się, że wartość uśrednionego promienia polimeru zachowuje się inaczej niż w przypadku błędzenia przypadkowego, co prawdopodobnie również jest efektem wprowadzenia oddziaływań daleko zasięgowych. Z analizy zachowania się energii całkowitej oraz ciepła właściwego układu, można w przybliżeniu oszacować temperaturę w jakiej polimer rozwija się najszybciej.

Efektom zbadania dynamicznych właściwości polimerów, jest stworzenie pakietu programów, służących do przeprowadzania symulacji metodami dynamiki molekularnej, oraz analizy i wizualizacji danych eksperymentalnych.

A. Omówienie pakietu programów do symulacji dynamiki molekularnej.

W celu przeprowadzenia symulacji oraz późniejszej analizy otrzymanych danych, został stworzony specjalny pakiet programów. W skład tego pakietu wchodzi następujące programy:

- Program do symulacji polimerów,
- Program do wizualizacji danych,
- Program do analizy danych,
- Program do generowania polimerów.

W kolejnych rozdziałach omówię wszystkie programy, z położeniem nacisku na przedstawienie podstawowych własności każdego z nich. Szczegółowe omówienie wszystkich programów, wraz ze sposobem ich użytkowania, znajduje się w odpowiednich katalogach na załączonej płycie CD.

A.1 Program do symulacji polimerów Lennarda-Jonesa.

Przeprowadzenie symulacji polimeru Lennarda-Jonesa wymaga ogromnego nakładu obliczeniowego. Przy tworzeniu programu do symulacji należało przyjąć pewne założenia, dzięki którym można lepiej określić pożądane cechy programu. Do najważniejszych cech wyboru implementacji należała kwestia prędkości obliczeń oraz przenośności programu pod inne platformy systemowe. Kwestią stał się więc wybór odpowiedniego języka programowania, zdolnego sprostać tym wymaganiom. Z historycznego punktu widzenia, najlepszym wyborem byłby FORTRAN, który jest specjalnie zoptymalizowany pod kątem obliczeń numerycznych oraz którego kod jest przenośny pomiędzy wszystkimi środowiskami uniksowymi. Jednak współcześnie coraz częściej w obliczeniach numerycznych stosuje się język C, który posiada swoje implementacje na **wszystkich** obecnie stosowanych platformach a w szybkości obliczeń nie ustępuje FORTRANowi.

Do stworzenia programu wybrałem obiektową wersję języka C – język C++ [5]. Poniżej znajduje się spis plików programu, wraz z opisem przeznaczenia i najważniejszych metod zawartych w każdym z nich:

Generator.cpp – zawiera klasę *Generator* odpowiedzialną za generowanie liczb pseudolosowych z przedziału $[0, 1]$ metodą przedstawioną przez Marsiglię:

- metoda *init* – inicjalizacja generatora,
- metoda *makeVector* – stworzenie losowego N-wymiarowego wektora,

Vector.cpp – zawiera klasę *Vector* służącą do przeprowadzania operacji na wektorach (dodawanie, odejmowanie, dzielenie, itd.),

Simulation.cpp – zawiera klasę *Simulation* w której znajduje się między innymi implementacja algorytmów dynamiki molekularnej oraz kod sterujący przeprowadzaniem symulacji:

- metoda *simulate* – przeprowadza symulację dla zadanej temperatury,
- metoda *calculateForces* – oblicza siły działające na każdy węzeł łańcucha,

- metoda *calculateNoises* – oblicza szum dla członu losowego w równaniu Langevina,
- metoda *doMDStep* – przeprowadza pojedynczy krok symulacji dynamiki molekularnej.

Polymer.cpp - główny plik programu:

- procedura *performSimulation* – przeprowadza symulację dla wszystkich temperatur zadanych w pliku konfiguracyjnym,
- procedura *loadParametersFromFile* – wczytuje parametry symulacji z pliku.

Dodatkowo w skład programu wchodzi plik *Makefile*, służący do automatycznej kompilacji programu pod systemami uniksowymi, oraz plik *simin* zawierający dane przeprowadzanej symulacji. Pełny opis zawartości plików źródłowych znajduje się w komentarzach dołączonych do kodu źródłowego, oraz w pliku README umieszczonym w katalogu głównym programu (folder Programy\Simulation).

A.2 Program do wizualizacji danych.

Program do wizualizacji danych został stworzony dla lepszego zobrazowania otrzymanych w trakcie symulacji wyników. Podobnie jak w przypadku programu do symulacji, ważną cechą, narzucającą wybór języka programowania, była kwestia przenośności programu pomiędzy platformami. W przypadku bibliotek graficznych służących do operacji w przestrzeni trójwymiarowej, jedynym możliwym wyborem jest zastosowanie biblioteki *Open Graphics Library* (tzw *OpenGL*), która posiada swoje implementacje na wszystkich platformach. Projektując program zdecydowałem się na zastosowanie języka Java [6], którego implementacja bibliotek OpenGL (tzw Java3D lub Java OpenGL), jest w pełni obiektowa, co pozwala w prosty i przejrzysty sposób skonstruować kod programu.

Stworzony program pozwala na manipulowanie (skalowanie, obracanie) oglądanym łańcuchem polimerowym, prześledzenie trajektorii układu, oraz zapisanie wybranych fragmentów ewolucji czasowej w postaci pliku graficznego (w formacie JPEG) lub filmu (w formacie Microsoft AVI lub Quicktime).

Pełny opis programu znajduje się w pliku README umieszczonym w głównym katalogu programu (Programy/PolymerVisualizer). Teraz przedstawię wybrane klasy użyte w programie wraz z ich zastosowaniem:

PolymerVisualizer.java – zawiera główną klasę programu, odpowiedzialną za stworzenie interfejsu graficznego oraz komunikację z użytkownikiem,

PolymerGeometry.java – odpowiada za generowanie trójwymiarowego modelu polimeru,

PolymerTrajectoryReader.java – wczytuje plik zawierający trajektorię układu, wygenerowany przez program do symulacji dynamiki molekularnej,

ImageWriter.java – odpowiada za zapisywanie wymodelowanego polimeru jako pliku graficznego,

MovieWriter.java – klasa służąca do zapisywania sekwencji wymodelowanych polimerów (kolejnych kroków trajektorii) jako animacji.

A.3 Program do analizy danych.

Efektem przeprowadzenia symulacji polimeru Lennarda-Jonesa jest wygenerowanie plików zawierających trajektorię układu oraz parametry termodynamiczne każdego kroku trajektorii. W celu przeanalizowania otrzymanych danych stworzony został program w języku Java. Program ten składa się z dwóch części. Pierwsza część służy do analizy parametrów termodynamicznych, czyli, między innymi, do policzenia ich wartości średnich. Program składa się z następujących plików:

- *PolymerParametersAnalyzer.java* – główna część programu odpowiedzialna za analizę parametrów termodynamicznych,
- *PolymerParametersFileReader.java* – moduł programu, który wczytuje parametry termodynamiczne z pliku,
- *PolymerData.java* – moduł opisujący parametry termodynamiczne pojedynczego kroku symulacji.

Drugi program analizuje trajektorię układu i oblicza maksymalny oraz rozmyty promień polimeru. W jego skład wchodzi następujące moduły:

- *PolymerTrajectoryAnalyzer.java* – główna część programu przeprowadzająca analizę trajektorii układu,
- *PolymerTrajectoryFileReader.java* – moduł programu, który wczytuje trajektorię układu z pliku wygenerowanego przez program do symulacji,
- *PolymerData.java* – zawiera klasę opisującą pojedynczy krok trajektorii.

Oba programy znajdują się na dołączonej płycie CD w katalogach, odpowiednio, *Programy/PolymerParametersAnalyzer* oraz *Programy/PolymerTrajectoryAnalyzer*. Każdy z programów jest dokładnie opisany w odpowiednim pliku README.

A.4 Program do generowania łańcuchów polimerowych.

Ostatni stworzony program służy do generowania układów polimerowych. Program umożliwia wygenerowanie układu atomów równomiernie rozłożonych na prostej, okręgu lub spirali. Wygenerowane przy pomocy tego programu układy mogą posłużyć jako punkt startowy symulacji. Można również przedstawić wygenerowaną strukturę w omawianym wcześniej programie do wizualizacji. Program zawiera następujące moduły:

- *PolymerGenerator.java* – główny plik programu odpowiedzialny za wywołanie jednego z modułów tworzących łańcuch atomów,
- *AbstractGenerator.java* – klasa opisująca zachowanie się wszystkich generatorów,

- *PolymerLineGenerator.java* – generuje łańcuch atomów rozłożony równomiernie na odcinku,
- *PolymerCircleGenerator.java* – generuje łańcuch atomów rozłożony równomiernie na okręgu,
- *PolymerSpiralGenerator.java* – generuje łańcuch atomów rozłożony równomiernie na spirali.

Program znajduje się w katalogu *Programy/PolymerGenerator* na dołączonej płycie CD.

B. Jednostki zredukowane.

W symulacjach używane są jednostki zredukowane, które są zadane przez parametry potencjału Lennarda-Jonesa.

Nazwa jednostki	Symbol jednostki	Odpowiednik w układzie SI
czas	t^*	$t\sqrt{\frac{\varepsilon}{m\sigma^2}}$
długość	r^*	$\frac{r}{\sigma}$
energia	E^*	$\frac{E}{\varepsilon}$
temperatura	T^*	$\frac{k_B T}{\varepsilon}$
siła	F^*	$\frac{E\sigma}{\varepsilon}$
prędkość	V^*	$V\sqrt{\frac{m}{\varepsilon}}$

Spis tablic

1. Parametry symulacji	7
2. Test generatora liczb pseudolosowych	16
3. Temperatury charakterystyczne procesu rozwijania się polimeru Lennarda-Jonesa	23

Spis rysunków

1. Rozkład normalny $N(0, 1)$	33
2. Stan podstawowy polimeru Lennarda-Jonesa	34
3. Schemat procesu zwijania się polimeru	35
4. Zależność ilości atomów w jednostce objętości od ilości atomów w łańcuchu	36
5. Schemat procesu rozwijania się polimeru pod wpływem wzrastającej temperatury	37
6. Promień polimeru w funkcji temperatury dla łańcucha 100 atomów	38
7. Promień polimeru w funkcji temperatury dla łańcucha 500 atomów	39
8. Promień polimeru w funkcji temperatury dla łańcucha 1000 atomów	40
9. Średnia energia kinetyczna na cząstkę w funkcji temperatury	41
10. Średnia energia całkowita na cząstkę w funkcji temperatury	42
11. Średnie ciepło właściwe na cząstkę w funkcji temperatury	43

Bibliografia

- [1] F. A. Oliveira, P. L. Taylor J. Chem. Phys. **101** (1994) 10118
- [2] M. Doi. *Introduction to Polymer Physics*. Clarendon Press (1996)
- [3] M. P. Allen, D. J. Tildesley *Computer Simulation of Liquids*. Clarendon Press, Oxford (1987)
- [4] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling *Numerical Recipes in C*. Cambridge University Press (1988)
- [5] B. Eckel *Thinking in C++ 2nd Edition*. Prentice Hall (2000)
- [6] B. Eckel *Thinking in Java 3rd Edition*. Prentice Hall (2002)
- [7] M. Cieśla *Symulacje jednowymiarowych łańcuchów poddanych działaniu sił Langevina*. Praca magisterska, Instytut Fizyki, Uniwersytet Jagielloński (1998)
- [8] N. G. van Kampen *Statistical Processes In Physics and Chemistry*. Elsevier Science Publishers B.V. (1981)
- [9] D. Frenkel, B. Smit *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press (2002).

- [10] M. Toda, R. Kubo, N. Saitô *Fizyka Statystyczna*. PWN Warszawa (1991)
- [11] L. D. Landau, E. M. Lifszyc *Fizyka Statystyczna*. PWN Warszawa (1976)
- [12] L. Verlet Phys. Rev. **159** (1967) 98
- [13] D. Beeman J. Comput. Phys. **20** (1976) 130
- [14] R. W. Hockney Methods Comput. Phys. **9** (1970) 136
- [15] W. C. Swope, H. C. Andersen, P. H. Berens, K. R. Wilson *A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters*. J. Chem. Phys. **76** (1982) 637
- [16] P. C. Hammer Nat. Bur. Stand. Appl. Math. Ser **12** (1951) 33
- [17] G. Marsaglia, A. Zaman. Florida State University Report: FSU-SCRI-87-50 (1987)
- [18] S. T. Sun, I. Nishio, G. Swislow, J. Chem. Phys. **73** (1980) 5971
- [19] R. Nowak, *Statystyka dla fizyków*. PWN Warszawa (2002)
- [20] M. Jeżabek, K. Zalewski, J. Wosiek *Quasi-classical vacua for gauge groups $SU(N \rightarrow \infty)$* . Act. Phys. Pol. **B12** (1981) 597

