Performing density-functional tight-binding calculations with LAMMPS and the LATTE library


Los Alamos National Laboratory

August 4, 2017
1. Introduction
   - Quantum-based Molecular Dynamics

2. Computational framework

3. Latte scf-dftb physical model
   - Latte library

4. Linear scaling

5. Applications at LANL
Quantum based MD

- Takes the quantum nature of the chemical bond explicitly into account.
- Free from the arbitrariness of choosing a particular force field.
- Allows for formation and rupture of bonds (chemical reactions).
- Electronic properties can be monitored along the simulation.
- Computationally more demanding than classical MD.
Extending LAMMPS to DFTB-based QMD

**LAMMPS**: We benefit from most of the LAMMPS functionalities.
- Integration of the nuclear degrees of freedom (Verlet)
- Provides thermostats and barostats to perform NVT and NVP simulations.
- Provides all the infrastructure to do geometry optimization using different algorithms.
- We can compute transition barriers with the NEB method.

**LATTE (M. Cawkwell)**: Forces computed at the SCC-DFTB level.
- Provides forces and the potential component of the total energy.
- Integrates the electronic degrees of freedom (XL) (provides charges)
- Solves the electronic structure of the system.
- Several solvers such as SP2, Truncated SP2, Chebyshev polynomial expansion, etc.
Our computational framework:

- **LATTE**: A molecular dynamics package based around self-consistent-charge density functional tight binding theory (DFTB). We have also used this library in DFTB+.
- **PROGRESS**: This is a set of new capabilities offered as modules that rely entirely on BML. Several $\mathcal{O}(N)$ solvers for $\rho$ are available.
- **BML**: Only linear algebra or mathematically related operations are performed. We have a fast sparse matrix-matrix multiplication.

- [https://github.com/qmmd/bml](https://github.com/qmmd/bml)
- [https://github.com/lanl/qmd-progress](https://github.com/lanl/qmd-progress)
- [https://github.com/lanl/LATTE](https://github.com/lanl/LATTE)
LATTE scf-dftb physical model

\[ H_{i\alpha,j\beta}^0(r_{ij}) = A_0 \exp(A_1 r_{ij} + A_2 (r_{ij})^2 + A_3 (r_{ij})^3 + A_4 (r_{ij})^4) \]

\[ S_{i\alpha,j\beta}(r_{ij}) = B_0 \exp(B_1 r_{ij} + B_2 (r_{ij})^2 + B_3 (r_{ij})^3 + B_4 (r_{ij})^4) \]

\[ H_{i\alpha,j\beta} = H_{i\alpha,j\beta}^0 + \frac{1}{2} S_{i\alpha,j\beta}(V_i + V_j) \]

\[ P = [f(H)]_{SCF} \]

\[ q_i = \frac{1}{2} \sum_{\alpha \in i} (P_{j\beta,i\alpha} S_{i\alpha,j\beta} + P_{i\alpha,j\beta} S_{j\beta,i\alpha}) - n_i^e \]

\[ E = 2 \text{Tr}(PH) + \frac{1}{2} \sum_{i,j \neq i} \gamma_{ij} q_i q_j + E_{\text{pair}} \]

\[ F_{ij} = -\text{Tr} \left[ P \frac{\partial H}{\partial r_{ij}} \right] - \frac{1}{2} \sum_i \sum_{j \neq i} q_i q_j \frac{\partial \gamma_{ij}}{\partial r_k} - \frac{\partial E_{\text{pair}}}{\partial r_{ij}} \]
Why LATTE?

- Fast analytical physical model based on product of exponentials.
- Improves transferability.
- Provides linear scaling solvers for the density matrix.
- Provides linear scaling solvers for the inverse overlap.
- Provides the Extended Lagrangian method.

- Efforts in obtaining highly accurate parametrizations are ongoing (EPO code developed by **N. Aguirre**)
- Solvers will be ported to Exascale computing through ongoing projects.
Compiling latte as a library

- The standalone LATTE code can be easily compiled as a library (lattelib.a).

  ```
  $ cd; git clone git@github.com:lanl/LATTE.git
  $ cd ~/LATTE; make
  ```

- Both ifort (Intel) and gfortan (GNU) fortran compilers have been tested.

- LATTE can be easily linked to LAMMPS.

  ```
  $ cd ~/lammps/src
  $ make yes-latte
  $ make yes-molecule
  $ make serial
  ```
What do we need to run?

- LAMMPS usual input files (data.system and in.system)
- Add a “fix latte” to the input file:
  ```
  fix 2 all latte NULL
  ```
- LATTE control file (latte.in) and parameters folder (TBparams)
- Everything should be inside the same folder we are running.
- And we will run as:
  ```
  $ ~/lammps/src/lmp_serial < in.system
  ```

That’s all!
Linear scaling up to 3000 atoms system. Water boxes of different sizes. Calculations were performed on a 18-core 2.10 GHz Intel Xeon E5-2695 v4 processors.
Energy deviations for a 2955 atoms water box showing small energy fluctuations and energy drift. For this case the average fluctuation is no larger than $5 \mu\text{eV/atom}$ and the drift is about $3 \mu\text{eV/ps-atom}$
Linear scaling XLBO MD

- Linear scaling with low prefactor (green curve in graph)
- Full accuracy control.
- Perfect energy stability (green curve in the inset)

Sucrose conformers in water (A. Redondo)

Linear scaling XLBO NVT QMD.

Left: Sucrose molecule solvated in 535 water molecules. Right: Temperature as a function of the molecular dynamics time step for an simulation with constant number of particles, temperature and pressure (NTP).
Full SCF NVT QMD with low electronic temperature.

\[ \text{O}_2 + 2 \text{H}_2\text{O} + 4 \text{e}^- \rightarrow 4 \text{OH}^- \quad \text{(Reduction)} \]

\[ \text{Surf} \rightarrow \text{Surf}^{(4+)} + 4 \text{e}^- \quad \text{(Oxidation)} \]
Hugoniot shock compression (R. Perriot)

XL with limited SCF NVP QMD with high electronic and ionic temperature.

- 30 nitromethane molecules
- $P=24$ GPa, $T \sim 2000K$
- NVE for 500 + ps
- Monitor $T$, $P$, HOMO-LUMO
- Reaction starts at $\sim 230$ ps
- $\sim 15$ ps/day on 16 threads
NEB (CG geometry optimization) with high-accuracy SCF convergence and moderate electronic temperature.
... We have extended the capabilities of LAMMPS to do fast quantum-based molecular dynamics simulations. The interface to LATTE will be available in the next LAMMPS release.