3SPN.2 LAMMPS Implementation

Daniel M. Hinckley[1] and Juan J. de Pablo[2]

[1]Department of Chemical and Biological Engineering, University of Wisconsin-Maidon
[2]Institute for Molecular Engineering, University of Chicago

October 8, 2013

1. Introduction

3SPN.2 is an improvement on the previous version of the **3**-**S**ite-**P**er-**Nucleotide** (3SPN) coarse-grained DNA model. Key improvements include the replacement of Gō-like interactions with angle-dependent potentials, a reduction in the magnitude of the explicit charge on the phosphate sites, and modification of the dihedral potential to increase the flexibility of ssDNA. These improvements remedy a number of limitations that were identified by the members of the de Pablo group and others. The resulting model can capture the persistence lengths of both ssDNA and dsDNA, proper melting temperatures for duplexes and hairpins, and has stable major and minor grooves. For further details please see Hinckley, D. H. et al. J. Chem. Phys. 139,144903 (2013).

The 3SPN.2 coarse-grained model has been implemented in LAMMPS as a user package. This `USER-3SPN2` package contains this documentation file, sample .in input files for serial and parallel tempering calculations, a folder `DSIM_ICNF` that contains a configuration generator, and the source file to be added to the LAMMPS source. The following sections explain the particulars of the LAMMPS implementation, how to generate initial configurations for B–DNA, steps for running and visualizing trajectories, and limitations to this implementation. Also included are instructions for compiling the serial version of LAMMPS with 3SPN.2 from source. If you have any questions or problems, contact Dan Hinckley (dhinckley@wisc.edu).

2. Implementation Details

3SPN.2 consists of bond, angle and dihedral bonded interactions as well as numerous nonbonded interactions. For functional forms of these potentials, we refer the user to the model publication in JCP. The bonded interactions are implemented using existing LAMMPS potentials or slight modifications thereof. Bonds and angles are modeled using the *class2* and *harmonic* bond and angle styles, respectively. The Gaussian well dihedral potential is modeled using a new *3spn2* dihedral style (`dihedral_3spn2.cpp`). The base-stacking nonbonded interactions currently only occur between predetermined sets of sites. As such, they are implemented as a modified angle potential in `angle_3spn2_stacking.cpp`. Standard harmonic angles, as well as these "stacking" angles are both applied through a hybrid stacking interaction. The energies of both interactions are combined and output as `Eangle` in the thermo output.

All remaining interactions are captured by the `pair_3spn2.cpp`. When initialized, this pair style sets the value of equilibrium angles and distances. It also populates the arrays

specifying the strength of interactions and any modulating parameters. Lastly, it also creates a base pairing array that assigns flags used to determine whether or not cross-stacking interactions are to be calculated. This is because a base that is base pairing with a base at the end of a DNA strand often has no base with which to cross-stack. In order to determine whether or not a base is at the 5' or 3' end of DNA, special types (types 7-14) are assigned to the bases at the 5' and 3' ends. This is similar to the notation used in all-atom forcefields, with the obvious difference that in our coarse-grained representation the bases are topologically identical.

When the 3SPN.2 pair style is computed, first it determines whether or not the pair of sites (i,j) are on different molecules, separated by more than 3 nucleotides for the purposes of base pairing, and if the base pairing flag is set. It is also determined whether or not the bases are separated by at least 5 sites. If base pairing is to be performed, the absolute indices of the sites i and j are used to get the indices of the neighboring sites that participate in the angle-dependent potentials. Then a instance of the BasePair object is created and populated with the instantaneous angles and distances. That done, member functions are called to calculate the cross stacking and base pairing interactions. If base pairing interactions are not present, excluded volume interactions are calculated. Electrostatics interactions are then calculated and the resulting force and that from excluded volume (if applicable), are then applied to sites i and j.

The energies are saved to a vector that allows for extraction of these energies using a compute. The `3spn2.in` input file displays file to screen as follows:

```
<step> <num. bp> <Ebond> <Eangle(harmonic and stacking)> <Edihedral> ...
       <Ebp> <Ecstk> <Eelectro> <T>
```

3. Coefficients for New Styles

The new styles have the following coefficients:

- `dihedral_3spn2.cpp` -
  `dihedral_coeff [dihedral number] 3spn2 [`K_ϕ`] [`ϕ_o`] [`$\sigma_{\phi,o}$`]`
- `angle_3spn2_stacking.cpp` -
  `angle_coeff [angle number] stacking/3spn2 [`ϵ`] [`r_o`] [`θ_o`] [`α`] [`K`]`
- `pair_3spn2.cpp` -
  `pair_style [`T` (Kelvin)] [`I` (mM)] [Short Range Cutoff]`
  `pair_coeff 1 1 3spn2 [`ϵ`][`σ`]`

See the sample input files and the `conf_lammps.in` generated using the configuration generator described below for additional examples.

4. Generating an Initial Configuration

A configuration generator for generating B-DNA is provided in `DSIM_ICNF/` inside `USER-3SPN2`. Navigate into the directory and type `make` to build `icnf.exe`, which generates all of the needed files for simulation and visualization. It takes the following arguments:

```
./icnf.exe <sequence file> <complementarity flag (0-ssDNA; 1-dsDNA)> <output directory>
```

The sequence file is formatted as follows:

```
<NBPS>
<sense sequence (5'->3')>
<antisense sequence (3'->5')>
```

for example,

```
14
ATATATATATATAT
TATATATATATATA
```

If the antisense strand is not specified, it is assumed that you want completely complementary DNA. Three files are generated. The first is `in00_cvmd.psf`, a topology file that can be used in VMD to visualize the initial configuration. The second is `in00_conf.xyz`, an .xyz file containing the coordinates of each coarse-grained site. The last file is `conf_lammps.in`, the topology file used in LAMMPS.

After running `incf.exe`, it is possible to visualize the configuration in VMD. This is done with the following command:

```
vmd in00_cvmd.psf in00_conf.xyz
```

To capture the correct excluded volume of each site, go to Extensions>TkConsole and type `source path/to/spheres.vmd`. The `spheres.vmd` file is also found inside `USER-3SPN2`. Then, if you set the visualization style to VDW, you will have the correct excluded volume of the sites.

5. Input Files

Included in USER-3SPN2 are `3spn2.in`, `3spn2_restart.in` and `3spn2_temper.in`, the input files for serial, restarted, and parallel tempering simulations, respectively. Inside each input file, the temperature and ionic strength are specified in Kelvin and mM, respectively. Each file also reads the topology file, `conf_lammps.in`. It must be in the same working directory as the input file.

6. Generating a Movie

After performing a simulation, it is easy to visualize its trajectory in VMD with the following command:

```
vmd in00_cvmd.psf traj.xyz
```

As before, source the .vmd script to use the correct excluded volume. If you would like to render a particular frame of the trajectory, it can be done by going to File>Render. To make to a movie, go to Extensions>Visualization>Movie Maker. Most of the time it is desirable to only render a fraction of the total frames. The other frames can be dropped by right clicking on the trajectory in the main window.

7. Performing Parallel Tempering Calculations

3SPN.2 incorporates a temperature-dependent dielectric that must be varied with temperature. To account for this, the pair style *3spn2* takes a temperature fix as an optional 4th argument when performing parallel tempering calculations. An example is include below:

```
variable T world 300.0 315 330 345 360 375
...
pair_style      3spn2 ${T} ${salt} 18.0 tempfix
...
fix tempfix all langevin ${T} ${T} 1000 ${random}
...
temper 100000 1000 ${T} tempfix 0 ${random}
vmd in00_cvmd.psf traj.xyz
```

Note that the temperature is defined using a world-type variable and that this variable is used in the pair style and the Langevin thermostat fix. The temper command, which requires the REPLICA package to run, takes both the temperature and the thermostat fix as arguments. For a complete example, see `3spn2_temper.in`. To run the simulation, issue the command

```
mpirun -np 6 ../../lmp_openmpi -partition 6x1 -in 3spn2_temper.in
```

8. Limitation to LAMMPS Implementation

There are a number of limitations to the LAMMPS implementation that each user should be aware of. These are enumerated below:

(1) 3SPN.2 is performed in an implicit solvent. Consequently, the simulation box is mostly empty. This makes parallelization via LAMMPS's spatial decomposition horribly ineffective. 3SPN.2 will not scale well on multiple processors unless parallel tempering is being performed with each box being run by one processor.
(2) The virial (fdotr) is not currently being calculated properly.
(3) The Langevin integrator is specified with a default damping constant. This damping constant has not been rigorously optimized.

9. Compiling the source

The following instructions should be sufficient to compile the serial version of LAMMPS with 3SPN.2.

```
svn co svn://svn.icms.temple.edu/lammps-ro/trunk mylammps
cp -r USER-3SPN2 mylammps/src
cd mylammps/src
cd STUBS
make
cd ..
make yes-MOLECULE
```

```
make yes-CLASS2
make yes-USER-3SPN2
make serial
```