Computational Science

Molecule Dynamic simulation

Task:

Implementation of an MD-simulation for Lennard-Jones particles/ ions/ water in 2D and 3D. →interaction between the particles an thereby causes spatial motion by iterativ calculations

Computational Science 1

- Task: 1
- Concept: 2
- Workflow 3
- So, 13.01.19 3
- Selbstständige Arbeitsphase 3
- So, 20.01.19? 4
- Overview: 5
- Information about modules 7

Data storage module 7

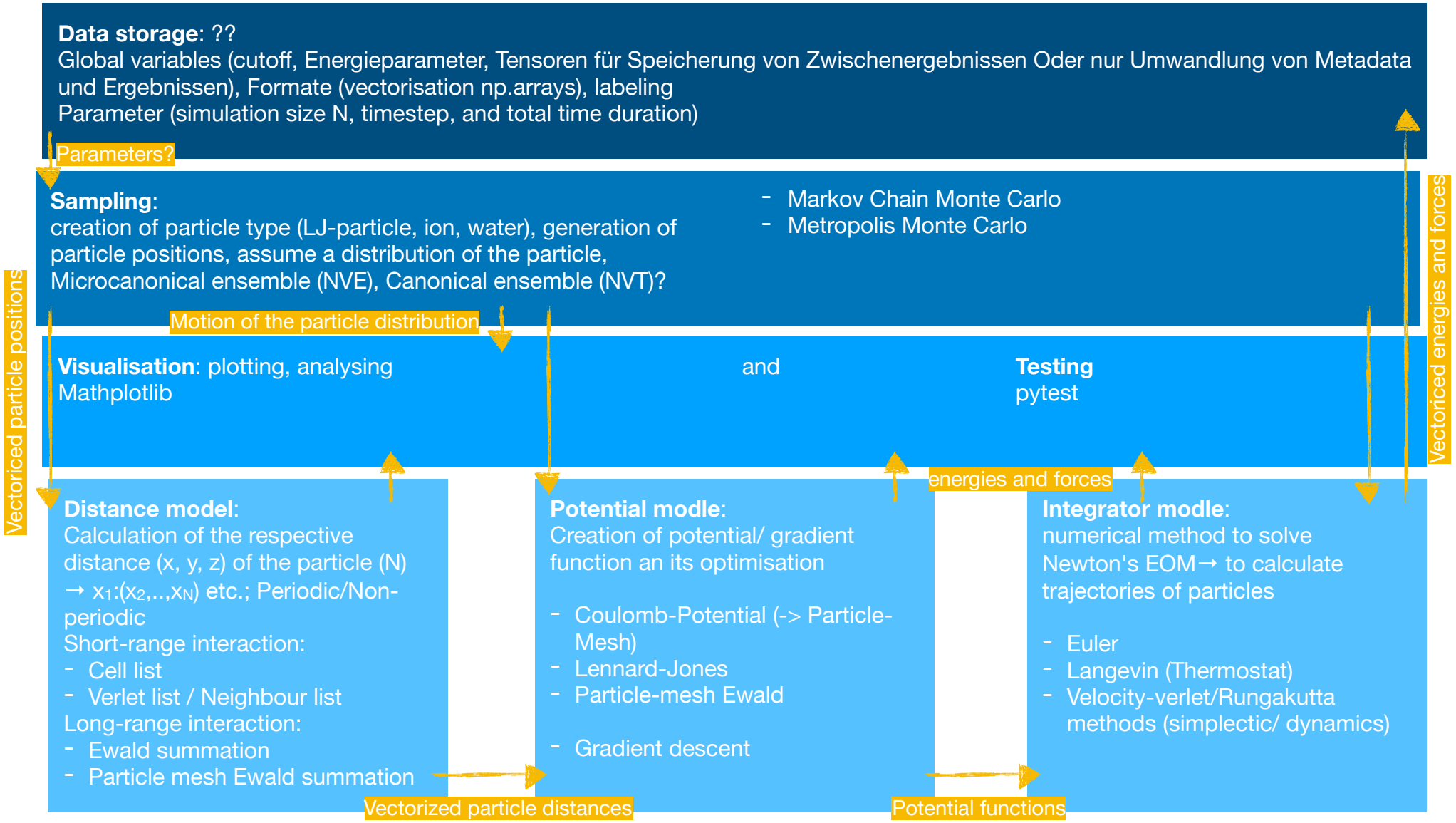
Sampling module 7

Distance module 8

Potential module 9

Integrator module 9

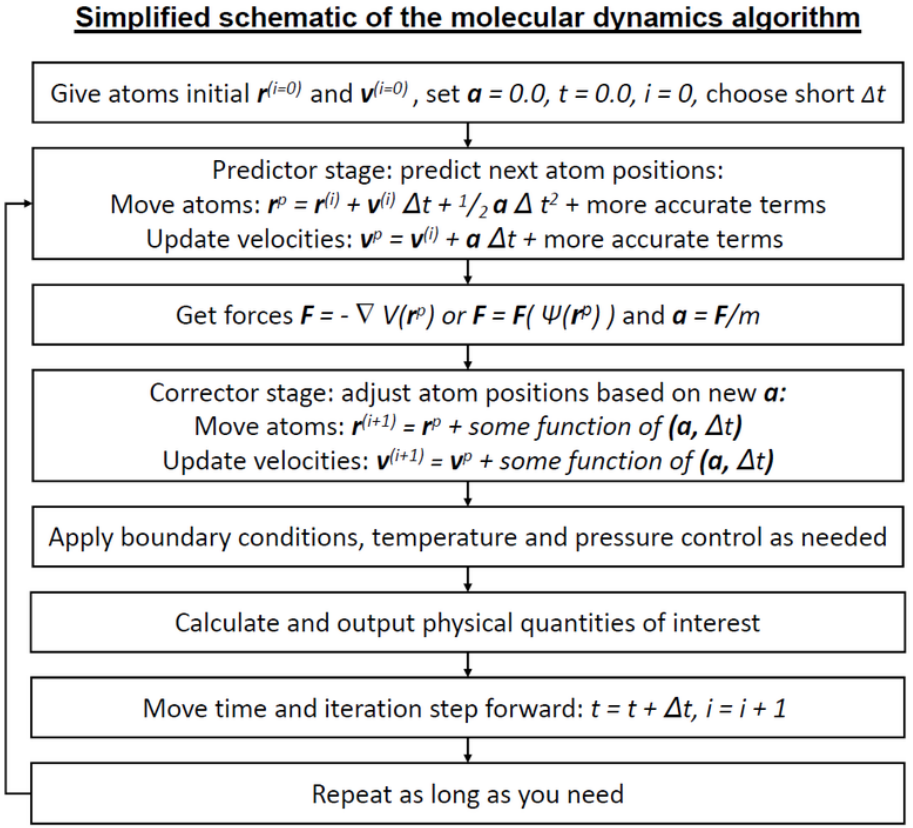
Concept:


Data storage: ??
Global variables (cutoff, Energieparameter, Tensoren für Speicherung von Zwischenergebnissen Oder nur Umwandlung von Metadata und Ergebnissen), Formate (vectorisation np.arrays), labeling
Parameter (simulation size N, timestep, and total time duration)
Parameters?
Sampling:
creation of particle type (LJ-particle, ion, water), generation of particle positions, assume a distribution of the particle, Microcanonical ensemble (NVE), Canonical ensemble (NVT)?
- Markov Chain Monte Carlo
- Metropolis Monte Carlo
Vectoriced particle positions
Motion of the particle distribution
Vectoriced energies and forces
Visualisation: plotting, analysing
Mathplotlib
and
Testing
pytest
energies and forces
Distance model:
Calculation of the respective distance (x, y, z) of the particle (N) → x_1:(x_2,..,x_N) etc.; Periodic/Non-periodic
Short-range interaction:
- Cell list
- Verlet list / Neighbour list
Long-range interaction:
- Ewald summation
- Particle mesh Ewald summation
Potential modle:
Creation of potential/ gradient function an its optimisation
- Coulomb-Potential (-> Particle-Mesh)
- Lennard-Jones
- Particle-mesh Ewald
- Gradient descent
Integrator modle:
numerical method to solve Newton's EOM→ to calculate trajectories of particles
- Euler
- Langevin (Thermostat)
- Velocity-verlet/Rungakutta methods (simplectic/ dynamics)
Vectorized particle distances
Potential functions

Workflow

So, 13.01.19

- finale aufgaben Verteilung
- modules diskutieren
- Labeling festlegen (formate, Bezeichnungen, Abkürzungen, Parameter)
- Input und Output formate und Benennungen definieren
- Ziele festlegen für Fortschritt bis zum nächsten Treffen
- Parallelization strategies ? Domain decomposition method
- (Distribution of system data for parallel computing)
-



Selbstständige Arbeitsphase

- Ziele umsetzen für Fortschritt bis zum nächsten Treffen

So, 20.01.19?

- Fortschritte präsentieren Fehler Probleme diskutieren
-

Overview: Aufgabe für (Person): Phillip (P), Felix (F), Malte (M), Joana (J)

MODUL	INPUT	TASK	OUTPUT	TESTING	VISUALISATION
DATA STORAGE		? Configuration ?			
SAMPLING 13.01.19 (P, F, M, J)	N = number of particle - generate configuration - -	- configuration ? - Dann in: ? Markov Chain Monte Carlo Metropolis Monte Carlo	2D: Np.array([x_position 1:N],[y_position 1:N]) 3D: Np.array([x_position 1:N],[y_position 1:N], [z_position 1:N])	-	- distribution of particle plotten -
DISTANCE	2D: Np.array([x_position 1:N],[y_position 1:N]) 3D: Np.array([x_position 1:N],[y_position 1:N], [z_position 1:N])		2D: Np.array([x_i 1:N], [dis_x_i+1 1:N],…, [dis_x_N 1:N]); wie y, z 3D:…		
POTENTIAL				- energy consistency - Use simple potential with work and compare	
INTEGRATOR					

Information about modules

DATA STORAGE MODULE
????

SAMPLING MODULE
Initial configuration (markow chain, Particles equispaced in a box; 2D/ 3D)

MONTE CARLO

INPUT	OUTPUT	
initial configuration r(0), potential, size, beta, step	Position r, potential(r)	

(1) Start with initial configuration $\mathbf{r}^{(0)}$ and set k = 0.

(2) Fork=1,....,K:

(a) Sample random vector $\eta \in \mathbb{R}^N$ with $\eta_i \sim N(0,\sigma^2)$, i.e. an isotropic multivariate Gaussian distribution/ or function with variances σ^2.

(b) Propose new configuration $\mathbf{r}' = \mathbf{r}^{(k)} + \eta$.

(c) Accept new configuration with probability $p_{\text{acc}} = \min\left\{1, \exp\left(\frac{\phi(\mathrm{r}') - \phi(\mathrm{r}^k)}{k_B T}\right)\right\}$

If accepted, set $\mathbf{r}^{(k+1)} = \mathbf{r}'$, otherwise $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)}$.

Markov Chain Monte Carlo
- create samples from a possibly multi-dimensional continuous random variable, with probability density proportional to a known function.
- these samples can be used to evaluate an integral over that variable, as its expected value or variance.
- in Markov chain Monte Carlo methods are autocorrelated.
- a equilibrium distribution which is proportional to the function given.

Metropolis Monte Carlo
- Konfigurationen gemäß ihrer Boltzmann-Wahrscheinlichkeit + einfachen arithmetischen Durchschnitt
- zuerst eine zufällige Bewegung ausgeführt
- dann die Boltzmann-Wahrscheinlichkeit einer solchen Bewegung ausgewertet wird und die Wahrscheinlichkeit mit einer Zufallszahl verglichen wird.
- Ist die Boltzmann-Wahrscheinlichkeit der Bewegung größer als die Zufallszahl, wird die Bewegung akzeptiert. Andernfalls wird das System auf seine ursprüngliche Konfiguration zurückgesetzt.

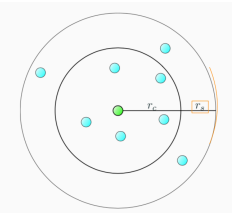
Vor- und Nachteile: ?

- not interested in the statistics of momenta anymore, so we integrate them out
- sampling the probability density.
- simple version of Monte Carlo is Metropolis Monte Carlo

DISTANCE MODULE

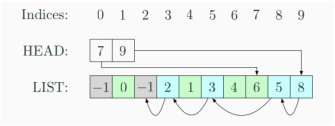
INPUT	OUTPUT	
Position r, potential(r)	Matrix distances x1 x2 … xN	

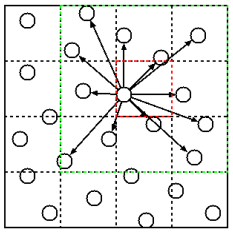
- calculation of distances between particles in MD Energy and force Calculation
- Introduce a cutoff and considering only the nearest neighbours within the radius; for LJ.Potential rc= 2.5 σ
  - Reduce effort: loop over all pairs of particle (i,j) = (j,i)
  -



Verlet list
- For each particle i, a Verlet list hat lists all other particles j within a smaller distance than rc+ rs
- Update the neighbour list every Ns tilmestep
- $r_s > N_s v_{\text{typ}} \delta t$ type speed of particles
- Choice between hash table and array

Cell-linked list : get complexity down
- domain into cells with an edge length greater than or equal to the cut-off radius of the interaction to be computed
- particles sorted into cells and the interactions are computed between particles in the same or neighbouring cells
- Needed array HEAD and array LIST
- Think about parallelisation





Cell decomposition approach
- partition the domain into cells and build a cell-linked list
- Verlet list by searching only in the neighbouring cells instead of considering all particles
- Dirty flag ?

POTENTIAL MODULE
description how the particles will interact in the simulation; two main approximations: Born–Oppenheimer approximation (dynamics of electrons), point particles that follow classical Newtonian dynamics (nuclei, which are much heavier than electrons)

Coulomb-Potential (-> Particle-Mesh)

$$\phi_i(\mathbf{r}) = \frac{1}{4\pi\varepsilon_0}\frac{q_i}{|\mathbf{r}-\mathbf{r_i}|}$$

$$\frac{1}{4\pi\varepsilon_0} = c_0^2\mu_0 = {}^{c^2{}_0\mu_0} = c^2{}_0 \times 10^{-7}\,\mathrm{H\cdot m^{-1}} = 8.987\times 10^9\,\mathrm{Nm^2C^{-2}}$$

Lennard-Jones

Particle-mesh Ewald

INTEGRATOR MODULE

Präsentation