Heinrich–Heine–Universität Düsseldorf
Institut für Theoretische Physik II
Computational Physics
Wintersemester 2018/2019

Prof. Dr. J. Horbach
Dr. S. Ganguly
(saswati@thphy.uni-duesseldorf.de.de)
M. Golkia (mehrdad.golkia@hhu.de)
Blatt 4 vom 6.11.2018
Abgabe bis 10:30 Uhr am 13.11.2018

"Crystals are living beings at the beginning of creation."

Nikola Tesla

Problem 4.1: **1D Harmonic Crystal**

Consider a one-dimensional chain of N particles of mass m interacting via springs (spring constant k) with each of its nearest neighbours. In this system, periodic boundary conditions are present, i.e. the first particle interacts with the last and vice versa (see Figure 1).

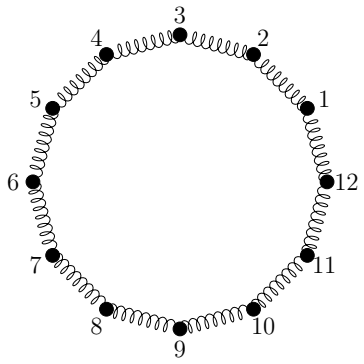

Figure 1: The first particle (1) interacts with the last particle (12). Note that the coordinates are actually one-dimensional.

a) Implement the harmonic chain to solve Newton's equations of motion. To this end, use the velocity-Verlet algorithm with a discrete time step $\delta t = 0.1$ and $N = 100$ particles. The particles have a mass $m = 1$ and the spring constant $k = 1$.
Make sure you take advantage of Newton's third law, i.e. calculate each interaction only once each time step.

b) Particles are placed initially at positions $x_i^0 = i - 1,\ (i = 1, \ldots, N)$ (therefore, the total system length is $L = N$). Give each particle a random initial velocity. Do not forget to subtract the velocity of center of mass from it.

c) Measure, plot and discuss

   i) Kinetic and potential energy as well as the total momentum of the system as a function of time. What do you observe for the total momentum?

   ii) Total energy as a function of time for different $\delta t = \{0.05, 0.1, 0.2\}$. The total energy of the system is supposed to be constant. Is the obtained total energy constant? Explain the behaviour of total the energy as a function of δt.

Hint: You may use the following steps to implement your code.

1) Use arrays of size N (number of particles) to store the current velocity, position and force of each particle.

2) It is better to use separate, independent functions for each task:

   i) `initialize()`: Sets up the arrays and initialize the phase space of the system.
      - To generate a random velocity for each particle you can use

        ```
        v[i] = ((double)(rand()%RAND_MAX )) / RAND_MAX - 0.5
        ```

        It gives a random value between -0.5 and 0.5 to the particle i. Do not forget to use #include <stdlib.h>.

   ii) `md_step()`: Propagate the system by one single time step (md = molecular dynamics) and update the force.

   iii) `calculate_force()`: Updates the force arrays depending on the current positions of the particles (take care of boundary conditions!).

   iv) `potential()`: Calculates the potential and the force of two particles, only using their distance from each other.

   v) `write_phase_space()`: Depending on the current time step, write a file that contains position and velocity of the particles. You can create a file including the step as a file name as follows in C++ (this requires the `-std=c++11 or later versions` switch for `g++`):

   ```
   #include <string>
   // this would generate "phase_space_1" or "phase_space_2", ...
   string fname = "phase_space_" + to_string(step);
   ofstream output(fname.c_str());
   ```

   vi) `update_thermodynamics()`: To calculate thermodynamic variables like kinetic energy, centre of mass, etc.

   vii) Then the `main()` can do very simple tasks (pseudo-code):

   ```
   int main()
   {
     // first read in dt, N
     initialize(..);
     // update the force at the beginning
     calculate_force(..);

     for(i = 0; i < steps; ++i) {
       md_step(..)
       if(..) {
         update_thermodynamics(...);
         write_phase_space(..)
       }
     }
   }
   ```

3) Please try to test the functions independently to be sure that all of them work perfectly.

4) Make sure that the expected physical properties of the system are maintained (e.g. energy conservation, momentum conservation, and force balance).

5) To avoid the common mistakes, be sure to check the sign of the force, implement the periodic boundary condition and be careful about initial position and velocity of particles.