PotC: Many-body potential implementations à la carte

Reproducability Description Appendix

Markus Höhnerbach
RWTH Aachen University
hoehnerbach@aices.rwth-aachen.de

Paolo Bientinesi*
RWTH Aachen University
pauldj@aices.rwth-aachen.de

ACM Reference Format:
Markus Höhnerbach and Paolo Bientinesi. 2018. PotC: Many-body potential implementations à la carte: Reproducability Description Appendix. In *Proceedings of ACM Student Research Competition at SC17 (ACM SRC/SC18).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 ABSTRACT

The poster contains computational results from the execution of generated potentials. We provide the generated files online, as well as the benchmark input files and parameter sets. LAMMPS itself is open-source software, and can easily be obtained. The experimental runs were performed on a 2nd generation Xeon Phi system.

2 DESCRIPTION

2.1 Check-list (artifact meta information)

- **Algorithm: Molecular Dynamics/Tersoff+Stillinger-Weber**
- **Program: LAMMPS**
- **Compilation:** `make intel_serial`**, ICC 18 (w/ GCC 7 stdlib)**
- **Binary:** `lmp_intel_serial`
- **Data set: LAMMPS benchmarks sw and tersoff**
- **Run-time environment: ICC 18/GCC 7**
- **Hardware: Xeon Phi 7210**
- **Publicly available?: Yes**

2.2 How software can be obtained (if available)

LAMMPS is available at http://lammps.sandia.gov/. We are using version 16Mar18. The generated output files from our generator are available at https://github.com/v0i0/poster-gen-pot. The repository also contains the Makefile, input and parameter files.

2.3 Hardware dependencies

The benchmark runs were performed on a Intel Xeon Phi 7210 system.

2.4 Software dependencies

The code was compiled using icc (ICC) 18.0.3 20180410 with the standard library of GCC 7.2.0. While LAMMPS would usually also depend on MPI, we opted to build a serial version of the code to reduce dependencies.

2.5 Datasets

The code uses the benchmarks provided with LAMMPS. These perform a simulation of 32000 atoms of crystalline Silicon for 100 timesteps, in both the Tersoff and Stillinger-Weber case. The input files are provided in our repository.

3 INSTALLATION

(1) Download the LAMMPS tarball and extract it.
(2) Copy `Makefile.intel_serial` into `lammps-16Mar18/src/MAKE`.
(3) Copy `pair_*_gen*` into `lammps-16Mar18/src`.
(4) Change directory into `lammps-16Mar18/src`.
(5) Execute `(cd STUBS; make)`.
(6) Execute `make intel_serial`.
(7) This produces a binary called `lmp_intel_serial`.

4 EXPERIMENT WORKFLOW

Change into the directory where you cloned https://github.com/v0i0/poster-gen-pot. It contains the input files of the form `{sw,tersoff}.in[.ref]`. Let `$BIN` be the path to the binary from the installation step. An input file can be executed using:

```
$BIN -in <input-file>
```

There are three dimensions along which we can obtain results:

(1) To choose between Stillinger-Weber or for Tersoff, choose corresponding input file prefix (either "sw" or "tersoff").
(2) To choose between the generated and handwritten code, choose the input file without suffix for generated, and the one with the ".ref" suffix for handwritten code.
(3) To choose between vectorized and non-vectorized code, add `-pk intel 0 mode double -sf intel` as command line options to enable vectorized execution, and add not options for non-vectorized execution.

The relevant information in the output are (1) the lines starting with "0" and "100" and (2) the line tyhat starts with "Loop". The outputs (1) contain data from the simulation and should match up for each potential. The output (2) contains the runtime of the simulation that was not spent on set-up, and is the time we display in our plot.

For the reader's convenience, the following command line will run all the input files and extract the relevant data:

```
for i in sw.in sw.in.ref tersoff.in tersoff.in.ref;
do
  echo $i;
  $BIN -in $i -pk intel 0 mode double -sf intel |
    grep '100\|Loop';
  $BIN -in $i | grep '100\|Loop';
done
```

*Advisor

ACM SRC/SC18, November 2018, Dallas, Texas USA
2018. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

5 EVALUATION AND EXPECTED RESULT

That command line should output the following text (with slight deviation in terms of the "Loop time", but none in the simulation data lines).

```
sw.in
  max neighbors/atom: 2000, page size: 100000
      0       1000   -138771.2          0     -134635   6866.6499
    100   508.80533  -136736.12          0   -134631.6   6361.7858
Loop time of 35.8894 on 1 procs for 100 steps with 32000 atoms
  max neighbors/atom: 2000, page size: 100000
      0       1000   -138771.2          0     -134635   6866.6499
    100   508.80533  -136736.12          0   -134631.6   6361.7858
Loop time of 38.5878 on 1 procs for 100 steps with 32000 atoms
sw.in.ref
  max neighbors/atom: 2000, page size: 100000
      0       1000   -138771.2          0     -134635   6866.6486
    100   508.80533  -136736.12          0   -134631.6   6361.7856
Loop time of 3.11945 on 1 procs for 100 steps with 32000 atoms
  max neighbors/atom: 2000, page size: 100000
      0       1000   -138771.2          0     -134635   6866.6499
    100   508.80533  -136736.12          0   -134631.6   6361.7858
Loop time of 21.1647 on 1 procs for 100 steps with 32000 atoms
tersoff.in
  max neighbors/atom: 2000, page size: 100000
      0       1000  -148173.19          0  -144036.99   7019.4434
    100   429.24901   -145810.1          0  -144034.64  -14610.887
Loop time of 13.0774 on 1 procs for 100 steps with 32000 atoms
  max neighbors/atom: 2000, page size: 100000
      0       1000  -148173.19          0  -144036.99   7019.4434
    100   429.24901   -145810.1          0  -144034.64  -14610.887
Loop time of 32.3869 on 1 procs for 100 steps with 32000 atoms
tersoff.in.ref
  max neighbors/atom: 2000, page size: 100000
      0       1000  -148173.19          0  -144036.99   7019.4434
    100   429.24901   -145810.1          0  -144034.64  -14610.887
Loop time of 9.61098 on 1 procs for 100 steps with 32000 atoms
  max neighbors/atom: 2000, page size: 100000
      0       1000  -148173.19          0  -144036.99   7019.4434
    100   429.24901   -145810.1          0  -144034.64  -14610.887
Loop time of 30.3192 on 1 procs for 100 steps with 32000 atoms
```