

1 Exercise: Simple coarse grained model

2 Exercise: Nosè-Hoover

```
1 #include <iostream>
2 #include <fstream>
3 #include <cmath>
4
5 using namespace std;
6
7 //v is fixed, calculation of v(t+1/2dt) not necessary
8
9 double v(double t, double dt, double fixed){
10     if (t <= 0){
11         return 1.0;
12     }
13     else {
14         // fix velocity to 0.8 or 1.2
15         return fixed; //1/2*(v(t-1/2*dt,dt)+v(t+1/2*dt,dt));
16     }
17 }
18
19 // x(t) for a harmonic oscillator with m=k=1
20 double x(double t){
21     double x0 = 1.0;
22
23     if(t == 0){
24         return x0;
25     }
26     else {
27         return (x0*cos(t));
28     }
29 }
30
31 double x(double t, double dt, double fixedv){
32     return (x(t) + dt*v(t,dt,fixedv));
33 }
34
35 int main(int argc, char* argv[]) {
36
37     ofstream outfile1;
38     ofstream outfile2;
39     outfile1.open("NoseHooverOut08.txt");
40     outfile2.open("NoseHooverOut12.txt");
```

```

41
42 outfile1 << "t x(t) v(t)" << endl;
43 outfile2 << "t x(t) v(t)" << endl;
44 double dt = 0.1;
45
46 for (double t = 0; t < stod(argv[1]); t+=dt){
47     outfile1 << t << " " << x(t, dt, 0.8) << " " << v(t, dt,
48     0.8) << endl;
49     outfile2 << t << " " << x(t, dt, 1.2) << " " << v(t, dt,
50     1.2) << endl;
51 }
52
53 outfile1.close();
54 outfile2.close();
55
56 // plot x(t) and v(t)
57 system("gnuplot plot.gnu ");
58
59 return 0;
60 }

```

```

1 # gnuplot script to plot Nose-Hoover output
2 set autoscale # scale axes automatically
3 unset log # remove any log-scaling
4 unset label # remove any previous
   labels
5 set xtic auto # set xtics automatically
6 set ytic auto # set ytics automatically
7 set title "Simulation of a harmonic oscillator with fixed velocity"
   "
8 set xlabel "t"
9 set ylabel "x(t), v(t)"
10 set xr [0.0:100]
11 set yr [-2:2]
12
13
14 set size 2,2
15 set origin 0,0
16 set multiplot layout 2,1 columnsfirst scale 1,1
17
18 plot "NoseHooverOut08.txt" using 1:3 title 'v(t) fixed to 0.8'
   with linespoints, "NoseHooverOut08.txt" using 1:2 title 'x(t)'
   with linespoints
19
20 plot "NoseHooverOut12.txt" using 1:3 title 'v(t) fixed to 1.2'
   with linespoints, "NoseHooverOut12.txt" using 1:2 title 'x(t)'
   with linespoints
21
22 unset multiplot

```

3 Exercise: Molecular Dynamics - BALL

```
1 #include <BALL/KERNEL/system.h>
2 #include <BALL/KERNEL/selector.h>
3 #include <BALL/FORMAT/PDBFile.h>
4 #include <BALL/MOLMEC/MDSIMULATION/microCanonicalMD.h>
5 #include <BALL/STRUCTURE/fragmentDB.h>
6 #include <BALL/STRUCTURE/residueChecker.h>
7 #include <BALL/MOLMEC/AMBER/amber.h>
8 #include <BALL/MOLMEC/MINIMIZATION/conjugateGradient.h>
9
10 using namespace std;
11 using namespace BALL;
12
13 int main(int argc, char* argv[]){
14
15     PDBFile sourceFile;
16     System mdSystem;
17
18     //Sanity checks for command-line arguments
19     if(argc == 3){
20         sourceFile.open(argv[1], ios::in);
21     }else{
22         cout << "Wrong amount of Parameters\n\n Usage: prog
23         sourceFile simulationTime\n";
24         return 1;
25     }
26
27     if(sourceFile.is_open()){
28         sourceFile.read(mdSystem);
29         sourceFile.close();
30     }
31
32     FragmentDB db("");
33     mdSystem.apply(db.normalize_names);
34     mdSystem.apply(db.add_hydrogens);
35     mdSystem.apply(db.build_bonds);
36
37     ResidueChecker rc(db);
38     mdSystem.apply(rc);
39
40     // create hydrogen bonds and force field
41     AmberFF amber(mdSystem);
42     Selector hydrogen_selector("element(H)");
43     mdSystem.apply(hydrogen_selector);
44
45     amber.options[PeriodicBoundary::Option::PERIODIC_BOX_ENABLED]="
46     true";
```

```

45  amber . setup (mdSystem) ;
46
47  MicroCanonicalMD md(amber) ;
48  md . setReferenceTemperature (300.0) ;
49  md . setEnergyOutputFrequency (500.0) ;
50
51  // redirect std::cout to file . Found no other possibility to
52  // write MD simulation output directly to file
53  ofstream finalMD ("finalMD.txt") ;
54  streambuf *coutbuf = cout . rdbuf () ;
55  cout . rdbuf (finalMD . rdbuf ()) ;
56  md . simulateTime ( stod (argv [2]) ) ;
57  cout . rdbuf (coutbuf) ;
58  finalMD . close () ;
59  return 0 ;

```