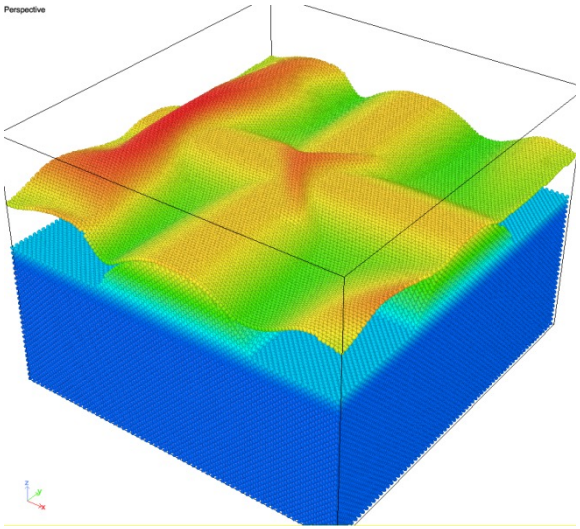
Molecular Dynamics (part 1)

Computational Material Science
Lecture 4



Last time

- We model solids by using interaction potentials, obtained by an educated guess on how the interaction between atoms should be.
- While pair potentials like Lennard-Jones work well for rare gases, we need progressively more fancy potentials for more complicated systems, like ionic solids, metals and covalent solids, for which directionality of the bonds are important.
- All potentials are approximations and contain parameters that are fit to experiments or smaller-scale simulations (more precise simulations)
- AN ADDITIONAL POINT: The potential does a good job when it is capable of capturing properties that were not in the *training set.*

Today

Periodicity

Basic concepts of molecular dynamics

Numerical integration of Newton's equation

How to check the soundness of an MD simulation

LEARNING GOALS:

Ability to describe the basic concepts of molecular dynamics

Capability of writing the subroutine that determines the initial atomic velocity of a molecular dynamics code

Distinguish between a clearly wrong MD simulation and a possibly correct one

Limitation of pair potentials

Pair potentials are also called central-force potentials

For linear elastic isotropic solids with symmetric lattice positions, the Cauchy relation should hold:

$$c_{12} = c_{44}$$

Deviations from the Cauchy relation are a measure of the deviation from central force interactions in the material.

experimental values:

Material	c_{12}/c_{44}
"LJ"	1.00
Ar	1.12
Mo	1.54
Cu	1.94
Au	4.71
NaCl	0.99
Si	0.77
MgO	0.53
diamond	0.16

Reminder: linear elasticity

Fundamental assumptions:

1) Stress is small (far from yield point)
2) Strains are small
3) Stress-strain relationship linear

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & \dots & C_{16} \\ C_{21} & C_{22} & \dots & C_{26} \\ C_{31} & C_{32} & \dots & C_{36} \\ C_{41} & C_{42} & \dots & C_{46} \\ C_{51} & C_{52} & \dots & C_{56} \\ C_{61} & C_{62} & \dots & C_{66} \end{pmatrix} \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_x \\ \gamma_y \\ \gamma_z \end{pmatrix} = \mathbf{C}\boldsymbol{\epsilon}$$

isotropic solids:

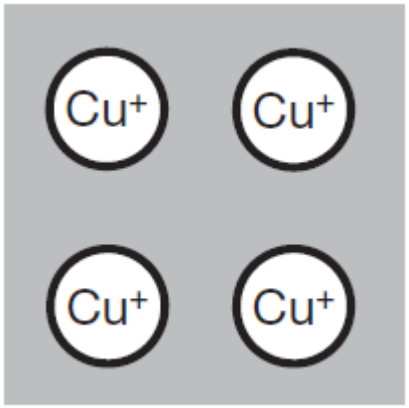
$$\mathbf{C} = \begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}$$

$$\mu = C_{44} = \frac{1}{2}(C_{11} - C_{12})$$

$$\lambda = C_{12}$$

$$\lambda + 2\mu = C_{11}$$

Metals



Metals are made of cores surrounded by delocalized electrons

Elastic properties, especially at high T, are poorly captured by pair-potentials, even more so when there are defects.

Pair potentials neglect the energy related to the electronic gas, overestimating repulsion between atoms.

$$U = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}{}'\phi(r_{ij}) + U_{eg}(V)$$

A possible fix is to take a volume dependent corrective term, electron density depends on volume

$$U_{eg}(V) = U_{known}(V) - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}{}'\phi(r_{ij})$$

Calculated for a perfect system, to be used for `defective' one

experimental

Embedded Atom Model potentials

Limitation: The volume-dependent term cannot account for asymmetries in the electronic gas (defects, grain boundaries), so it is best to add to the pair potential a functional of the local electron density which reflects the atomic positions.
EAM potentials are motivated by DFT, and have the form:

$$U = \sum_i F_i\left[\sum_{j\neq i} f_{ij}(r_{ij})\right] + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}{}'\phi_{ij}(r_{ij})$$

f represents an approximation of the electron density
F represents the energy to embed an atom *i* in a uniform electron gas of a local electron density:

$$\bar{\rho}_i = \sum_{j\neq i} \rho_j(r_{ij})$$

$$\rho(r) = r^6 e^{-\beta r}$$

electron density of a 4s hydrogen horbital

$$U_{EAM} = \sum_i F_i(\bar{\rho}_i) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}{}'\phi_{ij}(r_{ij})$$

The various EAM potentials have different choices for F, ρ and Φ.
The functions depend on parameters fit to experiments or simulations.

Periodicity

Mostly, modeling for material science is about describing systems of discrete objects.

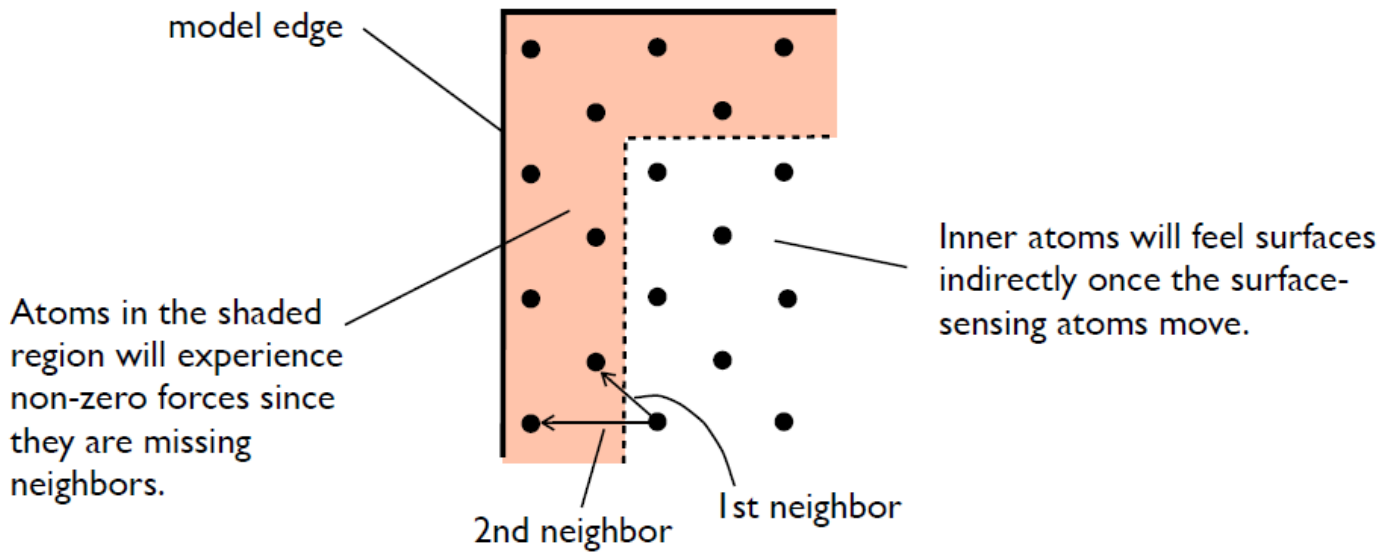
Objects: atoms, molecules, ions, vacancies, dislocations, slip systems, grains

Often, modeling a really large number of objects is too expensive, thus very large systems are modeled exploiting the concept of periodic boundary conditions

Limitation of finite systems

- Atomistic simulations are by necessity of finite systems. As a result, atoms near the edges of the model will experience surface effects and will relax.

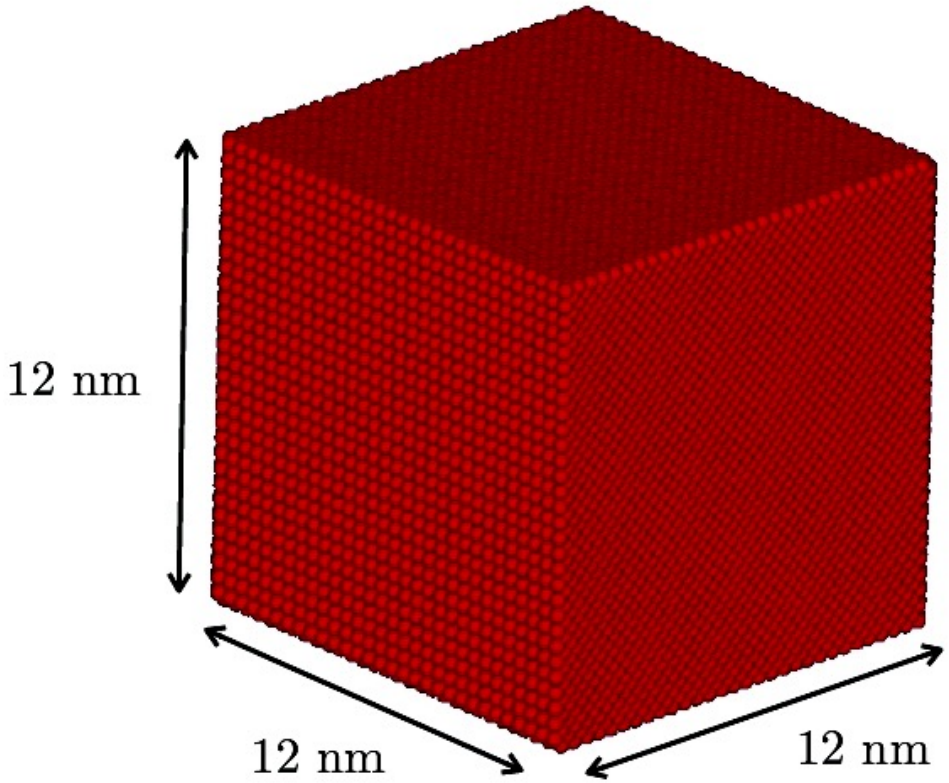
For example, consider a 2D lattice in which particles have second-neighbor interactions:



- If the model edge is a physical surface, then the relaxation is real. Otherwise this is a limitation of the simulation.

Finite crystals

- **Example:** surface effects in a finite nanosystem



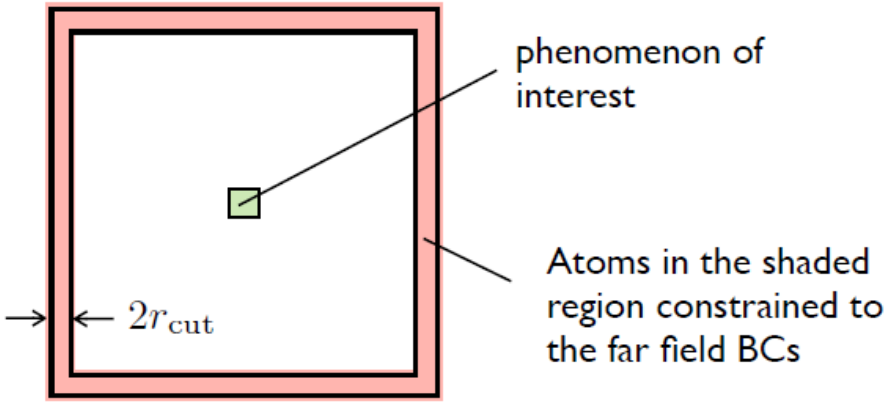
- 108,000 aluminum atoms in a perfect FCC arrangement.
- 25% of the atoms are within a cutoff radius of the surface and therefore sense it.
- Sometimes we are interested in modeling "infinite" crystals or phenomena deep inside a crystal.

Boundary conditions

- When the objective is to simulate a system without surfaces two possible options are:

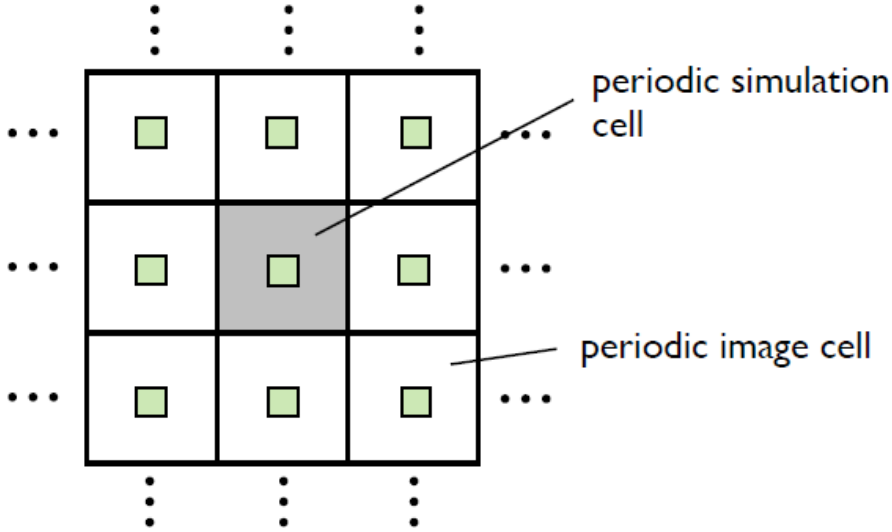
(1) Simulate a much larger system than needed and constrain the edge atoms.

- Verify solution by increasing (or decreasing) cell size and seeing that no significant change in observed phenomena occurs.



(2) Apply periodic boundary conditions (PBCs).

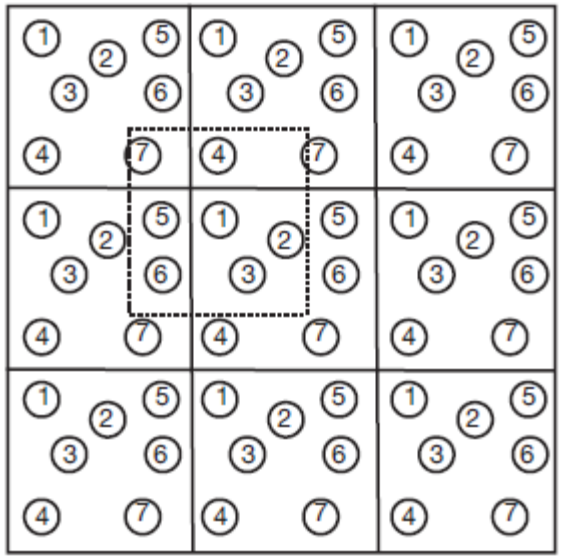
- Finite-sized simulation cell is replicated in space, creating an infinite system.
- Must verify that the periodic cell is large enough to preclude interaction between studied phenomena.



Periodic boundary conditions

The goal is to mimic a real system, much larger than what can be modeled in a reasonable amount of time on a computer.
The trick is to model a representative part of the system and apply boundary conditions to approximate the effect of the rest of the material.

A common approach, mostly employed in MD is to use periodic boundary conditions.



The simulation is performed on a unit cell and all replicas display the same behavior.

All particles i have the same properties.

When one particle moves out of the cell, its replica comes in. The number of particles in the simulation cell stay unchanged.

When summing the interactions between particles, replicas are considered.

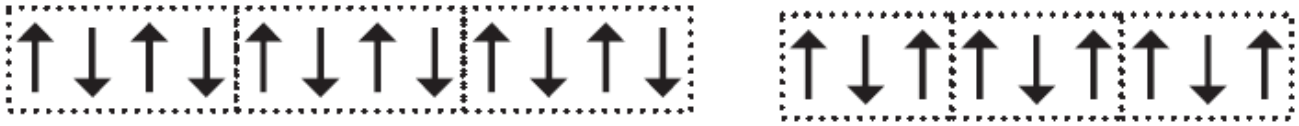
Periodic boundary conditions

Advantages: powerful to simulate bulk behavior, time efficient

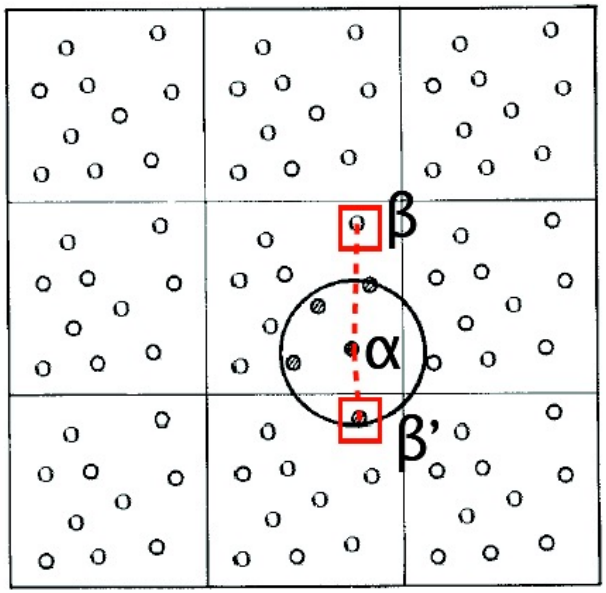
Drawbacks:
1) periodic boundary conditions induce correlations between cells with a length scale that has the same size as the cell. These correlations are artifacts and can have a non-negligible influence on the results.

How can you check the influence of this correlations on the results of your simulation?

2) The fundamental simulation cell must be commensurate with the system of study.



Minimum image convention



The interaction is calculated with the closest atom or image inside the cutoff.

Example the interaction between α and β atoms:

$$
\begin{aligned}
x'_{\alpha\beta} &= \min(x_\alpha - x_\beta, x_\alpha - x_\beta + L, x_\alpha - x_\beta - L), \\
y'_{\alpha\beta} &= \min(y_\alpha - y_\beta, y_\alpha - y_\beta + L, y_\alpha - y_\beta - L), \\
z'_{\alpha\beta} &= \min(z_\alpha - z_\beta, z_\alpha - z_\beta + L, z_\alpha - z_\beta - L),
\end{aligned}
$$

$$
r_{\alpha\beta} = \sqrt{{x'_{\alpha\beta}}^2 + {y'_{\alpha\beta}}^2 + {z'_{\alpha\beta}}^2}
$$

in this example the interaction is computed between α and the image β'

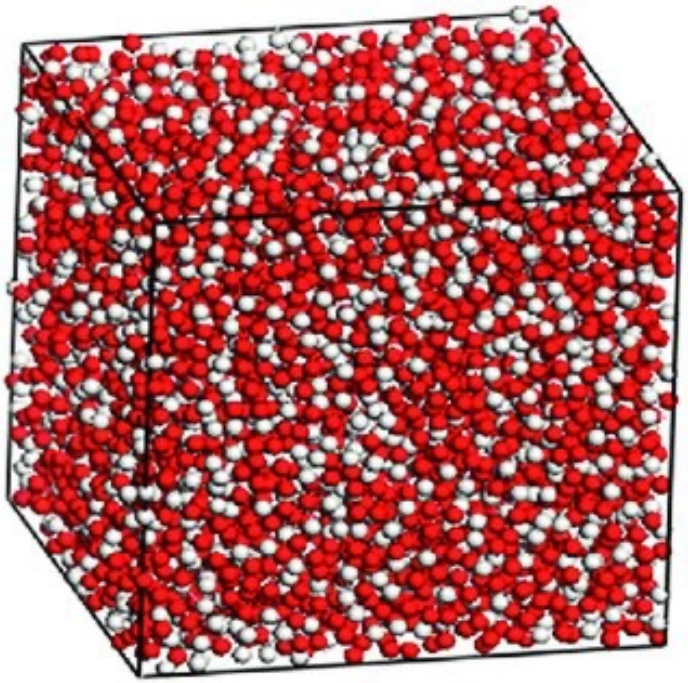
When periodic boundary conditions and cutoff are being used, the cutoff should not be so large that a particle "sees" its own image => the cutoff have to be no more than half the length of the cell.

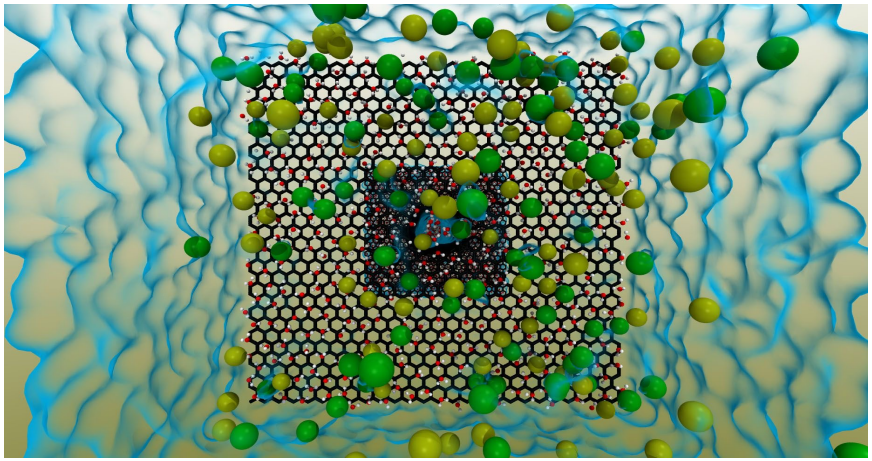
Basics of molecular dynamics

The basic idea is very simple:

Calculate the forces on atoms, solve Newton's equation to determine where they move.

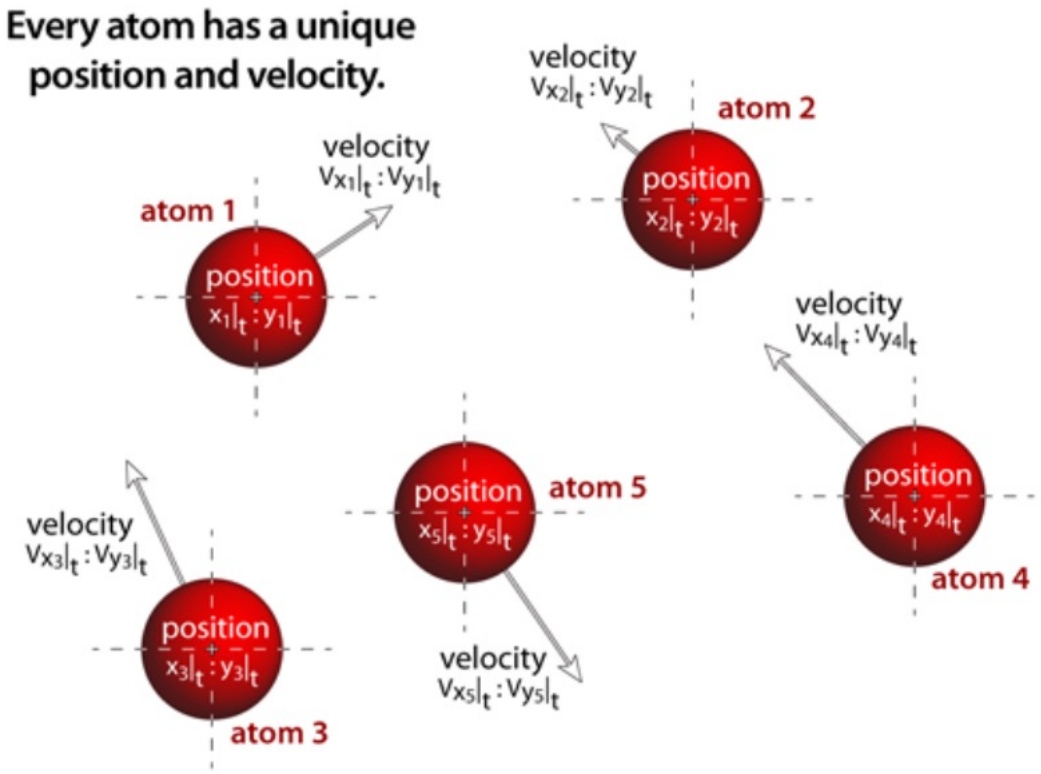
This means applying classical mechanics to atoms





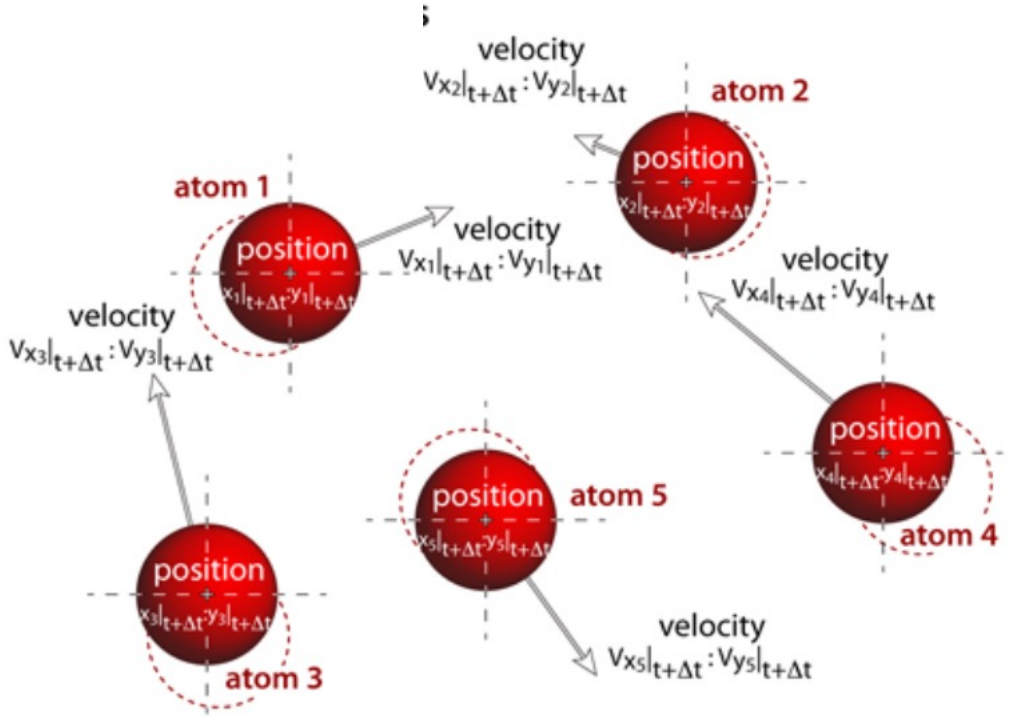
Initial situation

A MD simulation starts from a certain distribution of atoms, in our case a solid crystal. The atoms have a certain position in space, given by their coordinates in a cartesian reference system and a given assigned velocity. The system of atoms is subject to given boundary conditions.

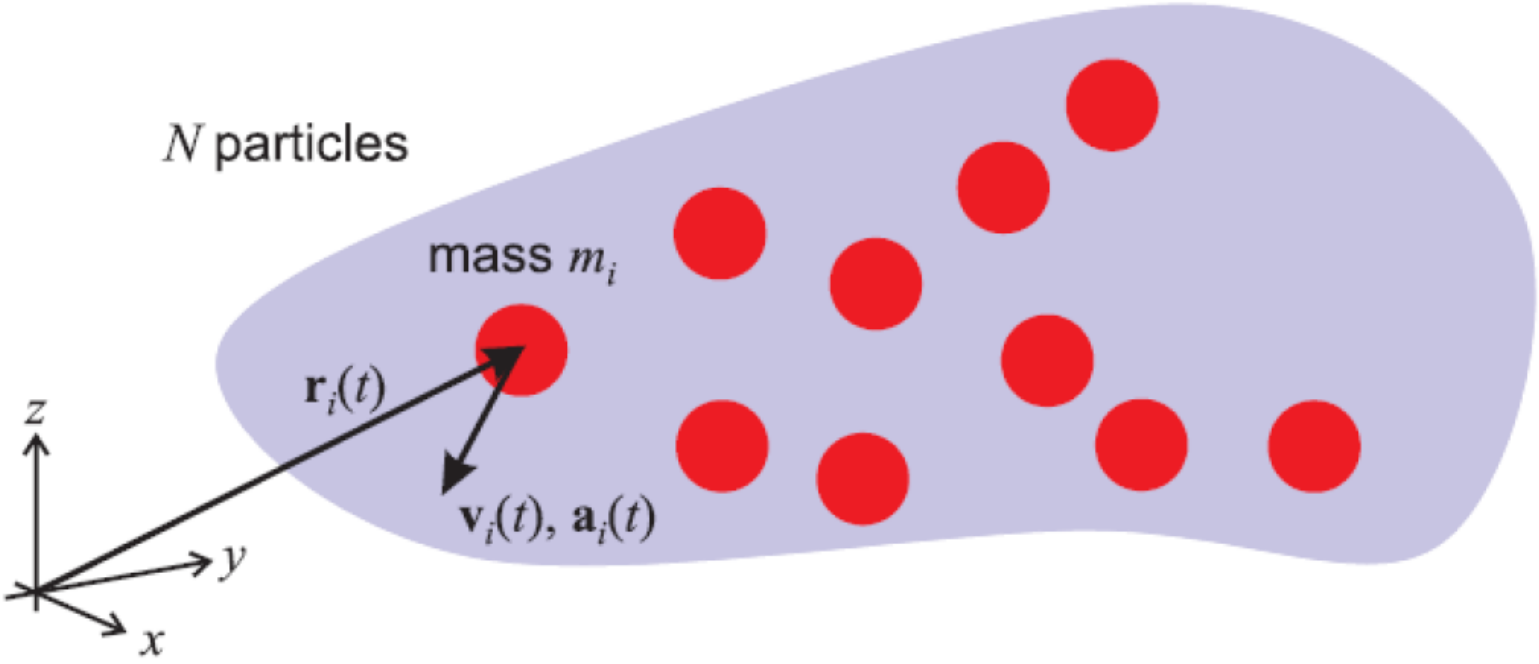


Final situation

The main goal of an atomistic simulation is to find the new positions and velocities of the atoms after a period in time.



MD: calculate trajectories in time



GOAL: find an algorithm to compute positions, velocities, accelerations, as a function of time

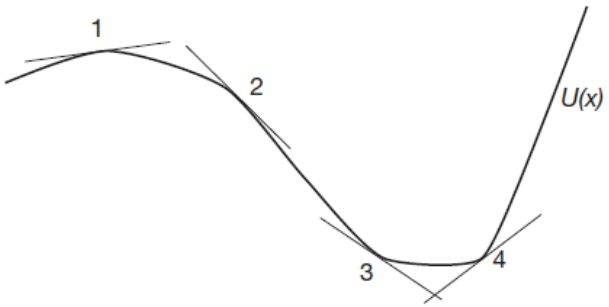
Classical mechanics in a nutshell

We want to determine the motion of an object subjected to a force
If the force is only a function of position and we know initial position and velocity, we can solve:

$$\mathbf{F} = m\mathbf{a} \qquad \mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d^2\mathbf{r}}{dt^2}$$

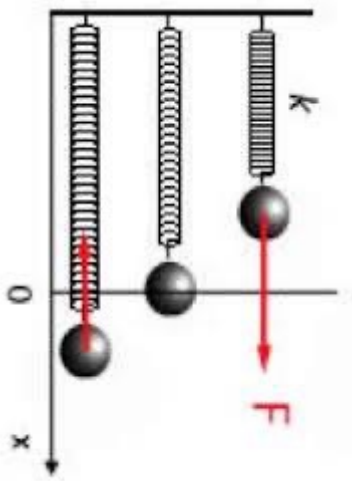
The potential energy is the work required to move an object against the force from an initial to a final position

$$U(x) = -\int_{x_o}^{x} F(x)dx$$



$$F = -\frac{\partial U(x)}{dx} \qquad \mathbf{F} = -\nabla U(\mathbf{r})$$

The harmonic oscillator



$$F = -k(x - x_o) = ma \qquad z = x - x_o$$

$$\textit{equation of motion} \qquad m\frac{d^2z}{dt^2} = -kz$$

$$z(t) = A\cos(\omega t) + B\sin(\omega t)$$

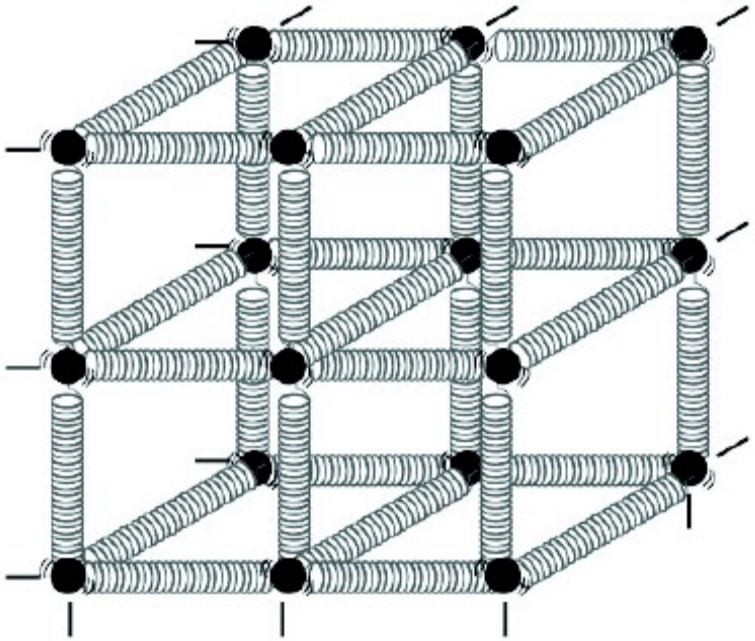
$$\textit{angular frequency} \qquad \omega = \sqrt{k/m}$$

$$v(t) = \frac{dz(t)}{dt} = -\omega A\sin(\omega t) + \omega B\cos(\omega t)$$

A, B are determined through the initial conditions

$$U = \frac{1}{2}k(x - x_o)^2 = \frac{1}{2}kz^2$$

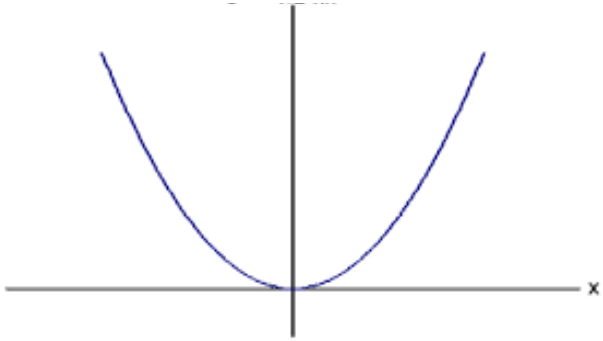
Atomic Model



Spring-ball model for the illustration of elastic behaviour of solids.

Potential energy of a spring:

$$U=1/2kx^2$$



Pair of interacting atoms

$$U(\mathbf{r}_i, \mathbf{r}_j) = \phi(r_{ij})$$

$$\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i = (x_j - x_i)\hat{x} + (y_j - y_i)\hat{y} + (z_j - z_i)\hat{z}$$

The force on atom *i* caused by *j*:

$$\mathbf{F}_i(\mathbf{r}_{ij}) = -\nabla_i \phi(r_{ij}) = -\left(\frac{\partial}{\partial x_i}\hat{x} + \frac{\partial}{\partial y_i}\hat{y} + \frac{\partial}{\partial z_i}\hat{z}\right)\phi(r_{ij})$$

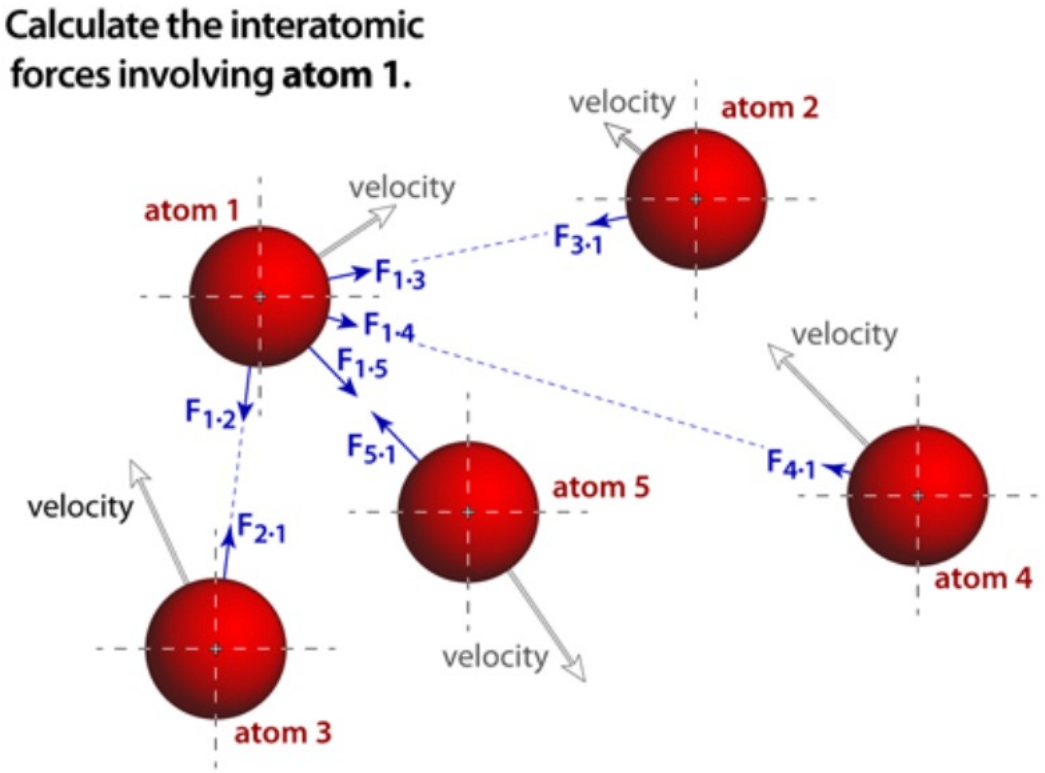
$$\frac{\partial \phi(r_{ij})}{\partial x_i} = \frac{\partial \phi(r_{ij})}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial x_i} = -\frac{\partial \phi(r_{ij})}{\partial r_{ij}}\frac{x_j - x_i}{r_{ij}}$$

$$\mathbf{F}_i(\mathbf{r}_{ij}) = \frac{\partial \phi(r_{ij})}{\partial r_{ij}}\left(\frac{x_j - x_i}{r_{ij}}\hat{x} + \frac{y_j - y_i}{r_{ij}}\hat{y} + \frac{z_j - z_i}{r_{ij}}\hat{z}\right) = \frac{\partial \phi(r_{ij})}{\partial r_{ij}}\frac{\mathbf{r}_{ij}}{r_{ij}}$$

$$\mathbf{F}_j(\mathbf{r}_{ij}) = -\frac{\partial \phi(r_{ij})}{\partial r_{ij}}\frac{\mathbf{r}_{ij}}{r_{ij}}$$

The form of a general pair potential is: $\phi(r_{ij}) = \dfrac{C}{r^n}$ $\qquad$ $\mathbf{F}_i = -n\dfrac{C}{r^{(n+1)}}\hat{r}_{ij}$

Interatomic forces



In principle each atom is interacting with all of the other atoms in the simulation, and thus the forces must be computed between every possible pair of atoms in the simulation.
Atoms respond also to external forces, if present.

Equations of motion for an atomic crystal

For each of the N atoms in the crystal, the force reads

$$\mathbf{F}_i = m_i \mathbf{a}_i = m_i \frac{d^2 \mathbf{r}_i}{dt^2}$$

$$\mathbf{F}_i = -\nabla_i U(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) = -\nabla_i U(\mathbf{r}^N)$$

For a pair potential like L_J: $U(\mathbf{r}^N) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N}{}' \phi_{ij}(r_{ij})$

$$\nabla_i U(\mathbf{r}^N) = \sum_{j \neq i} \left\{ \frac{d\phi_{ij}(r_{ij})}{dr_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}} \right\} = \sum_{j \neq i} \mathbf{f}_{ij}(r_{ij})$$

Equation of motion of atom i $\frac{d^2 \mathbf{r}_i}{dt^2} = \frac{1}{m_i} \mathbf{F}_i = \frac{1}{m_i} \sum_{j \neq i} \mathbf{f}_{ij}(r_{ij})$ 3N coupled equations

Solving these equations, one per coordinate of each atom in the system is no different than solving the equation of motion for an harmonic oscillator. One would then obtain the trajectory of all atoms.

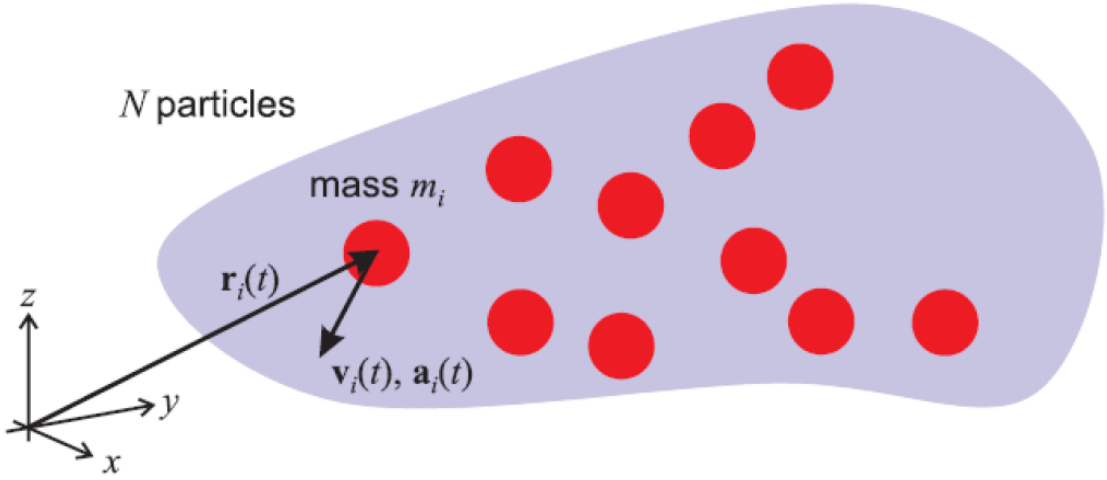
MD: calculate trajectories in time

These 3N equations (one per each d.o.f. per each atom):

$$\frac{d^2\mathbf{r}_i}{dt^2} = \frac{1}{m_i}\mathbf{F}_i = \frac{1}{m_i}\sum_{j\neq i}\mathbf{f}_{ij}(r_{ij})$$

are however highly coupled and no analytical solution exists

The key of effective MD simulations is to solve the N coupled equations **numerically** accurately and efficiently



Numerical integration of Newton's equation

To solve the equation of motion for the atoms numerically, one firstly needs to specify positions and velocities at time zero. Then time is broken into intervals δt, called *time steps*. The equations of motion are solved over those increments.

$$r_i(t_0) \rightarrow r_i(t_0 + \Delta t) \rightarrow r_i(t_0 + 2\Delta t) \rightarrow r_i(t_0 + 3\Delta t) \rightarrow \ldots \rightarrow r_i(t_0 + n\Delta t)$$

Remember Taylor expansion of a function around a point:

$$f(x + \Delta x) = f(x) + \frac{\Delta x}{1!} f'(x) + \frac{\Delta x^2}{2!} f''(x) + \frac{\Delta x^3}{3!} f'''(x) + \frac{\Delta x^4}{4!} f''''(x) + O(\Delta x^5)$$

$$r_i(t_0 + \Delta t) = r_i(t_0) + v_i(t_0)\Delta t + \frac{1}{2} a_i(t_0)\Delta t^2 + \ldots$$

Do we know what $a_i(t_0)$ is?

Numerical integration of Newton's equation

1) Assumption→ positions and velocities at $t+\delta t$ are found from forces evaluated at time t
2) This implies that $\mathbf{a}_i = d^2\mathbf{r}_i/dt^2 = \mathbf{F}_i(t)/m$ is assumed constant over a time interval

$$\mathbf{r}_i(t+\delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\delta t + \frac{1}{2}\mathbf{a}_i(t)\delta t^2$$

$$\mathbf{v}_i(t+\delta t) = \mathbf{v}_i(t) + \mathbf{a}_i(t)\delta t$$

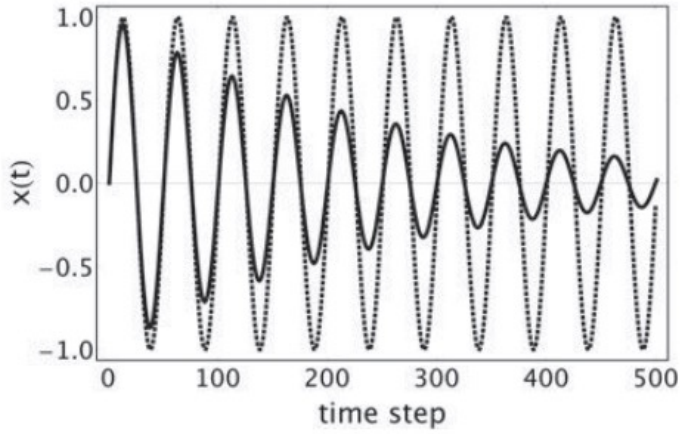
Numerical integration of Newton's equation

1) Assumption→ positions and velocities at $t+\delta t$ are found from forces evaluated at time t
2) This implies that also $\mathbf{a}_i = d^2\mathbf{r}_i/dt^2 = \mathbf{F}_i(t)/m$ is constant over a time interval

$$\mathbf{r}_i(t+\delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\delta t + \frac{1}{2}\mathbf{a}_i(t)\delta t^2$$

$$\mathbf{v}_i(t+\delta t) = \mathbf{v}_i(t) + \mathbf{a}_i(t)\delta t$$

Let's apply this integration scheme to a harmonic oscillator:



The Verlet algorithm

$$r_i(t_0 + \Delta t) = r_i(t_0) + v_i(t_0)\Delta t + \frac{1}{2}a_i(t_0)\Delta t^2 + \ldots$$

$$r_i(t_0 - \Delta t) = r_i(t_0) - v_i(t_0)\Delta t + \frac{1}{2}a_i(t_0)\Delta t^2 + \ldots$$

The Verlet algorithm

$$r_i(t_0 + \Delta t) = r_i(t_0) + v_i(t_0)\Delta t + \frac{1}{2}a_i(t_0)\Delta t^2 + \ldots$$

$$+ \left[r_i(t_0 - \Delta t) = r_i(t_0) - v_i(t_0)\Delta t + \frac{1}{2}a_i(t_0)\Delta t^2 + \ldots \right]$$

$$\downarrow$$

$$r_i(t_0 - \Delta t) + r_i(t_0 + \Delta t) = 2r_i(t_0) - v_i(t_0)\Delta t + v_i(t_0)\Delta t + a_i(t_0)\Delta t^2 + \ldots$$

The Verlet algorithm

$$r_i(t_0+\Delta t) = r_i(t_0) + v_i(t_0)\Delta t + \frac{1}{2}a_i(t_0)\Delta t^2 + \ldots$$

$$+\left[\, r_i(t_0-\Delta t) = r_i(t_0) - v_i(t_0)\Delta t + \frac{1}{2}a_i(t_0)\Delta t^2 + \ldots \,\right]$$

$$\downarrow$$

$$r_i(t_0-\Delta t) + r_i(t_0+\Delta t) = 2r_i(t_0) - v_i(t_0)\Delta t + v_i(t_0)\Delta t + a_i(t_0)\Delta t^2 + \ldots$$

$$\downarrow$$

$$r_i(t_0+\Delta t) = 2r_i(t_0) - r_i(t_0-\Delta t) + a_i(t_0)\Delta t^2 + \ldots$$

The Verlet algorithm

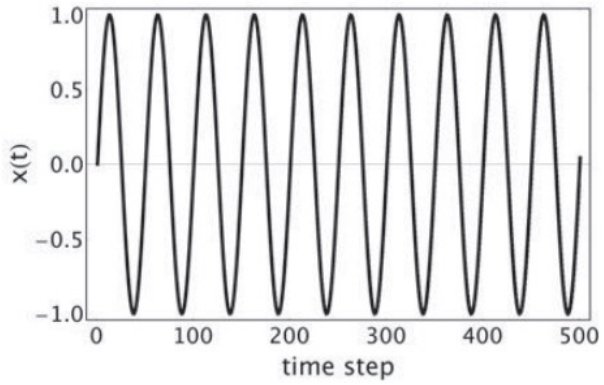
Position forward in time $$\mathbf{r}_i(t+\delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\delta t + \frac{1}{2}\mathbf{a}_i(t)\delta t^2$$

Position backwards in time $$\mathbf{r}_i(t-\delta t) = \mathbf{r}_i(t) - \mathbf{v}_i(t)\delta t + \frac{1}{2}\mathbf{a}_i(t)\delta t^2$$

$$\mathbf{r}_i(t+\delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\delta t) + \mathbf{a}_i(t)\delta t^2$$

The acceleration is then evaluated at each time step from the force

$$\mathbf{v}_i(t) = \frac{\mathbf{r}_i(t+\delta t) - \mathbf{r}_i(t-\delta t)}{2\delta t}$$ needed?



The velocity Verlet algorithm

$$\mathbf{r}_i(t+\delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\,\delta t + \frac{1}{2m}\mathbf{F}_i(t)\delta t^2$$

$$\mathbf{v}_i(t+\delta t) = \mathbf{v}_i(t) + \frac{1}{2m}\big(\mathbf{F}_i(t) + \mathbf{F}_i(t+\delta t)\big)\delta t$$

Take home messages

- Molecular dynamics simulations treat atoms as if they were Newtonian particles
- The goal of an MD simulation is to track the trajectory of all atoms while the system evolves in time. Atoms respond to external forces and to forces exchanged between themselves.
- The equation of motion of all atoms is computed numerically using an integration scheme of the type of the Verlet algorithm.

Create your own MD code

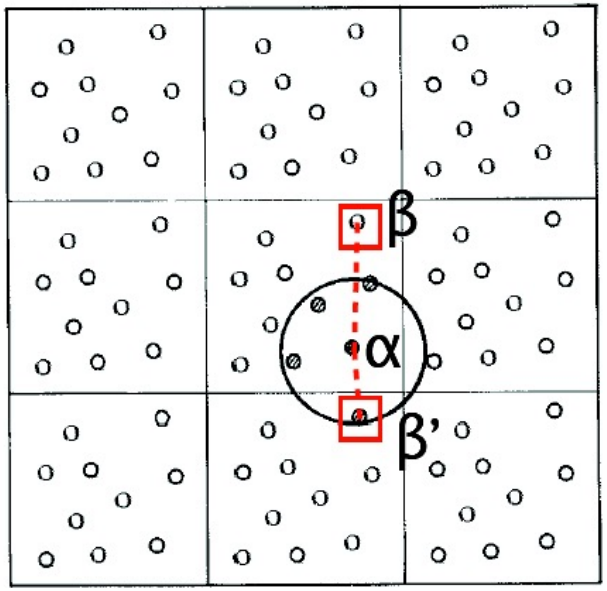
1. Initialize positions (done!) and velocities (to be done)
2. Initialize energies (almost done) and forces (to be done)
3. Apply the Verlet algorithm to find trajectories of atoms (next class)
4. Calculate forces and energies at new positions (next class)

In class assignment 1

Apply periodicity to the unit cell representing your FCC and BCC crystals by means of the minimum image convention and calculate the energy per atom of the crystal.

Check the effect that the size of the cell and the cut-off have on convergence.

In class assignment 1



The interaction is calculated with the closest atom or image inside the cutoff.

Example the interaction between α and β atoms:

$$
\begin{aligned}
x'_{\alpha\beta} &= \min(x_\alpha - x_\beta, x_\alpha - x_\beta + L, x_\alpha - x_\beta - L),\\
y'_{\alpha\beta} &= \min(y_\alpha - y_\beta, y_\alpha - y_\beta + L, y_\alpha - y_\beta - L),\\
z'_{\alpha\beta} &= \min(z_\alpha - z_\beta, z_\alpha - z_\beta + L, z_\alpha - z_\beta - L),
\end{aligned}
$$

$$
r_{\alpha\beta} = \sqrt{{x'_{\alpha\beta}}^2 + {y'_{\alpha\beta}}^2 + {z'_{\alpha\beta}}^2}
$$

in this example the interaction is computed between α and the image β'

When periodic boundary conditions and cutoff are being used, the cutoff should not be so large that a particle "sees" its own image => the cutoff have to be no more than half the length of the cell.

In class assignment 1

TIP:

x = x – round(x)

Let us examine what x = x - round(x) does.
If x >1/2, it replaces x by x-1.
If x <-1/2, it adds 1,
otherwise leaves x unchanged.

Thus, it selects only the nearest image of atom i, if in a cell 1*1*1

Minimum image convention

```
function[ucell]= latsum(n,s)
nc=5;
amin=1.2;
amax=2.3;
step=0.005
nsteps=round((amax-amin)/step)+1
[s,i1]=fccmke(nc);
n=i1;
m=0;
ucell=zeros(nsteps,1);
for a=amin:step:amax
    m=m+1;
    b(m)=a;
    for i = 1:n
        for j=1:n
            xij = s(j,1)-s(i,1);
            yij = s(j,2)-s(i,2);
            zij = s(j,3)-s(i,3);
            xij = xij - round(xij);
            yij = yij - round(yij);
            zij = zij - round(zij);
            dist = a*nc*sqrt(xij^2+yij^2+zij^2);
                if dist>0
                phi=4*(1/dist^(12)-1/dist^6); %(Lennard-Jones)
                else
                phi=0;
                end
        ucell(m) = ucell(m) + phi;
        end
    end
    ucell(m) = ucell(m)/2; %because I am double counting contributions
    ucell(m) = ucell(m)/n; %lattice energy per atom (in reduced units)
end
plot(b,ucell)
```

In class assignment 2

Generate the initial velocities of the atoms in your FCC crystal to be used in a molecular dynamics simulation. Choose the velocities from a Maxwell-Boltzmann distribution, ensuring that the system is at a prescribed temperature.

Instead of using s(i, alpha) for the coordinates of the atoms, use x(j), y(j), z(j). The code becomes more readable.

There is an important link between kinetic energy and temperature

$$K = 3Nk_BT / 2$$

InitMDLJ

```
function[n,sx,sy,sz,vx,vy,vz]= initLJMD(nc,tin)
%  created scaled coordinates in an fcc lattice
nc=5;
tin=0.6;
ncell=4; % number of atoms in a cell
x=[0 .5 0 .5];
y=[0 .5 .5 0];
z=[0 0 .5 .5];
i1=0;
n = ncell*nc^3;
sx=zeros(n,1);
sy=zeros(n,1);
sz=zeros(n,1);
vx=zeros(n,1);
vy=zeros(n,1);
vz=zeros(n,1);
for k=1:nc
    for l = 1:nc
        for m = 1:nc
            for i = 1:ncell
                i1=i1+1;
                sx(i1) = (x(i) + k-1)/nc;
                sy(i1) = (y(i) + l-1)/nc;
                sz(i1) = (z(i) + m-1)/nc;
            end
        end
    end
end
```

Generate initial velocities

Step 1: Generate velocities

$$\rho(v_x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-v_x^2/2\sigma^2}$$

where the standard deviation is $\sigma = \sqrt{k_B T/m}$

We can generate a set of random numbers from a Gaussian (normal) distribution with zero mean and unit standard deviation $\sigma = 1$ by choosing

y1 = sqrt(- 2 ln(x1)) cos(2 pi x2)

x1 and x2 are two random numbers

Random velocities

```
for i=1:n
    vx(i) = sqrt(-2*log(rand))*cos(2*pi*rand);
    vy(i) = sqrt(-2*log(rand))*cos(2*pi*rand);
    vz(i) = sqrt(-2*log(rand))*cos(2*pi*rand);
```

Generate initial velocities

Step 2. Eliminate any drift of the atoms.

Generate initial velocities

Step 2. Eliminate any drift of the atoms.

Calculate total momentum
Calculate momentum per atom
Subtract momentum from atomic velocity

What is now the temperature of the crystal? Is it the one we have imposed?

Find a way to go back to the required temperature

InitMDLJ

```
k = 0;
px = 0;
py = 0;
pz = 0;
for i=1:n
    vx(i) = sqrt(-2*log(rand))*cos(2*pi*rand);
    vy(i) = sqrt(-2*log(rand))*cos(2*pi*rand);
    vz(i) = sqrt(-2*log(rand))*cos(2*pi*rand);
    px = px + vx(i);
    py = py + vy(i);
    pz = pz + vz(i);
end
histogram(vy);
    % set net momentum to zero and calcuate K
    px = px/n;
    py = py/n;
    pz = pz/n;
    for i=1:n
        vx(i) = vx(i)-px;
        vy(i) = vy(i)-py;
        vz(i) = vz(i)-pz;
        k = k + vx(i)^2 + vy(i)^2 + vz(i)^2;
    end
    k = .5*k;
% kinetic energy of desired temperature (tin)
kin = 3*n*tin/2;
% rescale velocities
sc=sqrt(kin/k);
for i=1:n
    vx(i) = vx(i)*sc;
    vy(i) = vy(i)*sc;
    vz(i) = vz(i)*sc;
end
histogram(vy);
```