



What is LAMMPS?

LAMMPS is a freely available (GPL) classical molecular dynamics code and an acronym for Large-scale Atomic/Molecular Massively Parallel Simulator. LAMMPS has potentials for solid-state materials (metals, semiconductors) and soft matter (biomolecules, polymers) and coarse-grained or mesoscopic systems. It can be used to model atoms or, more generically, as a parallel particle simulator at the atomic, meso, or continuum scale. LAMMPS runs on single processors or in parallel using message-passing techniques and a spatial-decomposition of the simulation domain. The code is designed to be easy to modify or extend with new functionality. It is distributed by Sandia National Laboratories, a US Department of Energy laboratory.

This guide to running LAMMPS provides instructions on how to run standard test cases on the Supercomputing Wales systems.

LAMMPS Functionality

LAMMPS includes the following features and functionality:

- Running in parallel using MPI or OpenMP
- Various particle and model types (atoms, coarse-grained particles, united-atom polymers or organic molecules, all-atom polymers, organic molecules, proteins, DNA, metals, granular materials, coarse-grained mesoscale models, and many more)
- Various types of force fields (pairwise potentials, charged pairwise potentials, manybody potentials, charge equilibration, electron force field, coarse-grained potentials, mesoscopic potentials, bond potentials, angle potentials, dihedral potentials, improper potentials, and many more)
- Atom creation (read in atom coordinates from files, create atoms on one or more lattices, delete geometric or logical groups of atoms, replicate existing atoms multiple times, displace atoms)
- Ensembles, constraints and boundary conditions (2D or 3D systems, orthogonal or non-orthogonal, simulation domains, constant NVE, NVT, NPT, NPH, Parinello/Rahman integrators, thermostating options for groups and geometric regions of atoms, pressure control via Nose/Hoover or Berendsen barostatting in 1 to 3 dimensions, simulation box deformation, harmonic (umbrella) constraint forces, and many more)
- Integrators (velocity-Verlet integrator, Brownian dynamics, rigid body integration, energy minimization via conjugate gradient or steepest descent relaxation, rRESPA hierarchical timestepping, rerun command for post-processing of dump files)

- Diagnostics, multi-replica models, real-time visualisation, dynamic load balancing and many more

Step 1 - Log in

The examples used in this guide is configured to run on the Cardiff Skylake *Hawk* cluster. Connect to *hawklogin.cf.ac.uk* with your Supercomputing user credentials using your preferred method (e.g. PuTTY from a Windows machine or ssh from any Linux terminal, thus

```
ssh -l your_username hawklogin.cf.ac.uk
```

The steps below involve typing commands (**in bold font**) in the terminal window.

Step 2 - Load a BLAST module

In common with most of the application guides in this series, we assume at the outset that the module of choice would be selected from those originally available on HPC Wales. Thus a necessary first step would be to gain access to that module set by issuing the commands

```
module purge  
module load hpcw
```

Note that the “module purge” command is required to prevent any unintentional collision with pre-existing modules. Subsequently issuing the module command

```
module avail
```

will provide visibility of and access to the entire set of HPCW modules.

A number of LAMMPS binary packages are available.

- List pre-installed LAMMPS versions:

```
module avail lammmps
```
- Load the default LAMMPS module (Intel build, MPI):

```
module load lammmps
```
- Confirm the loaded modules. All dependencies are handled automatically via the module file:

```
module list
```

Step 3 - Create a directory

From your home directory, create a directory to hold LAMMPS user files:

```
cd ~  
mkdir lammmps
```

Step 4 - Obtain a test case

Two test cases are provided with the installation at

```
/app/chemistry/lammmps/27Oct11/sb/intel-13.0/intel-4.1/example
```

Copy the input files and job scripts for both examples to your user space:

```
cd lammmps  
cp /app/chemistry/lammmps/27Oct11/sb/intel-13.0/intel-
```

```
4.1/example/in.* .  
cp /app/chemistry/lammps/27Oct11/sb/intel-13.0/intel-  
4.1/example/lammps*SLURM.q .
```

Step 5 - Submit a job

Now you are ready to run the examples with the supplied job scripts.

- From your working directory, submit the job by: `sbatch lammptest.SLURM.q`
- Check the job queue by: `squeue`
- When the job finishes, it will have created an output file called `LammpsEx.Hawk.o.<JobID>` where `<JobID>` is the ID generated by the system.

- Compare your job output with the reference output file

```
/app/chemistry/lammps/27Oct11/sb/intel-13.0/intel-4.1/example/LammpsEx.Hawk.o.15470
```

- Repeat the above for the 2nd example, submitting the second example by: `sbatch lammptest1.SLURM.q`. When the job finishes, it will have created an output file called `LammpsEx1.Hawk.o.<JobID>`. The job will take around 90 seconds on 80 cores of Hawk. Compare your job output with the reference output file

```
/app/chemistry/lammps/27Oct11/sb/intel-13.0/intel-4.1/example/LammpsEx1.Hawk.o.15472
```

References

Documentation and source code is available at the LAMMPS website:

<http://lammps.sandia.gov/>

A variety of useful tutorials are also available at: <http://lammps.sandia.gov/tutorials.html>