Computational Materials

-

Exercise

Introduction

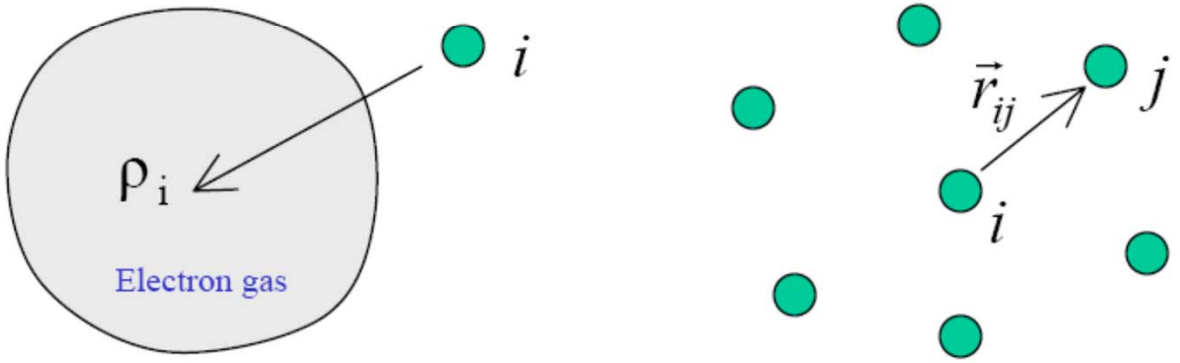
- Use LAMMPS code to study defect properties and their dynamics behavior
- How to run LAMMPS code to determine the formation energies of defects
- Use EAM potentials and compare defect properties with experimental values
- Analyze the results (VMD to reveal the migration of interstitials)

EAM Potential

- Embedded-atom method (EAM)

$$E_{tot} = \sum_i E_i \qquad E_i = F_i(\rho_i) + \frac{1}{2}\sum_{j \neq i} \varphi_{ij}(r_{ij}) \qquad \rho_i = \sum_{j \neq i} f_j(r_{ij})$$

embedding energy **two-body term**



Install LAMMPS

Download LAMMPS:

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

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

SVN checkout and update

$ svn co svn://svn.lammps.org/lammps-ro/trunk mylammps
$ cd mylammps/src/
$ make serial & make g++_openmpi …

Pre-built Ubuntu Linux executables:

$ sudo apt-get install lammps

Windows installer package

Pre-compiled Windows installers which install LAMMPS executables on a Windows system can be downloaded from this site:
http://rpm.lammps.org/windows.html

Login Server

$ ssh –l lammps 202.127.205.158

$ Password: 000000

$ cp –r exercise5-LAMMPS **your_folder_name**

$ cd **your_folder_name**

$ ls

Interstitial migration_int Vacancy

Load LAMMPS Module on Flux

- The flux provides a compiled LAMMPS code.
- We can check available versions of LAMMPS by typing below commands.

```
$ module av lammps
---------------------------- /home/software/rhel6/Modules/modulefiles ----------------------
lammps/1Feb14          lammps/22Jul14          lammps/4Jul12(default)  lammps/8Oct12
```

We are going to use the version of LAMMPS code named “lammps/22Jul14”. Before loading this module, we need to know the pre-required modules to run this version of LAMMPS code:

```
$ module show lammps/22Jul14
-------------------------------------------------------------------
/home/software/rhel6/Modules/modulefiles/lammps/22Jul14:
conflict        lammps
prereq   openmpi/1.6.5/intel/14.0.2
prereq   mkl/11.1
```

MPI – Message Passing Interface

Load LAMMPS Module on Flux

```
$ module show lammps/22Jul14
-------------------------------------------------------------------
/home/software/rhel6/Modules/modulefiles/lammps/22Jul14:
conflict        lammps
prereq   openmpi/1.6.5/intel/14.0.2
prereq   mkl/11.1
```

- These results indicate that we need
  "openmpi/1.6.5/intel/14.0.2"
  and "mkl/11.1"

to run LAMMPS code. Thus, we need to upload these two modules.

```
$ module load openmpi/1.6.5/intel/14.0.2

$ module load mkl/11.1
```

Load LAMMPS Module on Flux

- Loading LAMMPS code

```
$ module load lammps/22Jul14
```

- Check if LAMMPS module has been loaded successfully:

```
$ module list
```

If the “lammps/22Jul14” is shown in the list, this means that LAMMPS has been loaded successfully.

RUN LAMMPS MD Simulations

A Vacancy in Cu

$ cd Vacancy

$ ls

Cu_mishin1.eam.alloy - interatomic potential for Cu
Cu.vacancy – input file for a vacancy in Cu

$ vi Cu.vacancy

```
# Input file for Vacancy Formation Energy

# ---------- INITIALIZATION ----------
clear
units           metal (eV, A, ps)
dimension       3
boundary        p       p    p
atom_style      atomic
```

A Vacancy in Cu

```
# ----------------- ATOM DEFINITION ------------------
variable ao equal 3.615

lattice         fcc  ao
region          simbox block -4 4 -4 4 -4 4

create_box      1 simbox

lattice         fcc 3.615  orient x 1 0 0 orient y 0 1 0 orient z 0 0 1 (lattice)
create_atoms    1 region simbox (atoms)

# ---------------------- FORCE FIELDS ----------------------
pair_style      eam/alloy
pair_coeff * * Cu_mishin1.eam.alloy Cu
#--------------------------Settings---------------------------
compute eng all pe/atom
compute eatoms all reduce sum c_eng
```

A Vacancy in Cu

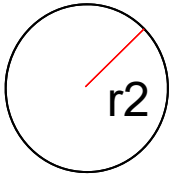
```
#---------------------Run Minimization-----------------------
reset_timestep  0

thermo 10 (every 10 timesteps – output thermo properties)
thermo_style custom step pe lx ly lz press pxx pyy pzz c_eatoms

dump 1 all custom 400 dump.relax.1.* id type xs ys zs c_eng

min_style cg (perfect crystal)
minimize 1e-15 1e-15 5000 5000 (energy, force, maximum iterations)

run 0 (only for outputting initial information)
undump 1

#variable N equal count(all), counts the total number of atoms in the cell
#the total number of atoms is stored to the variable N

variable N equal count(all)
variable No equal $N (total number of atoms)
```

A Vacancy in Cu

#variable Ei equal "c_eatoms" computes the initial energy of the cell system before the vacancy
#E is needed to store the initial energy of the system to the variable Ei

variable E equal "c_eatoms"
variable Ei equal $E (outputting the total energy of perfect crystal)

#--
variable r2 equal sqrt(${ao}^2+${ao}^2)/4 (size of an atom)
#r2 is the radius of the copper atom



#region select is a region defined so that all atoms within this region are removed
region select sphere 0 0 0 ${r2} units box
delete_atoms region select compress yes (forming a vacancy)
#--

A Vacancy in Cu

```
reset_timestep  0

thermo 10
thermo_style custom step pe lx ly lz press pxx pyy pzz c_eatoms

dump 1 all custom 400 dump.relax.2.* id type xs ys zs c_eng


min_style cg
minimize 1e-15 1e-15 5000 5000

#variable Ef equal "c_eatoms" computes the final energy of the cell system after
the vacancy
#The final energy is stored to the variable Ef

variable Ef equal "c_eatoms” (total energy of a crystal with a vacancy)
```

A Vacancy in Cu

```
#########################################
# SIMULATION DONE
print "All done"
print "Total number of atoms = ${No}"
print "Initial energy of atoms = ${Ei}"
print "Final energy of atoms = ${Ef}"
```

Run LAMMPS Code for a Vacancy

1) Single process serial execution

```
$ lmp_linux  < Cu.vacancy
```

2) Multiple process for parallel calculation

```
$ mpirun  -np 2 lmp_linux < Cu.vacancy
```

3) To run LAMMPS with pbs, you can perform

```
$ cp ~/lammps.pbs
$ cp Cu.vacancy inputfile
$ qsub lammps.pbs
```

Formation Energy of A Vacancy in Cu

Formation energy of vacancy

$$E_{vac}^{f} = E(N-1) - \frac{N-1}{N}E(N) = E(N-1) - (N-1)E_{coh}$$

Here, E(N) is the energy of a perfect crystal containing of N atoms, E_{coh} is the cohesive energy/atom of FCC copper (E_{coh}=E(N)/N), and E(N-1) is the total energy of a crystal with a vacancy.

Please note that all the energies are relaxed using CG

- Using VMD to visualize the vacancy - dump.relax.2.24

Formation Energy of An Interstitial in Cu

$ cd ..

$ cd Interstitial

$ ls

100 110 octahedral tetrahedral

Let us calculate a <100> dumbbell interstitial

$ cd 100

$ ls
Cu.100 Cu_mishin1.eam.alloy

$ vi Cu.100

Formation Energy of An Interstitial in Cu

```
#-----------------------------------------------------------
variable r2 equal sqrt(${ao}^2+${ao}^2)/4
#r2 is the radius of the copper atom
region select sphere 0 0 0 ${r2} units box
#region select is a region defined so that all atoms within this region are
removed
delete_atoms region select compress yes
create_atoms 1 single 0.90375 0.0 0.0 units box
create_atoms 1 single -0.90375 0.0 0.0 units box
# create a 100 dumbbell interstitial centring at (0 0 0)
# 0.90375 is half of the lattice
#-----------------------------------------------------------
```

Single process serial execution

```
$ lmp_linux  < Cu.100
```

Formation Energy of An Interstitial in Cu

- Formation energy of an interstitial

$$E_{int}^{f} = E(N+1) - \frac{N+1}{N}E(N) = E(N+1) - (N+1)E_{coh}$$

Here, E(N) is the energy of a perfect crystal consisting of N atoms, E_{coh} is the cohesive energy/atom of FCC copper (E_{coh}=E(N)/N). E(N+1) is the total energy of a crystal containing an interstitial and N is the total number of atoms in the perfect crystal.

Please note that all the energies are relaxed using CG

- Using VMD to visualize the interstitial - dump.relax.2.**

Formation Energy of An Interstitial in Cu

Case 2 – <110> crowdion (in the directory of 110)

Case 3 – octahedral (in the directory of octahedral)

Case 4 – tetrahedral (in the directory of tetrahedral)

- Calculate their formation energies and determine their relative stabilities.

Migration of An Interstitial in Cu

go to /exercise5-LAMMPS2/migration_int

$ vi Cu.migration

```
#-------------------------------------------------------------
reset_timestep  0

thermo 100
thermo_style custom step temp pe lx ly lz press pxx pyy pzz c_eatoms

min_style cg
minimize 1e-15 1e-15 5000 5000

dump 1 all custom 400 dump.relax.2.* xs ys zs type (output/400 timesteps)

velocity all create 600 678345 mom yes rot no
fix 1 all nvt temp 600 600 1 drag 1 (1 – 1ps adjusting temperature)

run 20000
#############################################
```

Migration of An Interstitial in Cu

$ qsub lammps.pbs

- Plot MSD as a function of time – msd.txt

- Using VMD to visualize the migration of the interstitial - dump.relax.2.**

$ ./extract

Running wsdefect to identify defects from the output of lammps code. Using mig_mov.lammpstrj (VMD file) to show defects as a function of time.