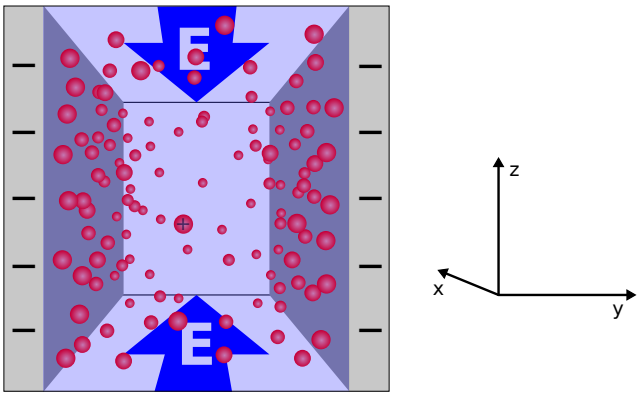
Tutorial 5

ESPResSo 3: The Lattice-Boltzmann-Method in ESPResSo: Polymer Diffusion

Olaf Lenz* Stefan Kesselheim Georg Rempfer

June 24, 2011
Institute for Computational Physics, Stuttgart University



Contents

1 Introduction

In this tutorial, you will learn basics about the Lattice-Boltzmann- Method (LBM) with special focus on the application on soft matter simulations, or more precisely on how to apply it in combination

*olenz@icp.uni-stuttgart.de

with molecular dynamics to take into account hydrodynamic solvent effects without the need to introduce explicit solvent particles.

The LBM – its theory as well as its applications – is still a very active field of research. After almost 20 years of development there are many cases in which the LBM has proven to be fruitful, in other cases the LBM is considered promising, and in some cases it has not been of any help.

2 Activating Lattice-Boltzmann in ESPResSo

By default, ESPResSo does not contain the LBM. Instead, it is a feature that can be activated and deactivated. To activate a feature, you need to recompile ESPResSo. To learn how to activate a feature, refer to the ESPResSo User's Guide.

The command **code_info** within ESPResSo returns a list of activated features and can be used to check whether recompilation has been successful.

Task:
Activate the feature `LB` *of* ESPResSo*:*

1. *Change to the* ESPResSo *directory.*
2. *Create a copy of the file* `src/myconfig-default.h` *by the name of* `myconfig.h`*. The new file should be in the* ESPResSo *directory itself,* not *in the* `src/` *subdir!*
3. *Edit* `myconfig.h` *and add the line*

   ```
   #define LB
   ```

   to activate the Lattice-Boltzmann feature.
4. *Recompile* ESPResSo *by calling* `make`*. This will rebuild* ESPResSo

3 The LB interface of ESPResSo

In ESPResSo the LB scheme and the MD scheme are not synchronized: In one LB time step typically several MD steps are performed. This allows to speed up the simulations and is adjusted with the parameter tau. ESPResSo has three main commands for the LB module: **lbfluid**, **lbnode**, and **lb_boundary**. **lbfluid** is mainly used to set up parameters and does everything that concerns the whole fluid. **lbnode** involves readout and manipulation of single LB cells. Additionally the command **thermostat** lb is used to activate the LB thermostat and to set the temperature.

For more details on these commands, please refer to the ESPResSo User's Guide.

4 Polymer Diffusion

In these exercises we want to use the LBM-MD-Hybrid to reproduce a classic result of polymer physics: The dependence of the diffusion coefficient of a polymer on its chain length. If no hydrodynamic interactions are present, one expects a scaling law $D \propto N^{-1}$ and if they are present, a scaling law $D \propto N^{-\nu}$ is expected. Here ν is the Flory exponent that plays a very prominent role in polymer physics. It has a value of $\sim 3/5$ in good solvent conditions in 3D. Discussions of these scaling laws can be found in polymer physics textbooks like [1, 2, 3].

We want to determine the diffusion coefficient from the mean square displacement (MSD) that a particle travels in the time t. For large t it should be proportional to the time and the diffusion coefficient occurs as prefactor:

$$\frac{\partial \langle r^2(t) \rangle}{\partial t} = 2dD. \tag{1}$$

Here d denotes the dimensionality of the system, in our case 3. This equation can be found in virtually any simulation textbook, like [4]. We will therefore set up a polymer in an LB fluid, simulate for an appropriate amount of time, calculate the mean square displacement as a function of time and obtain the diffusion coefficient from a linear fit. However we make a couple of steps in between and divide the full problem into subproblems that allow to (hopefully) fully understand the process.

4.1 Diffusion of a single particle

Our first step is to investigate the diffusion of a single particle that is coupled to an LB fluid by the point coupling method. Investigate the script `tutorial5-1.tcl`.

In this script an LB fluid and a single particle are created and LB is used to thermalize the system. The random forces on the particle and within the LB fluid will cause the particle to move, and its position is recorded in the file `pos.dat`. To calculate the mean-square displacement of the particle, the helper script `msd.pl` can be employed. The command

```
perl msd.pl pos.dat
```

will create the file `msd_plot.dat` which contains the MSD over time.

Task: (5 points)
Run the simulation script.
Plot the MSD. Why is there a difference between short times and long times? Can you give an explanation for the parabolic shape at short times? Use a linear fit to the curve to determine the diffusion coefficient.
The file `energy.dat` *contains the kinetic energy of the particle as a function of the elapsed simulation time. Investigate it, by plotting it. Calculate the average value of the kinetic energy* e.g. *by fitting a constant function. What value would you expect from a correct thermostat?*
Now change the box size from 16 *to* 8, 24, 32. *What can you observe? Can you explain your observation? What happens if you replace the lb thermostat (and the LB fluid) by a standard Langevin thermostat?*
Run the simulation again with different values for the friction coefficient, e.g. 1., 5., 20., 50. *Calculate the diffusion coefficient for all cases and make a plot of D as a function of γ. What do you observe? Is there any difference between the friction coefficient that you put in, and the diffusion coefficient that you obtain?*

4.2 Diffusion of a polymer

The simulation script `tutorial5-2.tcl` sets up and simulates a single bead-spring polymer in a LB fluid. The script contains a quite long warmup section so that also longer polymers can are possible. You can probably make it shorter. Also the runtime can be reduced. You should find out about necessary number of steps by yourself.

Task: (5 points)
*Run the script for different chain lengths (*8, 16, 32, 64, 128*) and look at the output files which are identical to the output files of the single particle diffusion simulation script, only this time they contain the positions of the center-of-mass of the polymer.*
Compute the diffusion coefficient, and plot it as a function of the chain length. Try to find out, why the results do not show the expected $N^{3/5}$ behaviour.

References

[1] P. G. de Gennes. *Scaling Concepts in Polymer Physics*. Cornell University Press, Ithaca, NY, 1979.

[2] M. Doi. *Introduction do Polymer Physics*. Clarendon Press, Oxford, 1996.

[3] Michael Rubinstein and Ralph H. Colby. *Polymer Physics*. Oxford University Press, Oxford, UK, 2003.

[4] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation*. Academic Press, San Diego, second edition, 2002.