165.144 Simulation of condensed matter

2025 Dr. Esther Heid esther.heid@tuwien.ac.at

Summary chapter 1

Classical mechanics: System of point particles with conservative force field:

$\mathbf{x}(t) = (x_1, y_1, z_1, \cdots, x_N, y_N, z_N)$, $\mathbf{v}(t) = \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t}$, $\mathbf{a}(t) = \frac{\mathrm{d}^2\mathbf{x}(t)}{\mathrm{d}t^2}$, $\mathbf{f}_i(t) = m_i\mathbf{a}_i(t)$

Summary chapter 1

Classical mechanics: System of point particles with conservative force field:

$\mathbf{x}(t) = (x_1, y_1, z_1, \cdots, x_N, y_N, z_N)$, $\mathbf{v}(t) = \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t}$, $\mathbf{a}(t) = \frac{\mathrm{d}^2\mathbf{x}(t)}{\mathrm{d}t^2}$, $\mathbf{f}_i(t) = m_i \mathbf{a}_i(t)$

- $E = E_k + E_{\mathrm{pot}}$
- $\Delta_{A\rightarrow B} E_k = W_{A\rightarrow B}[C] = -\Delta_{A\rightarrow B} E_{\mathrm{pot}}$
- $\mathbf{f} = -\nabla E_{\mathrm{pot}}$

Summary chapter 1

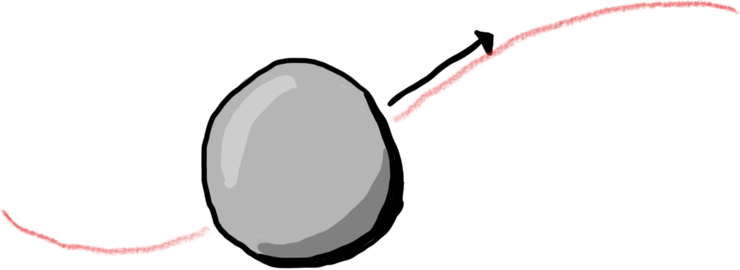
Classical mechanics: System of point particles with conservative force field:

$\mathbf{x}(t) = (x_1, y_1, z_1, \cdots, x_N, y_N, z_N)$, $\mathbf{v}(t) = \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t}$, $\mathbf{a}(t) = \frac{\mathrm{d}^2\mathbf{x}(t)}{\mathrm{d}t^2}$, $\mathbf{f}_i(t) = m_i \mathbf{a}_i(t)$

- $E = E_k + E_{\mathrm{pot}}$
- $\Delta_{A\to B} E_k = W_{A\to B}[C] = -\Delta_{A\to B} E_{\mathrm{pot}}$
- $\mathbf{f} = -\nabla E_{\mathrm{pot}}$

Symmetries vs conservation laws:

- **Translational symmetry:** Conservation of linear momentum $\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} = 0$
- **Rotational symmetry:** Conservation of angular momentum $\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}t} = 0$
- **Time translational symmetry:** Conservation of mechanical energy $\frac{\mathrm{d}E}{\mathrm{d}t} = 0$



Chapter 2: Numerical integration over time

Molecular dynamics simulation

Recap from Chapter 1: For an MD simulation, we need to know:

1. How atoms interact, i.e. given coordinates $\overrightarrow{r}$, obtain forces $\overrightarrow{f}(\overrightarrow{r})$
2. How atoms move, i.e. given forces $\overrightarrow{f}$, obtain coordinates at next time step $\overrightarrow{r}(t + \Delta t)$

Molecular dynamics simulation

Recap from Chapter 1: For an MD simulation, we need to know:

1. How atoms interact, i.e. given coordinates $\vec{r}$, obtain forces $\vec{f}(\vec{r})$
2. How atoms move, i.e. given forces $\vec{f}$, obtain coordinates at next time step $\vec{r}(t + \Delta t)$

Newton's classical mechanics:

1. $\vec{f} = -\nabla E_{\mathrm{pot}}$ and $\vec{f_i} = -\frac{\partial E_{\mathrm{pot}}}{\partial \vec{x_i}}$
2. Numeric integration of Newton's equations of motions, with $\vec{a} = \frac{\vec{f}}{m}$

Molecular dynamics simulation

Recap from Chapter 1: For an MD simulation, we need to know:

1. How atoms interact, i.e. given coordinates $\overrightarrow{r}$, obtain forces $\overrightarrow{f}(\overrightarrow{r})$
2. How atoms move, i.e. given forces $\overrightarrow{f}$, obtain coordinates at next time step $\overrightarrow{r}(t + \Delta t)$

Newton's classical mechanics:

1. $\overrightarrow{f} = -\nabla E_{\mathrm{pot}}$ and $\overrightarrow{f_i} = -\frac{\partial E_{\mathrm{pot}}}{\partial \overrightarrow{x_i}}$
2. Numeric integration of Newton's equations of motions, with $\overrightarrow{a} = \frac{\overrightarrow{f}}{m}$

Chapter 2: Given a simple E_{pot}, create a trajectory via numeric integration

A simple potential energy

- Spherical particles (atoms) that interact with each other
- Interactions between pairs of atoms based on two terms
- First term: Atoms repel each other at close range (=resistance to compression)
- Second term: Atoms attract each other over a range of separations (=e.g. in solids or liquids)

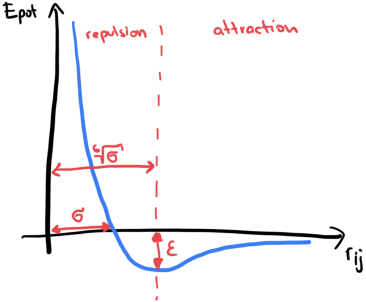
Different potentials available, best known: **Lennard-Jones (LJ) potential**

$$E_{\mathrm{pot}}(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right]$$

The Lennard-Jones potential

$$E_{\mathrm{pot}}(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right]$$

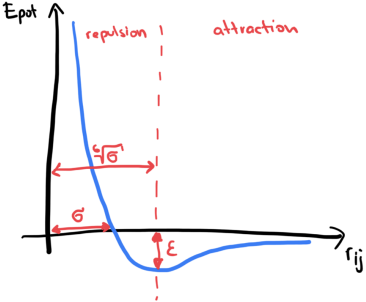
- pair of atoms i and j located at $\vec{r_i}$ and $\vec{r_j}$
- $\vec{r}_{ij} = \vec{r_j} - \vec{r_i}$ and $r_{ij} = |\vec{r}_{ij}|$
- ϵ governs the strength of the interaction
- σ defines the length scale, minimum at $2^{1/6}\sigma$



The Lennard-Jones potential

$$E_{\text{pot}}(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]$$

- pair of atoms i and j located at $\vec{r_i}$ and $\vec{r_j}$
- $\vec{r}_{ij} = \vec{r_j} - \vec{r_i}$ and $r_{ij} = |\vec{r}_{ij}|$
- ϵ governs the strength of the interaction
- σ defines the length scale, minimum at $2^{1/6}\sigma$



Force that atom j exerts on atom i:

$$\vec{f}_{ij} = 4\epsilon\left[-12\sigma^{12}r_{ij}^{-13} - -6\sigma^{6}r_{ij}^{-7}\right]\frac{\vec{r}_{ji}}{r_{ij}} = \frac{48\epsilon}{\sigma^2}\left[\left(\frac{\sigma}{r_{ij}}\right)^{14} - 0.5\left(\frac{\sigma}{r_{ij}}\right)^{8}\right]\vec{r}_{ij}$$

The Lennard-Jones potential

The force on atom i with mass m_i (in this chapter all $m_i = m$, i.e. all particles are similar) is then

$m_i \frac{d^2 \overrightarrow{r}_i}{dt^2} = \overrightarrow{f}_i = \sum_{j=1}^{N} \overrightarrow{f}_{ij}$ with $j \neq i$ and N atoms

In general: $\overrightarrow{f}_{ij} = -\overrightarrow{f}_{ji}$

The Lennard-Jones potential

The force on atom i with mass m_i (in this chapter all $m_i = m$, i.e. all particles are similar) is then

$m_i \frac{d^2 \overrightarrow{r}_i}{dt^2} = \overrightarrow{f}_i = \sum_{j=1}^{N} \overrightarrow{f}_{ij}$ with $j \neq i$ and N atoms

In general: $\overrightarrow{f}_{ij} = -\overrightarrow{f}_{ji}$

Dimensionless units: Choose σ, m, ϵ and ϵ / k_B as units of length, mass, energy, and temperature respectively. We then get:

- Equation of motion $\frac{d^2 \overrightarrow{r}_i}{dt^2} = 48 \sum_{j \neq i} \left(r_{ij}^{-14} - 0.5 r_{ij}^{-8} \right) \overrightarrow{r}_{ij}$
- Kinetic energy: $E_K = \frac{1}{2} \sum_{i=1}^{N} v_i^2$
- Potential energy: $E_{\mathrm{pot}} = 4 \sum_{1 \leq i < j \leq N} \left[r_{ij}^{-12} - r_{ij}^{-6} \right]$

Integration

- MD trajectories are obtained using finite difference techniques, i.e. numerical integration in many short time steps Δt

Integration

- MD trajectories are obtained using finite difference techniques, i.e. numerical integration in many short time steps Δt
- Compute force on each particle at time t as the vector sum of interactions with other particles

Integration

- MD trajectories are obtained using finite difference techniques, i.e. numerical integration in many short time steps Δt
- Compute force on each particle at time t as the vector sum of interactions with other particles
- Obtain the acceleration from the forces

Integration

- MD trajectories are obtained using finite difference techniques, i.e. numerical integration in many short time steps Δt
- Compute force on each particle at time t as the vector sum of interactions with other particles
- Obtain the acceleration from the forces
- Combine acceleration, positions and velocities at time t to get positions and velocities at $t + \Delta t$

Integration

- MD trajectories are obtained using finite difference techniques, i.e. numerical integration in many short time steps Δt
- Compute force on each particle at time t as the vector sum of interactions with other particles
- Obtain the acceleration from the forces
- Combine acceleration, positions and velocities at time t to get positions and velocities at $t + \Delta t$
- Several methods exist, all based on Taylor expansions:

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \Delta t \vec{v}(t) + \frac{1}{2}\Delta t^2 \vec{a}(t) + \frac{1}{6}\Delta t^3 \vec{j}(t) + \dots$$

$$\vec{v}(t + \Delta t) = \vec{v}(t) + \Delta t \vec{a}(t) + \frac{1}{2}\Delta t^2 \vec{j}(t) + \dots$$

$$\vec{a}(t + \Delta t) = \vec{a}(t) + \Delta t \vec{j}(t) + \dots$$

Euler algorithm

Simply truncate Taylor expansion

Euler algorithm

$$\vec{r}(t+\Delta t) = \vec{r}(t) + \Delta t \vec{v}(t) + \frac{1}{2}\Delta t^2 \vec{a}(t)$$

$$\vec{v}(t+\Delta t) = \vec{v}(t) + \Delta t \vec{a}(t)$$

Euler algorithm

Simply truncate Taylor expansion

Euler algorithm

$$\vec{r}(t+\Delta t) = \vec{r}(t) + \Delta t \vec{v}(t) + \frac{1}{2}\Delta t^2 \vec{a}(t)$$

$$\vec{v}(t+\Delta t) = \vec{v}(t) + \Delta t \, \vec{a}(t)$$

Drawbacks:

- not time-reversible
- not very accurate

Verlet algorithm

Use $\overrightarrow{r}(t+\Delta t) = \overrightarrow{r}(t) + \Delta t\overrightarrow{v}(t) + \frac{1}{2}\Delta t^2\overrightarrow{a}(t) + \dots$
and $\overrightarrow{r}(t-\Delta t) = \overrightarrow{r}(t) - \Delta t\overrightarrow{v}(t) + \frac{1}{2}\Delta t^2\overrightarrow{a}(t) - \dots$

Verlet algorithm

$$\overrightarrow{r}(t+\Delta t) = 2\overrightarrow{r}(t) - \overrightarrow{r}(t-\Delta t) + \Delta t^2\overrightarrow{a}(t)$$

Verlet algorithm

Use $\vec{r}(t+\Delta t) = \vec{r}(t) + \Delta t \vec{v}(t) + \frac{1}{2}\Delta t^2 \vec{a}(t) + \dots$
and $\vec{r}(t-\Delta t) = \vec{r}(t) - \Delta t \vec{v}(t) + \frac{1}{2}\Delta t^2 \vec{a}(t) - \dots$

Verlet algorithm

$$\vec{r}(t+\Delta t) = 2\vec{r}(t) - \vec{r}(t-\Delta t) + \Delta t^2 \vec{a}(t)$$

Advantages:

- straightforward implementation

Verlet algorithm

Use $\vec{r}(t+\Delta t) = \vec{r}(t) + \Delta t \vec{v}(t) + \frac{1}{2}\Delta t^2 \vec{a}(t) + \ldots$
and $\vec{r}(t-\Delta t) = \vec{r}(t) - \Delta t \vec{v}(t) + \frac{1}{2}\Delta t^2 \vec{a}(t) - \ldots$

Verlet algorithm

$$\vec{r}(t+\Delta t) = 2\vec{r}(t) - \vec{r}(t-\Delta t) + \Delta t^2 \vec{a}(t)$$

Advantages:

- straightforward implementation

Drawbacks:

- adding a small term ($\Delta t^2 \vec{a}(t)$) to the difference of two large terms ($2\vec{r}(t) - \vec{r}(t-\Delta t)$) causes a loss of numerical precision
- lack of explicit velocity term (but can be recovered, e.g. $\vec{v}(t) = \frac{\vec{r}(t+\Delta t) - \vec{r}(t-\Delta t)}{2\Delta t}$)
- not self-starting, i.e. we need to estimate $\vec{r}(t-\Delta t)$ when starting at $t=0$

Leapfrog algorithm

Positions:

Rewrite Taylor expansion $\vec{r}(t+\Delta t) = \vec{r}(t) + \Delta t \vec{v}(t) + \frac{1}{2}\Delta t^2 \vec{a}(t) + \frac{1}{6}\Delta t^3 \vec{j}(t) + \dots$

as $\vec{r}(t+\Delta t) = \vec{r}(t) + \Delta t \left[\vec{v}(t) + \frac{\Delta t}{2}\vec{a}(t)\right] + \dots$

where $\left[\vec{v}(t) + \frac{\Delta t}{2}\vec{a}(t)\right] = \vec{v}(t+\Delta t/2)$

Velocities:

Combine Taylor expansion at $\vec{v}(t+\Delta t/2)$ and $\vec{v}(t-\Delta t/2)$

Leapfrog algorithm

$$\vec{r}(t+\Delta t) = \vec{r}(t) + \Delta t \vec{v}(t+\Delta t/2)$$

$$\vec{v}(t+\Delta t/2) = \vec{v}(t-\Delta t/2) + \Delta t \vec{a}(t)$$

Leapfrog algorithm

Coordinates and velocities are evaluated at different times:

- We first evaluate $\overrightarrow{v}(t + \Delta t/2)$ from $\overrightarrow{v}(t - \Delta t/2)$ and $\overrightarrow{a}(t)$
- We then obtain the positions $\overrightarrow{r}(t + \Delta t)$ from $\overrightarrow{v}(t + \Delta t/2)$ and $\overrightarrow{r}(t)$

Leapfrog algorithm

Coordinates and velocities are evaluated at different times:

- We first evaluate $\vec{v}(t + \Delta t/2)$ from $\vec{v}(t - \Delta t/2)$ and $\vec{a}(t)$
- We then obtain the positions $\vec{r}(t + \Delta t)$ from $\vec{v}(t + \Delta t/2)$ and $\vec{r}(t)$

Advantages:

- explicitly includes velocity term
- no calculation of difference of large numbers
- exactly time-reversible
- preserves energy well
- straightforward implementation

Leapfrog algorithm

Coordinates and velocities are evaluated at different times:

- We first evaluate $\vec{v}(t + \Delta t/2)$ from $\vec{v}(t - \Delta t/2)$ and $\vec{a}(t)$
- We then obtain the positions $\vec{r}(t + \Delta t)$ from $\vec{v}(t + \Delta t/2)$ and $\vec{r}(t)$

Advantages:

- explicitly includes velocity term
- no calculation of difference of large numbers
- exactly time-reversible
- preserves energy well
- straightforward implementation

Drawbacks:

- Positions and velocities are not synchronized (we therefore cannot compute the kinetic energy contribution to the total energy at the same times as the positions are defined which determine the potential energy)
- not self-starting, i.e. we need to estimate $\vec{v}(t - \Delta t/2)$ when starting at $t = 0$

Velocity-Verlet

Velocity Verlet algorithm

$$\vec{r}(t+\Delta t) = \vec{r}(t) + \Delta t \vec{v}(t) + \tfrac{1}{2}\Delta t^2 \vec{a}(t)$$

$$\vec{v}(t+\Delta t) = \vec{v}(t) + \Delta t \tfrac{\vec{a}(t)+\vec{a}(t+\Delta t)}{2})$$

In practice: Three-stage procedure (there is also an alternative four-stage procedure involving half time-steps for the velocities)

- Compute $\vec{r}(t+\Delta t)$ from $\vec{r}(t)$, $\vec{v}(t)$ and $\vec{a}(t)$
- Compute $\vec{a}(t+\Delta t)$ using the interaction potential at $\vec{r}(t+\Delta t)$
- Compute $\vec{v}(t+\Delta t)$ from $\vec{v}(t)$, $\vec{a}(t)$ and $\vec{a}(t+\Delta t)$

Velocity-Verlet

Advantages:

- Synchronized positions and velocities
- Self-starting
- explicitly includes velocity term
- no calculation of difference of large numbers
- straightforward implementation

Velocity-Verlet

Advantages:

- Synchronized positions and velocities
- Self-starting
- explicitly includes velocity term
- no calculation of difference of large numbers
- straightforward implementation

Drawbacks:

- requires a bit more memory than Verlet or Leapfrog

Velocity-Verlet

Advantages:

- Synchronized positions and velocities
- Self-starting
- explicitly includes velocity term
- no calculation of difference of large numbers
- straightforward implementation

Drawbacks:

- requires a bit more memory than Verlet or Leapfrog

Other integrators

There are many more, e.g. Beeman's algorithm (more accurate expression for velocities but more expensive), or Predictor-Corrector Integration Methods (much more complex and expensive)

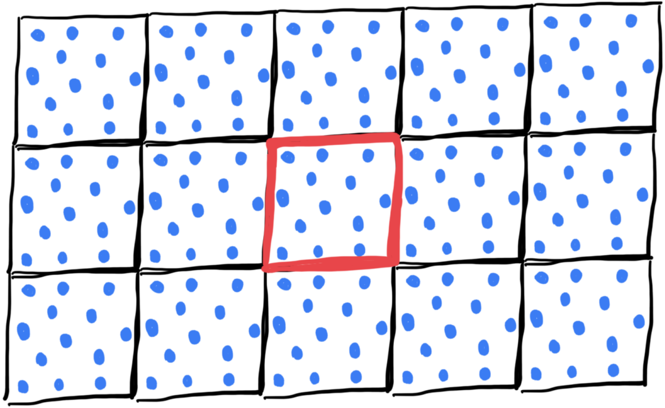
Let’s put this to the test!

jupyter

(Demonstration of different integrators for a toy system: Throwing a ball)

Periodic boundary conditions

To get a real system working with the Lennard Jones potential, (e.g. liquid argon) we need to think about the size of our simulation system.



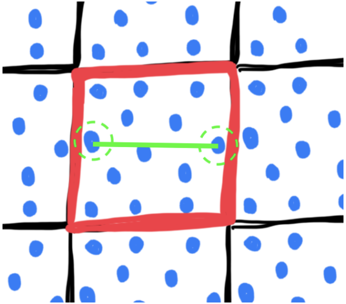
Periodic boundary conditions (cube of length L)

Coordinates (similar in y and z direction):

- if $r_{ix} \geq L/2$, replace it by $r_{ix} - L$
- if $r_{ix} \leq -L/2$, replace it by $r_{ix} + L$

Interactions (similar in y and z direction):

- if $r_{ijx} \geq L/2$, replace it by $r_{ijx} - L$
- if $r_{ijx} \leq -L/2$, replace it by $r_{ijx} + L$



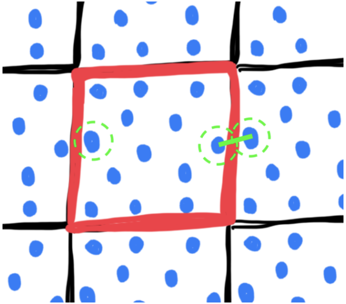
Periodic boundary conditions (cube of length L)

Coordinates (similar in y and z direction):

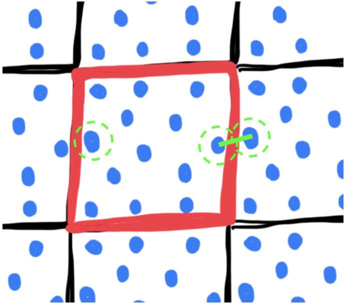
- if $r_{ix} \geq L/2$, replace it by $r_{ix} - L$
- if $r_{ix} \leq -L/2$, replace it by $r_{ix} + L$

Interactions (similar in y and z direction):

- if $r_{ijx} \geq L/2$, replace it by $r_{ijx} - L$
- if $r_{ijx} \leq -L/2$, replace it by $r_{ijx} + L$



Periodic boundary conditions (cube of length L)



Coordinates (similar in y and z direction):

- if $r_{ix} \geq L/2$, replace it by $r_{ix} - L$
- if $r_{ix} \leq -L/2$, replace it by $r_{ix} + L$

Interactions (similar in y and z direction):

- if $r_{ijx} \geq L/2$, replace it by $r_{ijx} - L$
- if $r_{ijx} \leq -L/2$, replace it by $r_{ijx} + L$

Important:

Limits interaction range to $L/2$

Long-range interactions need to be described differently

A simple initial state

- Results of a simulation of adequate duration are insensitive to the initial state

$$\vec{x}(t) = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \vdots & \\ x_N & y_N & z_N \end{pmatrix} \qquad \vec{v}(t) = \begin{pmatrix} v_1^x & v_1^y & v_1^z \\ & \vdots & \\ v_N^x & v_N^y & v_N^z \end{pmatrix} \qquad \vec{a}(t) = \begin{pmatrix} a_1^x & a_1^y & a_1^z \\ & \vdots & \\ a_N^x & a_N^y & a_N^z \end{pmatrix}$$

A simple initial state

- Results of a simulation of adequate duration are insensitive to the initial state
- Here we will start with atoms on a lattice

$$\vec{x}(t) = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \vdots & \\ x_N & y_N & z_N \end{pmatrix} \qquad \vec{v}(t) = \begin{pmatrix} v_1^x & v_1^y & v^z_1 \\ & \vdots & \\ v_N^x & v_N^y & v_N^z \end{pmatrix} \qquad \vec{a}(t) = \begin{pmatrix} a_1^x & a_1^y & a_1^z \\ & \vdots & \\ a_N^x & a_N^y & a_N^z \end{pmatrix}$$

A simple initial state

- Results of a simulation of adequate duration are insensitive to the initial state
- Here we will start with atoms on a lattice
- Initial velocities: random directions, with fixed magnitude based on temperature $T = \frac{1}{dN-d} \sum_i \vec{v_i}^2$ with d being the number of dimensions (2 or 3), and $dN - d$ being the total degrees of freedom ($-d$ because of conservation of momentum)

$$\vec{x}(t) = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \vdots & \\ x_N & y_N & z_N \end{pmatrix} \qquad \vec{v}(t) = \begin{pmatrix} v_1^x & v_1^y & v^{z_1} \\ & \vdots & \\ v_N^x & v_N^y & v_N^z \end{pmatrix} \qquad \vec{a}(t) = \begin{pmatrix} a_1^x & a_1^y & a_1^z \\ & \vdots & \\ a_N^x & a_N^y & a_N^z \end{pmatrix}$$

A simple initial state

- Results of a simulation of adequate duration are insensitive to the initial state
- Here we will start with atoms on a lattice
- Initial velocities: random directions, with fixed magnitude based on temperature $T = \frac{1}{dN-d} \sum_i \vec{v_i}^2$ with d being the number of dimensions (2 or 3), and $dN - d$ being the total degrees of freedom ($-d$ because of conservation of momentum)
- Adjust initial velocities to ensure that the center of mass of the system is at rest

$$\vec{x}(t) = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \vdots & \\ x_N & y_N & z_N \end{pmatrix} \qquad \vec{v}(t) = \begin{pmatrix} v_1^x & v_1^y & v^{z_1} \\ & \vdots & \\ v_N^x & v_N^y & v_N^z \end{pmatrix} \qquad \vec{a}(t) = \begin{pmatrix} a_1^x & a_1^y & a_1^z \\ & \vdots & \\ a_N^x & a_N^y & a_N^z \end{pmatrix}$$

A simple initial state

- Results of a simulation of adequate duration are insensitive to the initial state
- Here we will start with atoms on a lattice
- Initial velocities: random directions, with fixed magnitude based on temperature $T = \frac{1}{dN-d} \sum_i \vec{v_i}^2$ with d being the number of dimensions (2 or 3), and $dN - d$ being the total degrees of freedom ($-d$ because of conservation of momentum)
- Adjust initial velocities to ensure that the center of mass of the system is at rest
- Set initial accelerations to zero

$$\vec{x}(t) = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ & \vdots & \\ x_N & y_N & z_N \end{pmatrix} \qquad \vec{v}(t) = \begin{pmatrix} v_1^x & v_1^y & v_1^z \\ & \vdots & \\ v_N^x & v_N^y & v_N^z \end{pmatrix} \qquad \vec{a}(t) = \begin{pmatrix} a_1^x & a_1^y & a_1^z \\ & \vdots & \\ a_N^x & a_N^y & a_N^z \end{pmatrix}$$

Properties

- Total energy per atom: $\overbrace{\frac{1}{2N}\sum_i v_i^2}^{\text{kinetic energy per atom}} + \overbrace{\frac{4}{N}\sum_{i<j}\left[r_{ij}^{-12} - r_{ij}^{-6}\right]}^{\text{potential energy per atom}}$
- Temperature $T = \frac{1}{dN}\sum_i \overrightarrow{v_i}^2 = \frac{2E_k}{d}$ with the per-atom kinetic energy E_k

  ↳ d... dimensions (2 or 3)
- Pressure: derived from the virial expression $PV = NT + \frac{1}{d}\sum_{i=1}^{N} \overrightarrow{r_i} \cdot \overrightarrow{f_i}$, leading to $P = \frac{1}{dV}\left[\sum_i \overrightarrow{v_i}^2 + 48\sum_{i<j}(r_{ij}^{-12} - \frac{1}{2}r_{ij}^{-6})\right]$

  V ... Volume for three dimensions

  A ... Area for two dimensions

Let’s put this to the test!

(Demonstration of Soft-sphere potential (adapted Lennard-Jones potential))