

## PSE Molekulardynamik

### Sheet 3: XML, Linked-Cell-Algorithm and “falling drops”

Exercise at 07th November 2014

#### Task 1 “XML-Input”

As the configuration of a simulation is getting more complex, we will use xml input files.

- Download the tool Codesynthesis XSD ( <http://www.codesynthesis.com/products/xsd/> ).
- Make yourself familiar with the tool!  
(Have a look at the example “Hello” as it is contained in the examples-folder, and as described in the Getting Started Guide.)
- Develop an input file reader using the the C++/Tree-Mapping. It should be possible to specify at least the following things:
  - Base name of the output files
  - Write frequency of the output files
  - $\delta t$ ,  $t_{end}$ , etc.
  - Input files and specification of the cuboids
- Encapsulate the use of the XML-component in a suitable way! Also take care of a testable design.

#### Note:

- You can find more information about XML and XML Schema at
  - XML: <http://www.xml.com/pub/a/98/10/guide0.html>
  - XSD: [http://www.w3schools.com/Schema/schema\\_intro.asp](http://www.w3schools.com/Schema/schema_intro.asp)

- Note that the XML DOM object may be constructed by a method taking `std::istream` as a parameter.
- Note that the parser tries to validate against an .xsd-file by default. Thus take care to specify the .xsd document correctly, or switch validation off by passing the flag `xml_schema::flags::dont_validate` to the parser.

#### Task 2 “Linked-Cell Algorithm”

As you experienced on the last sheet, the computational effort for a naive molecular-dynamics simulation is pretty high ( $\mathcal{O}(n^2)$ ). Thus we simplify the model and make use of the linked-cell algorithm, as discussed in the lecture.

- Implement a new particle container, realizing the linked-cell algorithm. However, it should still be possible to use the existing implementation.
- Adapt the XML format. Especially, it is now necessary to specify the size of the domain (that is the size of the inner area including the boundary of the domain, but not the halo particles), and the cutoff-radius.
- The container has to offer access to the halo particles as well as to the boundary particles. It has to support the deletion of the particles in the halo.
- Perform the simulation “Collision of two bodies“ once again, now with the following parameters:

$$\begin{aligned}x_1 &= \{20, 20, 0\} & x_2 &= \{70, 60, 0\} \\v_1 &= \{0, 0, 0\} & v_2 &= \{0, -10, 0\} \\N_1 &= \{100, 20, 1\} & N_2 &= \{20, 20, 1\} \\ \epsilon &= 5 & h &= 2^{1/6} \cdot \sigma \approx 1.1225 \\ m &= 1 & \sigma &= 1 \\ \delta t &= 0.0005 & t_{end} &= 20 \\ \text{Size of the domain: } L &= \{180, 90\} & r_{cutoff} &= 3.0\end{aligned}$$

- How does the runtime per iteration behave for the two different implementations? Compare the times for a simulation of a 2d-square with e.g. 1000, 2000, 4000 and 8000 molecules

Embed a graph comparing the two implementations in the doxygen documentation.

### Task 3 "Boundary Conditions"

Up to now we neglected boundary conditions. Implement different boundary conditions for the Linked-Cell particle container:

- Outflow: particles are being removed when they cross the boundary of the domain
- Reflecting Boundary Conditions. Realize the reflection in the following way: Place a "counter"-particle at the boundary. The force is calculated according to the Lennard-Jones-Potential, but it is only if the particles is closer to the boundary than  $2^{\frac{1}{6}} \cdot \sigma$ , thus only the repulsing component is effective.  
**Note:** For this implementation of reflecting boundaries the time step  $\delta t$  has to be small enough in order to guarantee the stability of the method.
- Adapt the xml input file correspondingly so that it is possible to specify at which boundary of the domain which type of condition is applied.

### Task 4 "Simulation of a falling drop"

One of the tasks of the next tutorial will be the simulation of a drop of liquid falling into a basin filled with the same fluid. In preparation of that, implement a generator for spheres.

- Extend the particle generator so that you can generate spheres of molecules. The molecules should be arranged in a grid, the sphere is described by the following properties:
  - Coordinates of the center
  - Initial velocity of the sphere
  - The radius as the number of molecules along the radius
  - distance (= meshwidth)  $h$  of the molecules.
- Adapt the xml input format correspondingly.
- Conduct a first little experiment to test your implementation and let a sphere fly against a reflecting boundary with the following parameters:

$$x = \{60, 25, 0\}$$

$$N = 15$$

$$\epsilon = 5$$

$$\delta t = 0.00005$$

$$r_{cutoff} = 3.0$$

$$\text{Size of the domain: } L = \{120, 50\}$$

$$v = \{0, -10, 0\}$$

$$h = 2^{1/6} \cdot \sigma \approx 1.1225$$

$$\sigma = 1$$

$$t_{end} = 10$$

Have fun!

Deliver by 27th November 2014, 3 PM per mail to [uekerman@in.tum.de](mailto:uekerman@in.tum.de)!