Practicing Dissipative Particle Dynamics

Yu-Hang Tang

CSRC, September 24, 2015
Format: Discussion-based
Feel free to interrupt at any time!
Dissipative Particle Dynamics is a stochastic, particle-based technique for simulating simple and complex fluids.
In DPD we track the motion of particles.

\[ m \cdot a(t) = f = -\frac{\partial H}{\partial x(t)} \]

\[ v \left( t + \frac{\Delta t}{2} \right) = v(t) + \frac{1}{2} a(t) \Delta t \]

\[ x(t + \Delta t) = x(t) + v \left( t + \frac{\Delta t}{2} \right) \Delta t \]

\[ a(t + \Delta t) = -\frac{\partial H}{\partial x(t + \Delta t)} \]

\[ v(t + \Delta t) = v \left( t + \frac{\Delta t}{2} \right) + \frac{1}{2} a(t + \Delta t) \Delta t \]
Short-range pairwise interactions

Lenard-Jones Potential

$$\sigma \left[ -2 \left( \frac{r}{r_0} \right)^6 + \left( \frac{r}{r_0} \right)^{12} \right]$$

DPD Linear Force

$$a_{ij} \left[ 1 - \frac{r}{r_0} \right]$$
**Bonded potential**

- **Bond**
- **Angle**
- **Dihedral**
- **Improper**
Periodic boundary conditions
approximate infinite systems by tiling unit cells.
A particle passing through the boundary re-appears on the opposite side with the same velocity.
minimum image convention

Bounce forward/back conditions
The **cell list** algorithm is ubiquitously employed to accelerate the evaluation of pairwise forces.
A neighbor list with skin distance may be even faster.
**Domain decomposition**

**Task decomposition**
LAMMPS

= 

Large-scale Atomic/Molecular Massively Parallel Simulator

http://lammps.sandia.gov/

✓ model particles at the atomic, meso, or continuum scale.
✓ serial or parallel using message-passing spatial decomposition
✓ easy to modify or extend
✓ open source (GPL)
Installation

Step 1. Download source
svn co svn://svn.lammps.org/lammps-ro/trunk lammps

Step 2. Compile
cd lammps/src
make serial OR make mpi

Step 3. Verify compilation
cd lammps/bench
../src/lmp_serial -in in.lj
mpirun -np 2 ../src/lmp_mpi -in in.lj
Enable **optional packages**

**Molecule package**

```
make yes-molecule
```

**USER MESO package (requires CUDA and MPI)**

```
svn co http://www.cfm.brown.edu/repo/release/USER-MESO/ meso
cd meso
make yes-molecule
make yes-user-meso
make meso
```
Example I: Binary Immiscible liquid

Script at

http://www.cfm.brown.edu/repo/release/DPD-workshop/binary.lmp

Once finished, visualize with VMD

vmd binary.lammpstrj
# Line-by-line: # DPD binary mixture

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td># DPD binary mixture</td>
<td></td>
</tr>
<tr>
<td>dimension</td>
<td>3</td>
</tr>
<tr>
<td>units</td>
<td>lj</td>
</tr>
<tr>
<td>comm_modify</td>
<td>vel yes</td>
</tr>
<tr>
<td>newton</td>
<td>off</td>
</tr>
<tr>
<td>atom_style</td>
<td>atomic</td>
</tr>
<tr>
<td>neighbor</td>
<td>0.3 bin</td>
</tr>
<tr>
<td>neigh_modify</td>
<td>delay 0 every 4 check no</td>
</tr>
<tr>
<td>region</td>
<td>domain block -5 5 -5 5 -5 5 units box</td>
</tr>
<tr>
<td>create_box</td>
<td>2 domain</td>
</tr>
<tr>
<td>lattice</td>
<td>fcc 4</td>
</tr>
<tr>
<td>create_atoms</td>
<td>1 random 1500 15255 domain</td>
</tr>
<tr>
<td>create_atoms</td>
<td>2 random 1500 47214 domain</td>
</tr>
<tr>
<td>mass</td>
<td>* 1.0</td>
</tr>
<tr>
<td>pair_style</td>
<td>dpd 1.0 1.0 419084618</td>
</tr>
<tr>
<td>pair_coeff</td>
<td>1 1 25.0 4.5</td>
</tr>
<tr>
<td>pair_coeff</td>
<td>1 2 150.0 4.5</td>
</tr>
<tr>
<td>pair_coeff</td>
<td>2 2 25.0 4.5</td>
</tr>
<tr>
<td>thermo</td>
<td>100</td>
</tr>
<tr>
<td>dump</td>
<td>dump0 all atom 5 binary.lammpstrj</td>
</tr>
<tr>
<td>velocity</td>
<td>all create 1.0 8178251</td>
</tr>
<tr>
<td>fix</td>
<td>integrator all nve</td>
</tr>
<tr>
<td>timestep</td>
<td>0.01</td>
</tr>
<tr>
<td>run</td>
<td>10000</td>
</tr>
</tbody>
</table>

A comment. Specifically, the # sign may appear anywhere in a line, and all characters from there onward are treated as comment and discarded.
Line-by-line: **dimension 3**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code># DPD binary mixture</code></td>
<td></td>
</tr>
<tr>
<td><code>dimension 3</code></td>
<td>Set the dimensionality of the simulation</td>
</tr>
<tr>
<td><code>units lj</code></td>
<td></td>
</tr>
<tr>
<td><code>comm_modify vel yes</code></td>
<td></td>
</tr>
<tr>
<td><code>newton off</code></td>
<td></td>
</tr>
<tr>
<td><code>atom_style atomic</code></td>
<td></td>
</tr>
<tr>
<td><code>neighbor 0.3 bin</code></td>
<td></td>
</tr>
<tr>
<td><code>neigh_modify delay 0 every 4 check no</code></td>
<td></td>
</tr>
<tr>
<td><code>region domain block -5 5 -5 5 -5 5 units box</code></td>
<td></td>
</tr>
<tr>
<td><code>create_box 2 domain</code></td>
<td></td>
</tr>
<tr>
<td><code>lattice fcc 4</code></td>
<td></td>
</tr>
<tr>
<td><code>create_atoms 1 random 1500 15255 domain</code></td>
<td></td>
</tr>
<tr>
<td><code>create_atoms 2 random 1500 47214 domain</code></td>
<td></td>
</tr>
<tr>
<td><code>mass * 1.0</code></td>
<td></td>
</tr>
<tr>
<td><code>pair_style dpd 1.0 1.0 419084618</code></td>
<td></td>
</tr>
<tr>
<td><code>pair_coeff 1 1 25.0 4.5</code></td>
<td></td>
</tr>
<tr>
<td><code>pair_coeff 1 2 150.0 4.5</code></td>
<td></td>
</tr>
<tr>
<td><code>pair_coeff 2 2 25.0 4.5</code></td>
<td></td>
</tr>
<tr>
<td><code>thermo 100</code></td>
<td></td>
</tr>
<tr>
<td><code>dump dump0 all atom 5 binary.lammpstrj</code></td>
<td></td>
</tr>
<tr>
<td><code>velocity all create 1.0 8178251</code></td>
<td></td>
</tr>
<tr>
<td><code>fix integrator all nve</code></td>
<td></td>
</tr>
<tr>
<td><code>timestep 0.01</code></td>
<td></td>
</tr>
<tr>
<td><code>run 10000</code></td>
<td></td>
</tr>
</tbody>
</table>
### Line-by-line: `units lj`

```plaintext
# DPD binary mixture

dimension 3
units lj
comm_modify vel yes
newton off

atom_style atomic
neighbor 0.3 bin
neigh_modify delay 0 every 4 check no

region domain block -5 -5 5 -5 5 units box
create_box 2 domain
lattice fcc 4
create_atoms 1 random 1500 15255 domain
create_atoms 2 random 1500 47214 domain
mass * 1.0

pair_style dpd 1.0 1.0 419084618
pair_coeff 1 1 25.0 4.5
pair_coeff 1 2 150.0 4.5
pair_coeff 2 2 25.0 4.5

thermo 100
dump dump0 all atom 5 binary.lammpstrj

velocity all create 1.0 8178251
fix integrator all nve

timestep 0.01
run 10000
```

Set the unit used in the simulation to be 'lj', i.e. reduced units.
Line-by-line: **comm_modify vel yes**

Exchange velocity data of particles during MPI parallel execution; necessary for DPD.
### Line-by-line: newton off

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>DPD binary mixture</td>
</tr>
<tr>
<td>dimension</td>
<td>3</td>
</tr>
<tr>
<td>units</td>
<td>lj</td>
</tr>
<tr>
<td>comm_modify vel yes</td>
<td>newton off</td>
</tr>
<tr>
<td>atom_style</td>
<td>atomic</td>
</tr>
<tr>
<td>neighbor 0.3 bin</td>
<td>neigh_modify delay 0 every 4 check no</td>
</tr>
<tr>
<td>region domain block -5 -5 -5 5 5 units box</td>
<td>create_box 2 domain</td>
</tr>
<tr>
<td>lattice fcc 4</td>
<td>create_atoms 1 random 1500 15255 domain</td>
</tr>
<tr>
<td>create_atoms 2 random 1500 47214 domain</td>
<td>mass * 1.0</td>
</tr>
<tr>
<td>pair_style dpd 1.0 1.0 419084618</td>
<td>pair_coeff 1 1 25.0 4.5</td>
</tr>
<tr>
<td>pair_coeff 1 2 150.0 4.5</td>
<td>pair_coeff 2 2 25.0 4.5</td>
</tr>
<tr>
<td>thermo 100</td>
<td>dump dump0 all atom 5 binary.lammpstrj</td>
</tr>
<tr>
<td>velocity all create 1.0 8178251</td>
<td>fix integrator all nve</td>
</tr>
<tr>
<td>timestep 0.01</td>
<td>run 10000</td>
</tr>
</tbody>
</table>

Whether or not to exploit Newton's 3\textsuperscript{rd} law to accelerate computation.

Syntax: newton on|off

Setting newton to on accelerates serial computation but may lower performance at high parallelism; newton off mandatory for GPU acceleration.
Line-by-line: **atom_style atomic**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td># DPD binary mixture</td>
<td>Select style of atom container.</td>
</tr>
<tr>
<td>dimension 3</td>
<td>Syntax: atom_style style</td>
</tr>
<tr>
<td>units lj</td>
<td>style = atomic, bond, full, etc.</td>
</tr>
<tr>
<td>comm_modify vel yes</td>
<td>Each atom style carries a different set of degrees of freedom.</td>
</tr>
<tr>
<td>newton off</td>
<td></td>
</tr>
<tr>
<td>atom_style atomic</td>
<td></td>
</tr>
<tr>
<td>neighbor 0.3 bin</td>
<td></td>
</tr>
<tr>
<td>neigh_modify delay 0 every 4 check no</td>
<td></td>
</tr>
<tr>
<td>region domain block -5 -5 -5 5 5 units box</td>
<td></td>
</tr>
<tr>
<td>create_box 2 domain</td>
<td></td>
</tr>
<tr>
<td>lattice fcc 4</td>
<td></td>
</tr>
<tr>
<td>create_atoms 1 random 1500 15255 domain</td>
<td></td>
</tr>
<tr>
<td>create_atoms 2 random 1500 47214 domain</td>
<td></td>
</tr>
<tr>
<td>mass * 1.0</td>
<td></td>
</tr>
<tr>
<td>pair_style dpd 1.0 1.0 419084618</td>
<td></td>
</tr>
<tr>
<td>pair_coeff 1 1 25.0 4.5</td>
<td></td>
</tr>
<tr>
<td>pair_coeff 1 2 150.0 4.5</td>
<td></td>
</tr>
<tr>
<td>pair_coeff 2 2 25.0 4.5</td>
<td></td>
</tr>
<tr>
<td>thermo 100</td>
<td></td>
</tr>
<tr>
<td>dump dump0 all atom 5 binary.lammpstrj</td>
<td></td>
</tr>
<tr>
<td>velocity all create 1.0 8178251</td>
<td></td>
</tr>
<tr>
<td>fix integrator all nve</td>
<td></td>
</tr>
<tr>
<td>timestep 0.01</td>
<td></td>
</tr>
<tr>
<td>run 10000</td>
<td></td>
</tr>
</tbody>
</table>
Line-by-line: `neighbor 0.3 bin`

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension</td>
<td>3</td>
</tr>
<tr>
<td>units</td>
<td>lj</td>
</tr>
<tr>
<td>comm_modify</td>
<td>vel yes</td>
</tr>
<tr>
<td>newton</td>
<td>off</td>
</tr>
<tr>
<td>atom_style</td>
<td>atomic</td>
</tr>
<tr>
<td>neighbor</td>
<td>0.3 bin</td>
</tr>
<tr>
<td>neigh_modify</td>
<td>delay 0 every 4 check no</td>
</tr>
<tr>
<td>region</td>
<td>domain block -5 5 -5 5 -5 5 units box</td>
</tr>
<tr>
<td>create_box</td>
<td>2 domain</td>
</tr>
<tr>
<td>lattice</td>
<td>fcc 4</td>
</tr>
<tr>
<td>create_atoms</td>
<td>1 random 1500 15255 domain</td>
</tr>
<tr>
<td>create_atoms</td>
<td>2 random 1500 47214 domain</td>
</tr>
<tr>
<td>mass</td>
<td>* 1.0</td>
</tr>
<tr>
<td>pair_style</td>
<td>dpd 1.0 1.0 419084618</td>
</tr>
<tr>
<td>pair_coeff</td>
<td>1 1 25.0 4.5</td>
</tr>
<tr>
<td>pair_coeff</td>
<td>1 2 150.0 4.5</td>
</tr>
<tr>
<td>pair_coeff</td>
<td>2 2 25.0 4.5</td>
</tr>
<tr>
<td>thermo</td>
<td>100</td>
</tr>
<tr>
<td>dump</td>
<td>dump0 all atom 5 binary.lammpstrj</td>
</tr>
<tr>
<td>velocity</td>
<td>all create 1.0 8178251</td>
</tr>
<tr>
<td>fix</td>
<td>integrator all nve</td>
</tr>
<tr>
<td>timestep</td>
<td>0.01</td>
</tr>
<tr>
<td>run</td>
<td>10000</td>
</tr>
</tbody>
</table>

Select style of neighbor list.

Syntax: `neighbor skin style`  
`skin = skin distance`  
`style = bin | nsq`
Line-by-line: `neigh_modify delay 0 every 4 check no`

Configure neighbor list update frequency.

Syntax: every M delay N check yes|no

delay: wait for N steps before rebuilding
every: rebuild neighbor list every M steps
check: only rebuild if some atom has moved more than half the skin.
Line-by-line: **region domain block -5 5 -5 5 -5 5 units box**

Define a region in the 3D space for later use; multiple regions can be define within each simulation

Syntax: region ID style args

In this case we built a cubic region of size 10x10x10 centered at the origin
Line-by-line: create_box 2 domain

Create a simulation box based on the region previously defined.

Syntax: create_box N region

N: max number of atom types used in the simulation
Line-by-line: **lattice fcc 4**

Define a lattice for the subsequent 'create_atoms' command

Syntax: lattice style scale
style = none | sc | bcc | fcc etc...

Note: this line is actually unused in the presented script but will be part of an exercise.
Line-by-line: create_atoms 1 random 1500 15255 domain

Create atoms randomly in the simulation box

Syntax: create_atoms type style args
Specifically: create_atoms type random N seed region

type: type of the created particles
N: number of particles to create
seed: seed for random number generator
region: only create atoms in the region
Line-by-line: \texttt{mass} * 1.0

Set the mass for each type of particle

Syntax: \texttt{mass} type value

type: number or range or *
m: type m

\texttt{n*: all types from n to N}

\texttt{m*n: all types from m to n}

\texttt{*: all types}
Line-by-line: pair_style dpd 1.0 1.0 419084618

Use the DPD style for pairwise force evaluation.

Syntax: pair_style dpd k_B T r_c seed
k_B: temperature
r_c: cutoff distance
seed: seed for random number generator
Line-by-line: `pair_coeff 1 1 25.0 4.5`

Set up pairwise coefficients

Syntax: `pair_coeff type1 type2 A gamma [r_c]`

A: conservative force constant
gamma: dissipation strength

`r_c` (optional): cutoff for a specific pair
**Line-by-line: thermo 100**

# DPD binary mixture

dimension 3
units lj
comm_modify vel yes
newton off

atom_style atomic
neighbor 0.3 bin
neigh_modify delay 0 every 4 check no

region domain block -5 5 -5 5 -5 5 units box
create_box 2 domain
lattice fcc 4
create_atoms 1 random 1500 15255 domain
create_atoms 2 random 1500 47214 domain
mass * 1.0

pair_style dpd 1.0 1.0 419084618
pair_coeff 1 1 25.0 4.5
pair_coeff 1 2 150.0 4.5
pair_coeff 2 2 25.0 4.5

thermo 100
dump dump0 all atom 5 binary.lammpstrj

velocity all create 1.0 8178251
fix integrator all nve

timestep 0.01
run 10000

display diagnostics every N steps
# DPD binary mixture

dimension 3
units lj
comm_modify vel yes
newton off

atom_style atomic
neighbor 0.3 bin
neigh_modify delay 0 every 4 check no

region domain block -5 5 -5 5 -5 5 units box
create_box 2 domain
lattice fcc 4
create_atoms 1 random 1500 15255 domain
create_atoms 2 random 1500 47214 domain
mass * 1.0

pair_style dpd 1.0 1.0 419084618
pair_coeff 1 1 25.0 4.5
pair_coeff 1 2 150.0 4.5
pair_coeff 2 2 25.0 4.5

thermo 100
dump dump0 all atom 5 binary.lammpstrj

velocity all create 1.0 8178251
fix integrator all nve

timestep 0.01
run 10000

Configure trajectory output

Syntax: dump ID group style args

ID: ID for the dump; multiple dumps with different IDs may be defined within a simulation.

group: the groups of particles to dump; a group 'all' containing all particles is predefined in LAMMPS

style: atom – LAMMPS’s own trajectory format; may also be dcd, xyz etc.

args: *atom = N* filename
Line-by-line: velocity all create 1.0 8178251

Set particle initial velocity

Syntax: velocity group style args
Specifically:
velocity group create T seed
velocity group set vx vy vz
# DPD binary mixture

dimension 3
units lj
comm_modify vel yes
newton off

atom_style atomic
neighbor 0.3 bin
neigh_modify delay 0 every 4 check no

region domain block -5 5 -5 5 -5 5 units box
create_box 2 domain
lattice fcc 4
create_atoms 1 random 1500 15255 domain
create_atoms 2 random 1500 47214 domain
mass * 1.0

pair_style dpd 1.0 1.0 419084618
pair_coeff 1 1 25.0 4.5
pair_coeff 1 2 150.0 4.5
pair_coeff 2 2 25.0 4.5

thermo 100
dump dump0 all atom 5 binary.lammpstrj

velocity all create 1.0 8178251
fix integrator all nve

timestep 0.01
run 10000

**Line-by-line: fix integrator all nve**

Uses the NVE integrator as a 'LAMMPS fix'.

**Syntax: fix ID group style**
# DPD binary mixture

dimension 3
units lj
comm_modify vel yes
newton off

atom_style atomic
neighbor 0.3 bin
neigh_modify delay 0 every 4 check no

region domain block -5 -5 -5 5 5 5 units box
create_box 2 domain
lattice fcc 4
create_atoms 1 random 1500 15255 domain
create_atoms 2 random 1500 47214 domain
mass * 1.0

pair_style dpd 1.0 1.0 419084618
pair_coeff 1 1 25.0 4.5
pair_coeff 1 2 150.0 4.5
pair_coeff 2 2 25.0 4.5

thermo 100
dump dump0 all atom 5 binary.lammpstrj

velocity all create 1.0 8178251
fix integrator all nve

timestep 0.01
run 10000
Line-by-line: run 10000

Run the simulation for 10000 steps
Instead of 'creating' particles we can also load them through data file.

Each atom style takes a slightly different format, i.e. different # of columns in the Atoms section, etc.

Check LAMMPS website for the specifics.

# data file sample
1000 atoms
1 atom types
0 10 xlo xhi
0 10 ylo yhi
0 10 zlo zhi
Masses
1 1.000000
Atoms
1 1 8.401877172 3.943829268 7.830992238 7.830992238
2 1 7.984400335 9.116473579 1.975513693 1.975513693
3 1 3.352227557 7.682295948 2.777747108 2.777747108
...
Style

Styles is the mechanism used by LAMMPS to allow scalable functionality extension.

From a user's perspective they serve as options to various LAMMPS commands.

From a developer's perspective they are C++ derived classes of a virtual base class. Overloading the virtual interfaces allows the custom functionalities to be implemented.
Groups

A group is a collection of particles. The group ID can be used in other commands such as fix, compute, dump, or velocity to act on those atoms together.

Syntax:

group ID style args

style = region, type, etc...
Fix

A 'fix' is any operation that is applied to the system during time stepping. Examples include updating of atom positions and velocities due to time integration, controlling temperature, applying constraint forces to atoms, enforcing boundary conditions, computing diagnostics, etc.

Fixes perform their operations at different stages of the timestep.

Advanced topic: implementing custom fixes
Verlet scheme, stages and fix class
Thank you for attending!

Y.H.T. appreciates invitation and support from Dr. Zhen Li, Dr. Zhewei Zhou, Dr. George Karaniadakis, and Shanghai Institute of Applied Mathematics and Mechanics, Shanghai Center for Nonlinear Sciences, Pacific Northwest National Lab, and Beijing Computational Science Research Center.