

Molecular Dynamics Simulations

Oleg N. Starovoytov

HPC User Services

LSU HPC / LONI

sys-help@loni.org

Louisiana State University

Baton Rouge

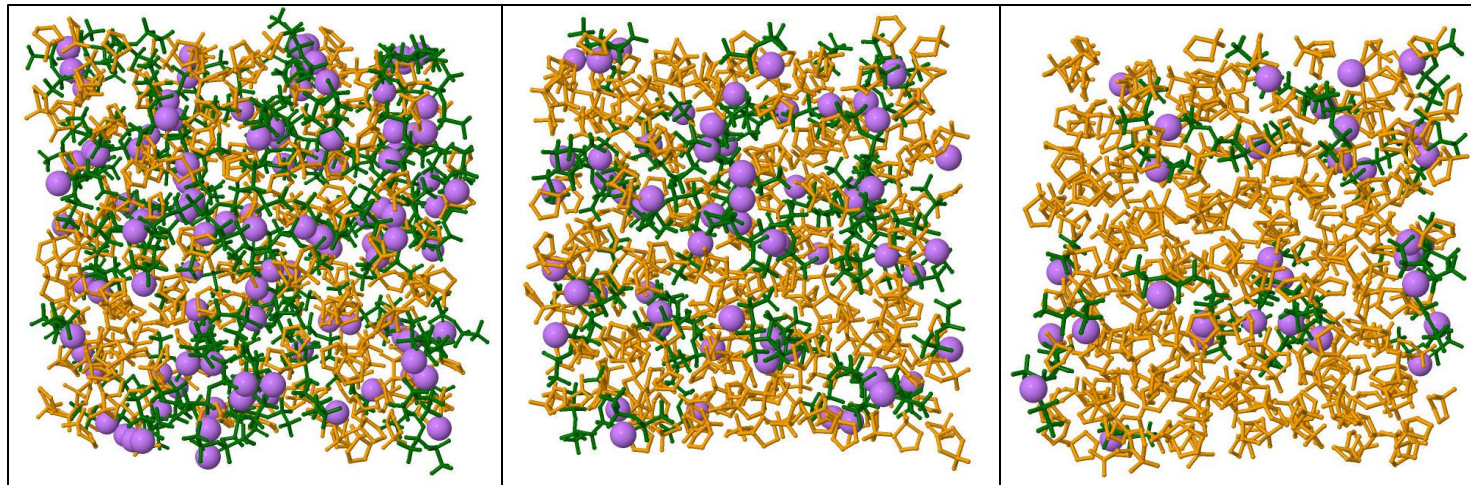
November 5, 2025

- Part 1

1. Introduction to Molecular Dynamics Simulations
2. Molecular Dynamics Simulation packages
3. HPC LSU and LONI Software environment

- Part 2

1. Running MD simulations using available packages on an HPC System



LiTFSI: 2.17 mol/dm³

LiTFSI: 1.77 mol/dm³

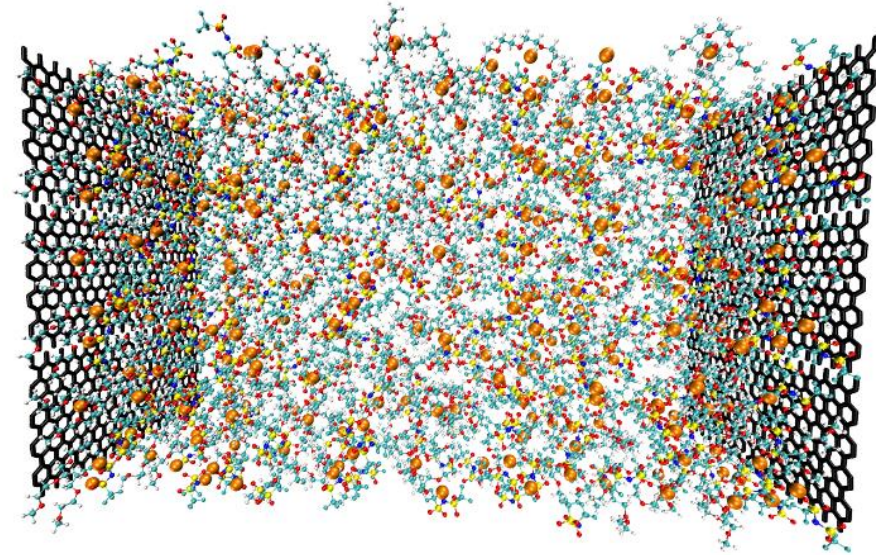
LiTFSI: 1.03 mol/dm³



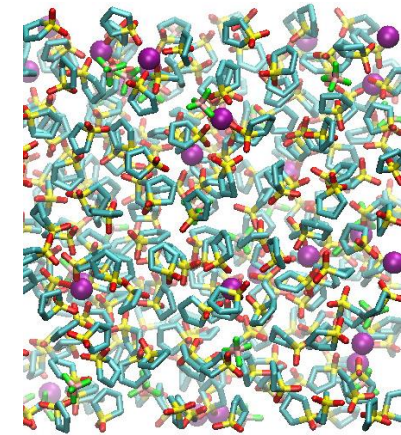
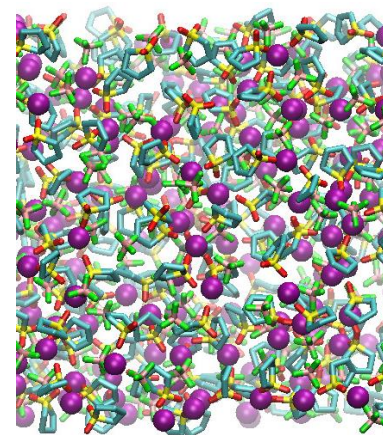
Introduction to Molecular Dynamics Simulations

Molecular dynamics (MD) is a versatile tool for calculating structural and dynamics properties of molecular systems at equilibrium as a function of time. The molecular systems should obey the laws of classical physics.

Molecular dynamics simulations are widely used in various fields like chemistry, physics, biology, and material science. The most popular research studies include the dynamics of proteins, DNA and RNA structures, ionic liquids (ILs), lipid bilayers and membranes, battery electrolytes, and ...



Ion dynamics in battery electrolytes



Molecular dynamics simulation models

1. Atom and void - Democritus
2. An ensemble of atoms, each has a point mass m .
3. Group of atoms (OPLS-UA)
4. Coarse-grained models (MARTINI model)
5. Machine learning models (potentials)

Force fields

1. Pair-wise classical force fields (AMBER, CHARMM, OPLS, GROMOS)
2. Many-body force fields include (EAM, Tersoff, REBO)
3. Reactive force fields (ReaxFF)
4. Machine learning models (potentials)

Molecular dynamics simulations

1. Integrate Newton's equation of motion, $F = ma$
2. Set $3N$ ODEs to propagate over time (Velocity Verlet Algorithm)

Thermodynamic properties

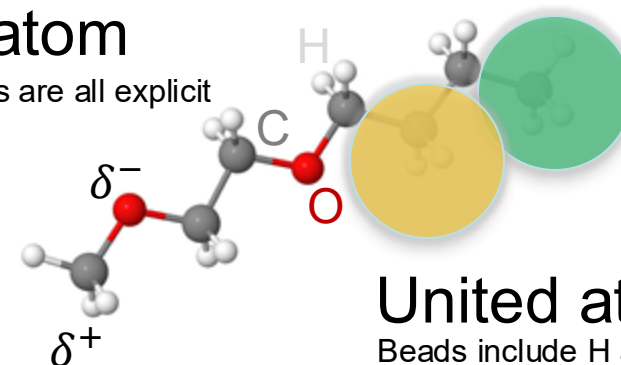
1. Calculate structural and dynamic properties as a time average of an ensemble of atoms.



Democritus ~400 B.C.
The Greek word ατομος – invisible.

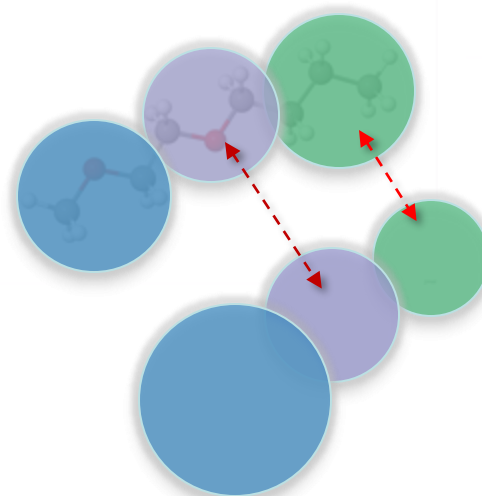
All atom

H atoms are all explicit



United atom

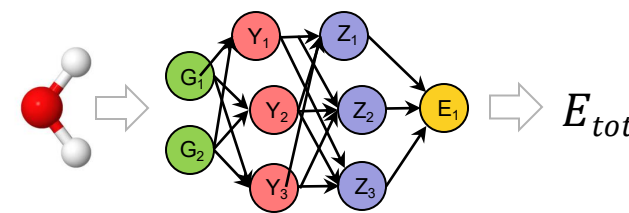
Beads include H atoms in CH₂ CH₃



Coarse-Grained models

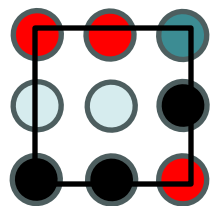
Beads include entire functional groups

Per-atom NN



Machine Learning (ML) potentials: SNAP, FLARE, ...

QM/MD

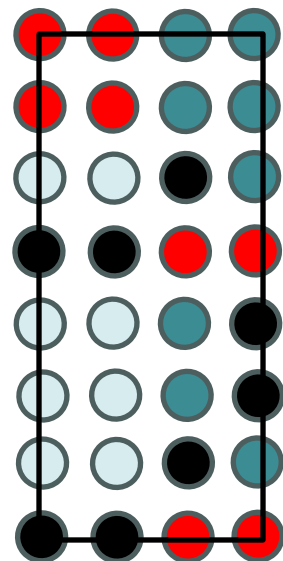


$O(N_e^\alpha)$

$N \sim 100-1000$ atoms

System size is too small and too expensive to calculate.

Accurate description of interatomic interactions.
There are no fixed interatomic potentials.



$O(N_{atoms})$

$N = 10^5 - 10^6$ atoms

It can capture some chemical and physical phenomena.

Inaccurate description of interatomic interactions.

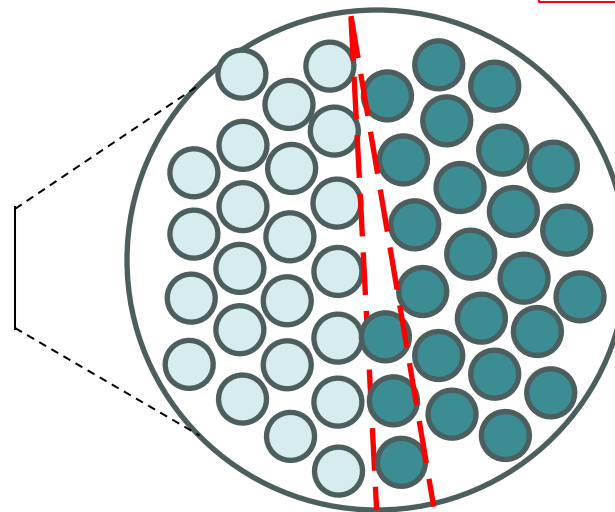
$$E^{Total} = \sum_{i=1}^N E'_i$$

A need for the QM accuracy to describe interatomic interactions.

A need for simulating large systems, $N \sim 10^5 - 10^6$.

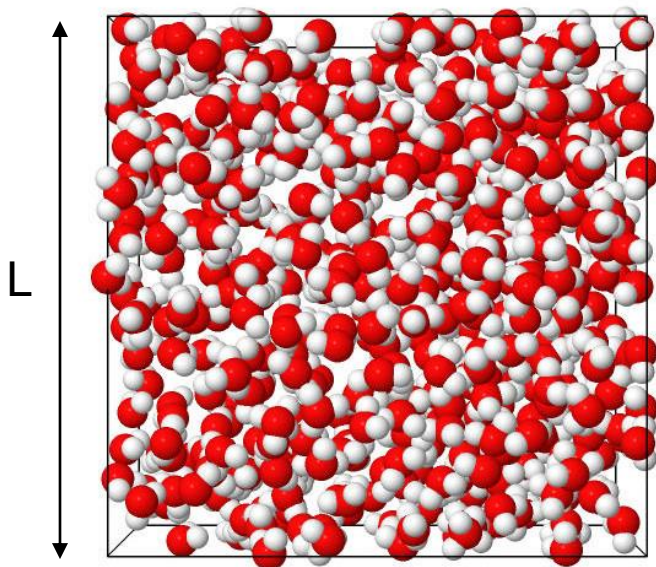
A need for low simulation costs.

Machine Learning (ML) potentials to reproduce energies (E), forces (F), and stress tensor (σ).



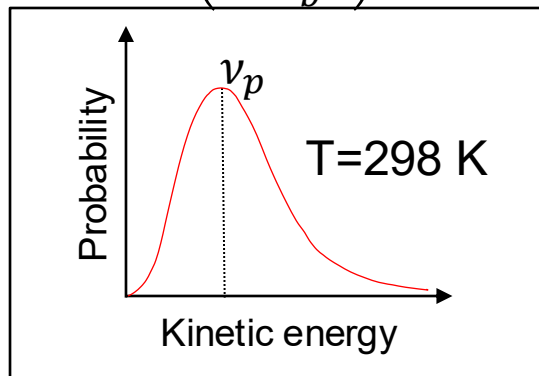
Schematic representation of the arrangement of atoms at a grain boundary in a polycrystalline sample.

Simulation box, L



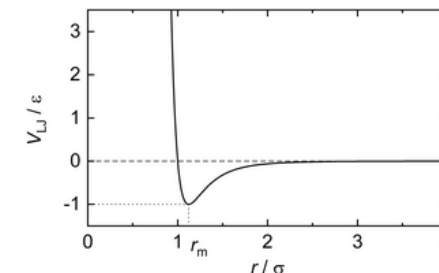
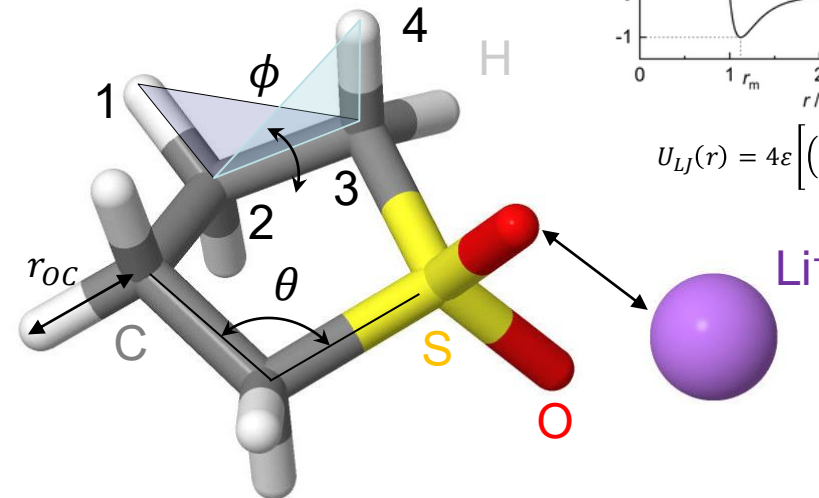
$$N = 1536 \quad r_i = x, y, z$$

$$P(v_{i,r}) = \left(\frac{m}{2\pi k_b T} \right)^{\frac{1}{2}} e^{-\frac{mv_{i,r}^2}{2\pi k_b T}}$$



1. Set up a system of N atoms.
2. Assign x, y, and z coordinates to each atom
3. Assign velocities using Maxwell-Boltzmann Distribution
4. Choose the right force field (Potential function)
5. Propagate atomic positions using integration algorithms (Velocity Verlet, Leap Frog, and ...)

Force Field



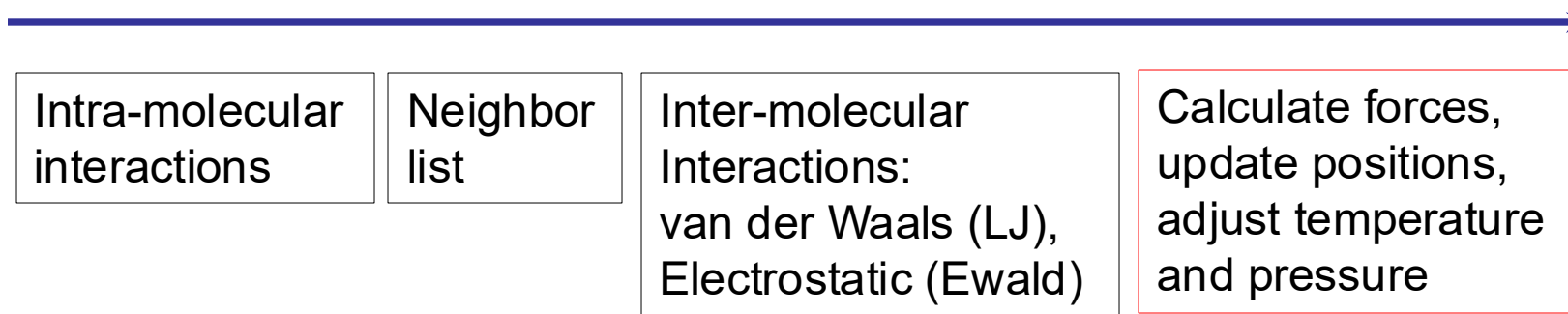
$$U_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

$$U_{total} = U_{intra} + U_{inter}$$

$$U_{intra} = U_{bond} + U_{bend} + U_{torsion} + U_{out\ of\ plane}$$

$$U_{inter} = U_{Coulomb} + U_{van\ der\ Waals}$$

Time step: Δt



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

$$a(t + \Delta t) = \frac{f(t + \Delta t)}{m}$$

$$v(t + \Delta t) = v(t) + \frac{1}{2} \Delta t [a(t) + a(t + \Delta t)]$$

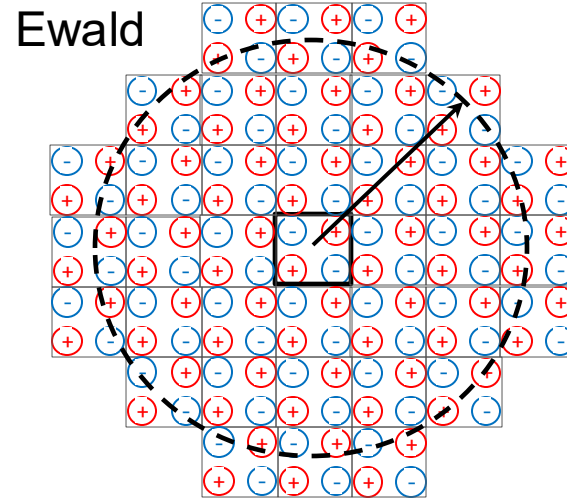
1. Ewald summation (by Peter P. Ewald in 1921)
2. PPME (Particle-Particle mesh Ewald, Hockney 1981)
3. PME (Particle mesh Ewald, Darden 1993)

$$U^{Ewald} = U^{real} + U^{reciprocal} + U^{self}$$

$$\mathcal{O}(N^2) \quad \longrightarrow \quad \mathcal{O}(N \cdot \log(N))$$

A three-dimensional grid is introduced to optimize the computation of long-range interactions and calculate the reciprocal space contribution.

A discrete set of points is introduced where charge densities and potentials are calculated significantly reducing the number of calculations needed for reciprocal space.



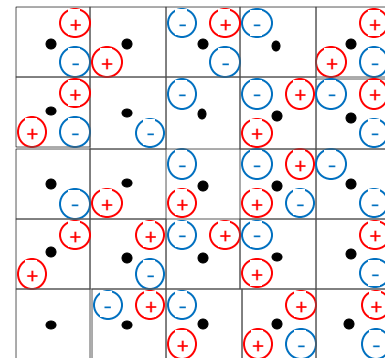
$$\sum_{i=1}^N q_i = 0$$

$$\mathcal{O}(N)$$

$$U^{real} = \frac{1}{2} \sum_{i,j} \sum_n q_i q_j \frac{\text{erfc}(\alpha r_{ij,n})}{r_{ij}}$$

$$U^{self} = -\frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^N q_i^2$$

$$U^{reciprocal} = \frac{1}{2\pi V} \sum_{i,j} q_i q_j \sum_{k \neq 0} \frac{\exp\left(-\frac{\pi k^2}{\alpha}\right) + 2\pi i k \cdot (r_i - r_j)}{k^2}$$



PME

$$\mathcal{O}(N \cdot \log(N))$$

A background image showing a dense collection of molecular models. The molecules are rendered in a stick-and-ball style, with atoms represented by spheres in shades of purple, green, and yellow, and bonds represented by thin yellow rods. The molecules are scattered across the frame, creating a complex, interconnected network.

Molecular Dynamics Simulation Packages

- LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator, Sandia National Lab, 1995) is a molecular dynamics simulation package, <https://www.lammps.org>
- GROMACS (GROningen Machine for Chemical Simulations, University of Groningen, 1991) is a molecular dynamics simulation package, <https://www.gromacs.org>
- NAMD (Not Another Molecular Dynamics Program, University of Illinois Urbana-Champaign, 1995) is a molecular dynamics simulation package (CHARMM force field), <https://www.ks.uiuc.edu/Research/namd>
- AMBER (Assisted Model Building with Energy Refinement, University of California, 2002) is a molecular dynamics simulation package (DNA force fields), <https://ambermd.org>

MD Simulation Packages

Name	Model builder	Min	MD	MC	GPU	License
LAMMPS	Yes	Yes	Yes	Yes	Yes	Free
GROMACS	No	Yes	Yes	No	Yes	Free
NAMD	Yes	Yes	Yes	No	Yes	Free
AMBER	Yes	Yes	Yes	Yes	Yes	Proprietary

- <module available>

	LAMMPS	GROMACS	NAMD	AMBER	VMD
QBC (QB3)	2023/08/02	2021.7	3.0b7	18/22	1.9.3
QBD (QB4)	2020/2023	2021.7	2.14/3.0bX	22	1.9.3
SMIC	2023	2021	3.07	22	1.9.3
MIKE	2022/2023	2021.3	2.14	18/22	1.9.3

HPC LSU and LONI Software Environment

```
[user@mike2 ~]$ module av
```

```
amber/18/intel-2021.5.0-intel-mpi-2021.5.1
```

```
amber/22/intel-2021.5.0-cuda-11.5.0-intel-mpi-2021.5.1
```

```
amber/22/intel-2021.5.0-intel-mpi-2021.5.1
```

```
.
```

```
gromacs/2021.3/intel-2021.5.0-intel-mpi-2021.5.1
```

```
.
```

```
lammps/02Aug2023/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1
```

```
lammps/02Aug2023/intel-2021.5.0-intel-mpi-2021.5.1
```

```
.
```

```
namd/2.14/intel-2021.5.0
```

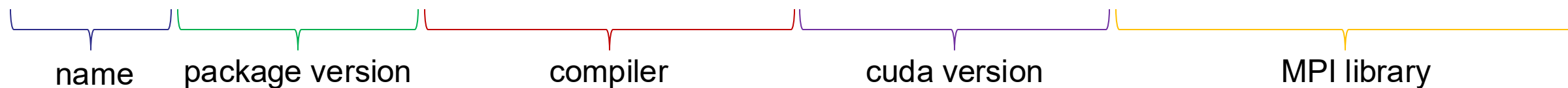
```
namd/2.14/intel-2021.5.0-cuda
```

No GPU

lammps/02Aug2023/intel-2021.5.0-intel-mpi-2021.5.1



lammps/02Aug2023/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1



GPU

```
[user@mike2 ~]$ module purge
```

```
[user@mike2 ~]$ module load lammmps/23Jun2022/intel-2021.5.0-intel-mpi-2021.5.1
```

```
[user@mike2 ~]$ module list
```

Currently Loaded Module files:

- 1) intel/2021.5.0
- 2) intel-mpi/2021.5.1
- 3) lammmps/23Jun2022/intel-2021.5.0-intel-mpi-2021.5.1

```
[user@mike2 ~]$ module display lammmps/02Aug2023/intel-2021.5.0-intel-mpi-2021.5.1
```

```
module-whatism {LAMMPS stands for Large-scale Atomic/Molecular Massively Parallel Simulator. This package uses patch releases, not stable release. See https://github.com/spack/spack/pull/5342 for a detailed discussion. }
```

```
conflict      lammmps
```

```
prepend-path  PATH /usr/local/packages/lammmps/02Aug2023/intel-2021.5.0-intel-mpi-2021.5.1/bin
```

```
prepend-path  MANPATH /usr/local/packages/lammmps/02Aug2023/intel-2021.5.0-intel-mpi-2021.5.1/share/man
```

```
[user@mike2 ~]$ ls /usr/local/packages/lammps/02Aug2023/intel-2021.5.0-intel-mpi-2021.5.1/bin
```

```
binary2txt chain.x Imp_mpi micelle2d.x msi2Imp phana stl_bin2txt
```

There are two types of jobs in the High-Performance Computing (HPC) environment.

1. Interactive jobs

Interactive jobs allow the user to run tasks interactively, usually directly connected to the system through a terminal or GUI.

These jobs are intended for tasks that require interaction from the user.

`salloc` and/or `srun` commands

2. Batch jobs

Batch jobs are computational tasks that are submitted to the HPC cluster without the need for interaction from the user.

These jobs are queued and executed by the Slurm resource manager.

`sbatch` <script.sh>



Running Molecular Dynamics Simulations Using Available Packages

Every LAMMPS simulation needs two essential files:

1. structure / topology (.lammps),
2. parameters (.in).

Structure/topology

1536 atoms

Atoms

```
1 1 1 -1.04840 23.067397 25.992172 12.516813
2 1 2 0.52420 23.651513 25.756170 13.277936
3 1 2 0.52420 23.106625 25.196754 11.981115
```

...

Parameters

```
units      real
atom_style full
boundary   p p p
```

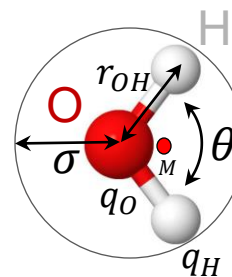
Force Field

```
pair_style lj/cut/tip4p/cut 1 2 1 1 0.125 8.0
bond_style harmonic
angle_style harmonic
kpace_style none
```

#Read data

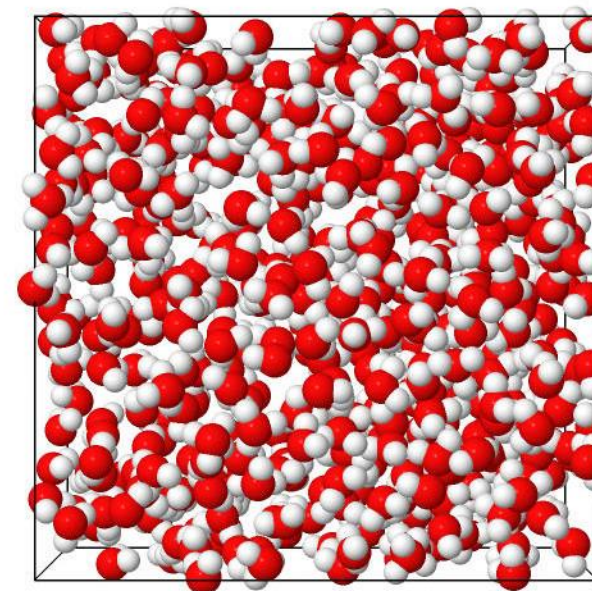
```
read_data tip4p_512.lammps
```

...



$r_{OH} = 0.9572 \text{ \AA}$
 $\theta = 104.52^\circ$
 $q_M = -1.040 e$
 $q_H = 0.52 e$
 $\sigma = 3.1536 \text{ \AA}$

TIP4P water model



Running LAMMPS interactively

```
[user@mike2 LAMMPS]$ salloc -A hpc_hpcadmin10 -p workq -t 01:00:00 -J Traning
[user@mike2 LAMMPS]$ module purge
[user@mike2 LAMMPS]$ module load lammeps/02Aug2023/intel-2021.5.0-intel-mpi-2021.5.1
[user@mike2 LAMMPS]$ module list
```

Currently Loaded Modulefiles:

1) intel/2021.5.0 2) intel-mpi/2021.5.1 3) lammeps/02Aug2023/intel-2021.5.0-intel-mpi-2021.5.1

```
[user@mike171 LAMMPS]$ srun -n 1 imp_mpi -in tip4p_512.in > tip4p_512.out &
[user@mike171 LAMMPS]$ top
```

```
[user@mike171 LAMMPS]$ srun -n64 imp_mpi -in tip4p_512.in > tip4p_512.out &
[user@mike171 LAMMPS]$ ls
[user@mike171 LAMMPS]$ log.lammps tip4p_512.in tip4p_512.lammps tip4p_512.out tip4p_512.traj
```

```
[user@mike171 LAMMPS]$ tail -f log.lammps
[user@mike171 LAMMPS]$
```

Running LAMMPS jobs using SLURM system

```
#!/bin/bash
#SBATCH -p workq
#SBATCH -N 1
#SBATCH -n 64
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A hpc_allocation
#SBATCH -J test
#SBATCH -o lammmps_%j_%N.out
#SBATCH -e lammmps_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load lammmps/23Jun2022/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1
```

```
echo $SLURM_NNODES
echo $SLURM_NTASKS
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N1 -n64 Imp_mpi -in water_tip4p.in > water_tip4p.out
```

Running LAMMPS jobs using SLURM system

```
#!/bin/bash
#SBATCH -p workq
#SBATCH -N 2
#SBATCH -n 128
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A hpc_allocation
#SBATCH -J test
#SBATCH -o lammmps_%j_%N.out
#SBATCH -e lammmps_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load lammmps/23Jun2022/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1
```

```
echo $SLURM_NNODES
echo $SLURM_NTASKS
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N2 -n128 Imp_mpi -in water_tip4p.in > water_tip4p.out
```

Every GROMACS simulation needs three essential files:

Structure

TIP3P water

1536

```
1SOL OW 1 2.308 1.150 1.290 0.0374 -0.1946 0.1896
1SOL HW1 2 2.242 1.208 1.328 -0.7293 -1.1860 0.4498
1SOL HW2 3 2.376 1.143 1.357 1.6678 2.7051 -0.9958
```

...

Topology

; Include forcefield parameters

```
#include "charmm27.ff/forcefield.itp"
```

; Include water topology

```
#include "charmm27.ff/tip3p.itp"
```

...

Parameters

; Run parameters

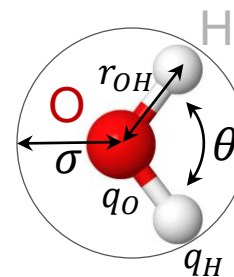
```
integrator = md ;leap-frog integrator
```

```
nsteps = 5000 ;
```

```
dt = 0.002 ;
```

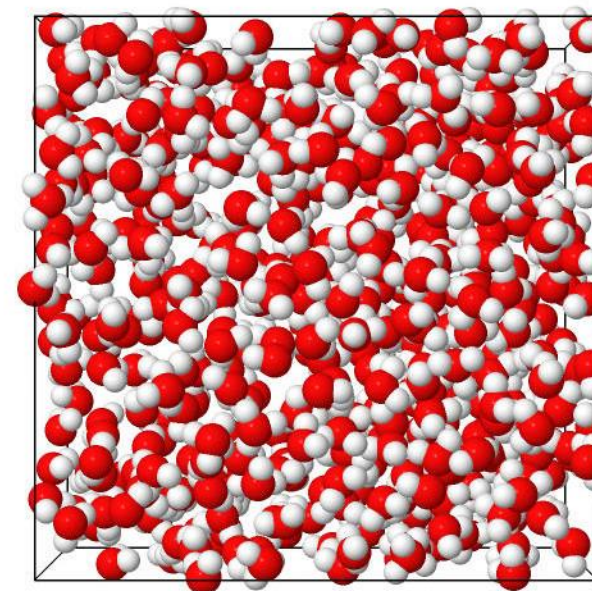
...

1. structure (.gro/.pdb),
2. topology (.top), and
3. parameters (.mdp).



$r_{OH} = 0.9572 \text{ \AA}$
 $\theta = 104.52^\circ$
 $q_O = -0.834 e$
 $q_H = 0.417 e$
 $\sigma = 3.15061 \text{ \AA}$

TIP3P water model



Running GROMACS interactively

```
[user@mike2 GROMACS]$ salloc -A hpc_hpcadmin10 -p workq -t 01:00:00 -J Traning
[user@mike2 GROMACS]$ module purge
[user@mike2 GROMACS]$ module load gromacs/2021.3/intel-2021.5.0-intel-mpi-2021.5.1
[user@mike2 GROMACS]$ module list
```

Currently Loaded Modulefiles:

1) intel/2021.5.0 2) intel-mpi/2021.5.1 3) gromacs/2021.3/intel-2021.5.0-intel-mpi-2021.5.1

```
[user@mike2 GROMACS]$ srun gmx_mpi grompp -f min.mdp -c npt.gro -p topol.top -o min.tpr
[user@mike2 GROMACS]$ srun gmx_mpi mdrun --deffnm min
[user@mike2 GROMACS]$ ls
[user@mike2 GROMACS]$ min.edr min.gro min.log min.mdp min.tpr min.trr
```

```
[user@mike2 GROMACS]$ srun gmx_mpi grompp -f eql.mdp -c min.gro -p topol.top -o eql.tpr
[user@mike2 GROMACS]$ srun -N1 -n64 gmx_mpi mdrun --deffnm eql
[user@mike2 GROMACS]$ ls
[user@mike2 GROMACS]$ eql.cpt eql.edr eql.gro eql.log eql.mdp eql.tpr eql.trr eql.xtc
```

Running GROMACS jobs using SLURM system

```
#!/bin/bash
#SBATCH -p workq
#SBATCH -N 1
#SBATCH -n 64
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A hpc_allocation
#SBATCH -J test
#SBATCH -o gromacs_%j_%N.out
#SBATCH -e gromacs_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load gromacs/2021.3/intel-2021.5.0-intel-mpi-2021.5.1
```

```
echo $SLURM_NNODES
echo $SLURM_NTASKS
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N1 -n64 gmx_mpi mdrun -deffnm npt -v
```

Running GROMACS jobs using SLURM system

```
#!/bin/bash
#SBATCH -p workq
#SBATCH -N 2
#SBATCH -n 128
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A hpc_allocation
#SBATCH -J test
#SBATCH -o gromacs_%j_%N.out
#SBATCH -e gromacs_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load gromacs/2021.3/intel-2021.5.0-intel-mpi-2021.5.1
```

```
echo $SLURM_NNODES
echo $SLURM_NTASKS
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N2 -n128 gmx_mpi mdrun -deffnm npt -v
```

Every NAMD simulation needs three essential files:
Input files are identical to the input files used by **X-PLOR** and **CHARMM**.

Coordinates (.pdb)

```
REMARK original generated coordinate pdb file
ATOM 1 OH2 TIP3W 5 3.668 10.082 15.904 1.00 0.00 WW1 O
ATOM 2 H1 TIP3W 5 3.224 10.451 15.101 1.00 0.00 WW1 H
ATOM 3 H2 TIP3W 5 3.092 10.379 16.627 1.00 0.00 WW1 H
...
```

Structure (.psf)

```
1536 !NATOM
1 WW1 5 TIP3 OH2 OT -0.834000 15.9994 0
2 WW1 5 TIP3 H1 HT 0.417000 1.0080 0
3 WW1 5 TIP3 H2 HT 0.417000 1.0080 0
...
```

Topology (.xplor)

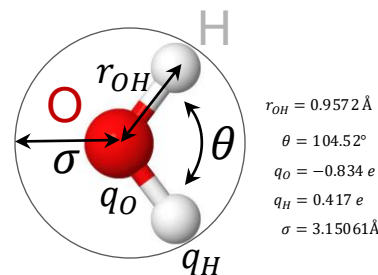
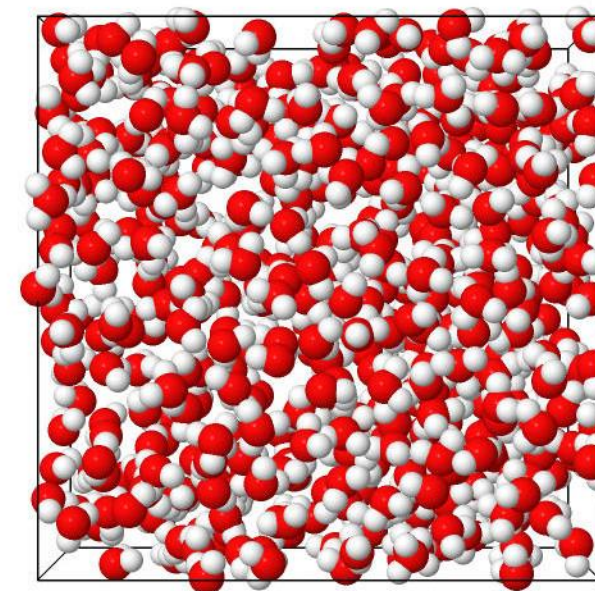
```
BOND OT HT 450.000 0.9572 !ALLOW WAT
BOND HT HT 0.000 1.5139 !ALLOW WAT
ANGLE HT OT HT 55.000 104.5200 !ALLOW WAT
...
```

Parameters (.input)

```
timestep 1.0
fullElectFrequency 4
numsteps 50000
outputtiming 20
...
```

1. coordinates (.pdb),
2. structure (.psf),
3. topology (.xplor),
4. parameters (.namd).

TIP3P water model



Running NAMD interactively

```
[user@mike2 NAMD]$ salloc -A hpc_hpcadmin10 -p workq -t 01:00:00 -J Traning
[user@mike2 NAMD]$ module purge
[user@mike2 NAMD]$ module load namd/2.14/intel-2021.5.0
[user@mike2 NAMD]$ module list
```

Currently Loaded Modulefiles:

1) intel/2021.5.0 2) namd/2.14/intel-2021.5.0

```
[user@mike2 NAMD]$ srun -N1 -n64 namd2 tip3p_512.namd > tip3p_512.out &
[user@mike2 NAMD]$ ls
[user@mike2 NAMD]$ par_all22_prot_lipid.xplor tip3p_512.out tip3p_512.out.coor.BAK
tip3p_512.out.vel.BAK tip3p_512.out.xsc.BAK tip3p_512.psf tip3p_512.nam p3p_512.out.coor
tip3p_512.out.vel tip3p_512.out.xsc tip3p_512.pdb
```

Running NAMD jobs using SLURM system

```
#!/bin/bash
#SBATCH -p workq
#SBATCH -N 1
#SBATCH -n 64
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A hpc_allocation
#SBATCH -J test
#SBATCH -o NAMD_%j_%N.out
#SBATCH -e NAMD_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load namd/2.14/intel-2021.5.0
```

```
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N1 -n64 namd2 tip3p_512.namd > tip3p_512.out
```

Running NAMD jobs using SLURM system

```
#!/bin/bash
#SBATCH -p workq
#SBATCH -N 2
#SBATCH -n 128
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A hpc_allocation
#SBATCH -J test
#SBATCH -o NAMD_%j_%N.out
#SBATCH -e NAMD_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load namd/2.14/intel-2021.5.0
```

```
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N2 -n128 namd2 tip3p_512.namd > tip3p_512.out
```

Every AMBER simulation needs three essential files:
Initial coordinate, topology, and parameter files.

Structure (.inpcrd)

```
default_name
1536
16.5307255 19.4975686 18.1539268 13.3987255 16.6285686 12.9069268
15.2747255 15.8905686 11.7989268 17.0747255 16.7645686 10.4629268
...
```

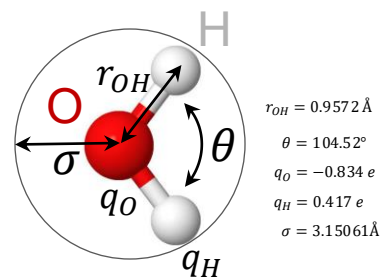
Topology (.prmtop)

```
%FORMAT(10I8)
1536 2 1024 0 512 0 ...
2048 512 0 0 0 2 ...
0 0 0 0 0 0 ...
...
```

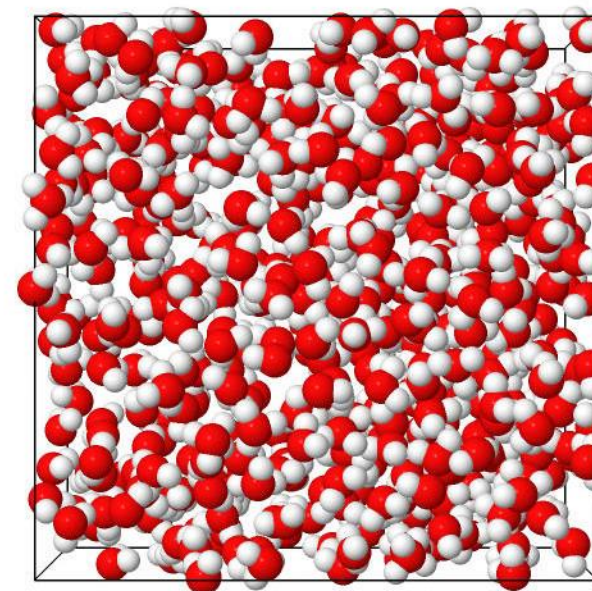
Parameters (.inp)

```
Production
&cntrl
imin=0,
ntx=1,
ntwv = 1
...
```

1. coordinates (.inpcrd),
2. topology (.prmtop),
3. parameter (.inp).



TIP3P water model



Running AMBER interactively

```
[user@mike2 AMBER]$ salloc -A hpc_hpcadmin10 -p workq -t 01:00:00 -J Training
[user@mike2 AMBER]$ module purge
[user@mike2 AMBER]$ module load amber/22/intel-2021.5.0-intel-mpi-2021.5.1
[user@mike2 AMBER]$ module list
```

Currently Loaded Modulefiles:

1) intel/2021.5.0 2) intel-mpi/2021.5.1 3) amber/22/intel-2021.5.0-intel-mpi-2021.5.1

```
[user@mike2 AMBER]$ srun sander -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd &
[user@mike2 AMBER]$ ls
```

```
[user@mike2 AMBER]$ srun -n1 -N64 sander.MPI -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd &
[user@mike2 AMBER]$ ls
```

```
[user@mike2 AMBER]$ srun -n1 -N64 pmemd.MPI -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd &
[user@mike2 AMBER]$ ls
```

Running AMBER jobs using SLURM system

```
#!/bin/bash
#SBATCH -p workq
#SBATCH -N 1
#SBATCH -n 64
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A loni_allocation
#SBATCH -J test
#SBATCH -o amber_%j_%N.out
#SBATCH -e amber_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load amber/22/intel-2021.5.0-intel-mpi-2021.5.1
```

```
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N1 -n64 sander.MPI -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd -r tip3p_512.rst
```

Running AMBER jobs using SLURM system

```
#!/bin/bash
#SBATCH -p workq
#SBATCH -N 2
#SBATCH -n 128
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A loni_allocation
#SBATCH -J test
#SBATCH -o amber_%j_%N.out
#SBATCH -e amber_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load amber/22/intel-2021.5.0-intel-mpi-2021.5.1
```

```
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N2 -n128 sander.MPI -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd -r tip3p_512.rst
```

- Marrink, S. J. *et al*, “The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations” J. Phys. Chem. B 2007, 111, 27, 7812-7824, <https://doi.org/10.1021/jp071097f>
- Van der Waals, Johannes Diderik (1837 - 1923). *Over de Continuïteit van den Gas- en Vloeistofoestand*. Leiden, 1873, (Nobel Prize 1910, van der Waal’s equation of state) <http://rbx-exhibit2000.scs.illinois.edu/vanderwaals.htm>
- Fabbrizzi, L. “Beyond the Molecule: Intermolecular Forces from Gas Liquefaction to X–H... π Hydrogen Bonds” ChemPlusChem, Volume 87, Issue 1, 2022, Pages 1-23, ISSN 2192-6506, <https://doi.org/10.1002/cplu.202100243>
- Darden, T. York, D. and Pederson, L. “Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems” J. Chem. Phys. 1993, 98, 10089-10092
- Plimpton, S. “Fast Parallel Algorithms for Short-Range Molecular Dynamics”, Journal of Computational Physics, Volume 117, Issue 1, 1995, Pages 1-19, ISSN 0021-9991, <https://doi.org/10.1006/jcph.1995.1039>.
- Behler, J. and Parrinello, M. “Generalized Neural-Network Representation of High-Dimensional Potential Energy Surfaces” Physical Review Letters, 2007, 98, 146401 <https://doi.org/10.1103/PhysRevLett.98.146401>

Thank You



Environment	Slurm
Job ID	\$\$SLURM_JOBID
Submit directory	\$\$SLURM_SUBMIT_DIR
Submit host	\$\$SLURM_SUBMIT_HOST
Node list	\$\$SLURM_JOB_NODELIST
Total number of nodes	\$\$SLURM_NNODES
Total number of tasks	\$\$SLURM_NTASKS

SLURM stands for Simple Linux Utility for Resource Management, which also provides a workload management system. It enables users to submit, monitor, and manage jobs on a supercomputer. It provides a framework for distributing computational tasks and a job scheduling system.

Running LAMMPS interactively

```
[user@mike2 LAMMPS]$ salloc -N1 -n16 -p gpu --gres=gpu:1 --time=12:00:00 -A hpc_hpcadmin8  
[user@mike2 LAMMPS]$ module purge  
[user@mike2 LAMMPS]$ module load lammops/02Aug2023/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1  
[user@mike2 LAMMPS]$ module list
```

Currently Loaded Modulefiles:

1) intel/2021.5.0 2) intel-mpi/2021.5.1 3) lammops/02Aug2023/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1

```
[user@mike179 LAMMPS]$ time srun -N1 -n16 lmp_mpi -sf gpu -pk gpu 1 neigh yes newton off -in lj.in > lj.out &
```

```
[user@mike179 LAMMPS]$ nvidia-smi -l
```

Running LAMMPS interactively

```
[user@mike2 LAMMPS]$ salloc -N1 -n32 -p gpu --gres=gpu:2 --time=12:00:00 -A hpc_hpcadmin8  
[user@mike2 LAMMPS]$ module purge  
[user@mike2 LAMMPS]$ module load lammops/02Aug2023/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1  
[user@mike2 LAMMPS]$ module list
```

Currently Loaded Modulefiles:

1) intel/2021.5.0 2) intel-mpi/2021.5.1 3) lammops/02Aug2023/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1

```
[user@mike179 LAMMPS]$ time srun -N1 -n32 Imp_mpi -sf gpu -pk gpu 2 neigh yes newton off -in lj.in > lj.out &
```

```
[user@mike179 LAMMPS]$ nvidia-smi -l
```

Running LAMMPS jobs using 1 GPU

```
#!/bin/bash
#SBATCH -p gpu
#SBATCH --gres=gpu:1
#SBATCH -N 1
#SBATCH -n 16
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A hpc_allocation
#SBATCH -J test
#SBATCH -o lammmps_%j_%N.out
#SBATCH -e lammmps_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load lammmps/02Aug2023/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1
```

```
echo $SLURM_NNODES
echo $SLURM_NTASKS
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N1 -n16 lmp_mpi -sf gpu -pk gpu 1 neigh yes newton off -in lj.in > lj.out
```

Running LAMMPS jobs using 2 GPUs

```
#!/bin/bash
#SBATCH -p gpu
#SBATCH --gres=gpu:2
#SBATCH -N 1
#SBATCH -n 32
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A hpc_allocation
#SBATCH -J test
#SBATCH -o lammmps_%j_%N.out
#SBATCH -e lammmps_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load lammmps/02Aug2023/intel-2021.5.0-cuda-11.6.0-intel-mpi-2021.5.1
```

```
echo $SLURM_NNODES
echo $SLURM_NTASKS
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun -N1 -n32 lmp_mpi -sf gpu -pk gpu 2 neigh yes newton off -in lj.in > lj.out
```

Running AMBER interactively

```
[user@mike2 AMBER]$ salloc -N1 -n16 -p gpu --gres=gpu:1 --time=12:00:00 -A hpc_hpcadmin8  
[user@mike2 AMBER]$ module purge  
[user@mike2 AMBER]$ module load amber/22/intel-2021.5.0-cuda-11.5.0-intel-mpi-2021.5.1  
[user@mike2 AMBER]$ module list
```

Currently Loaded Modulefiles:

1) intel/2021.5.0 2) intel-mpi/2021.5.1 3) amber/22/intel-2021.5.0-cuda-11.5.0-intel-mpi-2021.5.1

```
[user@mike2 AMBER]$ export CUDA_VISIBLE_DEVICES="0"
```

```
[user@mike2 AMBER]$ time srun pmemd.cuda_DPFP.MPI -AllowSmallBox -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd -r tip3p_512.rst &
```

```
[user@mike2 AMBER]$ nvidia-smi -l
```

DPFP – stands for Double Precision Floating point (64 bits)

SPFP – stands for Single Precision Floating point (32 bits)

Running AMBER interactively

```
[user@mike2 AMBER]$ salloc -N1 -n32 -p gpu --gres=gpu:2 --time=12:00:00 -A hpc_hpcadmin8  
[user@mike2 AMBER]$ module purge  
[user@mike2 AMBER]$ module load amber/22/intel-2021.5.0-cuda-11.5.0-intel-mpi-2021.5.1  
[user@mike2 AMBER]$ module list
```

Currently Loaded Modulefiles:

1) intel/2021.5.0 2) intel-mpi/2021.5.1 3) amber/22/intel-2021.5.0-cuda-11.5.0-intel-mpi-2021.5.1

```
[user@mike2 AMBER]$ export CUDA_VISIBLE_DEVICES="0,1"
```

```
[user@mike2 AMBER]$ time srun pmemd.cuda_DPFP.MPI -AllowSmallBox -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd -r tip3p_512.rst &
```

```
[user@mike2 AMBER]$ nvidia-smi -l
```

DPFP – stands for Double Precision Floating point (64 bits)

SPFP – stands for Single Precision Floating point (32 bits)

Running AMBER jobs using 1 GPU

```
#!/bin/bash
#SBATCH -p gpu
#SBATCH -gres=gpu:1
#SBATCH -N 1
#SBATCH -n 16
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A loni_allocation
#SBATCH -J test
#SBATCH -o lammps_%j_%N.out
#SBATCH -e lammps_%j_%N.err
#SBATCH --mail-user=your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load amber/22/intel-2021.5.0-cuda-11.5.0-intel-mpi-2021.1
```

```
export CUDA_VISIBLE_DEVICES="0"
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun pmemd.cuda_DPFP.MPI -AllowSmallBox -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd -r tip3p_512.rst
```

Running AMBER jobs using 2 GPUs

```
#!/bin/bash
#SBATCH -p gpu
#SBATCH -gres=gpu:2
#SBATCH -N 1
#SBATCH -n 32
#SBATCH -c 1
#SBATCH -t HH:MM:SS
#SBATCH -A loni_allocation
#SBATCH -J test
#SBATCH -o lammps_%j_%N.out
#SBATCH -e lammps_%j_%N.err
#SBATCH --mail-user your@email.address
#SBATCH --mail-type=ALL
```

```
module purge
module load amber/22/intel-2021.5.0-cuda-11.5.0-intel-mpi-2021.1
```

```
export CUDA_VISIBLE_DEVICES="0,1"
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
echo $SLURM_SUBMIT_DIR
cd $SLURM_SUBMIT_DIR
```

```
time srun pmemd.cuda_DPFP.MPI -AllowSmallBox -O -i eql.inp -o eql.out -p tip3p_512.prmtop -c tip3p_512.inpcrd -r tip3p_512.rst
```