Simulation Methods in Physics I

Worksheet 3: Molecular Dynamics 2 and Observables

Students:	Michael Marquardt	Cameron Stewart
matriculation numbers:	3122118	3216338

1 Command line parameters

For the simulation different command line parameters were defined. In the following code blocks you can see them. They will be explained during the report.

Code block 1: Commands for ljsim.py — script: src/ljsim.py

```
9  # command line arguments
10 parser = argparse.ArgumentParser()
11 parser.add_argument("--cont", type=double, help="continue
      calculation with for cont further time")
12 parser.add_argument("--time", type=double, help="How long
      do you want to run the simulation? | default time=10s")
13 parser.add_argument("--tstat", type=double, help="Uses a
      thermostat with a given temperature")
14 parser.add_argument("--warm", type=double, help="Use the
      warming up | pass force")
15 parser.add_argument("--ctstat", type=double, help="
      continue for the simulation with tstat")
16 parser.add_argument("--cwarm", type=double, help="continue
       for the simulation with warm")
17
18 args = parser.parse_args()
```

Code block 2: Commands for ljanalyze.py — script: src/ljanalyze.py

```
8  # command line arguments
9  parser = argparse.ArgumentParser()
10 parser.add_argument("--M", type=int, default=10, help="
      window size | default: M=10")
11 parser.add_argument("--datafile", type=str, default='../
      dat/ljsim.dat', help="datafilename | default: '../dat/
      ljsim.dat'")
12 parser.add_argument("--teq", type=double, help="
      equilibration time [s] | calculates averages after
      equilibration time")
13 parser.add_argument("--tlim", type=double, nargs='+', help
      ="type in the time limits you want to plot")
14
15 args = parser.parse_args()
```

2 Restart the program where left it

First we want to change the programm in a way, that it is able to restart the simulation where it ended last time. The first thing that we have to do is to store the necessary information about the end state of the system, the position x and velocity v of each particle, into the file ljsim.dat. This happens in code block 3. Furthermore there are the new variables Ts and Ps. The represent the temperature and pressure of the system over time and will be explained in the next chapter.

Code block 3: Storing data — script: src/ljsim.py

```
208 # write out simulation data
209 print("Writing simulation data to {}.".format(datafilename
    ))
210 datafile = open(datafilename, 'w')
211 pickle.dump([ts, Es, Ts, Ps, x, v], datafile)
212 datafile.close()
```

Now this data must be read by ljsim.py. Therefore we define a command line parameter −−cont which accepts as argument the time for which the simulation should be continued (code block 1). If the parameter is not used the simulation will start new. Furtehrmore there is another argument −−time which takes the simulationtime for a new simulation. The code in code block 4 does exactly this.

Code block 4: Continue simulation — script: src/ljsim.py

```
65 # Import previous data
66 if args.cont:
67         # open datafile
68         datafile = open(datafilename,'r')
69         ts, Es, Ts, Ps, x, v = pickle.load(datafile)
70         datafile.close()
71
72         # length of run
73         t = ts[-1]
74         tmax = t+args.cont
75         step = 0
76
77 else:
78         # length of run
79         if args.time:
80                 tmax = args.time
81         else:
82                 tmax = 10.0
```

3 Calculating Temperature and Pressure

Deviation of the pressure

The virial of a system is defined as:

$$G = \sum_i^N \boldsymbol{p}_i \cdot \boldsymbol{r}_i \tag{1}$$

In the case

$$0 = \left\langle \frac{\mathrm{d}G}{\mathrm{d}t} \right\rangle = \left\langle \sum_i^N \frac{\boldsymbol{p}_i^2}{m_i} \right\rangle + \sum_i^N \langle \boldsymbol{F}_i \cdot \boldsymbol{r}_i \rangle \tag{2}$$

the following equation can be derived:

$$-\sum_i^N \langle \boldsymbol{F}_i \cdot \boldsymbol{r}_i \rangle = 2 \left\langle \sum_i^N \frac{\boldsymbol{p}_i^2}{2m_i} \right\rangle = 2 \langle E_{\mathrm{kin}} \rangle = 3Nk_BT \overset{\text{id. gas}}{=} 3PV \tag{3}$$

For an pair interaction (like in the simulation) let the forces $\boldsymbol{f}_{ij}$ and the vectors $\boldsymbol{r}_{ij}$ be defined as follows:

$$\boldsymbol{r}_{ij} = \boldsymbol{r}_j - \boldsymbol{r}_i \tag{4}$$

$$\boldsymbol{f}_{ij} = f_{\mathrm{lj}}(\boldsymbol{r}_{ij}^2) \cdot \frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|} \tag{5}$$

With this knowldege we can derive the pressure of the system:

$$P = P_{\text{id. gas}} + P_{\text{interaction}} \tag{6}$$

$$= \frac{Nk_BT}{V} + \frac{1}{3V} \sum_i^N \langle \boldsymbol{F}_i \cdot \boldsymbol{r}_i \rangle \tag{7}$$

$$= \frac{1}{3V} \left[\left\langle \sum_i^N \frac{\boldsymbol{p}_i^2}{2m_i} \right\rangle + \left\langle \sum_{i,j \neq i}^N -\boldsymbol{f}_{ij} \cdot \boldsymbol{r}_i \right\rangle \right] \tag{8}$$

$$= \frac{1}{3V} \left[\left\langle \sum_i^N \frac{\boldsymbol{p}_i^2}{2m_i} \right\rangle + \left\langle \sum_{i,j>i}^N \boldsymbol{f}_{ij} \cdot \boldsymbol{r}_j - \boldsymbol{f}_{ij} \cdot \boldsymbol{r}_i \right\rangle \right] \tag{9}$$

$$= \frac{1}{3V} \left[\left\langle \sum_i^N \frac{\boldsymbol{p}_i^2}{2m_i} \right\rangle + \left\langle \sum_{i,j>i}^N \boldsymbol{f}_{ij} \cdot \boldsymbol{r}_{ij} \right\rangle \right] \tag{10}$$

$$= \frac{1}{3V} \left[2 \langle E_{\mathrm{kin}} \rangle + \left\langle \sum_{i,j>i}^N \boldsymbol{f}_{ij} \cdot \boldsymbol{r}_{ij} \right\rangle \right] \tag{11}$$

4 Molecular Dynamics at a Desired Temperature

For the *velocity rescaling* thermostat we can derive the rescaling-factor f_{re} from equation (12).

$$\frac{3}{2}k_B T_0 = \frac{E_{\mathrm{kin},0}}{N} \tag{12}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\frac{(f_{\mathrm{re}}\cdot \boldsymbol{v}^{(i)})^2}{2m} \tag{13}$$

$$= f_{\mathrm{re}}^2 \frac{E_{\mathrm{kin}}}{N} \tag{14}$$

$$= f_{\mathrm{re}}^2 \frac{3}{2}k_B T \tag{15}$$

$$f_{\mathrm{re}} = \sqrt{\frac{T_0}{T}} \tag{16}$$