Computational Physics – Lecture 10: Molecular dynamics III

Kristel Michielsen
Institute for Advanced Simulation
Jülich Supercomputing Centre
Research Centre Jülich
k.michielsen@fz-juelich.de
http://www.fz-juelich.de/ias/jsc/qip

Contents

- Molecular dynamics simulations
  - Time step
  - Equilibration
  - Measurements
  - Pseudocode
  - Thermostats
- Molecular dynamics ←→ Monte Carlo
- Exercises

Some pictures are taken from the internet for illustrational and educational purposes

Molecular dynamics simulation

Ingredients

- Boundary conditions
- Forces
- Initial conditions
- Integration algorithm
- **Time step**
- Equilibration
- Measurements

Molecular dynamics

Time step

- Choice is crucial
  - Too short → phase space is sampled inefficiently
  - Too long → energy will fluctuate (wildly)
- The maximum time step is defined by the fastest motion in the system
  - In standard MD with any real material at ordinary temperature $\Delta t = O(\mathrm{fs})$
    10^6-10^8 time steps can be simulated (depends in general on the computational cost of the calculation of the forces) → simulate processes that occur within 1 – 100 ns

Molecular dynamics

Time step

- Optimization and parallelization of software
- Special purpose hardware

→ Processes that occur within 1 μs - 1 ms can be simulated

Molecular dynamics simulation

Ingredients

- Boundary conditions
- Forces
- Initial conditions
- Integration algorithm
- Time step
- Equilibration
- Measurements

Molecular dynamics
Equilibration

- Since the system does not start from an equilibrium state, a certain number of time steps are to be taken until the system has reached an equilibrium → short MD simulation by itself

Molecular dynamics simulation

Ingredients

- Boundary conditions
- Forces
- Initial conditions
- Integration algorithm
- Time step
- Equilibration
- Measurements

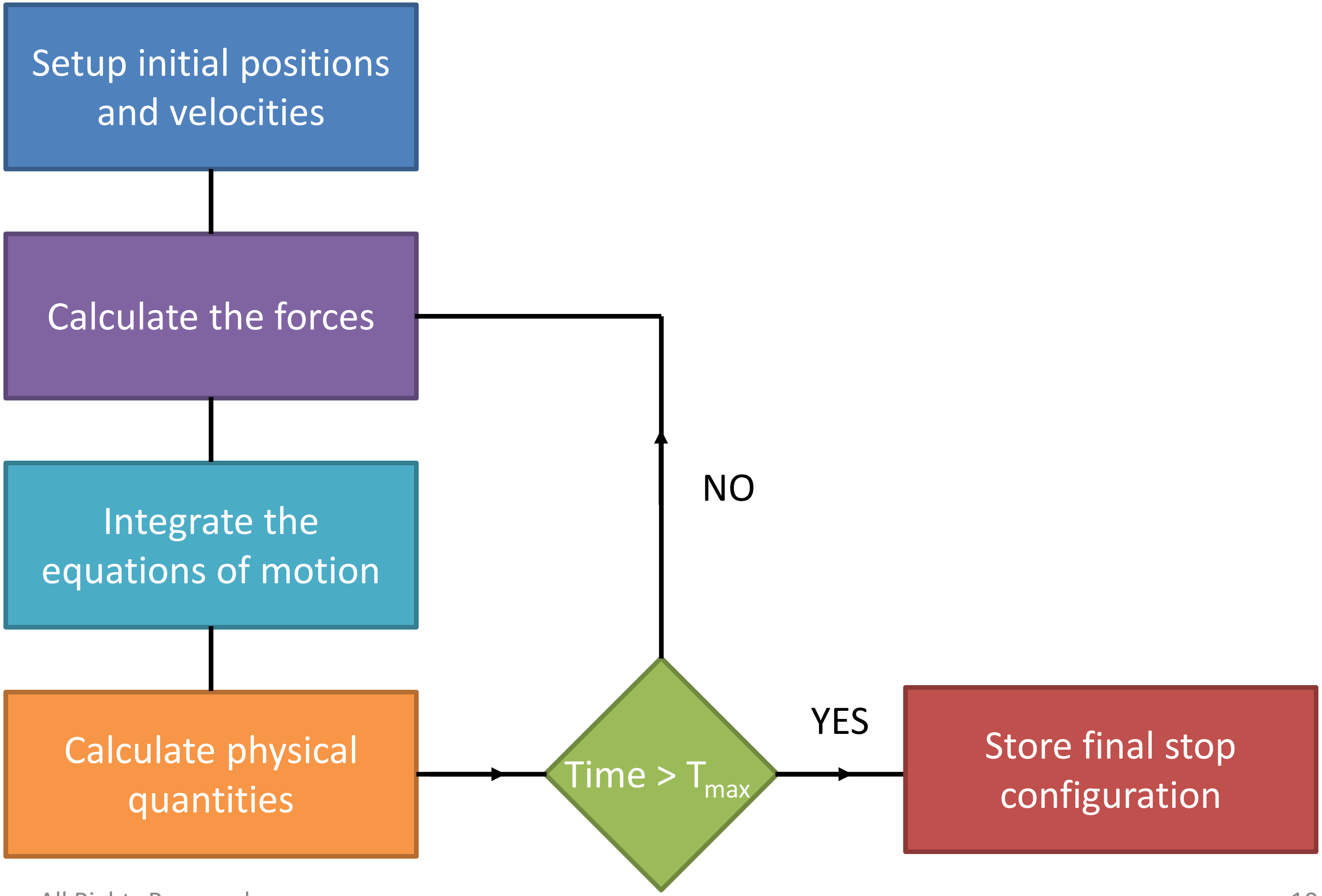
Molecular dynamics

Measurements

- Construct estimators for physical quantities of interest

Molecular dynamics

Scheme


Setup initial positions and velocities
Calculate the forces
Integrate the equations of motion
Calculate physical quantities
Time > T_{max}
NO
YES
Store final stop configuration

Molecular dynamics

Pseudocode: Program

Understanding Molecular Simulation, D. Frenkel and B. Smit, Academic Press (2001)

```
program md                          simple MD program
call init                           initialization
t=0
do while (t.lt.tmax)                MD loop
   call force(f,en)                 determine the forces
   call integrate(f,en)             integrate equations of motion
   t=t+delt
   call sample                      sample averages
enddo
stop
end
```

subroutines

Molecular dynamics

Pseudocode: Initialization

Understanding Molecular Simulation, D. Frenkel
and B. Smit, Academic Press (2001)

```
subroutine init                      initialization of MD program
sumv=0
sumv2=0
do i=1,npart
  x(i)=lattice pos(i)                place the particles on a lattice
  v(i)=(ranf()-0.5)                  assign random velocities in [-0.5,0.5]
  sumv=sumv+v(i)                     velocity center of mass
  sumv2=sumv2+v(i)**2                kinetic energy (mass = 1)
enddo
sumv=sumv/npart                      velocity center of mass
sumv2=sumv2/npart                    mean squared velocity
fs=sqrt(3*temp/sumv2)                scale factor of the velocities (Ekin = 3NT/2)
do i=1,npart                         set desired kinetic energy
  v(i)=(v(i)-sumv)*fs                set velocity center of mass to zero
  xm(i)=x(i)-v(i)*dt                 position previous time step
enddo
return
end
```

Molecular dynamics

Pseudocode: Calculation of the forces

Understanding Molecular Simulation, D. Frenkel and B. Smit, Academic Press (2001)

```
subroutine force(f,en)                 determine the force and energy
en=0                                   set energy to zero
do i=1,npart
  f(i)=0                               set forces to zero
enddo
do i=1,npart-1
  do j=i+1,npart                       loop over all pairs
    xr=x(i)-x(j)
    xr=xr-box*nint(xr/box)             periodic boundary conditions
    r2=xr**2
    if (r2.lt.rc2) then                test cut-off
      r2i=1/r2
      r6i=r2i**3
      ff=48*r2i*r6i*(r6i-0.5)          Lennard-Jones potential
      f(i)=f(i)+ff*xr                  update force
      f(j)=f(j)-ff*xr
      en=en+4*r6i*(r6i-1)-ecut         update potential energy
    endif
  enddo
enddo
return
end
```

Molecular dynamics

Pseudocode: Integrate eqns. of motion

Understanding Molecular Simulation, D. Frenkel
and B. Smit, Academic Press (2001)

```
subroutine integrate(f,en)               integrate equations of motion
sumv=0
sumv2=0
do i=1,npart                             MD loop
   xx=2*x(i)-xm(i)+delt**2*f(i)          Verlet algorithm $\mathbf{r}(t+\Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t-\Delta t) + \mathbf{a}(t)(\Delta t)^2$
   vi=(xx-xm(i))/(2*delt)                velocity $\mathbf{v}(t) = \frac{\mathbf{r}(t+\Delta t) - \mathbf{r}(t-\Delta t)}{2\Delta t}$
   sumv=sumv+vi                          velocity center of mass
   sumv2=sumv2+vi**2                     total kinetic energy
   xm(i)=x(i)                            update positions previous time
   x(i)=xx                               update positions current time
enddo
temp=sumv2/(3*npart)                     instantaneous temperature
etot=(en+sumv2)/(2*npart)                total energy (potential + kinetic) per particle
return
end
```

Molecular dynamics

Temperature

- MD naturally samples from the microcanonical ensemble (N, V, E)
  - Calculate forces on all particles from the derivative of the force field
  - Integrate the equations of motion with some time step
  - Recalculate the forces and repeat the process to generate a dynamical trajectory in the (N, V, E) ensemble
  - The mean kinetic energy is constant → average kinetic temperature T_K is constant

Molecular dynamics
Temperature

- Sampling from canonical ensembles (N,V,T) requires particles to interact with a thermostat
  - In thermal equilibrium T_K fluctuates. Sampling from the (N,V,T) ensemble requires keeping the statistical temperature T_S constant

Molecular dynamics

Temperature

– Velocity scaling

$$\mathbf{v}_n^{new} = \sqrt{T_S / T_K(t)}\mathbf{v}_n^{old} \text{ with } 3NT_K(t) = \sum_{n=1}^{N} m_n \mathbf{v}_n^2$$

- Straightforward to implement
- Results do not correspond to any known ensemble

– Berendsen method: Velocities are rescaled after each step by a factor

$$\lambda = \left[1 + \frac{\Delta t}{\tau}\left(\frac{T_S}{T_K(t)} - 1\right)\right]^{1/2}, \tau :$$ Parameter to adjust the strength of the coupling to the heat bath

- Straightforward to implement and robust
- Results do not correspond to any known ensemble

Berendsen et al, J. Chem. Phys. 81, 3684 (1984)

Molecular dynamics

Temperature

– Nosé-Hoover thermostat: Modify Newton equations

$$\frac{d\mathbf{r}_n}{dt} = \mathbf{p}_n / m_n, \frac{d\mathbf{p}_n}{dt} = \mathbf{F}_n - \varsigma \mathbf{p}_n$$

$$\frac{d\varsigma}{dt} = \frac{1}{\tau_T^2}\{T_K(t)/T_S - 1\}$$

where

τ_T : thermostat relaxation time (to be adjusted)

T_S : parameter

- Gives canonical properties

Nosé, J. Chem. Phys. 81, 511 (1984)
Nosé, Mol. Phys. 52, 255 (1984)
Hoover, Phys. Rev. A 31, 1696 (1985)

MD ←→ MC

MD

- Microcanonical ensemble (NVE) → constant temperature simulations require special thermostat techniques
- Only input is the description of the interatomic interaction
- Provides detailed molecular/atomic level information
- Success depends on the availability of good potential functions
- Classical description

MC

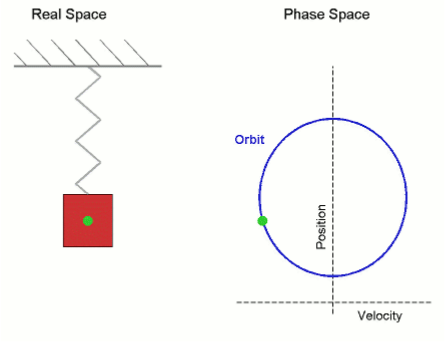
- Canonical ensemble (NVT) → simple to perform constant temperature simulations
- Speed up of equilibration process due to the flexibility in choosing random moves
- No discrete time approximations (and accompanying inaccuracies)
- Trajectories are based on energies only →fictitious dynamics

Molecular dynamics
Exercise 1

- Equation of motion $m_n \mathbf{a}_n = \mathbf{F}_n$
  - Mass of particle n : m_n
  - Acceleration of particle n : $\mathbf{a}_n = \dfrac{d^2\mathbf{r}_n}{dt^2}$
  - Force acting on particle n : $\mathbf{F}_n$
- Harmonic oscillator

$$m\frac{d^2x}{dt^2} = m\ddot{x} = -kx$$

http://en.wikipedia.org/wiki/Simple_harmonic_motion



Molecular dynamics

Exercise 1

- Implement the Euler algorithm for the harmonic oscillator ($V'(x) = kx$)
- Choice of units: $m = 1 \quad , \quad k = 1$
- Initial position and velocity: $x(0) = 0$ and $v(0) = 1$
- Solve using $\Delta t = 0.1, 0.01, 0.001$ for $j = 1, \ldots, [10000/\Delta t]$
- Plot $x(j\Delta t)$ and compare with $\sin(j\Delta t)$
- Argue whether this algorithm is useful or not
  - It is not!

Molecular dynamics

Exercise 2

- Implement the two variants (a) and (b) of the Euler-Cromer algorithm for the harmonic oscillator ($V'(x)=x$) and repeat the calculations of exercise 1
- Implementation:

$$(a)=\begin{cases} p((j+1)\Delta t)=p(j\Delta t)-\Delta t V'(x(j\Delta t)) \\ x((j+1)\Delta t)=x(j\Delta t)+\Delta t p((j+1)\Delta t) \end{cases}$$

$$(b)=\begin{cases} x((j+1)\Delta t)=x(j\Delta t)+\Delta t p(j\Delta t) \\ p((j+1)\Delta t)=p(j\Delta t)-\Delta t V'(x((j+1)\Delta t)) \end{cases}$$

Argue whether these algorithms are useful or not

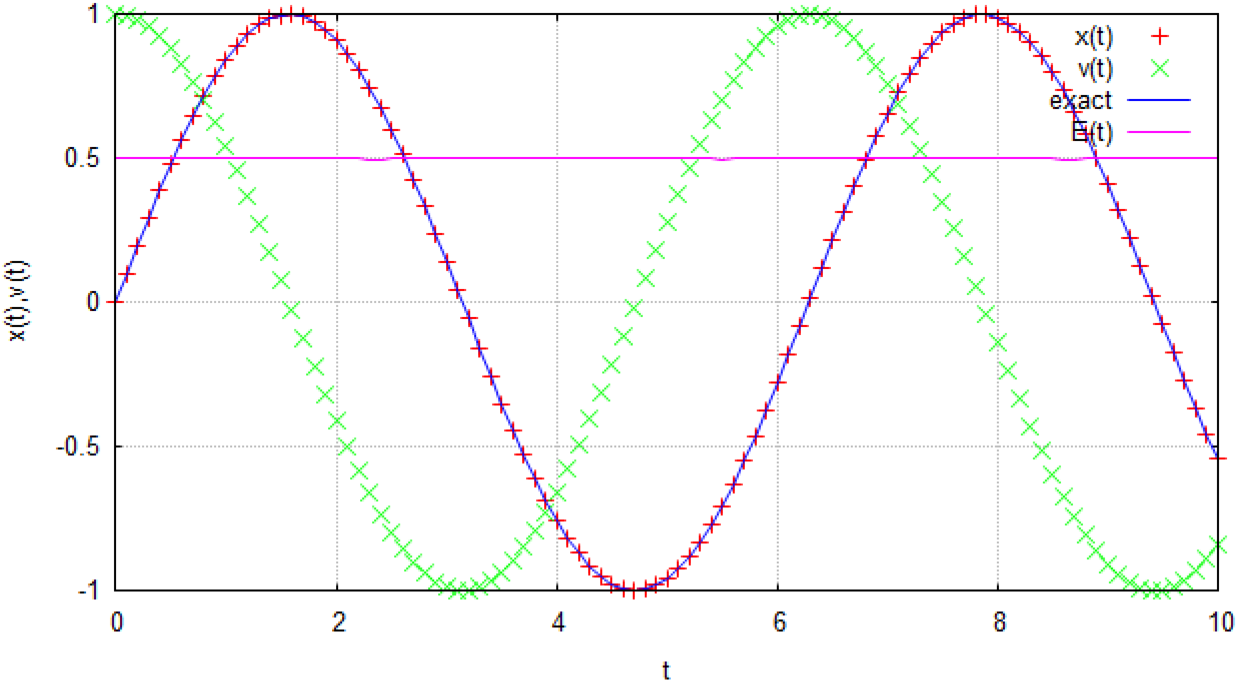
Molecular dynamics

Exercise 2

- Plot $x(t)$, $v(t)$, and the energy $E(t) = v^2(t)/2 + V(x(t))$ and also plot the analytical solution $x(t) = \sin t$ and $E = 1/2$ for comparison. Explain why the energy $E(t)$ is not exactly constant, as it should be according to classical mechanics
  - Use e.g $\Delta t = 0.01$ for $j = 1,\ldots,1000$ for plotting purposes

Molecular dynamics
Exercise 2

- Example of such a plot



Molecular dynamics

Exercise 3

- Use the velocity Verlet algorithm and repeat the simulations of exercise 2
- Use $\Delta t = 0.1, \quad 0.01$
- Plot $x(t), v(t)$ and the energy $E(t) = v^2(t)/2 + V(x(t))$ and also plot the analytical solution $x(t) = \sin t$ and $E = 1/2$ for comparison.
- Discuss the differences with the results of the Verlet and Euler-Cromer algorithms

Molecular dynamics

Exercise 4

- Use the velocity Verlet algorithm to solve the equation of motion of many coupled oscillators:

$$H = K + V = \frac{1}{2}\sum_{n=1}^{N} v_n^2 + \frac{1}{2}\sum_{n=1}^{N-1}(x_n - x_{n+1})^2$$

$$\frac{\partial V}{\partial x_k} = 2x_k - x_{k-1} - x_{k+1} \quad , \quad 1 < k < N$$

$$\frac{\partial V}{\partial x_1} = x_1 - x_2 \quad , \quad \frac{\partial V}{\partial x_N} = x_N - x_{N-1}$$

Molecular dynamics
Exercise 4

- For $N = 4, 16, 128$ and $\Delta \mathrm{t} = 0.1, \;\; 0.01$
- Initial configurations

1. $v_1(0), \ldots, v_N(0) = 0$

   $x_1(0), \ldots, x_N(0) = 0$ except $x_{N/2}(0) = 1$

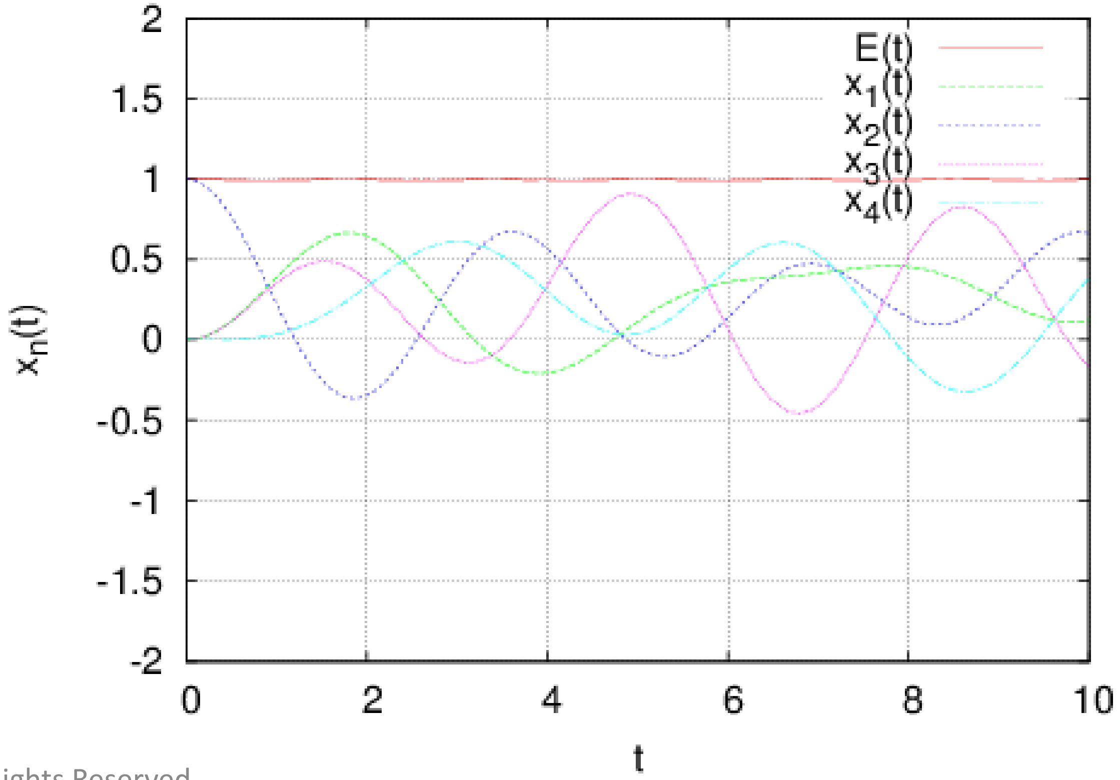
2. $v_1(0), \ldots, v_N(0) = 0$

   $x_k(0) = \sin \dfrac{\pi jk}{N+1}$ for $k = 1, \ldots, N$ and $j = 1, N/2$

- Plot the results for several $x_k(t)$ and interpret!

Molecular dynamics
Exercise 4

- Example for $N = 4, \Delta t = 0.1$ and initial configuration (1)



Report

Ms. Vrinda Mehta
v.mehta@fz-juelich.de

Dr. Madita Willsch
m.willsch@fz-juelich.de

Dr. Fengping Jin
f.jin@fz-juelich.de

Dr. Dima Nabok
d.nabok@fz-juelich.de

- Filename: Report_5_Surname1_Surname2.pdf, where Surname1 < Surname2 (alphabetical order).
  Example: Report_5_Jin_Willsch.pdf (Do not use “umlauts” or any other special characters in the names)
- Content of the report:
  - Names + matricle numbers + e-mail addresses + title
  - Introduction: describe briefly the problem you are modeling and simulating (write in complete sentences)
  - Simulation model and method: describe briefly the model and simulation method (write in complete sentences)
  - Simulation results: show figures (use grids, with figure captions !) depicting the simulation results. Give a brief description of the results (write in complete sentences)
  - Discussion: summarize your findings
  - Appendix: Include the listing of the program

Due date: 10 AM, June 5, 2023