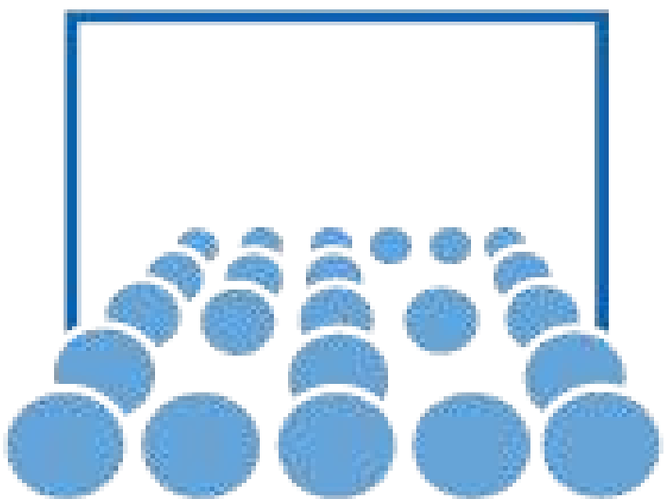
PSE Molekulardynamik

Thermostats, Lorentz-Berthelot mixing rule, Linux Cluster

Wolfgang Eckhardt

06.12.2013



Outline

- Schedule
- Presentations: Worksheet 3
- Thermostats
- Lorentz-Berthelot mixing rule
- Linux Cluster
- Preparation: Worksheet 4

Schedule (big meetings)

Date	Worksheet
18.10.2013	
31.10.2013	1
15.11.2013	2
06.12.2013	3
20.12.2013	4
17.01.2014	5 (preliminary results)
31.01.2014	5

Presentations: Worksheet 3

Institut für Informatik — TU München
Scientific Computing in Computer Science
Prof. Dr. H.-J. Bungartz
Dipl.-Inf. W. Eckhardt
Dipl.-Math. A. Breuer

WS 2012/13

PSE Molekulardynamik
Sheet 3: XML, Linked-Cell-Algorithm and "falling drops"

Exercise at 16th November 2012

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`
- 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 naiv molecular-dynamics simulation is pretty high ($\mathcal{O}(n^2)$). Thus we simplify the model and make use of the linked-cell algorithmus, as discussed in the lecture.

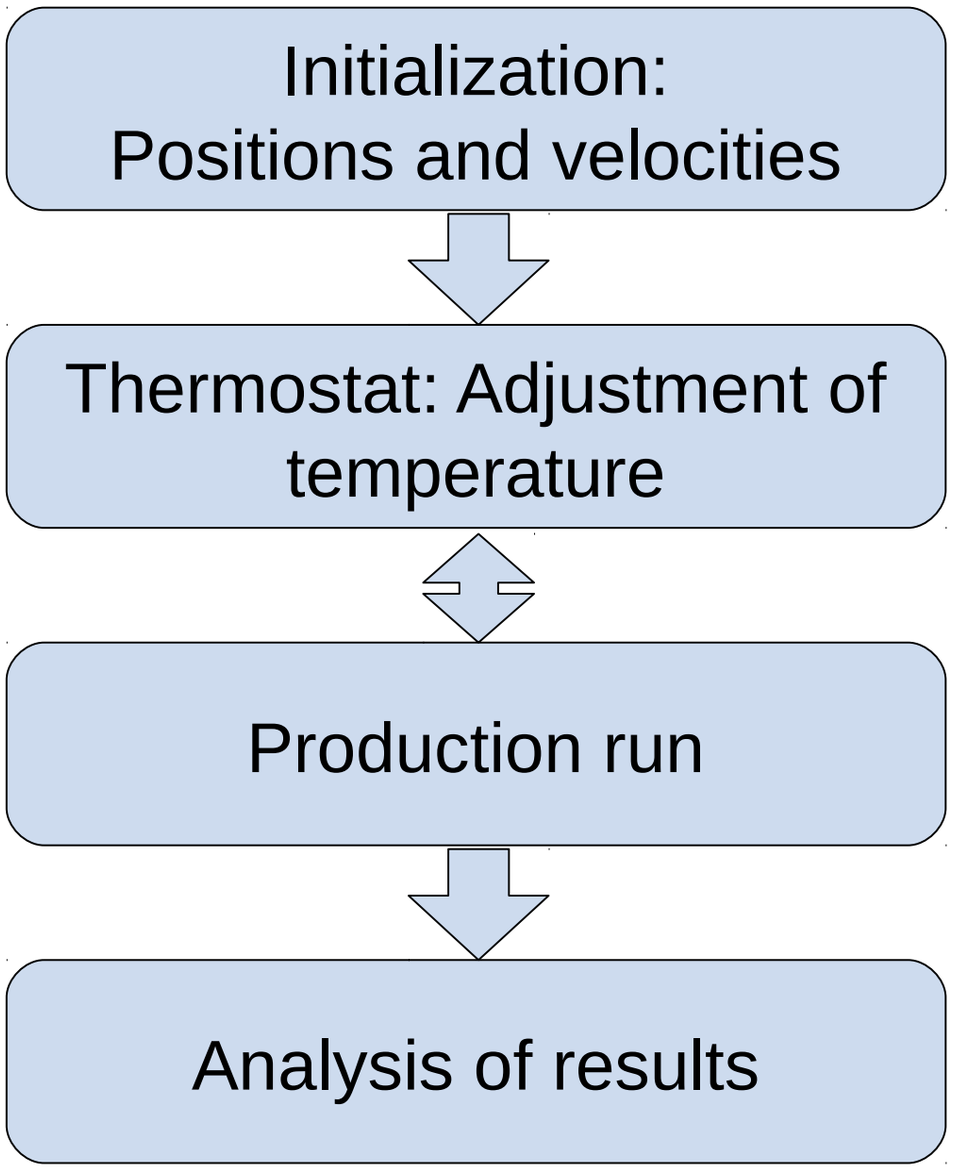
- Implement a new particle container, realizing the linked-cell algorithm. However, it should still be possible to use the existing implementation.
- Adapt the XML format. Especially, it is now neccessary to specify the size of the domain (that is the size of the inner area including the boundary of 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:

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

- 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

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

Thermostats: Simulation overview



Thermostats

- Kinetic energy

$$E_{kin} = \sum_{i=1}^{\#particles} \frac{m_i \langle v_i, v_i \rangle}{2}$$

- Equipartition theorem (thermodynamics): Correlation temperature ↔ kinetic energy

$$E_{kin} = \frac{\#particles \cdot \#dimensions}{2} k_B T$$

- Scale to achieve temperature

$$\beta = \sqrt{\frac{E_{kin}^D}{E_{kin}}}, \quad v_i^n := \beta v_i^n$$

Thermostats

- Heating/Cooling smooth process → Smooth energy transfer
- “Adjustment of temperature” over multiple time steps

- Example (worksheet 4):
  - Target temperature: 40
  - Application of thermostat every 1000 time steps

Lorentz-Berthelot mixing rule

- Different materials → Lennard-Jones potential?
- Generalized formulation

$$F_i = \sum_{\substack{j=1 \\ j \neq i}}^{\#particles} F_{ij}, \quad F_{ij} = \frac{24\epsilon_{ij}}{(\|x_i - x_j\|_2)^2} \left(\left(\frac{\sigma_{ij}}{\|x_i - x_j\|_2} \right)^6 - 2 \left(\frac{\sigma_{ij}}{\|x_i - x_j\|_2} \right)^{12} \right) (x_j - x_i)$$

- Mixing rule

$$\sigma_1 = \sigma_{11}$$
$$\epsilon_1 = \epsilon_{11}$$
$$\sigma_2 = \sigma_{22}$$
$$\epsilon_2 = \epsilon_{22}$$

$$\Rightarrow \quad \sigma_{12} = \sigma_{21} = \frac{\sigma_{11} + \sigma_{22}}{2}$$
$$\epsilon_{12} = \epsilon_{21} = \sqrt{\epsilon_{11}\epsilon_{22}}$$

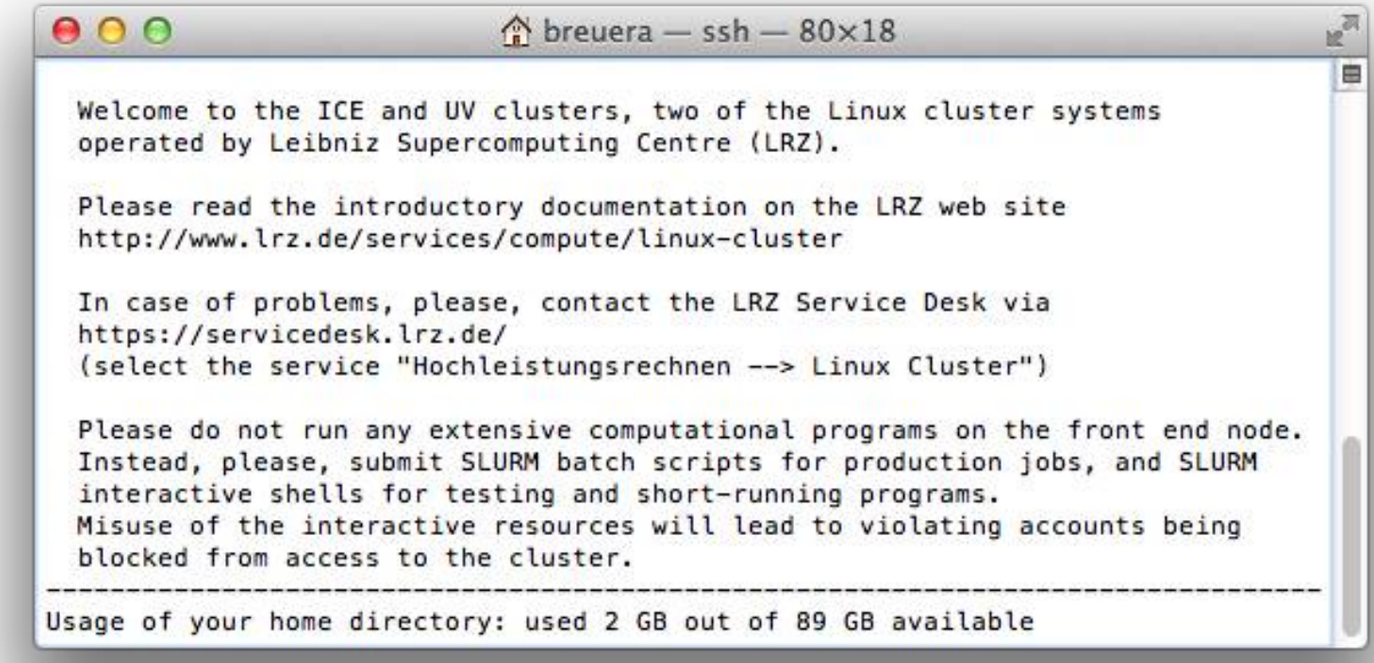
Linux Cluster / MAC Cluster: Overview

- http://www.lrz.de/services/compute/linux-cluster/
- http://www.mac.tum.de/wiki/index.php/MAC_Cluster
- Read usage instructions carefully!
- Here: ICE (optional) and MAC-Cluster AMD-BDZ / Intel SNB Partition

	SGI ICE	MAC-Cluster SNB Partition	MAC-Cluster BDZ Partition
Architecture	Dual socket quad-core Intel Nehalem	Intel SandyBridge-EP Xeon E5-2670 (2.6 GHz)	AMD Opteron 6274 (2.2 GHz)
#Nodes	64	28	19
#Processor cores	512	448	1216
Aggregate peak performance (Tflop/s)	5.2	9.3	10.7
Aggregate memory (TByte)	1.5	3.5	4.9

Linux Cluster: Basics

- Login
- Checkout & build (libraries see course page)
- Compute: Interactive & batch
- Profiling with VTune: Interactive & batch



Linux Cluster: Basics

- Login Linux-Cluster:

```
ssh USER@lxlogin1.lrz.de
```

- Login MAC-Cluster:

```
ssh USER@mac-login-amd.tum-mac.cos.lrz.de
```

- Module environment:

```
module list | avail | load | unload | show ...
```

- Compute: Interactive

```
salloc --partition=snb –ntasks=1 --cpus-per-task=32
       -–time=01:00:00

srun ./MolSim ARGS...
```

- Compute: batch (see example-script at login)

```
squeue | sbatch | sinfo
```

Preparation: Worksheet 4

- Reminder: Use external literature.
  - M. Griebel et al: Numerical Simulation in Molecular Dynamics: Numerics, Algorithms, Parallelization, Applications
  - M. Griebel et al: Numerische Simulation in der Moleküldynamik: Numerik, Algorithmen, Parallelisierung, Anwendungen
  - http://software.intel.com/sites/products/documentation/doclib/stdxe/2013/composerxe/compiler/cpp-mac/index.htm
  - ...?

Preparation: Worksheet 4

Institut für Informatik — TU München WS 2012/13
Scientific **C**omputing in **C**omputer **S**cience
Prof. Dr. H.-J. Bungartz
Dipl.-Inf. W. Eckhardt
Dipl.-Math. A. Breuer

PSE Molekulardynamik
Sheet 4: Thermostats, Rayleigh-Taylor-Instability and "falling drops"

Exercise at 07th December 2012

Task 1 "Thermostats"

In order to keep the simulated material at a constant temperature, as well as to be able to simulate heating or cooling processes, it is neccessary to have a thermostat.

The temperature of the phase is determined by the velocity of the particles. The dependency is given by

$$E_{\text{kin}} = \frac{\#\text{dimensions} \cdot \#\text{particles}}{2} k_B T$$

or dimensionless:

$$E_{\text{kin}} = \frac{\#\text{dimensions} \cdot \#\text{particles}}{2} T$$

where E_{kin} can be calculated as the sum of the kinetic energy of all particles:

$$E_{kin} = \sum_{i=1}^{\#\text{particles}} \frac{m_i \langle v_i, v_i \rangle}{2}$$

As seen in the tutorial, the scaling factor can be calculated as:

$$\beta = \sqrt{\frac{E_{\text{kin}}^D}{E_{\text{kin}}}}$$

- Implement such a thermostat.
- The input parameters are
  - initial temperature:
    the initial scaling value for the velocity for the Maxwell-Boltzmann distribution is calculated as:

$$v = \sqrt{\frac{2E_{\text{kin}}}{\#\text{dimensions} \cdot \#\text{particles} \cdot m}}$$

    The initialization of the phase to a given temperature may be moved to the thermostat and be called after all particles have been instantiated.
    For our applications (at least for the moment) it is ok to set $m = 1$ and hardcode it.
    However, the initialization with the Brownian Motion should be optional.
  - the number of timesteps after which the thermostat is applied
- Furthermore for simulations with a different target temperature:
  - target temperature
  - temperature difference, i.e. the step size in which the temperature should be changed
  - the number of timesteps after which the temperature has to be changed
- Adapt the XML and tests accordingly.

Task 2 "Simulation of the Rayleigh-Taylor instability"

The Rayleigh-Taylor instability occurs at the interface of two fluids when a fluid with higher density resides on a fluid with lower density and is subject to gravity.
In this task we will perform the simulation of a Rayleigh-Taylor instability. We will impose reflecting boundary conditions on the horizontal boundaries of the domain and periodic boundary conditions on the vertical boundaries.
The two different fluids are represented by two molecular cuboids.

- Implement periodic boundaries:
  - particles which left the domain at a boundary have to be inserted at the opposite boundary
  - particles from the boundary layer of the opposite boundary have to be inserted into the halo layer of the boundary.
- The fluids are exposed to gravity. Implement a force calculation routine which adds a gravitational force G (along the y-axis) according to

$$G = m \cdot g_{grav}$$

  It should be possible to specify the factor g_{grav}.
- We have to model two different fluids. Extend the molecules so they can be of different type. Especially they may have different