

Poniższe ćwiczenia przewidziane są do wykonania z wykorzystaniem programów wchodzących w skład pakietu Tinker 5.0.

Ogólne zasady tworzenia plików z danymi i zapisywania wyników są następujące.

Niezbędne dane to struktura badanego układu, zapisana w pliku z rozszerzeniem `.xyz` (np. `geom.xyz`). Zapisane są w nim informacje o liczbie atomów, ich rodzaju, współrzędnych, typie atomu w polu siłowym i wiązaniach z sąsiednimi atomami. Plikowi `.xyz` towarzyszyć powinien plik z rozszerzeniem `.key` zawierający słowa kluczowe określający sposób wykonywania pewnych zadań. Niezbędne jest też podanie pliku `.prm` z parametryzacją pola siłowego (w przypadku prostych pól można zeń zrezygnować a parametryzację wpisać bezpośrednio do pliku `.key`).

Programy stosujące mechanikę molekularną (np. optymalizacja geometrii) zapisują strukturę wynikową dodając do rozszerzenia `.xyz` odpowiedni numer. Kolejny start programu wykorzystuje strukturę zapisaną w pliku z najwyższym numerem; np. po minimalizacji energii dla geometrii struktury zapisanej w pliku `geom.xyz` utworzony zostanie plik `geom.xyz_2`, kolejne uruchomienie optymalizacji struktury `geom` użyje jako danych wejściowych pliku `geom.xyz_2`, wyniki zapisując w `geom.xyz_3`.

Program symulujący dynamikę molekularną może przy odpowiednich poleceniach w pliku `.key` zapisywać w określonych odstępach czasu aktualną geometrię układu w pliku o rozszerzeniu `.arc`. Zapisywany jest także plik o rozszerzeniu `.dyn` zawierający informację o położeniach atomów i ich prędkościach. Jeśli podczas startu dynamiki istnieje plik `.dyn`, to informacja o połączeniu atomów zostanie wczytana z pliku `.xyz`, natomiast symulacja zacznie się od położenia i prędkości zapisanych w pliku `.dyn`. Umożliwia to restart symulacji.

Większość plików `.xyz` daje się obejrzeć programem `molden`, po wydaniu polecenia typu:

```
molden nazwa.xyz
```

Pliki te można też obejrzeć w programie `VMD`. Po jego uruchomieniu poleceniem

```
vmd
```

należy wczytać odpowiednią strukturę informując jednocześnie program, że jest zapisana w formacie Tinkera: otwieramy menu *File* i wybieramy *New Molecule...*, po naciśnięciu *Browse* należy odszukać plik i zatwierdzić przez *OK*, w okienku „*Determine File Type*” należy odszukać typ *Tinker* i nacisnąć *Load*.

Pliki `.arc` wygodniej oglądać w programie `VMD`, są one rozpoznawane jako pliki Tinkera, więc wystarcza wydanie polecenia:

```
vmd nazwa.arc
```

Po załadowaniu pliku do `VMD` warto zmienić sposób rzutowania przestrzeni: menu *Display*, wybieramy *Ortographic*.

Kartotekę `mnch-md` z niezbędnymi do wykonania ćwiczeń danymi uzyskasz pobierając plik <http://www.chemia.uj.edu.pl/~eilmes/mnumch/dane5.zip>

i rozpakowując go poleceniem

```
unzip dane5
```

1. Zbadaj, jaka będzie najkorzystniejsza konfiguracja dwu i czterech obojętnych atomów oddziałujących zgodnie z potencjałem Lennarda-Jonesa o $r_{\min} = 2 \text{ \AA}$ i $\varepsilon = 0.25 \text{ kcal/mol}$. Jak zmieni się geometria, jeśli połowa atomów będzie obdarzona ładunkiem $-e$ a połowa $+e$?

Rozwiązanie.

Plik `atoms.prm` zawiera definicję parametrów pola siłowego. Wygląda on następująco:

```
vdwtype      lennard-jones
radiustype    r-min
radiussize    radius
dielectric    1.0

atom  1  X  "X"          18    39.948  0
atom  2  Y  "Y"          18    39.948  0

vdwpr  1  2    2.00    0.25
vdwpr  1  1    2.00    0.25
vdwpr  2  2    2.00    0.25

charge  1    1.0
charge  2   -1.0
```

Zapisy te mówią, że zdefiniowane mamy dwa typy atomów 1 i 2 oznaczone X i Y (o masie równej atomowi argonu, ale z punktu widzenia minimalizacji energii to nieistotne), ładunkach odpowiednio 1 i -1 oraz oddziałujących potencjałem Lennarda-Jonesa o zadanych parametrach (po `vdwpr` podane są typy oddziałujących atomów, oraz wartości $r_{\min}$ i ε).

Plik `atoms2.xyz` zawiera startową geometrię dwu atomów:

```
2
1  X    0.0    0.0    0.0    1
2  Y    5.0    0.0    0.0    2
```

Zapis ten mówi, że mamy dwa atomy, pierwszy typu 1 położony w (0,0,0), drugi typu 2 w (5,0,0).

Skopiujmy ten plik na plik `atoms.xyz`:

```
cp atoms2.xyz atoms.xyz
```

Plik `atoms.key` ma następującą zawartość:

```
parameters ./atoms.prm
```

```
chargeterm none
```

Pierwszy zapis mówi, gdzie znaleźć parametry pola siłowego, drugi, że na razie wyłączamy oddziaływania ładunek-ładunek (co odpowiada neutralnym atomom).

Minimalizację przeprowadzamy wydając polecenie

```
minimize.x atoms
```

i zadając kryterium zbieżności 0.0001

Wynikową strukturę `atoms.xyz_2` można obejrzyć w edytorze tekstu, jej wyświetlenie w molidenie prawdopodobnie się nie powiedzie (ale da się obejrzyć w VMD).

Jak widać, atomy znalazły się w odległości 2 Å od siebie (czego należało oczekiwać, bo tam jest minimum energii van der Waalsa). Możemy uzyskać bliższe informacje o składowych tej energii wydając polecenie

```
    analyze.x atoms
i wybierając opcję d
```

Dowiadujemy się stąd tego, co już wiemy: atomy są w odległości 2 Å a ich energia potencjalna wynosi -0.25 kcal/mol.

Przed zbadaniem czterech atomów musimy usunąć plik `atoms.xyz_2` poleceniem

```
rm atoms.xyz_2
```

Zbadajmy teraz przypadek czterech atomów startując z atomów umieszczonych w narożach kwadratu.

Wydając polecenia

```
cp atoms4p.xyz atoms.xyz
minimize.x atoms 0.0001
analyze.x atoms d
```

i ewentualnie oglądając strukturę wynikową `atoms.xyz_2` w molidenie, VMD lub edytorze tekstu stwierdzamy, że atomy rozmieszczone są w narożach kwadratu o boku 1.98 Å a energia układu wynosi -1.12 kcal/mol.

Start z pozycji o wysokiej symetrii nie jest jednak dobrym pomysłem. Zauważmy bowiem, że ze względu na symetrię składowa gradientu w kierunku prostopadłym do płaszczyzny jest równa 0, więc atomy początkowo rozmieszczone na płaszczyźnie mogą stać się nie-koplanarne tylko w wyniku błędów numerycznych.

Sprawdźmy zatem przypadek, kiedy jeden z atomów będzie minimalnie wychylony z płaszczyzny:

```
rm atoms.xyz_2
cp atoms4.xyz atoms.xyz
minimize.x atoms 0.0001
analyze.x atoms d
```

Widzimy, że atomy ustawiły się w tetraedr o krawędzi 2 Å. Analiza przyczynków do energii pokazuje, że w takim ustawieniu każda para atomów jest odległa o 2 Å, czyli jest w minimum potencjału van der Waalsa, co przy zsumowaniu 6 oddziaływań daje energię potencjalną równą -1.5 kcal/mol.

Zbadajmy teraz efekt włączenia oddziaływań elektrostatycznych. W tym celu w pliku `atoms.key` wykomentujmy polecenie wyłączające te oddziaływania umieszczając przed nim znak `#`. Plik ma wtedy postać:

```
parameters ./atoms.prm

#chargeterm none
```

Możemy teraz sprawdzić przypadek dwu atomów:

```
rm atoms.xyz_2
cp atoms2.xyz atoms.xyz
minimize.x atoms 0.0001
analyzer.x atoms d
```

Stwierdzamy, że odległość między atomami zmalała do 1.37 Å. Analiza składowych energii wyjaśnia, dlaczego tak jest. Przyciąganie elektrostatyczne jest silniejsze niż oddziaływanie van der Waalsa, opłaca się więc zbliżyć atomy: odpychanie van der Waalsa 17.8 kcal/mol jest kompensowane z nadwyżką przez przyciąganie elektrostatyczne -241.6 kcal/mol dając całkowitą stabilizację -223.8 kcal/mol.

Sprawdźmy jeszcze 4 atomy (startując z atomów niekoplanarnych):

```
rm atoms.xyz_2
cp atoms4.xyz atoms.xyz
minimize.x atoms 0.0001
analyzer.x atoms d
```

Dostaliśmy kwadrat o boku 1.43 Å i energii -559.9 kcal/mol. Płaska struktura kwadratowa jest w tym przypadku korzystniejsza, ponieważ umożliwia 4 silne przyciągania elektrostatyczne (po bokach kwadratu) i 2 słabsze odpychania (po przekątnej). W tetraedrze wszystkie odległości są równe, więc zysk na oddziaływaniach elektrostatycznych byłby mniejszy (można to sprawdzić wyłączając elektrostatykę na czas optymalizacji startującej z niepłaskiej geometrii a później włączając ją do analizy energii).

2. Zbadaj optymalne struktury klastrow Lennarda-Jonesa dla N atomów, $N = 4 - 7$.

Rozwiązanie.

Wykorzystamy program `monte.x` łączący metodę Monte-Carlo z optymalizacją gradientową. Po wykonaniu próbnego ruchu MC optymalizowana jest struktura układu i na podstawie znalezionej energii minimum wyznaczane jest prawdopodobieństwo akceptacji nowej konfiguracji w algorytmie Metropolis.

Przed rozpoczęciem pracy należy wyłączyć oddziaływania elektrostatyczne w pliku `atoms.key` (atomy mają oddziaływać jedynie potencjałem Lennarda-Jonesa).

Startowe geometrie znajdują się w plikach `monteN.xyz`, gdzie $N = 4, 5, 6, 7$.

Dla znalezienia optymalnej geometrii 4 atomów LJ wydajemy polecenia

```
rm atoms.xyz_2
cp monte4.xyz atoms.xyz
monte.x atoms
```

W programie akceptujemy dwie pierwsze opcje (liczbę kroków i rodzaj kroku), zadajemy wielkość kroku na 0.3, akceptujemy temperaturę i ustawiamy warunek zbieżności na 0.001.

Oglądając moldelem plik `atoms.xyz_2` stwierdzamy, że atomy ustawiły się w narożach tetraedru (co już wiemy z poprzedniego zadania). Powtarzając procedurę dla kolejnych

plików stwierdzamy, że dla pięciu atomów dostajemy bipiramidę trygonalną, dla 6-ciu – oktaedr a dla 7 – bipiramidę pentagonalną.

Oczywiście, ze względu na losowość procesu, wyniki dla tej samej geometrii startowej mogą się okazać różne dla kolejnych przebiegów programu. Chcąc się upewnić, że rzeczywiście znaleziono minimum należałoby sprawdzić po kilka razy różne geometrie startowe.

3. Zbadaj częstość drgania rozciągającego C=O w octanie etylu korzystając z pola siłowego MM3.

Rozwiązanie.

Zacznijmy od utworzenia pliku z geometrią cząsteczki w programie molden.

Wygodniej może być wyłączyć cieniowanie (przycisk *Shade*). Klikając w *ZMAT Editor* rozpoczynamy edycję geometrii. Wybieramy *Add Line* i atom C. Następnie klikamy w atom i wydajemy polecenie *Substitute atom by Fragment* wybierając CH₄. Później analogicznie jeden z atomów wodoru zastępujemy grupą –COOH, a następnie wodór z tej grupy grupą CH₃. Na końcu leżący w płaszczyźnie cząsteczki atom H z dodanej właśnie grupy CH₃ zastępujemy kolejną grupą CH₃ i klikamy *Close*. Wydajemy polecenie *Write* i wybieramy *Tinker QM/MM*, podając nazwę pliku `octan.xyz`.

Następnie należy w pliku `octan.xyz` ustawić właściwe rodzaje atomów (6 kolumna). Atomy wodoru mają typ 5, węgle o hybrydyzacji sp³ typ 1 a węgiel związany z atomem tlenu wiązaniem podwójnym – typ 3. Atom tlenu z grupy C=O jest typu 7 a drugi – typu 6. (Widać, że w większości przypadków ostatnia cyfra numeru wpisanego przez molden jest poprawna, co ułatwia modyfikacje.)

Plik `octan.key` powinien zawierać informacje o polu siłowym:

```
parameters      ./mm3
```

Możemy teraz zoptymalizować geometrię:

```
minimize.x octan 0.00001
```

i obliczyć drgania normalne:

```
vibrate.x octan
```

W wyniku dostajemy tabelę wartości własnych hesjanu oraz częstości drgań normalnych. Zauważmy, że 3 stopnie swobody odpowiadają translacji cząsteczki jako całości a 3 rotacjom. Sześć pierwszych wartości własnych hesjanu i częstości drgań normalnych powinno być zatem równe 0. W praktyce warunek ten spełniony bywa w przybliżeniu (tak jest i u nas), znaczna różnica w stosunku do 0 oznacza punkt siodłowy (wartość wyraźnie ujemna) lub problemy z wyznaczeniem drgań (wartości dodatnie). Pozostałe 3N -6 częstości odpowiada oscylacjom.

Z chemii organicznej wiemy, że drgania C=O powinniśmy szukać w ok. 1700 cm⁻¹. W naszym przypadku mamy drganie nr 34 o częstości 1747 cm⁻¹. Podajmy programowi jego numer (34), w wyniku otrzymamy plik `octan.034` z informacjami o wychyleniach atomów w tym drganiu. Charakter drgania możemy obejrzeć w moldenie ładując odpowiedni plik

```
molden octan.034
```

i używając przycisków *Next* i *Prev* do przechodzenia pomiędzy kolejnymi obrazkami.

Możemy też spróbować VMD. Ze względu na odkształcenie cząsteczki brakuje wyrysowanych dwu wiązań; możemy je dodać poleceniem *Add/Remove Bonds* w menu *Mouse*. Następnie sposób wyświetlania trajektorii zmieniamy z *loop* na *rock* i możemy już uruchomić animację.

Widać, że drganie 1747 cm^{-1} rzeczywiście odpowiada rozciąganiu wiązania $\text{C}=\text{O}$. Wg podręcznika „*Chemia organiczna*” Morrisona i Boyda doświadczalnie zmierzona częstość to 1740 cm^{-1} , nasz wynik jest zatem bliski eksperymentowi.

4. Wyznacz strukturę stanu przejściowego pomiędzy konformacją krzesłową i skręconej łódki w cykloheksanie.

Rozwiązanie.

Po pierwsze musimy znaleźć te dwie konformacje. Wyznaczymy je optymalizując geometrię cząsteczki w polu siłowym MM3. Oczywiście dla znalezienia dwu konformacji musimy spróbować dwu różnych punktów startowych.

Wydajemy polecenia

```
cp cyclohex.startb.xyz cyclohex.xyz
minimize.x cyclohex 0.00001
mv cyclohex.xyz_2 cyclohex.tboat
```

```
cp cyclohex.startc.xyz cyclohex.xyz
minimize.x cyclohex 0.00001
mv cyclohex.xyz_2 cyclohex.chair
```

W tym momencie w plikach *cyclohex.tboat* i *cyclohex.chair* znajdują się geometrie konformacji skręconej łódki i krzesła. Zauważmy, że ich energie to ok. 13.8 oraz 8. kcal/mol, zgodnie z przewidywaniami konformacja krzesłowa jest stabilniejsza.

W pliku *cyclohex.key* usuwamy komentarz (znak #) z dyrektywy *saddlepoint* (będziemy teraz szukać punktu siodłowego).

Wydajemy polecenie

```
saddle.x cyclohex.chair cyclohex.tboat 0.01 y
```

W wyniku dostajemy plik *tstate.xyz* z geometrią stanu przejściowego.

Dooptymalizujmy go jeszcze trochę:

```
mv tstate.xyz cyclohex.xyz
newton.x cyclohex a a 0.00001
```

Geometria stanu przejściowego znajduje się teraz w *cyclohex.xyz_2*

Zauważmy, że jego energia to 18.9 kcal/mol, czyli jest wyższa od obu konformacji odpowiadających minimum energii.

Sprawdźmy wartości własne hesjanu dla tej konformacji:

```
vibrate.x cyclohex
```

i widzimy, że mamy jedną wyraźnie ujemną wartość własną, co oznacza punkt siodłowy.

5. Zbadaj na przykładowym układzie zmiany całkowitej energii układu dla algorytmów całkowania Beemana i velocity Verlet dla różnych kroków czasowych.

Rozwiązanie.

Symulacje poprowadzimy dla cząsteczki lipidu (`lipid.xyz`) w polu CHARM22.

Zobaczmy zawartość pliku `lipid.key`:

```
parameters    ./charmm22
archive
```

```
integrate beeman
randomseed 135792468
```

Instrukcja `integrate` określa algorytm całkowania (w tym przypadku algorytm Beemana).

Rozpocznijmy symulację dynamiki:

```
dynamic.x lipid 100000 1.0 0.5 1 > lipid-beem-1.log &
```

Poszczególne liczby oznaczają:

100000 - liczba kroków

1.0 - krok w fs

0.5 – częstość zapisu geometrii w pliku `.arc` (w ps)

1 – wybór symulacji dla stałej energii

Po zakończeniu pracy programu mamy w pliku `lipid-beem-1.log` informacje o zmianach energii podczas symulacji. Możemy je wydobyć w sposób umożliwiający ich wykreślenie:

```
grep "Total E" lipid-beem-1.log > 1-b1.dat
```

a następnie usuwając zbędne kolumny w tym pliku, tak, by została tylko jedna kolumna z energią (w programie emacs prostokątny blok tekstu można usunąć stając w jednym jego narożniku, naciskając `Ctrl_Space`, przenosząc kursor do drugiego narożnika i naciskając kolejno `Ctrl_x r k`)

Analogicznie wykonujemy symulacje dla kroku 2 i 0.5 fs (pamiętając o usunięciu przed rozpoczęciem symulacji pliku `lipid.dyn` !):

```
rm lipid.dyn
rm lipid.arc
dynamic.x lipid 50000 2.0 0.5 1 > lipid-beem-2.log
rm lipid.dyn
rm lipid.arc
dynamic.x lipid 200000 0.5 0.5 1 > lipid-beem-05.log
```

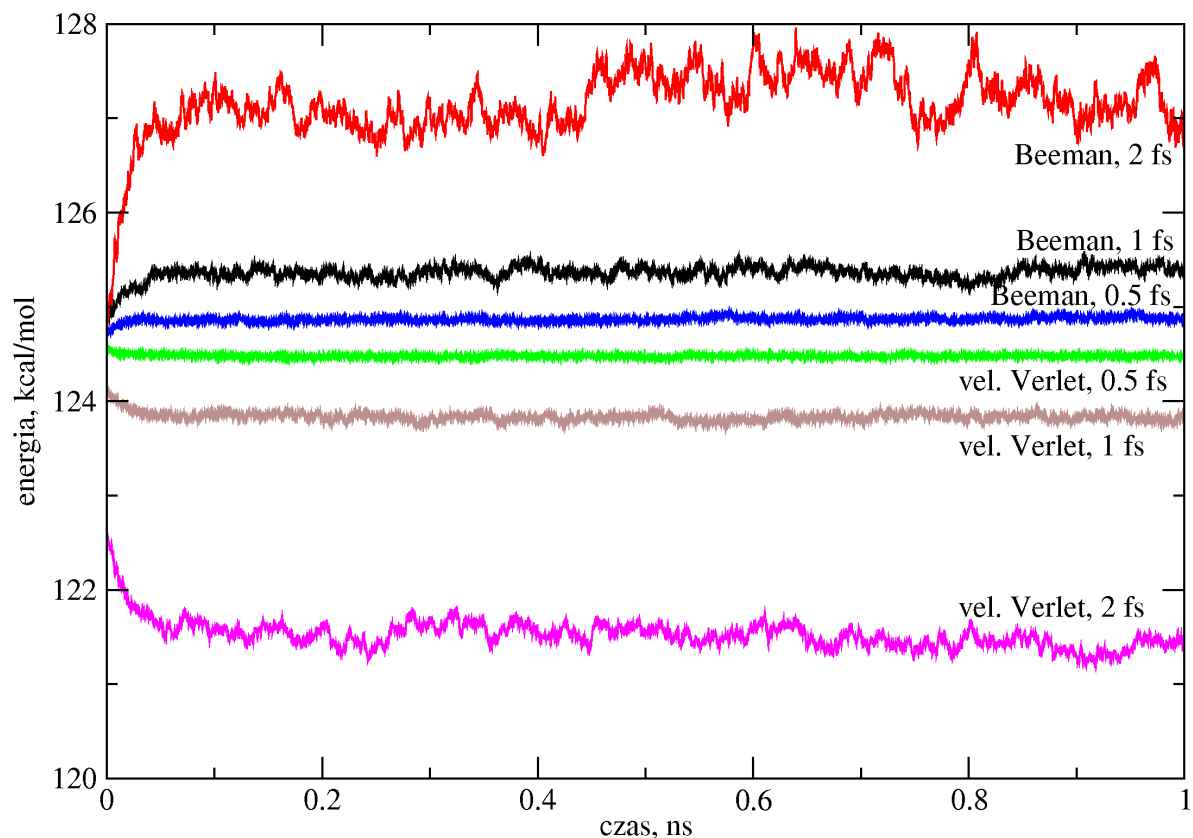
(Uwaga: nie uruchamiaj dwu symulacji równocześnie!)

Następnie zmieniamy parametr `integrate` w pliku `lipid.key` na `verlet` i powtarzamy operacje dla trzech czasów.

Po wydobyciu wartości energii do plików `.dat` możemy je wykreślić w programie `xmgrace`. Pamiętajmy, że używaliśmy różnych kroków czasowych, więc dla kroku czasowego 2 fs dane na osi *x* należy przemnożyć przez 2 a dla kroku 0.5 fs – podzielić przez 2. Robimy to

poleceniem *Data, Transformations, Evaluate expression*, zaznaczamy w obu okienkach żądany zbiór danych, zapisujemy w oknie *Formula* wzór $x=2*x$ lub $x=x/2$ i klikamy *Apply*.

Poniżej przedstawiono przykładowy wynik otrzymany dla całkowitego czasu symulacji 1 ns.



Widać, że w obu algorytmach fluktuacje średniej wartości energii maleją przy zmniejszaniu kroku czasowego symulacji a wyniki obu algorytmów zbliżają się do siebie. Zmniejsza się jednocześnie dryft energii w czasie.

6. Wyznacz współczynnik autodyfuzji acetonu w temperaturze 300 K.

Rozwiązanie.

Plik `aceton.xyz` zawiera informację o geometrii układu 150 cząsteczek acetonu a plik `aceton.dyn` dane o startowych położeniach i prędkościach (dzięki jego użyciu skrócimy czas dochodzenia do równowagi). Użyte pole siłowe to pole OPLS-AA.

Symulację przeprowadzimy w stałej objętości i temperaturze. Plik `aceton.key` podaje m.in. długość boku sześcianu, w którym znajduje się układ (używamy okresowych warunków brzegowych). Jest to 26.411 Å, co daje gęstość układu ok. 0.785 g/cm³.

Symulację uruchamiamy poleceniem

```
dynamic.x aceton 20000 1.0 0.2 2 300. > aceton.log &
```

Poszczególne liczby oznaczają:

20000 - liczbę kroków

1.0 – krok w fs

0.2 – odstęp (w ps) pomiędzy zapisami geometrii do pliku z trajektorią

2 – typ symulacji (tu: w stałej temperaturze i objętości)

300. – temperaturę termostatu w K

Po zakończeniu symulacji możemy obejrzeć trajektorię w VMD

```
vmd aceton.arc
```

Do wyznaczenia współczynnika autodyfuzji posłuży program `diffuse.x`, który na podstawie pliku `.arc` oblicza średnie przemieszczenia cząsteczek (z uwzględnieniem okresowych warunków brzegowych) a potem współczynnik dyfuzji ze wzoru Einsteina.

$$D = \lim_{t \rightarrow \infty} \frac{\langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle}{6t},$$

gdzie $\mathbf{r}(t)$ położenie cząsteczki w czasie t .

Wywołujemy program:

```
diffuse.x aceton > diff.dat
```

i wpisujemy następujące dane

```
1 100 1
```

```
0.2
```

(są to:

nr pierwszej i ostatniej geometrii do analizy oraz odstęp pomiędzy geometriami

odstęp czasu między geometriami w ps)

W pliku `diff.dat` można odczytać informacje, o które byliśmy pytani – gdybyśmy nie przekierowali wyjścia programu do pliku, zobaczylibyśmy te informacje na ekranie, ale zaraz potem zasypani zostalibyśmy serią liczb wypisanych wprost na ekran.

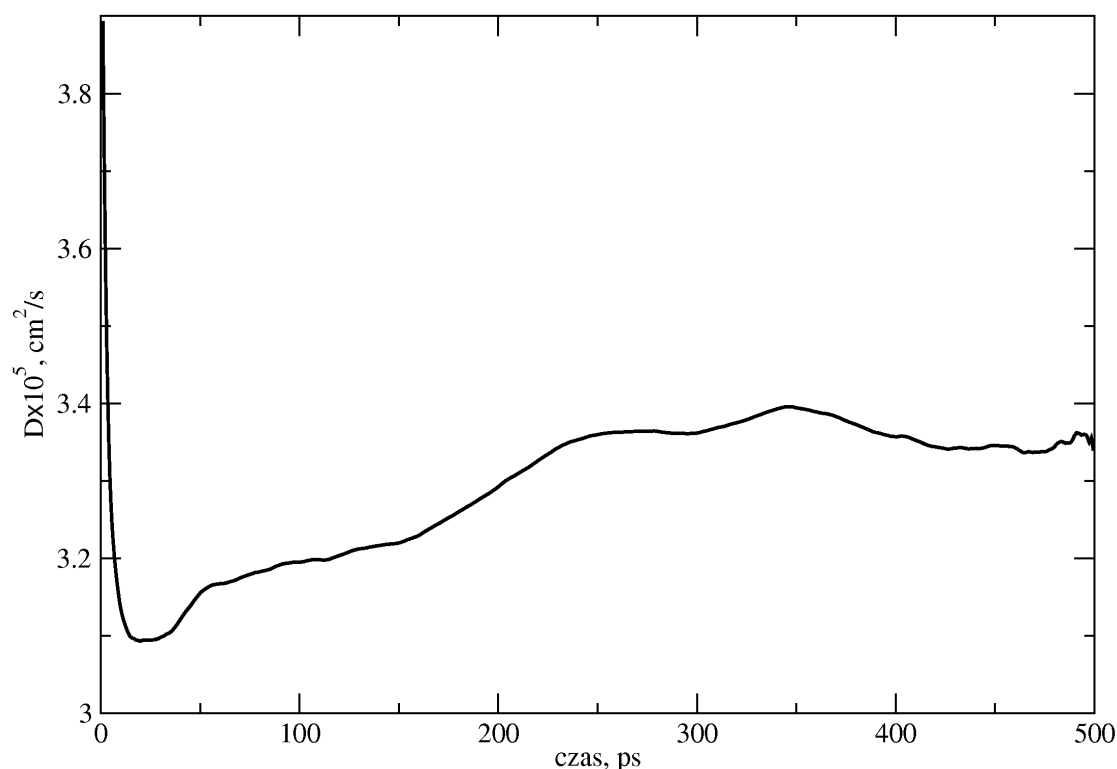
Plik `diff.dat` musimy opracować z użyciem edytora tekstu, aby usunąć początkowe linie niezawierające danych.

W programie `xmgrace` możemy teraz obejrzeć zależność 6 kolumny (D) od 1-szej (czas) wydając polecenia *Data*, *Import*, *ASCII*, wybierając nazwę pliku (`diff.dat`), ustawiając *Load as* na *Block data*, naciskając *OK*, wybierając *X from column 1*, *Y from column 6* i zatwierdzając przez *Accept*.

Na następnej stronie przedstawiono przykładowy wykres dla trajektorii zbieranej przez 0.5 ns (co 0.5 ps).

Ponieważ we wzorze występuje przejście z czasem do granicy w nieskończoności, powinniśmy zwracać uwagę na prawą stronę wykresu – czyli wartości obliczone dla dużej różnicy czasów. Z drugiej strony pamiętać należy, że wartości te zostały obliczone dla mniejszej liczby danych (skrajna wartość z tylko dwu geometrii – pierwszej i ostatniej). Widać, że współczynnik autodyfuzji wynosi ok. $3.3 - 3.4 \times 10^{-5} \text{ cm}^2/\text{s}$, co dobrze zgadza się z wartościami cytowanymi w pracy: *J. Phys. Chem. B*, **2009**, *113*, 11463.

Współczynnik dyfuzji dla acetonu wyznaczony w polu siłowym OPLS-AA:



7. Wyznacz z symulacji MD średnią liczbę koordynacji jonu Li^+ przez cząsteczki *N*-metyloacetamidu jako rozpuszczalnika.

Rozwiązanie.

Wykorzystamy symulację w stałej temperaturze i objętości (NVT) układu 125 cząsteczek *N*-metyloacetamidu i jednego jonu Li^+ (w bardziej realistycznym podejściu należałoby uwzględnić przeciwjony) w polu siłowym AMBER94.

Potrzebne pliki mają nazwy `nmali.*`.

Symulację uruchamiamy poleceniem

```
dynamic.x nmali 20000 1.0 0.2 2 300. > nmali.log &
```

(znaczenie argumentów opisano w zadaniu poprzednim).

Przeglądając w VMD plik trajektorii `nmali.arc` możemy stwierdzić, że jon Li^+ koordynowany jest przez atom tlenu z cząsteczki *N*-metyloacetamidu a liczba tych atomów waha się między 4 a 5.

Warto obejrzeć wykres funkcji rozkładu radialnego (*radial distribution function*, RDF) dla odległości Li^+ - O. Odpowiednie dane wygenerujemy za pomocą jednego z programów Tinkera. Musimy jednak wcześniej usunąć plik `nmali.key` (aby mieć możliwość zadania odległości) – po prostu zmienimy mu nazwę:

```
mv nmali.key nmali.KEY
```

Możemy teraz wygenerować funkcję rozkładu radialnego:

```
radial.x nmali.arc > distr.dat
```

Podajemy następujące dane:

```
1 100  
2003 343  
10  
0.01  
N
```

są to odpowiednio

nr pierwszej i ostatniej ramki danych

typy atomów (2003 to Li^+ a 343 O)

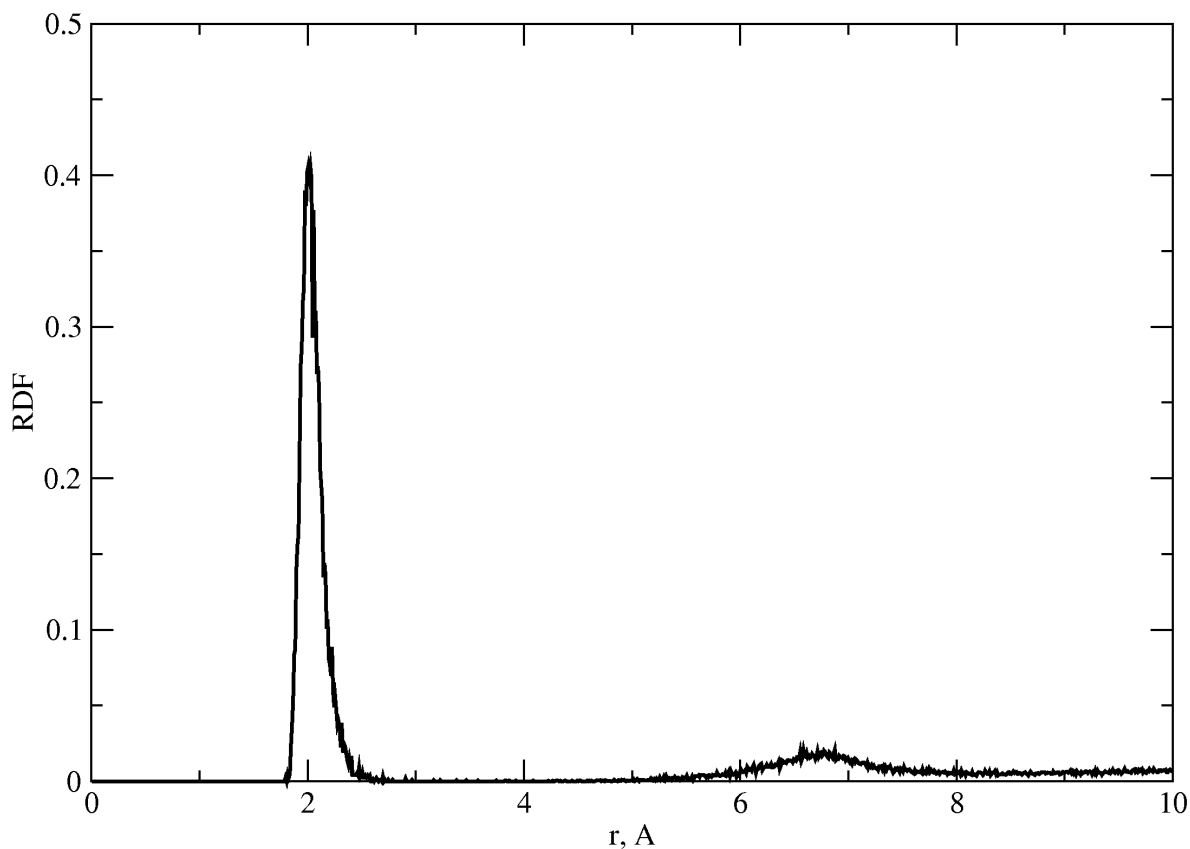
odległość (w Å), do której należy zliczać atomy

krok odległości (w Å)

N oznacza, by nie uwzględniać par atomów z tej samej cząsteczki (u nas to i tak niemożliwe)

Możemy teraz użyć programu xmgrace do wyświetlenia 3 i 4 kolumny danych: polecenia *Data*, *Import*, *ASCII*, wybieramy nazwę pliku (`distr.dat`), ustawiamy *Load as* na *Block data*, naciskamy *OK*, wybieramy *X from column 3*, *Y from column 4* i zatwierdzamy przez *Accept*.

Poniżej przedstawiono przykładowy wykres dla danych zbieranych przez 0.5 ns co 0.5 ps.



Jak widać, maksimum gęstości rozkładu radialnego jest przy ok. 2 Å. Odpowiada ono atomom tlenu znajdujących się w pierwszej sferze koordynacyjnej jonu Li^+ . Drugie, słabo zaznaczone maksimum przy ok. 7 Å odpowiada drugiej sferze koordynacyjnej.

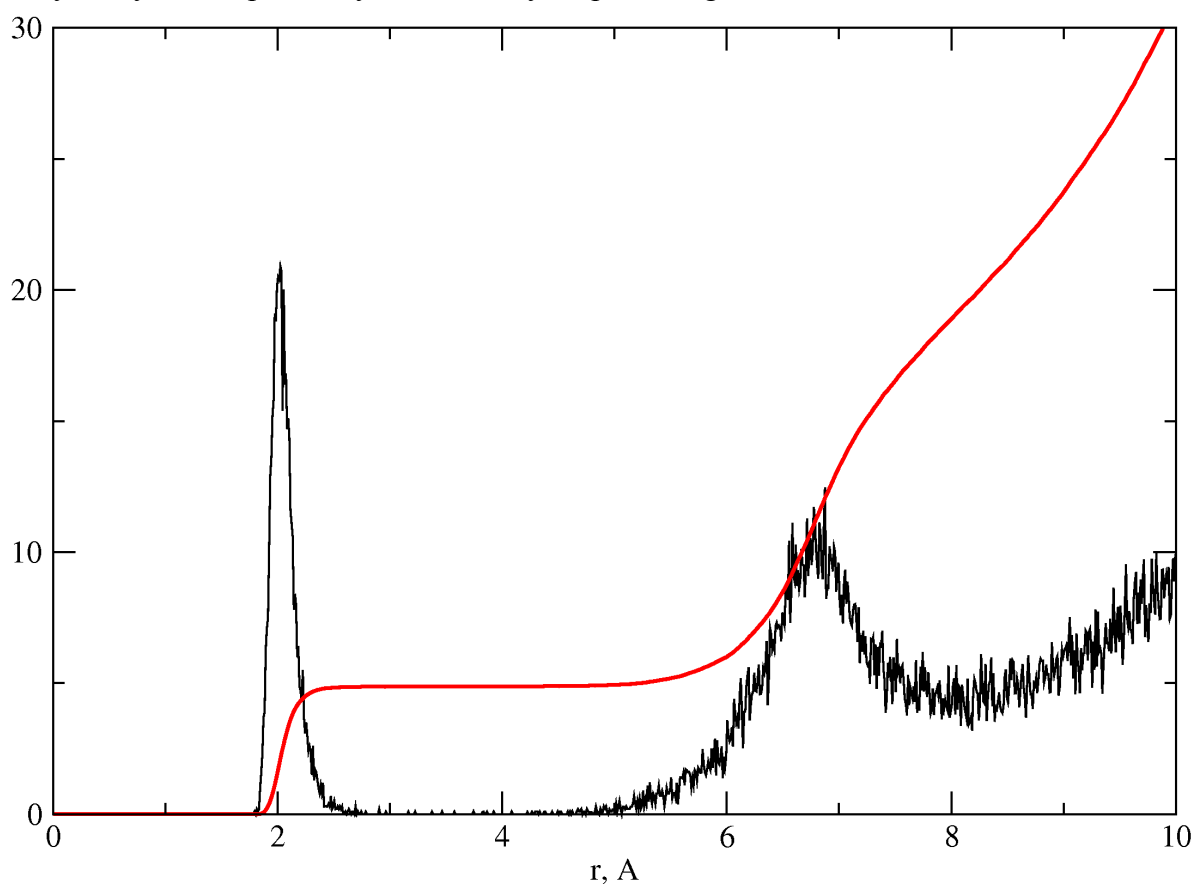
Aby ustalić, jak zmienia się liczba atomów O znajdujących się w odległości r od jonu Li^+ , musimy pamiętać, że funkcję RDF trzeba przemnożyć przez element objętości $4\pi r^2$:

Data, Transformations, Evaluate expression, zaznaczamy w obu oknach zestaw danych, wpisujemy w oknie *Formula* wzór $y=4*\pi*x^2*y$, zatwierdzamy przez *Accept*.

a później scałkować po odległości:

Data, Transformations, Integration, naciskamy *Accept*.

Otrzymamy w ten sposób wykres zbliżony do poniższego:



Czerwona linia pokazuje liczbę atomów O znajdujących się w kuli o promieniu r dookoła jonu. Dla odległości około 2 – 6 Å występuje plateau równe około 5, co odpowiada 5 atomom tlenu w pierwszej sferze koordynacyjnej jonu. Średnia liczba koordynacyjna jest nieco niższa niż 5.

8. Zbadaj, jaką gęstość wody otrzyma się z symulacji z użyciem modelu TIP3P cząsteczki wody w temperaturze 0, 25 i 60 °C pod ciśnieniem 1 atm.

Rozwiązanie.

Wykonamy symulację dynamiki w stałej temperaturze i pod stałym ciśnieniem (NPT) układu 500 cząsteczek wody (waterbox.xyz). Przykładowa parametryzacja TIP3P dla wody jest zapisana w pliku waterbox.key (dostarczany jako jeden z przykładów Tinkera):

```
parameters                      NONE
verbose
archive

a-axis                          24.662
ewald
thermostat                      BERENDSEN
tau-temperature                 0.1

vdwtype                        LENNARD-JONES
radiusrule                     GEOMETRIC
radiustype                     SIGMA
radiussize                     DIAMETER
epsilon                         GEOMETRIC
dielectric                      1.0

atom      1      0      "O Water (TIP3P)"      8      15.999      2
atom      2      H      "H Water (TIP3P)"      1      1.008      1
vdw        1      3.150656111      0.152072595
vdw        2      0.000      0.000
bond        1      2      529.6      0.9572
angle       2      1      2      34.05      104.52
ureybrad    2      1      2      38.25      1.5139
charge      1      -0.834
charge      2      0.417
```

W parametryzacji tej użyty jest człon Ureya-Bradley'a opisujący energię zginania wiązania (1) — (2) — (3) jako funkcję odległości między atomem (1) i (3):

$$v(r_{13}) = \frac{k_{13}}{2} (r_{13} - r_{(0)13})^2$$

Przeprowadzamy symulację dla temperatury 0 °C (czyli 273 K) i ciśnienia 1 atm:

```
dynamic.x waterbox 50000 1. 0.5 4 273. 1. > waterbox.273.log &
```

Wartości liczbowe oznaczają po kolei: liczbę kroków (20000), krok (1 fs), odstęp czasu między zapisami trajektorii (0.5 ps), rodzaj symulacji (4 oznacza NPT), temperaturę (273 K) i ciśnienie (1 atm).

Analogicznie prowadzimy symulacje dla temperatury 25 i 60 °C (298 i 333 K):

```
dynamic.x waterbox 50000 1. 0.5 4 298. 1. > waterbox.298.log &
dynamic.x waterbox 50000 1. 0.5 4 333. 1. > waterbox.333.log &
```

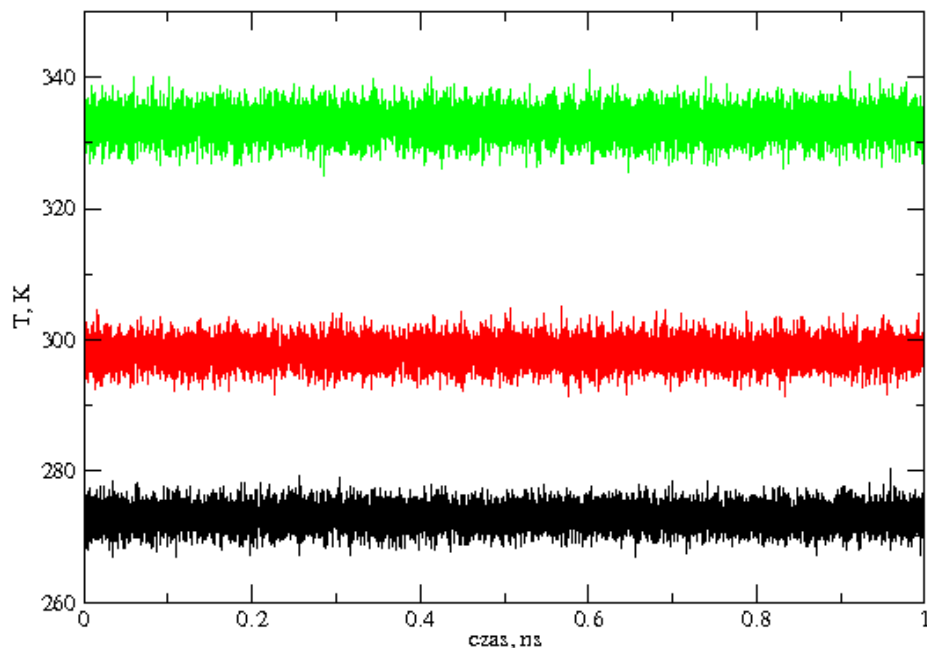
(Uwaga: nie należy uruchamiać tych symulacji jednocześnie !)

Po zakończeniu symulacji możemy wydostać z logów informacje o gęstości układu i temperaturze, np.:

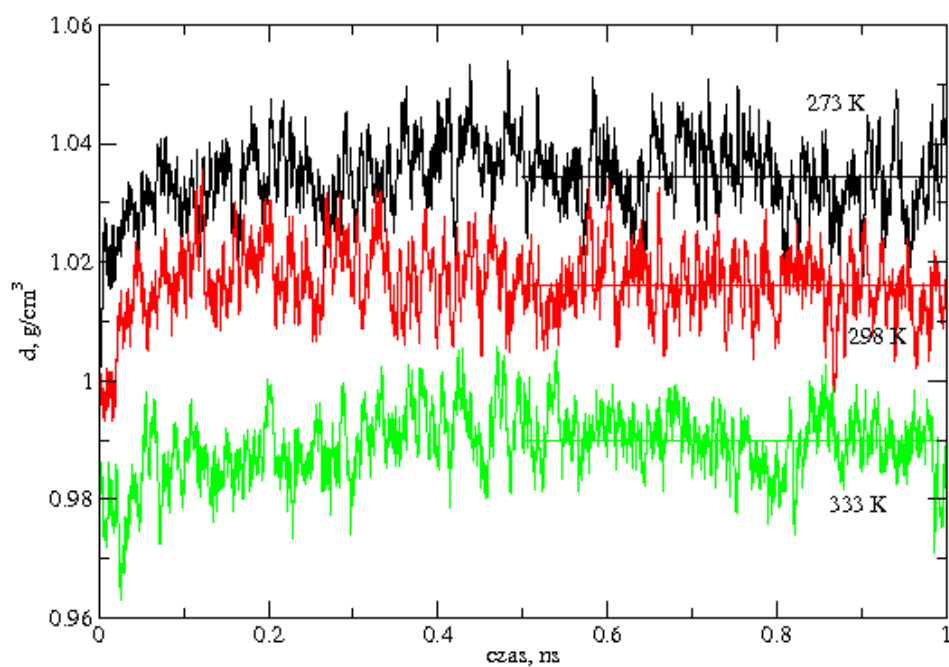
```
grep "Density" waterbox.273.log > den-273.dat  
grep "Temperature" waterbox.273.log > temp-273.dat
```

Edytując te pliki należy zostawić w nich tylko kolumnę liczb (wycinanie prostokątnego bloku tekstu w programie emacs – patrz przykład 5).

Po wydobyciu danych dla wszystkich temperatur możemy sprawdzić, czy rzeczywiście średnia temperatura utrzymywała się na zadanym poziomie wykreślając odpowiednie dane, jak na poniższym przykładzie dla 1 ns symulacji:



Z wykresu wynika, że średnia temperatura w czasie symulacji utrzymywała się na zadanym poziomie. Zmiany gęstości przedstawione są na następnym wykresie:



Początkowo obserwowane są systematyczne zmiany gęstości a po dojściu układu do stanu równowagi obserwuje się fluktuacje względem średniej wartości. Liniami poziomymi zaznaczono wartości uśrednione po końcowych 0.5 ns symulacji.

Uzyskane wartości gęstości to: 1.034 g/cm³ dla 273 K, 1.016 g/cm³ dla 298 K oraz 0.990 g/cm³ w 333 K. Odpowiednie dane tablicowe (Wikipedia wg CRC Handbook of Chemistry and Physics) to 0.9998, 0.997 i 0.983 g/cm³. Dla wszystkich temperatur wynik jest zawyżony, najbardziej dla temperatury 273 K. Niższe wartości z użyciem tego pola siłowego można uzyskać prowadząc symulację dla cząsteczek wody traktowanych jako ciała sztywne (zamrożona geometria równowagowa); w testowej symulacji otrzymano 0.998 g/cm³ w 273 K, 0.978 g/cm³ w 298 K i 0.944 g/cm³ w 333 K, jak widać tym razem gęstość w miarę wzrostu temperatury jest coraz bardziej zaniżona.

9. Rozważmy łańcuch Markowa związany ze zmianami cen akcji na giełdzie. Załóżmy, że giełda może być w 3 stanach: hossy (H), neutralnym (N) i bessy (B). Jeśli w danym dniu panuje hossa, to w dniu następnym hossa wystąpi z prawdopodobieństwem 0.6, stan neutralny i bessa z prawdopodobieństwami 0.2. Jeśli giełda jest w stanie neutralnym, w następnym dniu prawdopodobieństwa stanów H, N i B wynoszą odpowiednio 0.3, 0.4 i 0.3. W dniu następującym po bessie prawdopodobieństwa stanów są równe H=0.2, N=0.3, B=0.5.

a) W dniu dzisiejszym na giełdzie panuje hossa. Jakie jest prawdopodobieństwo znalezienia się giełdy w każdym ze stanów pojutrze, za trzy dni, za siedem sesji.

b) Jaki jest równowagowy rozkład prawdopodobieństwa.

Rozwiązanie.

Do obliczeń wykorzystamy program R.

a) Elementy wektora stanu będą oznaczać kolejno H, N i B.

Macierz przejścia ma postać

$$P = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

a rozkład prawdopodobieństwa stanów giełdy w dniu dzisiejszym jest równy

$$p_0 = [1 \ 0 \ 0]$$

(panuje hossa, czyli występuje stan pierwszy).

Definiujemy odpowiednie wielkości w programie R:

```
P=matrix(c(0.6,0.2,0.2,0.3,0.4,0.3,0.2,0.3,0.5),c(3,3),byrow=T)
```

```
      [,1] [,2] [,3]
[1,] 0.6 0.2 0.2
[2,] 0.3 0.4 0.3
[3,] 0.2 0.3 0.5
p0=c(1,0,0)
```

i możemy obliczyć prawdopodobieństwo wystąpienia stanów giełdy za dwa i trzy dni:

```
(p1=p0%*%P)
      [,1] [,2] [,3]
[1,] 0.6 0.2 0.2
```

```
(p2=p1%*%P)
      [,1] [,2] [,3]
[1,] 0.46 0.26 0.28
```

```
(p3=p2%*%P)
      [,1] [,2] [,3]
[1,] 0.41 0.28 0.31
```

Po drodze policzyliśmy rozkład prawdopodobieństwa w dniu następnym, równy po prostu pierwszemu wierszowi macierzy **P**.

Rozkład prawdopodobieństwa po 7 sesjach uzyskamy przez kolejne mnożenia

```
p=p0
for (i in 1:7) p=p%*%P
p
      [,1] [,2] [,3]
[1,] 0.3823 0.290725 0.326975
```

Prawdopodobieństwa są dość zbliżone, ze wskazaniem na hossę.

b) Rozkład stacjonarny w naszym łańcuchu Markowa możemy znaleźć wyznaczając macierz będącą granicą wyrażenia $\mathbf{P}^n$ przy n dążącym do nieskończoności. Naiwną i siłową metodą może być próba wymnożenia macierzy przez siebie wiele razy:

```
Pi=P
for (i in 1:10000) Pi=P %*% Pi
Pi
      [,1] [,2] [,3]
[1,] 0.3818182 0.2909091 0.3272727
[2,] 0.3818182 0.2909091 0.3272727
[3,] 0.3818182 0.2909091 0.3272727
```

Do obliczenia potęgi macierzy lepiej wykorzystać diagonalizację. Jeśli bowiem macierz **A** da się sprowadzić do postaci diagonalnej $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ przez podobieństwo:

$$\mathbf{D} = \mathbf{P}^{-1} \mathbf{A} \mathbf{P},$$

to n -tą potęgę **A** możemy obliczyć następująco:

$$\mathbf{A}^n = \underbrace{\mathbf{A} \mathbf{A} \mathbf{A} \dots \mathbf{A}}_{n \text{ razy}} = \underbrace{\mathbf{P}(\mathbf{P}^{-1} \mathbf{A} \mathbf{P})(\mathbf{P}^{-1} \mathbf{A} \mathbf{P}) \dots (\mathbf{P}^{-1} \mathbf{A} \mathbf{P})}_{n \text{ razy}} \mathbf{P}^{-1} = \underbrace{\mathbf{P} \mathbf{D} \mathbf{D} \mathbf{D} \dots \mathbf{D} \mathbf{P}^{-1}}_{n \text{ razy}} = \mathbf{P} \mathbf{D}^n \mathbf{P}^{-1}$$

Przypomnijmy też, że macierz **P**, to macierz wektorów własnych macierzy **A** i że rzeczywista macierz niesymetryczna (a taka jest macierz przejścia), która ma wszystkie wartości własne różne, jest diagonalizowalna.

Reasumując, potęgę macierzy obliczamy diagonalizując ją, podnosząc zdiagonalizowaną macierz do potęgi (co jest łatwe, ponieważ sprowadza się do podniesienia do potęgi elementów diagonalnych) i poddając wynik transformacji odwrotnej do diagonalizującej.

Zatem diagonalizujemy naszą macierz przejścia **P**:

```
eigen(P)
$values
[1] 1.0000000 0.3618034 0.1381966

$vectors
      [,1] [,2] [,3]
[1,] -0.5773503 -0.7083577 0.07663858
[2,] -0.5773503 0.1550220 -0.78793294
```

```
[3,] -0.5773503  0.6886200  0.61097316
```

i tworzymy macierz wektorów własnych oraz macierz odwrotną do niej:

```
v=eigen(P)$vectors
v1=solve(v)
```

Wartości własne macierzy **P** to 1, 0.3618 i 0.1382. Ich n -te potęgi przy n dążącym do nieskończoności dążą zatem do 1, 0 i 0. n -ta potęga zdiagonalizowanej macierzy dąży zatem do

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Wykorzystując ten wynik, dostajemy ostatecznie po pomnożeniu przez **v** i **v1** granicę potęgi **Pⁿ**:

```
Pn=matrix(c(1,0,0,0,0,0,0,0,0),c(3,3))
Pn
```

```
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    0    0
[3,]    0    0    0
```

```
(Pinf=v %*% Pn %*% v1)
```

```
      [,1] [,2] [,3]
[1,] 0.3818182 0.2909091 0.3272727
[2,] 0.3818182 0.2909091 0.3272727
[3,] 0.3818182 0.2909091 0.3272727
```

Wynik jest taki sam, jak w metodzie naiwno-siłowej.

Ponieważ

$$\lim_{n \rightarrow \infty} \mathbf{P}^n = \mathbf{1}\pi,$$

gdzie π jest rozkładem w stanie stacjonarnym, każdy wiersz otrzymanej macierzy podaje rozkład prawdopodobieństwa w stanie stacjonarnym.

Wynika stąd, że prawdopodobieństwo wystąpienia hossy po dostatecznie długim czasie to ok. 0.38, stanu neutralnego 0.29 a bessy 0.33.

Widać, że znajdując rozkład prawdopodobieństwa po siedmiu sesjach byliśmy już blisko opisu stanu stacjonarnego.

Sprawdźmy jeszcze, że rzeczywiście jest to stan stacjonarny, czyli nie zmieni się po zadziałaniu nań macierzą przejścia:

```
(pe=Pinf[1,])
[1] 0.3818182 0.2909091 0.3272727
(pe %*% P)
      [,1] [,2] [,3]
[1,] 0.3818182 0.2909091 0.3272727
```