Worksheet 5: Hydrodynamics and the Lattice-Boltzmann-Method

Georg Rempfer Johannes Zeman

July 2, 2015

Institute for Computational Physics, University of Stuttgart

Contents

General Remarks

- Deadline for the report is **Monday, 13th of July 2015, 12:00 noon**

- In this worksheet, you can achieve a maximum of 20(+3) points.
- To hand in your report, send it to your tutor via email.
  - Johannes (`zeman@icp.uni-stuttgart.de`) (Thursday 14:00-15:30)
- Please attach the report to the email. For the report itself, please use the PDF format (we will *not* accept MS Word doc/docx files!). Include graphs and images into the report.
- The report should be 5–10 pages long. We recommend using LaTeX. A good template for a report is available online.
- The worksheets are to be solved in **groups of two or three** people.

1 Introduction

In this worksheet, you will first have to answer some theoretical questions about hydrodynamics. In the remaining practical part, you will investigate the transition from laminar to turbulent flow using one of the Lattice-Boltzmann implementations included in ESPResSo. There are actually two implementations, one running on the CPU and one running on the GPU. The GPU version is significantly faster (up to a factor of 50 depending on the graphics card), so you are better off using the GPU implementation.

There are two other topics this tutorial will cover, which are relevant to simulations in general: visualization of your results with with high quality pictures and videos and how to run simulations unattendedly over long periods of time.

2 Short Questions - Short Answers

Task (4 points)

Answer the following questions:

- What could be considered the main advantages of the Lattice-Boltzmann-Method (LBM) over "classical" computational fluid dynamics (CFD) methods (Finite Volume or Finite Elements Navier-Stokes solvers)?
- In which cases does the LBM *not* work?
- What's the meaning of a Reynolds number and how is it defined?
- How is Stokes' law defined and what does it describe?

3 Stokes' Law, Computer Game Superheros and Low Reynolds Numbers

Imagine you are in your living room after a long day of studying. As a refreshment, you just got yourself a nice glass of delicious dihydrogen monoxide (H_2O), as suddenly Duke Nukem kicks in the door, shouts "Hail to the king, baby!", and fires at you with his supercharged shrink ray. Ouch! Within the blink of an eye, you are shrunk to micrometer size and, to complete your misery, fall into your own glass of water.

Fortunately, supercharged shrink rays cause the majority of the target's mass to be temporarily beamed to a parallel universe, so that the mass density of the target remains unchanged. However, you just happen to wear that brand new heavy-duty spherical body armor you got from *www.gimmemoreofthatfreakygeekstuffilikesomuch.com*, so your mass density is about twice the density of H_2O.

Task (5 points)

- Now being a sphere of $2R = 1\ \mu m$ in diameter, how do you expect to fall into the water? Will you plunge in smoothly or will you be stopped rather abruptly? Give reasons for your answer!
- Use Stokes' law in combination with Archimedes' principle to calculate your terminal velocity under water. Assume $\rho_s = 2000\ \frac{kg}{m^3}$ (your density), $\rho_{H_2O} = 1000\ \frac{kg}{m^3}$ (density of water), $g = 9.81\ \frac{m}{s^2}$ (gravitational acceleration), and $\mu = 0.001\ Pa\ s$.
- Why is Stokes' law applicable in this case?
- As you just figured out, without doing anything, you will sink deeper and deeper into the water. Since giving up and drowning is not an option, you start swimming to get back to the surface. Is it better to use butterfly strokes, breast strokes, or freestyle strokes to propel yourself upwards? If you had the choice, would you prefer a paddle or a ship's propeller to facilitate moving?
- Could Stokes' law still be applied in air? ($\mu_{air} = 1.8 \times 10^{-5}\ Pa\ s$, $\rho_{air} = 1.225\ \frac{kg}{m^3}$ and $v \approx 1\ \frac{\mu m}{s}$)

Hint

- In order to get an idea what this task is about, read Purcell's (very entertaining!) paper *Life at low Reynolds number*.
  You can find it under `/group/sm/2015/purcell77a.pdf`.

4 Simulation of Hydrodynamic Flow Around a Rigid Object

Now we will go back to more serious problems and perform hydrodynamic simulations using the LBM. Our system consists of a pattern of 3d periodic stripes as obstacles, surrounded by a newtonian fluid and a homogeneous force density acting on the fluid as shown in fig. 1.

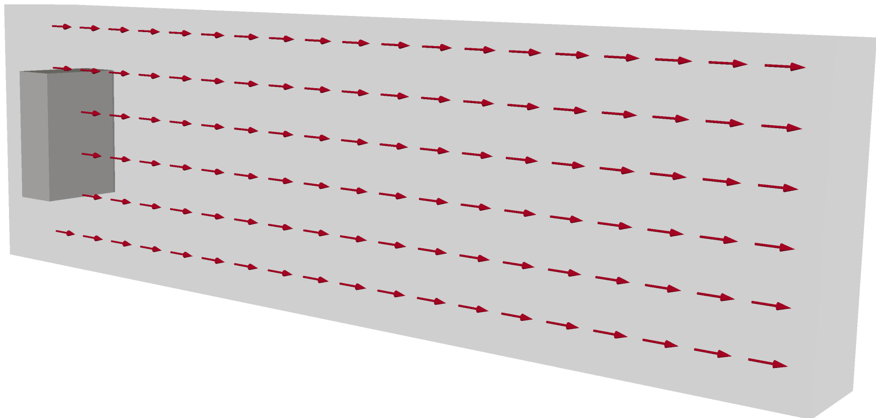

Figure 1: System to investigate. Light gray is the simulation volume with periodic boundaries and the LB fluid. Dark grey represents an obstacle with no-slip boundary conditions and red stands for the applied constant force density

Turbulence is a phenomenon comprising processes on very different time- and length scales, which makes it hard to simulate with moderate computational effort. You are welcome to experiment with the parameters to produce turbulence and any working set of parameters will be appreciated but be warned that this is a time-consuming task.

The following set of parameters will work for this task:

- fluid viscosity $\nu = 1.1$
- fluid density $\rho = 1$
- fluid force density $f = 0 \dots 0.2$
- fluid grid spacing $a = 1$
- time step $\tau = 0.01$
- box size $= 200 \times 60 \times 20$
- obstacle dimensions $= 10 \times 30 \times 20$

You will need to compile ESPResSo with the following features defined in `myconfig.h`:

`EXTERNAL_FORCES`, `LB`, and `LB_BOUNDARIES`.

Should you want to take advantage of the GPU Lattice-Boltzmann implementation, you need to compile ESPResSo with CUDA support

```
$> ./configure --with-cuda=/usr/local/cuda
```

and enable the following features in `myconfig.h`:

`EXTERNAL_FORCES`, `LB_GPU`, and `LB_BOUNDARIES_GPU`

To set up the simulation, you will need the following ESPResSo commands, apart from the commands you already used in previous tutorials:

- `thermostat off`
- `lbfluid <cpu|gpu> dens` ρ `agrid` a `tau` τ `visc` ν `ext_force` f `0 0`
- `lbboundary rhomboid corner` p_x p_y p_z `a` a_x a_y a_z
  `b` b_x b_y b_z `c` c_x c_y c_z `direction outside`

Please consult the ESPResSo user's guide if you need further information on those commands. The commands to produce output are given in section 5.1. Also note that although we are not going to use particles, you will have to set a Verlet skin with `setmd skin 0.1`

For the following simulations, you can use the Tcl script provided on the course website.

5 Visualization

For this part of the tutorial we are going to use `paraview`, an easy to use, open source visualization program for scientific data. You should find a preinstalled version on the CIP pool computers.

If you want to use it on other computers, it is part of most linux distributions' repositories or you can get a copy at http://www.paraview.org/. Versions up to 3.13 may contain a bug that causes a memory leak while rendering movies. So if your repositories contain a version up to that, better get the current release from the website.

5.1 Laminar Picture

Task (2 points)

- Create a nice plot of the stationary state of a laminar flow field around the obstacle and hand it in as jpg or png.

You can output the LB velocity field and the boundary geometry from ESPResSo in a paraview compatible format with the following commands:

- `lbfluid print vtk boundary` *`filename`*

- `lbfluid print vtk velocity` *`filename`*

For this tutorial, it will be sufficient to know about the 7 `paraview` controls shown in fig. 2.

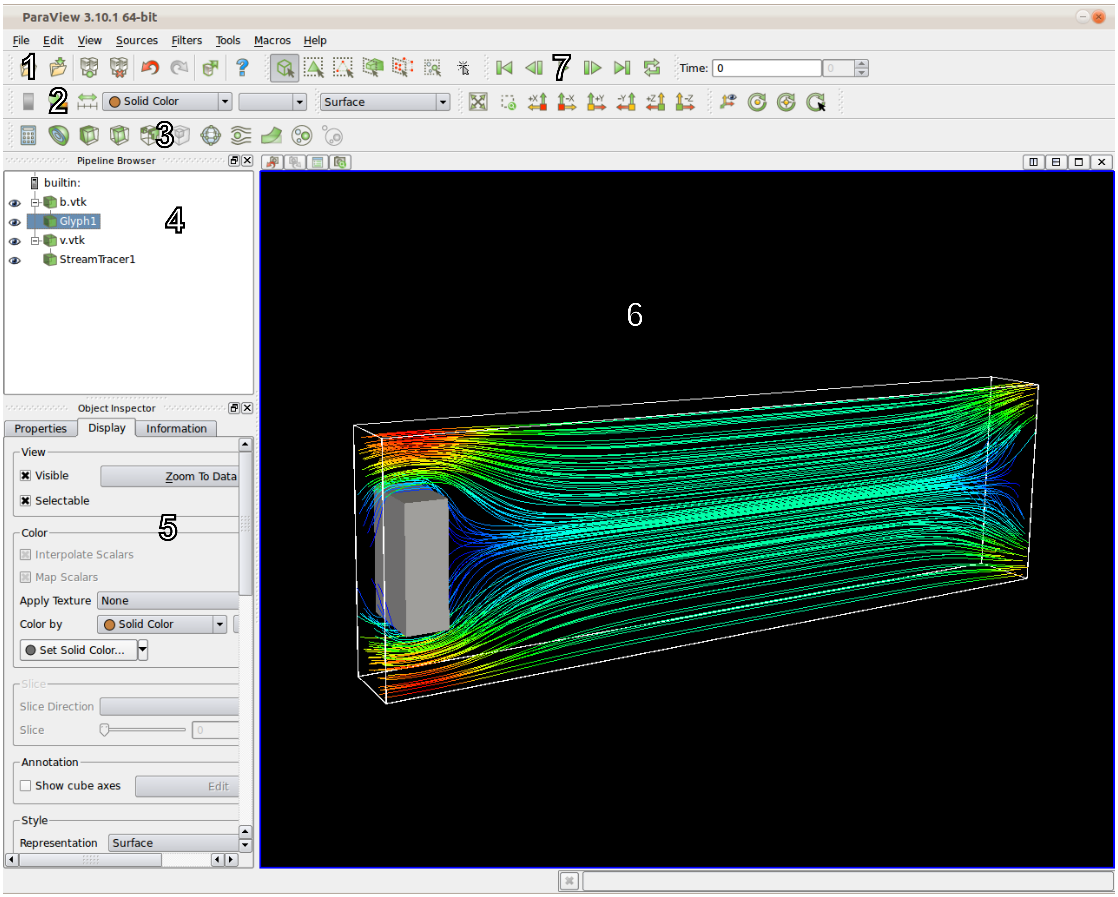

Figure 2: Paraview with a sample visualization and the most important controls.

1. Load data files into the pipeline for visualization
2. Adjust color settings of the filter selected in the pipeline browser
3. Add one of the most used filters to the element selected in the pipeline browser
4. The pipeline browser – shows loaded data files and filter applied to them for visualization
5. Configuration panel for the filter chosen in the pipeline browser
6. The preview panel – you can rotate, zoom and move this with the mouse
7. Controls for videos

With a little exploration, these controls should be self explanatory. If you need help, consult your tutor, the paraview help (F1) or the paraview online documentation at http://www.paraview.org.

5.2 Turbulent Video

> **Task** (2 points)
>
> - Create a video of the time evolution of the system with a force density from the upper end. Choose the simulation time between frames such that the video looks smooth and the occuring turbulences are interesting to watch.

Outputting so much data significantly slows down the simulation and requires 6.6 MB per frame. If you don't want to spend that much time on the simulation and the rendering or do not have that much memory available, you can just output a few hundred frames from time step 9000 on.

Creating videos with paraview works the same way as creating images. The only difference is that you have to output one vtk file for every frame in ESPResSo (about one frame every 10 integration steps will do) and include a counter in the file names. You can then open all those files at once with paraview and use the video controls to watch a preview of your animation.

You can save an animation through the "Files" menu. How the video is going to be encoded depends on the codecs available on the machine. Watch out, some codecs only allow certain video dimensions or frame rates (powers of 2).

If the resulting video is too big to hand in, store it on your CIP account and just hand in the path.

6 Unattended Simulations

After having verified that the chosen system actually produces laminar and turbulent flow, we want to quantitatively investigate the flow rate. In section 7.1 you will show that for low forces, the net flow rate scales linearily with the force density. For flow velocities that produce a non neglible convective contribution, however, there is no theory that predicts the scaling for such a complex geometry. We will therefore run the simulation for many force densities ranging from $f = 0.0$ to $f = 0.2$. Since it is a computation time intensive task to obtain data of sufficient resolution, you should automize this process so that no user interaction is required.

Task (1 point)

- Change the simulation script such that the force density can be specified as a command line argument. Determine the time needed for one simulation and adjust the given bash script such that it takes a couple of hours to sample the specified range of force densities.

Running the same simulation several times while varying a parameter can easily be achieved by writing a second program that executes the simulation in a loop and passes the parameter as a command line argument. A simple bash script will suffice in this case. The command line arguments can be accessed in an ESPResSo Tcl script through the list `$argv`.

One problem is that the bash itself does not support calculations involving floating point numbers. This can however be fixed by using `bc`. The following example demonstrates how to do this:

```
f=0
f=$(echo $f+0.1 | bc -l)
```

Due to the lack of support for floating point numbers, you will only be able to compare for equality as stings in loops and conditions. Therefore, if more complicated calculations are needed for the batched execution, a full featured language might be better suited for this task.

The given bash script `batch_execute.sh` works in the way described above and also executes the analyzation as described in section 6.1.
Adjust it so that it will run on one of the CIP pool machines for a couple of hours, sampling the force density range $f \in [0 \ldots 0.2]$ in such a way, that about a third of the calculation time is spent on the interval $f \in [0 \ldots 0.002]$, a third on $f \in (0.002 \ldots 0.02]$ and another third on $f \in (0.02 \ldots 0.2]$. Of course you can also prepare several of those scripts to split the work amongst several computers.

6.1 Automatic Analyzation

Task (1 point)

- Change the existing simulation script to calculate the time dependent net flow rate of each run and store it into a file. Complete the existing analyzation so that it reliably calculates the averages asymptotic net flow rate for each run and briefly explain how it works.

The ESPResSo command

`lbnode` x y z `print <velocity|boundary|...>`

allows you to access LB information during the simulation. Change the simulation script so that the simulation is interrupted every 100 steps and the time and the current flow rate is written to a file called `flowrate_$f.dat` with `gnuplot` readable format in a folder `results`.
The flow rate can be calculated by summing up the flow in force direction of all elements on a plane perpendicular to the force direction.

It is important that the determination of the averaged asymptotic flow rate from this time resolved flow rate profile works in a completely automatic fashion for all force densities ranging from $f = 0.00001$ to $f = 0.2$. So first run some simulations in this range (specifically at the upper and lower limit) by hand to see how long it takes to reach a stationary or oscillatory state and how much time you need for averaging to get a reliable mean value in the oscillatory case. Then adjust the analyzation routine in `batch_execute.sh` and `average.gp` so that it averages over the determined time interval.

7 Investigating the Force-Flowrate Relation

7.1 Analytical Theory for low Re

Task	(2 points)
• Prove that in the stationary state for low force densities, there is a linear relation between the applied force density and the net volume flux in our system.	

Flow of Newtonian fluids is conventionally described by the equation of continuity (1) and the Navier-Stokes equations (2)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0\,, \tag{1}$$

$$\rho \frac{\partial \vec{v}}{\partial t} + \underbrace{\rho\,(\vec{v} \cdot \vec{\nabla})\vec{v}}_{\substack{\text{convective}\\ \text{momentum}\\ \text{transport}}} = -\nabla p + \underbrace{\eta \vec{\Delta} \vec{v}}_{\substack{\text{viscous}\\ \text{momentum}\\ \text{transport}}} + \vec{f}\,, \tag{2}$$

with ρ the density, $\vec{v}$ the velocity and η the viscosity of the fluid. $\vec{\nabla}$ and $\vec{\Delta}$ stand for the in cartesian coordinates component wise gradient and laplacian respectively. By rescaling all the occuring variables, one can eliminate one of the two occuring parameters and formulate (2) in a completely unitless way, depending only on the dimensionless Reynold's number:

$$\frac{\partial \vec{v}'}{\partial t'} + (\vec{v}' \cdot \vec{\nabla})\vec{v}' = -\nabla p' + \frac{1}{Re}\vec{\Delta}\vec{v}' + \vec{f}' \tag{3}$$

Check http://en.wikipedia.org/wiki/Reynolds_number#Derivation for more information on the nondimensionalization and Reynold's number. From equation (3), one deduces that for very low Re, e.g. small densities, small systems, low velocities or big viscosities, the viscous momentum transport dominates over the convective one. Assuming this is the case, one can neglect the convective contribution and find the stationary state of an incompressible Newtonian fluid by solving the resulting Stoke's equations

$$\vec{\Delta}\vec{v} = \frac{1}{\eta}\nabla p - \frac{1}{\eta}\vec{f}\,, \tag{4}$$

$$\nabla \cdot \vec{v} = 0\,. \tag{5}$$

Pay attention to the fact that the pressure in this equation is not a given quantity but also a variable. In fact it is what couples the two equations and makes the solutions interesting. As boundary conditions we impose $\vec{v} = 0$, so called no-slip boundary conditions.

From this you should be able to deduce that the resulting $\vec{v}$ scales linearily with the applied force density, as it would be the case for a simple linear, uncoupled differential equation.

7.2 Results from the Simulations

Task	(3 points)
• Plot the relation between applied force density and asymptotic average net flow rate. Discuss the result.	

Questions you could address are:

- For which range of f does the law deduced in section 7.1 hold?
- Where does the flow field start to really look turbulent?
- Can you propose any law that describes the scaling of the asymptotic average net flow rate on the whole investigated interval of f?
- Is there information in the literature for this behaviour and if yes, is it theoretically deduced and from what or is it experimentally or heuristically obtained?

8 Bonus Questions

Task (3 points)

- ESPResSo does not specify a unit system and depending on how one sets certain units, other ones are rescaled. Assuming our system consists of water with density $\rho = 1000\,\mathrm{kg/m^3}$ and kinematic viscosity $\nu = 10^{-6}\,\mathrm{m^2/s}$ and the video you created shows the system in real time – how big is the system?
- In the produced video you saw that the flow lost its mirror symmetry in the y-direction after a while although the geometry of the problem and the applied force density is symmetric. What caused this symmetry break in the simulation? What causes it in reality? How could one cause it earlier to maybe save some computational time?
- Assuming one naively set $k_\mathrm{B}T = 1$, what temperature would that correspond to in the real world?

Hint

- You can find information about how to figure out the unit system in section 4.1 of the bachelor thesis delivered with this homework.