

機器學習於材料資訊的應用

Machine Learning on Material Informatics

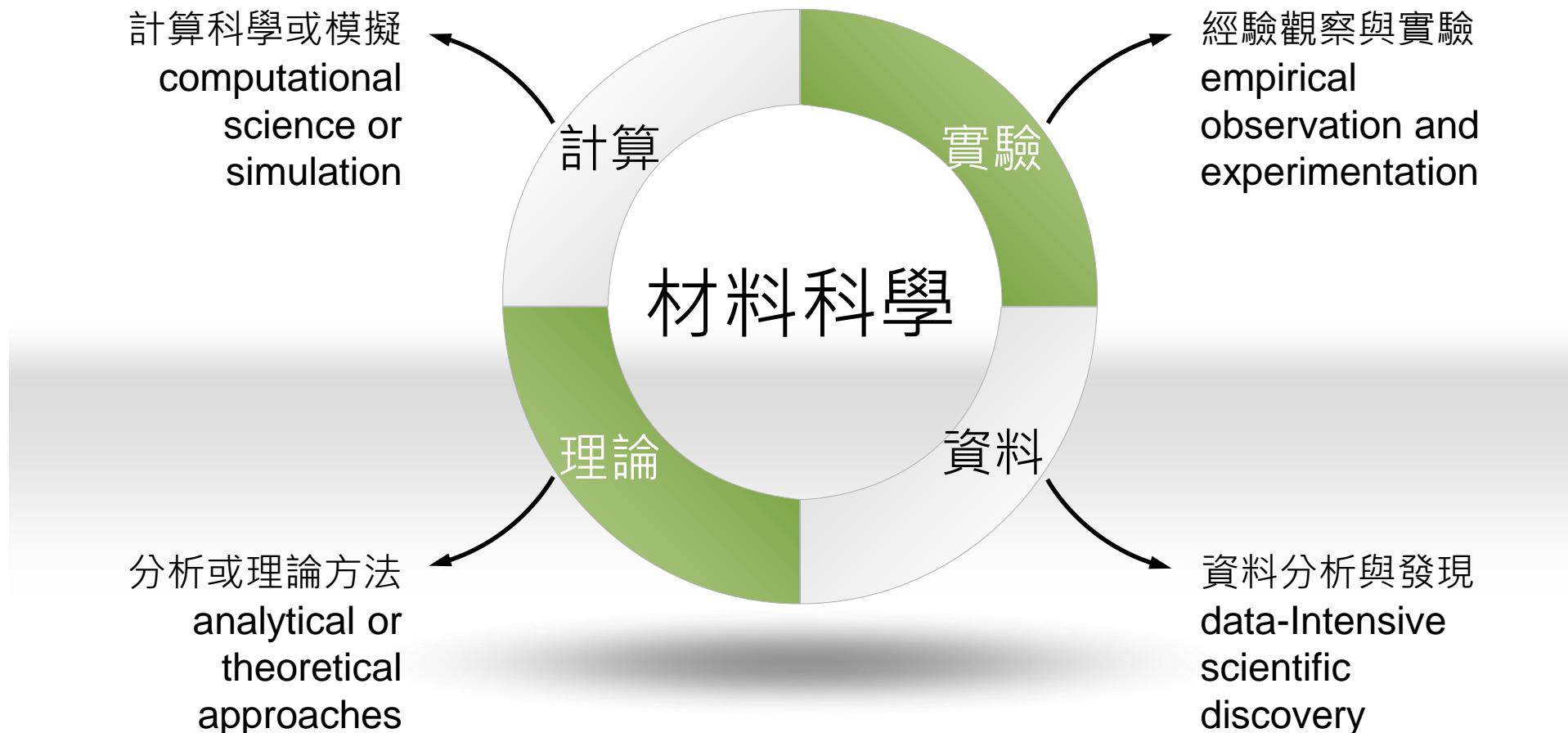
陳南佑(NAN-YOW CHEN)

nanyow@narlabs.org.tw

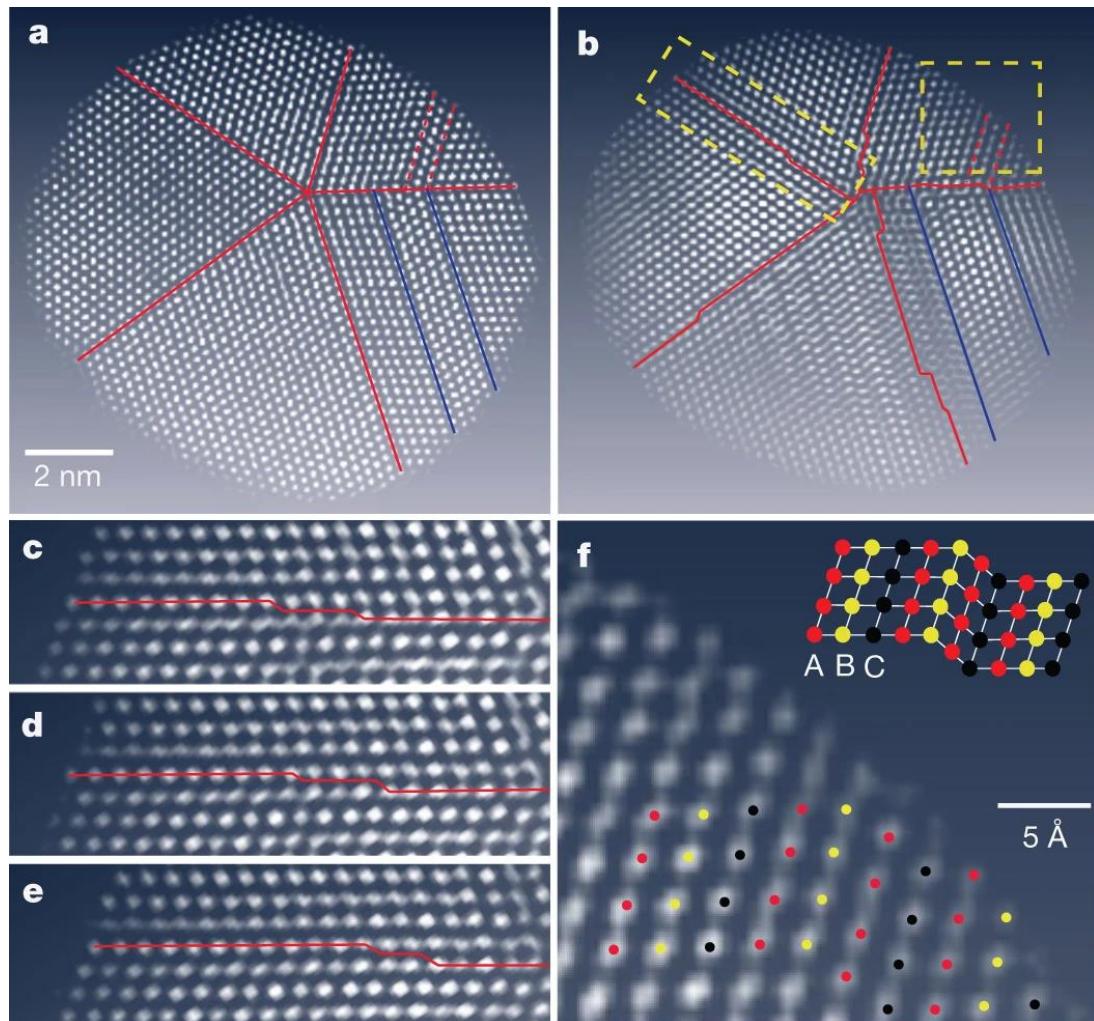
楊安正(AN-CHENG YANG)

acyang@narlabs.org.tw

材料科學的分支

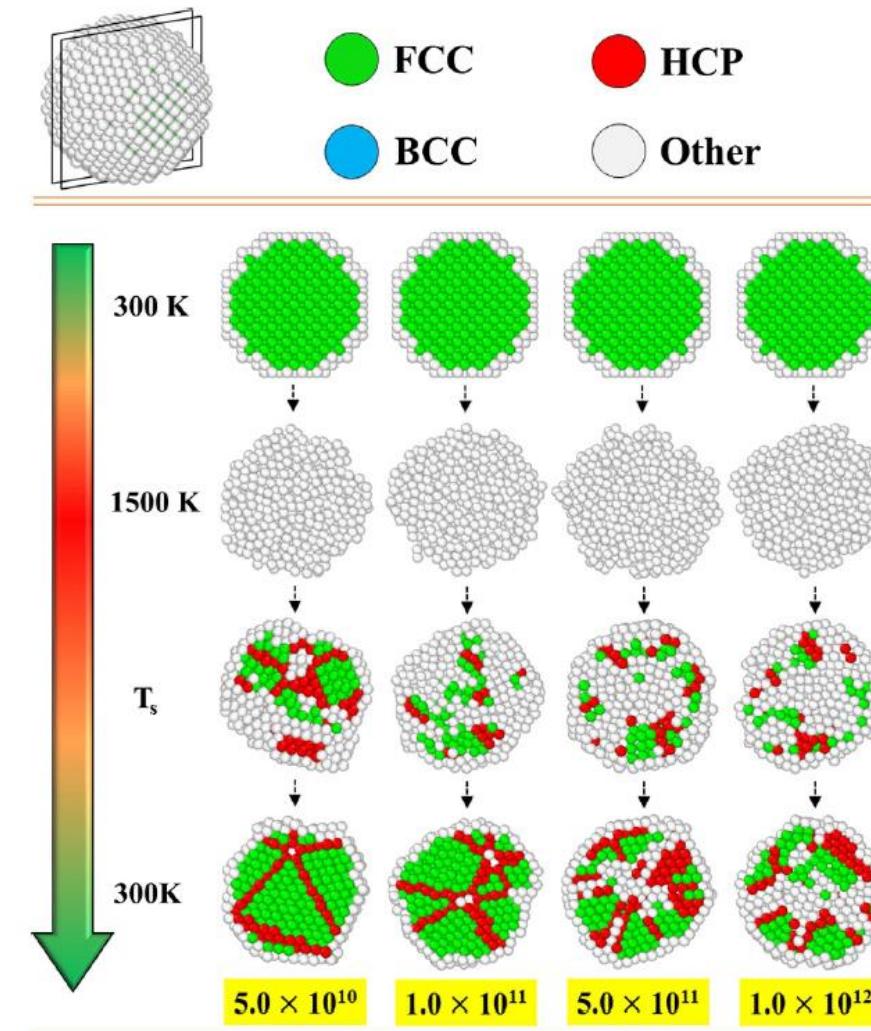


實驗與測量



Chen, C., Zhu, C., White, E. et al. Three-dimensional imaging of dislocations in a nanoparticle at atomic resolution. *Nature* 496, 74–77 (2013).

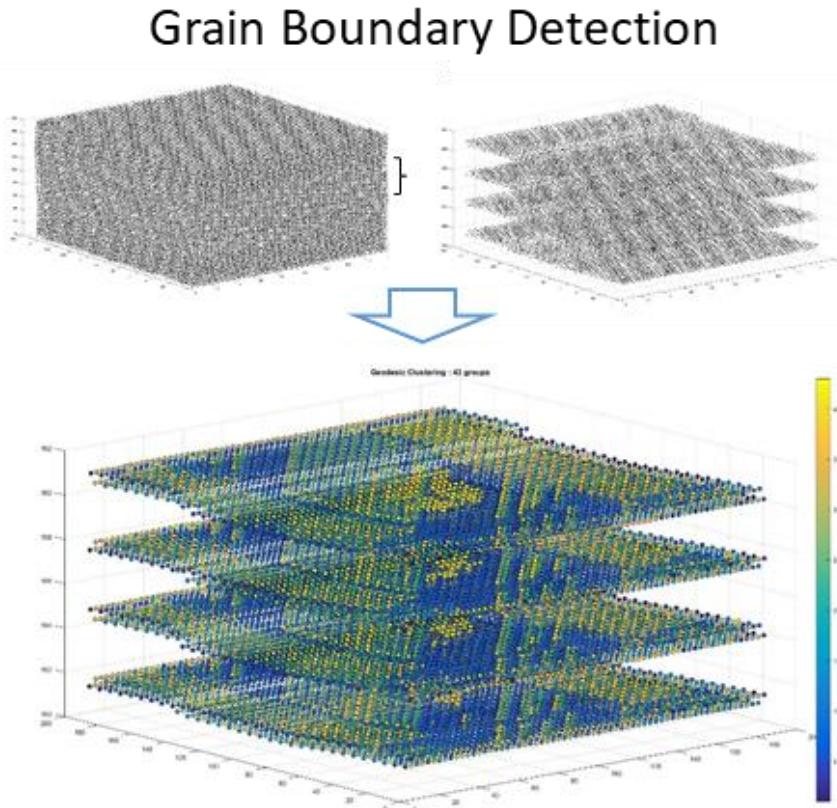
計算或模擬



Vo, T.Q., Kim, B. A molecular dynamics study on cooling rate effect on atomic structure of solidified silver nanoparticles. *Eur. Phys. J. D* 73, 183 (2019).

Case1-晶粒邊界辨識

晶粒邊界分群機器

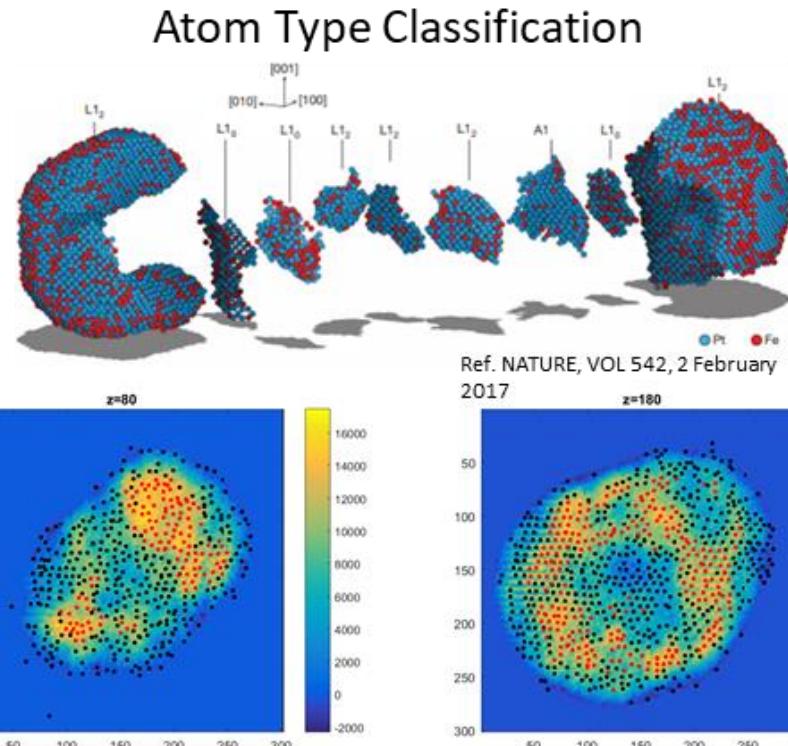


合作計畫	無
合作團隊	交大材料鄒年棣實驗室

案例	透過非監督式學習，完成晶粒邊界辨識、結構分群
客戶目標	晶粒邊界辨識，達成結構分群，以進行區塊結構相似度分析，並與人為經驗公式相互驗證
問題困難描述	人工觀察大量模擬結果資料耗工費時，傳統分群方法需要先前知識才能協助分群。
訓練資料來源	分子動力學模擬結果
機器學習引擎	Modularity
結果	針對晶粒邊界辨識與其結構達成自動分群，與人工經驗法則結果一致。
Status	會議論文已發表 論文撰寫中

Case2-奈米粒子影像重建

種類與缺陷型態深度辨識機器

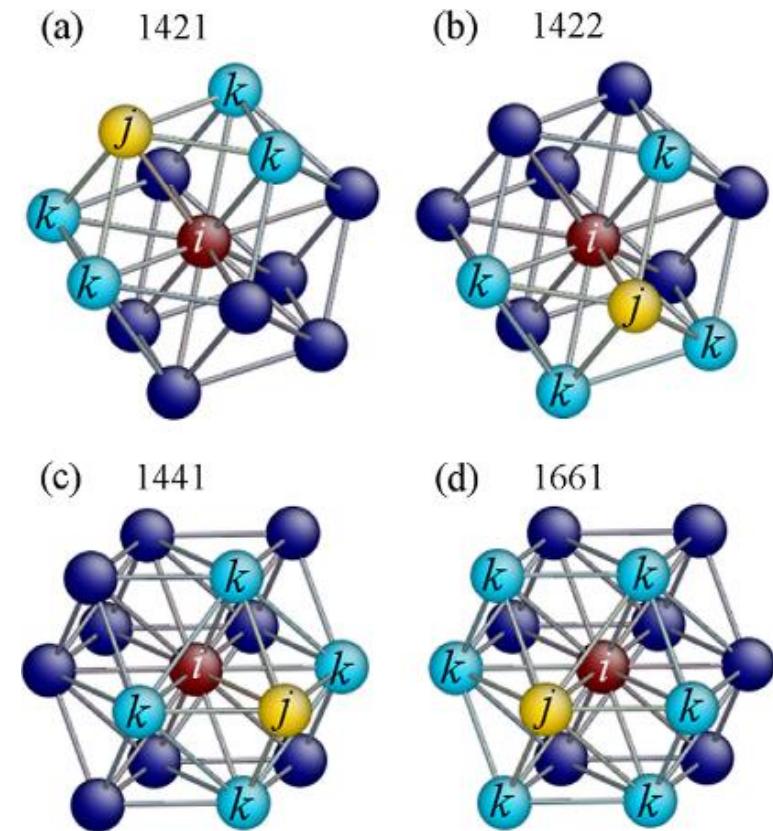
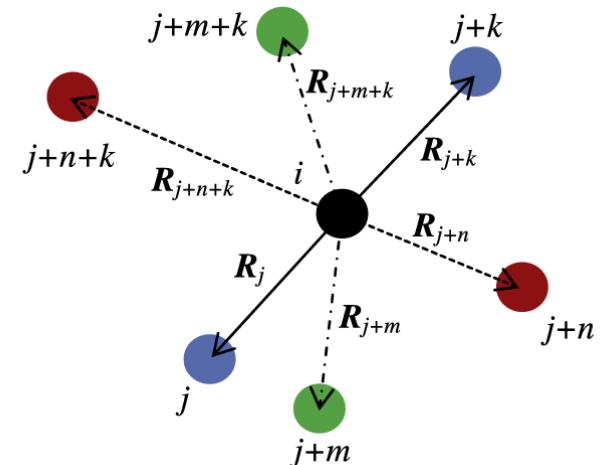
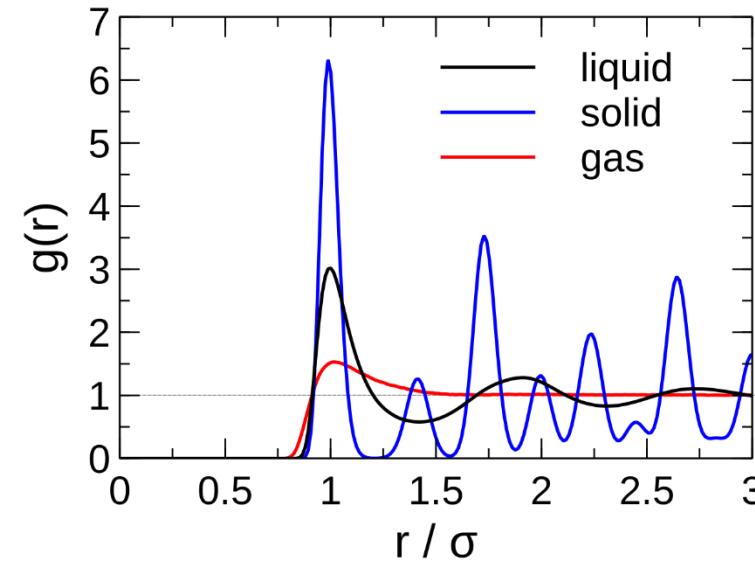


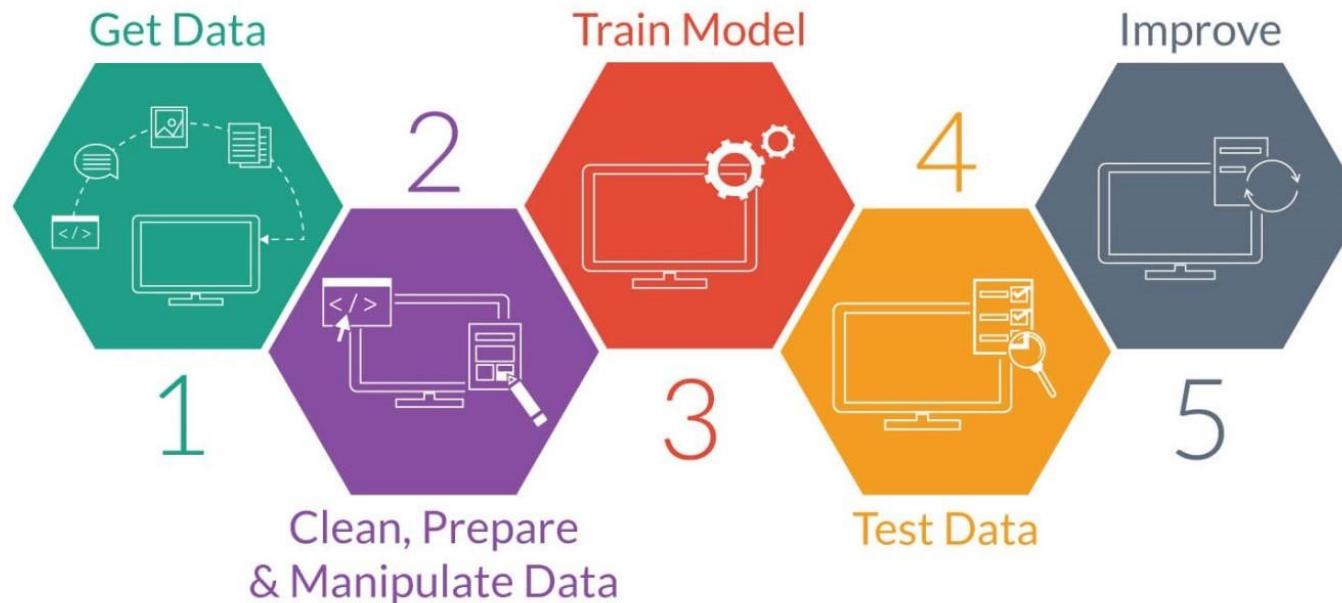
合作計畫	國研院創新計畫
合作團隊	清大工科陳健群實驗室

案例	三維斷層造影之原子種類與缺陷型態辨識
客戶目標	透過深度學習技術，定位原子座標、辨識原子種類、分析缺陷型態
問題困難描述	需要人工判斷原子種類與缺陷邊界，會因人為偏見造成不一致性的誤判。
訓練資料來源	原子級三維斷層造影顯微技術之實驗資料
機器學習引擎	Convolutional Neural Network, Ensemble Learning, Active Learning
結果	成功完成原子座標定位、種類辨識、缺陷邊界型態與相似度分析。降低人為誤判機會。
Status	通過實驗組驗證 論文撰寫中(Nature Materials)

Classification of Crystallographic Groups by Machine Learning

A new powerful approach helping us to identify the microstructure.





使用軟體產生資料

檔案處理

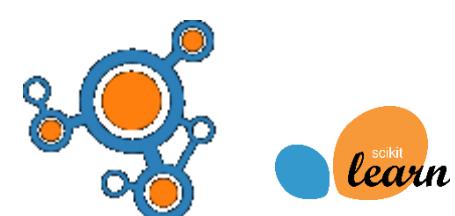
特徵萃取



ASE

建立網路

分群演算法



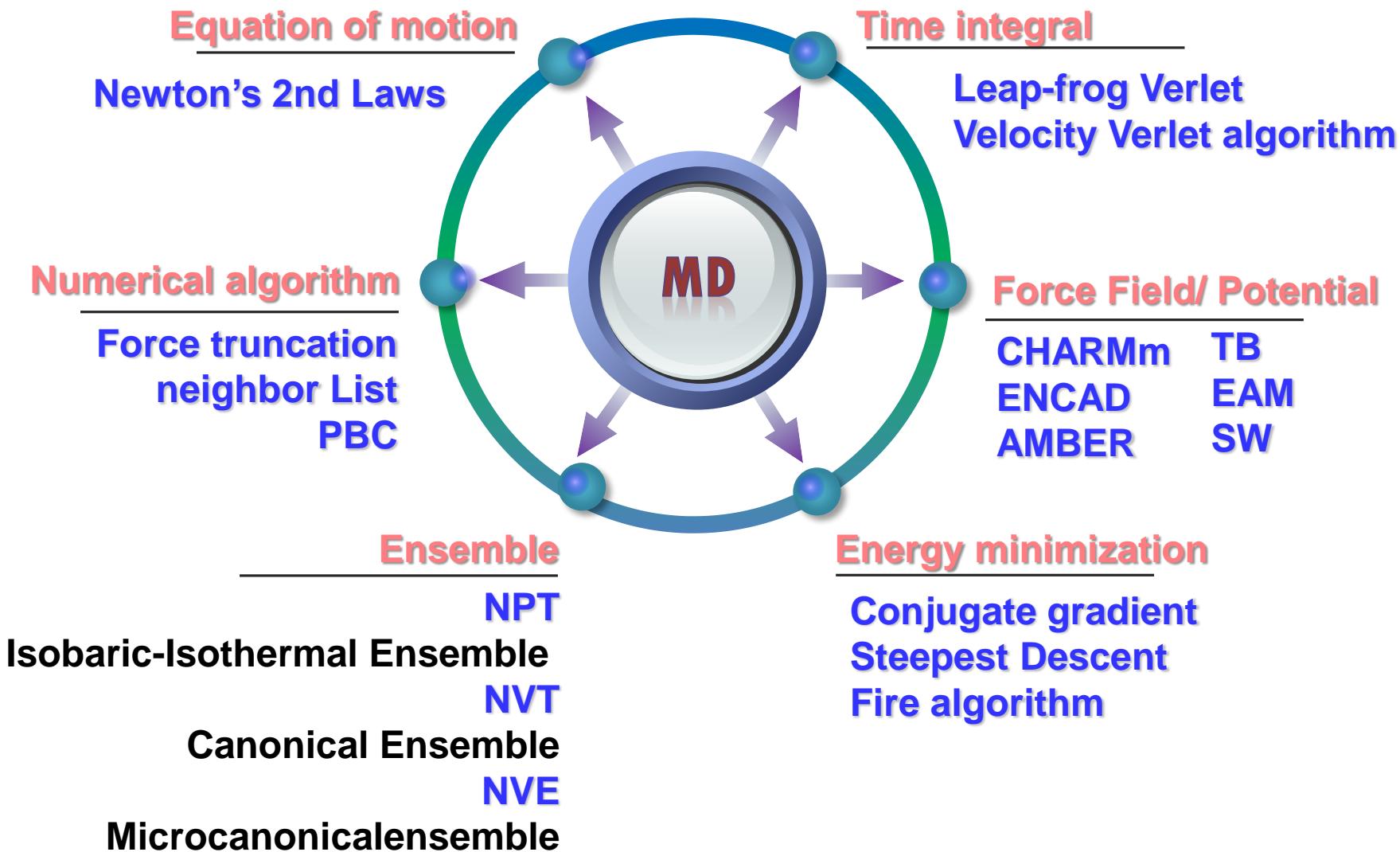
用測試資料
檢驗演算法

調整萃取特徵
方法

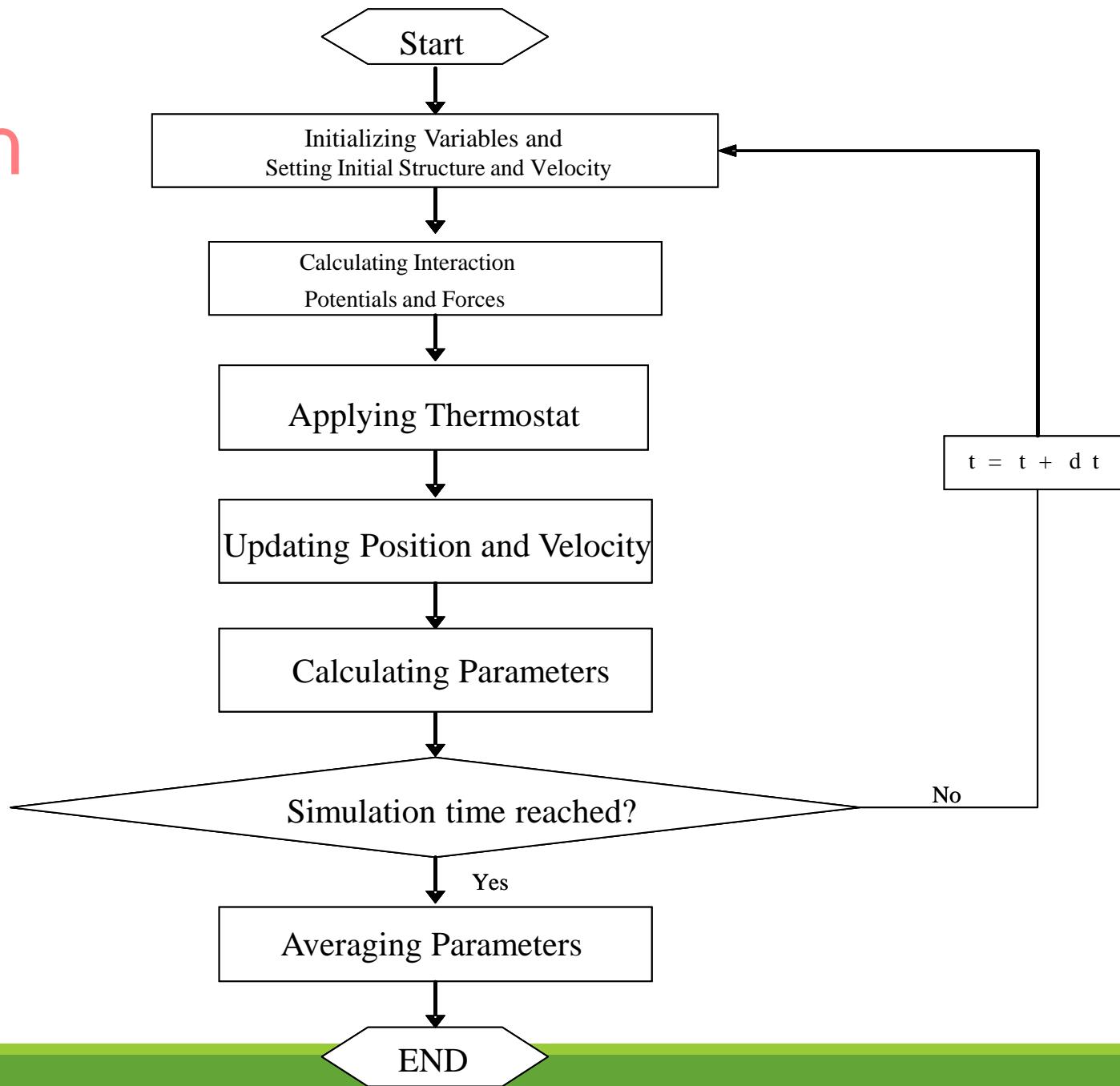
Molecular Dynamics (MD)

- A computer simulation technique that allows one to predict the time evolution of a system of interacting particles (atoms, molecules,.. etc.)
- N-body simulation
- 利用一個描述原子間作用關係來描述一個系統。
- 系統內的所有原子遵循著牛頓運動定律。

Quantum Molecular Dynamics; Classical Molecular Dynamics



Flow Chart of MD simulation



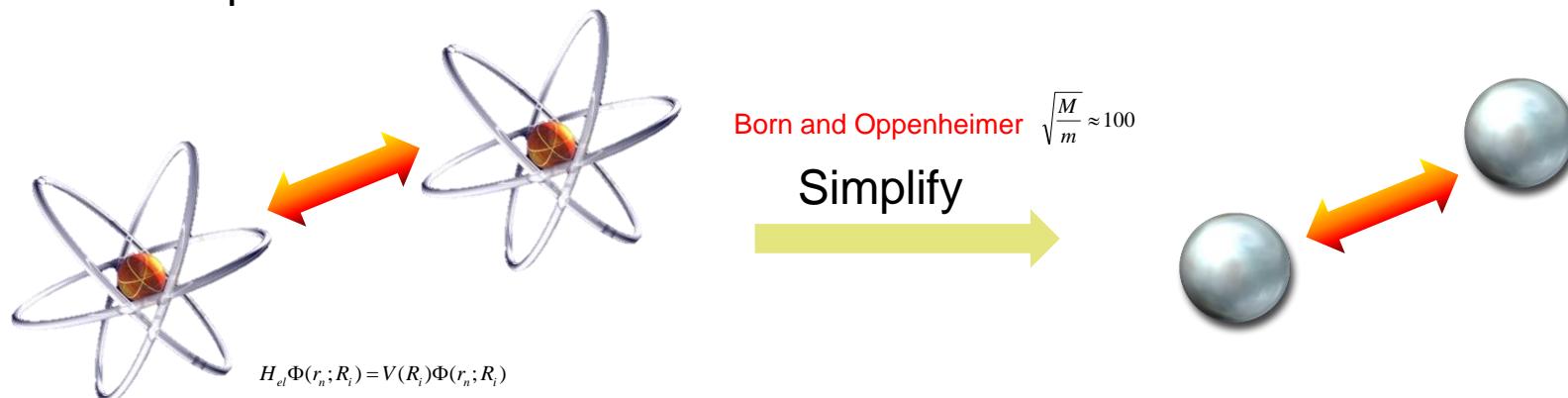
Initial velocity and energy conservation

- Velocities are randomly assigned to the atoms. To achieve faster equilibration atoms can be assigned velocities with the expected equilibrium velocity, i.e. Maxwell distribution.
- Momentum and energy are to be conserved throughout the simulation period.
- Energy conservation is sensitive to the choice of integration method and size of the time step.
- Angular momentum conservation.

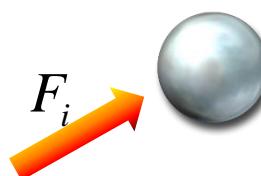
Equation of motion

How Does Molecular Dynamics Work?

In the molecular dynamics method, each atom is treated as a point mass in space



Once the force on each atom is computed, atomic motion is determined through application of Newton's Laws



$$F_i(t) = m_i \frac{d^2 r_i(t)}{dt^2} = -\frac{\partial \Phi(r_i)}{\partial r_i}$$

Second-order ordinary differential equation which can be numerically integrated to find new atomic positions!

Time integration

First-order

Second-order

High-order

Leap-frog Verlet algorithm

$$r(t + \delta t) = r(t) + v(t + \frac{1}{2} \delta t) \delta t$$

$$v(t + \frac{1}{2} \delta t) = v(t - \frac{1}{2} \delta t) + a(t) \delta t$$

Runge–Kutta methods

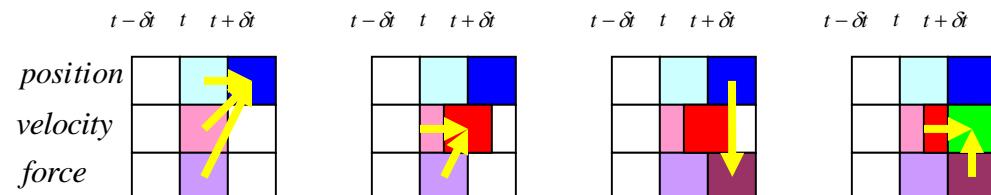
SSPRK3

...

Velocity Verlet algorithm

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

$$v(t + \delta t) = v(t) + \frac{1}{2} (a(t) + a(t + \delta t)) \delta t$$



Ensembles

An ensemble is a collection of all possible systems which have different microscopic states but have an identical macroscopic or thermodynamic states. There exist different ensembles with different characteristics.

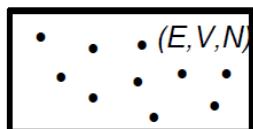
- **Microcanonical ensemble** (NVE) : The thermodynamic state characterized by a fixed number of atoms, N, a fixed volume, V, and a fixed energy, E. ([isolated system](#))

- **Canonical Ensemble** (NVT): This is a collection of all systems whose thermodynamic state is characterized by a fixed number of atoms, N, a fixed volume, V, and a fixed temperature, T. ([system in thermostat](#))

- **Isobaric-Isothermal Ensemble** (NPT): This ensemble is characterized by a fixed number of atoms, N, a fixed pressure, P, and a fixed temperature, T. ([system in barostat](#))

- Microcanonical ensemble

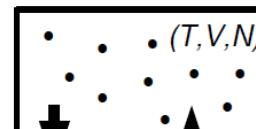
isolated system



*energy transfer not allowed
E constant*

- Canonical ensemble

heat reservoir T



*energy transfer allowed
T constant*

Temperature Control

- Velocity Rescaling

This method is appropriate and efficient for a **homogeneous** system or an inhomogeneous system with **slight difference of mass among atoms**. In the latter case, this method is only appropriate in initial stages of equilibrating the system to relax the random noise of initial velocity.

- Weak coupling method :Berendsen Thermostat / Brownian motion

This method does generate a fluctuation at about the desired temperature, but has disadvantages similar to the rescaling method.

- Stochastic collision method : Andersen thermostat

The drawback of this method is that it cannot generate a smooth trajectory and has **less realistic dynamics**. Therefore, it has been rarely used for molecular dynamics simulations.

- Extended system coupling (Extended Lagrangian) method :

Although the addition of the extended Lagrangian has definitely carried out a correct canonical distribution in property and produced a fluctuation of energy, this assumption of an extended system still cannot make a connection to statistical mechanics.

Temperature Control

- Velocity Rescaling $V_i^{\text{new}} = V_i^{\text{old}} \lambda$ $\lambda = \sqrt{T_d / T_\alpha}$
 - Weak coupling method : Berendsen Thermostat / Brownian motion
-

$$m\dot{v} = F_i - m_i \gamma_i v_i + R(t) \quad (\text{Langevin's equation})$$

- Stochastic collision method : Andersen thermostat
- Extended system coupling (Extended Lagrangian) method :
 - The Nose-Hoover extended system method

$$\boxed{\begin{aligned}\dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} \\ \dot{\mathbf{p}}_i &= -\nabla_i V - \zeta \mathbf{p}_i \\ \zeta &= \frac{\left(\sum_i \frac{\mathbf{p}_i^2}{m_i} - g k_B T \right)}{Q} = \nabla_T \left[\frac{\sum_i \frac{\mathbf{p}_i^2}{m_i}}{g k_B T} - 1 \right] = \nabla_T \left[\frac{T^*}{T} - 1 \right]\end{aligned}}$$

Notice that: (1) for $T^* > T$ (system too hot), ζ will increase (if it is positive the system cools down)
(2) for $T^* < T$ (system too cold), ζ will decrease (if it is negative the system heat up)

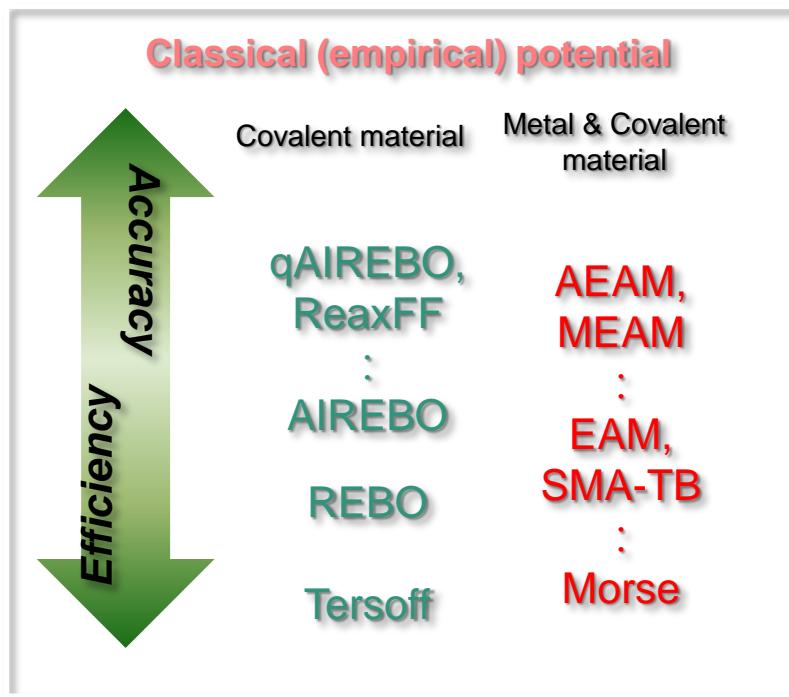
Force Field

In the context of molecular modeling, a force field refers to the form and parameters of mathematical functions used to describe the potential energy of a system of particles (typically molecules and atoms). Force field functions and parameter sets are derived from both experimental work and high-level quantum mechanical calculations. "All-atom" force fields provide parameters for every type of atom in a system.

The usage of the term "force field" in chemistry and computational biology differs from the standard usage in physics. In chemistry it is a system of potential energy functions rather than the gradient of a scalar potential.

To be effective, an analytic Force Field (potential) must possess the following critical properties:

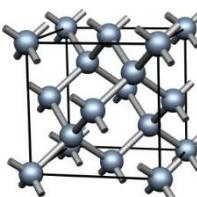
- Accuracy (reproduce properties of interest as closely as possible)
- Computational speed (calculations are fast with simple potentials)
- Transferability (can be used to study a variety of properties for which it was not fit)
- Flexibility (accommodate a wide range of structures in a fitting database)



The potential parameters in the force field are derived from

- ab-initio quantum mechanics
- Crystallography.
- Spectroscopy
-

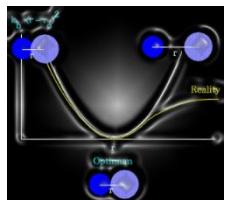
$$\nu = \frac{1}{2\pi} \sqrt{\frac{K}{\mu}}$$



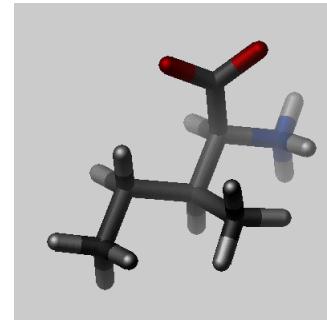
Nonreactive force field (AMBER,CHARMM...)

$$U_{total} = U_{bond} + U_{bending} + U_{dihedral} + U_{vdw} + U_{col}$$

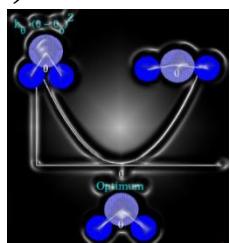
(a) Bond strength



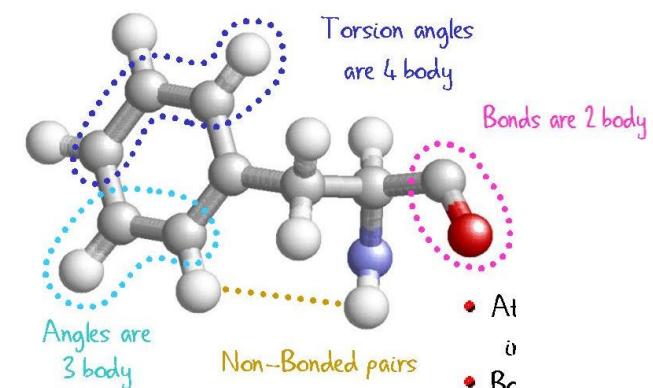
$$U_{bond\ length} = \sum_{N=1}^{N_0} \frac{1}{2} K_b^i (b_i - b_0^i)^2$$



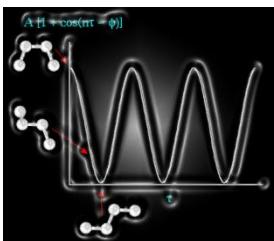
(b) Bond bending



$$U_{bending} = \sum_{N=1}^{N_0} \frac{1}{2} K_\theta^i (\theta_i - \theta_0^i)^2$$



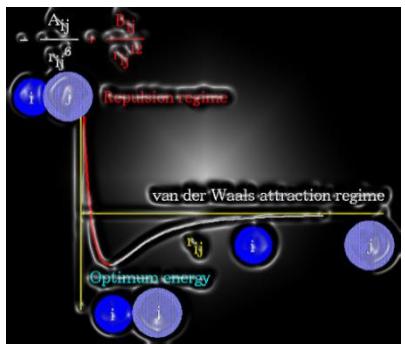
(c) Dihedral Angle



$$U_{dihedral} = \sum_{N=1}^{N_\phi} K_\phi^i \left\{ 1 - \cos[N^i (\varphi_i - \varphi_0^i)] \right\}$$

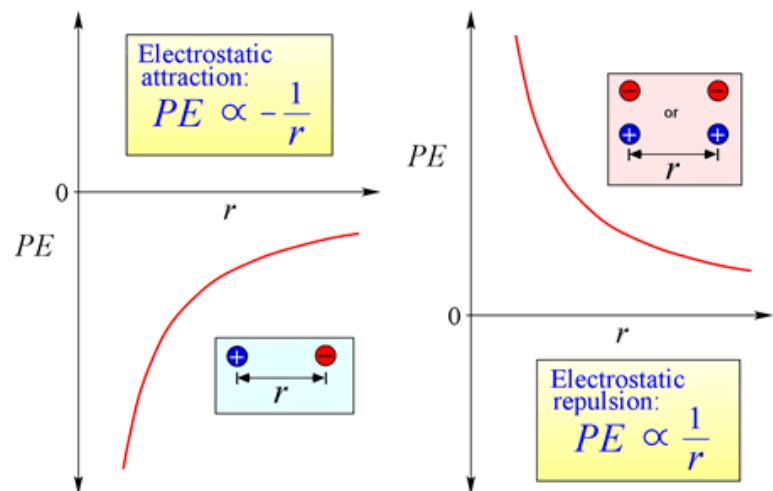
- Atoms are connected in a simple graph.
- Bonded Interactions: 2, 3, 4 body.
- Non-Bonded Interactions.

(d) Van der Waal



$$U_{\text{vdw}} = \left[\epsilon^{ij} \left(\frac{r_0^{ij}}{r_{ij}} \right)^{12} - 2 \epsilon^{ij} \left(\frac{r_0^{ij}}{r_{ij}} \right)^6 \right]$$

(e) Coulomb force



$$U_{\text{col}} = \sum_{\text{partialcharges}} \frac{q^i q^j}{r_{ij} \epsilon}$$

Treating Long-Range Electrostatic Interactions
particle-mesh Ewald (PME); Ewald sum

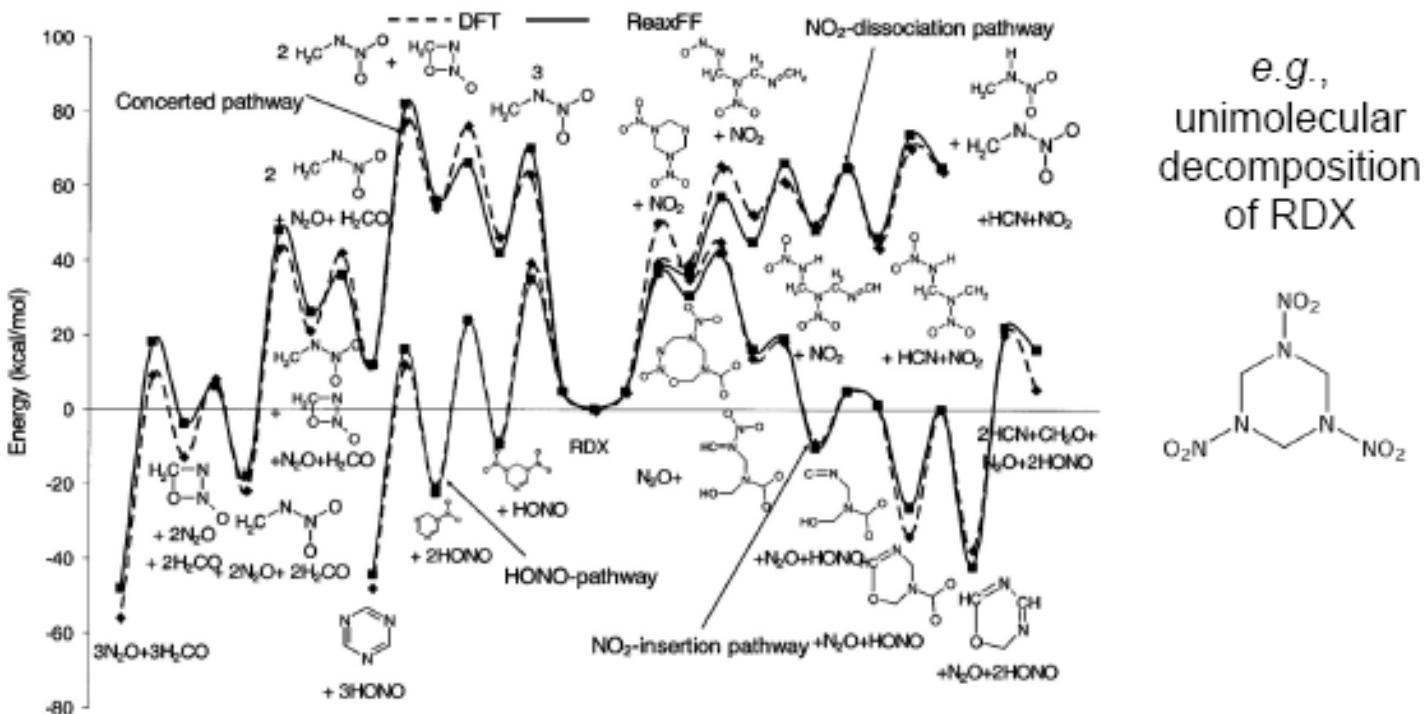
ReaxFF Potential Energy Function

$$E_{\text{system}} = E_{\text{bond}} + E_{\text{lp}} + E_{\text{over}} + E_{\text{under}} + E_{\text{val}} + E_{\text{pen}} + E_{\text{coa}} + E_{\text{C2}} + E_{\text{triple}} + E_{\text{tors}} + \\ E_{\text{conj}} + E_{\text{H-bond}} + E_{\text{vdWaals}} + E_{\text{Coulomb}}$$

- E_{bond} : bond energy; attractive term, directly derived from bond orders
- E_{lp} : Lone pair energy; penalty for breaking up lone pairs in O, N
- E_{over} : Overcoordination energy: penalty for overcoordinating atoms
- E_{under} : Undercoordination energy: stabilizes undercoordinated atoms
- E_{val} : Angle strain; equilibrium angle depends on bond order central atom
- E_{pen} : Penalty for 'allene'-type molecules (H2C=C=CH2)
- E_{coa} : Angle conjugation; stabilizes -NO2 groups
- E_{C2} : C2 correction: destabilizes C=C
- E_{triple} : triple bond related, first mentioned in publications from 2008
- E_{tors} : Torsion energy: bond-order dependent V2-term
- E_{conj} : Torsion conjugation: general conjugation stability
- $E_{\text{H-bond}}$: Hydrogen bond
- E_{vdWaals} : van der Waals: calculated between every atom
- E_{Coulomb} : Coulomb interaction: calculated between every atom; polarizable charges get updated every iteration

ReaxFF/*Ab Initio* Comparison

ReaxFF can describe a wide variety of chemical reactions.



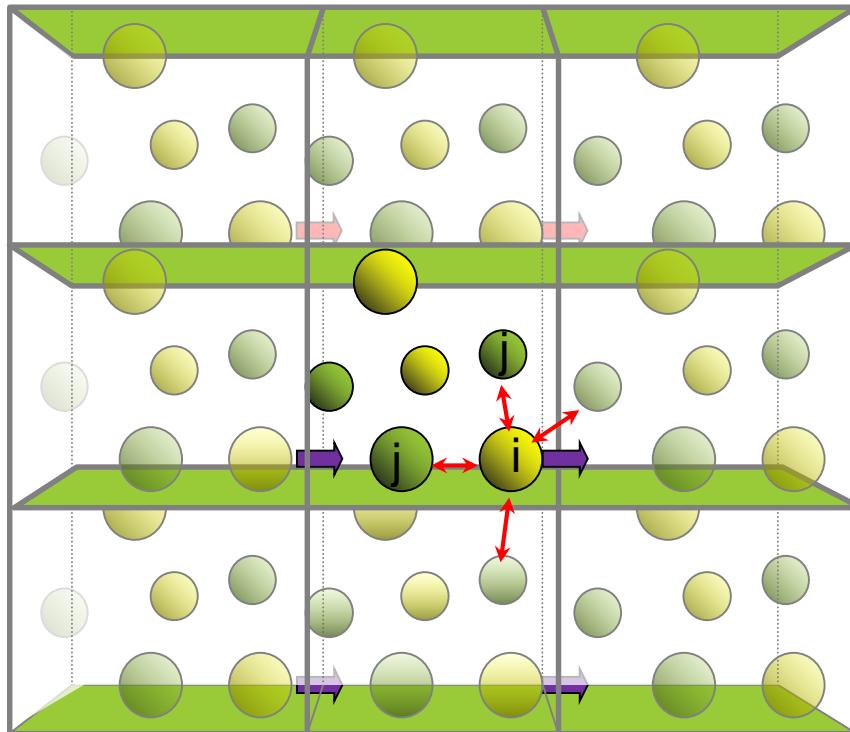
Strachan, et al, JCP, 122, 054502 ('05).

(Original slide from Quenneville presentation)

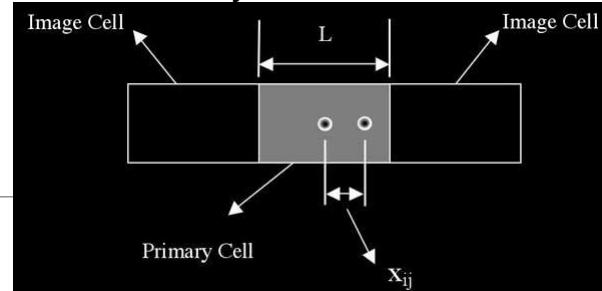
Numerical algorithm

- Periodic Boundary Condition
- Minimum Image Criterion

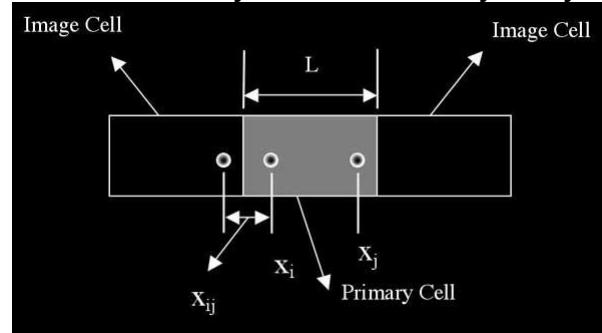
Case B: $x_{ij} < -0.5L \rightarrow x_{ij} = x_{ij} + L$



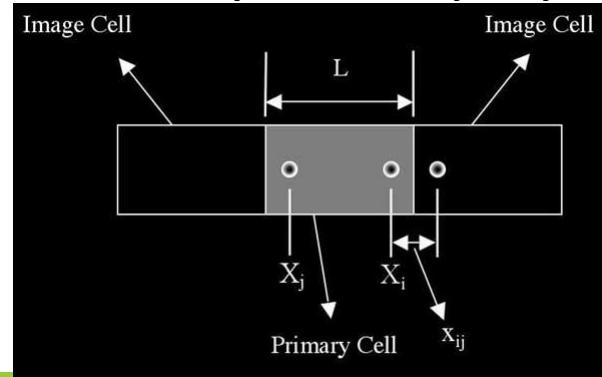
Case A: x_{ij} is less than $0.5L$



Case B: $x_{ij} < -0.5L \rightarrow x_{ij} = x_{ij} + L$



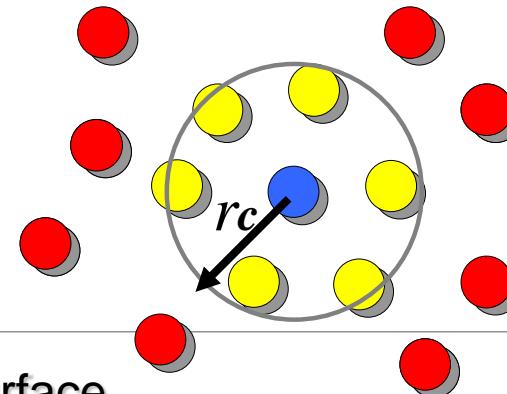
Case C: $x_{ij} > 0.5L \rightarrow x_{ij} = x_{ij} - L$



Force Truncation

● Truncation

$$U(r) = \begin{cases} U(r) & r < R_c \\ 0 & r \geq R_c \end{cases}$$



- **Shift function:** modifies the entire potential energy surface

$$U_{nb}(r) = \begin{cases} U_{nb}(r) - \left[U_{nb}(R_C) + (r - R_C) \left[\frac{dU_{nb}(R_C)}{dr} \right] \right] & r < R_C \\ 0 & r \geq R_C \end{cases} \dots \text{Levitt et. al.}$$

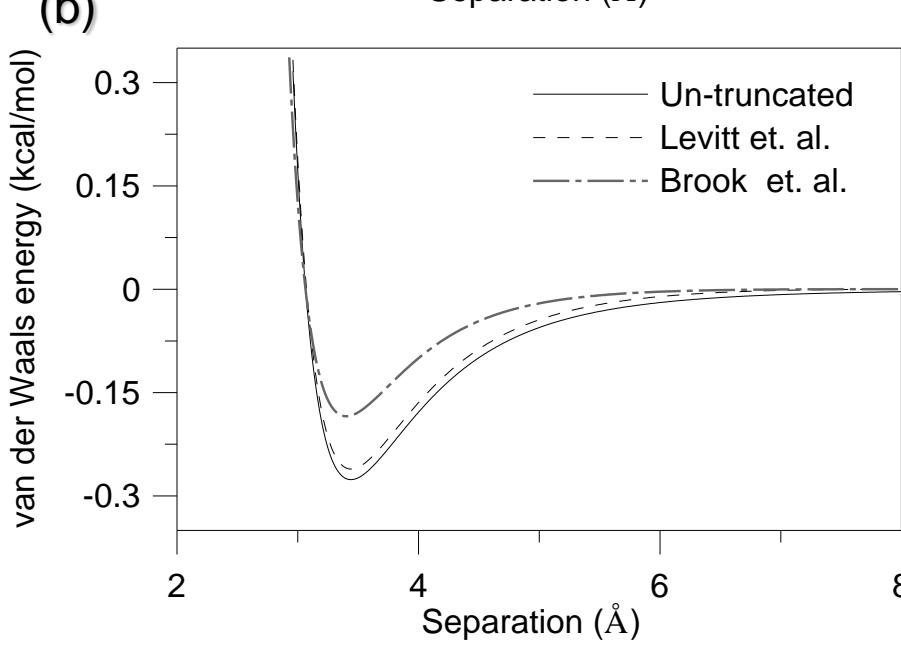
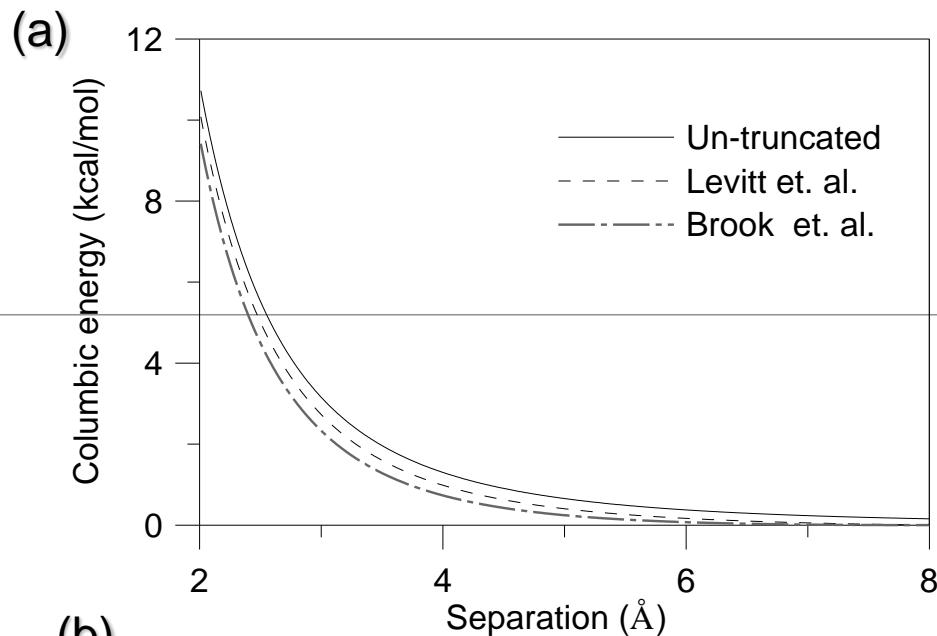
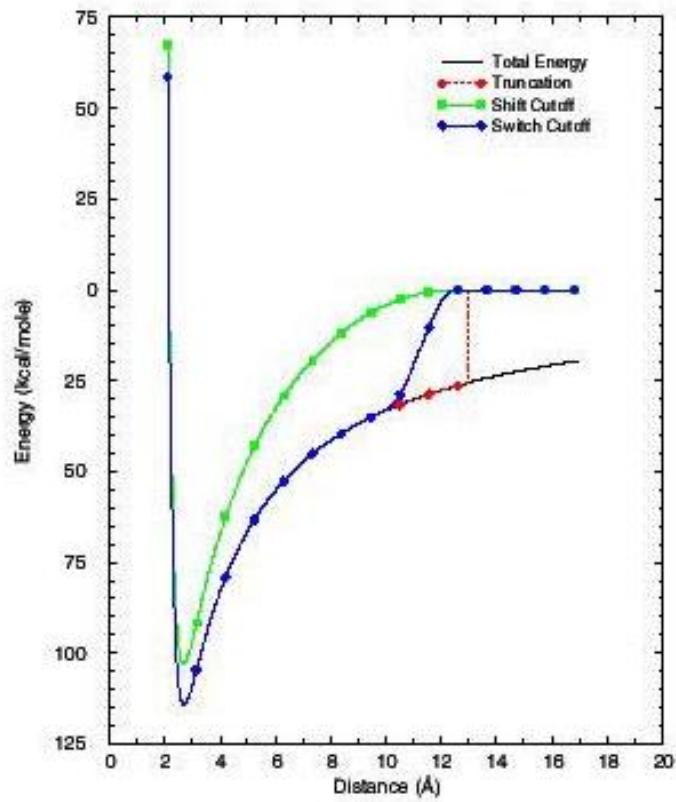
- **Switch function:** modifies the interaction potential over a predefined range of distances

$$U_{nb}(r) = \begin{cases} U_{nb}(r) & r < r_l \\ U_{nb}(r) \left\{ \frac{(r_c - r)^2(r_c + 2r - 3r_l)}{(r_c - r_l)^3} \right\} & r_l \leq r < r_c \\ 0 & r_c \leq r \end{cases}$$

CHARMm

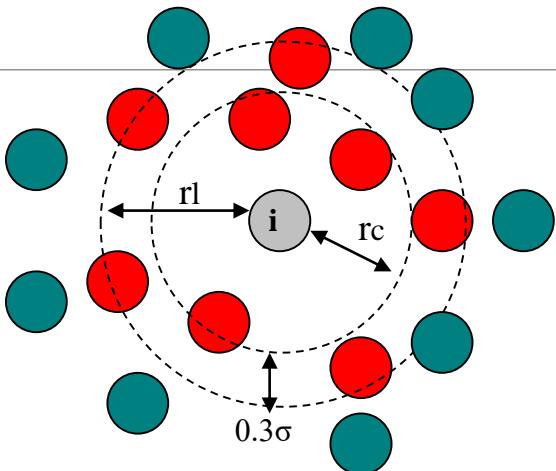
● The Ewald summation technique

(a) Columbic energy profile between two water molecules and (b) the van der Waals energy profile between oxygen atom in C=O group and nitrogen atom in N-H group with and without truncation methods.

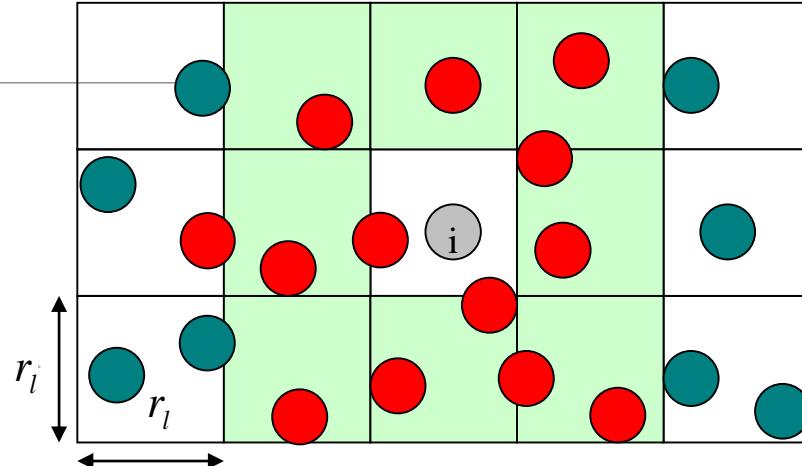


Efficient Neighbours List

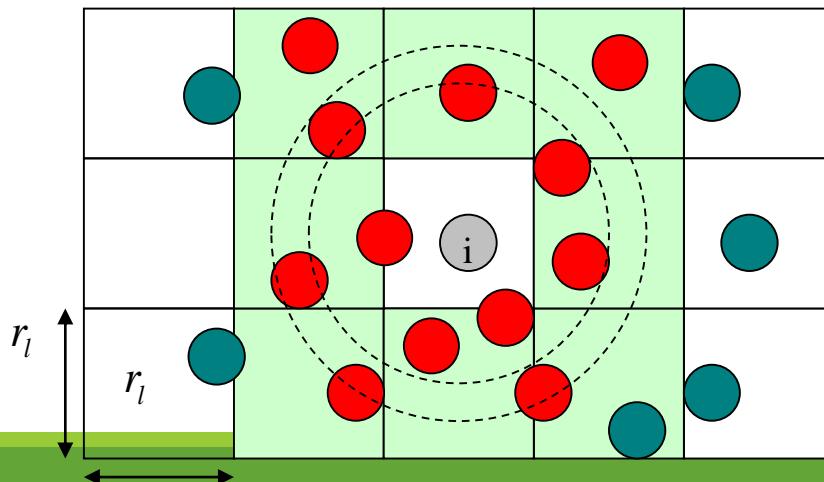
● Verlet List



● Cell Link



● Verlet List + Cell Link



Packages for molecular modeling

Package Name	Developer	Features
LAMMPS	Sandia National Labs	atomic, polymeric, biological, metallic, granular, and coarse-grained systems
NAMD	Parallel Programming Labs	structural biology
GROMACS	Science for Life Labs	general purpose
DL_POLY	Daresbury Labs	general purpose
Desmond	D. E. Shaw Research	biological and chemical systems
AMBER	Peter Kollman et. Al.	biomolecules
SPaSM	Los Alamos National Labs	Large scale simulation

Installation of Windows pre-built binary (ICMS version)

weppage Screenshot

LAMMPS-ICMS Windows Installer Repository

This repository is hosting pre-compiled Windows installers of the [LAMMPS](#) molecular dynamics simulation software package. The binaries are built semi-automatically with MinGW64 Linux to Windows cross compilers using up-to-date snapshots of the [LAMMPS-ICMS git repository](#) hosted at the [Institute for Computational Molecular Science](#) at Temple University. The LAMMPS binaries contain **all** optional packages included in the source distribution **except**: KIM (license is not GPL compatible), USER-CUDA (CUDA does not support cross compilation), KOKKOS (does not support cross-compilation), and REAX (superseded by the USER-REAXC package).



Some Notes on GPU Support

These Windows binaries include (experimental on Windows) GPU acceleration via the [GPU](#) package. This is achieved through compiling the GPU package in OpenCL mode and linking to an [OpenCL v1.2](#) compatible ICD loader. This means the executables do not contain any vendor provided code and should be compatible with GPUs from both [AMD](#) and [Nvidia](#). The GPU package has been compiled for mixed precision computation and is currently somewhat tuned for Nvidia (Fermi generation) GPUs. It does not yet work with OpenCL drivers for CPUs (like those included in the Intel and AMD OpenCL SDKs). Work is under way to improve and expand OpenCL support in LAMMPS and particularly provide means so that the individual kernels are optimized for specific platforms at runtime. Please watch [Mike Brown's LAMMPS accelerator library web page](#) for status updates.

When reporting problems, please always include the exact [version](#) of the installer and the output of the `ocl_get_devices` tool.

Looking for pre-compiled Linux binary RPMs? They are just a [mouse click away](#).

Installing LAMMPS on Windows

There are installer packages for 32-bit and 64-bit versions of Windows available.

- [Latest version for 32-bit Windows](#)
- [Latest version for 64-bit Windows](#)

[32-bit Windows download area](#)
[64-bit Windows download area](#)

The respective download directory will contain installer packages that are labeled with the date they were compiled on and one package labeled as *latest*. It is usually recommended to download and install the latest package via the link above. The other packages are provided in case there is a problem with it. Download the installer executable suitable for your machine, execute it, and follow the instructions in the dialogs. Each version will install into a different directory, so it is possible to have multiple versions installed at the same time. Both kinds of packages contain:

- A regular multi-threaded LAMMPS executable called `lmp_serial`. This should always work.
- A multi-threaded LAMMPS executable that also supports parallel execution via MPI message passing. This executable is called `lmp_mp1` and requires installation of a suitable MPICH2 package to work.
- the LAMMPS manual in PDF format
- the [colvirs](#) reference manual in PDF format
- the potential files bundled with the LAMMPS source code
- most of the example inputs, reference outputs and related files
- the benchmark inputs and reference outputs
- the tools `restart2data`, `binary2txt`, `chain`, `msi2lmp`, `ocl_get_devices`

Both executables will run in serial when executed directly. Please see below for instructions on how to perform parallel runs. To use the MPI based parallelism, you also need to install [MPICH2 from Argonne lab](#). For 32-bit Windows you have to download and install [mpich2-1.4.1p1-win-ia32.msi](#) or any compatible version. Correspondingly, for 64-bit Windows you have to download and install [mpich2-1.4.1p1-win-x86-64.msi](#) or any compatible version.

Running LAMMPS on Windows

General Comments

<http://rpm.lammps.org/windows.html> Fri Aug 15 2014 10:35:07 GMT+0800 (台北標準時間)

Choose executable file based on the architecture of your PC, then double click on it to install. If you want to run LAMMPS in parallel mode, [mpich2-1.4.1p1-win-x86-64.msi](#) is also needed.

LAMMPS Windows Installer Repository

<https://packages.lammps.org/windows.html>



.../LAMMPS-ICMS-20140625/Windows/

Run LAMMPS on Windows

1. Install lammps on your windows system, put the path of lammps exe file(Ex. `lmp_serial.exe`) in the PATH environment variable (Ex. `C:\Program Files\LAMMPS 64-bit 16Aug2018\bin`).
2. Put input file(Ex. `in.xxx`), potential file or coordinate file (optional)in project folder (`X:\project_folder`).
3. Open cmd , and change directory to project folder("X:" then "cd project_folder").
4. Or press shift and right click to open cmd in the project_folder.
5. Use the following command to run lammps "`C:\Program Files\LAMMPS 64-bit 16Aug2018\bin\lmp_serial.exe -in in.xxx`"

LAMMPS input script

設定檔可區分為4個部分

Initialization

Atom Definition

Settings

Run a Simulation

Settings

Run a Simulation

...

命令可以分類成9群

1. Initialization
2. Atom Definition
3. Force Fields
4. Settings
5. Fixes
6. Computes
7. Outputs
8. Actions
9. Miscellaneous

LAMMPS

INITIALIZATION

ATOM DEFINITION

SETTINGS

Force fields

Parameters

Fixes

Computes

Output

RUN

Actions

Miscellaneous

Commands

[atom_modify](#), [atom_style](#), [boundary](#), [dimension](#), [newton](#), [processors](#), [units](#)

[create_atoms](#), [create_box](#), [lattice](#), [read_data](#), [read_dump](#), [read_restart](#), [region](#), [replicate](#)

[angle_coeff](#), [angle_style](#), [bond_coeff](#), [bond_style](#), [dielectric](#), [dihedral_coeff](#), [dihedral_style](#), [improper_coeff](#), [improper_style](#), [kspace_modify](#), [kspace_style](#), [pair_coeff](#), [pair_modify](#), [pair_style](#), [pair_write](#), [special_bonds](#)

[neighbor](#), [neigh_modify](#), [group](#), [timestep](#), [reset_timestep](#), [run_style](#), [min_style](#), [min_modify](#).

[fix](#), [fix_modify](#), [unfix](#)

[compute](#), [compute_modify](#), [uncompute](#)

[dump](#), [dump_image](#), [dump_modify](#), [dump_movie](#), [restart](#), [thermo](#), [thermo_modify](#), [thermo_style](#), [undump](#), [write_data](#), [write_dump](#), [write_restart](#)

[run](#), [temper](#), [minimize](#), [neb_prd](#), [rerun](#)

[delete_atoms](#), [delete_bonds](#), [displace_atoms](#), [change_box](#)

[clear](#), [echo](#), [if](#), [include](#), [jump](#), [label](#), [log](#), [next](#), [print](#), [shell](#), [variable](#)

Command Format

- ✓ *COMMAND* (+ group-ID) (+*args...*) (style (+*args...*)) (+*其它(+ values ...)*) (+ keyword (+*values ...*))

Example:

neb etol ftol N1 N2 Nevery file-style arg

- etol = stopping tolerance for energy (energy units)
- ftol = stopping tolerance for force (force units)
- N1 = max # of iterations (timesteps) to run initial NEB
- N2 = max # of iterations (timesteps) to run barrier-climbing NEB
- Nevery = print replica energies and reaction coordinates every this many timesteps
- file-style=*final* or *each* or *none*

final arg = filename

filename = file with initial coords for final replica coords for intermediate replicas are linearly interpolated between first and last replica

each arg = filename

filename = unique filename for each replica (except first) with its initial coords

none arg = no argument

all replicas assumed to already have their initial coords

- ✓ fix +ID + group-ID + style (+*args...*) (+ keyword (+*values ...*))
- ✓ compute +ID + group-ID + style (+*args...*) (+ keyword (+*values ...*))
- ✓ variable+name + style_name +*args...*

※若*arg* 或*value*為斜字體即為字元(*yes, no, final ...*)，若非則為數字

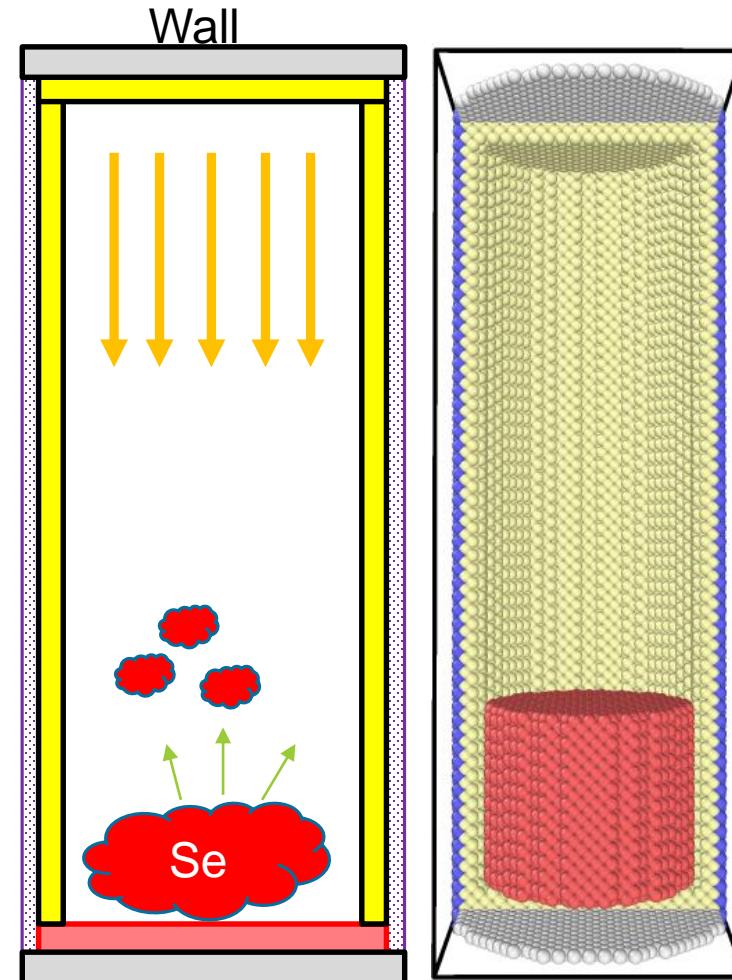
command參數	說明
style(+arg)	<p>擁有多重功能參數之command 會有style參數</p> <ul style="list-style-type: none"> ✓ 功能參數 -style · 名稱有特定 -arg · 部份名稱為特定，部份由使用者給數據，可為數字或字元 ✓ 無指定群組 · e.g.: <ul style="list-style-type: none"> min_style (選擇能量最小化方法) lattice (選擇晶格) pair_style (選擇勢能函數) ✓ 指定群組 · e.g.: <ul style="list-style-type: none"> fix (+ID + group - ID + style_name) compute (+ID + group - ID + style_name)
ID, group-ID, region-ID, fix-ID,	參數中的指定代號,指定群組代號,指定區間代號 , fix代號 (使用者給定)
keyword (+value)	<p>可額外選用之子功能/子條件之參數(可給可不給)</p> <ul style="list-style-type: none"> -keyword · 名稱有特定 -value · 部份名稱為特定，部份由使用者給數據，可為數字或字元
file	檔案名稱 · e.g.: dump ID group-ID style N file args
name	名稱 · e.g.: variable name
其它	以上參數以外的名稱通常為供特定功能所設置之參數名稱，其定義請閱讀 manual

You should know about this

region	area
region-ID	name of region
group	a cluster of atoms
group-ID	Name of group
type	atom type
molecule	molecule id #
id	atom id #

Addforce zone
→

Heating zone



4 group, 2 type

compute command

- centro/atom - centro-symmetry parameter for each atom
- cna/atom - common neighbor analysis (CNA) for each atom
- com - center-of-mass of group of atoms
- com/molecule - center-of-mass for each molecule
- coord/atom - coordination number for each atom
- group/group - energy/force between two groups of atoms
- gyration/molecule - radius of gyration for each molecule
- heat/flux - heat flux through a group of atoms
- inertia/molecule - inertia tensor for each molecule
- ke - translational kinetic energy
- msd - mean-squared displacement of group of atoms
- msd/molecule - mean-squared displacement for each molecule
- pe - potential energy
- pe/atom - potential energy for each atom
- pressure - total pressure and pressure tensor
- rdf - radial distribution function $g(r)$ histogram of group of atoms
- reduce - combine per-atom quantities into a single global value
- reduce/region - same as compute reduce, within a region
- stress/atom - stress tensor for each atom
- temp - temperature of group of atoms
- voronoi/atom - Voronoi volume and neighbors for each atom

fix command (1/2)

addforce - add a force to each atom

ave/atom - compute per-atom time-averaged quantities

ave/time - compute/output global time-averaged quantities

deform - change the simulation box size/shape

deposit - add new atoms above a surface

efield - impose electric field on system

freeze - freeze atoms in a granular simulation

gravity - add gravity to atoms in a granular simulation

gcmc - grand canonical insertions/deletions

heat - add/subtract momentum-conserving heat

indent - impose force due to an indenter

langevin - Langevin temperature control

momentum - zero the linear and/or angular momentum of a group of atoms

move - move atoms in a prescribed fashion

neb - nudged elastic band (NEB) spring forces

npt - constant NPT time integration via Nose/Hoover

npt/asphere - NPT for aspherical particles

npt/sphere - NPT for spherical particles

nve - constant NVE time integration

nvt - constant NVT time integration via Nose/Hoover

fix command (2/2)

nvt - constant NVT time integration via Nose/Hoover

press/berendsen - pressure control by Berendsen barostat

print - print text and variables during a simulation

restrain - constrain a bond, angle, dihedral

rigid - constrain many small clusters of atoms to move as a rigid body with NVE integration

setforce - set the force on each atom

shake - SHAKE constraints on bonds and/or angles

srd - stochastic rotation dynamics (SRD)

temp/berendsen - temperature control by Berendsen thermostat

temp/rescale - temperature control by velocity rescaling

thermal/conductivity - Muller-Plathe kinetic energy exchange for thermal conductivity calculation

viscosity - Muller-Plathe momentum exchange for viscosity calculation

viscous - viscous damping for granular simulations

wall/lj93 - Lennard-Jones 9-3 wall

wall/reflect - reflecting wall(s)

Data Visualization for Atomistic/Molecul- ar Simulations

These are high-quality visualization packages we have used and recommend.

LAMMPS can either write output files directly in an input format recognized by these programs

[VMD](#)

[AtomEye](#)

[OVITO](#)

[ParaView](#)

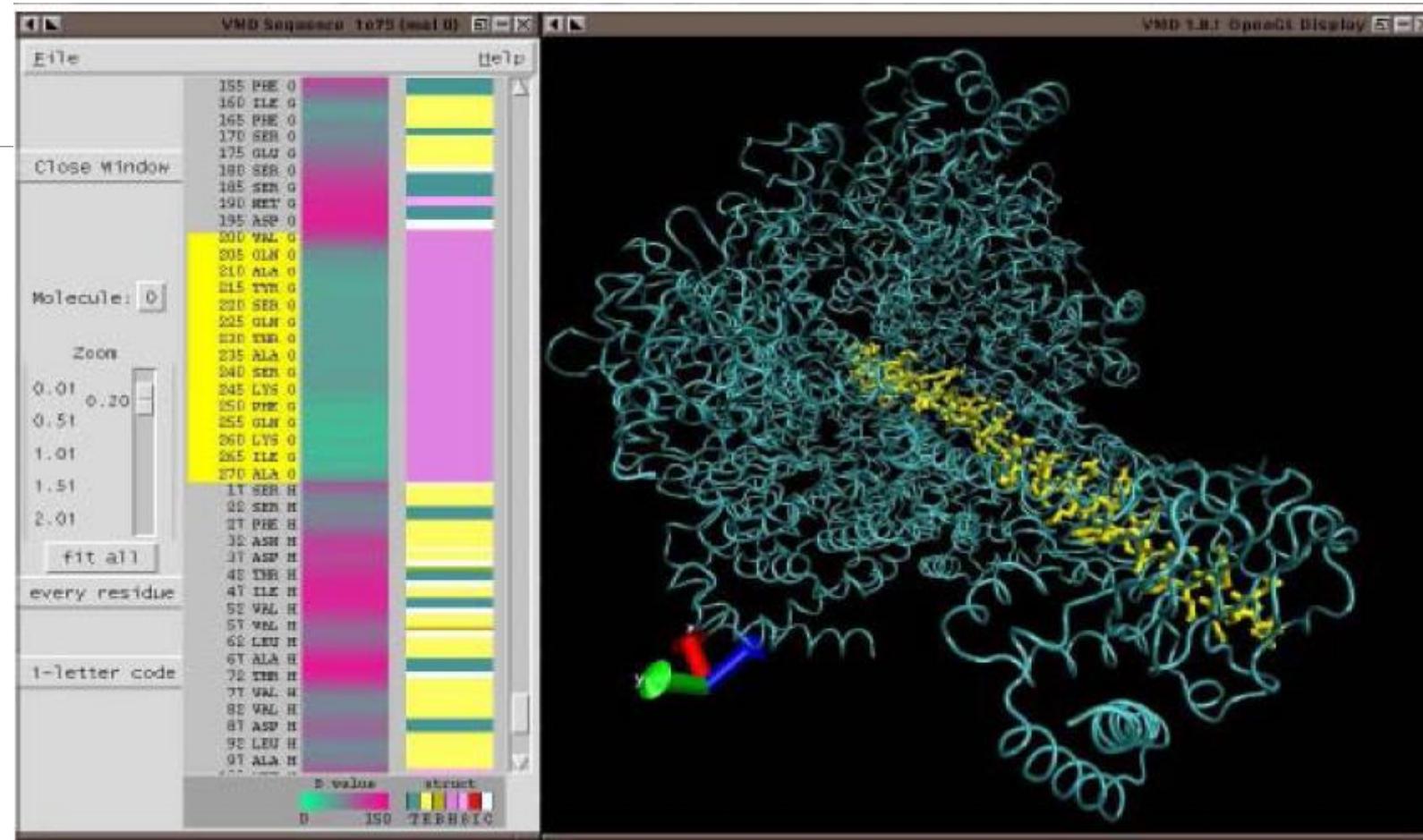
[PyMol](#)

[Raster3d](#)

[RasMol](#)

VMD

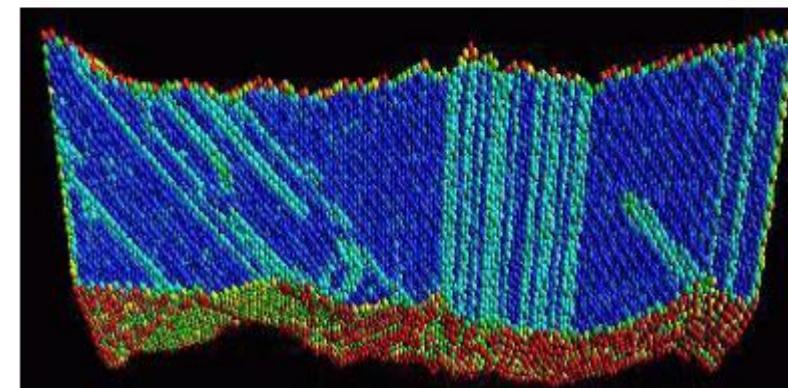
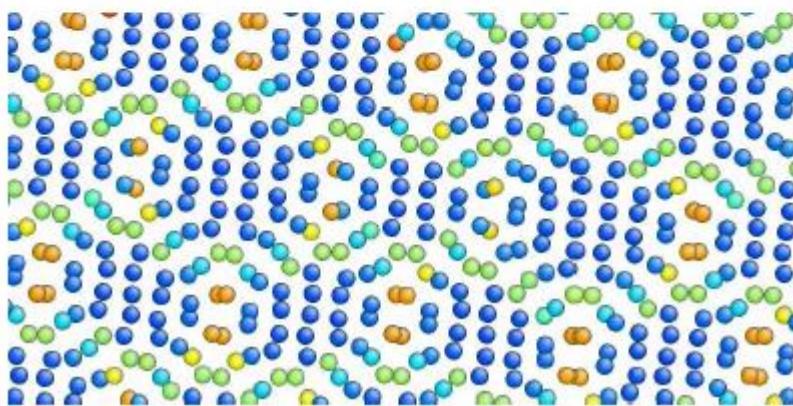
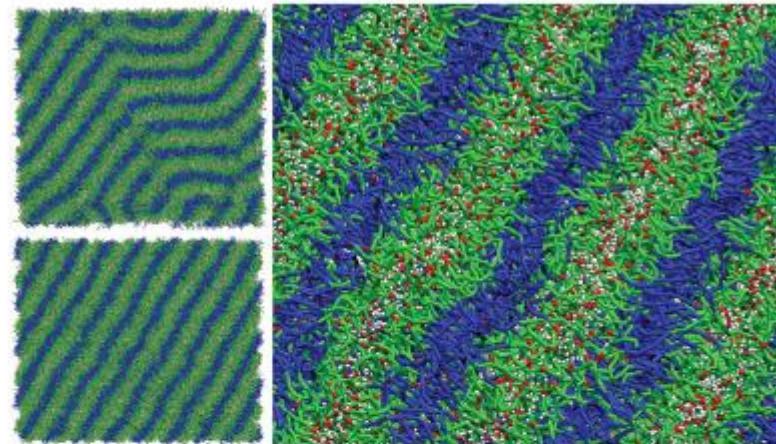
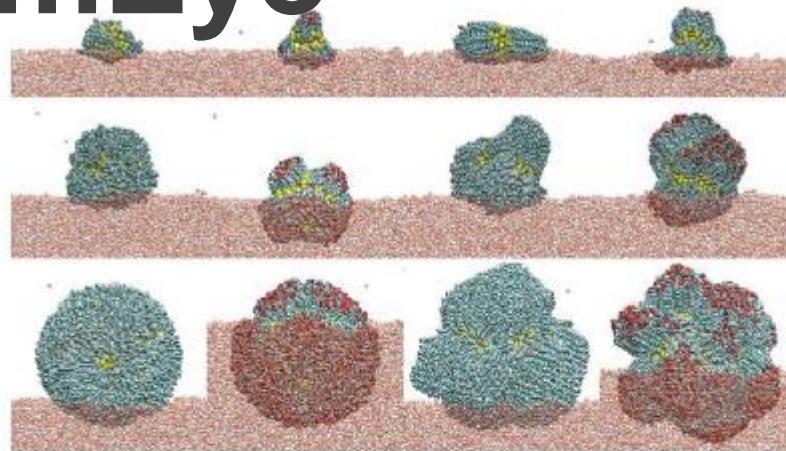
<http://www.ks.uiuc.edu/Research/vmd>



- Feature : surface, electronic density, hydrogen bond...
- Format : .pdb, mol, xyz, POSCAR, gjf ...

AtomEye

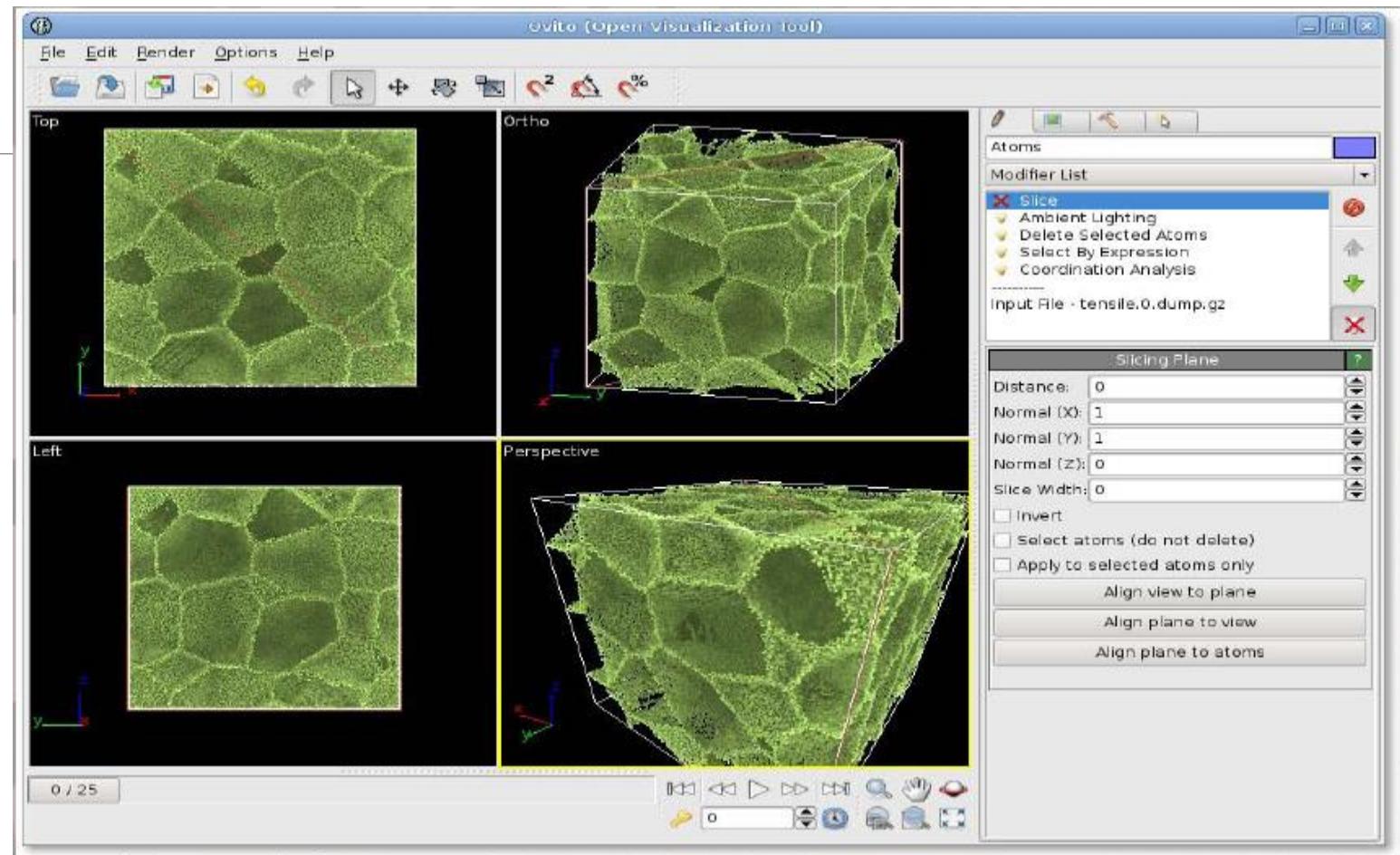
<http://li.mit.edu/Archive/Graphics/A/>



- Feature : default → coordination number, central symmetry parameter, local von Mises shear strain
- Format : cfg

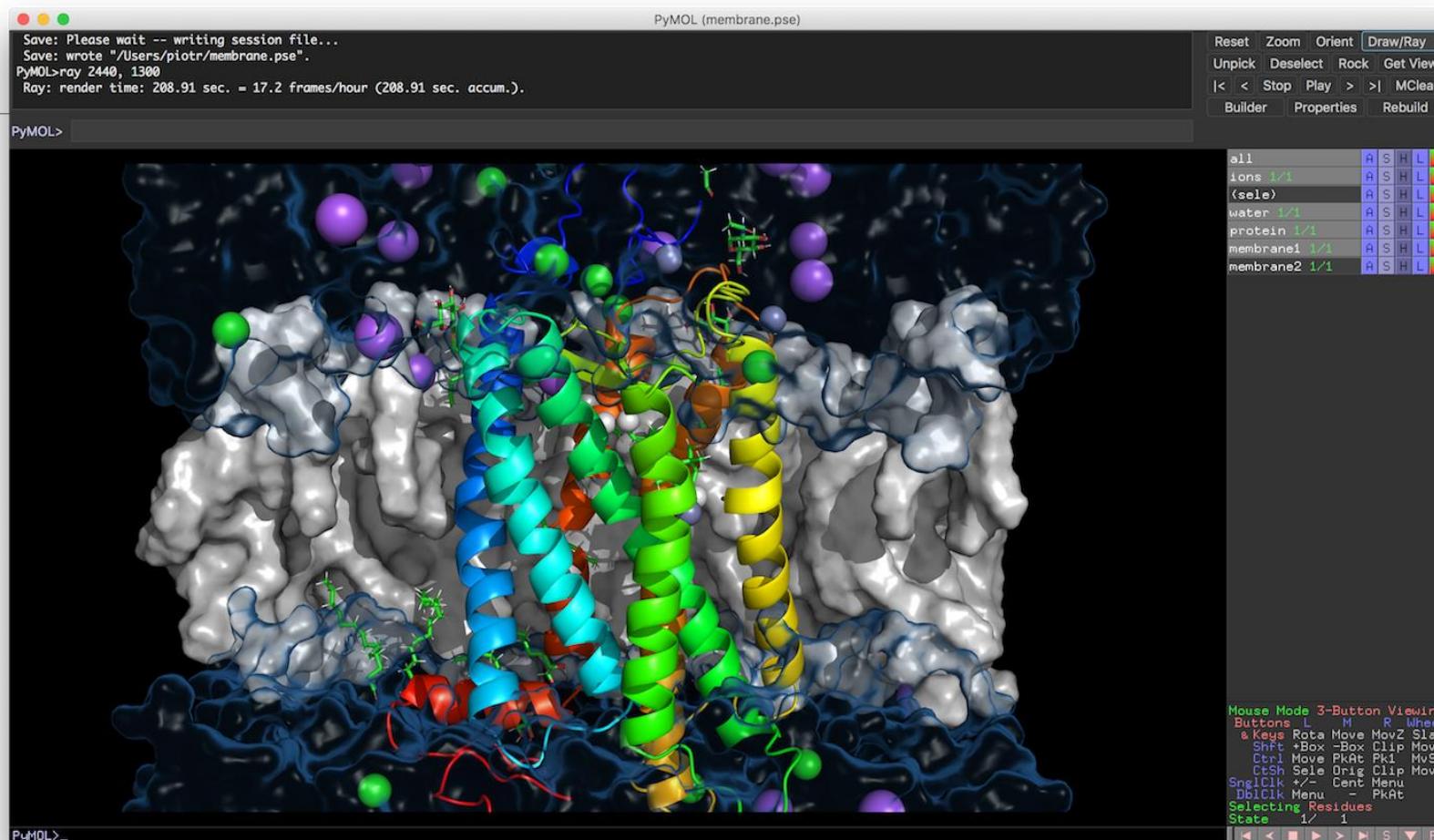
Ovito

<http://www.ovito.org/>



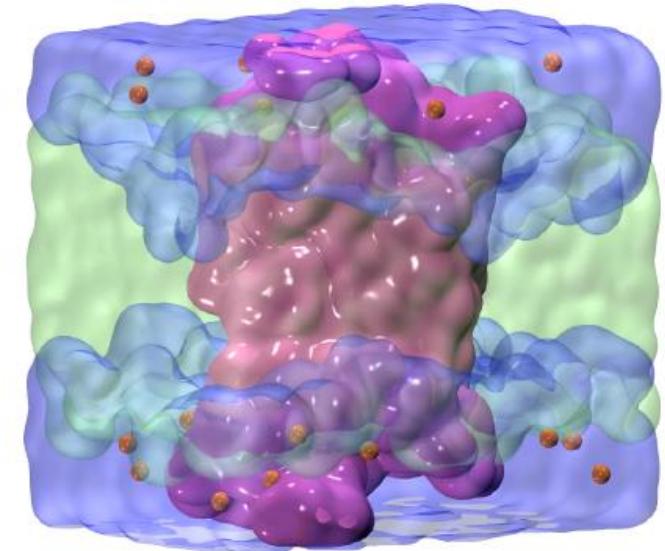
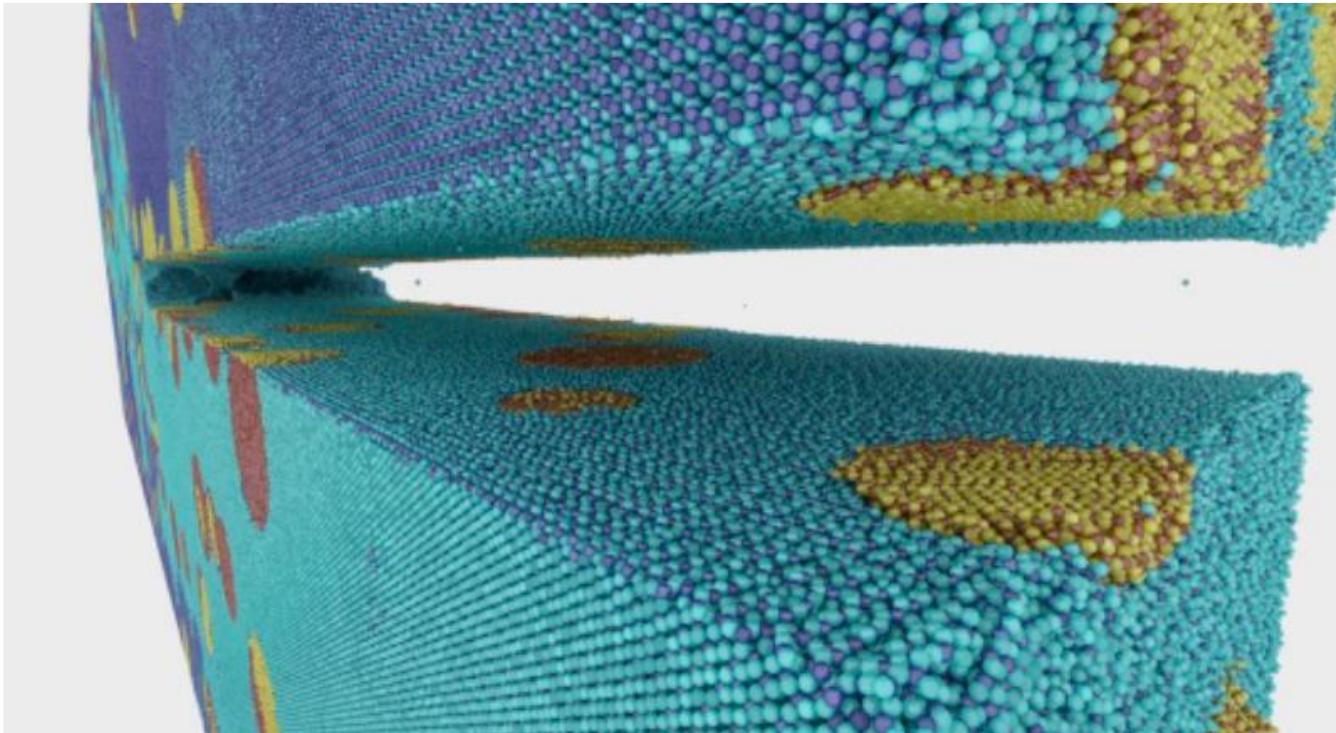
- Feature : A fast analyzing configuration viewer for crystal structures → FCC BCC HCP
- Format : xyz, cfg

Pymol



- Feature : surface, electronic density, hydrogen bond...
- Format : .pdb, mol, xyz, POSCAR, gjf ...

paraview



- Feature : Highly customized analyze tool and large scale visualization.
- Format : xyz, cfg, vtk,...

Melting-Cu

Model system Cu, Al

Initial conditions T, P...

Supercell number of atoms, PBCs ...

Interatomic potentials EAM

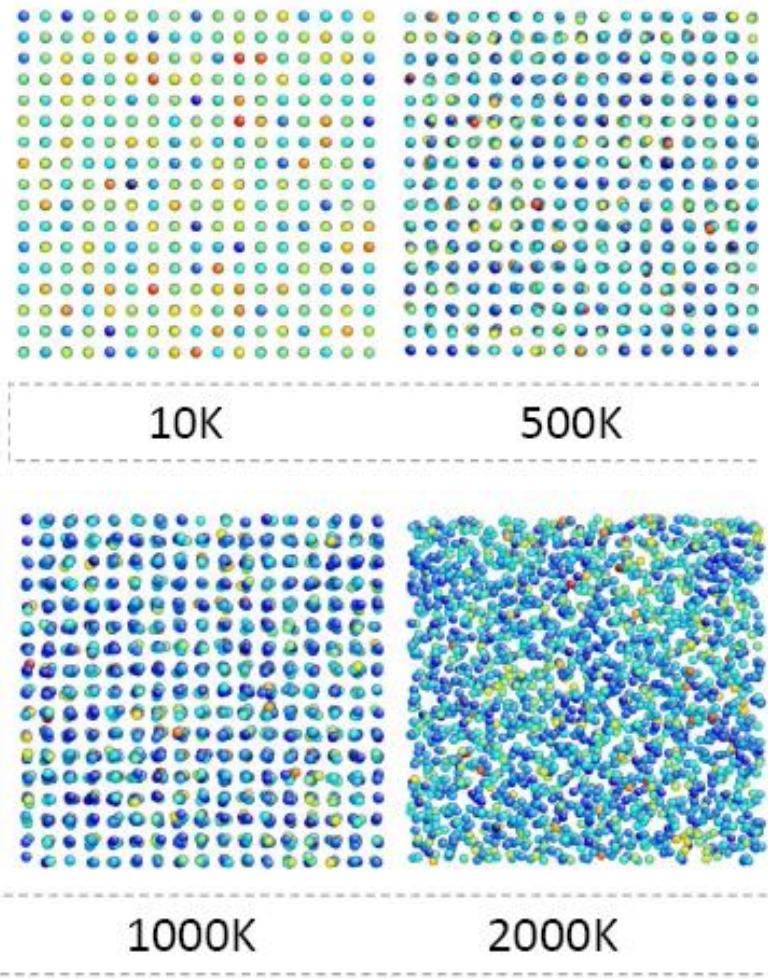
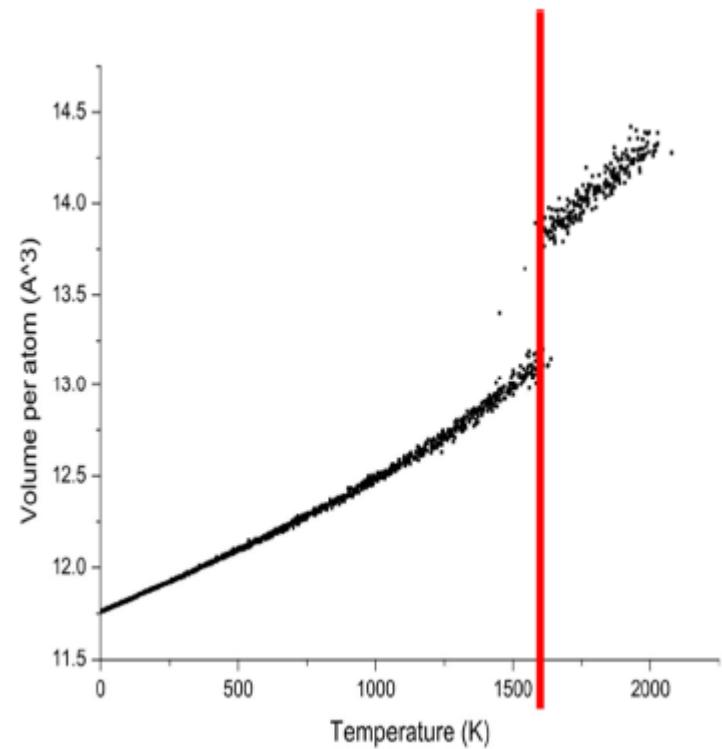
Ensembles NPT (Nosé-Hoover thermostat)

input script

Melting

```
units          metal
boundary      p p p
atom_style    atomic
lattice       fcc 3.61
region        box block 0 4 0 4 0 4
create_box    1 box
create_atoms  1 box
pair_style    eam
pair_coeff   * * Cu_u3.eam
timestep     0.01
thermo       1000
variable      N equal step
variable      pote equal pe
variable      T equal temp
variable      Press equal press
variable      V equal vol
velocity      all create 10 825577 dist gaussian
fix           extra all print 100 "${N} ${T} ${V} ${pote} ${Press}" file data
dump          1 all cfg 10000 a*.cfg id type xs ys zs c_ke
dump_modify   1 element Cu
fix           1 all npt temp 10 2000 1 iso 0 0 10
run           120000
```

若超過三組輸出參數 需使用{}



Melting-Si

Model system Si

Initial conditions T, P...

Supercell number of atoms, PBCs ...

Interatomic potentials Tersoff

Ensembles NPT (Nosé-Hoover thermostat)

input script-1

```
units          metal
atom_style    atomic

lattice        diamond 5.431
region         cubic block 0 3 0 3 0 5
create_box     1 cubic
create_atoms   1 box
mass           1 28.0855
pair_style     tersoff
pair_coeff     * * Si.tersoff Si

velocity       all create 300.0 456236 sum yes mom
yes dist gaussian
neighbor       1 bin
neigh_modify   every 1 delay 20 check yes ## 20+1
step=>check
thermo         1000
timestep       0.002
```

input script-2

```
variable  
variable  
variable  
fix  
screen no  
variable           vv atom "(vx*vx+vy*vy+vz*vz)^(1/2) "  
#####compute average energy/atom#####  
#compute          pe_a    all pe/atom  
#fix              vva all ave/atom 10 70 1000 vx vy vz  
#fix              pe_a all ave/atom 10 70 1000 c_pe_a  
#dump             id all custom 1000 dump.melt x y z mass f_pe_a  
#####compute average energy/atom#####  
  
fix               1 all nvt temp 300 300 1  
run               500  
unfix            1  
  
#fix              1 all nve  
#fix              2 all temp/rescale 10 300.0 300.0 10.0 1.0  
fix               2 all npt temp 2000 4000 1 x 1 1 1 y 1 1 1  
#####deform#####  
#fix              3 all deform 1 z erate 0.005 units box  
  
run              200000
```