

Основы МД: Анализ радиальных функций распределения

Автор *gorkovets*

Создано 07/15/2011 - 15:14

Опубликовано gorkovets в Пт, 07/15/2011 - 15:14

Описание задачи

Данная задача посвящена построению функции радиального распределения после расчета молекулярной динамики в микроканоническом ансамбле.

Цель.

Изучение работы базовых алгоритмов молекулярной динамики на примере моделирования инертного газа (аргона), построение радиальной функции распределения. Расчеты осуществляются при помощи программного пакета *Gromacs*.

Подготовка к работе

Для выполнения задачи необходим компьютер, работающий под *OS Linux*, либо родственными ОС. Расчеты молекулярной динамики рекомендуется выполнять на высокопроизводительных компьютерных кластерах.

Необходимое ПО :

1 *Gromacs* - осуществляет расчеты молекулярной динамики (в данной статье использована версия 4.5.4) как в обычной, так и в MPI версиях (для распределенных вычислений на кластере)

2 *Gnuplot* - используется для построения графиков (в статье рассмотрена версия 5)

3 *LAM-MPI* - драйвер, необходимый для распределенных вычислений

Для удобства работы рекомендуем использовать файловые менеджеры, например, *Midnight Commander*.

Описание системы и ее подготовка.

Система представляет собой инертный газ аргон в коробке 10x10x10 нм.

Скачайте архив в интересующую Вас директорию

```
wget http://molsim.org/sites/default/files/tutor01.tar\_0.gz [1]
```

Распакуйте его

```
tar -xvf file.tar.gz
```

Создаем систему с помощью файла *argongen.pl*, в качестве аргумента выступает необходимое количество атомов. *Argongen.pl* создает .gro-файл и файл топологии (*.top).

Например, мы вводим

```
./argongen.pl 1000
```

На выходе мы получим файлы *argon1000.gro* и *topol1000.top*

Рассмотрим скрипт *Argongen.pl*, создающий эти файлы.

```
#!/usr/bin/perl
$number=$ARGV[0];      #????????? ????????, ? ?????? ?????? ??? 1000
use POSIX;              #??????????? ?????? ?????? ? ???????

#????????? ?????? ???? ? ??????????????, ???? ??????? ?? ??????:
$box_size=POSIX::ceil($number**(1/3));
printf ("The size of the cub's edge = $box_size\n");

#????????? ?????????????? ???????????? ??????? ??? ?????????????????? ??????:
$num=($box_size**3);
printf ("corrected number of atoms = $num\n");

#????????? ? ?????????? gro-????? ?? ??????? ? ?????? ??????????????? ???????
{open(TE,">argon$num.gro") || die "Can't create the main file";
$i=1;$h=0;$r=1;
print(TE "ARGON\n");
print(TE "$num\n");

#????????? ???? ???????????? ?? ??????? ? ???? ??????????????? ???????
open(TT,">topol$num.top") || die "I can't create topology file";#?????????
print TT ("\n#include \"oplsaa.ff/forcefield.itp\" \n");
print TT (" \n");
print TT ("[ moleculetype ]\n");
print TT ("; Name          nrexcl\n");
print TT ("ARGON          5 \n");
print TT (" \n");
print TT (" \n");

#????? ??? ???????????? ??????????
for ($xi=0; $xi<=($box_size-1); $xi++)
{for ($yi=0; $yi<=($box_size-1); $yi++)
{for ($zi=0; $zi<=($box_size-1); $zi++)
{
$l="GRO";          #??? ???????
$ar="Ar";          #??? ??????
$c=0;              #?????
$m=40;             #?????
$x=($xi/$box_size*10); #????????????? ? ?????????????????? ? ???????????????
$y=($yi/$box_size*10);
$z=($zi/$box_size*10);
if ($r==10){$h=$h+1;$r=1}else{$r=$r+1}

#??????? ??????? ? ???? ? ??????? ???????
print TE sprintf "%5u%5s%5s%5u%8.3f%8.3f%8.3f\n",$h,$l,$ar,$i,$x,$y,$z;
print TT sprintf "%5u%5s%5u%6s%5s%6u%5u%5u\n",$i,$ar,$h,$l,$ar,$i,$c,$m;
$i=$i+1}}}}

#????????? ??????????????? ?????? ? ???? ?????????????
print TT ("[ system ]\n");
print TT ("; Name\n");
print TT ("ARGON\n");
print TT (" \n");
print TT ("[ molecules ]\n");
print TT ("; Compound          #mols\n");
print TT ("ARGON          1\n");
close(TT);          #??????????? ???????
```

```
close(TE);  
}
```

Формат gro-файла:

```
?????_??????? ????_????? ???_????? ????_????? ?????????_x_y_z
```

Формат файла топологии:

```
?????_????? ???_????? ?????_??????? ????????_??????? ???? ?????????_????? ????  
? ?????
```

Поместите наш газ в необходимую коробку:

```
editconf -f argon1000.gro -o argon1000_newbox.gro -c -box 10 10 10 -bt cubic
```

Будет использован файл *argon1000.gro*, запись данных пойдет в *argon1000_newbox.gro*, опция -c говорит, что система будет помещена в центр коробки, -box позволяет установить размеры коробки, -bt cubic задает форму коробки, в данном случае — куб.

Затем необходимо подготовить систему, а именно — минимизировать ее энергию. Для этого используйте следующую команду:

```
grompp -f en_minim.mdp -c argon1000_newbox.gro -p topol1000.top -o em1000.tpr
```

Опция -f указывает на файл, в котором прописаны параметры действия, совершаемого над системой, -c входной файл системы, -p файл топологии, -o выходной файл.
Для запуска МД используется:

```
mddrun -v -deffnm em1000
```

Опция -v принуждает печатать всю возможную информацию в лог, -deffnm общее имя группы файлов, которые мы используем.

Далее запустите МД в NVE-ансамбле, то есть сохраняется количество вещества, объем и энергия системы. Делается это аналогично минимизации энергии системы.

```
grompp -f nve_conf.mdp -c em1000.gro -p topol1000.top -o nve1000.tpr  
mddrun -deffnm nve1000
```

Построение функции радиального распределения.

В ходе выполнения команд были получены файлы траектории. На основании них можно построить функцию радиального распределения системы.

Для этого необходимо сначала создать .tpr-файл с топологией:

```
grompp -f nve_conf.mdp -c argon1000_newbox.gro -p topol1000.top -o topol1000.tpr
```

Для построения функции радиального распределения используется следующая команда:

```
g_rdf -f nve1000.trr -s topol1000.tpr -o rdf1000.svg
```

Опция -f указывает на входной файл траектории, -s на файл топологии, -o на выходной файл с графиком.

При выполнении команды появится запрос о необходимости выбора группы атомов, используйте группу *System (0)*, на второй запрос ответьте так же.

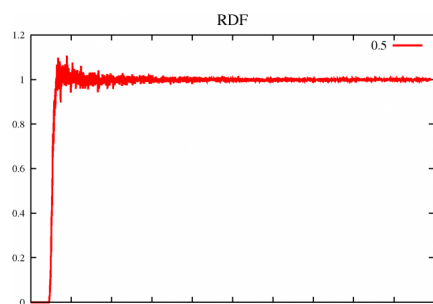
На выходе будут получены файлы в формате xvg, чтобы получить из них файлы в формате tif необходимо:

```
gnuplot gnuplotenergy1.scr  
./eps_to_img.sh
```

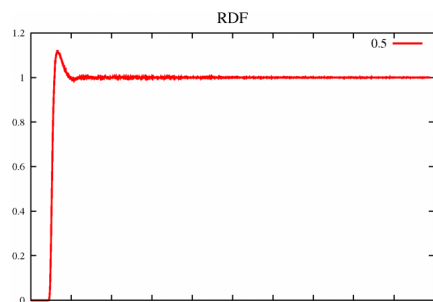
gnuplotenergy1.scr находится в папке RDF.

Для того, чтобы построить графики функции радиального распределения для системы с другим числом атомов необходимо изменить названия соответствующих файлов в скрипте.

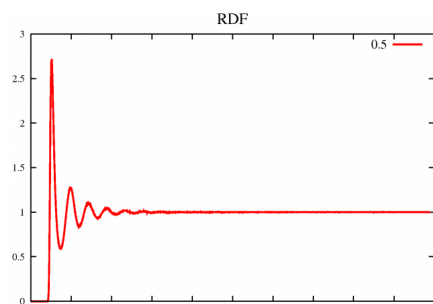
Функция радиального распределения для системы с 1000 атомов аргона



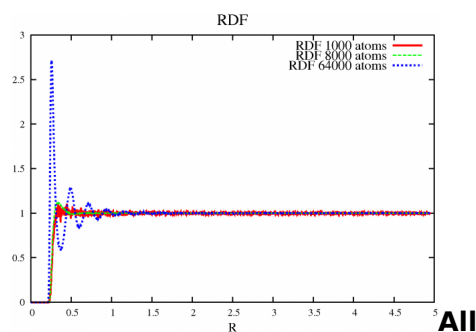
Функция радиального распределения для системы с 8000 атомов аргона



Функция радиального распределения для системы с 64000 атомов аргона



Общий график для всех трех систем



На общем графике видно, что все три системы имеют разные функции радиального распределения и на разных r достигают единицы. С увеличением плотности ρ увеличивается, то есть увеличивается расстояние, на котором влияние частиц друг на друга отсутствует.

Прикрепленный файл

Размер

[tutor01.tar.gz](#) [2]

27.36 кб

Источник: http://www.molsim.org/ru/methods/MD_basics_3

Ссылки:

[1] http://molsim.org/sites/default/files/tutor01.tar_0.gz

[2] http://www.molsim.org/sites/default/files/tutor01.tar_0.gz