Introduction to Molecular Dynamics Simulations using LAMMPS

Dr. Ali Kerrache

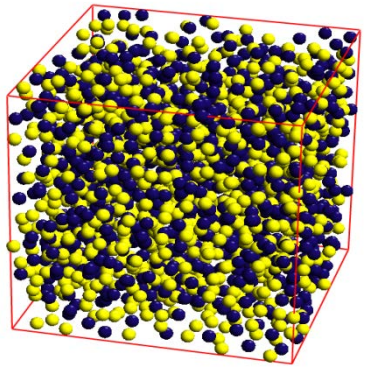
Faculty of Science, Univ. of Manitoba, Winnipeg
Compute Canada / WestGrid

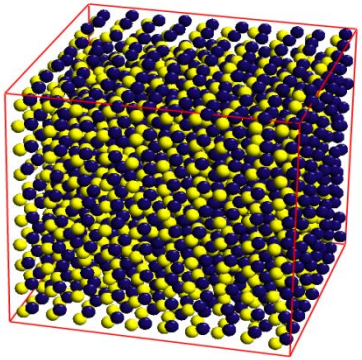
E-mail: ali.kerrache@umanitoba.ca

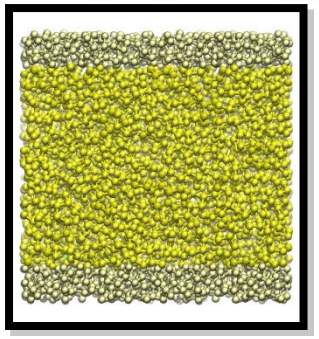
Home page: www.ali-kerrache.comeze.com

Introduction to MD Simulations using LAMMPS

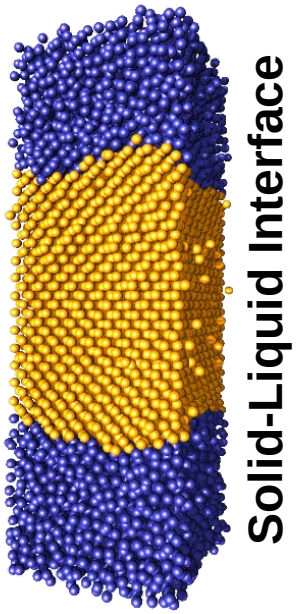
Outline:



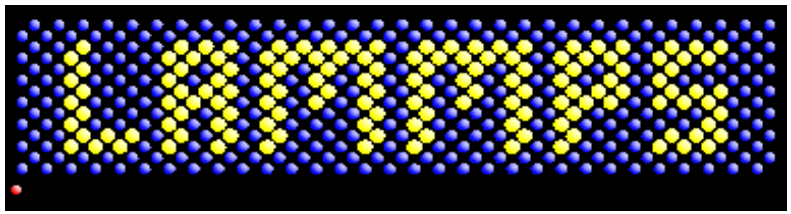




- **Part I**
  - Introduction
  - Molecular Dynamics Simulations
  - Examples of Simulations using Molecular Dynamics



- **Part II**
  - LAMMPS: Molecular Dynamics Simulator
  - Building and Running LAMMPS



- **Readings and references**

Introduction to Molecular Dynamics Simulations

Previous projects:

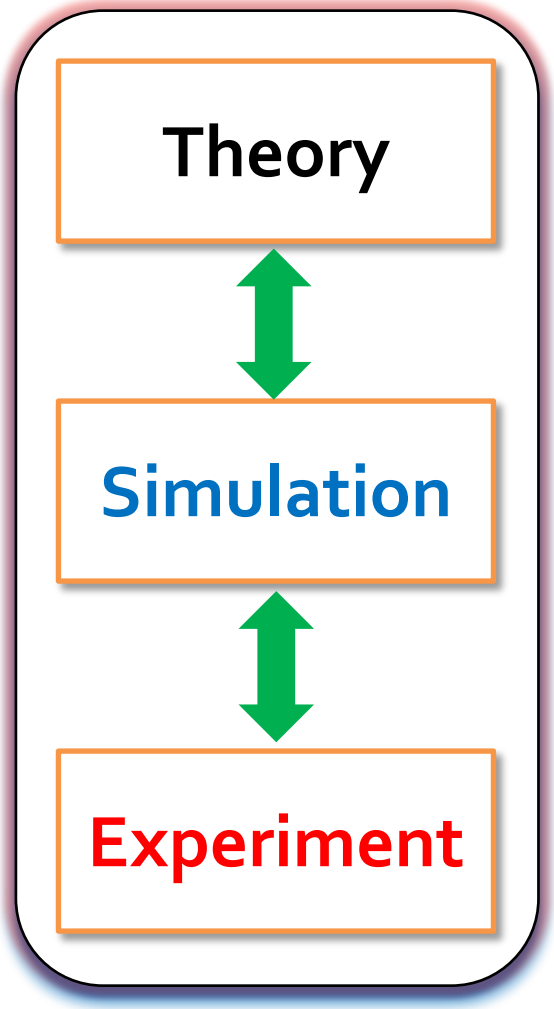
- Order – Disorder transitions in binary alloys
- Aging processes in super-cooled and amorphous silica
- Melting, Crystallization and Solid-Liquid Interfaces in Binary Metallic Systems: B2-AlNI, CuZr
- Effects of Shear Deformations on the Structure of Amorphous Silicon
- Crystallization and Solid-Liquid Interfaces in binary metallic systems: CuZr
- Structure of Glasses with three and five Oxides

Introduction to Molecular Dynamics Simulations

Why do we need simulations?

- **In most cases, experiments are:**
  - Difficult or impossible to perform
  - Too dangerous perform
  - Expensive and time consuming
  - Blind and too many parameters to control

- **Simulation is a powerful tool:**
  - can replace experiment
  - provoke experiment
  - explain and understand experiment
  - complete the theory



Computer Simulations

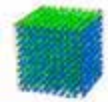
❑ Computer simulation:

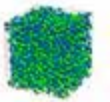
- ➢ a tool to get insight about the properties of materials at atomic or molecular level.
- ➢ used to predict and/or verify experiments.
- ➢ considered as a bridge between theory and experiment.
- ➢ provide a numerical solution when analytical ones are impossible.
- ➢ used to resolve the behavior of nature (the physical world surrounding us) on different time- and length-scales.

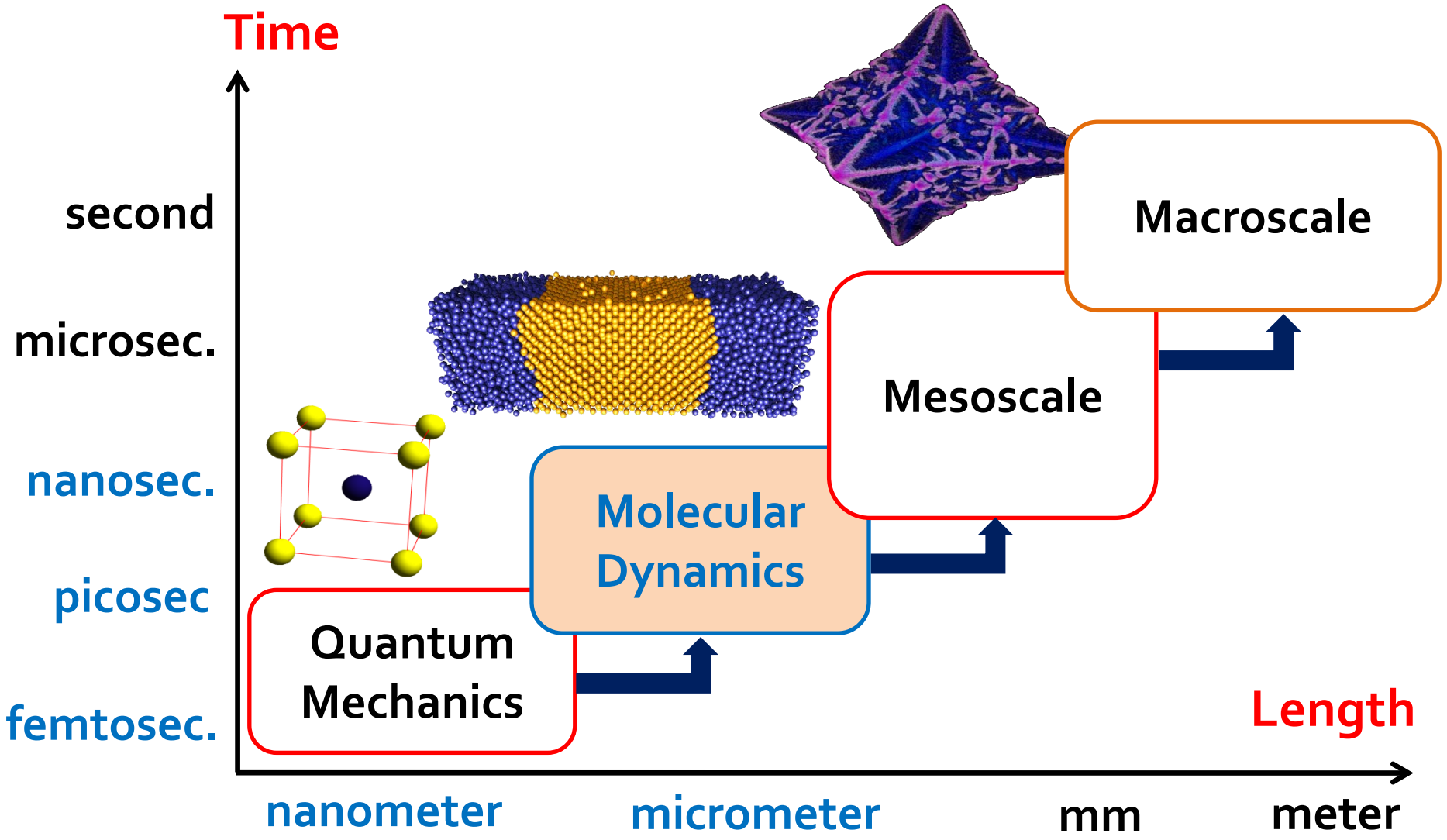
❑ Can be applied in, but not limited to:

- ✓ Physics and Applied Physics
- ✓ Chemistry
- ✓ Materials and Engineering

Simulations: Time and Length scales







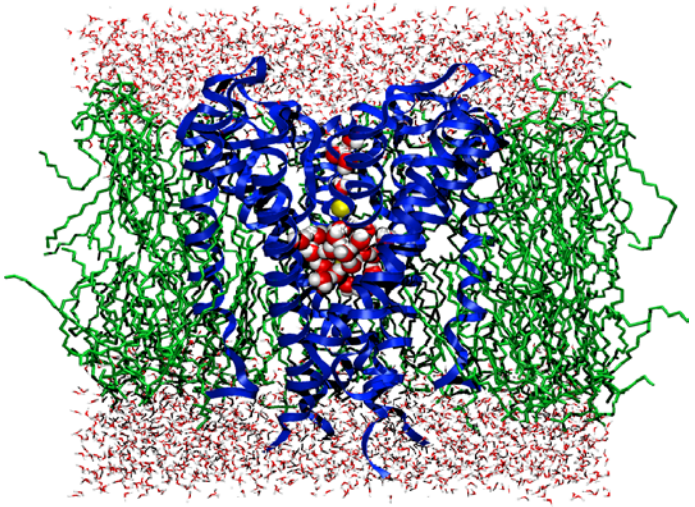
Molecular Dynamics simulations

❑ Newton equation

$$m_i \vec{a}_i = \vec{F}_i$$

$$\vec{F}_i = \sum_{j \neq i}^{N} \vec{f}_{ij}$$

➢ MD is the solution of the classical equations of motion for a system of N atoms or molecules in order to obtain the time evolution of the system.



➢ Applied to many-particle systems.

➢ Requires the definition of the potential to compute the forces.

$$\vec{f}_{ij} = -\vec{\nabla}_i V(r_{ij})$$

➢ Uses algorithms to integrate the equations of motion.

Compute the forces

- **Potential function:**

$$U(\mathrm{r}) = U_{bond}(\ldots) + U_{non-bond}(\ldots) + U_{ext}(\ldots)$$

- **Newton equation:**

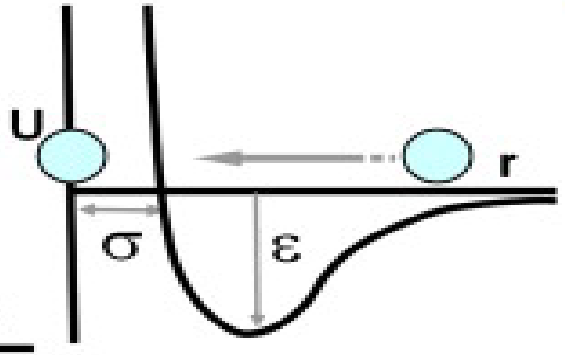
$$m_i \frac{d^2}{dt^2}\vec{x}_i = \vec{F}_i(\vec{x}_1, \ldots, \vec{x}_N) \qquad i = 1 \ldots N$$

- **Evaluate the forces acting on each particle:**

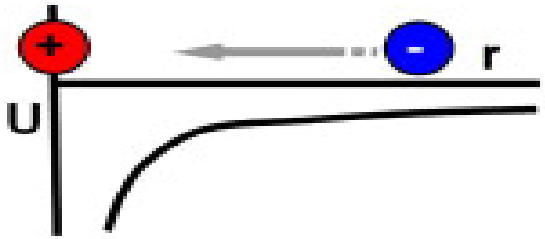
- **The force on an atom is determined by:** $\mathrm{F}_i = -\nabla U(\mathrm{r})$
  - $U(\mathrm{r})$: potential function
  - N : number of atoms in the system
  - r_{ij} : vector distance between atoms *i* and *j*

Example of potential function

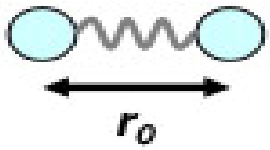
$$U = \sum_{i<j} \sum 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right]$$



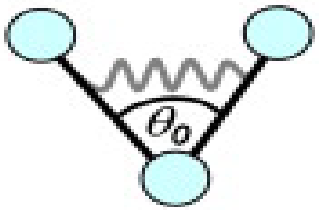
$$+ \sum_{i<j} \sum \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}$$



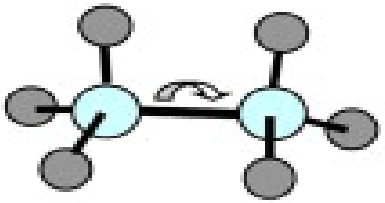
$$+ \sum_{bonds} \frac{1}{2} k_b (r - r_0)^2$$



$$+ \sum_{angles} \frac{1}{2} k_a (\theta - \theta_0)^2$$



$$+ \sum_{torsions} k_\phi [1 + \cos(n\phi - \delta)]$$



Verlet algorithm derivation

Taylor's expansions : position acceleration

$$r(t+\Delta t) = r(t) + \dot{r}(t)\Delta t + \tfrac{1}{2}\ddot{r}(t)\Delta t^2 + \tfrac{1}{6}\dddot{r}(t)\Delta t^3 + O(\Delta t^4) \qquad (I)$$

$$r(t-\Delta t) = r(t) - \dot{r}(t)\Delta t + \tfrac{1}{2}\ddot{r}(t)\Delta t^2 - \tfrac{1}{6}\dddot{r}(t)\Delta t^3 + O(\Delta t^4) \qquad (II)$$

Add (I) and (II) : velocity

$$r(t+\Delta t) + r(t-\Delta t) = 2r(t) + \ddot{r}(t)\Delta t^2 + O(\Delta t^4)$$

or :

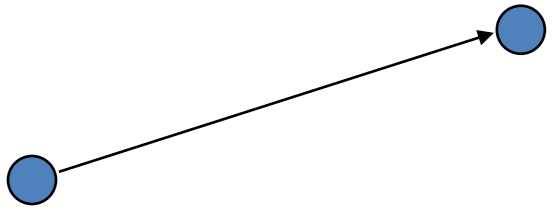
$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + f(t)\Delta t^2/m + O(\Delta t^4) \qquad (II)$$

Subtract (II) from (I) :

$$r(t+\Delta t) - r(t-\Delta t) = 2\dot{r}(t)\Delta t + O(\Delta t^3)$$

or :

$$v(t) = \big(r(t+\Delta t) - r(t-\Delta t)\big)/2\Delta t + O(\Delta t^2) \qquad (IV)$$



Verlet algorithm

- From the initial positions and velocities: $\mathbf{r}_i(t)$ $\mathbf{v}_i(t)$

$$\mathbf{a}(\mathbf{r}) = \frac{1}{m}\mathbf{F}(\mathbf{r}(t))$$

- Obtain the positions and velocities at: $t + \Delta t$

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

$$\mathbf{a}(t+\Delta t) = \frac{1}{m}F(\mathbf{r}(t+\Delta t))$$

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

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t+\Delta t/2) + \frac{1}{2}\mathbf{a}(t+\Delta t)\Delta t$$

- Velocity calculated explicitly
- Possible to control the temperature
- Stable in long simulation
- Most used algorithm

Leap-Frog algorithm

$$v(t+\frac{\Delta t}{2}) = v(t-\frac{\Delta t}{2}) + \frac{F(t)}{m}\Delta t$$

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

Predictor-Corrector algorithms

- **Predictor step:**
  - from the initial $\mathbf{r}_i(t)$, $\mathbf{v}_i(t)$ ➔ $\mathbf{a}(\mathbf{r}) = \frac{1}{m}\mathbf{F}(\mathbf{r}(t))$
  - predict $\mathbf{r}_i(t+\Delta t)$, $\mathbf{v}_i(t+\Delta t)$ using Taylor's series

$$\mathbf{r}^P(t+\Delta t) \cong \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{\mathbf{a}(t)}{2}\Delta t^2$$

$$\mathbf{v}^P(t+\Delta t) \cong \mathbf{v}(t) + \mathbf{a}(t)\Delta t$$

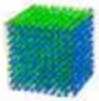
$$\mathbf{a}^P(t+\Delta t) \cong \mathbf{a}(t) + \mathbf{r}^{iii}(t)\Delta t$$

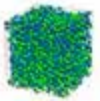
$\mathbf{r}^{iii}$: 3rd order derivatives

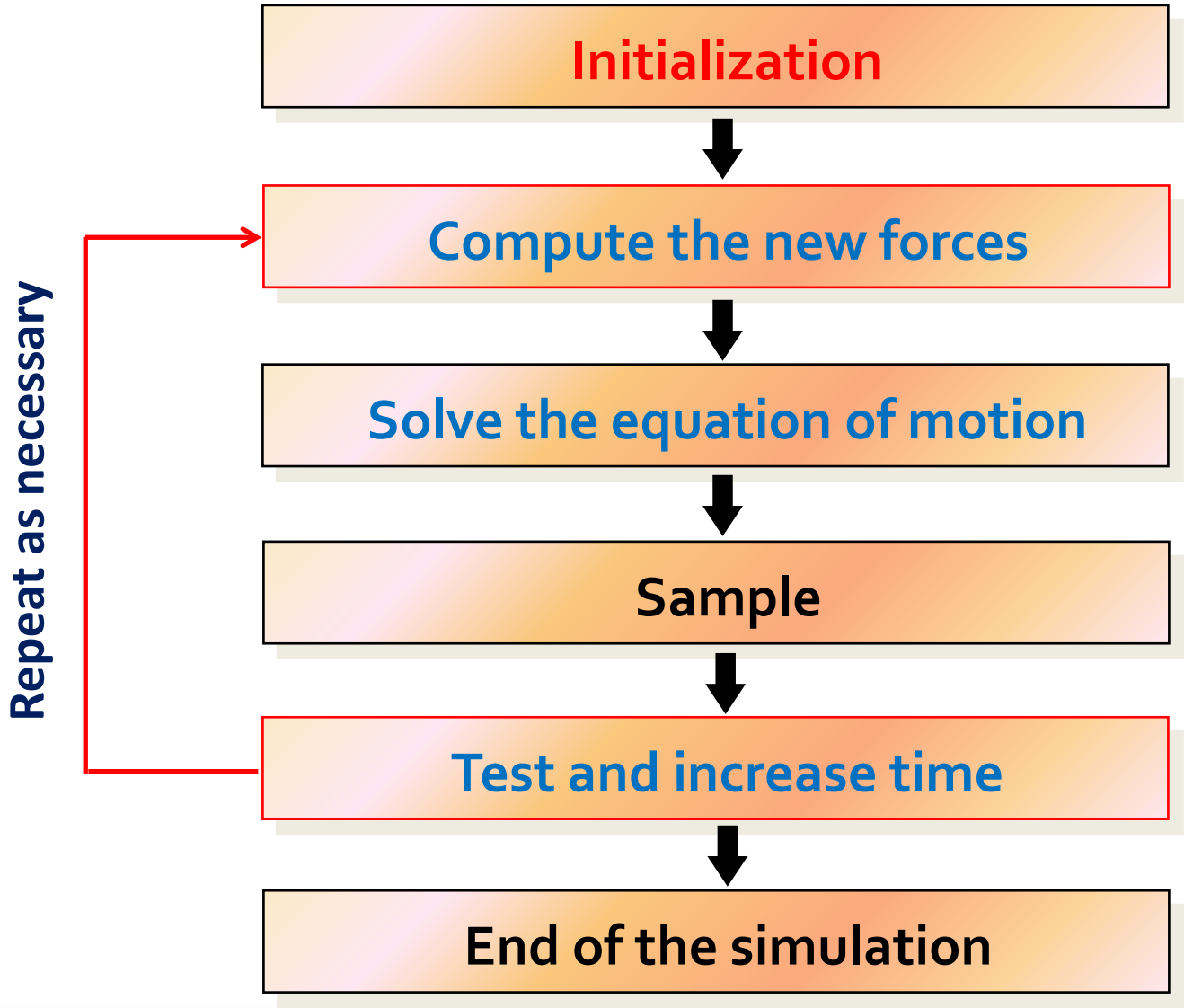
- **Corrector step:**
  - get corrected acceleration: $\mathbf{a}^C(\mathbf{r}) = \frac{\mathbf{F}(\mathbf{r}^P(t+\Delta t))}{m}$
  - using error in acceleration: $\Delta\mathbf{a}(t+\Delta t) \cong \mathbf{a}^C(t+\Delta t) - \mathbf{a}^P(t+\Delta t)$
  - correct the positions: $\mathbf{r}(t+\Delta t) \cong \mathbf{r}^P(t+\Delta t) + C_0\frac{\Delta t^2}{2}\Delta\mathbf{a}(t+\Delta t)$
  - correct the velocities: $\mathbf{v}(t+\Delta t) \cong \mathbf{v}^P(t+\Delta t) + C_1\Delta t\Delta\mathbf{a}(t+\Delta t)$

C_n : constants depending accuracy

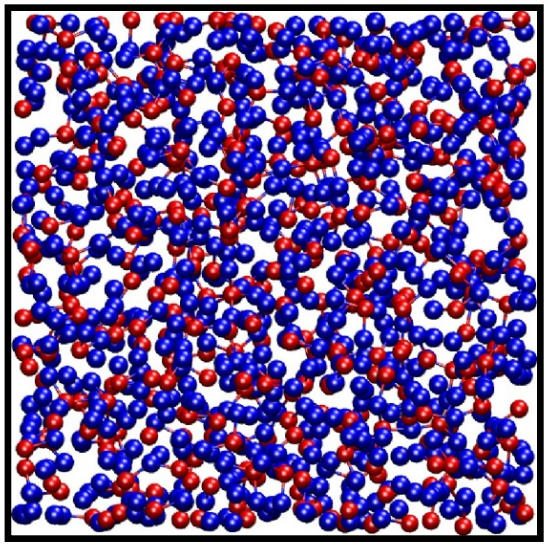
Basic Molecular Dynamics Program





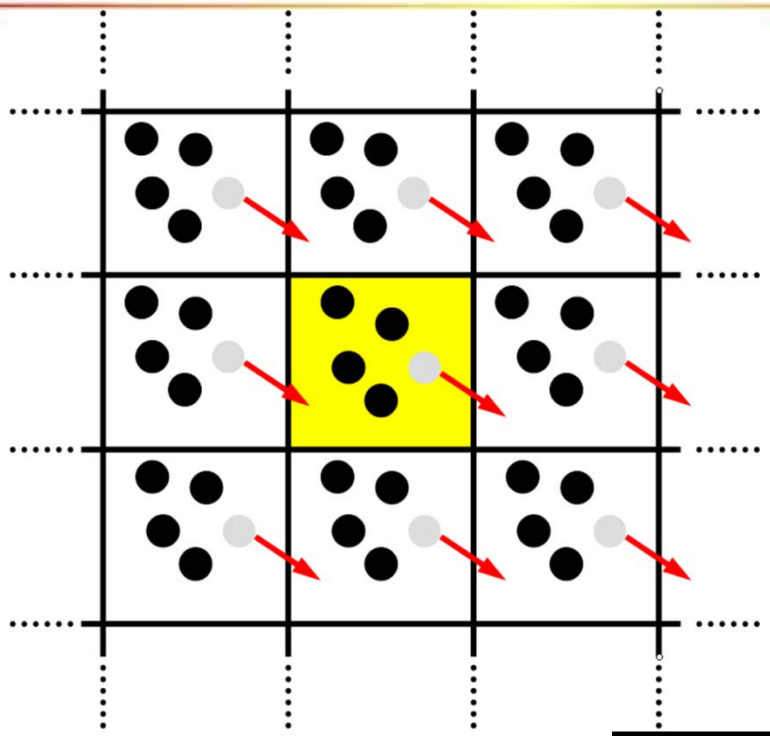


Simulation Setup

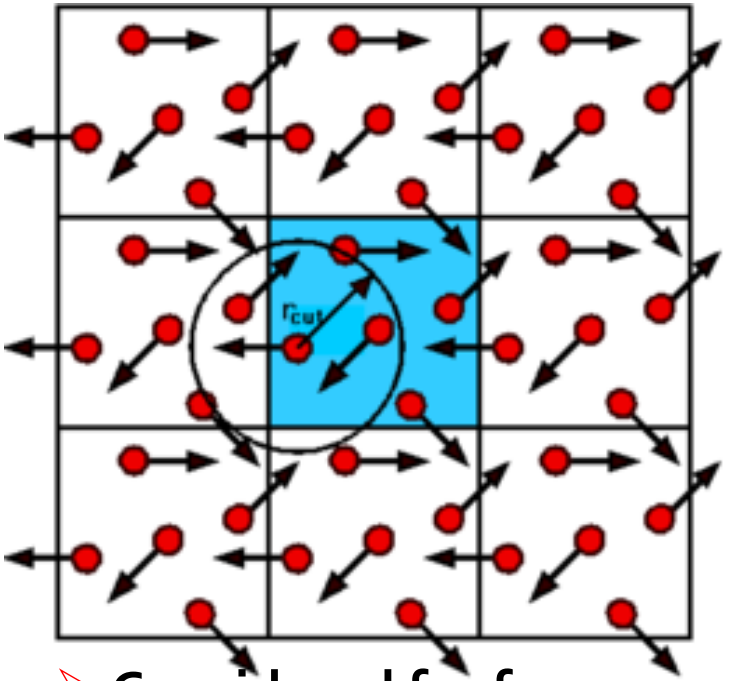


- Starting configuration:
  - Atomic positions (x,y,z)
  - density …
  - ….
- Initial velocities:
  - Depend on temperature
- periodic boundary conditions:
  - required to simulate bulk properties
- set the appropriate potential:
  - Depend of the system to simulate
- set the appropriate time step: should be short (order of 1fs).
- set the temperature control:
  - define the thermodynamic ensemble.

Periodic Boundary Conditions

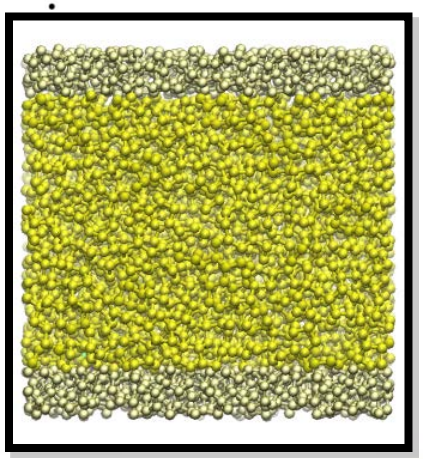


- Duplicate the simulation box in all directions (x, y and z)
- An atom moving out of boundary comes from the other side



- Considered for force calculations

- PBC in x and y direction
- **Walls:** fixed boundaries in z direction.

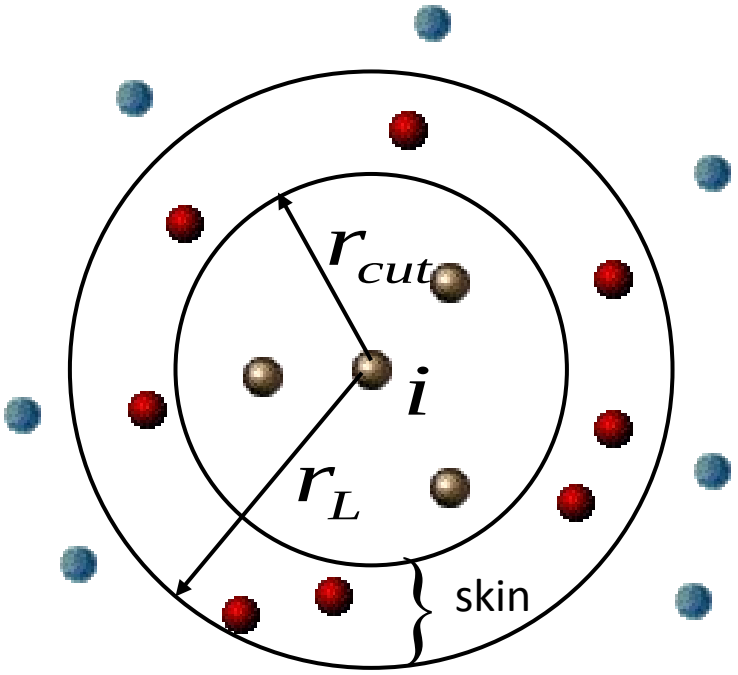


Use neighbor list to reduce execution time

- Evaluate forces is time consuming.
- Pair potential calculation: $\propto O(\mathrm{N}^2)$
- Atom moves $< 0.2\,\mathring{\mathrm{A}}$ per time step
- Not necessary to include all the possible pairs

❑ Solution: Verlet neighbor list

- Containing all neighbors of each atom within: r_L
- Update every N_L steps



$$r_L - r_{cut} > \frac{N_L \bar{\mathbf{v}} \Delta t}{2}$$

Thermodynamic ensembles

- **Ensembles:**
  - **NVE – micro-canonical ensemble**
  - **NVT – canonical ensemble**
  - **NPT – grand-canonical ensemble**

Each ensemble is used for a specific simulation.

- **Temperature control**
  - **Berendsen thermostat (velocity rescaling)**
  - **Andersen thermostat**
  - **Nose-Hoover chain**

Choose the ensemble that best fits your system and the properties you want to simulate then start the simulations

- **Pressure control**
  - **Berendsen volume rescaling**
  - **Andersen piston**

What can be computed from MD simulations

- **Statistical Mechanics**

  - **The prime purpose of MD is to sample the *phase space* of the statistical mechanics ensemble.**
  - **Most physical properties can be related the atomic trajectories and obtained as average as a function of time.**

  - **Structural properties:**
    obtained from spatial correlation functions e.g. radial distribution function.

  - **Dynamical Properties:**
    Time dependent properties (transport coefficients) obtained via temporal correlation functions e.g. velocity autocorrelation function.

Thermodynamic properties

- **Kinetic Energy**

$$\langle K.E. \rangle = \left\langle \frac{1}{2} \sum_{i}^{N} m_i v_i^2 \right\rangle$$

- **Temperature**

$$T = \frac{2}{3Nk_B} \rangle K.E. \langle$$

- **Configuration Energy**

$$U_c = \left\langle \sum_{i} \sum_{j>i}^{N} V(r_{ij}) \right\rangle$$

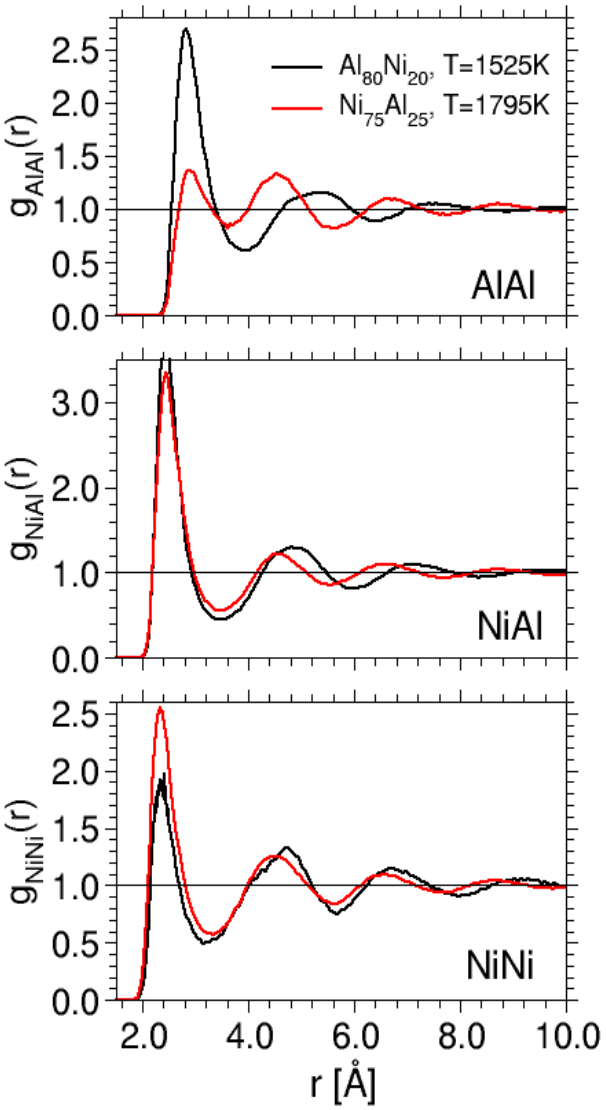
- **Pressure**

$$PV = Nk_B T - \frac{1}{3} \left\langle \sum_{i=1}^{N-1} \sum_{j>i}^{N} \vec{r}_{ij} \cdot \vec{f}_{ij} \right\rangle$$

- **Specific Heat**

$$\left\langle \delta(U_c)^2 \right\rangle_{NVE} = \frac{3}{2} Nk_B^2 T^2 \left(1 - \frac{3Nk_B}{2C_v}\right)$$

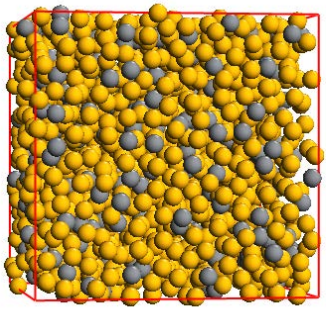
Structural properties



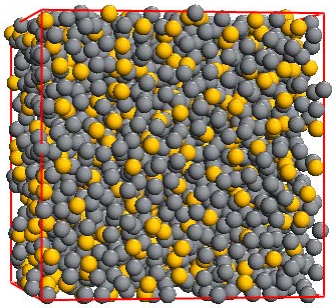
❖ **Radial Distribution Function**

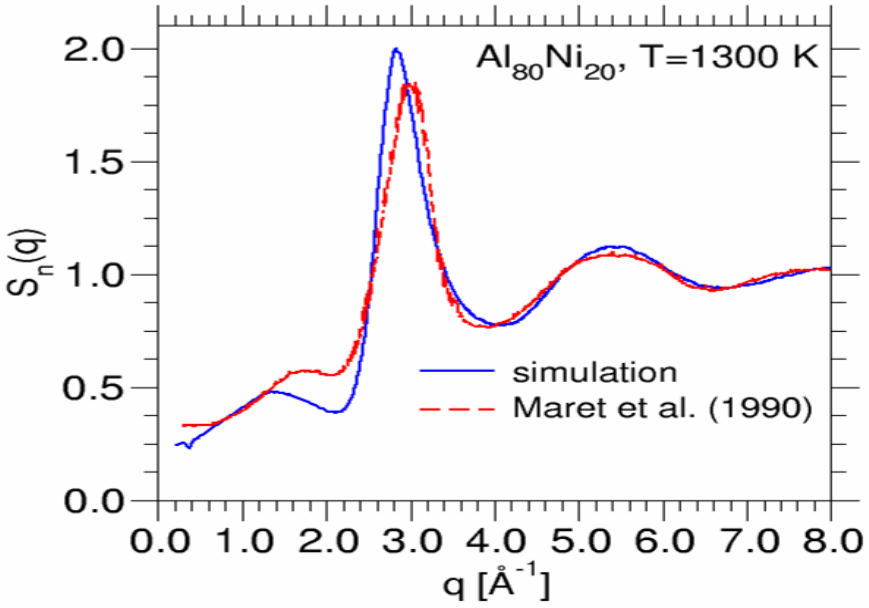
$$g(r) = \frac{\langle n(r) \rangle}{4\pi\rho r^2 \Delta r} = \frac{V}{N^2} \left\langle \sum_i \sum_{j \neq i}^{N} \delta(r - r_{ij}) \right\rangle$$

$Al_{25}Ni_{75}$:



$Al_{80}Ni_{20}$:



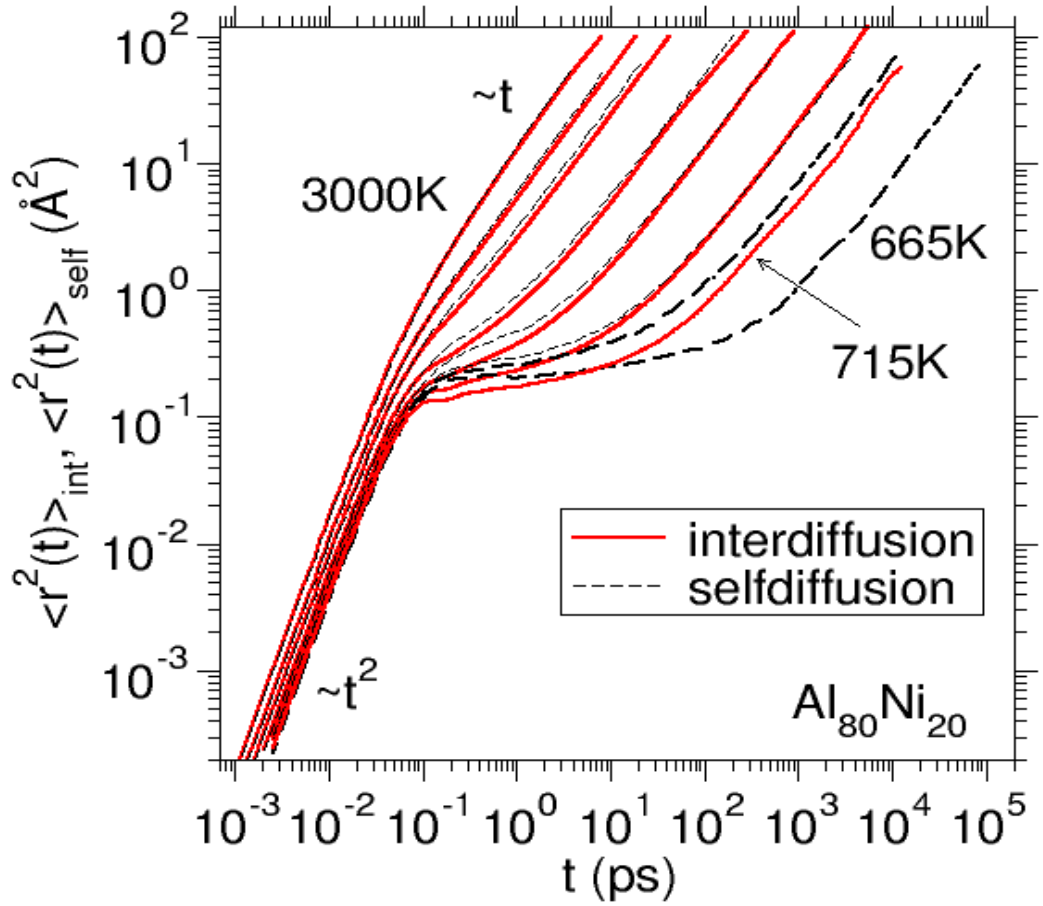


$$S(k) = 1 + 4\pi\rho \int_0^\infty \frac{\sin(kr)}{kr} (g(r) - 1)\, r^2 dr$$

➢ **Structure Factor**

Dynamical properties

Mean Square Displacement (Einstein relation)



$$2Dt = \frac{1}{3}\left\langle |\,\mathbf{r}_i(t) - \mathbf{r}_i(0)\,|^2 \right\rangle$$

$$\mathrm{MSD} = c_{\mathrm{Al}}\left\langle (\vec{r}_{\mathrm{s,\,Ni}}(t) - \vec{r}_{\mathrm{s,\,Ni}}(0))^2 \right\rangle + c_{\mathrm{Ni}}\left\langle (\vec{r}_{\mathrm{s,\,Al}}(t) - \vec{r}_{\mathrm{s,\,Al}}(0))^2 \right\rangle$$

Diffusion constants

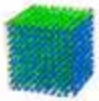
$$D = \lim_{t\to\infty} \frac{1}{N} \sum_{i=1}^{N} \frac{\left\langle (r(t) - r(0))^2 \right\rangle}{6t}$$

Recommended textbooks

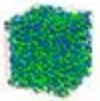
- *The Art of Molecular Dynamics Simulation, D.C. Rapaport, Camb. Univ. Press (2004)*
- *Understanding Molecular Simulation, D. Frenkel and B. Smit, Academic Press (2002).*
- *Computer Simulation of Liquids, M.P. Allen and D.J. Tildesley, Oxford (1989).*
- *Theory of Simple Liquids, J.-P. Hansen and I.R. McDonald, Academic Press (1986).*
- *Classical Mechanics, H. Goldstein, Addison Wesley (1980)*

How could we perform MD Simulation?

- **Open source: free access**
  - ✓ **LAMMPS:** http://lammps.sandia.gov/index.html
  - ✓ **DL_POLY:** http://www.scd.stfc.ac.uk/SCD/44516.aspx
  - ✓ **CP2K:** https://www.cp2k.org/about
  - ✓ **NAMD:** http://www.ks.uiuc.edu/Research/namd/
  - ✓ **GROMACS:** http://www.gromacs.org/
  - ✓ ….
- **Commercial software:**
  - ✓ **Amber:** http://ambermd.org/
- **Home made or private codes:**
  - ✓ **C, C++**
  - ✓ **Fortran, … etc**



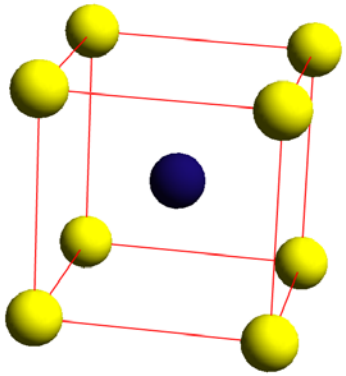
Molecular Dynamics Simulation Results



Melting, Crystallization and solid-liquid interfaces in binary metallic systems

Binary Metallic Systems: B2-$Al_{50}Ni_{50}$

Why B2-$Al_{50}Ni_{50}$?



- B2-$Al_{50}Ni_{50}$: prototype of binary ordered metals
- simulations of interfacial growth in binary systems rare
- growth kinetics of binary metals: diffusion limited?
- crystal growth slower than in one-component metals
- understand crystal growth of alloys on microscopic level

Questions:

Wilson H.A. Philos. Mag. , **50** (1900) 238.
Frenkel J., Phys. Z. Sowjetunion, **1** (1932) 498.

- crystal growth & accurate estimation of T_m?
- solid-liquid interface velocity from interface motion?
- kinetic coefficients and their anisotropy?
- solid-liquid interface motion controlled by mass diffusion?
- solid-liquid coexistence, interface structure?
- how to distinguish between solid-like & liquid-like particles?

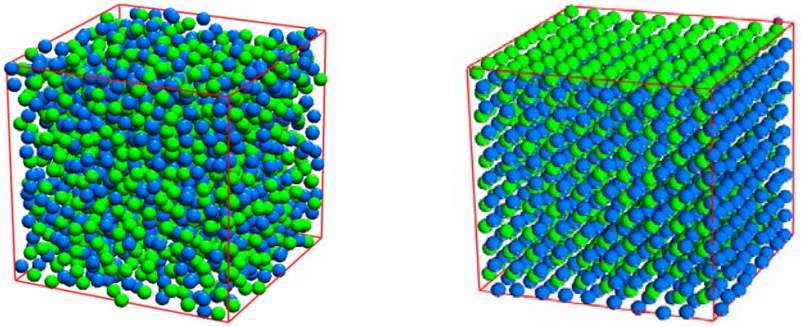
Melting, Crystallization & Solid-Liquid Interfaces

- solve Newton's equation of motion for system of *N* particles:

- **velocity Verlet algorithm (time step = 1 fs)**
- **NPT ensemble:**
  constant pressure (Anderson algorithm): p = 0
  constant temperature: stochastic heat bath
- **periodic boundary conditions in all directions**

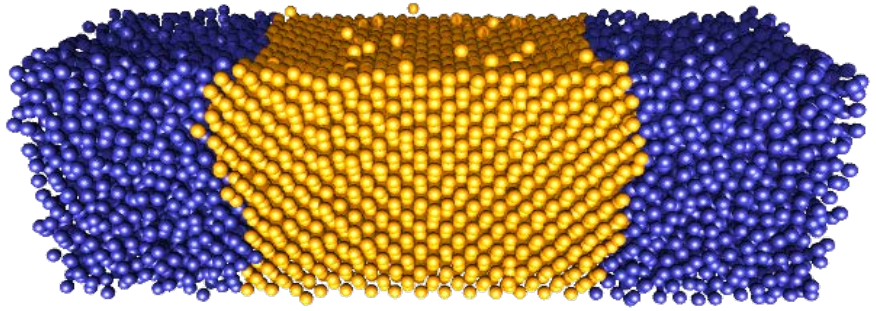
Allen M.P. and Tildeslay D.J., Computer simulation of liquids, 1987
Anderson H.C., JCP **72** (1980) 2384

MD of pure systems



- lattice properties
- T dependence of density
- Structural quantities
- Self-diffusion constant

MD of inhomogeneous systems



- melting temperature T_m
- kinetic coefficients & their anisotropy
- solid-melt interface structure
- crystal growth

Simulation details

- **Binary metallic mixtures - simple: Lenard-Jonnes potential**
  - better: EAM

- **EAM** potential:

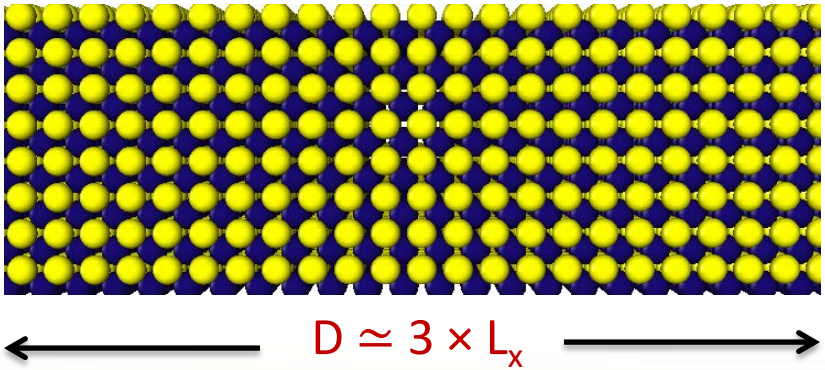
$$U_{\text{pot}} = \frac{1}{2}\sum_{k,l} u(r_{kl}) + \sum_{k} F(\bar{\rho}_k) \qquad \bar{\rho}_k = \sum_{l \neq k} \rho_l(r_{kl})$$

- two body interactions
- many body interactions (e-density)
- fitting to both experimental and *ab-initio* data

- reproduces the lattice properties & point defects
- structure and dynamics of AlNi melts:

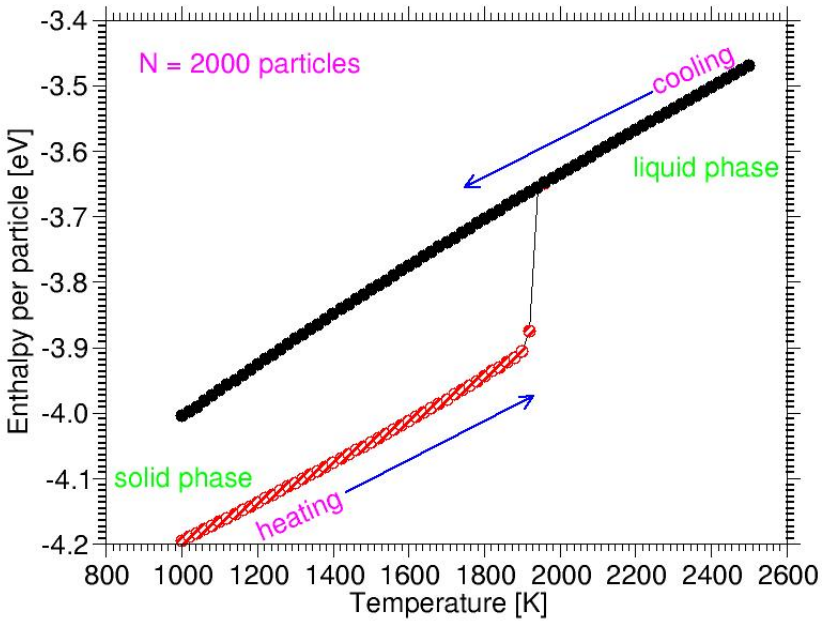
Y. Mishin *et al.*, PRB **65**, (2002) 224114.
J. Horbach *et al.*, PRB **75**, (2007) 174304.

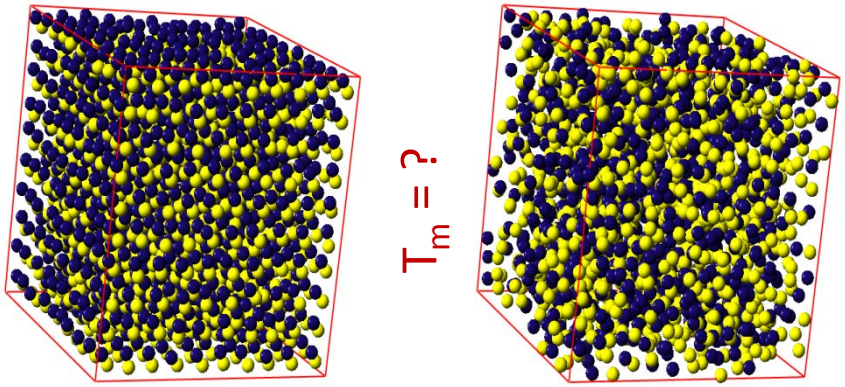
➢ Solid and liquid properties:
2000 particles ($L_x = L_y = L_z = 24.6$ Å)
➢ Solid-liquid interfaces (N particles):
$N_{Al} = N_{ni} \Rightarrow D = L_z \simeq 3 \times L_x \simeq 3 \times L_y$
10386 and 12672 particles



Melting and Crystallization

- How to go from crystal to melt & from melt to crystal?
  - start from B2 phase: equilibration at 1000 K
  - try to melt the crystal: heating process
  - cool down the melt: cooling process



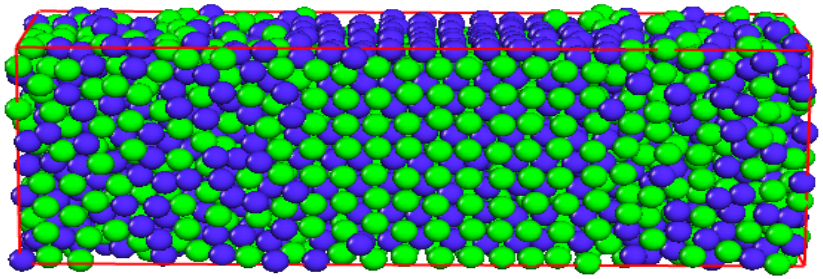


- How to solve this problem?

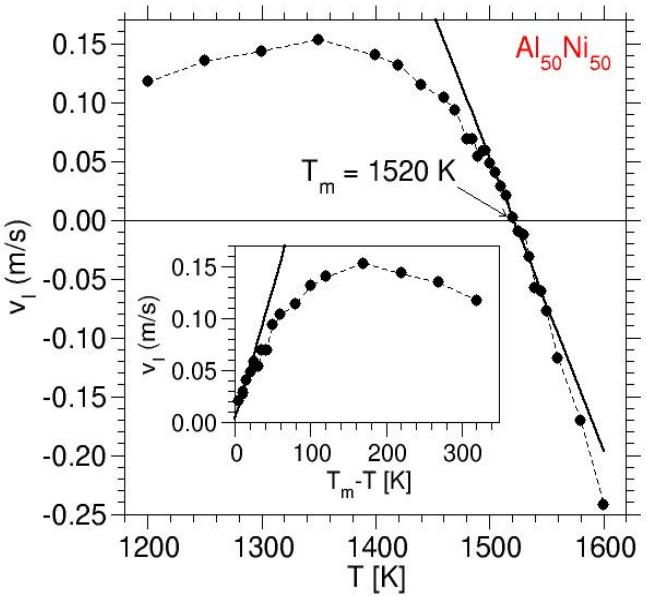
- binary alloys: glass formers in general
- crystallization process too slow
- brute force method: not appropriate to estimate T_M

Solid-Liquid Interface

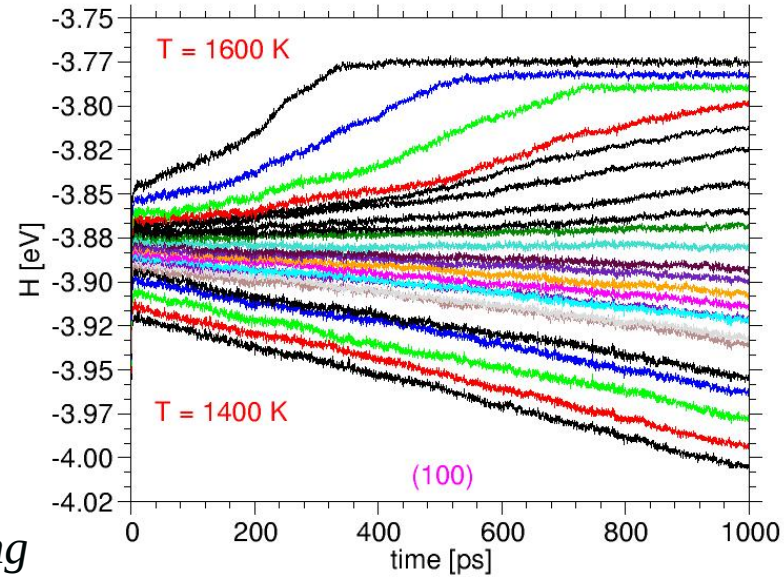
- Equilibrate a crystal (NPT)
- Fix the particles in the middle of the box
- Heat away the two other regions
- Quench at the target temperature



The Melting temperature T_M from motion of interface?

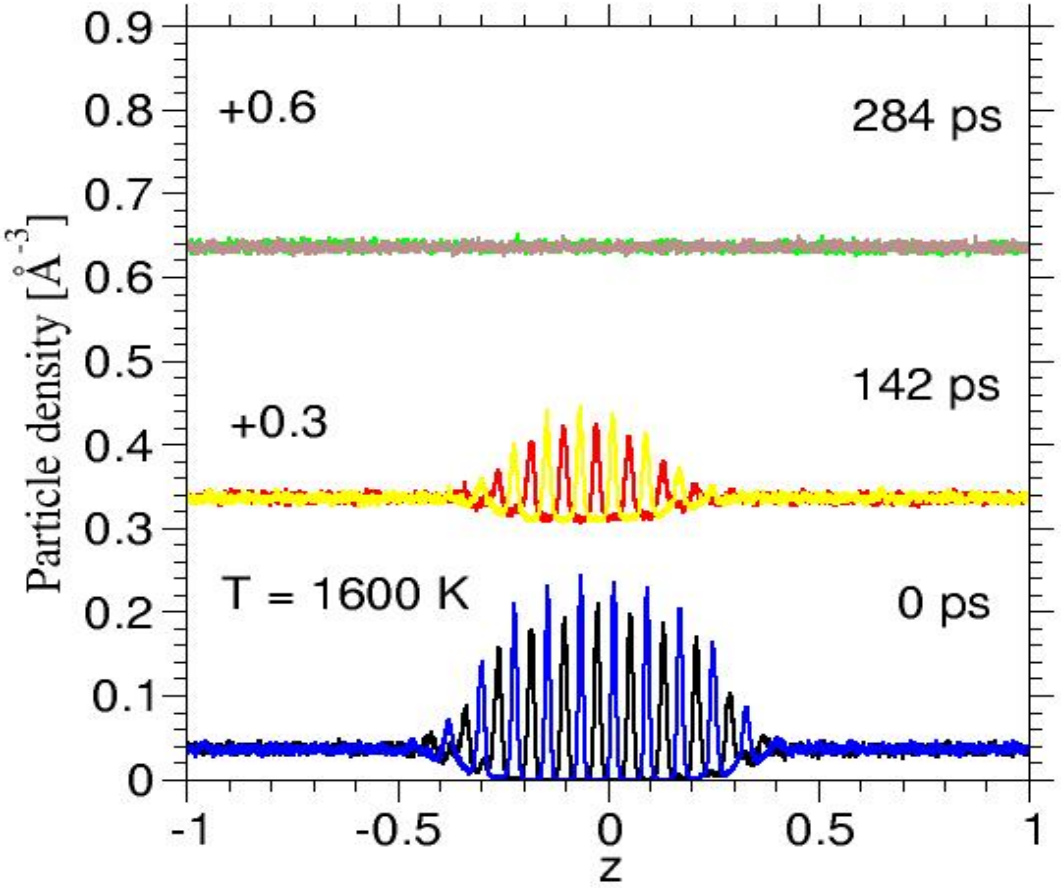


A. Kerrache et al. EPL 2008

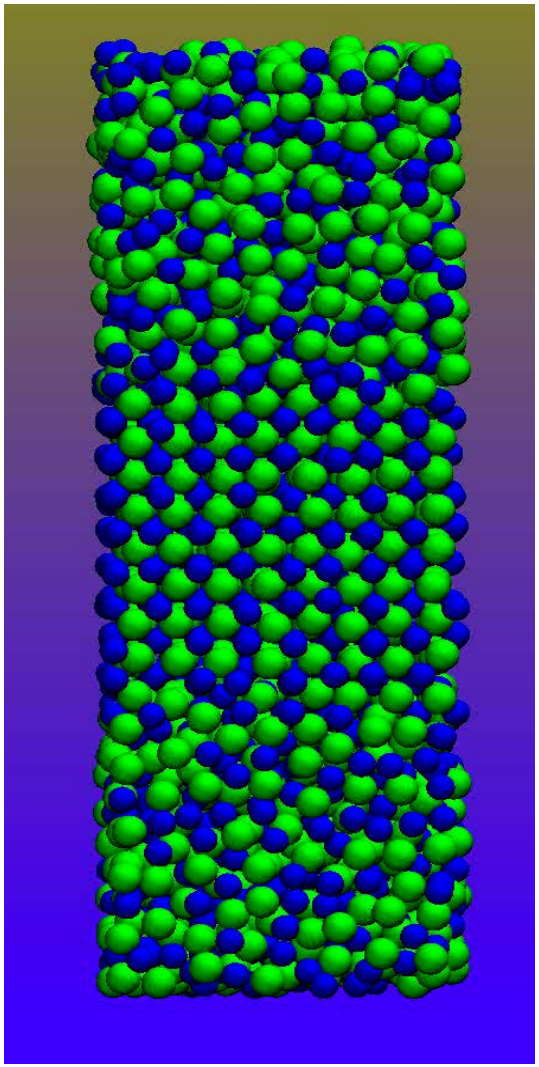


$T > T_M$: *melting*
$T = T_M$: *coexistence*
$T < T_M$: *crystallization*

Melting



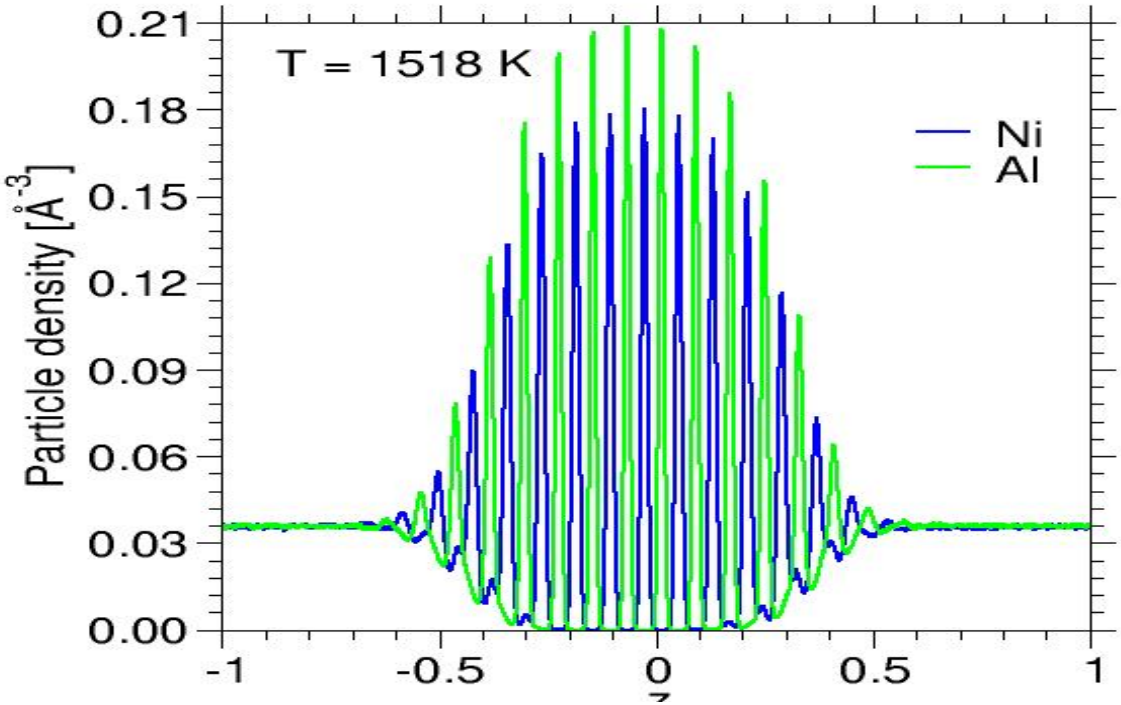
Particle density along the solid-liquid interface

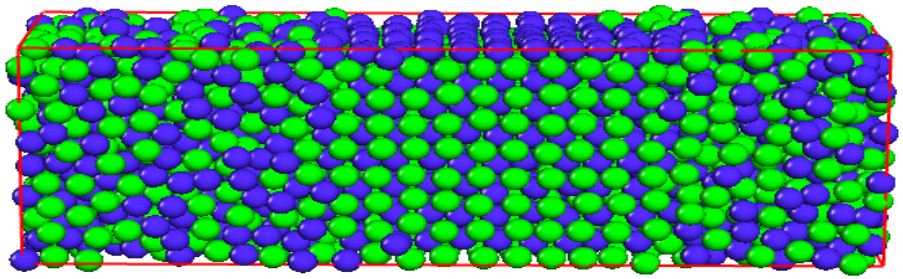


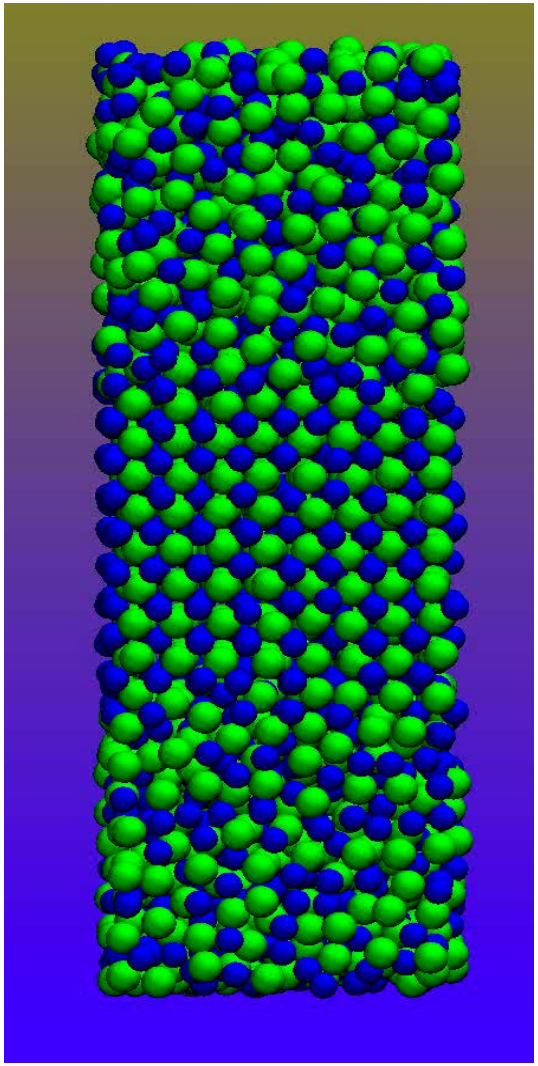
- Melting

Coexistence

Particle density along the solid-liquid interface

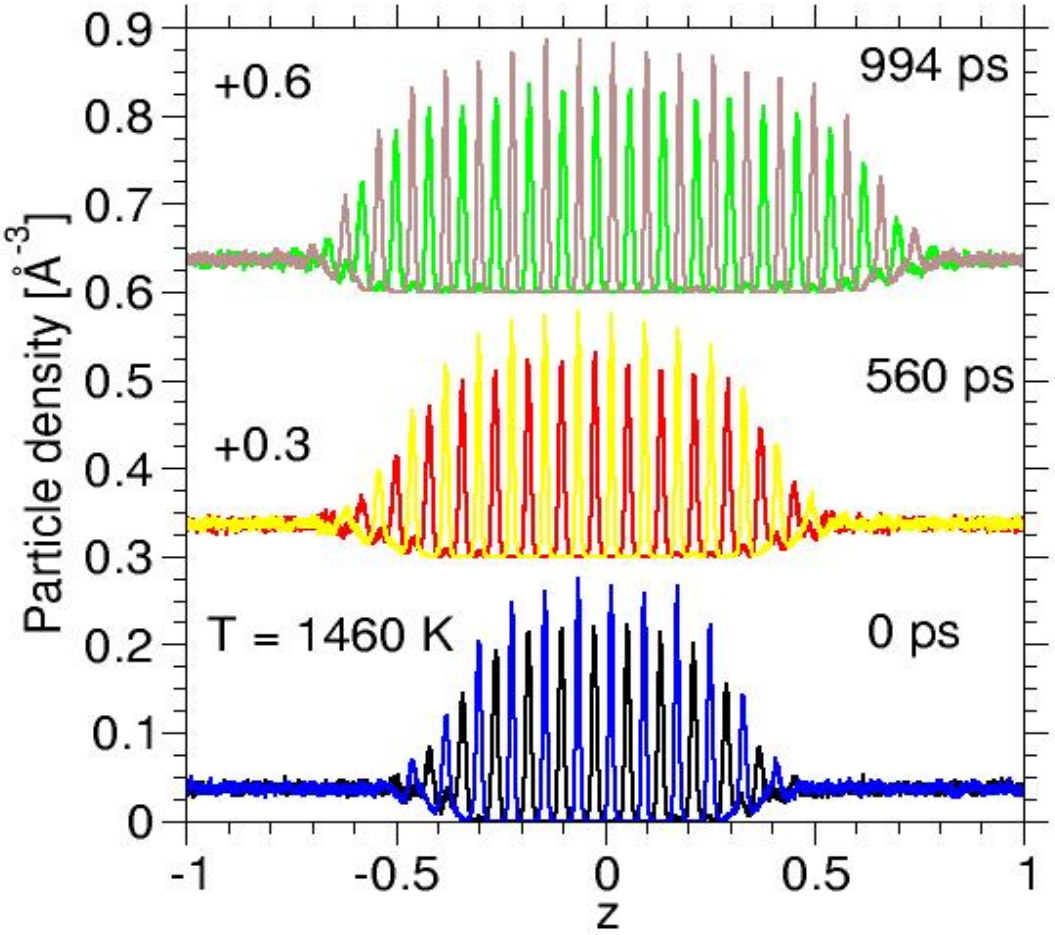




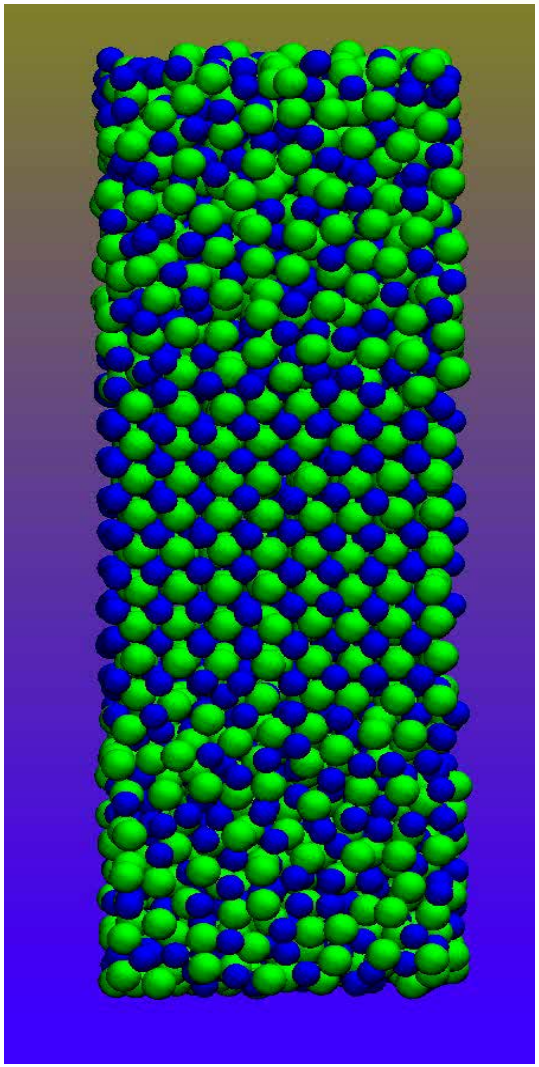


❑ **Coexistence**

Crystallization



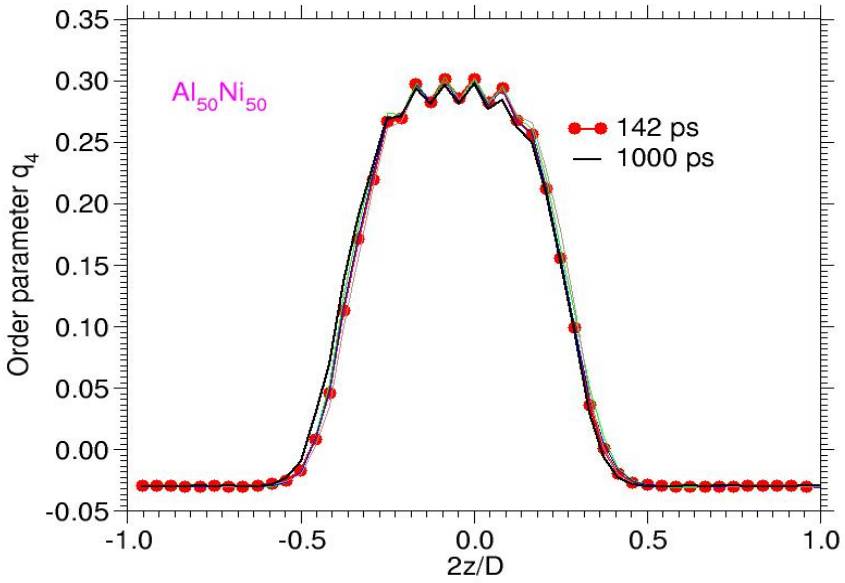
Particle density along the solid-liquid interface



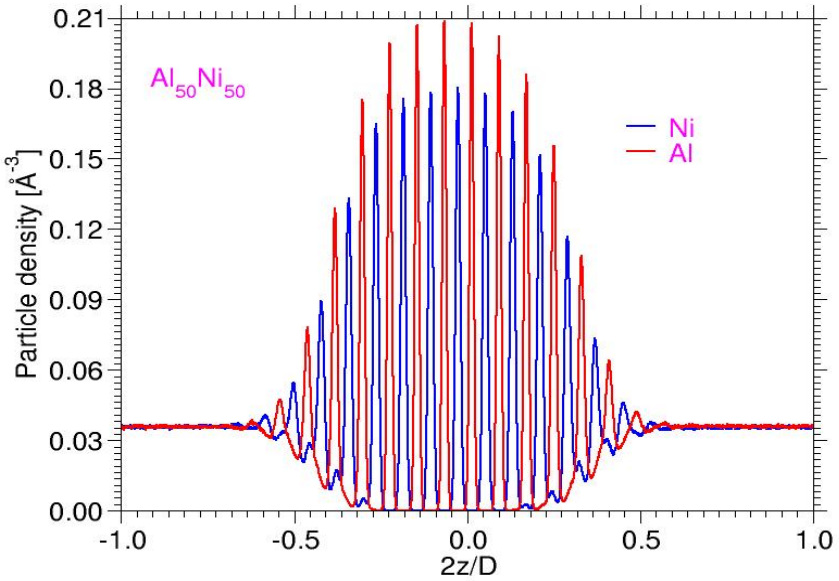
- Crystallization

Order Parameter and density at Coexistence

Bond order parameter profile for different times



partial particle density profile for different times



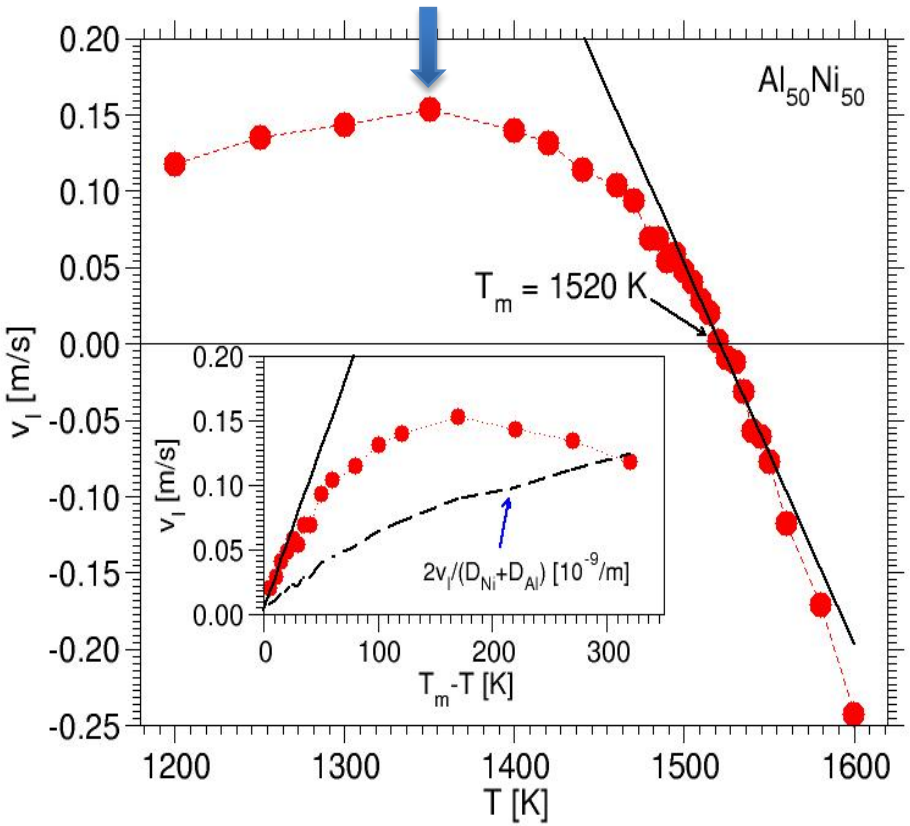
$$q_n = \left\langle \frac{1}{N} \sum_{i,j,k} \cos\left(n\theta_{xy}(i,j,k)\right) \right\rangle$$

$$n = 1, 2, \ldots, 6$$

I,j and k: indices for nearest neighbors, θ(I,j,k): bond angle formed by I, j and k atoms, rojected to xy plane, N_z: number of nearest neighbors

Constant density in the liquid region.
Solid-liquid interface over several layers.
Pronounced chemical ordering in the solid region:
Mass transport required for crystal growth.

Diffusion Limited Growth



Solid-liquid interface velocity as a function of temperature
Inset: as a function of under-cooling

Why the solid-liquid interface velocity presents a maximum?

Maximum of 0.15 m/s at 180 K
Interface velocity divided by the average self diffusion constant

Maximum due to decreasing of diffusion constant
Linear regime only up to 30 K of under-cooling

What about the mass transport across the solid-liquid interface?

Mass transport across the interface

- Order parameter to distinguish solid and liquid locally

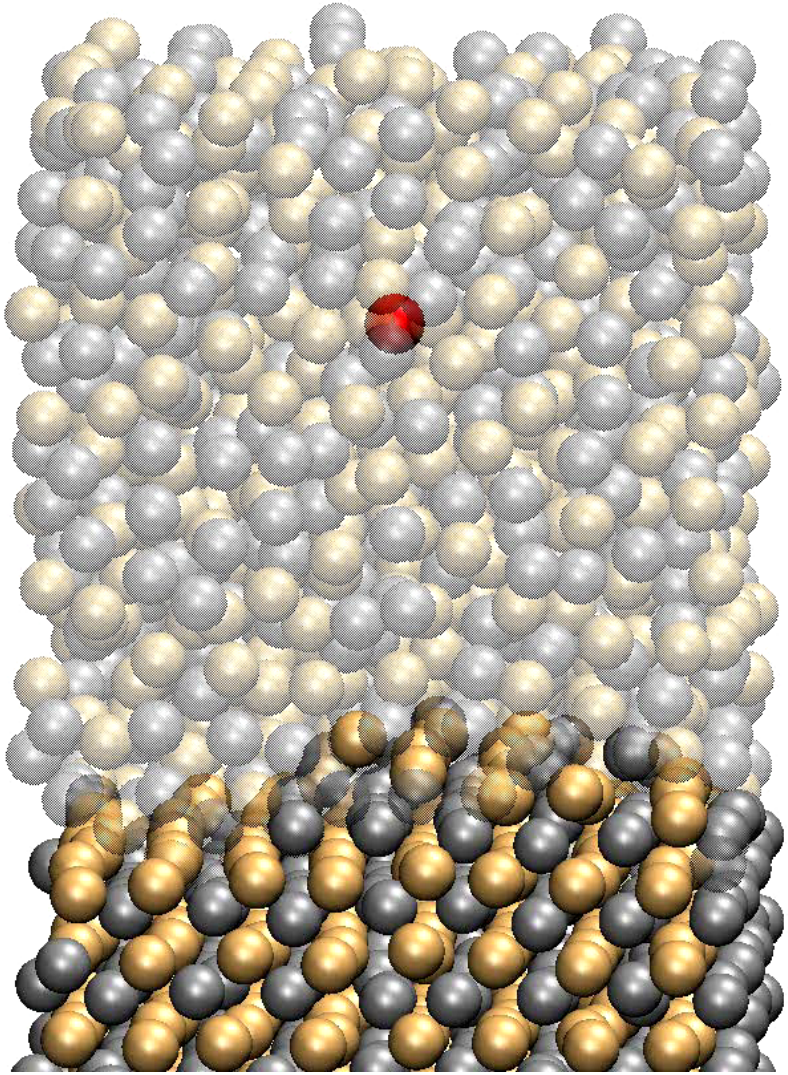
- compute the particle density and mass density profile

- Order parameter profile

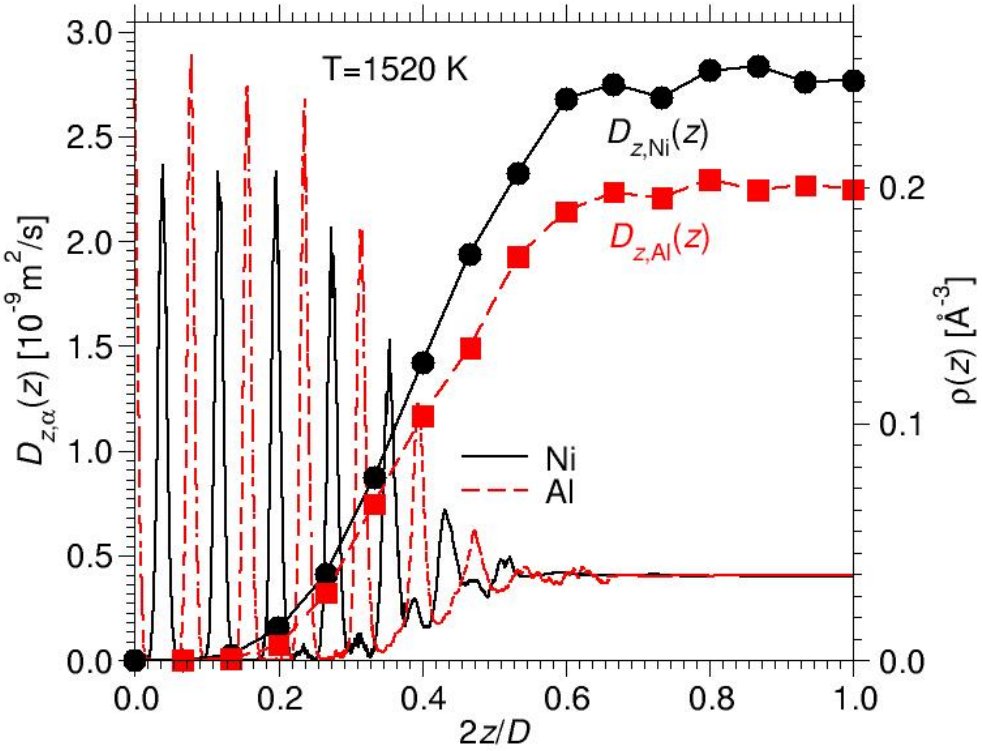
- Number of solid-like particles

- Solid-liquid interface velocities from the number of solid-like particles

- Diffusion along the interface



Mass transport across the interface



Mass transport and particle density across the solid-liquid interface

$$D_{z_s,\alpha}(z_s) = \lim_{t\to\infty} \frac{1}{N_s} \sum_{i_s=1}^{N_s} \frac{\left\langle (z_{i_s}(t) - z_{i_s}(0))^2 \right\rangle}{2t}$$

The diffusion constants decrease when we cross the solid-liquid interface

Wilson-Frenkel theory:
activated process controlled by mass diffusion in the liquid phase

Wilson H.A. Philos. Mag. , **50** (1900) 238.
Frenkel J., Phys. Z. Sowjetunion, **1** (1932) 498.
A. Kerrache et al. EPL, 2008.

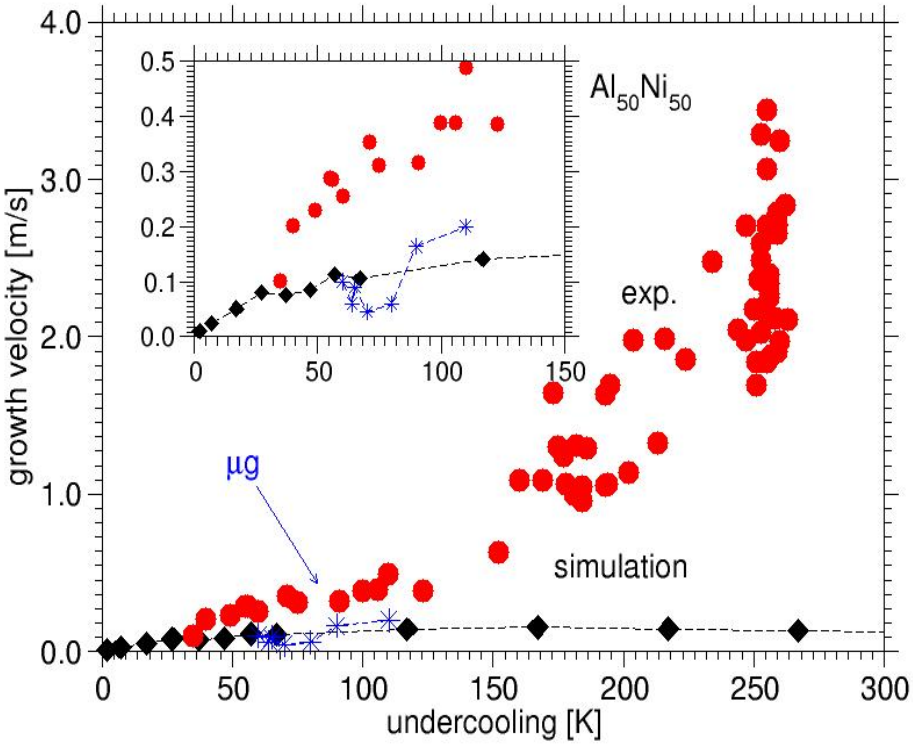
Crystal growth: controlled by mass transport in the liquid phase and solid-liquid interface

Experimental data?

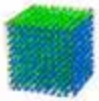
Comparison to experiment

- terrestrial data (Assadi *et al.*)
- μg data (parabolic flight)
  H. Hartmann (PhD thesis)

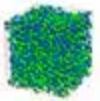
H. Assadi, *et al.*, Acta Mat. 54, 2793 (2006).



A. Kerrache *et al.*, EPL **81** (2008) 58001. good agreement with experimental data



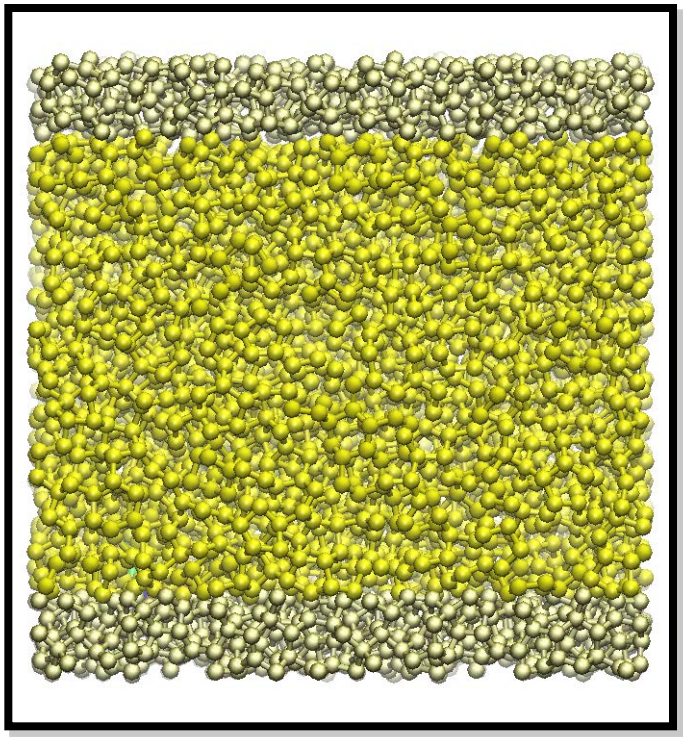
Molecular Dynamics Simulation Results



Effects of Shear deformations on the structure of amorphous silicon

Shear Deformations on amorphous silicon

- Equilibration of the sample at the desired temperature (Bulk simulation).
- Define the lower & upper walls.
- Move the particles of upper wall with a fixed shear velocity v_s
- Integrate the equation of motion of the mobile particles.
- Equilibration for 5 ns: fixed walls
- Periodic boundary conditions fixed in y direction.

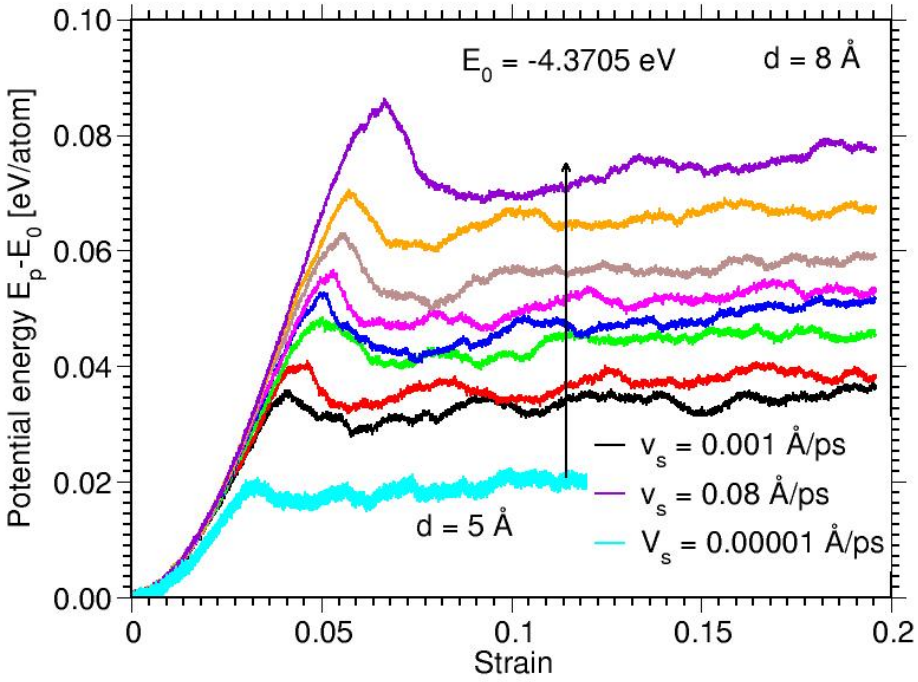


Typical starting configuration
Of amorphous silicon

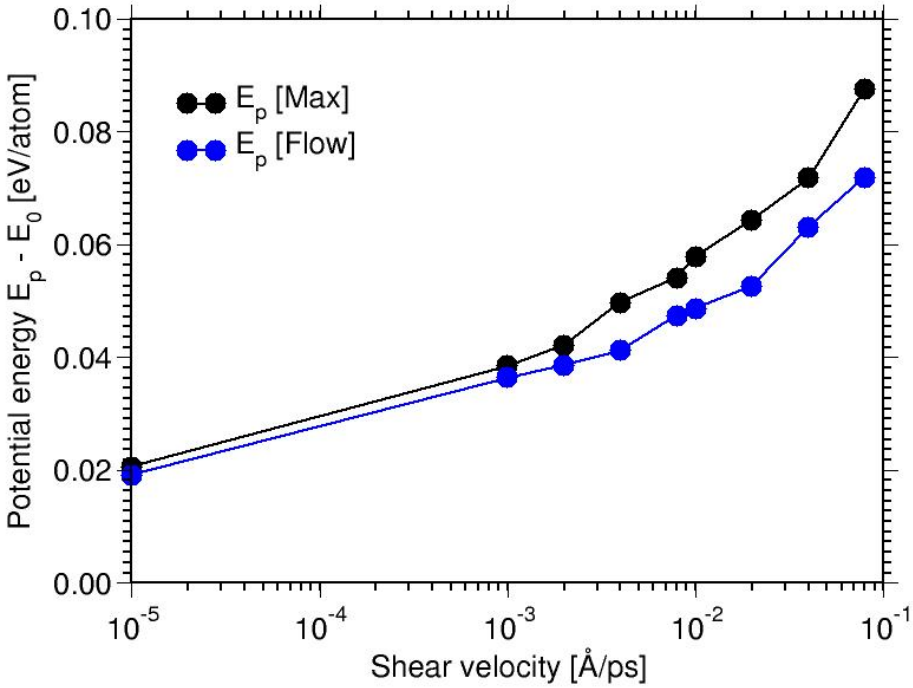
❑ Temperature:

rescaling the velocities using the y and z components

Shear Deformations at 300 K



Potential energy difference ΔE as a Function of imposed strain at 300 K



Shear velocity: 10^{-5} to $8x10^{-2}$ Å/ps

Quadratic increase of E_p and strain independent at small strain.
E_{max} increases with shear velocity.

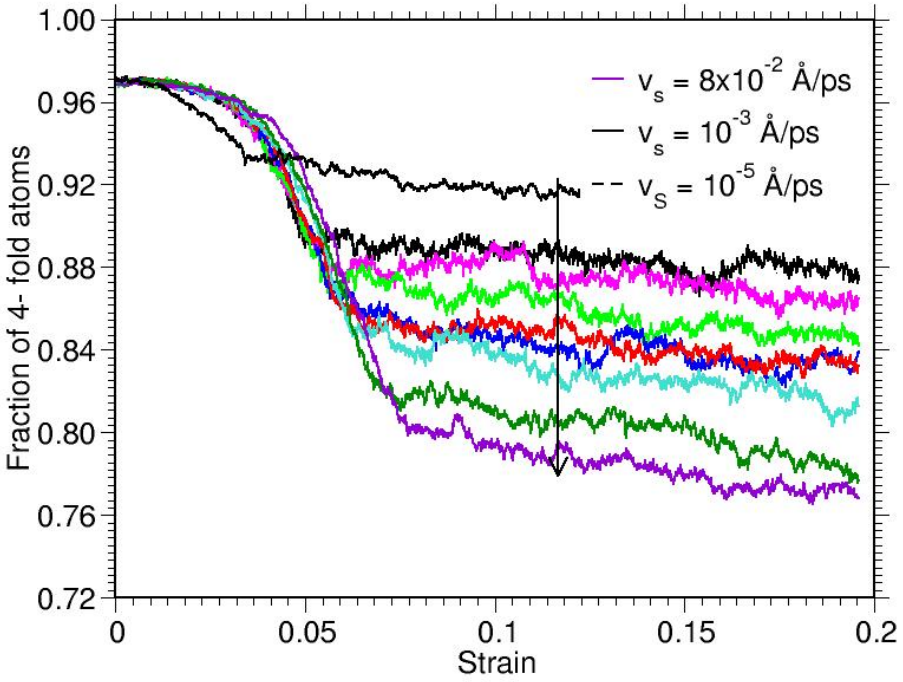
10^{-3} Å/ps ➔ 8 ns

10^{-5} Å/ps ➔ 800 ns

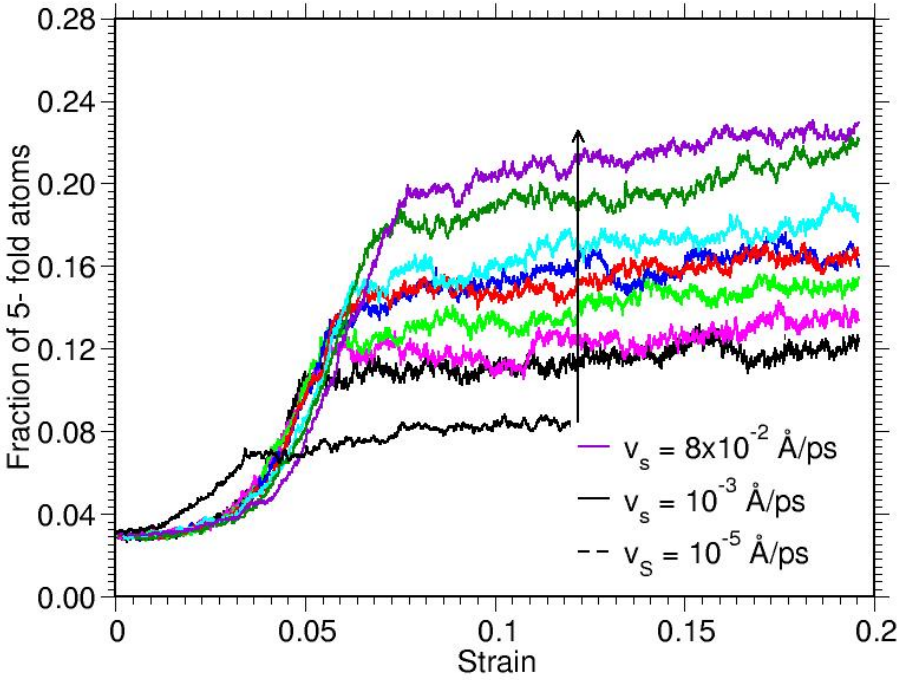
A. Kerrache et al. PRB **83** (2011) 134122.

Shear Deformations at 300 K

Fraction of perfect 4- fold atoms



Fraction of 5- fold atoms

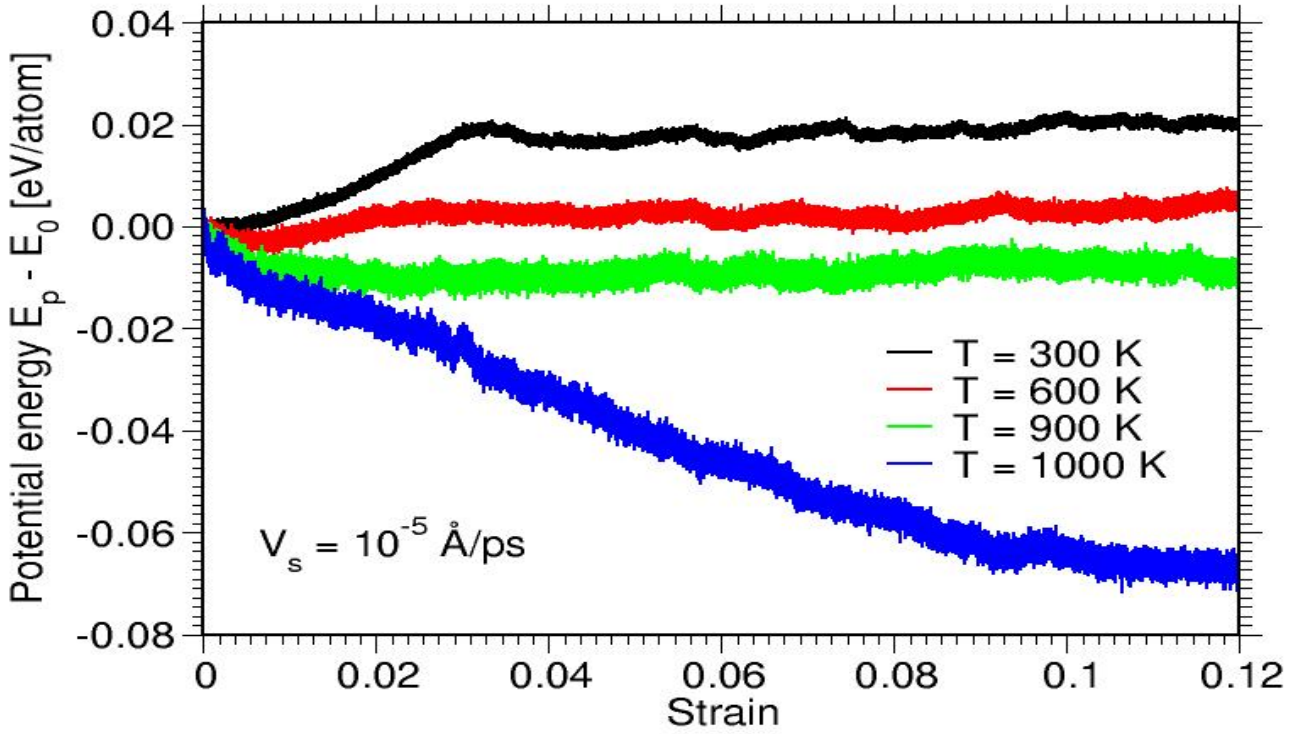


Increasing shear velocity leads:

- ➢ an increase of the fraction of 5- fold atoms
- ➢ a decrease of the fraction of 4- fold atoms

→ Increases the disorder and the defects

Shear induce crystallization



Temperatures:
300, 600, 900 and 1000 K
Shear velocity: 10^{-5} Å/ps
δ = 5 Å

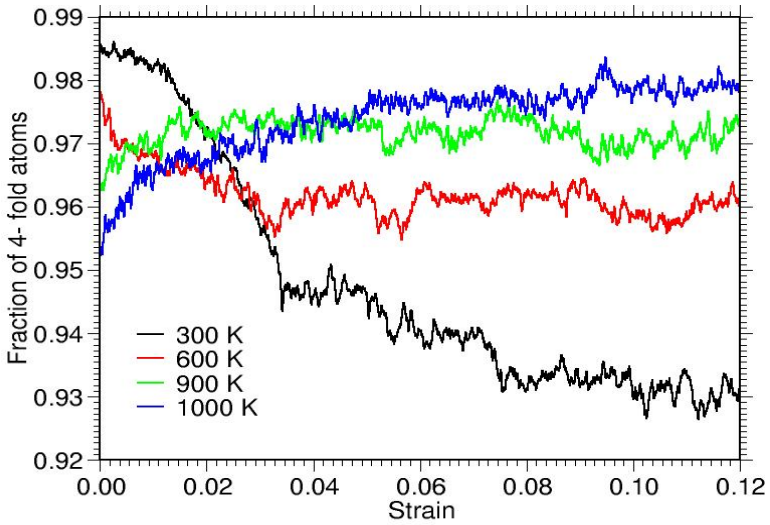
Potential energy ΔE_P as a function of imposed strain

Increase of the disorder at 300 and 600 K
At 900 K: increase of the order without crystallization
At 1000 K: shear induce the crystallization of a-Si

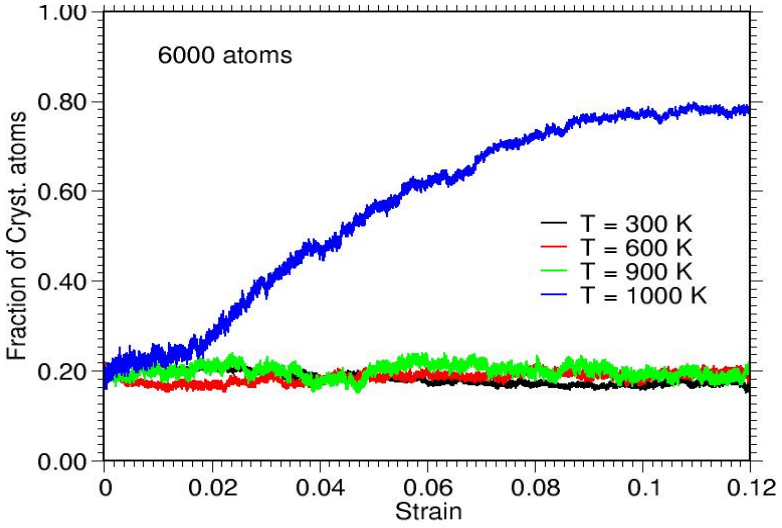
T	300	600	900	1000
ΔE_P	+0.02	+0.005	-0.01	-0.065

A. Kerrache et al. PRB **84** (2011) 041110.

Shear induce crystallization

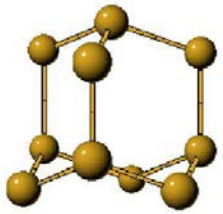


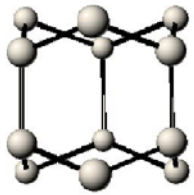
Coordination number



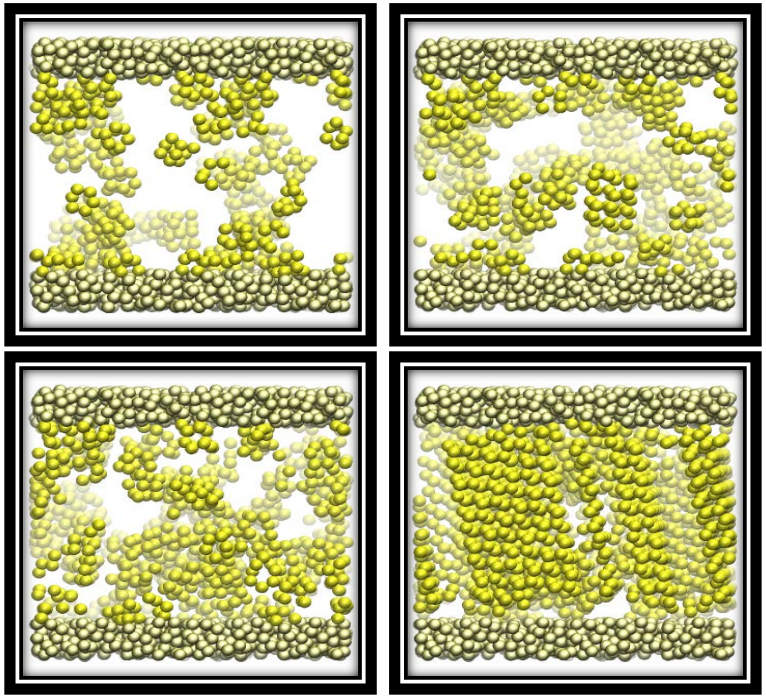
Number of crystalline particles

P. Beaucage et al. PRB (2005)





300 K 600 K



900 K 1000 K

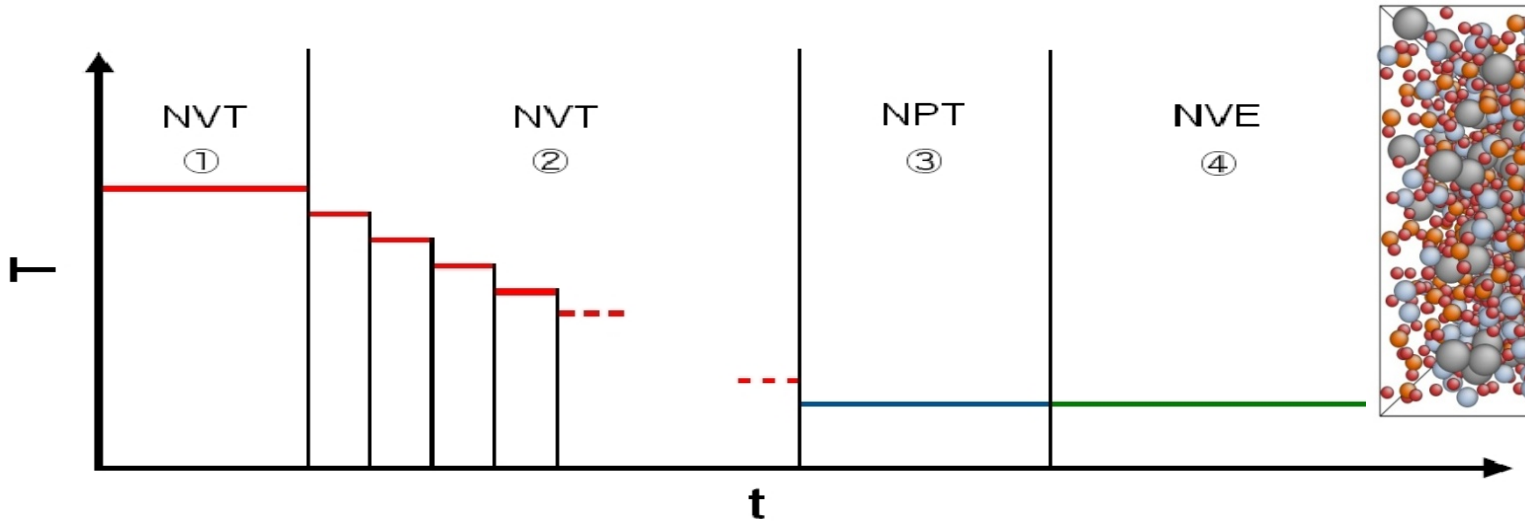
Molecular Dynamics Simulation Results

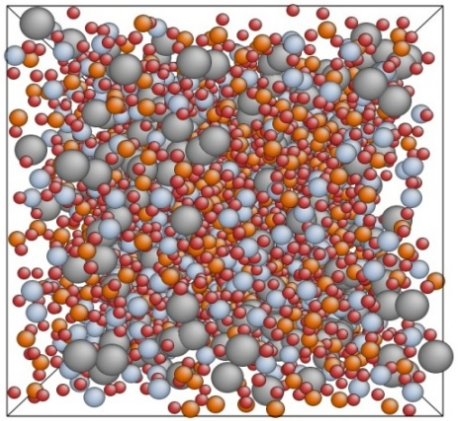
Study of borosilicate (SBN) glasses:

Indentation

Glass preparation

Glass preparation diagram





Cooling rates: 10^{12} à 10^{13} K/s

Glass preparation procedure:

- Random configuration (N atoms).
- Liquid equilibration du at 5000 K (NVT).
- Coming per steps of 100 K– (NVT).
- Glass equilibration at 300 K (NPT).
- Trajectory simulation at 300 K (NVE).

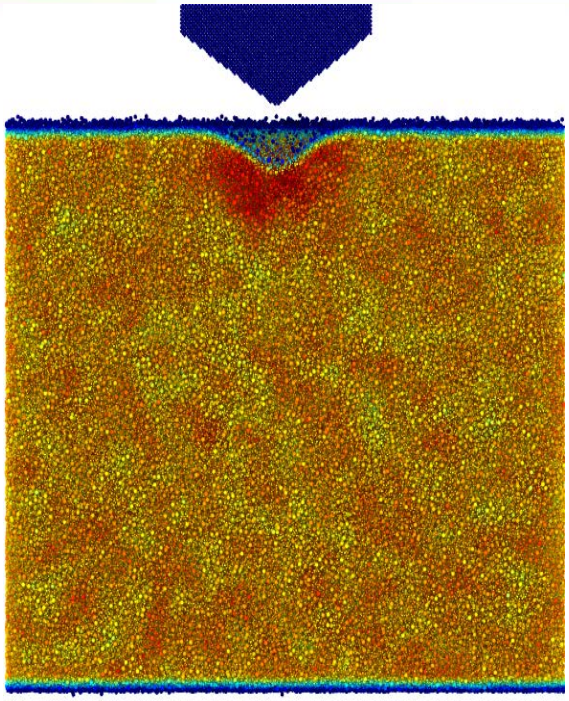
Model:

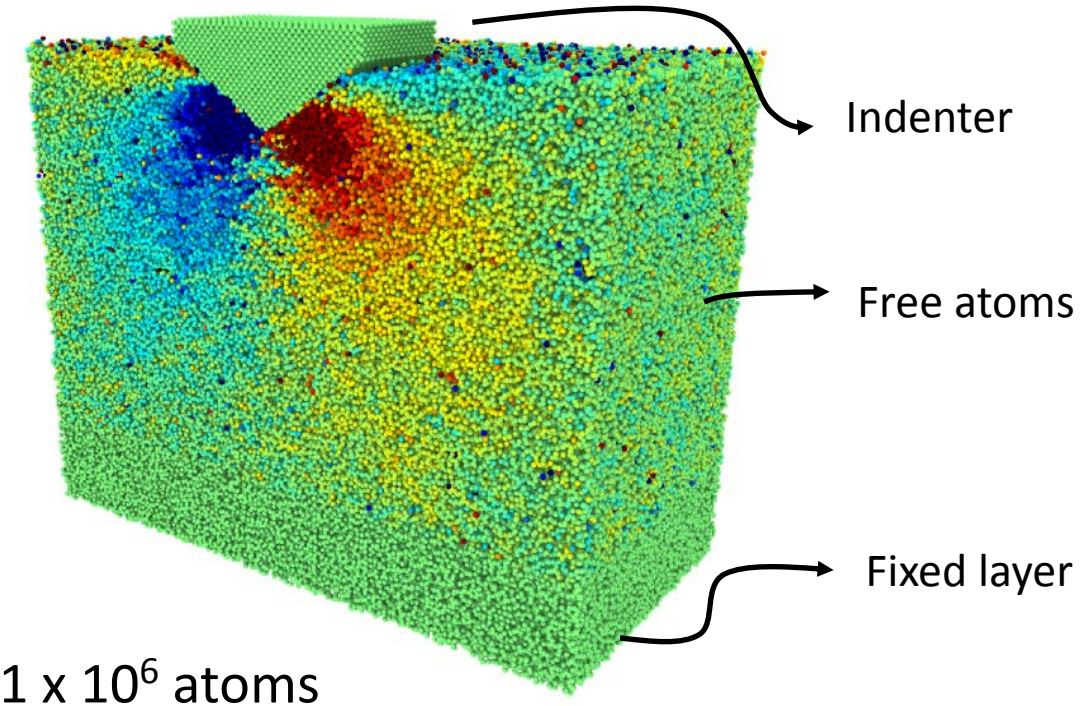
- MD Simulations (DL-POLY).
- Systems of N particules.
- Time step: 1 fs

SBN glasses:

- SiO_2-B_2O_3-Na_2O
  - R = [Na_2O] / [B_2O_3]
  - K = [SiO_2] / [B_2O_3]

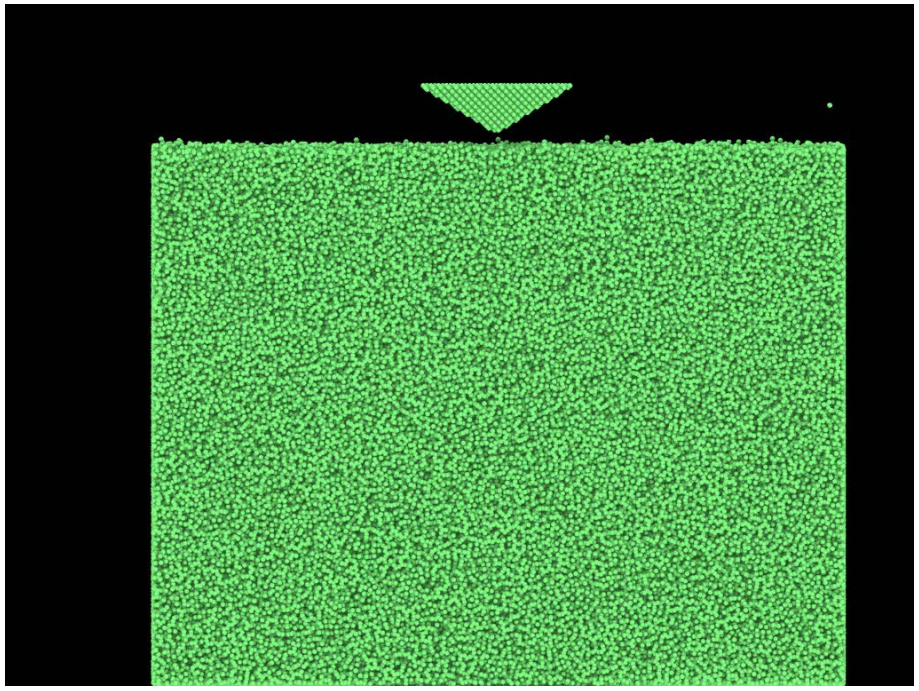
Indentation

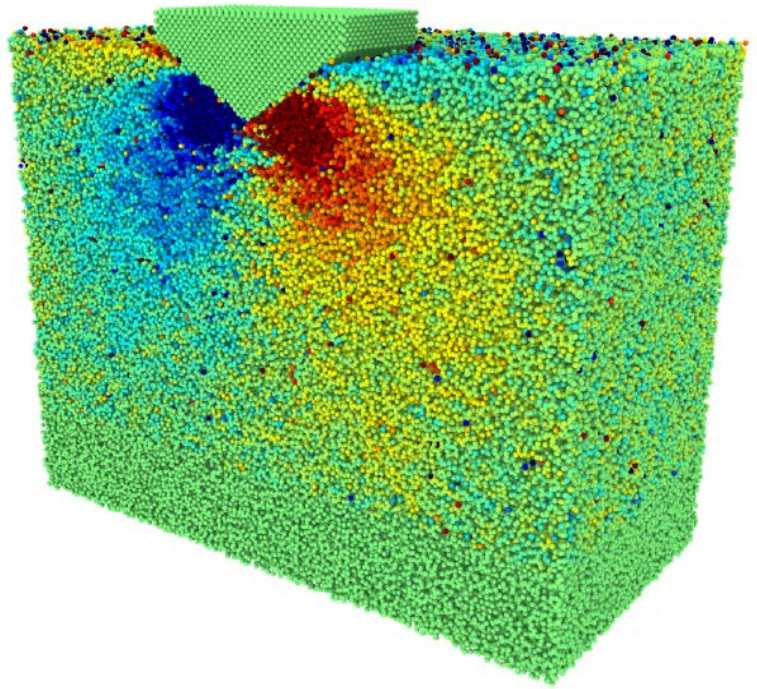




- 2.1 x 10^6 atoms
- Temperature : 300 K
- Indentation speed : 10 m/s
- Penetration depth : ~3.0 nm

Indentation





Movie provided by: **Dimitrios Kilymis**

Laboratoire Charles Coulomb (L2C), UMR 5221 CNRS-Univ. Montpellier, France.

Acknowledgments

Prof. Dr. Jürgen Horbach, Dusseldorf, Germany.
Prof. Dr. Kurt Binder (Mainz, Germany).
Prof. A. Meyer and Prof. D. Herlach (DLR).

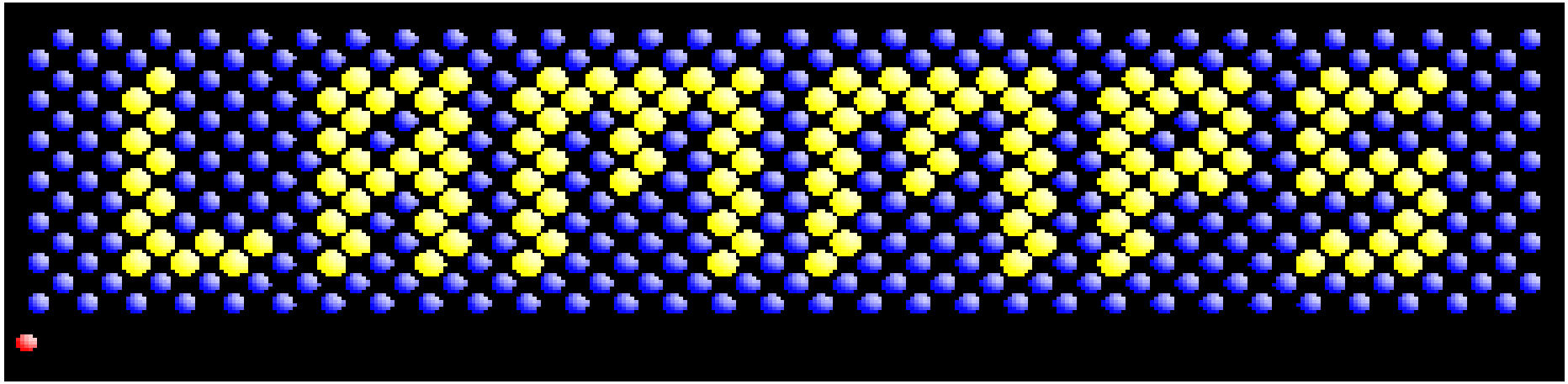
Prof. Normand Mousseau, Qc, Canada.
Prof. Laurent J. Lewis, Qc, Canada.

Dr. Dimitrios Kilymis, Montpellier, France.
Prof. Jean-Marc Delaye, CEA, France.

Dr. Victor Teboul, Angers, France.
Prof. Hamid Bouzar, UMMTO, Tizi-Ouzou, Algeria.

Introduction to LAMMPS

Large-scale Atomic / Molecular Massively Parallel Simulator



Source: some material and images were adapted from LAMMPS home page

LAMMPS: Classical Molecular Dynamics Code

Large-scale **A**tomic / **M**olecular **M**assively **P**arallel **S**imulator

S. Plimpton, A. Thompson, R. Shan, S. Moore, A. Kohlmeyer …

Sandia National Labs: *http://www.sandia.gov/index.html*

- **Home Page: http://lammps.sandia.gov/**

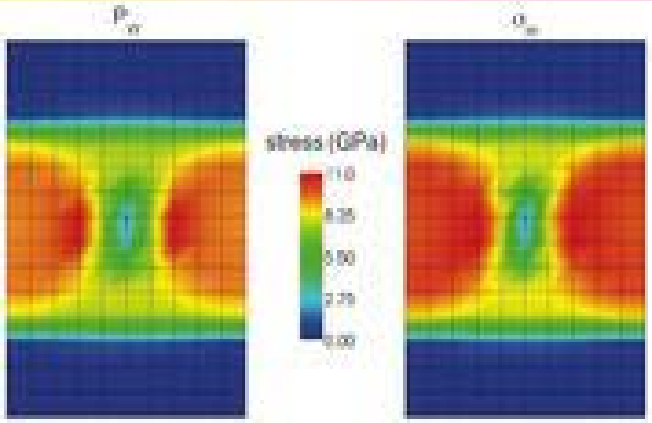
Results:

- **Papers: http://lammps.sandia.gov/papers.html**
- **Pictures: http://lammps.sandia.gov/pictures.html**
- **Movies: http://lammps.sandia.gov/movies.html**

Resources:

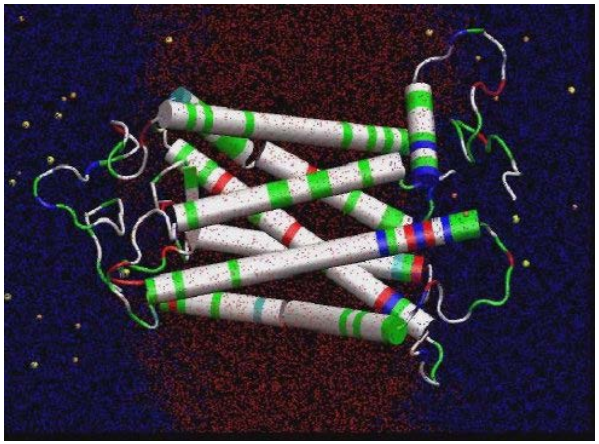
- **Online Manual: http://lammps.sandia.gov/doc/Manual.html**
- **Search the mailing list: http://lammps.sandia.gov/mail.html**
- **Search, Subscribe and Email the Mailing List: https://sourceforge.net/p/lammps/mailman/lammps-users/**

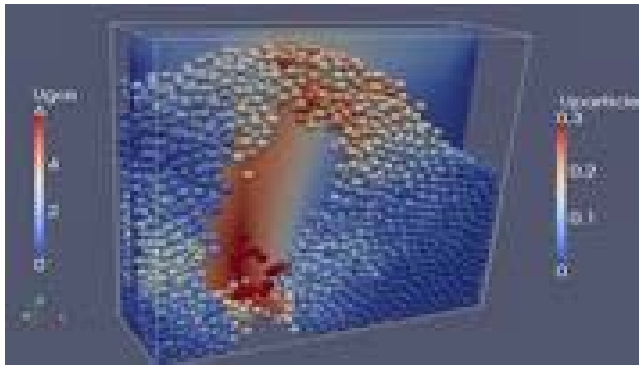
Where LAMMPS has been used?



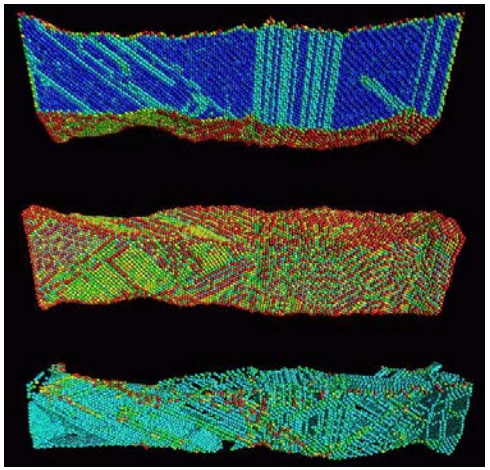
➢ Solid Mechanics

➢ Biophysics



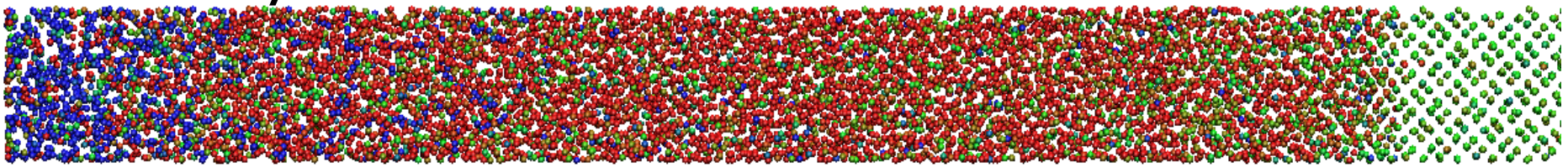


➢ Granular Flow



➢ Material Science

➢ Chemistry



Start with LAMMPS

Big Picture	Code	Documentation	Results	Related Tools	Context	User Support
Features	Download	Manual	Publications	Pre/Post processing	Authors	Mail list
Non-features	SourceForge	Developer guide	Pictures	Pizza.py Toolkit	History	Workshops
FAQ	Latest features & bug fixes	Tutorials	Movies	Offsite LAMMPS packages & tools	Funding	User scripts and HowTos
Wish list	Unfixed bugs	MD to LAMMPS glossary	Benchmarks	Visualization	Open source	Contribute to LAMMPS

Recent LAMMPS News

- NEW (1/17) Added a fix mscg command to enable building of multi-scale coarse-graining (MSCG) models via the Voth group's (U Chicago) MS-CG library.
- NEW (12/16) Significant features added to LAMMPS in the fourth quarter of 2016 include these new commands: compute global/atom global_atom.html, temper/grem and fix grem, pair tersoff/mod/c, pair agni, pair born/coul/dsf and pair_style born/coul/dsf/cs, dump nc and dump nc/mpiio, fix halt, fix dpd/energy, dump_modify thresh LAST option, and fix wall/gran/region. See authors here and details here.
- NEW (11/16) Added temper/grem and fix grem commands to enable tempering via the generalized replica exchange method (gREM) method.
- NEW (10/16) Added a fix wall/gran/region command which allows geometric regions to act as boundaries for granular particles.
- NEW (9/16) Significant features added to LAMMPS in the third quarter of 2016 include these new commands: allow for multiple procs per replica with the neb command, options for weighted load balancing via the

Design of LAMMPS

License

- LAMMPS is provided through **GNU Public License**
  https://www.gnu.org/licenses/licenses.en.html#GPL
- Free to Use, **Modify**, and Distribute.
- Contribute to LAMMPS:
  http://lammps.sandia.gov/contribute.html

Code Layout

- C++ and Object-Oriented approach
- Parallelization via MPI and OpenMP; runs on GPU.
- Is invoked by commands through input scripts.
- Possibility to customized output.
- Could be interfaced with other codes.

Obtaining LAMMPS

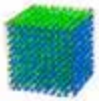
❖ Download Page:

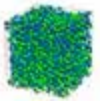
http://lammps.sandia.gov/download.html

➢ Distributions:

- ✓ Download a tarball ← Source Code
- ✓ Git checkout and update
- ✓ SVN checkout and update
- ✓ Pre-built Ubuntu executables ← Executable Ubuntu
- ✓ Pre-built binary RPMs for Fedora/RedHat/CentOS/openSUSE
- ✓ Pre-built Gentoo executable
- ✓ OS X with Homebrew ← RPMs - Linux
- ✓ Windows installer package ← Installation under Windows
- ✓ Applying patches

Download the source code





Download a tarball

Select the code you want, click the "**Download Now**" button, and your browser should download a gzipped tar file. Unpack it with the following commands, and look for a README to get you started.

```
tar -xzvf file.tar.gz
```

There have been ~256,700 downloads of LAMMPS from Sept 2004 thru Dec 2016.

LAMMPS molecular dynamics package:

- LAMMPS --- Stable version (17 Nov 2016) - Recent C++ version source tarball, GPL license, ~105 Mb. Includes all bug fixes and new features described on this page, up to the date of the most recent stable release.
- LAMMPS --- Development version - Most current C++ version source tarball, GPL license, ~105 Mb. Includes all bug fixes and new features described on this page.
- LAMMPS 2001 --- older f90 version source tarball, GPL license, 1.1 Mb, last updated 17 Jan 2005
- LAMMPS 99 --- older f77 version source tarball, GPL license, 840 Kb
- No package

Download Now

LAMMPS molecular dynamics package:

o **LAMMPS --- Stable version** (17 Nov 2016) - Recent C++ version source tarball, GPL license, ~105 Mb. Includes all bug fixes and new features described on this page, up to the date of the most recent stable release.

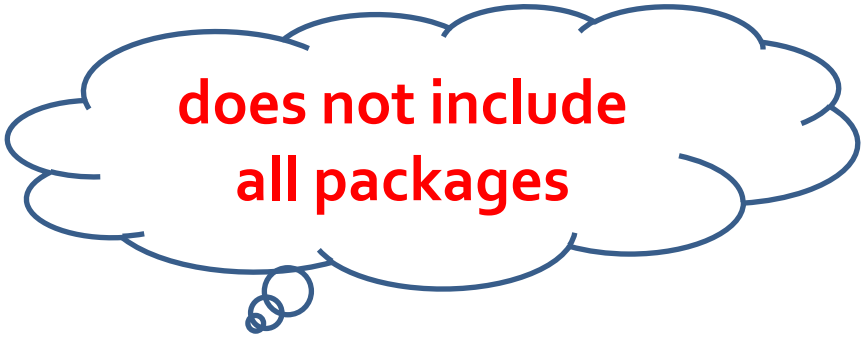
o **LAMMPS --- Development version** - Most current C++ version source tarball, GPL license, ~105 Mb. Includes all bug fixes and new features described on this page (folow the link: **Features**).

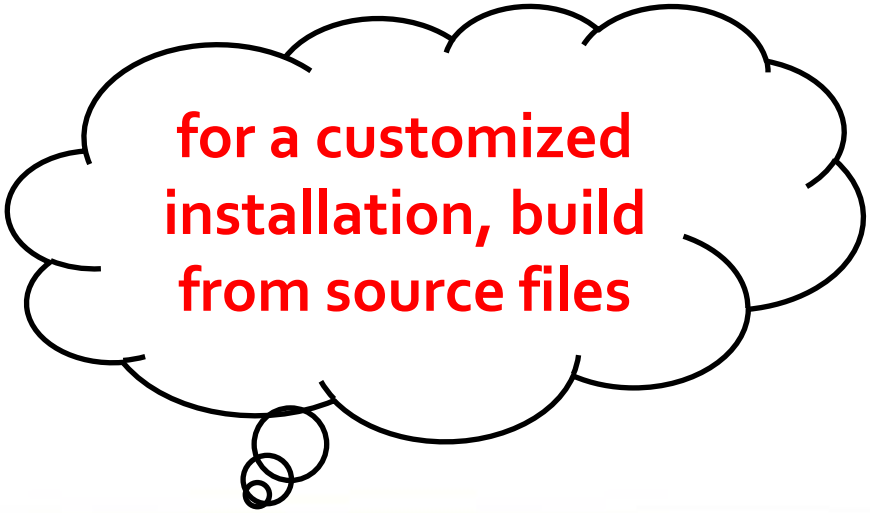
o **LAMMPS 2001** --- older f90 version source tarball, GPL license, 1.1 Mb, last updated 17 Jan 2005.

o **LAMMPS 99** --- older f77 version source tarball, GPL license, 840 Kb.

Installing and Building LAMMPS

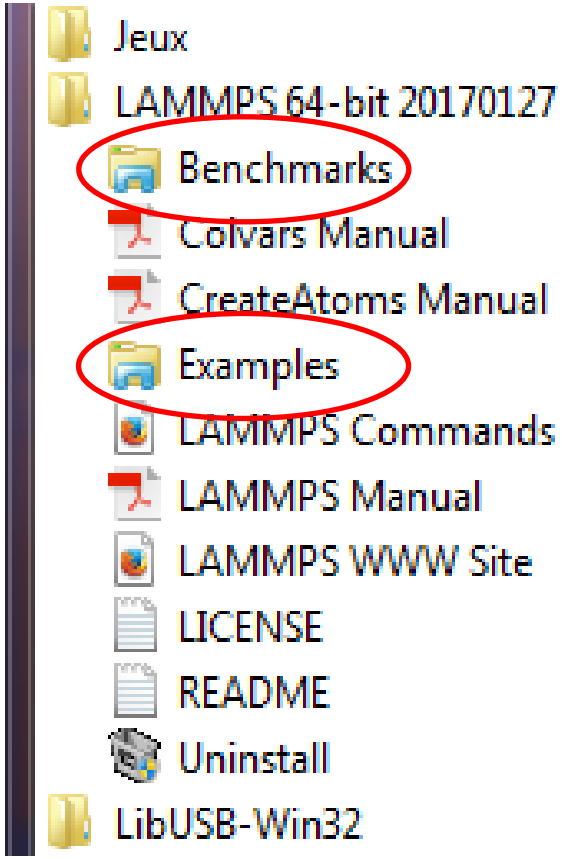
- **Build from RPMs**
  - ✓ Pre-built Ubuntu executables
  - ✓ Pre-built binary RPMs for Fedora/RedHat/CentOS/openSUSE
  - ✓ Pre-built Gentoo executable
  - ✓ OS X with Homebrew
- **Install under windows**
  - ✓ Windows installer package
- **Build from source code**
  - ✓ Download a tarball
  - ✓ Git checkout and update
  - ✓ SVN checkout and update
  - ✓ Applying patches





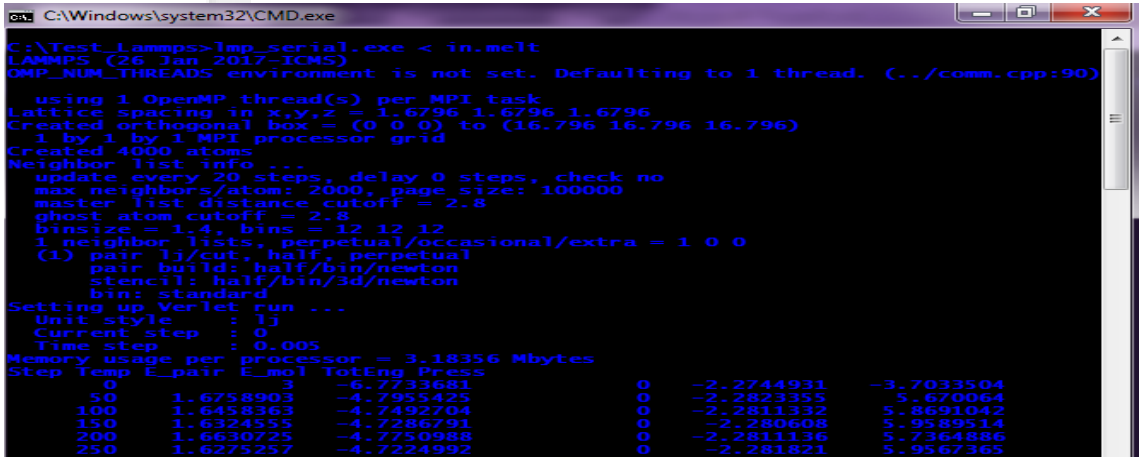
LAMMPS: Windows Installer

- **Download Page:** http://rpm.lammps.org/windows.html
- **Installer:** lammps-64bit-latest.exe



Directory:
Program Files\LAMMPS 64-bit 20170127
Executable under bin:
abf_integrate.exe ffmpeg.exe lmp_mpi.exe restart2data.exe binary2txt.exe lmp_serial.exe chain.exe msi2lmp.exe createatoms.exe

- **Execute:** lmp_serial.exe < in.lammps



LAMMPS: code source directories

- Download the source code: lammps-stable.tar.gz
- LAMMPS directory: lammps-17Nov16
  - bench: Benchmark tests (potential, input and output files).
  - doc: documentation (PDF and HTML)
  - examples: input and output files for some simulations
  - lib: libraries to build before building LAMMPS
  - LICENSE and README files.
  - potentials: some of the force fields supported by LAMMPS
  - python: to invoke LAMMPS library from Python
  - src: source files (*.cpp, PACKAGES, USER-PACKAGES, Makefile)
  - tools: some tools like xmovie (similar to VMD but at 2D).

LAMMPS: code source directories

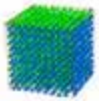
- **lib:** libraries to build before building LAMMPS

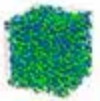
  atc, colvars, gpu, kim, linalg, molfile, poems, qmmm, smd, vtk, awpmd, compress, h5md, kokkos, meam, netcdf, python, quip, reax, voronoi

- **src:** source files (*.cpp, PACKAGES, USER-PACKAGES, Makefile):

  - **Setting and Building Files:** Makefile Makefile.list Make.sh Make.py Makefile.package.empty Makefile.package.settings.empty
  - **Source Files:** *.cpp *.h
  - **Standard packages:** asphere body class2 colloid compress coreshell dipole gpu granular kim kokkos kspace manybody …
  - **User-contributed packages:** user-atc user-awpmd user-cg-cmm user-colvars user-diffraction user-dpd user-drude user-eff user-fep user-h5md user-intel user-lb user-manifold …
  - **MAKE:** MACHINES Makefile.mpi Makefile.serial MINE OPTIONS

Building LAMMPS from source code





$ make

- make clean-all — delete all object files
- make clean-machine — delete object files for one machine
- make package — list available packages
- make package-status (ps) — status of all packages
- make yes-package — install a single package in src dir
- make no-package — remove a single package from src dir
- make yes-all — install all packages in src dir
- make no-all — remove all packages from src dir
- make yes-standard (yes-std) — install all standard packages
- make no-standard (no-std) — remove all standard packages
- make yes-user — install all user packages
- make no-user — remove all user packages
- make machine — build LAMMPS for machine
- make mode=lib machine — build LAMMPS as static lib for machine
- make mode=shlib machine — build LAMMPS as shared lib for machine

Building LAMMPS using GNU Makefile

- machine is one of these from src/MAKE:
  - # mpi = MPI with its default compiler
  - # serial = GNU g++ compiler, no MPI

- … or one of these from src/MAKE/OPTIONS:
  - # icc_openmpi = OpenMPI with compiler set to Intel icc
  - # icc_openmpi_link = Intel icc compiler, link to OpenMPI
  - # icc_serial = Intel icc compiler, no MPI

- … or one of these from src/MAKE/MACHINES:
  - # cygwin = Windows Cygwin, mpicxx, MPICH, FFTW
  - # mac = Apple PowerBook G4 laptop, c++, no MPI, FFTW 2.1.5
  - # mac_mpi = Apple laptop, MacPorts Open MPI 1.4.3, gcc 4.8, fftw
  - # ubuntu = Ubuntu Linux box, g++, openmpi, FFTW3

- … or one of these from src/MAKE/MINE:

Running LAMMPS

- Executable: lmp_machine

- Files:
  - Input File: in.lmp_file
  - Potential: see examples and the last slides for more details
  - Initial configuration: can be generated by LAMMPS, or another program or home made program.

- Interactive Execution:

```
$ ./lmp_machine < in.lmp_file
$ ./lmp_machine –in in.lmp_file
```

- Background Execution:

```
$ ./lmp_machine < in.lmp_file > log_lammps_output_file
$ ./lmp_machine –in in.lmp_file –l log_lammps_output_file
```

Command Line Options

❑ Command-line options:

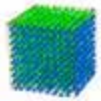
At run time, LAMMPS recognizes several optional command-line switches which may be used in any order.

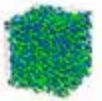
-e or -echo, -h or –help, -i or –in, -k or –kokkos, -l or –log, -nc or –nocite, -pk or –package, -p or –partition, -pl or –plog, -ps or –pscreen, -r or –restart, -ro or –reorder, -sc or –screen, -sf or –suffix, -v or –var

❑ For example:

mpirun -np 10 lmp_machine -l my.log -sc none -in in.alloy

LAMMPS simulation overview





INPUT	RUNNING	OUTPUT
• Initial positions • Initial velocities • PBC • Units • Potential • Ensemble • …. etc.	• Molecular Dynamics Simulation • Minimization • Monte Carlo	• Trajectories • Velocities • Forces • Energy • Temperature • Pressure • Density • Snapshots • Movies • … etc.

Outline of LAMMPS input script

- **Command Line:**
  - Every simulation is executed by supplying an input text script to the LAMMPS executable.

- **Parts of an input script**
  - **Initialize:** units, dimensions, etc.
  - Atomic positions and velocities
  - **Settings:**
    - Inter-atomic potential
    - Run time simulation parameters (e.g. time step)
    - Fixes - operations during dynamics (e.g. thermostat)
    - Computes - calculation of properties during dynamics

- **Run the simulation for N steps.**

LAMMPS: example of input script (LJ)

```
# 3d Lennard-Jones melt

units           lj
atom_style      atomic

lattice         fcc 0.8442
region          box block 0 10 0 10 0 10
create_box      1 box
create_atoms    1 box
mass            1 1.0

velocity        all create 3.0 87287

# Potential

pair_style      lj/cut 2.5
pair_coeff      1 1 1.0 1.0 2.5
```

Comment

Define units

Create the simulation box

Initialize the velocities

Define the potential

LAMMPS: example of input script (continue …)

```
# Neighbour list:
neighbor        0.3 bin
neigh_modify    every 20 delay 0 check no

#  set the thermodynamic ensemble:
fix             1 all nve

dump            id all atom 50 dump.melt
#dump_modify ….

log             log.melt
#thermo_style custum step temp etotal ….
thermo          50


run             250

# End of the simulation.
```

Monitor the neighbour list

Thermodynamic Ensemble

Store the trajectory

Log file
Printing frequency
Customize output

Run the simulation for N steps

LAMMPS: some commands

Initialization

- **Parameters: s**et parameters that need to be defined before atoms are created: ***units***, *dimension*, *newton*, *processors*, ***boundary***, ***atom_style***, *atom_modify*.

- If force-field parameters appear in the files that will be read: ***pair_style***, *bond_style*, *angle_style*, *dihedral_style*, *improper_style*.

- **Atom definition:** there are 3 ways to define atoms in LAMMPS.

  - Read them in from a data or restart file via the ***read_data*** or ***read_restart*** commands.
  - Or create atoms on a lattice (with no molecular topology), using these commands: *lattice*, *region*, *create_box*, *create_atoms*.
  - Duplicate the box to make a larger one the ***replicate*** command.

LAMMPS: some commands

- **Settings:**
  - Once atoms are defined, a variety of settings can be specified:
    force field coefficients, simulation parameters, output options …

  - **Force field coefficients:**
    pair_coeff, *bond_coeff*, *angle_coeff*, *dihedral_coeff*, *improper_coeff*, *kspace_style*, *dielectric*, *special_bonds*.

  - **Various simulation parameters:**
    neighbor, *neigh_modify*, *group*, *timestep*, *reset_timestep*, *run_style*, *min_style*, *min_modify*.

  - **Computations during a simulation:**
    compute, *compute_modify*, and *variable* commands.

  - **Output options:** *thermo*, *dump*, and *restart* commands.

LAMMPS: output log file

LAMMPS (30 Jul 2016)
using **1 O**penMP thread(s) per MPI task
 # 3d Lennard-Jones melt

units ljatom_style atomic
lattice fcc 0.8442Lattice spacing in x,y,z = 1.6796 1.6796 1.6796
region box block 0 10 0 10 0 10
create_box 1 box

Created orthogonal box = (0 0 0) to (16.796 16.796 16.796) 2 by 2 by 3 MPI processor grid

create_atoms 1 box
Created 4000 atoms

mass 1 1.0

LAMMPS: output log file

```
thermo          100
run             25000
Neighbor list info ...
    1 neighbor list requests
    update every 20 steps, delay 0 steps, check no
    max neighbors/atom: 2000, page size: 100000
    master list distance cutoff = 2.8
    ghost atom cutoff = 2.8
    binsize = 1.4 -> bins = 12 12 12
    Memory usage per processor = 2.05293 Mbytes
Step Temp E_pair E_mol TotEng Press
       0            3   -6.7733681            0   -2.2744931   -3.7033504
     100    1.6510577   -4.7567887            0   -2.2808214    5.8208747
     200    1.6393075   -4.7404901            0   -2.2821436    5.9139187
     300    1.6626896   -4.7751761            0   -2.2817652     5.756386
     400    1.6496817   -4.7563868            0   -2.2824829    5.7979802
```

LAMMPS: output log file

24900	1.5691281	-4.7849759	0	-2.4318721	5.605906
25000	1.552843	-4.7611011	0	-2.432419	5.7187477

Loop time of 15.4965 on 12 procs for 25000 steps with 4000 atoms
Performance: 696931.853 tau/day, 1613.268 timesteps/s
90.2% CPU use with 12 MPI tasks x 1 OpenMP threads

MPI task timing breakdown:

Section	min time	avg time	max time	%varavg	%total
Pair	6.6964	7.1974	7.9599	14.8	46.45
Neigh	0.94857	1.0047	1.0788	4.3	6.48
Comm	6.0595	6.8957	7.4611	17.1	44.50
Output	0.01517	0.01589	0.019863	1.0	0.10
Modify	0.14023	0.14968	0.16127	1.7	0.97
Other		0.2332			1.50

Total wall time: 0:00:15

Customize output: thermo & thermo_style

thermo **freq_steps**
thermo_style **style args**

- **style =** *one* or *multi* or *custom*
- **args =** list of arguments for a particular style
  - *one* args = none
  - *multi* args = none *custom*
  - args = list of keywords possible

❑ **keywords = step, elapsed, elaplong, dt, time, cpu, tpcpu, spcpu, cpuremain, part, timeremain, atoms, temp, press, pe, ke, etotal, enthalpy, evdwl, ecoul, epair, ebond, eangle, edihed, eimp, emol, elong, etail, vol, density, lx, ly, lz, xlo, xhi, ylo, yhi, zlo, zhi, xy, xz, yz, xlat, ylat, zlat, bonds, angles, dihedrals, impropers, pxx, pyy, pzz, pxy, pxz, pyz ….. etc**

LAMMPS: Potential classified by Material

- **Biomolecules:** CHARMM, AMBER, OPLS, COMPASS (class 2), long-range Coulombics via PPPM, point dipoles, …
- **Polymers:** all-atom, united-atom, coarse-grain (bead-spring FENE), bond-breaking, …
- **Materials:** EAM and MEAM for metals, Buckingham, Morse, Yukawa, Stillinger-Weber, Tersoff, EDIP, COMB, SNAP, …
- **Chemistry:** AI-REBO, REBO, ReaxFF, eFF
- **Mesoscale:** granular, DPD, Gay-Berne, colloidal, peri-dynamics, DSMC…
- **Hybrid:** can use combinations of potentials for hybrid systems: water on metal, polymers/semiconductor interface, colloids in solution, …

LAMMPS: Potential classified by Functional Form

- **Pairwise potentials:** Lennard-Jones, Buckingham, …
- **Charged Pairwise Potentials:** Coulombic, point-dipole
- **Manybody Potentials:** EAM, Finnis/Sinclair, modified EAM (MEAM), embedded ion (EIM), Stillinger-Weber, Tersoff, AI-REBO, ReaxFF, COMB
- **Coarse-Grained Potentials:** DPD, GayBerne, …
- **Mesoscopic Potentials:** granular, peri-dynamics
- **Long-Range Electrostatics:** Ewald, PPPM, MSM
- **Implicit Solvent Potentials:** hydrodynamic lubrication, Debye
- **Force-Field Compatibility with common**: CHARMM, AMBER, OPLS, GROMACS options

More about LAMMPS

- **Home Page:** http://lammps.sandia.gov/
- **Examples:** deposit, friction, micelle, obstacle, qeq, streitz, MC, body, dipole, hugoniostat, min, peptide, reax, tad, DIFFUSE, colloid, indent, msst, peri, rigid , vashishta, ELASTIC, USER, comb, eim, nb3b, pour, shear, voronoi, ELASTIC_T, VISCOSITY, coreshell, ellipse, meam, neb, prd, snap, HEAT, accelerate, crack, flow, melt, nemd
- **Results:**
  - Papers: http://lammps.sandia.gov/papers.html
  - Pictures: http://lammps.sandia.gov/pictures.html
  - Movies: http://lammps.sandia.gov/movies.html
- **Resources:**
  - Online Manual: http://lammps.sandia.gov/doc/Manual.html
  - Search the mailing list: http://lammps.sandia.gov/mail.html
  - Search, Subscribe and Email the Mailing List: https://sourceforge.net/p/lammps/mailman/lammps-users/

Acknowledgments

Thanks to LAMMPS developers

Thanks to LAMMPS contributors

Thank you for your attention

LAMMPS: pair style

pair_style none - turn off pairwise interactions
pair_style hybrid - multiple styles of pairwise interactions
pair_style hybrid/overlay - multiple styles of superposed pairwise interactions
pair_style zero - neighbor list but no interactions
pair_style adp - angular dependent potential (ADP) of Mishin
pair_style airebo - AIREBO potential of Stuart
pair_style airebo/morse - AIREBO with Morse instead of LJ
pair_style beck - Beck potential
pair_style body - interactions between body particles
pair_style bop - BOP potential of Pettifor
pair_style born - Born-Mayer-Huggins potential
pair_style born/coul/long - Born-Mayer-Huggins with long-range Coulombics
pair_style born/coul/long/cs - Born-Mayer-Huggins with long-range Coulombics and core/shell
pair_style born/coul/msm - Born-Mayer-Huggins with long-range MSM Coulombics

LAMMPS: pair style

pair_style born/coul/wolf - Born-Mayer-Huggins with Coulombics via Wolf potential
pair_style brownian - Brownian potential for Fast Lubrication Dynamics
pair_style brownian/poly - Brownian potential for Fast Lubrication Dynamics with polydispersity
pair_style buck - Buckingham potential
pair_style buck/coul/cut - Buckingham with cutoff Coulomb
pair_style buck/coul/long - Buckingham with long-range Coulombics
pair_style buck/coul/long/cs - Buckingham with long-range Coulombics and core/shell
pair_style buck/coul/msm - Buckingham long-range MSM Coulombics
pair_style buck/long/coul/long - long-range Buckingham with long-range Coulombics
pair_style colloid - integrated colloidal potential
pair_style comb - charge-optimized many-body (COMB) potential
pair_style comb3 - charge-optimized many-body (COMB3) potential

LAMMPS: pair style

pair_style coul/cut - cutoff Coulombic potential
pair_style coul/debye - cutoff Coulombic potential with Debye screening
pair_style coul/dsf - Coulombics via damped shifted forces
pair_style coul/long - long-range Coulombic potential
pair_style coul/long/cs - long-range Coulombic potential and core/shell
pair_style coul/msm - long-range MSM Coulombics
pair_style coul/streitz - Coulombics via Streitz/Mintmire Slater orbitals
pair_style coul/wolf - Coulombics via Wolf potential
pair_style dpd - dissipative particle dynamics (DPD)
pair_style dpd/tstat - DPD thermostatting
pair_style dsmc - Direct Simulation Monte Carlo (DSMC)
pair_style eam - embedded atom method (EAM)
pair_style eam/alloy - alloy EAM
pair_style eam/fs - Finnis-Sinclair EAM
pair_style eim - embedded ion method (EIM)
pair_style gauss - Gaussian potential

LAMMPS: pair style

pair_style gayberne - Gay-Berne ellipsoidal potential
pair_style gran/hertz/history - granular potential with Hertzian interactions
pair_style gran/hooke - granular potential with history effects
pair_style gran/hooke/history - granular potential without history effects
pair_style hbond/dreiding/lj - DREIDING hydrogen bonding LJ potential
pair_style hbond/dreiding/morse - DREIDING hydrogen bonding Morse potential
pair_style kim - interface to potentials provided by KIM project
pair_style lcbop - long-range bond-order potential (LCBOP)
pair_style line/lj - LJ potential between line segments
pair_style lj/charmm/coul/charmm - CHARMM potential with cutoff Coulomb
pair_style lj/charmm/coul/charmm/implicit - CHARMM for implicit solvent
pair_style lj/charmm/coul/long - CHARMM with long-range Coulomb
pair_style lj/charmm/coul/msm - CHARMM with long-range MSM Coulombics
pair_style lj/class2 - COMPASS (class 2) force field with no Coulomb
pair_style lj/class2/coul/cut - COMPASS with cutoff Coulomb

LAMMPS: pair style

pair_style lj/class2/coul/long - COMPASS with long-range Coulomb
pair_style lj/cubic - LJ with cubic after inflection point
pair_style lj/cut - cutoff Lennard-Jones potential with no Coulomb
pair_style lj/cut/coul/cut - LJ with cutoff Coulomb
pair_style lj/cut/coul/debye - LJ with Debye screening added to Coulomb
pair_style lj/cut/coul/dsf - LJ with Coulombics via damped shifted forces
pair_style lj/cut/coul/long - LJ with long-range Coulombics
pair_style lj/cut/coul/long/cs - LJ with long-range Coulombics and core/shell
pair_style lj/cut/coul/msm - LJ with long-range MSM Coulombics
pair_style lj/cut/dipole/cut - point dipoles with cutoff
pair_style lj/cut/dipole/long - point dipoles with long-range Ewald
pair_style lj/cut/tip4p/cut - LJ with cutoff Coulomb for TIP4P water
pair_style lj/cut/tip4p/long - LJ with long-range Coulomb for TIP4P water
pair_style lj/expand - Lennard-Jones for variable size particles
pair_style lj/gromacs - GROMACS-style Lennard-Jones potential

LAMMPS: pair style

pair_style lj/gromacs/coul/gromacs - GROMACS-style LJ and Coulombic potential
pair_style lj/long/coul/long - long-range LJ and long-range Coulombics
pair_style lj/long/dipole/long - long-range LJ and long-range point dipoles
pair_style lj/long/tip4p/long - long-range LJ and long-range Coulomb for TIP4P water
pair_style lj/smooth - smoothed Lennard-Jones potential
pair_style lj/smooth/linear - linear smoothed Lennard-Jones potential
pair_style lj96/cut - Lennard-Jones 9/6 potential
pair_style lubricate - hydrodynamic lubrication forces
pair_style lubricate/poly - hydrodynamic lubrication forces with polydispersity
pair_style lubricateU - hydrodynamic lubrication forces for Fast Lubrication Dynamics
pair_style lubricateU/poly - hydrodynamic lubrication forces for Fast Lubrication with polydispersity
pair_style meam - modified embedded atom method (MEAM)

LAMMPS: pair style

pair_style mie/cut - Mie potential
pair_style morse - Morse potential
pair_style nb3b/harmonic - nonbonded 3-body harmonic potential
pair_style nm/cut - N-M potential
pair_style nm/cut/coul/cut - N-M potential with cutoff Coulomb
pair_style nm/cut/coul/long - N-M potential with long-range Coulombics
pair_style peri/eps - peridynamic EPS potential
pair_style peri/lps - peridynamic LPS potential
pair_style peri/pmb - peridynamic PMB potential
pair_style peri/ves - peridynamic VES potential
pair_style polymorphic - polymorphic 3-body potential
pair_style reax - ReaxFF potential
pair_style rebo - 2nd generation REBO potential of Brenner
pair_style resquared - Everaers RE-Squared ellipsoidal potential
pair_style snap - SNAP quantum-accurate potential
pair_style soft - Soft (cosine) potential

LAMMPS: pair style

pair_style sw - Stillinger-Weber 3-body potential
pair_style table - tabulated pair potential
pair_style tersoff - Tersoff 3-body potential
pair_style tersoff/mod - modified Tersoff 3-body potential
pair_style tersoff/zbl - Tersoff/ZBL 3-body potential
pair_style tip4p/cut - Coulomb for TIP4P water w/out LJ
pair_style tip4p/long - long-range Coulombics for TIP4P water w/out LJ
pair_style tri/lj - LJ potential between triangles
pair_style vashishta - Vashishta 2-body and 3-body potential
pair_style yukawa - Yukawa potential
pair_style yukawa/colloid - screened Yukawa potential for finite-size particles
pair_style zbl - Ziegler-Biersack-Littmark potential