

Rozdział 1

Wstęp

Witamy Państwa na ćwiczeniach z Modelowania Molekularnego 2.

Zacznijmy od kwestii formalnych. Skrypt zawiera przede wszystkim opis trzech programów zaliczeniowych. Na ćwiczeniach będą one omawiane i oceniane. Przyśpieszenie do egzaminu jest możliwe jedynie po zaliczeniu ćwiczeń. Warunkiem dostatecznym i koniecznym do uzyskania zaliczenia ćwiczeń jest oddanie samodzielnie napisanych i działających programów zaliczeniowych, oraz przedstawienie wyników. W przypadku oddania obydwu programów w wyznaczonych terminach ocena z przedmiotu zostaje podniesiona o pół stopnia (gdy oceną z egzaminu nie jest ani 2.0, ani 5.0). Można więc uznać, że ocena z ćwiczeń jest „zero-jedynkowa”. Zachęcamy Państwa do podjęcia trudu i uzyskania z ćwiczeń „jedynek”!

Uczestników kursu (czyli głównych odbiorców tego skryptu) cechuje duże zróżnicowanie zarówno pod względem umiejętności programistycznych, jak i pod względem zrozumienia zagadnień fizycznych. Jednak od wszystkich studentów oczekujemy znajomości podstawowych algorytmów, pojęć programowania obiektowego, jak również znajomości obiektowego języka programowania (w domyśle Javy). Opracowując programy zaliczeniowe braliśmy pod uwagę osoby, które nie miały jeszcze okazji nabyć swobody w posługiwaniu się podstawowymi narzędziami programistycznymi i chcemy zapewnić, że choć przewidziany na ćwiczenia materiał nie należy do najprostszych, to trudności techniczne będzie można pokonać systematyczną pracą i konsultacjami z ćwiczeniowcem.

Na początku zajmiemy się dynamiką molekularną (*Molecular Dynamics*, MD). **Pierwszy program zaliczeniowy** jest implementacją prostej dynamiki molekularnej. Aby uprościć początkowe rozważania ograniczymy się do układu, składającego się z pojedynczej cząstki, z opisem 1-wymiarowym. Cząstka uwięziona będzie w studni z dwoma minimami, z barierą między nimi. Na takim układzie będziemy testować różne algorytmy całkujące (Verlet, Velocity Verlet, Leapfrog i Beeman). Będzie to układ modelowy, który będziemy w trakcie ćwiczeń rozwijać, umożliwiając zastosowanie:

1. dowolnego pola siłowego;
2. dowolnego algorytmu całkującego;
3. dowolnej liczby atomów tworzących układ;
4. opisu 1-, 2-, bądź 3-wymiarowego;

Następnie zajmiemy się metodą Monte Carlo (MC), a w szczególności - algorytmem Metropolis'a. W drugim programie zaliczeniowym modelować będziemy układ *w kontakcie z termostatem*¹. Związane z tym tematem zagadnienia fizyczne są dość złożone i do pełnego zrozumienia wymagają znajomości mechaniki statystycznej i termodynamiki. Mamy jednak nadzieję, że zainteresujemy studentów na tyle, by sami zechcieli pogłębiać swą wiedzę na kursach poświęconych tej tematyce.

Drugi program zaliczeniowy będzie umożliwiał przeprowadzenie symulacji Monte Carlo z symulowanym wyżarzaniem (*simulated annealing*) i wymianą replik (*Replica Exchange Monte Carlo, REMC*). Będziemy przeprowadzać symulacje dla bardzo prostego modelu białka - *modelu HP*. Powinno się Państwu spodobać to, że polimer związa się do konfiguracji o niskiej (jak nie najniższej) energii i możnaby ulec wrażeniu, że temu właśnie służy MC. Jednak odszukanie globalnego minimum energii potencjalnej dla modelu HP jest możliwe zwyczajnie poprzez wygenerowanie wszystkich możliwych konfiguracji (co również uczynimy) i wybranie tej o najniższej energii. Przekonają się Państwo, że istotą metody Monte Carlo jest próbkowanie zbioru możliwych konfiguracji według pewnego rozkładu prawdopodobieństwa. W przypadku algorytmu Metropolis'a jest to rozkład Boltzmanna (opisany wzorem (3.1)).

Kończąc, chcieliśmy szczególnie mocno podkreślić, że choć główną trudnością z jaką będą musieli zmierzyć się studenci będzie realizacja projektów programistycznych, to istotą proponowanych ćwiczeń jest przedstawienie subtelności fizycznych, wszytych w symulacje komputerowe układów molekularnych.

1.1 Dynamika molekularna

Dynamika molekularna może się Państwu wydawać bliska, choćby ze względu na wcześniejszy kurs, *Modelowanie Molekularne I*, w ramach którego poznali Państwo narzędzie NAMD do przeprowadzania tego typu symulacji. Częsteczką białka traktowana jest jak układ klasyczny, dla którego dynamika molekularna generuje ciąg *mikrostanów* wykorzystując algorytmy całkujące równania ruchu. Z tak uzyskanego zbioru mikrostanów wyznacza się rozmaite wielkości fizyczne, np. średnią energię układu.

Przez **mikrostan** rozumieć będziemy zbiór parametrów w pełni opisujących jedną realizację układu. W przypadku białka w ujęciu klasycznym mikrostan określają pędy i położenia wszystkich atomów. Przez **konfigurację** rozumieć będziemy

¹Mówimy też: *w kontakcie termicznym z otoczeniem*, bądź: *w stałej temperaturze*, bądź: *w zespole kanonicznym*, bądź: *w zespole NVT*.

jedną realizację, opisaną położeniami atomów, bez pędów. Zbiór wszystkich mikrostanów nazywać będziemy **przestrzenią fazową**. Analogicznie, przez **przestrzeń konfiguracyjną** rozumiemy zbiór możliwych konfiguracji układu.

Wróćmy do ciągu mikrostanów jako wyniku dynamiki molekularnej. Ciąg mikrostanów nazywać będziemy **trajektorią**. Jest to w pewnym sensie film z życia naszego modelu białka. Jednak trajektoria to coś więcej - to próba przestrzeni konfiguracyjnej (czyli zbiór konfiguracji), dzięki której wyznaczamy estymatory **wielkości makroskopowych** takie jak: średnia energia, średnia kwadratu energii i inne. Często stosowanym założeniem jest to, że tego typu estymatory są zgodne z wartościami mierzonymi w doświadczeniu, jednak tylko pod warunkiem, że uzyskana próba mikrostanów jest zgodna z rozkładem Boltzmanna.

Rodzi się pokusa, aby ostateczną weryfikacją tego, czy symulacja przebiegła poprawnie stanowił ogląd filmu uzyskanego z dynamiki molekularnej. Wówczas, o ile białko zachowuje się zgodnie z naszymi intuicjami, symulacja jest „dobra”, a wnioski wyniesione z symulacji - poprawne. Jednak czy nasze intuicje dotyczące białek są słuszne?

Przykładowo, często spotykana intuicja mówi, że *białko zwiija się do stanu natywnego, któremu odpowiada globalne minimum energii potencjalnej*. Stan natywny to postać białka, w której występuje ono w organizmie i w której może spełniać swoją biologiczną funkcję. Jednak skąd pomysł, że jest to jedna, konkretna konfiguracja, ta, dla której uzyskujemy globalne minimum energii potencjalnej?

Z drugiej strony, inna intuicja podpowiada nam, że białko wykonuje chaotyczne ruchy, niezbędne do wykonania powierzonych mu zadań. Jednak ruchy te, jakkolwiek „małe”, wykluczają możliwość, by białko przebywało w jednej (ustalonej przez minimum energii potencjalnej) konfiguracji.

Intuicje te stoją ze sobą w sprzeczności.

W jaki więc sposób białko realizuje swój stan natywny? Skoro nie utrzymuje jednej konfiguracji, to jakie konfiguracje przyjmuje? Które konfiguracje przyjmuje częściej, a które rzadziej? Co jest przyczyną takiego stanu rzeczy?

Na tych ćwiczeniach przyjmujemy założenie, że stan natywny białka jest dobrze opisany rozkładem Boltzmanna². Tzn. układ przyjmuje konfiguracje z pewnym prawdopodobieństwem, określonym przez rozkład Boltzmanna (więcej w rozdziale 3.).

²Liczmy, że uda się ten (teoretyczny) wynik wyprowadzić, ale istotą drugiego programu zaliczeniowego jest badanie konsekwencji tego założenia.

1.2 Metoda Monte Carlo

W zasadzie dobrze przeprowadzona symulacja dostarcza nam „dobrej” próby przestrzeni konfiguracyjnej. Dobrej, ale nie tylko dlatego, że obejrzawszy trajektorię stwierdzamy, że białko wykonuje chaotyczne ruchy, które wydają się „sensowne”. Dobre próbkowanie to próbkowanie zgodne z rozkładem Boltzmanna³. Dopiero dla tak wyznaczonej próby możemy wyznaczyć estymatory wielkości makroskopowych białka, które następnie jest sens porównywać z tymi wyznaczonymi doświadczalnie. Jakże w takim razie próbkowanie byłoby złe i prowadziło do błędnie wyznaczonych własności układu? Przykładowo, dla dynamiki liczącej sobie 10^6 kroków moglibyśmy wyprodukować 10^6 identycznych kopii pewnej wybranej konfiguracji. Bądź: 10^6 konfiguracji z pobliża globalnego minimum, losowane wg pewnego rozkładu innego niż rozkład Boltzmanna⁴.

Zanim przybliżymy Państwu metodę Monte Carlo, chcemy postawić pewien problem. Sama metoda MC z algorytmem Metropolisa opisana jest w rozdziale 4., a najlepiej zaznajomią się z nią Państwo, implementując drugi program zaliczeniowy.

W poprzednim paragrafie wspomnieliśmy o stanie natywnym białka. Warto wiedzieć, że istnieją tzw. *natywnie nieustrukturalizowane białka* (*natively unfolded proteins*). Biolog strukturalny może określić białka z tej klasy jako „nie mające ustalonej struktury”. Białka te natywnie przyjmują bardzo wiele znacznie różniących się między sobą konfiguracji. Nawet gdyby jakieś białko tego typu udało się skryształizować i określić strukturę jednego z mikrostanów, to ciężko oczekiwać, by o własnościach tego białka można było wnioskować na podstawie tej jednej konfiguracji. Szacuje się, że blisko 30% eukariotycznych białek zalicza się do tej klasy, bądź zawiera regiony o nieokreślonej strukturze [?].

Co mogłoby być przyczyną tego, że białko „nie chce przebywać” w pobliżu globalnego minimum energii potencjalnej? Innymi słowy: co sprawia, że białko zasługuje na miano natywnie nieustrukturalizowanego? Czy to, że takie minimum nie istnieje? Być może białko nie jest w stanie go osiągnąć? Czy może posiada wiele konfiguracji o niskiej energii, ale żadna z nich nie „przeważa” w istotny sposób? Odpowiedzi na te pytania uzyskają Państwo implementując metodę Monte Carlo z algorytmem Metropolisa w ramach drugiego programu zaliczeniowego.

³Przynajmniej przy podjętym na tych ćwiczeniach założeniu. Niewykluczone, że przyszłe zastosowania symulacji komputerowych rozszerzą ten „paradygmat”, jednak zasady rządzące przeprowadzaniem takich symulacji pozostaną te same, z dokładnością do rozkładu prawdopodobieństwa, jakim będzie opisany układ.

⁴Na ćwiczeniach zgłębimy ten przykład i zobaczymy jak (ewentualnie) wykorzystać dane uzyskane z drugiego podejścia.

Rozdział 2

Pierwszy program zaliczeniowy

Wraz ze skryptem otrzymali Państwo szkic programu do przeprowadzania dynamiki molekularnej. Pierwsze laboratoria poświęcimy omówieniu klas zdefiniowanych w tym programie. Jednak zachęcamy Państwa do samodzielnego zaprojektowania i implementacji programu.

2.1 Zarys programu

Będziemy zajmować się klasycznym opisem układu, w którym ewolucję i -tego atomu opisuje II zasada dynamiki Newtona:

$$\ddot{\mathbf{r}}_i = \frac{1}{m_i} \mathbf{F}_i, \quad (2.1)$$

gdzie m_i jest masą, $\ddot{\mathbf{r}}_i$ przyspieszeniem, zaś \mathbf{F}_i - siłą działającą na ten atom.

Znane Państwu dobrze z liceum równanie (2.1) jest opisem układu izolowanego, w którym położenia atomów wyrażone są we współrzędnych kartezjańskich. W bardziej zaawansowanych (i w praktyce: częściej stosowanych) podejściach układ opisany jest współrzędnymi uogólnionymi, a ewolucję układu wyrażają równanie Eulera-Lagrange'a, bądź równania Hamiltona. Jednak dużo ważniejszą uwagą jest ta, że równanie (2.1) opisuje układ izolowany¹, natomiast białko takim układem *NIE* jest.

W pierwszym programie zaliczeniowym ograniczymy się do układów izolowanych. Na ćwiczeniach postaramy się przybliżyć Państwu tematykę związaną z modelowaniem układów w kontakcie termicznym z otoczeniem.

W celu uzyskania trajektorii wszystkich atomów, $\mathbf{r}(t)$, należałoby rozwiązać układ równań różniczkowych II stopnia - równań (2.1) - co w przypadku podejścia analitycznego jest zazwyczaj bardzo złożonym problemem. Natomiast numeryczna procedura jest stosunkowo prosta i przedstawia się następująco (zapożyczone z [?]):

¹Nie wymieniający z otoczeniem ani energii, ani materii. W takim układzie zachowana jest energia, tzn. $E_{cakowita}(t) = const.$

PROSTA-DYNAMIKA-MOLEKULARNA

```

1   $U \leftarrow$  INICJALIZACJA
2   $\triangleright$  Zainicjalizuj układ  $U$ 
3   $t \leftarrow 0$ 
4  while  $t < t_{max}$ 
5      do
6          PROCEDURA-WYZNACZAJĄCA-SIŁY( $U$ )
7          PROCEDURA-CĄKUJĄCA( $U$ )
8          ZAPAMIĘTAJ-STATYSTYKI( $U$ )
9           $t \leftarrow t + 1$ 
10 WYPISZ-STATYSTYKI-DO-PLIKU

```

Analog powyższego znajdą Państwo w dołączonych do skryptu plikach ze szkicem programu.

2.2 Treść pierwszego programu zaliczeniowego

Zaimplementuj procedurę przeprowadzającą prostą dynamikę molekularną, która wykorzystywać będzie dowolny potencjał i dowolny algorytm całkujący (w szczególności - potencjały i algorytmy całkujące omówione dalej).

Symulację ograniczymy do przypadku 1-wymiarowego, ale z dowolną liczbą atomów. Położenie, prędkość i przyspieszenie atomu określać będą obiekty klasy `Vec` z metodami analogicznymi do operacji wykonywanych na wektorach.

Program powinien pobierać informacje dotyczące:

1. używanego potencjału,
2. algorytmu całkującego,
3. liczby atomów,
4. liczby kroków symulacji,
5. długości kroku czasowego,

z linii komend (argument `args` metody `main`), bądź z pliku.

Początkowo wszystkie atomy będą miały zerową prędkość. Położenia początkowe program może wybierać drogą losową, bądź według prostego algorytmu Państwa autorstwa (np. poukładane na osi x , w odległości 1 między kolejnymi atomami). Dla uproszczenia rozważmy przypadek, w którym wszystkie atomy mają tę samą masę równą 1.

Jako wynik program powinien zwracać trajektorię układu (generować plik `xyz`), oraz zależność energii: potencjalnej, kinetycznej i całkowitej, od czasu (plik `csv`). Symulacja powinna być „poprawna” w tym sensie, że energia układu powinna utrzymywać się na (względnie) stałym poziomie.

Na omówienie potencjałów i algorytmów całkujących poświęcamy resztę tego rozdziału, jak również ćwiczenia i laboratoria.

2.3 Energia potencjalna

Zakładamy, że energia potencjalna jest funkcją (jedynie) położenia atomów (czyli konfiguracji układu), $U = U(\mathbf{r})$, a siła działająca na i -ty atom wyraża się następująco:

$$\mathbf{F}_i = -\frac{\partial U}{\partial \mathbf{r}_i}$$

Zatem przyjmujemy założenie, że funkcja energii potencjalnej jest różniczkowalna.

2.3.1 Miękkie ścianki

Najprostszy potencjał gwarantujący, że układ nam „nie ucieknie” to potencjał imitujący „miękkie ścianki”. Energia potencjalna i -tego atomu zależy od jego odległości od początku układu współrzędnych:

$$U(\mathbf{r}_i) = \begin{cases} 0 & , \text{ jeżeli } r_i < L \\ \frac{1}{2}f(L - \|\mathbf{r}_i\|)^2 & , \text{ wpp} \end{cases}$$

gdzie f to stała determinująca „twardość” ścianek, zaś L - rozmiar wnęki o energii potencjalnej zero.

2.3.2 Potencjał „minimum-bariera-minimum”

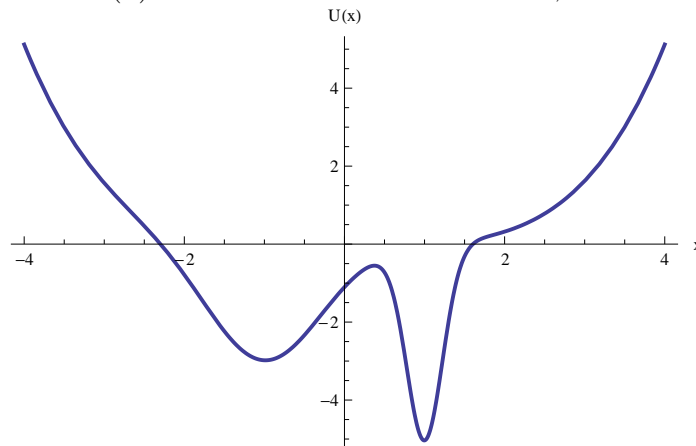
Będziemy przeprowadzać symulacje dla 1-wymiarowego atomu uwięzionego w studni o dwóch minimach. Wykorzystamy poniższy potencjał:

$$U(x) = -ae^{-b(x-1)^2} - ce^{-(x+1)^2} + dx^4,$$

z parametrami $a = 5$, $b = 10$, $c = 3$, $d = 0,02$. Przy takim zestawie parametrów wykres energii potencjalnej przedstawia Rys. 2.1. To na co warto zwrócić uwagę, to

Rysunek 2.1: Potencjał „minimum-bariera-minimum”:

$$U(x) = -5e^{-10(x-1)^2} - 3e^{-(x+1)^2} + 0,02x^4$$



że „leż” po lewej jest płytszy, ale szerszy od tego po prawej. Na ćwiczeniach zbadamy gęstość rozkładu Boltzmanna takiego potencjału, $\rho(x) \sim e^{-U(x)/k_B T}$, i wyciągniemy wnioski odnośnie tego, w którym „leju” atom będzie częściej przebywał.

2.3.3 Potencjał Lenarda-Jonesa

Na ćwiczeniach stosować będziemy tylko jeden potencjał przedstawiający oddziaływanie pary atomów - potencjał Lenarda-Jonesa:

$$U(\mathbf{r}_{ij}) = \mathcal{E} \left[\left(\frac{R}{\mathbf{r}_{ij}} \right)^{12} - 2 \left(\frac{R}{\mathbf{r}_{ij}} \right)^6 \right],$$

gdzie \mathbf{r}_{ij} to odległość między atomami i oraz j , zaś R i \mathcal{E} to parametry potencjału. Wystarczające będzie dla nas ustalenie stałych wartości parametrów, przykładowo $R = 1$, $\mathcal{E} = 1$.

2.4 Algorytmy całkujące

Algorytm całkujący pozwala wyznaczyć następne (w chwili czasu $t + \Delta t$) położenia i prędkości atomów, na podstawie poprzednich położen, prędkości i przyspieszeń (w chwilach czasu t , $t - \Delta t$, $t - 2\Delta t$, bądź dalszych, w zależności od złożoności algorytmu).

Algorytmy nie będą omawiane w tym skrypcie, a jedynie prezentowane w sposób umożliwiający szybką implementację.

2.4.1 Algorytm Verleta

To nasz algorytm wyjściowy. Najprostszy w implementacji i stabilny.

$$\mathbf{r}(t + \Delta t) := 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) + \mathbf{a}(t)\Delta t^2,$$

przy czym znak „ $:=$ ” należy interpretować jako „przypisz wartość” czy też „staje się”, znane Państwu z praktyki programistycznej. Jest to konwencja obowiązująca do końca tego rozdziału.

Prędkość nie jest potrzebna do wyznaczenia nowych położen, ale niezbędna do obliczenia energii kinetycznej. Można ją obliczyć (estymować) następująco:

$$\mathbf{v}(t) := \frac{\mathbf{r}(t) - \mathbf{r}(t - \Delta t)}{\Delta t},$$

2.4.2 Velocity Verlet

Składa się z dwóch kroków:

$$\mathbf{v}(t + \Delta t) := \mathbf{v}(t) + \frac{\mathbf{a}(t - \Delta t) + \mathbf{a}(t)}{2} \Delta t$$

oraz:

$$\mathbf{r}(t + \Delta t) := \mathbf{r}(t) + \mathbf{v}(t + \Delta t)\Delta t + \frac{\mathbf{a}(t)}{2}\Delta t^2$$

2.4.3 Leapfrog

Na algorytm *leapfrog* również składają się dwa kroki:

$$\mathbf{v}(t + \Delta t) := \mathbf{v}(t) + \mathbf{a}(t)\Delta t$$

oraz:

$$\mathbf{r}(t + \Delta t) := \mathbf{r}(t) + \mathbf{v}(t + \Delta t)\Delta t$$

Rozdział 3

Układ w kontakcie z termostatem

3.1 Rozkład Boltzmanna

Rozkład Boltzmanna, opisujący układ o stałej temperaturze T , wyraża prawdopodobieństwo tego, że układ znajdzie się w i -tej konfiguracji¹:

$$p(i) = \frac{1}{Z} \exp[-E(i)/k_B T], \quad (3.1)$$

gdzie $E(i)$ to energia i -tej konfiguracji, k_B to stała Boltzmanna, a stała normalizacyjna Z zapewnia normalizację rozkładu:

$$\sum_{i=1}^N p(i) = \frac{1}{Z} \sum_{i=1}^N \exp[-E(i)/k_B T] = 1$$

skąd:

$$Z = \sum_{i=1}^N \exp[-E(i)/k_B T],$$

gdzie N jest liczbą konfiguracji jakie może przyjąć układ. Niestety, stała Z jest praktycznie niemożliwa do wyznaczenia dla większych układów, co oznacza, że nie jesteśmy w stanie podać explicite prawdopodobieństw poszczególnych konfiguracji.

Na początku wspomnieliśmy, że gdybyśmy umieli odtworzyć przestrzeń konfiguracyjną układu, moglibyśmy wyznaczyć jego własności fizyczne. Dużo bardziej zaskakujące jest to, że wszystkie własności makroskopowe układu (a one nas zwykle interesują) bylibyśmy w stanie wyznaczyć, gdybyśmy umieli wyznaczyć stałą normalizacyjną Z , występującą w rozkładzie Boltzmanna²!

¹Ta postać rozkładu Boltzmanna sformułowana jest dla układu o przeliczalnym zbiorze możliwych konfiguracji.

²W rzeczywistości Z jest funkcją parametrów układu i nazywa się ją często *sumą statystyczną*.

3.2 Własność makroskopowa

Najpierw podamy podejście bardziej ogólne, w którym przestrzeń fazowa jest zbiorem gęstym, a nie przeliczalnym. Poniższy akapit nie jest związany z modelem HP, ale polecamy go Państwa uwadze.

Mikrostan układu oznaczmy przez $\mathbf{x} = (\mathbf{p}, \mathbf{q})$, gdzie \mathbf{p} to pędy uogólnione, a \mathbf{q} to współrzędne uogólnione - razem opisują one jedną realizację układu. Niech A będzie pewną własnością fizyczną badanego układu, przykładowo: energią potencjalną. Dla mikrostanu \mathbf{x}_i własność A przyjmuje wartość $A(\mathbf{x}_i)$. Przyjmujemy, że wartość A mierzona w doświadczeniu (tzw. **własność makroskopowa**) jest równa wartości oczekiwanej $\mathbb{E}A$ przy zadanej gęstości prawdopodobieństwa f , tzn.:

$$\mathbb{E}A = \lambda Z^{-1} \int_{\Omega} A(\mathbf{x}) f(\mathcal{H}(\mathbf{x}), T) d\mathbf{x}$$

gdzie λ jest stałą, T to temperatura, \mathcal{H} to hamiltonian układu, a Ω jest przestrzenią fazową, zaś:

$$Z = \int_{\Omega} f(\mathcal{H}(\mathbf{x}), T) d\mathbf{x}$$

to suma statystyczna (nazwana również *funkcją podziału*). Istotne jest, aby zechcieli Państwo zauważyć, że f jest funkcją (jedynie!) energii całkowitej układu i temperatury³.

W przypadku modelu HP (którego dotyczy drugi program zaliczeniowy) liczba mikrostanów układu jest skończona (ozn. N), zaś wartość oczekiwana A wyraża się w postaci sumy po dostępnych mikrostanach układu:

$$\mathbb{E}A = \sum_{i=1}^N A(i) p(i), \quad (3.2)$$

gdzie $p(i)$ jest prawdopodobieństwem i -tego mikrostanu. Prawdopodobieństwo, że układ o określonej, stałej temperaturze T znajdziemy w i -tym mikro stanie o energii E_i , dane jest rozkładem Boltzmanna, który dla utrwalenia zapiszemy jeszcze raz, w nieco innej postaci:

$$p(i) = \frac{\exp[-E(i)/k_B T]}{\sum_{j=1}^N \exp[-E(j)/k_B T]}. \quad (3.3)$$

Suma w mianowniku zapewnia normalizację rozkładu p :

$$\sum_{k=1}^N p(k) = 1.$$

³Tak jest w przypadku, gdy jedyne oddziaływanie układu z otoczeniem to oddziaływanie termiczne.

3.3 Metoda Monte Carlo (słowem wstępu)

Metoda Monte Carlo z algorytmem Metropolisa pozwala nam próbkować konfiguracje układu zgodnie z rozkładem Boltzmann, nawet jeżeli nie jesteśmy w stanie wyznaczyć stałej Z .

Powiedzieliśmy we wstępie, że implementując drugi program zaliczeniowy uzyskają Państwo odpowiedź na pytanie: *dla czego* białka natywnie nieustrukturalizowane nie przyjmują jednej konfiguracji. Odpowiedź zawarta jest we wzorze (3.1), a to, że dojście do niej wymaga obycia się z rozkładem Boltzmann pokazuje jak piękne potrafią być proste odpowiedzi na złożone pytania.

Rozdział 4

Drugi program zaliczeniowy

4.1 Model HP

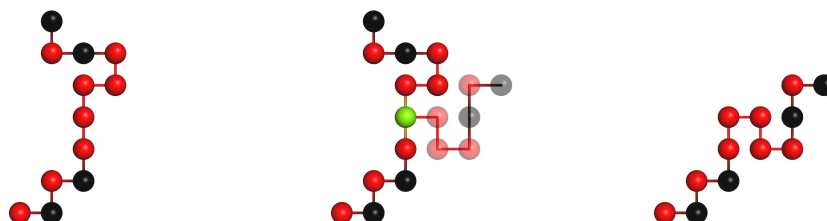
Model HP (*hydrophobic-polar protein folding model*) to model polimeru wykorzystywany w badaniach nad ogólnymi zasadami rządzącymi procesem zwijania białek. Badania tego typu w przypadku modeli pełnoatomowych wiążą się ze znacznymi kosztami obliczeniowymi, podczas gdy w modelu HP, ze względu na uproszczoną charakterystykę układu, możliwe jest przeprowadzenie krótkiej symulacji (trwającej od kilku minut do kilku godzin), w trakcie której układ jest w stanie osiągnąć wszystkie możliwe konfiguracje.

Rysunek 4.1: Dwuwymiarowy model HP o sekwencji: HPHPHHHHHPHP (wizualizacja w PyMOLu). Na czerwono zaznaczono aminokwasy hydrofobowe (H), na czarno aminokwasy polarne (P).

Po lewej: Ponieważ nie występują kontakty H-H, energia wynosi 0.

Środkowy: Obrót wokół szóstego aminokwasu (zaznaczono na zielono) skutkuje utworzeniem kontaktu H-H między ósmym i piątym aminokwasem.

Po prawej: Transformacja została zaakceptowana, liczba kontaktów H-H wynosi 1, zatem nowa energia układu wynosi $-\epsilon$.



Idea modelu HP opiera się na obserwacji, iż kluczową rolę w procesie zwijania białek pełni efekt hydrofobowy (w tym kontekście spotkać się można z terminem: „oddziaływania hydrofobowe”). W podstawowym modelu HP polimer zbudowany jest z monomerów H (hydrofobowych) oraz P (polarnych), przy czym wkład do energii pochodzi jedynie od H. Można więc myśleć o modelu HP jak o modelu

białka, w którym alfabet aminokwasów ograniczony został do zbioru $\{H, P\}$. Aminokwasy znajdują się w węzłach sieci kwadratowej (*square lattice*) w przypadku modelu dwuwymiarowego, zaś w przypadku modelu trójwymiarowego - w węzłach sieci sześcienniej (*cubic lattice*). Dwa aminokwasy nie mogą znajdować się w tym samym węźle. Natomiast jeśli dwa aminokwasy połączone są wiązaniem (przez analogię do wiązania peptydowego między aminokwasami w białkach), to muszą się one znajdować w sąsiednich węzłach. Mikrostan układu można określić poprzez: sekwencję peptydu oraz położenia poszczególnych aminokwasów (czyli konfigurację).

Ewolucję układu w modelu HP zadają:

1. zestaw transformacji struktury;
2. prawdopodobieństwa przejść między mikrostanami.

Przykładem transformacji może być obrót części białka o pewien kąt wokół wybranego aminokwasu (Rys. 4.1). W przypadku modelu 2D istnieją cztery możliwe obroty, z czego jeden z nich jest przekształceniem identycznościowym. Jeżeli po dokonaniu obrotu żadne dwa aminokwasy nie zajmują tego samego punktu w przestrzeni, obrót uznajemy za *dozwolony*.

Po dokonaniu dozwolonej transformacji, prawdopodobieństwo akceptacji nowego mikrostanu zależy jedynie od zmiany wartości energii, $\Delta E = E_{\text{nowa}} - E_{\text{stara}}$. Innymi słowy: to czy zaakceptujemy transformację zależy jedynie od tego jaki jest „nowy” i „stary” mikrostan - wcześniejsza historia układu nie ma tu znaczenia. Zatem ewolucja peptydu (ciąg konfiguracji wygenerowany w toku symulacji) jest realizacją procesu stochastycznego, w którym prawdopodobieństwo akceptacji nowej konfiguracji zależy jedynie od tego, jaka była poprzednia¹.

4.2 Sieć

Niech:

$$\mathbf{e}_x = (1, 0), \quad \mathbf{e}_y = (0, 1)$$

będą wektorami bazowymi w przypadku dwuwymiarowym, zaś:

$$\mathbf{e}_x = (1, 0, 0), \quad \mathbf{e}_y = (0, 1, 0), \quad \mathbf{e}_z = (0, 0, 1)$$

wektorami bazowymi w przypadku trójwymiarowym. Siecią kwadratową nazywać będziemy zbiór

$$LATTICE_{2D} = \{x\mathbf{e}_x + y\mathbf{e}_y : x, y \in \mathbb{Z}\}$$

zaś zbiór:

$$LATTICE_{3D} = \{x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z : x, y, z \in \mathbb{Z}\}$$

¹Proces stochastyczny w którym prawdopodobieństwa zdarzeń zależą jedynie od „stanu przed” i „stanu po” nazywany jest *łańcuchem Markowa*.

nazwiemy siecią sześcienną. Element sieci (węzeł) opisujemy przez podanie dwóch, bądź trzech liczb całkowitych (współrzędnych węzła), przykładowo dla sieci sześcienniej: (0,1,-10). Powiemy, że węzły \mathbf{a} i \mathbf{b} sąsiadują ze sobą (ozn. $\mathbf{a} \sim \mathbf{b}$), jeżeli istnieje wektor bazowy \mathbf{e} taki, że:

$$\mathbf{a} = \mathbf{b} + \mathbf{e} \quad \vee \quad \mathbf{b} = \mathbf{a} + \mathbf{e}$$

Niech $CHAIN = \{1, \dots, m\}$ będzie zbiorem aminokwasów tworzących peptyd, gdzie m to długość peptydu. Wówczas strukturę przestrzenną (dla sieci kwadratowej) wyrażać będziemy przez funkcję:

$$\mathbf{s}: CHAIN \rightarrow LATTICE_{2D}$$

spełniającą warunki

$$\mathbf{s}(1) = (0, 0)$$

$$\forall_{i \neq j} \quad \mathbf{s}(i) \neq \mathbf{s}(j)$$

$$\forall_{i < n} \quad \mathbf{s}(i+1) \sim \mathbf{s}(i)$$

Sama struktura (w postaci funkcji \mathbf{s}) nie wystarcza do określenia energii układu, potrzebna jest jeszcze sekwencja aminokwasowa.

4.3 Sekwencja

Sekwencja łańcucha określona jest przez wzorzec hydrofobowy:

$$Pat: CHAIN \rightarrow \{H, P\}.$$

Rozważany model dzieli aminokwasy ze względu na właściwości oddziaływań dalekozasięgowych na dwie kategorie: hydrofobowe (H) oraz polarne (P). Dalekozasięgowość oddziaływań odnosi się do wzajemnych położeń aminokwasów w sekwencji, a nie w przestrzeni. Przykładowo: o obecności oddziaływań dalekozasięgowych możemy mówić w przypadku pary aminokwasów o numerach 1 i 4, bądź: 2 i 9, ale nie w przypadku par: 1 i 3, czy też 4 i 5 (które są „blisko siebie” w sekwencji). Szczegóły w poniższej sekcji *Oddziaływania*.

4.4 Oddziaływania

W najprostszym modelu HP rozważa się jedynie oddziaływania dalekozasięgowe pomiędzy aminokwasami hydrofobowymi. Parę aminokwasów hydrofobowych, znajdujących się w sąsiednich węzłach, które nie są „blisko siebie w sekwencji”, nazwiemy *kontaktem*. Energia danej struktury zależy jedynie od liczby kontaktów i jest do niej proporcjonalna.

Ściślej, niech:

$$K_{HH}(\mathbf{s}) = \#\{\{i, j\} : |i - j| > 2, \quad \mathbf{s}(i) \sim \mathbf{s}(j), \quad Pat(i) = Pat(j) = H\}$$

oznacza liczbę kontaktów w strukturze \mathbf{s} . Energia układu wyraża się wzorem:

$$E(\mathbf{s}) = -\epsilon K_{HH}(\mathbf{s}),$$

gdzie $\epsilon > 0$.

Interakcje pomiędzy aminokwasami hydrofobowymi odzwierciedlają ich tendencję do kierowania się do wewnątrz białka i tym samym unikania kontaktu z wodą. Należy podkreślić, że model HP uwydatnia jeden aspekt procesu zwijania białek (efekt hydrofobowy), ignoruje natomiast oddziaływania lokalne występujące w rzeczywistym białku:

1. sztywność łańcucha (objawiającą się niedozwolonymi wartościami kątów ϕ - ψ na wykresie Ramachandrana)
2. wiązania wodorowe (istotne między innymi w α -helisach i β -kartkach, czyli w tzw. strukturze II-rzędowej).

Model HP jest niezwykle prosty, co nasuwa jedno oczywiste pytanie: *Czy nie jest przypadkiem ZBYT prosty?* Naturalnie, rzeczywiste białka (a nawet bardziej złożone modele białek) nie są rozpięte na sieci. Więcej: są opisane pędami. Ale to nie jedyne zastrzeżenia, różnic można wymienić więcej.

Jeżeli chcielibyśmy wszystkie wnioski wyniesione z modelu HP przenieść na rzeczywiste białka, to owszem - model HP okazałby się zbyt prosty. Jednak mamy nadzieję i okazję wyrobić sobie intuicję nie tyle związaną z modelem HP, ale z rozkładem Boltzmanna, który jest kluczową cechą, łączącą ten model z rzeczywistymi białkami.

4.5 Estymacja własności makroskopowych

Jeżeli dysponujemy dowolną próbą mikrostanów (\mathbf{x}_n), możemy własność makroskopową A *estymować*, obliczając jej średnią $\langle A \rangle$:

$$\mathbb{E}A \approx \langle A \rangle = \frac{1}{M} \sum_{\alpha=1}^M p(\mathbf{x}_\alpha) A(\mathbf{x}_\alpha),$$

gdzie $p(\mathbf{x}_\alpha)$ jest prawdopodobieństwem mikrostanu \mathbf{x}_α , zaś M jest licznością próby. Jednakże estymator tego typu, z dowolnie (źle) próbkowaną przestrzenią fazową może bardzo wolno zbiegać do wartości oczekiwanej $\mathbb{E}A$. W zasadzie można wymyślić próbkowanie, przy którym estymator nigdy nie zbiegne do wartości oczekiwanej.

Założmy, że dysponujemy ciągiem mikrostanów próbkowanych zgodnie z rozkładem Boltzmanna, (\mathbf{x}_n^B) . Wówczas możemy estymować własność makroskopową A przez średnią $\langle A \rangle_B$:

$$\mathbb{E}A \approx \langle A \rangle_B = \frac{1}{M} \sum_{\alpha=1}^M A(\mathbf{x}_\alpha^B), \quad (4.1)$$

gdzie M jest licznością próby. Jest szalenie istotne, aby umieli Państwo rozróżnić sumę po możliwych mikrostanach (3.2), która definiuje $\mathbb{E}A$, od sumy po mikrostanach z próby (4.1), która definiuje estymator $\mathbb{E}A$, oznaczony tutaj przez $\langle A \rangle_B$.

4.6 Metoda Monte Carlo - algorytm Metropolis

Algorytm Metropolis pozwoli nam otrzymać próbę zgodną z rozkładem Boltzmanna, czyli (\mathbf{x}_B) .

Wprowadźmy następujące oznaczenie:

$$\pi(j) = \exp[-E_j/k_B T],$$

gdzie j jest konfiguracją o energii E_j . Istotą algorytmu Metropolis jest stworzenie ciągu mikrostanów, będącego realizacją łańcucha Markowa z prawdopodobieństwem przejść, zależącym od różnicy energii kolejnych mikrostanów. W przypadku modelu HP algorytm Metropolis przebiega następująco:

1. Zainicjuj strukturę danych A , w której zapamiętywać będziemy mikrostany.
2. Stwórz mikrostan X .
3. Oblicz energię E_X .
4. Stwórz mikrostan Y , jako wynik dozwolonej transformacji mikrostanu X^2 .
5. Oblicz energię E_Y .
6. Zaakceptuj nowy mikrostan z prawdopodobieństwem:

$$p(\text{accept}\{X \rightarrow Y\}) = \min \left\{ 1, \frac{\pi(Y)}{\pi(X)} \right\}.$$

7. Jeżeli mikrostan został zaakceptowany wykonaj:

$$X := Y$$

8. Zapisz mikrostan:

$$A.\text{push}(X)$$

9. Jeżeli dokonano K kroków, zakończ i wypisz statystyki do pliku. W przeciwnym wypadku wróć do 4.

4.7 Transformacje

Będziemy zajmować się tylko modelem 2D, dla którego wystarczającą będzie transformacja „obróć wokół aminokwasu”. Wystarczającą, albowiem z każdej konfiguracji jesteśmy w stanie odtworzyć dowolną inną konfigurację na drodze ciągu transformacji „obróć wokół aminokwasu”. Transformacja prezentuje się następująco:

²Dokonujemy obrotu części peptydu wokół losowo wybranego aminokwasu.

losujemy aminokwas (dowolny, oprócz ostatniego), losujemy macierz obrotu i dokonujemy transformacji aminokwasów, które znajdują się w sekwencji za wylosowanym aminokwasem. Niektóre elementy polimeru mogą się w wyniku takiej transformacji nakładać, należy zatem sprawdzić czy obrót jest dozwolony.

Można wymyślić jeszcze wiele transformacji, które razem z obrotami prowadziłyby do szybciej zbieżnych estymacji. W szczególności, transformacje które pozwoliłyby osiągać mikrostanu trudne do zrealizowania przy użyciu samych obrotów (szczegół na ćwiczeniach). Choć nie jest to temat, który będziemy zgłębiać w ramach tych ćwiczeń, warto zaakcentować, że problem doboru transformacji jest niezwykle istotny w metodach MC.

4.8 Symulowane wyżarzanie (*simulated annealing*)

Najprostszym sposobem znajdowania konformacji o minimalnej energii jest systematyczne obniżanie temperatury podczas symulacji. Przy czym konfiguracją wyjściową w kolejnej, niższej temperaturze jest końcowa konfiguracja z poprzedniego kroku temperaturowego (szczegóły poniżej, w opisie programu zaliczeniowego). Wadą tego rozwiązania jest to, że układ może się łatwo zatrzymać w lokalnym minimum energii, z którego wyjście przy obniżonej temperaturze okaże się niezwykle mało prawdopodobne. Ponadto, zbieżność algorytmu przy niskich temperaturach jest nieznośnie powolna. Układ może stracić dużo czasu (kroków symulacji) w niecce reprezentującej lokalne minimum, bądź oscylując między stanami o tej samej energii.

4.9 Zamiana replik (*Replica Exchange Monte Carlo*)

W tym podejściu równolegle symuluje się wiele kopii układu, każdą w innej temperaturze. Załóżmy, że w pewnym momencie symulacji algorytmu Metropolis'a i -ta replika o temperaturze T_i jest w mikro stanie \mathbf{x}_i o energii $E(\mathbf{x}_i)$, zaś j -ta replika w odpowiednio: temperaturze T_j , w mikro stanie \mathbf{x}_j o energii $E(\mathbf{x}_j)$. Z rozkładu jednorodnego losujemy parę kolejnych replik (i, j) , które z prawdopodobieństwem p_{exch} zostaną zamienione miejscami:

$$p_{exch} = \min\{1, e^{-\Delta}\},$$

gdzie

$$\Delta = \left(\frac{1}{k_B T_j} - \frac{1}{k_B T_i} \right) (E(\mathbf{x}_i) - E(\mathbf{x}_j)).$$

Po zamianie i -ta replika symulowana jest w temperaturze T_j , a j -ta w temperaturze T_i . Ponieważ prawdopodobieństwo zamiany maleje wykładniczo wraz ze wzrostem różnicy temperatur, rozważamy wyłącznie repliki sąsiednie. Zamiana temperatur nie zmienia krajobrazu energetycznego, ale zmienia prawdopodobieństwa konfiguracji. Repliki, które utknęły w lokalnych minimach mogą zostać z nich wyzwolone

przez przeniesienie do wyższej temperatury. Wymiany nie powinny być zbyt częste, gdyż po zmianie temperatury układ przez pewien czas stabilizuje się.

Przypominamy, że po zakończonej symulacji własność A możemy estymować zgodnie ze wzorem:

$$\langle A \rangle_B = \frac{1}{M} \sum_{\alpha=1}^M A(\mathbf{x}_\alpha).$$

4.10 Symulacje wymagane do zaliczenia

Celem ćwiczeń jest zaimplementowanie modelu HP (domyślnie w języku Java) i sprawdzenie wyników przedstawionych w pracy Dilla [?]. Należy przeprowadzić symulowane wyżarzanie modeli dwuwymiarowych trzech polimerów o sekwencjach:

- PHPPHPPHHPHHPHPPHP
- HPPPHHPHPHHHHH
- HPPPHHPHPHHPHHH

W przypadku każdego peptydu temperaturą początkową będzie $T_{max} = 1$, a temperaturą końcową $T_{min} = 0.15$. Przyjmujemy, że wartość stałej Boltzmanna wynosi $k_B = 1$. Temperatura będzie w trakcie symulacji maleć o czynnik $\delta = 0.05$. Po przeprowadzeniu $K = 10000$ transformacji (dokonując akceptacji/odrzuceń wygenerowanych mikrostanów) w danej temperaturze T , przeprowadzamy kolejnych K kroków, w temperaturze $T - \delta$ i tak dalej, aż do osiągnięcia T_{min} . Konfiguracja początkowa układu w temperaturze $T - \delta$ będzie konfiguracją końcową z temperatury T .

W dużym uproszczeniu:

SYMULOWANE-WYŻARZANIE($T_{max}, T_{min}, \delta, K$)

```

1  INICJALIZACJA(U)
2  ▷ Zainicjalizuj układ U
3   $T \leftarrow T_{max}$ 
4  while  $T \geq T_{min}$ 
5      do
6          ZRÓB-K-TRANSFORMACJI-Z-ALGORYTMEM-METROPOLISA( $U, K, T$ )
7          WYPISZ-KONFIGURACJĘ-O-NAJNIŻSZEJ-ENERGII
8          WYPISZ-DOTYCHCZASOWĄ-TRAJEKTORIĘ
9           $T \leftarrow T - \delta$ 
10 WYPISZ-CV-ORAZ-ŚREDNI-MOMENT-BEZWŁADNOŚCI
```

Wykorzystując uzyskane z symulacji próby mikrostanów, należy wyznaczyć ciepło właściwe układu:

$$C_V(T) \approx \frac{\langle E^2 \rangle_B(T) - \langle E \rangle_B^2(T)}{k_B T^2},$$

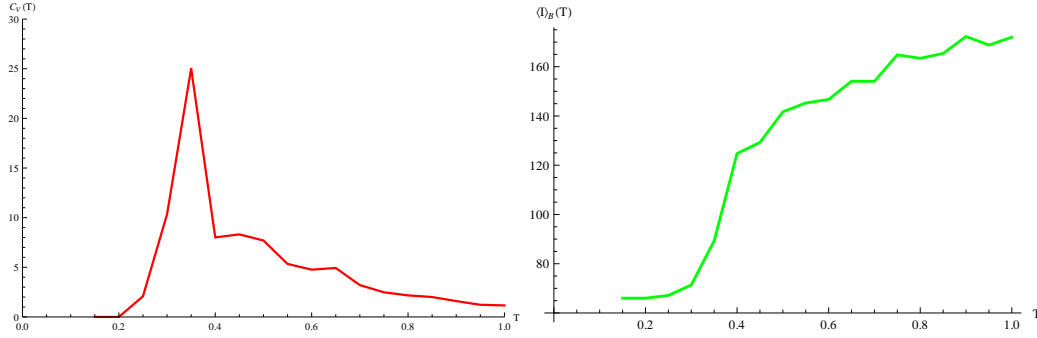
a także średni moment bezwładności $\langle I \rangle_B(T)$. Moment bezwładności polimeru o konfiguracji \mathbf{s} wyraża się wzorem:

$$I = \sum_{i=1}^m (\mathbf{s}(i) - \boldsymbol{\mu})^2,$$

gdzie $\boldsymbol{\mu}$ to środek masy polimeru, $\boldsymbol{\mu} = \frac{1}{m} \sum_{i=1}^m \mathbf{s}(i)$.

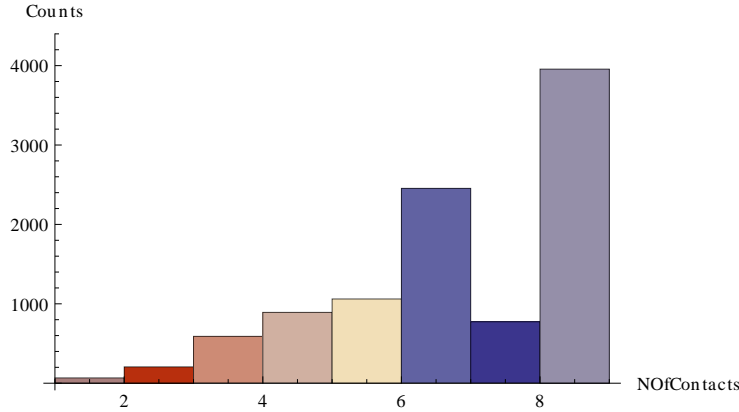
W każdej temperaturze wygenerujemy dokładnie K konfiguracji (zgodnie z algorytmem Metropolisa). Dla każdej temperatury należy sporządzić histogram liczby wystąpień konfiguracji w zależności od liczby kontaktów. Sporządzić należy również wykres zależności ciepła właściwego i średniego momentu bezwładności od temperatury.

Rysunek 4.2: Wykresy ciepła właściwego, $C_V(T)$, oraz średniego momentu bezwładności, $\langle I \rangle_B(T)$, polimeru o sekwencji PHPHPHPHHPPHHPPHPHP:



Przykładowy histogram wystąpień konfiguracji o zadanej liczbie kontaktów przedstawiono na Rys. 4.3.

Rysunek 4.3: Histogram kontaktów dla $T = 0.35$, dla polimeru o sekwencji PHPHPHPHHPPHHPPHPHP:



Wyniki należy przedstawić w postaci wykresów. Pisemne sprawozdania nie są obowiązkowe, sprawdzane będą wykresy (histogramy we wszystkich temperaturach, $C_V(T)$ i $\langle I \rangle_B(T)$) i działanie programu (na jednym z ćwiczeń należy uruchomić program i przeprowadzić symulacje).

Rozdział 5

Trzeci program zaliczeniowy

Celem trzeciego programu zaliczeniowego jest przeprowadzenie symulacji MD układu bardzo zbliżonego do znanego nam już modelu HP. W pierwszej kolejności należy zaimplementować poniższy algorytm całkujący (5.1). Następnie, potrzebujemy upewnić się, że nasz program działa dla wymiarów wyższych, niż 1. W końcu, musimy umożliwić, aby kilka potencjałów mogło opisywać nasz układ *jednocześnie* (np. Lenarda-Jonesa + “miękkie ścianki”) – w tym celu najlepiej jest skorzystać z dekoratora (wzorca projektowego omówionego na ćwiczeniach). Warto dodać potencjał “miękkie ścianki”, aby układ nie odpłynął nam w siną dal. Jednak uwaga: stała L powinna być na tyle duża, aby sfera zerowego potencjału pomieściła cały układ.

Do symulacji układu zbliżonego do modelu HP potrzebujemy jeszcze dwóch dodatkowych potencjałów: oddziaływania kolejnych aminokwasów i wiązania H–H (5.2 i 5.3). Dodatkowo, klasa *Atom* potrzebuje nowego atrybutu – *type* – który będzie znakiem P bądź H.

5.1 Równanie Langevina - integrator BBK

Oddziaływanie układu z termostatem o temperaturze T modeluje się na wiele sposobów. Jednym z nich jest integrator BBK:

$$\mathbf{r}(t + \Delta t) := \frac{2\mathbf{r}(t) - \mathbf{r}(t - \Delta t)(1 - \Delta t/2) + \frac{1}{m}(\mathbf{F}(t) + \mathbf{F}_R)(\Delta t)^2}{1 + \Delta t/2},$$

gdzie \mathbf{F}_R jest siłą losową, generowaną z 3-wymiarowego rozkładu normalnego, ze średnią $\mathbf{0}$ i wariancją $2k_B T m / \Delta t$ w każdym z trzech kierunków. Przyjmujemy, że stała Boltzmanna ma wartość 1.

5.2 Potencjał *bonded*

Jeżeli atomy i oraz j są kolejnymi elementami układu (“aminokwasami”), ich oddziaływanie wzbogacone będzie następującym potencjałem:

$$U(\mathbf{r}_{ij}) = \kappa(r_0 - \|\mathbf{r}_{ij}\|)^2,$$

gdzie κ jest stałą sprężystości (weźmy $\kappa = 200$), \mathbf{r}_{ij} to wektor łączący atom i oraz j , zaś r_0 to odległość równowagowa między atomami (weźmy $r_0 = 1$). Potencjał *bonded* zapewniać będzie odpowiednią długość wiązania (u nas 1), tak aby nasz łańcuch przypominał model HP, w którym długość wiązań była ustalona “na sztywno”.

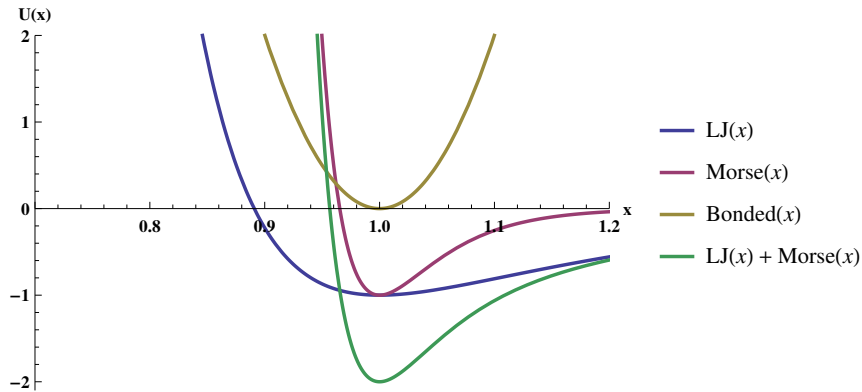
5.3 Potencjał Morse’a

Jeżeli atomy i oraz j są typu H i dzielą je co najmniej dwa inne atomy (tzn. $|i - j| > 2$), występuje między nimi oddziaływanie H–H, które modelować będziemy potencjałem Morse’a:

$$U(\mathbf{r}_{ij}) = D (1 - e^{-\alpha(\|\mathbf{r}_{ij}\| - r_0)})^2 - D,$$

w którym stała D wpływa na głębokość minimum, zaś parametr α na kształt (“stro-mość”) funkcji. Weźmiemy $D = 1$ oraz $\alpha = 20$.

Rysunek 5.1: Kształt potencjałów: Lenarda-Jonesa ($\mathcal{E}, R = 1$), Morse’a ($D = 1$, $\alpha = 20$) oraz *bonded* ($\kappa = 200$) w przypadku 1-wymiarowym.



5.4 Symulacja

Zaliczenie trzeciego programu zaliczeniowego polegać będzie na przeprowadzeniu symulacji przy zadanej długości i liczbie kroków. Na stałe można wszyć do programu: układ i algorytm całkujący (BBK). Prosiłbym, aby układem wyjściowym był pierwszy łańcuch HP z drugiego programu zaliczeniowego, z konfiguracją o najniższej energii. Przedział temperatur, który nas interesuje to $T \in [0, 1]$.

W odróżnieniu do pierwszego programu zaliczeniowego, tym razem sprawdzana będzie przede wszystkim trajektoria układu (plik `.xyz`).