

MD and Lammps Tutorial

UNLV

Pedro A Santos Florez
Sep 15, 2021
Group Meeting

Summary

Basic Principles

Code compilation

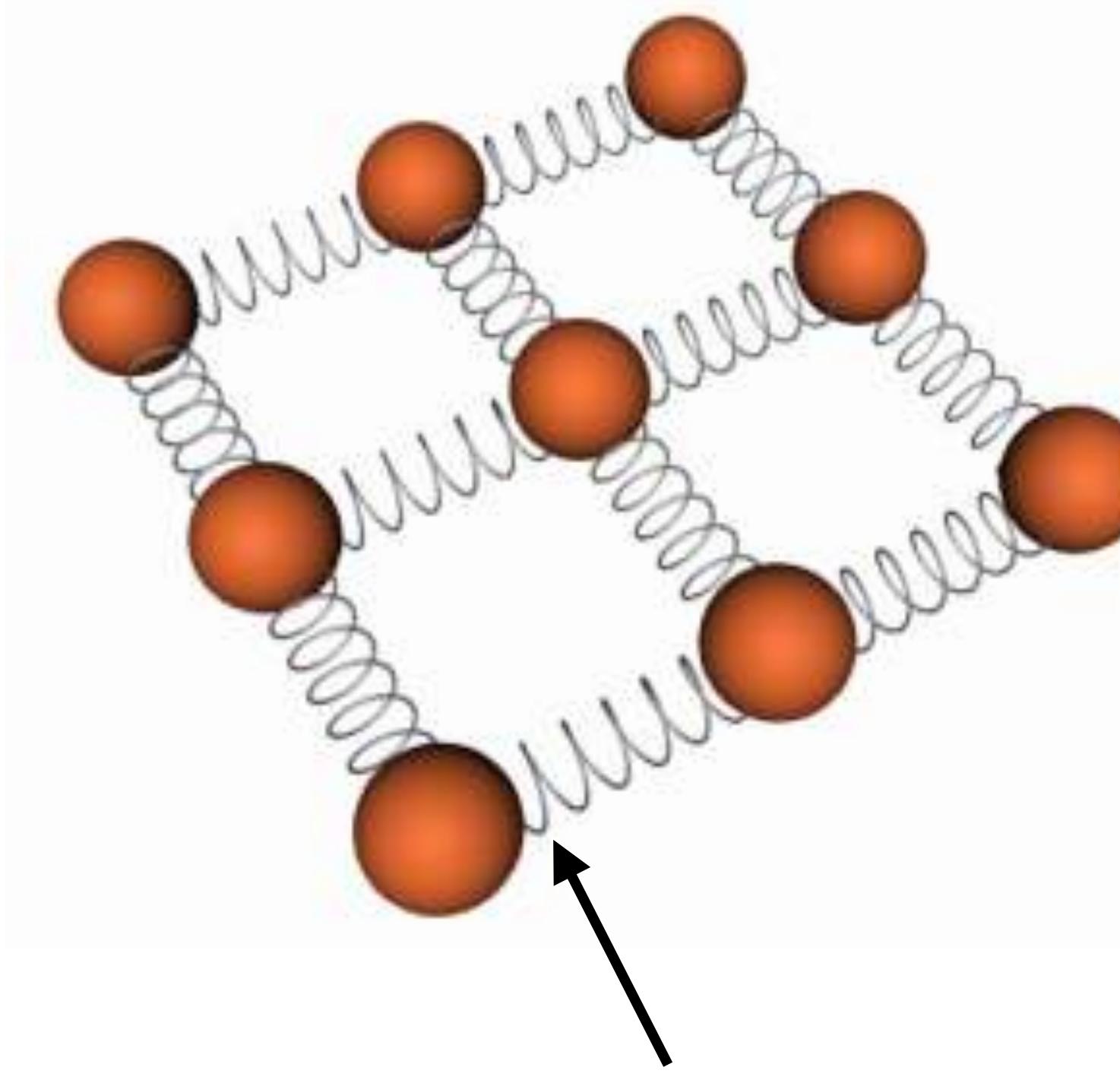
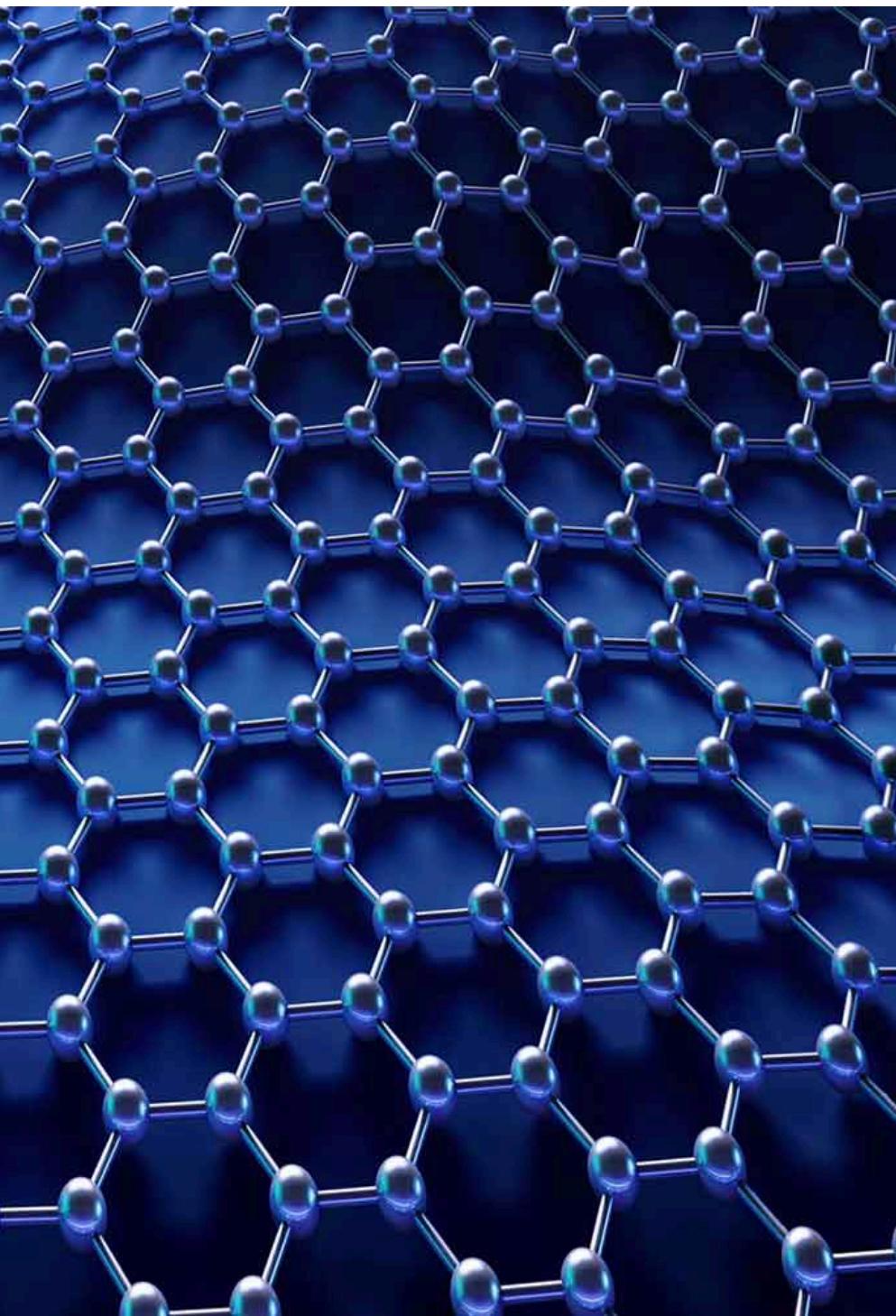
Examples

Basic Analysis of the results

Basic Principles

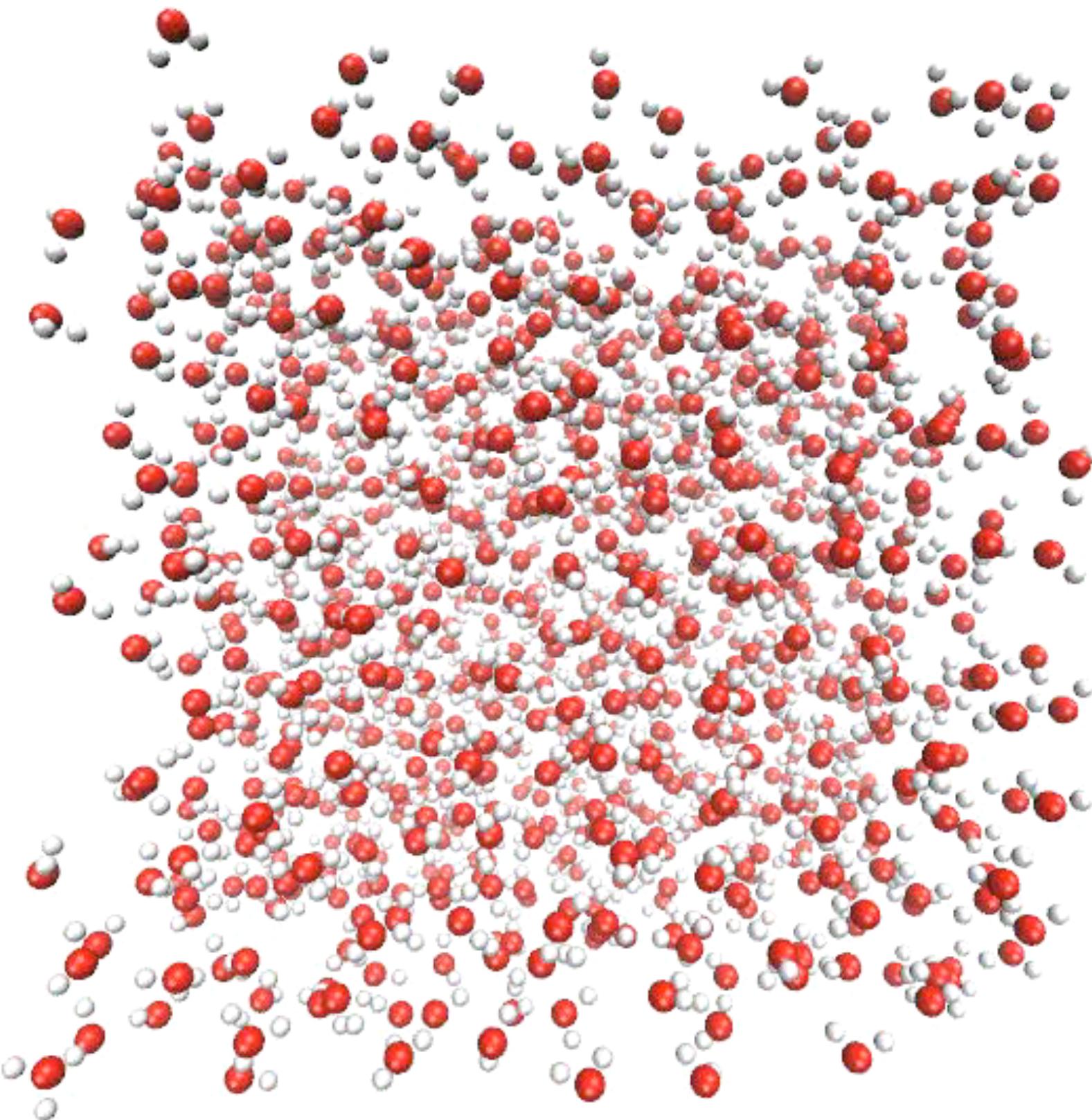
Atomistic Simulations

The techniques known as atomistic simulations, allow the study of the movement of atoms in materials, and the collective behaviour of these atoms allows us to understand, for example, how a material undergoes deformation, phase transitions, among other phenomena.



interactions between atoms

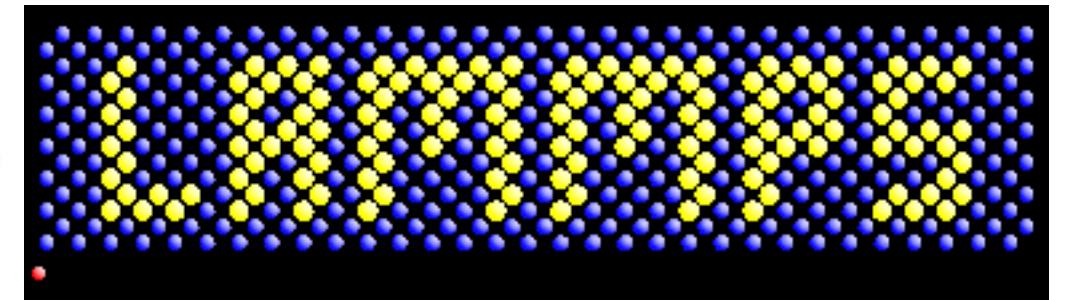
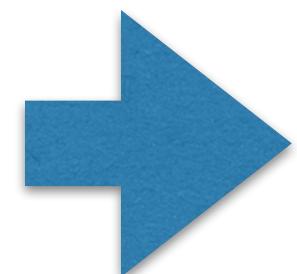
Molecular dynamics simulations



$$\frac{d^2\mathbf{r}_i(t)}{dt^2} = \frac{\mathbf{F}_i}{m_i} = -\frac{1}{m_i} \frac{\partial V(\mathbf{r}^N)}{\partial \mathbf{r}_i}$$

3N equações de movimento

Large-scale Atomic/Molecular Massively Parallel Simulator



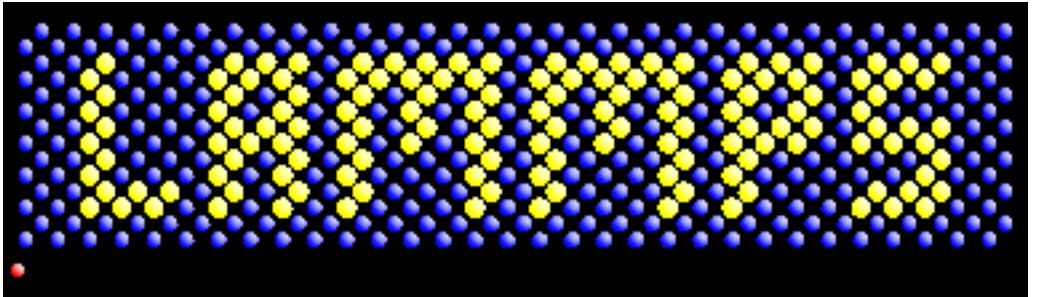
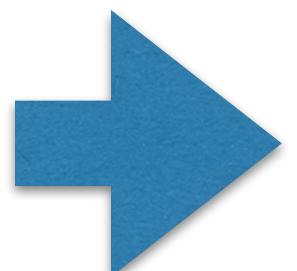
Molecular dynamics

Molecular Dynamics (MD) simulations are an extremely useful alternative avenue of investigation that can be considered as a method of "computational microscopy *in situ*", since they provide a detailed view of the mechanisms operating on atomic scales of length and time

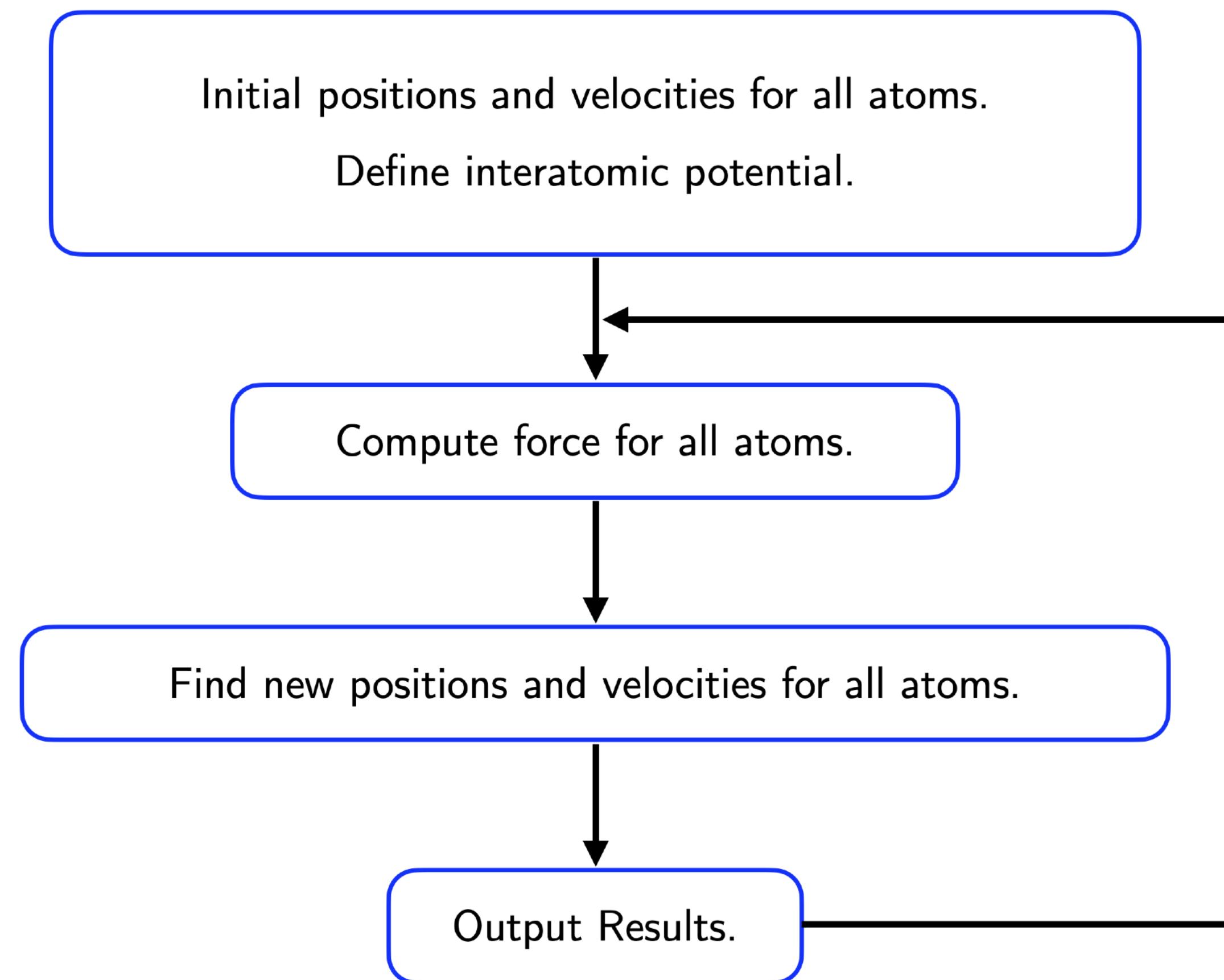
time: femto-seconds to nanoseconds

size: few atoms to millions and maybe billions of atoms.

Large-scale Atomic/Molecular Massively Parallel Simulator



Atomistic Simulations



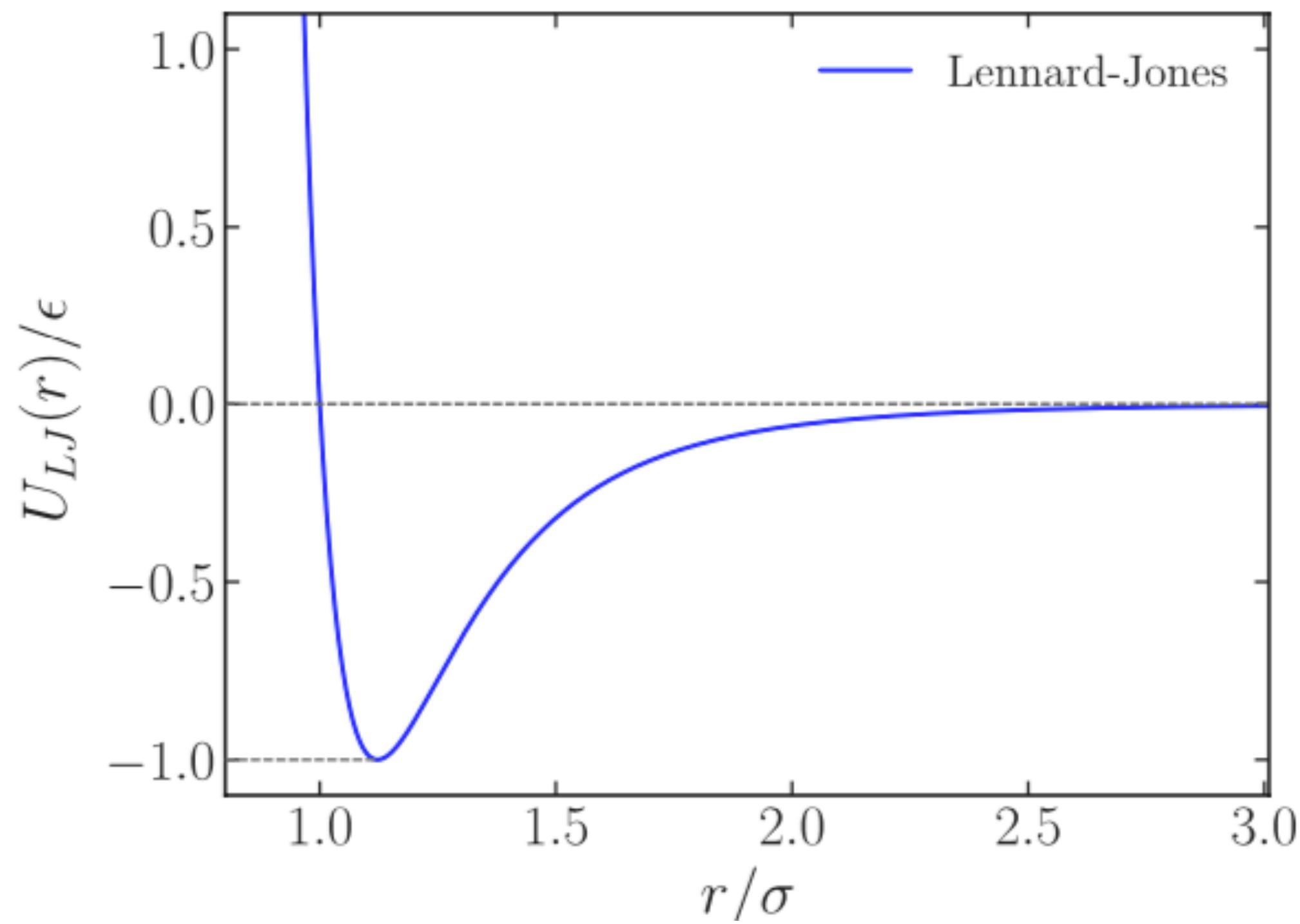
Repeat each Δt

The optimization of this process is the basis of MD!

Interatomic Potentials (Empirical)

Pairwise potentials

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$



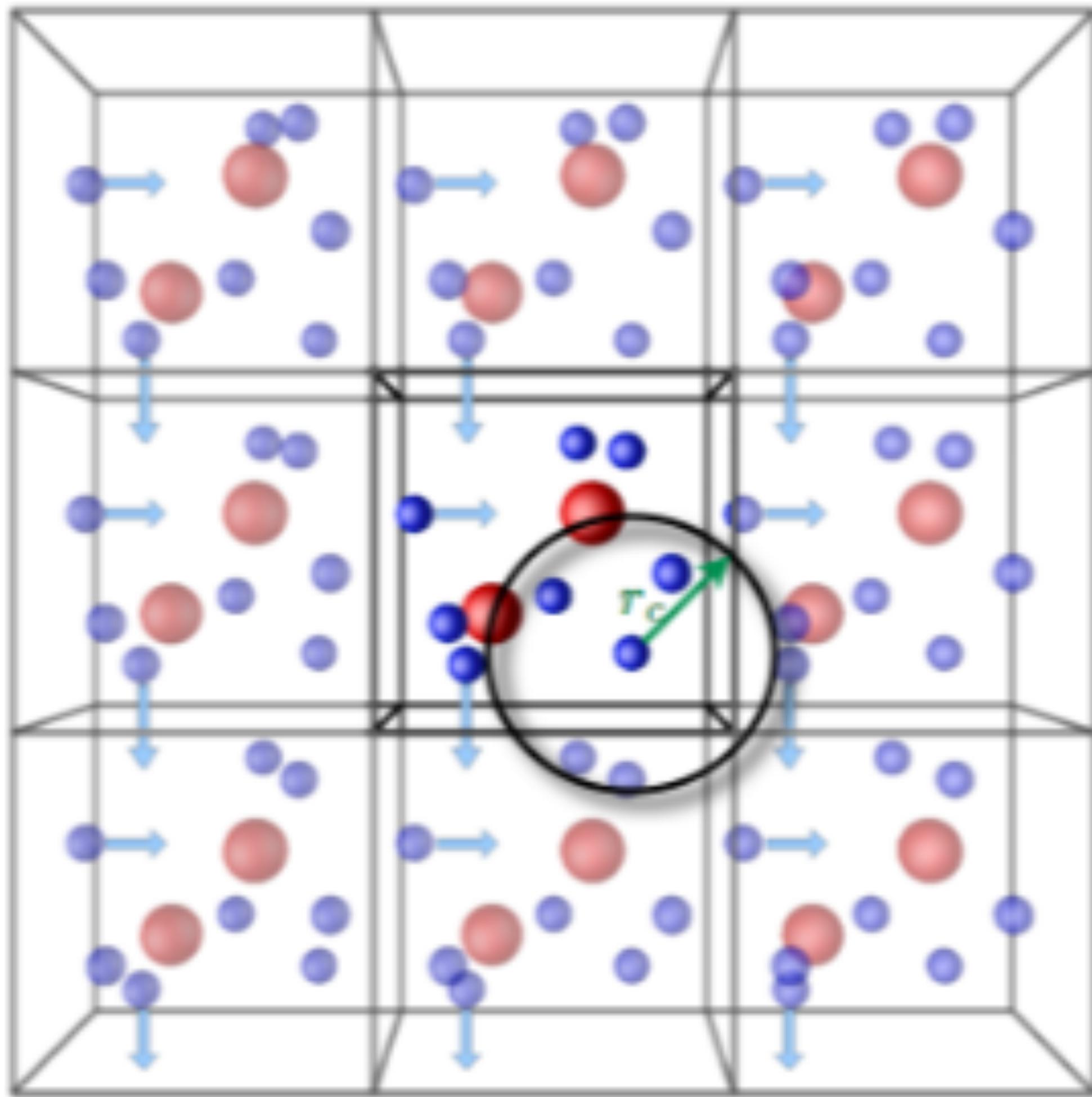
Bond-order potentials

$$U = \sum_i \sum_{i>i} \phi_2(r_{ij}) + \sum_i \sum_{i \neq j} \sum_{k > j} \phi_3(r_{ij}, r_{ik}, \theta_{ijk})$$

coarse- grained models

Interactions using "pseudo-atoms" to represent groups of atoms, allowing to perform MD simulations on very large systems.

Boundary Conditions



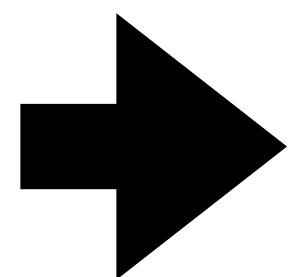
Periodic boundary conditions (PBC).

Cutting radius

$$r_c < 1/2 \text{ (box length)}$$

Integration of equations of motion

$$\frac{d^2\mathbf{r}_i(t)}{dt^2} = \frac{1}{m_i}\mathbf{F}_i(t) = -\frac{1}{m_i}\frac{\partial V(\mathbf{r}^N)}{\partial \mathbf{r}_i}.$$



$$\begin{aligned}\frac{d\mathbf{r}_i(t)}{dt} &= \mathbf{v}_i(t) \\ \frac{d\mathbf{v}_i(t)}{dt} &= \frac{\mathbf{F}_i(t)}{m_i},\end{aligned}$$

symplectic algorithms.

the equations remain unchanged under the exchange of t and $-t$ and \mathbf{p} and $-\mathbf{p}$) and the fact that a volume of the phase space is constant under canonical transformations (Liouville's theorem)

Integration of equations of motion

symplectic algorithm: Velocity-Verlet

scheme:

1. $\mathbf{p}(t + \frac{\Delta t}{2}) = \mathbf{p}(t) + \mathbf{F}(t) \frac{\Delta t}{2}$
2. $\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \frac{\mathbf{p}(t + \frac{\Delta t}{2})}{m} \Delta t$
3. Compute $\mathbf{F}(t + \Delta t)$ from $\mathbf{r}(t + \Delta t)$
4. $\mathbf{p}(t + \Delta t) = \mathbf{p}(t + \frac{\Delta t}{2}) + \mathbf{F}(t + \Delta t) \frac{\Delta t}{2}$

Integration of equations of motion

Key points in the construction of a MD simulation:

One of the key points in the construction of a MD simulation, besides a **good choice of the Δt time step**, is to implement an efficient integration algorithm, which satisfies criteria such as: speed of integration, conservation of the amplitude of movement (which indicates the veracity of the trajectories, because any small error in the numerical integration of the equations of motion, will blow up exponentially) and conservation of the energy of the system

Code compilation (Lammps)

Installing Lammmps

<https://docs.lammps.org/Install.html>

1. Download Lammmps.

2. Requirements:

C/C++ compiler (supporting the C++11 standard) **-std=c++11**
MPI

3. Build LAMMPS:

```
cd lammps/src/  
make mpi
```

The **src/MAKE** directory tree contains the **Makefile.<machine>** files included in the LAMMPS distribution. Typing **make example** uses **Makefile.example** from one of those folders, if available. Thus the **make serial** and **make mpi** lines above use **src/MAKE/Makefile.serial** and **src/MAKE/Makefile.mpi**, respectively. Other makefiles are in these directories:

OPTIONS	<i># Makefiles which enable specific options</i>
MACHINES	<i># Makefiles for specific machines</i>
MINE	<i># customized Makefiles you create (you may need to create this folder)</i>

4. Include packages in build:

```
cd lammps/src
```

```
make ps  
make yes-name  
make no-name  
make mpi
```

```
# check which packages are currently installed  
# install a package with name  
# un-install a package with name  
# build LAMMPS with whatever packages are now installed
```

Basics of running LAMMPS

LAMMPS is run from the command line, reading commands from a file via the `-in` command line flag, or from standard input. Using the “`-in in.file`” variant is recommended:

```
$ lmp_serial -in in.file  
$ lmp_serial < in.file  
$ /path/to/lammps/src/lmp_serial -i in.file  
$ mpirun -np 4 lmp_mpi -in in.file  
$ mpirun -np 8 /path/to/lammps/src/lmp_mpi -in in.file  
$ mpirun -np 6 /usr/local/bin/lmp -in in.file
```

You normally run the LAMMPS command in the directory where your input script is located.

Lammps on XSEDE, Expanse

Examples

Elastic constants

The elastic stiffness of a substance is described by a rank- four tensor with components C_{ijkl} that establishes the relation between the state of deformation, specified by the rank-two strain tensor ϵ_{ij} , and the corresponding applied stress tensor σ_{ij}

$$\sigma_{ij} = \sum_{\substack{kl= \\ x,y,z}} C_{ijkl} \epsilon_{kl},$$

Due to the symmetries, these rank-4 tensors can be represented in terms of 6×6 matrices by using Voigt notation:

(11) \rightarrow 1, (22) \rightarrow 2, (33) \rightarrow 3, (23), (32) \rightarrow 4, (13), (31) \rightarrow 5, and (12), (21) \rightarrow 6

$$\sigma_k = C_{kl} \epsilon_l, \quad k = 1, \dots, 6. \quad \longrightarrow \quad C_{kl} = - \frac{d\sigma^{\text{int}}_k}{d\epsilon_l}$$

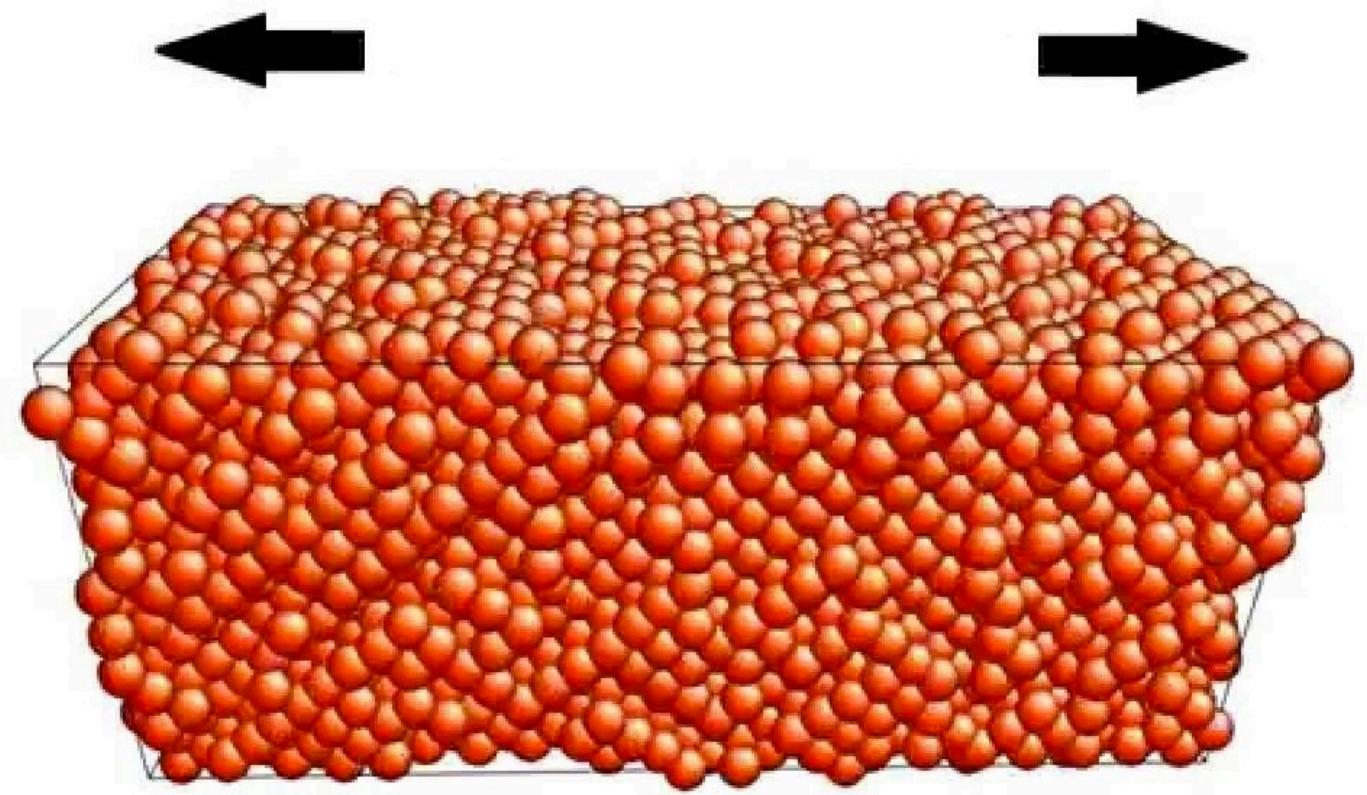
For T= 0K

Instantaneous deformation

$$C_{kl} = -\frac{\Delta \sigma_k^{\text{int}}}{\Delta \epsilon_l}$$

Small Deformations

xx
yy
zz
xy
xz
yz



Cubic Symmetries:

$$C_{12}=C_{21} \quad C_{11}=C_{22}=C_{33}$$

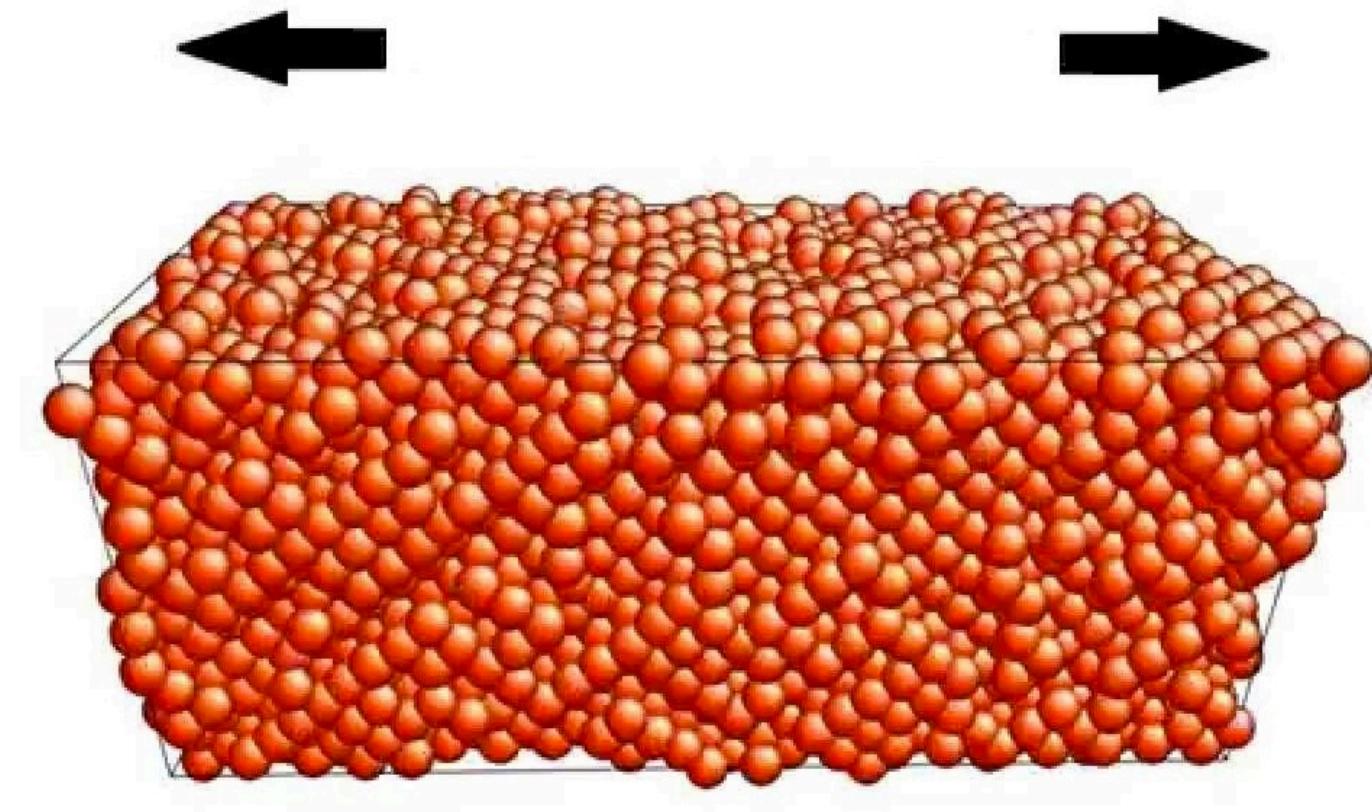
$$C_{13}=C_{31} \quad C_{12}=C_{13}=C_{23}$$

$$C_{23}=C_{32} \quad C_{44}=C_{55}=C_{66}$$

Uniaxial tensile deformation

To study mechanical properties and the reaction of a material when a load is applied.

Generally, a comparison is made between stress as a function of strain and stress as a function of deformation.



What we can predict?

Failure modes, stiffness and failure strength.

Lattice dislocations, defects and phase transitions.

Uniaxial tensile deformation

High Entropy Alloys

NbMoTaW (BCC)

using the spectral neighbor analysis potential (SNAP) approach

Let's see!

Basic Analysis of the results

To analyzes the results, we generally rely on thermodynamic variables that are calculated as a function of time ([outputs files](#)), or the average of a thermodynamic variable as a function of another controlled variable, such as temperature or pressure, ([plotting and calculating values using python, gnuplot, xmGrace, etc](#)), and we can also perform a structural analysis from the atomic positions as a function of time ([OVITO, VMD, python, etc](#)).

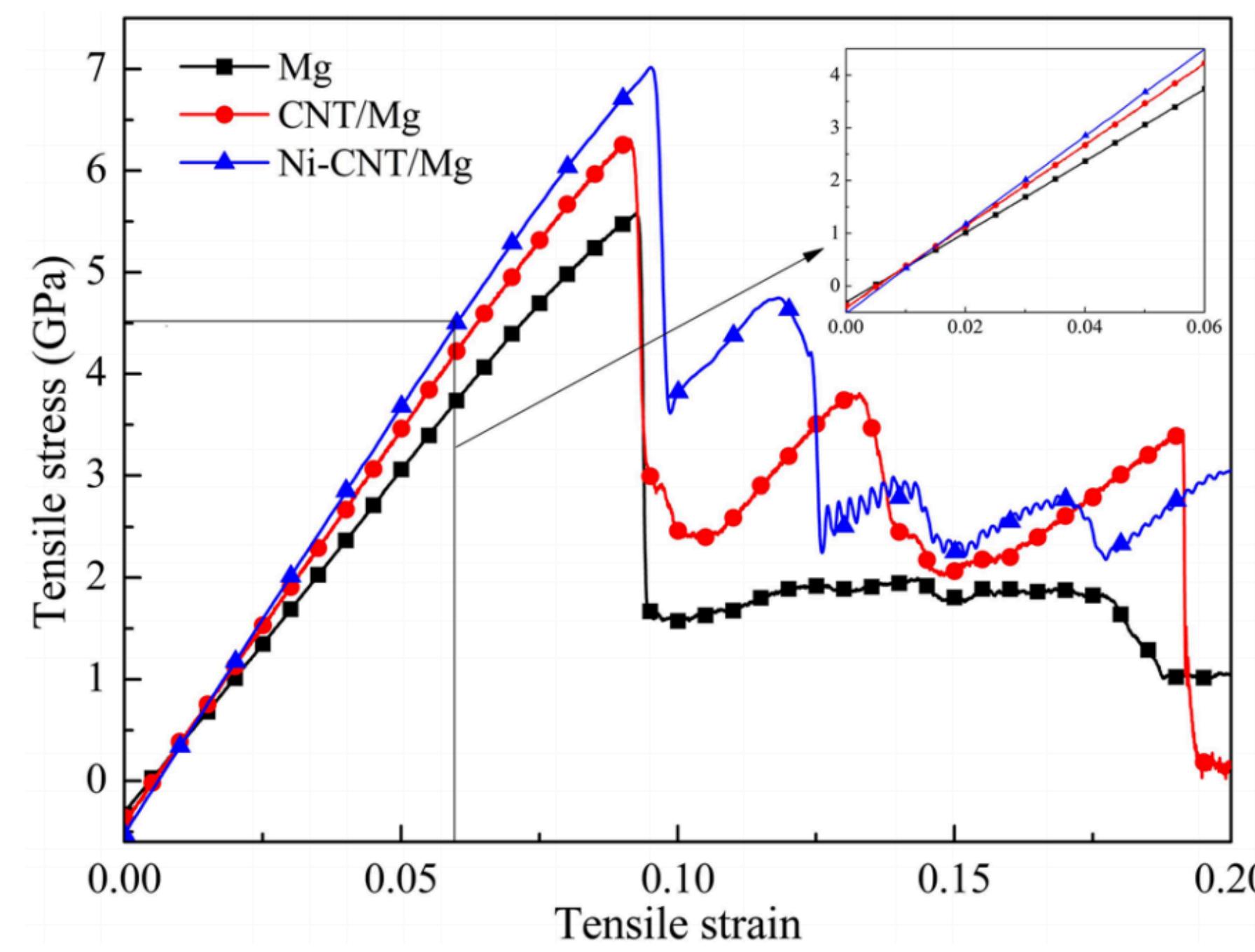


Fig. 3 Stress–strain curves for Ni-(6,6)CNT/Mg,(6,6)CNT/Mg and monolithic Mg at 300 K and $1 \times 10^9 \text{ s}^{-1}$

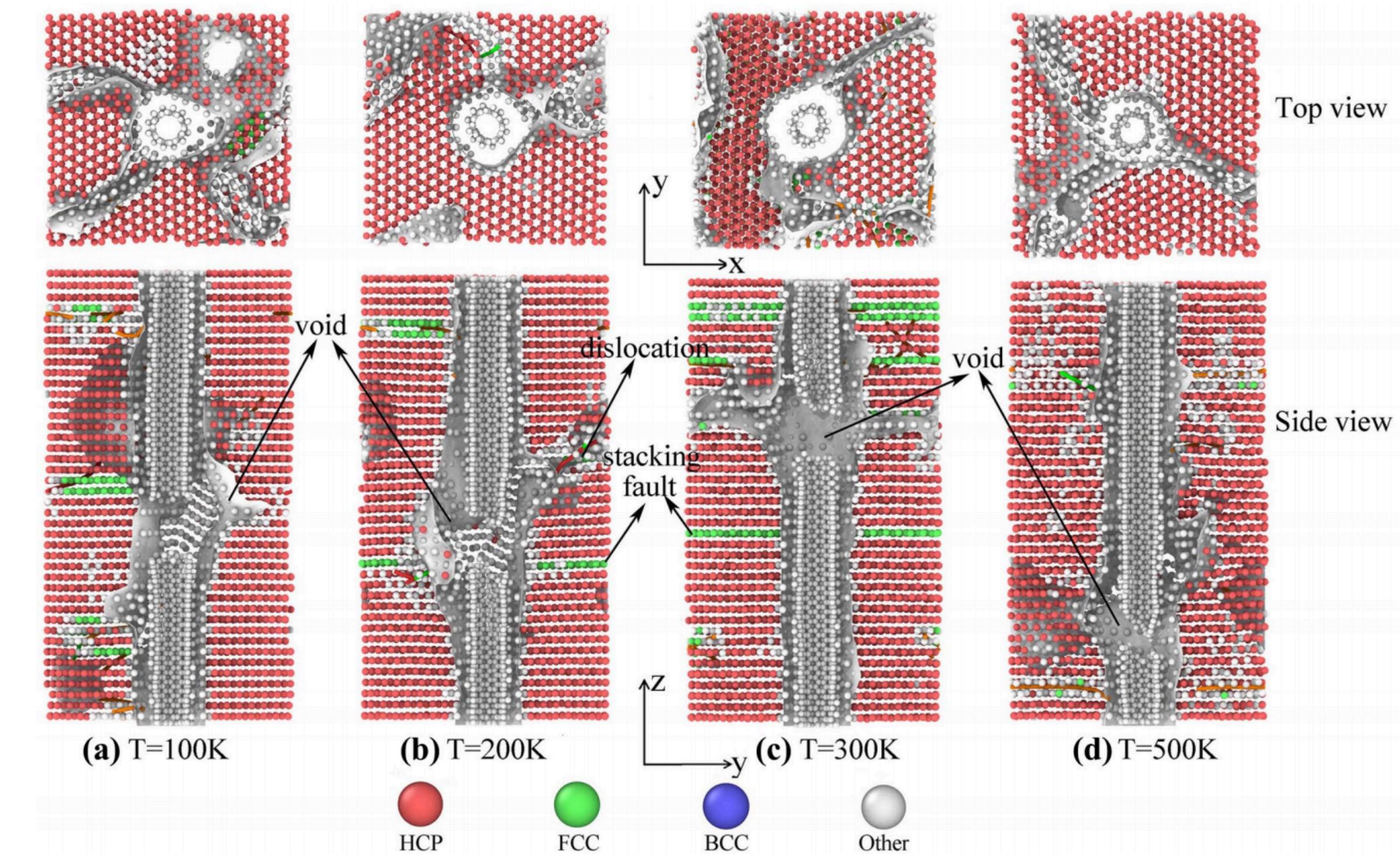


Fig. 8 Snapshots of tensile failure modes for Ni-CNT/Mg composites at different temperatures of **a** $T=100 \text{ K}$, **b** $T=200 \text{ K}$, **c** $T=300 \text{ K}$, and **d** $T=500 \text{ K}$