Simulating Soft Matter with ESPResSo, ESPResSo++ and VOTCA

6th Summer School at the ICP
Generic Program

• **Mornings** (9:00-12:30)
  2 Lectures, 1 1/2 h each; 1/2 h Coffee break
  Sometimes 2 tracks

• **Afternoon** (14:00-17:30)
  Hands-on Sessions, Up to 4 tutorials per session

• Poster Sessions plus Flash-talks on Monday, Posters can stay mounted the whole week

• BBQ on Monday

• Conference “Speakers” Dinner on Thursday:
  “Trollinger” in City center (S-Bahn Rotebühlplatz)
  Pay yourself for participants ~20€ for food and drinks
# Real Programm

<table>
<thead>
<tr>
<th>Monday, 9.10</th>
<th>Tuesday, 10.10</th>
<th>Wednesday, 11.10</th>
<th>Thursday, 12.10</th>
<th>Friday, 13.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.00-9.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.00-9.45</td>
<td>Welcome and Introduction (C. Holm)</td>
<td>ESPResSo Electrostatics + GPU Electrokinektesis (M. Kuron/F. Weik)</td>
<td>Intro to Lattice Boltzmann (U. Schiller)</td>
<td>U. Schiller (Clemson)</td>
</tr>
<tr>
<td>9.45-10.30</td>
<td>Intro to ESPResSo, Intro to ESPResSo++</td>
<td>ESPResSo Electrostatics + Electrokinetics (M. Kuron/F. Weik)</td>
<td>Intro to Lattice Boltzmann (U. Schiller)</td>
<td>S. Gekle (Bayreuth)</td>
</tr>
<tr>
<td>10.30-11.00</td>
<td>COFFEE BREAK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.00-11.45</td>
<td>First Simulations in ESPResSo</td>
<td>Votca Coarse-graining (D. Adrienko)</td>
<td>Concurrent coarse-graining with AdRes (R. Cortes-Huerto)</td>
<td>T. Bereau (MPI-P Mainz)</td>
</tr>
<tr>
<td>11.45-12.30</td>
<td>First Simulations in ESPResSo++</td>
<td></td>
<td></td>
<td>O. Valsson (MPI-P Mainz)</td>
</tr>
<tr>
<td>12.30-14.00</td>
<td>LUNCH BREAK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.00-15.30</td>
<td>ESPResSo (install+LI) (tutorials)</td>
<td>ESPResSo (tutorials)</td>
<td>ESPResSo (Lattice Boltzmann)</td>
<td>ESPResSo (Aktiv matter tutorial)</td>
</tr>
<tr>
<td>15.45-16.15</td>
<td>ESPResSo under the hood (tutorials)</td>
<td>ESPResSo (Electrostatics + Electrokinetics)</td>
<td>ESPResSo (Lattice Boltzmann)</td>
<td>ESPResSo++ (under the hood)</td>
</tr>
<tr>
<td>16.15-18.00</td>
<td>ESPResSo under the hood (tutorials)</td>
<td>ESPResSo (Electrostatics + Electrokinetics)</td>
<td>ESPResSo (Lattice Boltzmann)</td>
<td>ESPResSo Developer Meeting</td>
</tr>
<tr>
<td>18.00-19.00</td>
<td>Posterflash + Poster Session</td>
<td></td>
<td></td>
<td>Speakers Dinner</td>
</tr>
<tr>
<td>19.00-22.00</td>
<td>BBQ + Poster Session (continued)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lecture | Tutorial | Scientific Talk | Social
Important POI

- **Kitchen**: Drugs for the mind
- **Computers**: Please get an account at the registration desk
- **WLAN**: Get the SSID and WPA2 Key
- **Lunch** at Mensa: Follow the crowd at around 12:30
Dynamische Simulation von Systemen mit großen Teilchenzahlen

Abbildung: Rissausbreitung in Aluminiumoxid, 75.000 Teilchen
Introduction to Soft Matter Simulations

Christian Holm

Institut für Computerphysik, Universität Stuttgart
Stuttgart, Germany
Intro to Soft Matter Simulations

• What is Soft Matter?

• What can simulations do for you?

• What is needed to perform good simulations?

• Bits and pieces of necessary background information for understanding molecular simulations

• ESPResSo: history, aim, background
What is Soft Matter?
What is Soft Matter?

....and why is it interesting?
What is Soft Matter?
What is Soft Matter?

- Gummy bears, gels, networks: Rubber, low fat food,
- Fibers (z.B. Goretx, Nylon)...
- Colloidal systems: milk, mayonnaise, paints, cosmetics...
- “Simple” plastics: yoghurt cups, many car parts, CDs, ...
- Membranes: cell walls, artificial tissue, vesicles...
- Many parts of the cell, cytoskeleton, nucleus
- Most biomolecules (RNA, DNA, proteins, amino-acids)
- Liquid crystals
- Many applications: smart materials (actuators, sensors, photonic crystals), biotechnology, biomedicine (hyperthermia, drug targeting, cell separation techniques), model systems for statistical physics
Length Scales of Soft Matter

- 1 Å
- 1 nm
- 10 µm
- 1 mm

- $10^{-15}$ m
- $10^{-12}$ m
- $10^{-9}$ m
- $10^{-6}$ m
- $10^{-3}$ m
- $10^{0}$ m
- $10^{3}$ m
- $10^{6}$ m
Length Scales of Soft Matter

- $1 \text{ Å}$
- $1 \text{ nm}$
- $10 \mu \text{m}$
- $1 \text{ mm}$

$10^{-15} \text{ m}$ $10^{-12} \text{ m}$ $10^{-9} \text{ m}$ $10^{-6} \text{ m}$ $10^{-3} \text{ m}$ $10^0 \text{ m}$ $10^3 \text{ m}$ $10^6 \text{ m}$
Who needs Simulations?

- **Goal:** Understanding and prediction of interesting systems
  - Computer science: Network simulations, “emulations” of not-yet-existing CPUs, …
  - Economy: Simulations of economical cycles
  - Biology: Simulations of metabolic networks, ecological simulations (e.g. Predator-prey-systems, population dynamics)
  - Physics: Simulations of quantum systems, simulations of mechanical systems, astronomical simulations, weather prediction

![Images](image1.jpg)
Why using simulations in physics?

- All laws of nature can be expressed as mathematical formulas
- However, only few physical systems can be solved analytically
- Simulations can be used to numerically solve the most complex formulas and to compare them to experimental results
- System properties can be estimated without creating the system (cheaper, simpler, faster and/or less dangerous, well controlled)
Natural Speed-ups + …

Moore’s Law: Transistor Counts double every two years

Physical limit to transistor density!
Future: Computer power/1000 Euro?
Need to exploit parallelism
GPU Motivation (I): Performance Trends

Peak Double Precision FLOPS

- NVIDIA GPU
- x86 CPU

Peak Memory Bandwidth

- NVIDIA GPU
- x86 CPU

Other architectures (GPUs) and...
Algorithms

- Not only the computing power has improved dramatically.
- In the last 60 years numerous novel algorithms and methods have been developed that lead to major breakthroughs:

\[ T_{CPU} \approx N^{7/3} \]

\[ \Rightarrow \quad \text{Multigrid (1984) } T_{CPU} \approx N \]

- Algorithms and improvements of the methodology are as important as more powerful computers.
In a molecular simulation, the evolution of the states of a molecular system needs to be simulated.

In principle, only a pure quantum mechanical description of such a system is exact (careful, even here are pitfalls! How many exact solutions are known?)
  - Only very small systems can be simulated on that level
  - The system has to be simplified ("coarse-grained")
  - First step: Classical Atoms and Interactions

Real systems have ~$10^{23}$ atoms
  - A statistical description is needed
  - Only a part of a molecular system can be simulated
  - The simulated system has significant boundaries
Coarse-graining

- A model consists of a number of Degrees of Freedom (e.g. the atom positions) and the Interactions between them
- Coarse-graining:
  - reduce the number of degrees of freedom by keeping only the “important” degrees of freedom
  - Use “effective” interactions
- Classical first step: atoms and interactions (all-atom or atomistic)
- Further coarse-graining is often needed and useful
- For soft matter we are often on the molecular and mesoscopic level

Quantum

All-atom

Molecular

Mesoscopic Fluid Methods

Continuum
Computational Approaches

• **Quantum**: ab-initio QM or first principles high-level QM, pHF, MP2, Car-Parrinello MD, Born-Oppenheimer MD, TB-DFT, hybrid embedded QM/MM, ...

• **Atomistic**: Classical Force Field AA MD, MC

• **Coarse-grained**: Classical DFT, Molecular Dynamics, Monte Carlo, Field theoretic methods (SCFT)

• **Mesoscopic Fluid**: Lattice-Boltzmann, MPC, DPD, SPH

• **Continuum Solvers**: Computational Fluid Dynamics codes (Navier-Stokes), Poisson-Boltzmann, Lattice-Boltzmann, FEM
Available Programs:

• First principles Quantum: TURBOMOLE, Molpro (Stuttgart), Gaussian, ORCA...

• DFT: CP2K, Car-Parrinello MD, Quantum Espresso, Wien2K,... Look on www.psi-k.org

• All-Atom: GROMOS, GROMACS, NAMD, AMBER, CHARM, DL_POLY, LAMMPS...

• Coarse-grained: DL_MESO, LAMMPS, ESPResSo, ESPResSo++, OCTA,...

• Mesoscopic Fluid: Lattice-Boltzmann: openLB, SailFish, , LBM-C, WALBERLA

• Continuum: For PB: Delphi, APBS,UHBD

For FEM: DUNE, more on http://www.cfd-online.com/Wiki/Codes

HARLEM, PROPHET, much more than I can list
Making Molecular Simulations

How to make a molecular simulation?

- Choose the system to be simulated
- Choose the model and coarse-graining level of the simulation
- Determine the initial state of the model
- **Simulate** the model (using an appropriate algorithm and appropriate tools)
- Analyze and interpret the results

→ Executing the simulation is only a small part of the work!
Possible errors

- Simulations have plenty of sources for errors!
  - Errors of the boundary of the system
  - Errors of the initial state
  - Errors of the model / level of coarse-graining
  - Numerical errors (Errors of the simulation)
  - Errors of the interpretation / analysis

Theory and experimental verifications are still needed

Remember Murphy‘s Law!
What do I need to know...

...before I start simulating?

- Statistical Mechanics
- Theory behind my system (i.e. Soft and Condensed Matter theory)
- The program I am using (best way is to write it yourself!)
- Background of the algorithm (strength, weakness, limitations)
- Clever ways of analyzing the data
Aim of this week long tutorial?

- Describe some Algorithms within ESPResSo and ESPResSo++:
- Long Range Interactions, LB Hydrodynamics, Active Matter and Simulation Strategies
- VOTCA, AdResS,

Some Sample Applications
...there is much more you need to know....

- Bits and pieces, get ideas
- Meet developers for specific questions
Periodic Boundary Conditions

- Simulated systems are much smaller than “real” systems
  - Boundaries make up a significant part of the system!
  - Surface/Volume not small (i.e. for N=1000 the boundary makes up 49%)

- Trick: *Periodic boundary conditions*

- The simulated system has infinitely many copies of itself in all directions

- A particle at the right boundary interacts with the particle at the left boundary in the image

- *Minimum image convention*: Each particle only interacts with the closest image of another particle (i.e. interaction range L/2)

- Pseudo-infinite system without boundaries

- Significantly reduces boundary-related errors

- More tricky for long range interactions…
Example: Modeling Liquid Argon

- Very simple system:
  - Noble gas: no bonds between atoms
  - Closed shell: almost spherical
- Contributions to the interaction (from QM):
  - Pauli exclusion principle: strongly repulsive core (exact functional form does not matter)
  - Van-der-Waals interaction: attractive interaction for larger distances $\sim 1/r^6$

Semi-empirical 
**Lennard-Jones-Potential:**

$$V^{LJ}(r) = 4\varepsilon \left( \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right)$$

Liquid argon: $s = \text{3.4 Å}$, $e = \text{100 cm}^{-1}$
All-Atom Models

- Most commonly used model
- Each atom is represented by one spherical particle
- *A force field (FF)* describes the interactions between the atoms and consists of
  - a set of equations
  - a long table of parameters for all atom type pairs
- For different applications, various different force fields exist (e.g. GROMOS, AMBER, OPLS, Charm…)
- The interactions can be split into two groups:
  - Non-bonded potentials: e.g. Lennard-Jones, Coulomb
  - Bonded potentials for bonded atoms
Non-bonded Potentials

- Non-bonded potentials model the interaction between atoms that do not have bonds
- Lennard-Jones potential accounts for Pauli exclusion and van-der-Waals interaction:
  \[ V_{LJ}(r_{ij}) = 4\epsilon_{ij} \left( \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right) \]
- Coulomb interaction for charged atoms:
  \[ V_{C}(r_{ij}) = \frac{q_{i}q_{j}}{\varepsilon_{r}r_{ij}} \]
  - Beware: The Coloumb interaction is long-ranged. This may require special measures to compute it!
  - In some force fields, usually uncharged atoms can carry partial charges to account for polarization effects in certain compounds (for example water)
Bonded Potentials

- Bonded potentials model the bonds between atoms.
- Bond-stretching: harmonic 2-body potential models bond length:
  \[ V_b(r_{ij}) = \frac{1}{2} k_{ij}^b (r_{ij} - b_{ij})^2 \]
  - Classical spring potential!
- Bond-angle potential (3-body) models bond angle:
  \[ V_a(\theta_{ijk}) = \frac{1}{2} k_{ijk}^\theta (\theta_{ijk} - \theta_{ijk}^0)^2 \]
  or
  \[ V_a(\theta_{ijk}) = \frac{1}{2} k_{ijk}^\theta \left( \cos(\theta_{ijk}) - \cos(\theta_{ijk}^0) \right)^2 \]
Dihedral Potentials (4-body)

- The dihedral angle is the angle between the planes of 4 bonded atoms
- *Improper dihedrals* keep planar groups planar (e.g. aromatic rings):
  \[ V_{id}(\xi_{ijkl}) = k_\xi (\xi_{ijkl} - \xi_0)^2 \]
  - again: harmonic potential
- *Proper dihedrals* model cis/trans conformations:
  \[ V_d(\phi_{ijkl}) = k_{\phi}(1 + \cos(n\phi - \phi_0)) \]
How to find the FF parameters?

- Fit experimental data (density, $g(r)$, diffusion, heat of vaporization, ...)
- use QM calculations to calculate some interaction parameters
- FF work for the situation where they were parametrized, hope carries us along...(transferability)
- Combining FF parameters is non-trivial, often needs reparametrization
Pros and Cons of FF

**PRO:**
- Fast and easy to use (practically linear scaling)
- Visualization of microscopic behavior
- Mechanistic insight

**CON:**
- Quality difficult to assess
- Chemical reactions difficult to model
- Orbital interactions (polarizability) often not included
Coarse-grained Models

- Large and complex molecules (e.g. long polymers) can not be simulated on the all-atom level
- Requires coarse-graining of the model
- Coarse-grained models are usually also particles (beads) and interactions (springs, …)
- A bead represents a group of atoms
- Coarse-graining a molecule is highly non-trivial, see systematic coarse-graining, VOTCA, AdResS
Gaussian Polymer in a $\Theta$-solvent

- Conformational properties of a Gaussian polymer in a $\Theta$-solvent are that of a random walk
- Basis for bead-spring model of a polymer!
- Use a harmonic potential for the bonds:
  \[ V_h(r) = \frac{k}{2} (r - r_0)^2 \]
- We can compute the partition function exactly
  \[ H_0 = \frac{1}{2} \frac{3k_BT}{b^2} \sum_{i=0}^{N-1} |\vec{r}_i - \vec{r}_{i+1}|^2 \]
- Random walk and bead-spring model generate the same partition function!
Gaussian Chains in Good Solvent

- $\Theta$-solvent is a special case!
- Solvents are good or poor w.r. to the polymer
- Good solvent can be modeled via a repulsive potential
  - Use the repulsive part of Lennard-Jones (aka Weeks-Chandler-Anderson)

$$V_{\text{WCA}}(r) = \begin{cases} 
  V_{\text{LJ}}(r) + \varepsilon & \text{if } r < 2^{1/6} \sigma \\
  0 & \text{otherwise}
\end{cases}$$

- FENE (Finite Extensible Nonlinear Elastic) bond
  - Has a maximal extension/compression
  - Very similar to harmonic potential at $r_0$

- $V_{\text{FENE}}(r) = -\frac{1}{2} \epsilon (\Delta r_{\text{max}})^2 \log \left( 1 - \frac{r - r_0}{\Delta r_{\text{max}}} \right)$
Gaussian Chains in Poor Solvent

- Poor solvent can be modeled via a full Lennar-Jones potential.
- Polymer monomers experience an attraction, since they want to minimize contact with solvent.
- The quality of the solvent can be changed by varying the attraction via the interaction parameter $\varepsilon$ and the cut-off.
- Scaling laws with Flory exponent $\nu$:
  - RW $\nu = 0.5$
  - SAW $\nu = 0.588 (3/5)$
  - Globule $\nu = 1/3$
  - Rod $\nu = 1$
Charged Polymers

Mapping procedure for the Computer Model

Example: NaPSS (sulfonated polystyrene)

monomers, ions → beads with charge fraction $f$

bond potential → nonlinear springs

solvent → dielectric background $\varepsilon$

solvent quality $\tau$ → effective bead–bead interaction
Molecular Dynamics
Basic idea of Molecular Dynamics (MD):
- The system consists of point particles and interactions (e.g. atoms and their interactions)
- Solve the classical equations of motion for the particles on the computer:

\[
\vec{F}_i = m\vec{a}_i \\
\text{or} \quad \ddot{x}_i = \frac{f(x_i)}{m}
\]

- Can be applied to a wide range of problems:
  - Molecular systems (gases, fluids, polymers, proteins, liquid crystals, …)
  - Granular materials (sand, sugar, salt, …)
  - Planetary motion
  - Nuclear missiles….
What is a good integration algorithm?
- Easy to implement, fast to compute
- Numerically stable for large time-steps to allow for long simulations
- Trajectory should be reproducible
- Should conserve energy, linear and angular momentum
Short-time Stability

- Depending on the problem at hand, different properties of the integration algorithm are important.
- For some systems, it is important that the algorithm has a minimal error in the trajectory (“short-time stable”) (e.g. satellite orbits).
- Note that the error in the trajectory always grows exponentially over time due to positive Lyapunov exponents.
Long-time Stability

- In molecular simulations, we want to compute statistical averages (i.e. ensemble averages) of observables.
- MD uses the *Ergodic hypothesis*: \( \langle A \rangle_{\text{trajectory}} = \langle A \rangle_{\text{ensemble}} \)
- Accurate trajectories are not important.
- Instead, the correct physical ensemble should be described throughout the simulation:
  - Conservation of energy, linear and angular momentum
  - Time-reversibility
  - (In fact: conservation of phase space)
- Integrators that do this are “long-time stable” (or “symplectic”)
Verlet Algorithm

- **Verlet Integrator** (1967) is more accurate than Euler's method, and it is long-time stable(!)
- To derive it, Taylor expand $x(t)$ forward and backward in time

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{f(t)}{2m} \Delta t^2 + \frac{\partial^3 x}{3! \partial t^3} \Delta t^3 + O(\Delta t^4)$$

$$x(t - \Delta t) = x(t) - v(t)\Delta t + \frac{f(t)}{2m} \Delta t^2 - \frac{\partial^3 x}{3! \partial t^3} \Delta t^3 + O(\Delta t^4)$$

- This results in

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + \frac{f(t)}{m} \Delta t^2 + O(\Delta t^4)$$

- Straightforward algorithm, long-time stable
- Bootstrapping problem: Requires $x(t - \Delta t)$ for the initial step
Velocity Verlet Algorithm

\[ x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 \]

\[ v(t + \Delta t) = v(t) + \frac{a(t) + a(t + \Delta t)}{2}\Delta t \]

- Mathematically equivalent to Verlet algorithm
- Same accuracy \( \mathcal{O}(\Delta t^4) \)
- No bootstrapping problem
- Requires initial velocities instead
- Symplectic- preserves shadow Hamiltonian

→ Standard algorithm for MD simulations of atomistic and molecular systems
MD in various Ensembles

- Equations of motion are energy conserving
  - NVE (micro-canonical) ensemble
- Dynamics can be modified to yield other ensembles:
  - NVT: canonical ensemble
  - NPT: isothermal – isobaric
  - $\mu$PT: Gibbs ensemble
- Often achieved via changing the equations of motions (i.e. barostats, thermostats,...)
- Methods that go beyond standard MD are often needed
Langevin Dynamics

- Simulated molecules are usually not in vacuum. Air or solvent molecules collide constantly with the molecules, leading to Brownian motion.
- Simulating all solvent particles would be tedious and time consuming.
- **Langevin Dynamics (LD)** models solvent kicks via a random force and a friction:
  \[ m_i \ddot{r}_i = -\nabla V (\{r_i\}) - \Gamma \dot{r}_i + \xi_i(t) \]
  - Drag force
  - Random force
- Nice side-effect: LD thermalizes the system (simulates constant temperature, NVT ensemble).
- Locality of the random force destroys hydrodynamic interactions.
Mean-squared deviation (MSD) in a Langevin Simulation

\[ \langle |x(t) - x(0)|^2 \rangle \]

Ballistic regime

Diffusive regime

Slope \( \propto t \)

\( \tau = m/\gamma \)
Advanced MD Techniques

- Parallel tempering
- Metadynamics, Wang-Landau sampling
- Widom insertion for measuring chem. pot. m
- Flux forward sampling / Transition path sampling / other rare event techniques
- Expanded ensemble techniques
- Umbrella sampling
- Steered MD
- MC/MD hybrids.... and many more...
MD versus Monte-Carlo (MC)

- Properties of Monte-Carlo as compared to Molecular Dynamics:
  - Does not (easily) allow to observe dynamics
  - Easier to implement
  - Harder to parallelize
  - No time-step required
  - Good random number generator required
  - Faster for some systems (special moves!)
  - Often need physical insight to select good MC moves
Some remarks on the history and story behind ESPResSo
History of ESPResSo

- Start in 2001 (13.11 first commit by A. Arnold) at the MPI-P in Mainz
- Codename "TCL_MD, written in C and tcl
- Later name changed to "Extensible Simulation Package for Research on Soft Matter:
- Main Idea: create an efficiently parallized MD code with fast P3M electrostatics as an extensible-flexible research tool for Soft Matter related problems
- Early 2005 => Journal paper ready
History of ESPResSo

- 2006 first Espresso Workshop at FIAS (Frankfurt)
- Code repository remained at MPI-P (T. Stühn)
- Since 2010 maintained and mainly developed at ICP Stuttgart

http://espressomd.org
My END