Electro-Osmotic flow

This part is practical. It is concerned with the movement of ions in an charged slit pore, which consists of two infinite charged, parallel walls. It is similar to the systems that are discussed in the Bachelors thesis of Georg Rempfer [1] which is recommended for reading. In this exercise, you should simulate such a system with ESPResSo. You are supposed to use a Lattice Boltzmann fluid coupled to explicit ions which are represented by charged Week-Chandler-Anderson spheres. In addition to the charge on the walls, the ions are also subject to an external electrical field parallel to the walls. Electrostatics should be handled by the P3M algorithm with ELC. A set of realistic parameters and an more in detail description of the system can be found in the thesis. You should measure the flow profile of the fluid and the density and velocity profiles of the ions. The case of the slit pore can be solved analytically either in the case of only counter ions (the so called salt free case) or in the high salt limit (Debye-Hueckel-Limit). Calculate the ion profiles in one or both of these cases and compare the results with the simulation.

Background

Electro-Osmotic flow (EOF) is a hydrodynamic flow created by the movement of ions driven by an external field. See your tutor for general info on EOF.

Tasks

You will build a simulation script using ESPResSo to simulate an Electro-Osmotic flow with explicit ions driven with an external electric field. In general it is a good idea to start with a simple system first and incrementally increase the complexity of the system. This approach makes it easier to spot mistakes and allows for verification during the development process. While building your system, you should be careful about keeping the simulation time short (development stage). Once you are satisfied with your script you can then try better (more expensive) simulations (production stage).

1. Create free particles between two plates. Add fixed particles as discretized surface charges.
   - inspect for correctness (visualization is fine, but provide a screenshot)
2. Add the interactions (particle-particle and particle-wall)
   - inspect for correctness (visualization is fine, but provide a screenshot)
3. Add electrostatics with P3M (don't expect valid results, P3M is not suited for 2D systems)
   - inspect for correctness (visualization is fine, but provide a screenshot)
4. Correct the electrostatics with ELC
   - validate with Poisson-Boltzmann prediction (It might be useful to have a look at *Observables* and *Accumulators* in the Analysis section of the documentation.)
5. Add hydrodynamic interactions with LB. Apply an external electric field to the mobile ions.
   - inspect for correctness
   - validate the velocity profile with expected statistical behavior (*Observables* might help)
6. Looking Further
   - EOF is characterized by a flat flow profile, explain your results
   - name three ways your simulations can be modified to attempt to achieve a flat profile
   - try one of them (be clever, stay away from situations which are computationally expensive)

A good starting point for your simulation scripts are the ESPResSo Tutorials `https://espressomd.github.io/tutorials4.1.4.html` and the ESPResSo Documentation `https://espressomd.github.io/doc4.1.4/index.html`. The tutorials provide a good insight on how to set-up your system to use specific algorithms. A detailed explanation of the parameters for the algorithms can be found the documentation. Sometimes it might also be useful to have a look in the ESPResSo Samples `https://github.com/espressomd/espresso/tree/4.1/samples` folder. Parameters for your simulation can be found in pp. 19 [1].

Common pitfalls

- Do NOT try to build your script in one go. Take small steps and validate as often as possible
- Visualize your system. Use the online visualizer or other tools by exporting the particle trajectories.
- Output necessary simulation data to disk and analyze/compare it after the simulations. If you dont and your analysis/comparison script is wrong you will have to rerun your entire simulation.

There are many parameters to choose, many will seem arbitrary but they can make or break your simulations. In the following some hints are given which also include simulation parameters. Be aware that changing those can break your simulaiton.

Integrators

```
system.time_step = 0.01
system.cell_system.skin = 0.4
```

Thermostats

- when you use the Langevin thermostat, keep gamma to unity

  ```
  system.thermostat.set_langevin(kT=kT, gamma=1.0, seed=42)
  ```

- when you want to switch from Langevin thermostat to LB

  ```
  # turn off Langevin thermostat
  system.thermostat.turn_off()
  # remove existing particle motion
  system.galilei.kill_particle_motion()
  # remove existing center of mass momentum
  system.galilei.galilei_transform()
  ```

- when you want to use LB

  ```
  # for CPU
  lbf = lb.LBFluid(agrid=1.0, dens=1.0, visc=1.0,
                   tau=0.01, kT=kT, seed=seed)
  # for GPU
  lbf = lb.LBFluidGPU(agrid=1.0, dens=1.0, visc=1.0,
                      tau=0.01, kT=kT, seed=seed)
  # remember to activate the friction coupling
  system.thermostat.set_lb(kT=kT, seed=seed, gamma=20.0)
  ```

Electrostatics

- start with a low Bjerrum length in simulation units

```
l_bjerrum = 2.0
```

- when you use P3M don't set the accuracy too small

```
p3m = electrostatics.P3M(prefactor=l_bjerrum, accuracy=1e-4)
```

- when you use ELC don't set the accuracy too small

```
elc = electrostatic_extension.ELC(gap_size=elc_gap, maxPWerror=1e-4)
```

- warmup (integrate a few timesteps) your system when adding a new actor

Geometry and Constraints

- visualizations using external tools (exporting trajectories with VTF or H5MD) dont show constraints, only the online visualizer does
- constraints with a potential actually occupy space, find out the accessible volume to the ions
- use a cubic simulation box but keep some empty space between the upper plate and simulation boundary. This is used for the ELC gap, ask your tutor for further informations.

Potentially useful reading material

Lattice Boltzmann

- ESPResSo Documentation: `https://espressomd.github.io/doc4.1.4/lb.html`
- ESPResSo Tutorials *Lattice Boltzmann*: `https://espressomd.github.io/tutorials4.1.4.html`
- For further details on the Lattice Boltzmann Method: [2]

Electrostatic Methods

- ESPResSo Tutorials *Charged systems*: `https://espressomd.github.io/tutorials4.1.4.html`
- ESPResSo Documentation: `https://espressomd.github.io/doc4.1.4/electrostatics.html`

Electro-Osmotic flow

- ESPResSo Tutorial: `https://espressomd.github.io/tutorials4.1.4/07-electrokinetics/07-electrokinetics.html`
  Careful, this tutorial is using a continuum model to simulate this system!
- Bachelor Thesis [1]

References

[1] Georg Rempfer. "Lattice-Boltzmann Simulations in Complex Geometries". Bachelor thesis. University of Stuttgart, 2010.

[2] Timm Krüger et al. The Lattice Boltzmann Method: Principles and Practice. Cham: Springer, 2017. ISBN: 9783319446479. DOI:10.1007/978-3-319-44649-3