Coarse-graining breakout session

Dan S. Bolintineanu

2017 LAMMPS Workshop and Symposium

8/3/2017
Outline

Meso/macroscopic models in LAMMPS
- Contributed talk: Svetoslav Nikolov
  “Modeling non-linear micromechanics of hydrogels using dissipative particle dynamics”
- DPD, SPH, SMD
- Hydrodynamics: SRD, Lattice Boltzmann, FLD, Mango-Selm, …
- Granular
- Peridynamics

Molecular and macroscopic:
- USER-ATC package

‘Detailed’ MD to coarse-grained MD
- Iterative Boltzmann, force matching, …
- Tools: VOTCA, fix mscg, PyCGTOOL, Auto_MARTINI
- CG force fields: MARTINI, USER-ELBA

Meso/macropscopic models in LAMMPS

“Top-down” : particles are simply a numerical device for solving governing physics
→ typically mass, momentum conservation, i.e. fluid dynamics (NS)
→ many other physics also possible

DPD:
See George Karniadakis talk! → USER-MESO
See Tim Mattox talk! → USER-DPD: NPT, NPH, equation of state → includes
chemical reaction
pair_style dpd: ‘plain vanilla’ DPD
pair_style dpd/tstat: apply pairwise DPD thermostat to any pair-style
angular-momentum conserving variant?

SPH:
Definitely not ‘physical’ particles, just a Lagrangian solver of continuum equations!
USER-SPH, USER-SMD (mitigates ‘tensile instabilities’)
iSPH – implicit time-stepping, coupled to Trilinos

Related methods:
SDPD: ‘thermostatted’ SPH, i.e. solver for fluctuating
hydrodynamics (Espanol and Revenga, PRE 2003)

\[ f(r) \approx \sum \frac{m_j}{\rho_j} f(r_j) W(|r - r_j|; h) \]
SPH gallery

USER-SPH (Georg Ganzenmuller):

DualSPHysics:

GPU: 64 x M2090 (BSC)
MPI: Dynamic balancing
Algorithm: Verlet & Wendland
Particles: 1.013 Million
Segs: 297,342
Runtime: 91.9 hours
Physical time: 12 seconds
Hydrodynamics in colloidal suspensions

Colloid particles macroscopic ~1 micron: COLLOID package, fix nve/sphere
Solvent: (fluctuating) hydrodynamics!

- Particle methods (DPD, etc.)
- Lattice Boltzmann (USER-LB), stochastic rotation dynamics (SRD)
- FLD: simplified version of Stokesian dynamics
- MANGO-SELM: Atzberger group; solvers of fluctuating hydrodynamics
Hydrodynamics in colloidal suspensions

All particle methods suffer from:
- Compressibility (not a big deal)
- ‘Lubrication gap’ resolution problems
- Boundary conditions can be tricky

Implicit methods (SD, FLD):
- Limited to simple particle shapes, domain geometries
- Newtonian only

Vázquez-Quesada et al, CPM 2016

Dynamics

FLD

SRD/MPCD

DPD*

Structure

Bolintineanu et al, CPM 2014
Granular

pair gran/hertz:

Analytical Hertz solution (1882 for normal contact force:

$\delta = R_i + R_j - \|r_i - r_j\| > 0$

$F_n = k_n \sqrt{R\delta^{3/2}} n - \sqrt{R}\delta m\gamma_n v_n$

What about oblique contact/tangential force?
→ friction, with option of accumulated shear

pair gran/hertz/history

Dissipative term;
Many damping models

$F_S = -k_S \int_{t_0}^{t} v_{tR}(\tau)d\tau - \eta_T v_{tR}$

$v_{tR} = v_t - (R_i\Omega_i + R_j\Omega_j) \times n$

$\|F_S\| \leq \mu_S \|F_n\|$

Also see: LIGGGHTS/CFDEM project
Peridynamics

- Nonlocal formulation of solid mechanics
- Good alternative to e.g. XFEM for modeling fracture/fragmentation
- PERI package in LAMMPS
- Not frequently maintained, much of peridynamics development now in Peridigm

Stewart Silling, Mike Parks, many others (SNL)
Molecular and macroscopic

USER-AtC package:

• Molecular simulation in one region coupled to continuum (e.g. FEM)
• Couple continuum equations throughout domain to molecular system
• On-the-fly analysis of MD to extract continuum quantities

Reese Jones, Jeremy Templeton (both SNL)
‘Detailed’ MD to coarse-grained MD

Basic idea: access larger length/time scales by using less detailed representations

This works because:
1. Fewer particles!
2. ‘Softer’, sometimes simpler potentials → enables larger time step

• Not limited to atomistic→bead-spring; hierarchical also possible

Methods

Need to:
1.) Select chemical groups to coarse-grain
2.) Obtain parameters

‘DIY’ coarse-graining, e.g.
• Force matching
• (Iterative) Boltzmann inversion

\[
U_{\alpha\beta,0}(r) = -k_B T \ln(g_{\alpha\beta}(r))
\]

\[
U_{\alpha\beta,i+1}(r) = U_{\alpha\beta,i}(r) + k_B T \ln \left( \frac{g_{\alpha\beta,i}(r)}{g_{\alpha\beta}(r)} \right)
\]

Note: all of these map only static properties, dynamics may be way off!

‘Pre-packaged’ CG force fields:
MARTINI, ELBA, USER-CGDNA, various UA (e.g. OLPS-UA)

Tools:
VOTCA, fix mscg, PyCGTOOL, Auto_MARTINI