Joint ICTP-TWAS Caribbean School on Electronic Structure Fundamentals and Methodologies

27 August-21 September, Cartagena, Colombia

Tutorial on Ab initio Molecular Dynamics

Coordinator: Eduardo Menéndez Proupin

Outline
Exercise 1. Born-Oppenheimer molecular dynamics with Pwscf (pw.x).
Exercise 2. Car-Parrinello molecular dynamics with CP (cp.x)
Exercise 3. Calculation of averages and statistical errors.

Download the exercise

```
scp  student@172.30.11.211:Tutorials/Exercise_7_MD/exercise_7_md.tar.gz .
```

Exercise 1. Born-Oppenheimer Molecular Dynamics

Add the Quantum ESPRESSO bin directory to the PATH variable, e.g.

```
PATH=$PATH:$HOME/ChemUtils/Espresso/espresso-5.0.1/bin/
export PATH
```

Go to directory ex1-si8

List the directory. You should see a file si.md8.in

Run the MD calculation,

```
pw.x < si.md8.in >si.md8.out
```

what happened? Examine the output file si.md8.out

Solution/ Find out the directory where the pseudopotential file
(Si.pz-vbc.UPF) is located, and fix the line pseudo_dir = ...

If the internet doee not work yet,   `cp ../ex2-si8-cp/Si.pz-vbc.UPF .`

Run again, and see the output file. Did it end correctly ?

Visualize the run with

```
xcrysden --pwo si.md8.out
```

Now let us do a longer run

```
cp si.md8.in si.md8.mod1.in
```

edit the new input file and set

```
                   nstep   = 1000,
                  tstress = .true.,
```

read espresso-${VERSION}/Doc/INPUT_PW.txt
to see the meaning and where to place the keywords.

Question: What are the time step and the simulation time in femtoseconds ?

Run

```
pw.x < si.md8.mod1.in >si.md8.mod1.out
```

Examine the output file.
Why has the simulation stopped before doing the 1000 steps ?
Look at the line near the end of `si.md8.mod1.out`

```
     convergence NOT achieved after 100 iterations: stopping
```

what has happened ?
How many steps did run?

A frequent reason for not achieving self-consistent field (SCF) convergence is that the system is in a configuration where it is metallic. Then we need to use the smearing technique

In the name list `&system`, add the following options

```
occupations = 'smearing',
   smearing = 'gaussian',
     degauss = 0.01,
```

Discuss the above choice. Questions: How to select the value of degauss ? Why to smearing='gaussian' and an alternative.

Run again. Dit it run? Has it sense?

It the simulation has run to the end we can go on.

Let us analyze the results of the dynamics. For this we will use the script analysis_pw.sh . Make sure that the first two lines are

```
INFILE=si.md8.mod1.out
SUFFIX=mod1.dat
```

and run

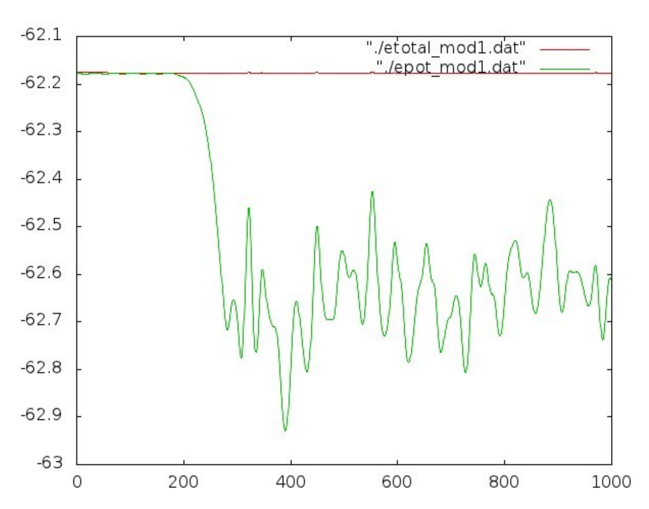
```
bash analysis_pw.sh
```

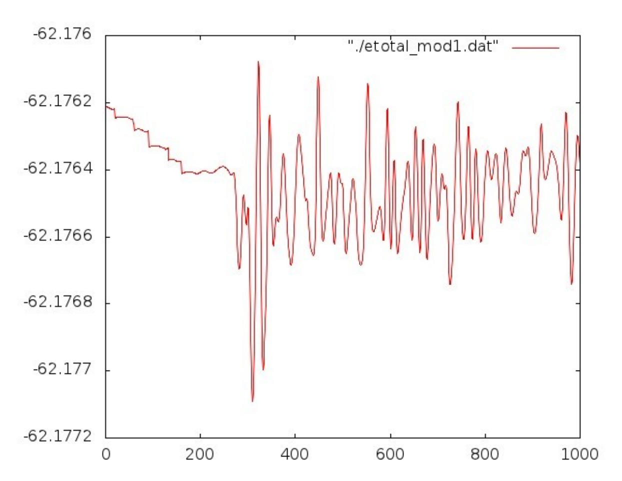
The following files must exist now:

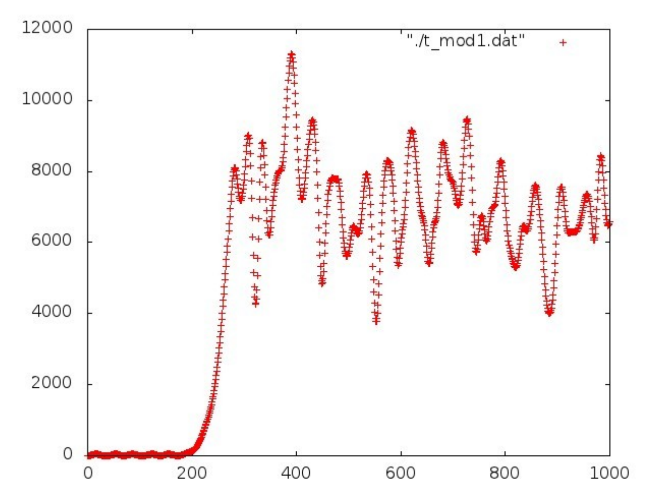
ekin_mod1.dat epot_mod1.dat etotal_mod1.dat p_mod1.dat t_mod1.dat

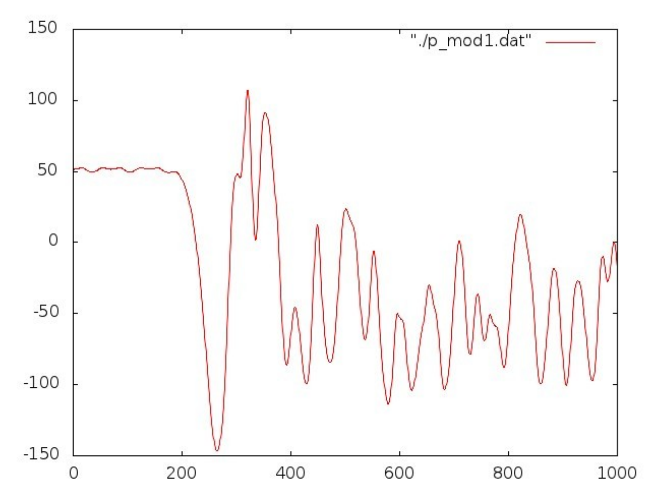
these files contain the kinetic energy, potential energy, their sum (total enegy), the pressure ant the temperature, for every step of the simulation.

Use gnuplot (or your favorite plotting program) to plot the variables. You should see something like the following









Question:
Look at the fluctuations of the total energy alone, and together with the potential energy. Check if there is drift in the total energy. Can you said that the energy is reasonably conserved ?

What do you think on the stabilized average temperature and pressure.

Do you think the calculated energies and forces are correct? What do you think of the parameters, e.g.
cutoff
k-points grid
degauss
the number of atoms and supercell size or other parameters ?

Let us experiment with some parameters. Create a new input file

```
mv si.md8.mod1.in si.md8.mod2.in
```

edit the new input file and set

```
K_POINTS {automatic}
 4 4 4 1 1 1
```

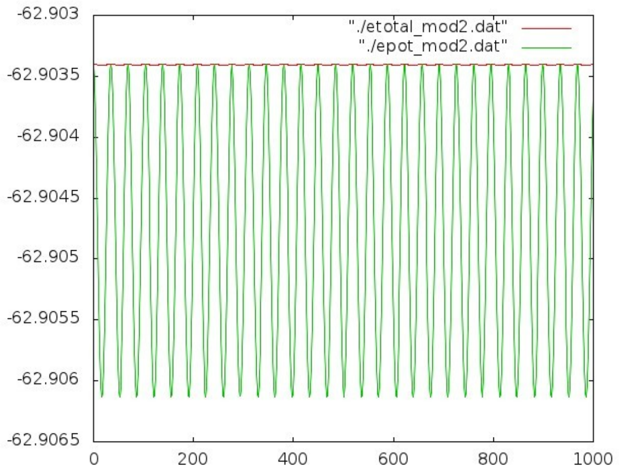
and run the simulation. Use paralellism if your computer allows it, e.g.,

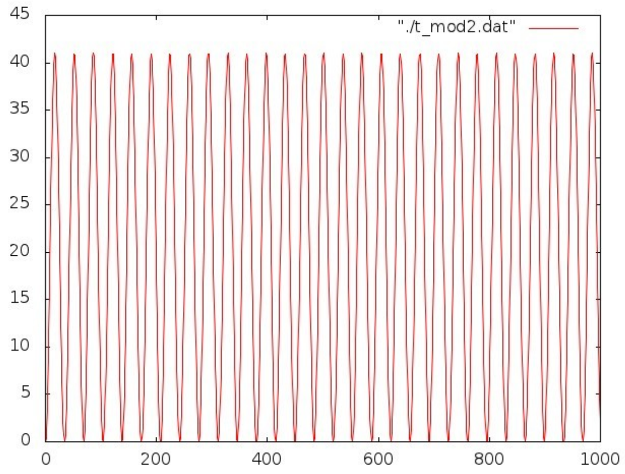
```
mpirun -np 2 pw.x < si.md8.mod2.in > si.md8.mod2.out
```

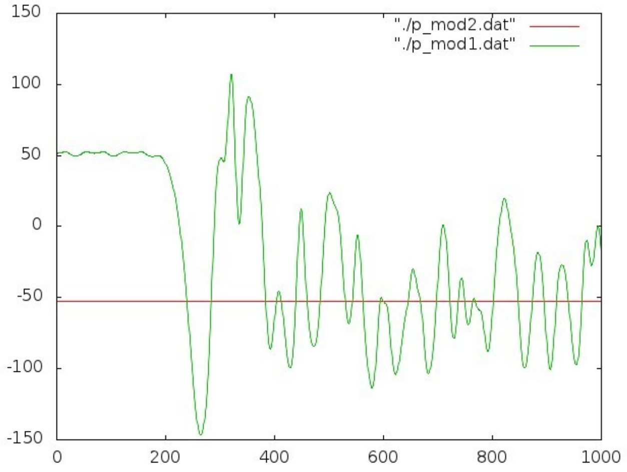
verify if it worked as expected. Look at the final temperature.

edit the script `analysis_pw.sh` and change mod1 by mod2, and run the script.

Then plot the temperatures, energies, and pressures and compare them with the previous simulation. You should see somethinglike this







It is evident that the physical result is quite different. Now the system keeps in oscillating around the zinc-blende structure, as expected for silicon. If you have time, repeat the simulation with an increased cutoff (ecutwfc), and use occupations = 'fixed'.

Now you can use the following formula to estimate the heat capacity per particle C_V (remember that *N=8* is far from the thermodynamics limit).

$$\frac{\langle (\delta K)^2 \rangle}{\langle K \rangle} = N k_B T \left(1 - \frac{3 k_B}{2 C_V} \right)$$

Is the result what you expect? Is it close to the Debye law or to the Dulong-Petit law ? Is it correct?

Why in this case cannot be used the well known formula

$$\langle (\delta E)^2 \rangle = N k_B T^2 C_V$$

Exercise 2. Car-Parrinello molecular dynamics.

Check that cp.x can be executed from your work directory. If you cannot, check if cp.x is compiled and installed in the Quantum ESPRESO binary directory and this directory is in the PATH variable. As in exercise 1, do

```
PATH=$PATH:$HOME/ChemUtils/Espresso/espresso-5.0.1/bin/
export PATH
```

First step, obtain the ground step

```
cp.x < si.cp8.in.1 > si.cp.out.1
```

Second step, run the dynamics. Create a new input file

```
cp  si.cp8.in.1  si.cp8.in.2
```

edit `si.cp8.in.2`, and change the options

```
                    nstep = 1200
          electron_dynamics='verlet'
               ion_dynamics='verlet'
```

note that different to `pw.x`, in restart `cp.x` runs `nstep` steps. In contrast, for `pw.x` nstep is the final step of the simulation. Restart the simulation with

```
cp.x < si.cp8.in.2 > si.cp.out.2
```

note `iprint=8`, this was made to print the simulation parameters in the file `cp.evp` for the same times of the exercise 1.
Recall the definition of dt (from `…../Doc/INPUT_CP.txt`)

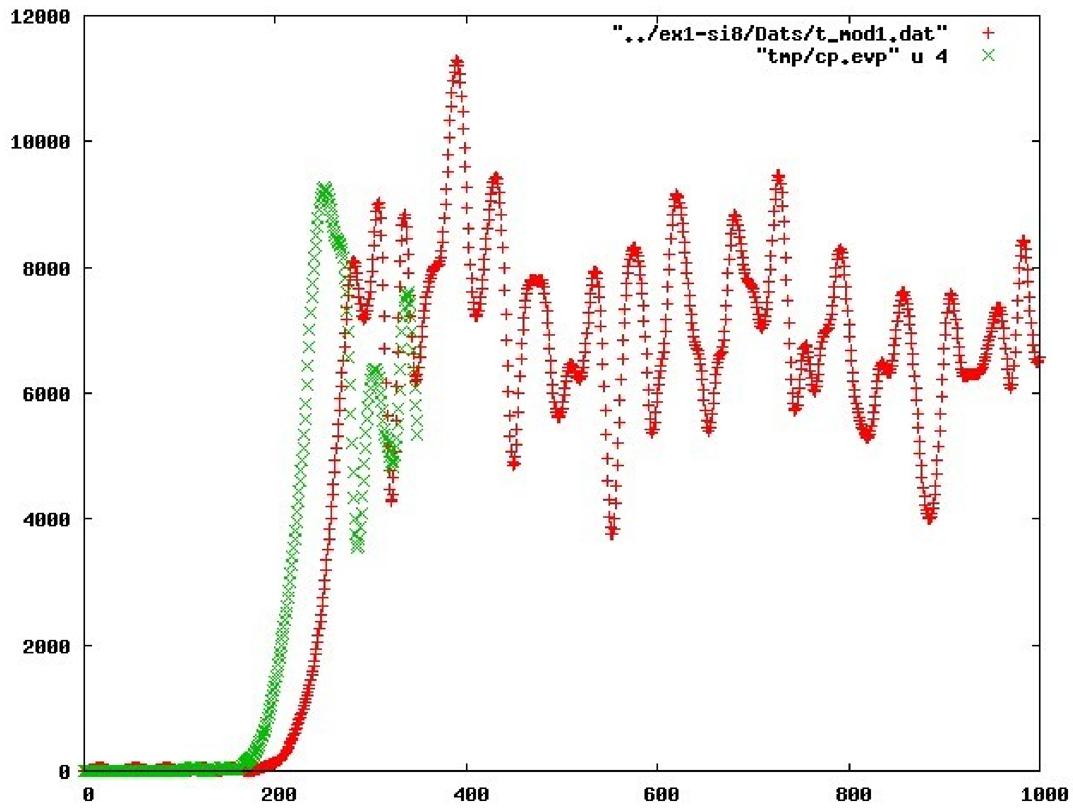
```
Variable:      dt

Type:          REAL
Default:       1.D0
Description:   time step for molecular dynamics, in Hartree atomic units
               (1 a.u.=2.4189 * 10^-17 s : beware, PW code use
                Rydberg atomic units, twice that much!!!)
```

Hence, to compare easily the results,

	dt	timeunit	dt*timeunit
CP	5	0.024 fs aprox	0.12 fs
PW	20	0.048 fs aprox	0.96 fs

Hence, with iprint=8 for CP, the variables are printed every 8*0.12=0.96 fs, as with PW.

With gnuplot, the temperatures with CP and with PW can be compared using the command



plot "../ex1-si8/Dats/t_mod1.dat","tmp/cp.evp" u 4

It is readily verified that CP and PW evolve the system to the same configuration. The difference in the time of rising of the temperature can be adscribed to the small difference in the electronic state of both methods, combined with the chaotic character of molecular dynamics trajectories.

The program CP does not allows k-points sampling (only the Γ point, k=0), hence we cannot use the 4x4x4 mesh and compare directly with the PW results. Let us note that in order to obtain meaningful results, what is really nead (with both codes) is to use a 4x4x4 supercell (with 256 atoms) or larger with only Γ point.

What program to use is in part a matter of taste. Both codes should produce the same physical results. For insulators with a large gap, the emass parameter can be chosen relatively large (but not much) and a long time step, and the CP simulation may run faster than BO with PW. For metals, CP can be used with a thermostat to keep cool the electronic system, this needs to set more parameters and the BO method is simpler. In general, CP method needs more care and fine tuning, while BO is more robust.
In general, the CP code offers some sophisticated capabilities that are not present in the PW code, and vice versa.
One example is that CP allows to control the temperature with the Nose thermostat and it also generate more output (like the file cp.evp and others). Is is possible that a future version of PW includes the Nose thermostat, although the advantage of this can be debated.

Exercise 3. Classical molecular dynamics with Lennard-Jones force field. Calculation of averages and errors.

Here will perform basic simulations of a Lennard-Jones liquid using examples codes of the book *Understanding molecular Simulation*, by D. Frenkel and Berend Smit.

The Case study 4 is the static properties of a Lennard-Jonnes fluid. The example runs a small code of molecular dynamics only for a gas of 108 particles interacting through Lennard-Jonnes potential, in a box with periodic boundary conditions.

One sets the density and the temperature and runs a MD, from which one extracts the temperature, pressure, and the internal energy. The kinetic and total energies are easily calculated from the previous data. This is the calculation of a point of the equation of state.

The goal of this exercise is to understand how to calculate averages and errors from a MD simulation.

Step1. Change to directory AIMD/ex3-lj
From here change to directory CaseStudy_4/Block and run

```
./block
```

If you see something like this

```
 ***** Calculate block averages ************
read start: end of file
apparent state: unit 31 named fort.31
```

it means that the code runs. If not, then do

```
make block
```

Then change to `../Source`
look for a fila named MD and run it

```
./MD
```

if you see

```
 **************** MC_NPT ***************
read start: end of file
apparent state: unit 15 named fort.15
```

it is OK. Otherwise, do

```
make MD
```

Then, change to the directory ../Run

The script run automates the MD calculations. It drives the following steps

1) Creates a file named fort.15 with the parameters of the simulation. See details in the file CaseStudy_4/Run/README.MD
2) runs the program MD (../Source/MD >> out)
3) Performs a block analysis of the error of the averages (../Block/block   >> out)

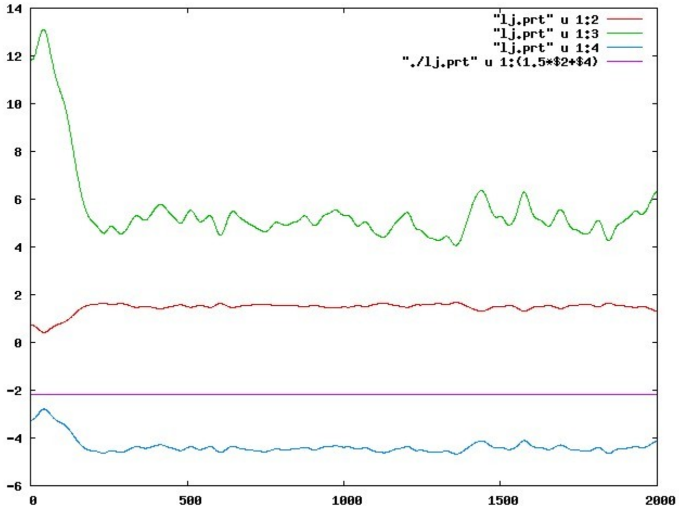
The main output files are
lj.res : last configuration to allow a restart
lj.prt : record of k*T, pressure and potential energy per particle.
out   : main output. Includes the averages of thermodynamical magnitudes and their statistical error.

Edit the script ./run. Modify the parameter tmax (simulation time) from 2.0 to 600.0 (or verify it is already set to 600.0). Then, run the script

```
./run
```

after some minutes the simulation ends and we can analyze the results.

First, let us plot the kinetic and potential energies, that are in columns 2 (multiply by 3/2) and 4 of the file lj.prt. The are gnuplot files are available (list them with ls *.gp).



Above can be seen the first 2000 steps and see the equilibration of the magnitudes. The configurations during the equilibration must not be used for the calculation of the averages. Let us discard the first steps, just erasing the first 1000 lines of lj.prt.
Check that the total energy is constant (the sum of column 2 (mult by 1.5) and 4).

Now let us obtain the average of the magnitudes and their statistical errors

If we have a serie of numbers A_n,

$$\langle A\rangle=\frac{1}{N_{steps}}\sum_{n=1}^{Nsteps} A_n$$

$$\sigma_A^2=\langle(A-\langle A\rangle)^2\rangle=\frac{1}{N_{steps}}\sum_{n=1}^{N_{steps}}(A_n-\langle A\rangle)^2 \quad \text{(Variance)}$$

$$E_A=\frac{\sigma_A}{\sqrt{N_{ind}}} \quad \text{(Error of the average)}$$

$$N_{ind}=\frac{N_{steps}}{N_{corr}}=\frac{t_{sampling}}{t_{corr}}$$ N_{ind} is the number of independent data, t_{corr} is the correlation time

The problem is that the correlation time must be determined somehow. The file statistical_errors.pdf that explaines the concepts and the method of “block analysis”.

After deleting the first 2000 lines from lj.prt, let us run the block analysis independently

```
cp lj.prth fort.31
cp lj.prt fort.32
../Source/block >> out2
```

For subsequent simulations of the same system, the discard of the equilibration steps can be done automatically setting the parameter tequil = 1.0 (1000 steps of 0.001) in the input file.

Let us

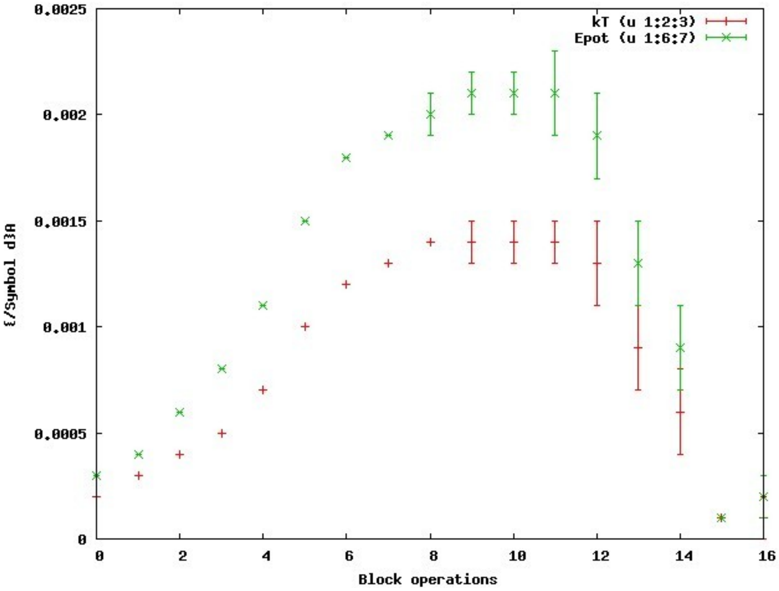
```
cp out2 blocks2
```

and delete lines from the beginning to the line 'over number of blocks 150000'

The block analysis can be displayed with the following gnuplot command:

```
plot "./blocks2" u 1:2:3 w yerrorbars,"./blocks2" u 1:6:7 w yerrorbars
```

the columns 2 and 6 are the standard deviations of the blocked data of kT and potential energies. The following plot is what must be obtained



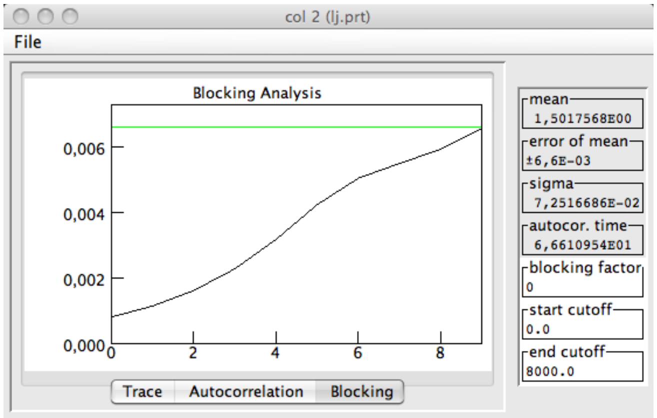
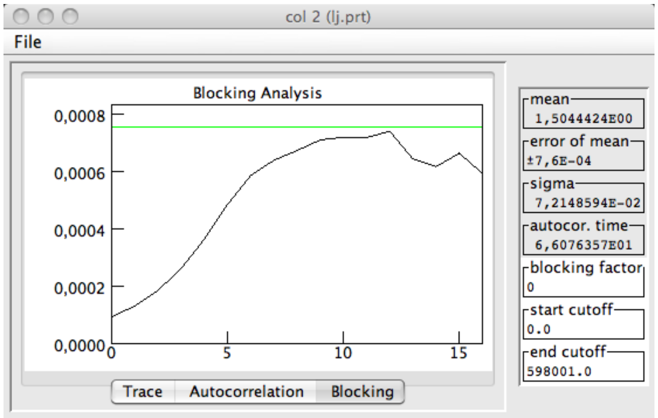
The values at the plateau represent the errors of the mean of the correlated data. If the data had been uncorrelated the error of the means would not depend on the block size. Columns 3 and 7 are the error bars of the errors.

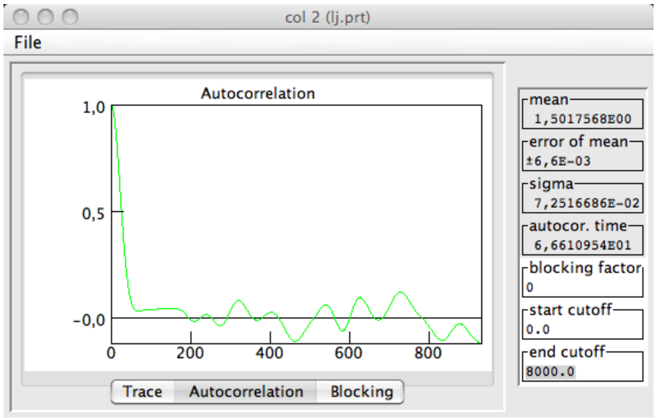
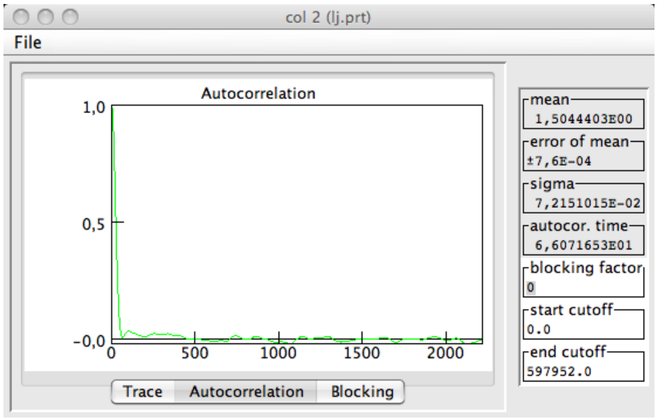
The same analysis can be done didacticaly using the java program Datasporkk. Execute the command

```
java -jar ../../../../Dataspork/dataspork.jar
```

Open the file lj.prt and explore the different options. In particular click the Tab Blocking. Note that you can select the range of values that account for the averages. You can see the effect of including or discarding the data of the equilibration part, the histogram.

Compare the block diagram using 148000 steps and using 8000 steps. What can you conclude?

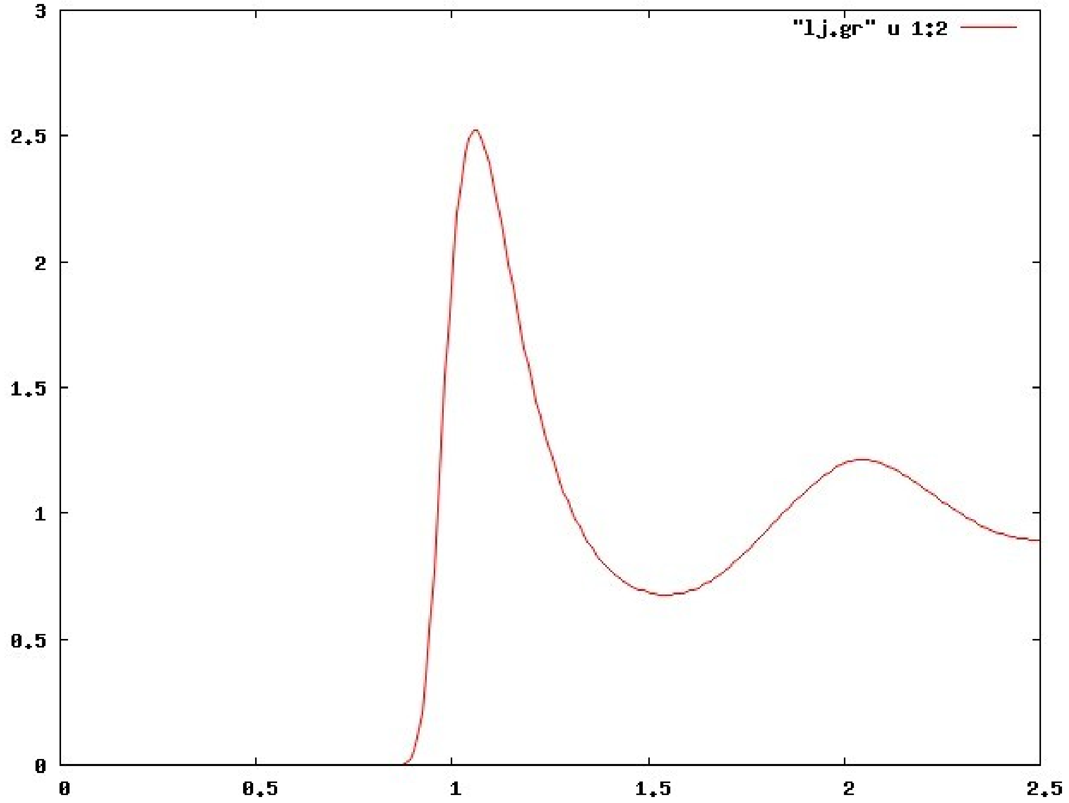
With 8000 steps	With 598000 steps
 col 2 (lj.prt) File Blocking Analysis 0,006 0,004 0,002 0,000 0 2 4 6 8 Trace Autocorrelation Blocking mean 1,5017568E00 error of mean ±6,6E-03 sigma 7,2516686E-02 autocor. time 6,6610954E01 blocking factor 0 start cutoff 0.0 end cutoff 8000.0	 col 2 (lj.prt) File Blocking Analysis 0,0008 0,0006 0,0004 0,0002 0,0000 0 5 10 15 Trace Autocorrelation Blocking mean 1,5044424E00 error of mean ±7,6E-04 sigma 7,2148594E-02 autocor. time 6,6076357E01 blocking factor 0 start cutoff 0.0 end cutoff 598001.0

Autocorrelation function of temperature using 8000 steps.	Autocorrelation function of temperature using 598000 steps.
 col 2 (lj.prt) File Autocorrelation 1,0 0,5 -0,0 0 200 400 600 800 Trace Autocorrelation Blocking mean 1,5017568E00 error of mean ±6,6E-03 sigma 7,2516686E-02 autocor. time 6,6610954E01 blocking factor 0 start cutoff 0.0 end cutoff 8000.0	 col 2 (lj.prt) File Autocorrelation 1,0 0,5 -0,0 0 500 1000 1500 2000 Trace Autocorrelation Blocking mean 1,5044403E00 error of mean ±7,6E-04 sigma 7,2151015E-02 autocor. time 6,6071653E01 blocking factor 0 start cutoff 0.0 end cutoff 597952.0

Pay attention to the correlation time and to the Self-correlation function shown above.

If you repeat the simulation, you can choose to increase the sampling time, parameter igr of the input file.
Multiply the present value by the correlation time and repeat the simulation.
If nsamp=1 and the atocorrelation time is 66 steps, we can set nsamp=60 and expect to obtain a new data with autocorrelation time = 1 or 2 step. We can also set tequil=2.0 to not sample the data during equilibration. Open the file lj.prt with dataspork and examine the reported correlation time.

The pair distribution function is one of the most common output of molecular dynamics of liquids. The file lj.gr contains it. It is proportional to the probability of findings two atoms as a function of distantce. The peaks indicate bond lengths, and the valleys after them, indicate the radii of coordination spheres.


"lj.gr" u 1:2
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5

Optional exercises.

Find the heat capacity at different temperature and do a plot of C_V vs temperature. What happens with low temperature results ?