Modeling Thermal Transport and Viscosity with Molecular Dynamics

Steve Plimpton
Sandia National Labs
sjplimp@sandia.gov

LAMMPS Users and Developers Workshop
International Centre for Theoretical Physics (ICTP)
March 2014 - Trieste, Italy

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.
Presentation: SAND2014-2225C

Tackling a new problem with LAMMPS

- Features: lammps.sandia.gov/features.html
- Commands page: doc/Section_Commands.html
- Papers: lammps.sandia.gov/papers.html
- Mail list: lammps.sandia.gov/mail.html
- Adding new features: doc/Section_modify.html
- Howto explanations in manual:
  - doc/Section_howto.html
  - 6.20 Calculating thermal conductivity
  - 6.21 Calculating viscosity

What is thermal conductivity?

- Propensity of a material to transmit heat (thermal energy)
- Solids or liquids or gases
- Temperature and density dependent
- High κ = good heat sink, low κ = good insulator

What is thermal conductivity?

- Propensity of a material to transmit heat (thermal energy)
- Solids or liquids or gases
- Temperature and density dependent
- High κ = good heat sink, low κ = good insulator
- Fundamental equation:

$$J = -\kappa \nabla T$$

$$J = \text{heat flux} = \frac{\Delta KE}{\text{Area time}}$$

$$\nabla T = \text{temperature gradient} = dT/dz$$

$$\kappa = \text{thermal conductivity} = \text{W / m K}$$

What is viscosity?

- Propensity of a fluid to transmit momentum perpendicular to direction of momentum flow (shear direction)
- Fluid "friction" or resistance to flow
- Fluid = Liquids and gases
- High η = honey, low η = water

What is viscosity?

- Propensity of a fluid to transmit momentum perpendicular to direction of momentum flow (shear direction)
- Fluid "friction" or resistance to flow
- Fluid = Liquids and gases
- High η = honey, low η = water
- Fundamental equation:

$$J_z(p_x) = -\eta \frac{\partial V_x}{\partial z}$$

$J_z(p_x)$ = momentum flux in perpendicular direction

$\frac{\partial V_x}{\partial z}$ = transverse velocity gradient

η = shear viscosity

4 methods for computing thermal conductivity

- Non-equilibrium methods:
  - basic idea: induce a temperature gradient or heat flux and monitor the other quantity
  - direct thermostatting method of Ikeshoji and Hafskjold
  - reverse perturbation method of Muller-Plathe
  - aggregate variant of Muller-Plathe method
- Equilibrium method:
  - Green-Kubo formalism

4 methods for computing thermal conductivity

- Non-equilibrium methods:
  - basic idea: induce a temperature gradient or heat flux and monitor the other quantity
  - direct thermostatting method of Ikeshoji and Hafskjold
  - reverse perturbation method of Muller-Plathe
  - aggregate variant of Muller-Plathe method
- Equilibrium method:
  - Green-Kubo formalism

- See examples/KAPPA for 4 sample scripts
- 3d LJ fluid, but adaptable to other systems (e.g. solids)

4 methods for computing shear viscosity

- Non-equilibrium methods:
  - basic idea: induce a flow gradient or momentum flux and monitor the other quantity
  - drag wall over fluid to induce shear
  - NEMD shear deformation with SLLOD thermostatting
  - Muller-Plathe reverse perturbation method
- Equilibrium method:
  - Green-Kubo formalism
  - auto-correlation of pressure tensor component

4 methods for computing shear viscosity

- Non-equilibrium methods:
  - basic idea: induce a flow gradient or momentum flux and monitor the other quantity
  - drag wall over fluid to induce shear
  - NEMD shear deformation with SLLOD thermostatting
  - Muller-Plathe reverse perturbation method
- Equilibrium method:
  - Green-Kubo formalism
  - auto-correlation of pressure tensor component

- See examples/VISCOSITY for 4 sample scripts
- 2d LJ fluid, but adaptable to other systems

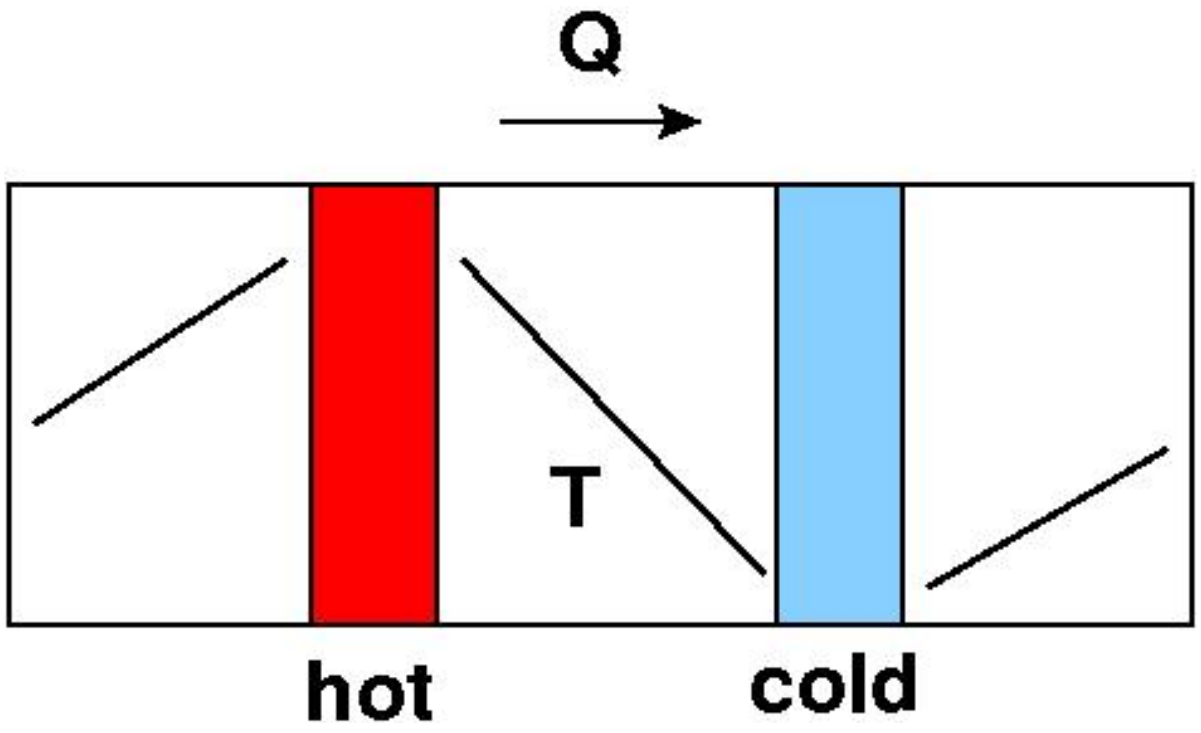
Caveats for atomistic MD

1. Missing electronic effects for κ
   - empirical atomistic simulations $\Rightarrow$
     heat is transported by phonons
   - electronic effects included only indirectly in potential
   - if electrons make large contribution to κ, won't see it
2. Homogeneous vs heterogeneous systems
   - formulas are for homogeneous bulk
   - κ in graphene sheets is 2d, possibly asymmetric
   - η for fluid flowing thru CNTs is radial BC
3. Mis-match to experiment
   - MD has severe length- and time-scale constraints
   - temperature gradients & shear rates are typically
     orders of magnitude larger than expt

(1) Direct thermostatting method

Ikeshoji and Hafskjold, Molecular Physics, 81, 251-261 (1994)

- 2 thermostats for 2 regions of simulation box
- One hot, one cold
- Monitor flux of energy needed to maintain ∇T



Direct thermostatting method

- LAMMPS implementation:
  - fix langevin using compute temp/region as "bias"
  - fix langevin can tally energy each thermostat adds/subtracts
  - fix ave/spatial monitors resulting temperature gradient

$$\kappa = \frac{\Delta Q}{2 Area \Delta t} \frac{\Delta z}{\Delta T}$$

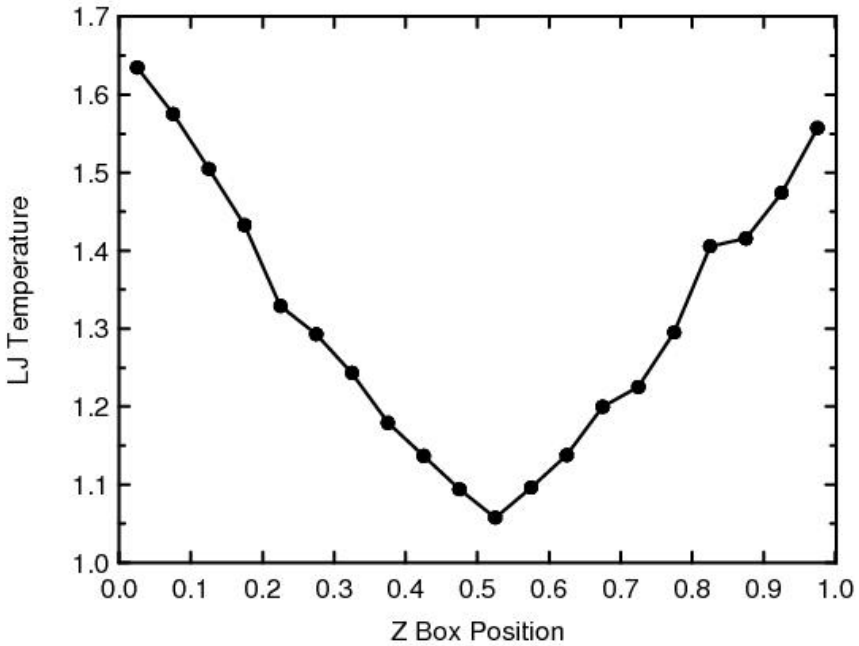
Script for direct thermostatting method

```
lattice fcc ${rho}
region box block 0 $x 0 $y 0 $z

# heat layers

region hot block INF INF INF INF 0 1
region cold block INF INF INF INF 10 11
compute Thot all temp/region hot
compute Tcold all temp/region cold

# 1st equilibration run

fix 1 all nvt temp $t $t 0.5
run 1000
unfix 1
```

More script for direct thermostatting method

```
# thermal conductivity calculation

compute ke all ke/atom
variable temp atom c_ke/1.5

fix hot all langevin ${thi} ${thi} 1.0 59804 tally yes
fix cold all langevin ${tlo} ${tlo} 1.0 287859 ...
fix_modify hot temp Thot
fix_modify cold temp Tcold

fix 2 all ave/spatial 10 100 1000 z lower 0.05 v_temp &
  file tmp.profile units reduced

thermo_style custom step temp c_Thot c_Tcold f_hot f_cold
run 20000
```

Output for direct thermostatting method



Step Temp Thot Tcold hot cold

...

30000 1.3011151 1.7275961 1.06067 -0.84589474 0.8965726

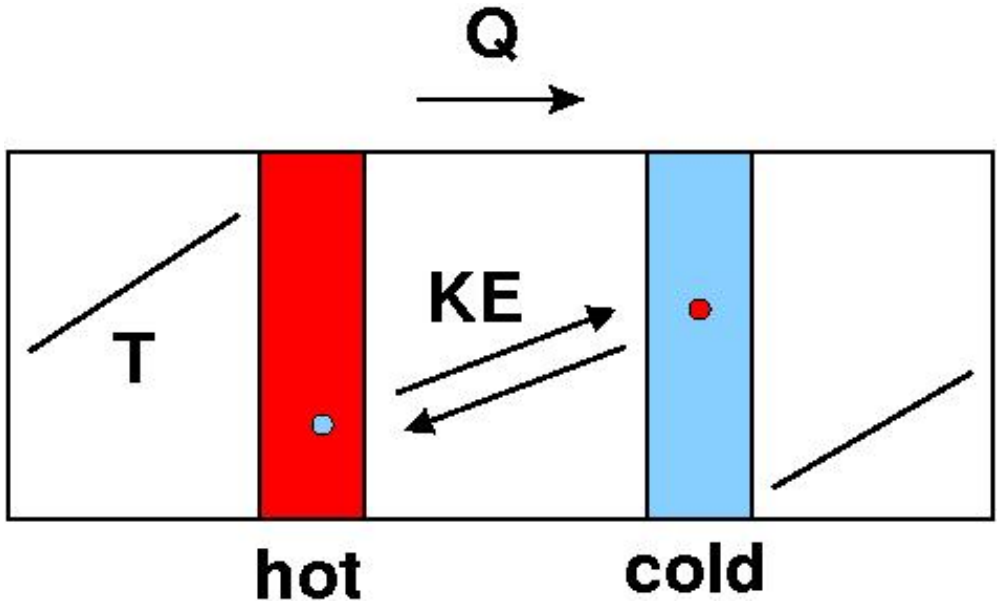
31000 1.3002026 1.5313418 1.0526131 -0.8964083 0.93984929

Loop time of 25.7381 on 8 procs for 20000 steps with 8000 atoms

(2) Muller-Plathe reverse perturbation method

Muller-Plathe, J Chem Phys, 106, 6082 (1997)

- Define hot and cold regions of simulation box
- Find hottest atom in cold region, coldest atom in hot region
- Swap velocity vector of these 2 atoms (energy)
- Tally heat flux due to KE exchanges
- Monitor the induced temperature profile
- Reverse of previous method



Muller-Plathe reverse perturbation method

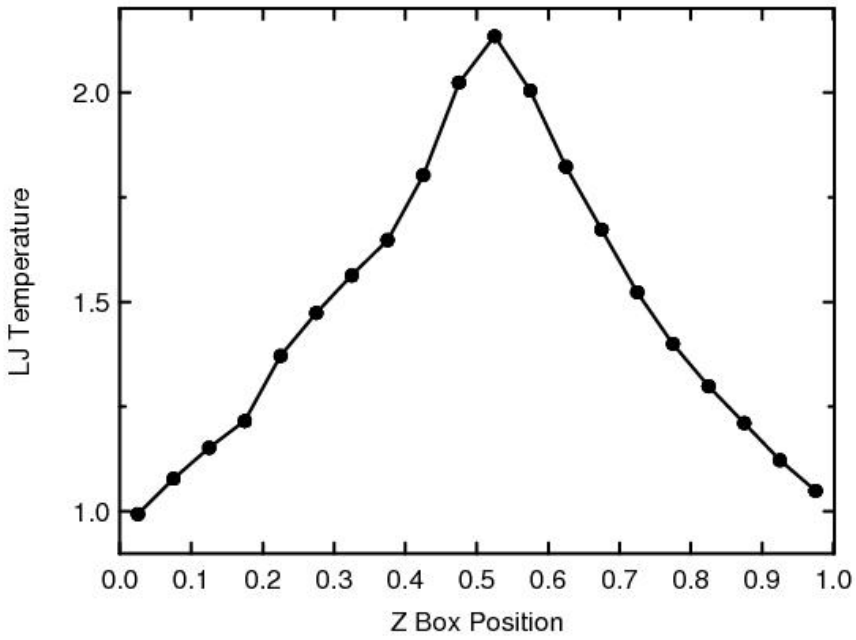
- LAMMPS implementation:
  - fix thermal/conductivity swaps KE and tallies heat flux
  - fix ave/spatial monitors induced temperature gradient

$$\kappa = \frac{\Delta Q}{2 Area \Delta t} \frac{\Delta z}{\Delta T}$$

Script for Muller-Plathe reverse method

```
# thermal conductivity calculation

compute ke all ke/atom
variable temp atom c_ke/1.5

fix 1 all nve
fix 2 all ave/spatial 10 100 1000 z lower 0.05 &
  v_temp file tmp.profile units reduced
fix 3 all thermal/conductivity 10 z 20

variable tdiff equal f_2[11][3]-f_2[1][3]
thermo_style custom step temp epair etotal &
  f_3 v_tdiff

run 20000
```

Output for Muller-Plathe reverse method



Step Temp E_pair TotEng 3 tdiff

...

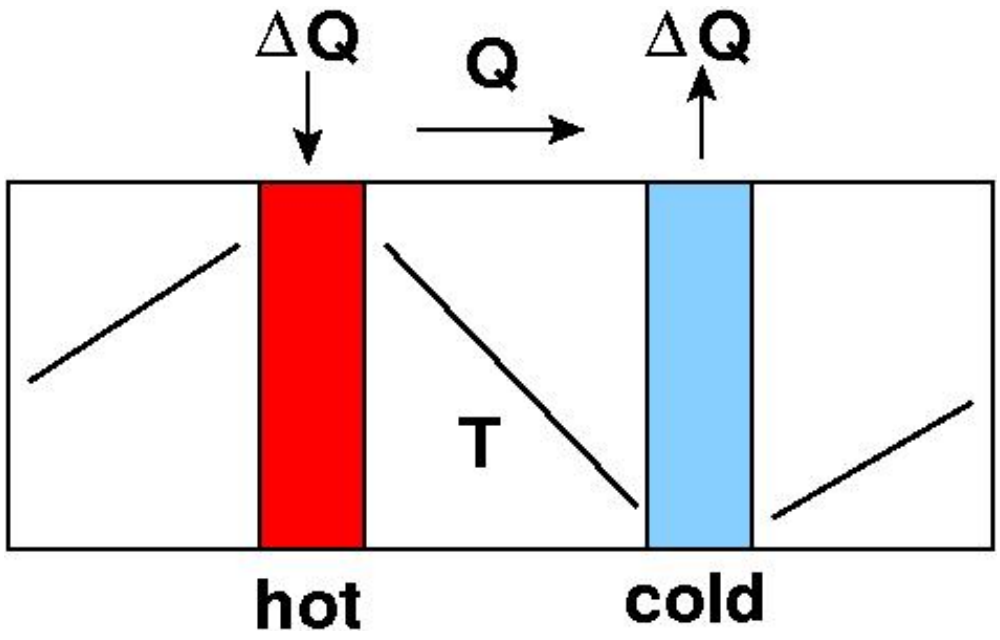
40000 1.4071151 -3.8068479 -1.6964391 14307.339 1.1772366

41000 1.4126121 -3.8153948 -1.6967416 15087.11 1.1408062

Loop time of 23.9599 on 8 procs for 20000 steps with 8000 atoms

(3) Variant of Muller-Plathe reverse perturbation method

- Define hot and cold regions of simulation box
- Add/subtract energy continuously to all atoms in these regions
- Equal and opposite heat flux
- Monitor the induced temperature profile



Variant of Muller-Plathe method

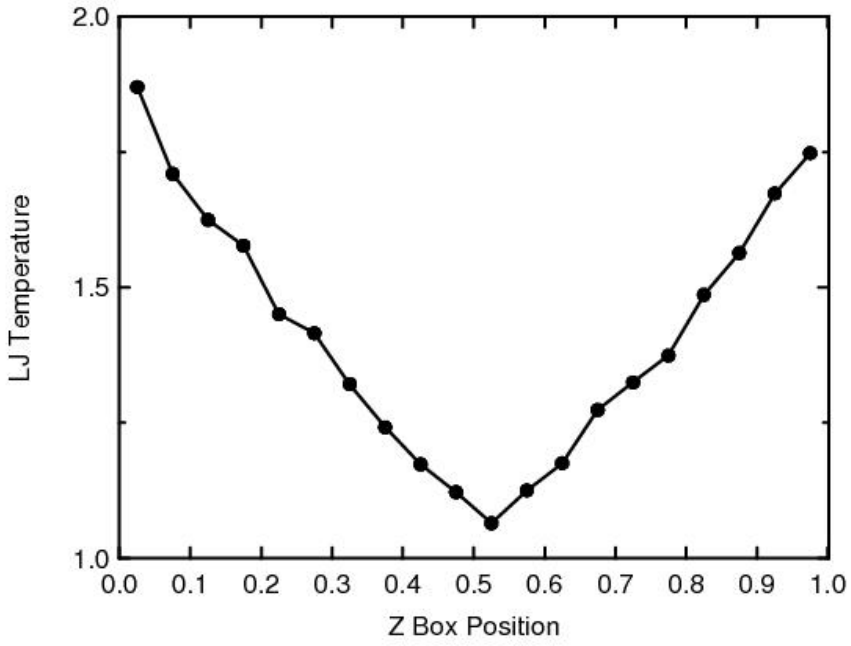
- LAMMPS implementation:
  - fix heat adds/subtracts KE in a region
  - fix ave/spatial monitors induced temperature gradient

$$\kappa = \frac{\Delta Q}{2 Area \Delta t} \frac{\Delta z}{\Delta T}$$

Script for variant of Muller-Plathe method

```
# thermal conductivity calculation

fix hot all heat 1 100.0 region hot
fix cold all heat 1 -100.0 region cold

compute ke all ke/atom
variable temp atom c_ke/1.5

fix 2 all ave/spatial 10 100 1000 z lower 0.05 &
  v_temp file tmp.heat.profile units reduced

variable tdiff equal f_2[11][3]-f_2[1][3]

run 20000
```

Output for variant of Muller-Plathe method



Step Temp Thot Tcold tdiff

...

30000 1.382101 1.9337034 1.0679145 -0.79821576

31000 1.3779178 1.8832819 1.0837774 -0.80611097

Loop time of 24.3193 on 8 procs for 20000 steps with 8000 atoms

(4) Green-Kubo equilibrium method

- Relate ensemble average of auto-correlation of J to κ
- Equilibrium J computable from per-atom KE, PE, virial

$$\kappa = \frac{V}{k_B T^2} \int_0^\infty \langle J_x(0) J_x(t) \rangle \, dt = \frac{V}{3 k_B T^2} \int_0^\infty \langle \mathbf{J}(0) \cdot \mathbf{J}(t) \rangle \, dt$$

$$\begin{aligned}
\mathbf{J} &= \frac{1}{V} \left[\sum_i e_i \mathbf{v}_i - \sum_i \mathbf{S}_i \mathbf{v}_i \right] \\
&= \frac{1}{V} \left[\sum_i e_i \mathbf{v}_i + \sum_{i<j} (\mathbf{f}_{ij} \cdot \mathbf{v}_j) \, \mathbf{x}_{ij} \right] \\
&= \frac{1}{V} \left[\sum_i e_i \mathbf{v}_i + \frac{1}{2} \sum_{i<j} (\mathbf{f}_{ij} \cdot (\mathbf{v}_i + \mathbf{v}_j)) \, \mathbf{x}_{ij} \right]
\end{aligned}$$

Green-Kubo method

- LAMMPS implementation:
  - compute heat/flux calculates J tensor
  - fix ave/correlate performs auto-correlation
  - variable trap() function performs time integration

Script for Green-Kubo method

```
compute myKE all ke/atom
compute myPE all pe/atom
compute myStress all stress/atom virial
compute flux all heat/flux myKE myPE myStress

fix JJ all ave/correlate $s $p $d &
  c_flux[1] c_flux[2] c_flux[3] type auto &
  file tmp.heatflux ave running

variable k11 equal trap(f_JJ[3])*${scale}
variable k22 equal trap(f_JJ[4])*${scale}
variable k33 equal trap(f_JJ[5])*${scale}

run 100000
```

Output for Green-Kubo method

Step Temp k11 k22 k33

...

98000 1.3477904 3.2534428 2.8638625 3.8437754

100000 1.3583776 3.3351133 2.859474 3.7715301

Loop time of 52.1737 on 8 procs for 100000 steps with 4000 atoms

```
variable kappa equal (v_k11+v_k22+v_k33)/3.0
print "thermal conductivity:  ${kappa}"
```

Comparing the 4 methods for thermal conductivity

Liquid Argon at state point: $\rho^* = 0.6$, $T^* = 1.35$, $R_c = 2.5\ \sigma$

D Evans, Phys Rev A, 34, 1449 (1986)

Method	κ
Direct thermostat	3.41
Muller-Plathe	3.45
M-P with fix heat	3.39
Green-Kubo	3.78
Evans paper	$\sim$3.3
Experiment	agrees with Evans

Comparing the 4 methods for thermal conductivity

Liquid Argon at state point: $\rho^* = 0.6$, $T^* = 1.35$, $R_c = 2.5\ \sigma$

D Evans, Phys Rev A, 34, 1449 (1986)

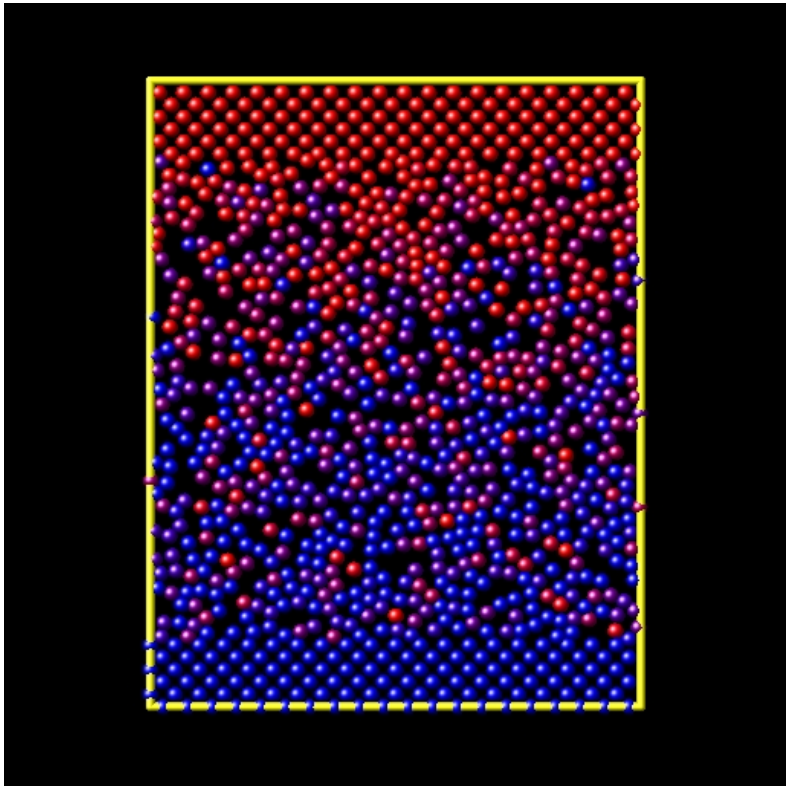
Method	κ
Direct thermostat	3.41
Muller-Plathe	3.45
M-P with fix heat	3.39
Green-Kubo	3.78
Evans paper	$\sim$3.3
Experiment	agrees with Evans

- Small systems have boundary effects
- Need to monitor equilibration and statistical noise
- Factors of 2 are easy to miss!

(1) Shearing via moving wall

LAMMPS methodology:

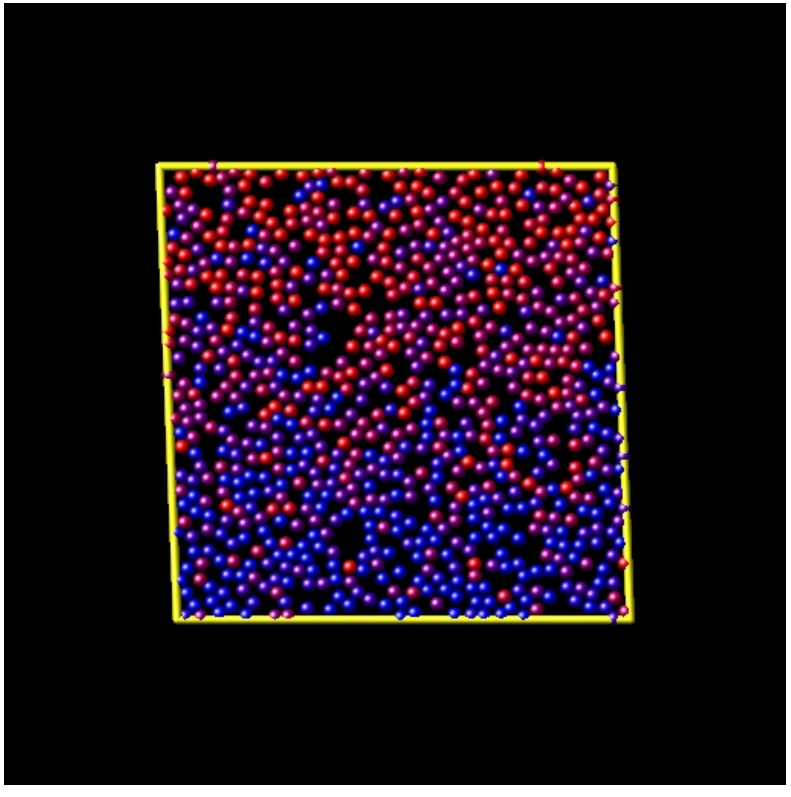
- Rigid, moving wall
- Fix addforce can apply load if desired
- Important to thermostat flow since adding energy
  - fix langevin on non-sheared dimensions
  - compute temp/profile to subtract flow profile
- Monitor P_{xz} and velocity profile of flow



(2) Shearing via deforming box

LAMMPS methodology:

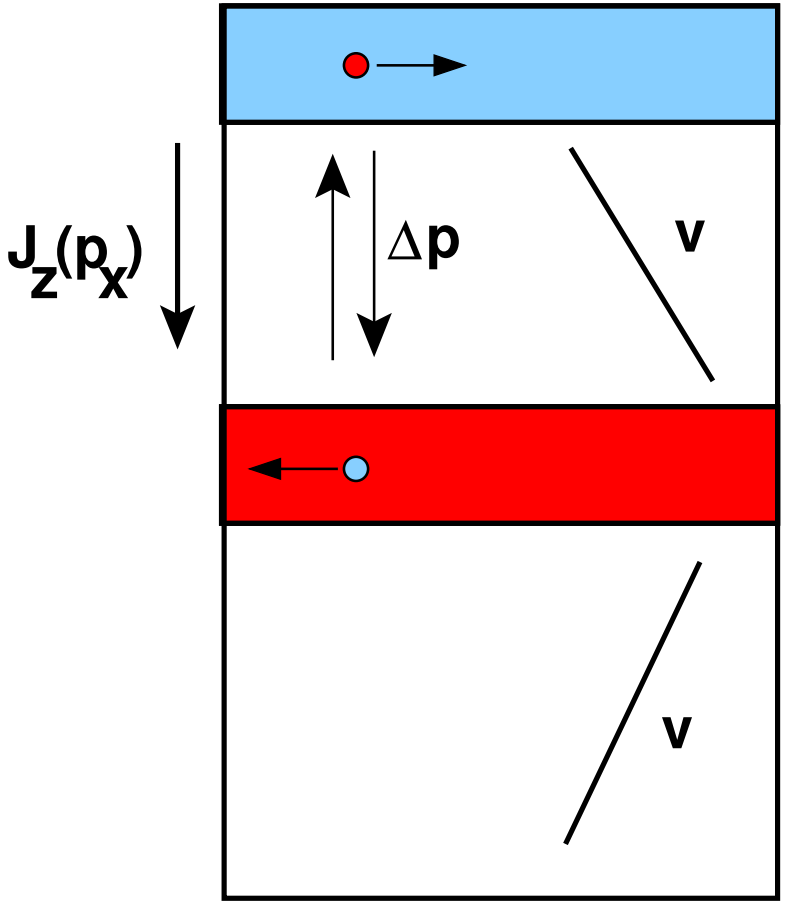
- Fix deform for box deformation
- Important to thermostat flow since adding energy
  - fix nvt/sllod for SLLOD equations of motion
  - *Evans and Morriss, Phys Rev A, 30, 1528 (1984)*
- Monitor P_{xz} and velocity profile of flow
  - insure flow profile agrees with box deformation



(3) Muller-Plathe reverse perturbation method

Muller-Plathe,
Phys Rev E, 59, 4894 (1999)

- Define two slabs within simulation box
- Find max V_x in one region, max $-V_x$ in other region
- Fix viscosity swaps momenta of these 2 atoms (or molecules)
- Tally momentum flux due to exchanges
- Monitor the induced velocity profile
- Reverse of previous methods



(4) Green-Kubo equilibrium method

- Relate ensemble average of auto-correlation of P_{xz} to η

$$\eta = \frac{V}{k_B T} \int_0^\infty \langle P_{xz}(0) P_{xz}(t) \rangle \, dt$$

- P_{xz} computable from virial
- Fix ave/correlate performs auto-correlation
- Variable trap() function performs time integration

Comparing the 4 methods for viscosity

LJ at state point: $\rho^* = 0.6$, $T^* = 1.0$, $R_c = 2.5\ \sigma$

Woodcock, AIChE Journal, 52, 438 (2006)

Method	η
Moving wall	0.946
Deforming box	1.18
Muller-Plathe	0.997
Green-Kubo	1.07
literature value	$\sim$1.0

Comparing the 4 methods for viscosity

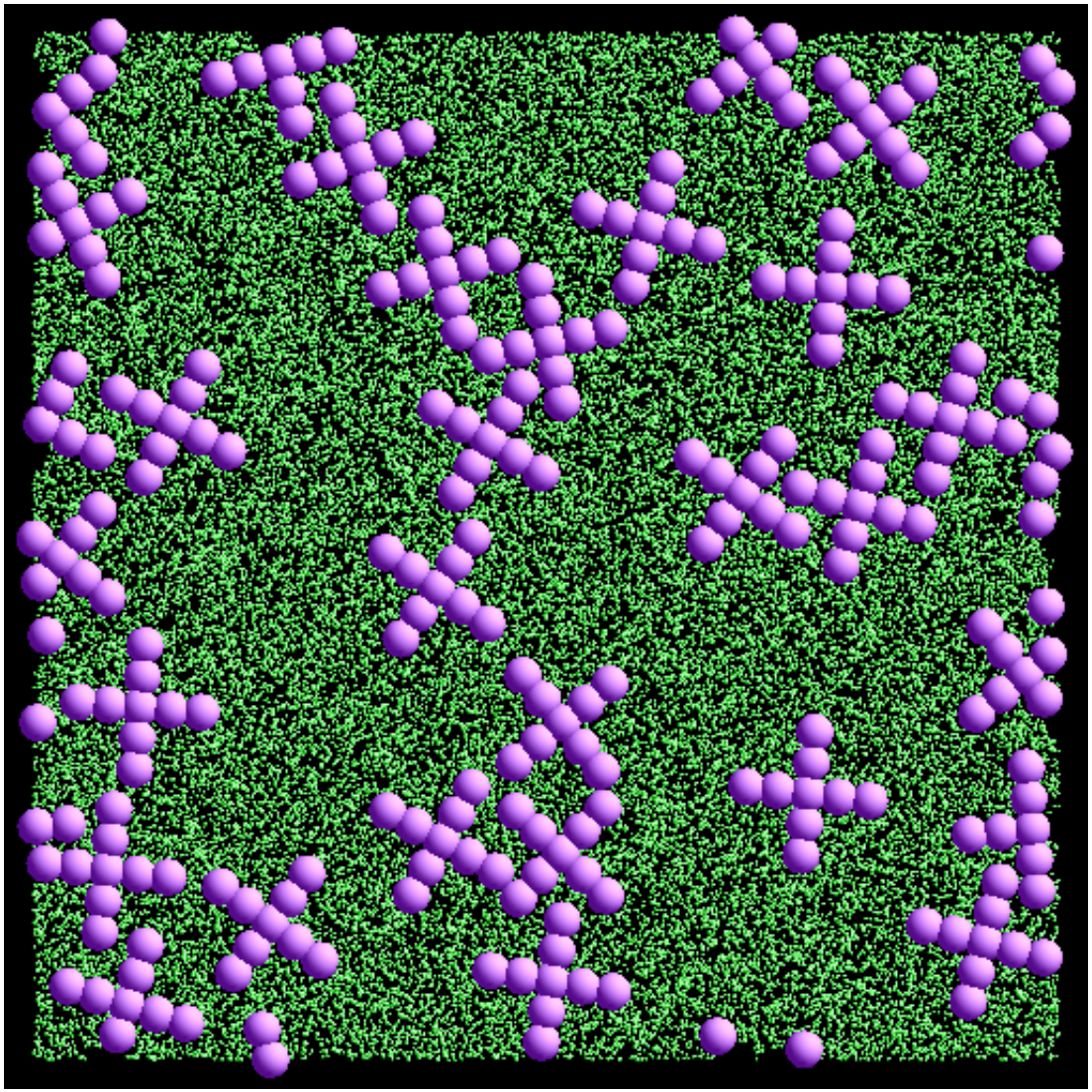
LJ at state point: $\rho^* = 0.6$, $T^* = 1.0$, $R_c = 2.5\ \sigma$

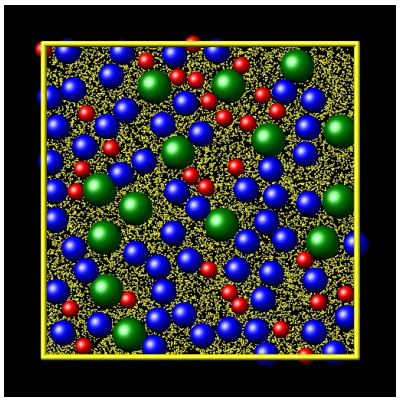
Woodcock, AIChE Journal, 52, 438 (2006)

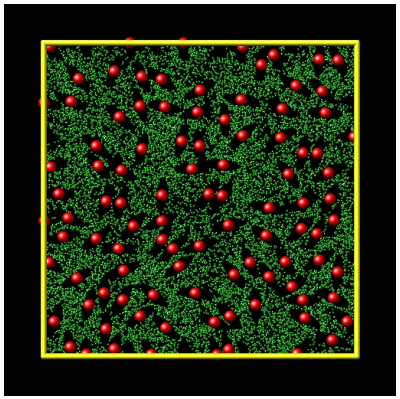
Method	η
Moving wall	0.946
Deforming box	1.18
Muller-Plathe	0.997
Green-Kubo	1.07
literature value	$\sim$1.0

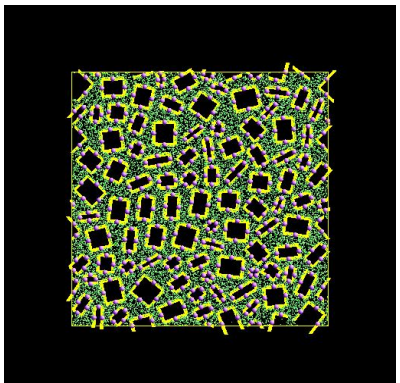
- Small systems have boundary effects
- Need to monitor equilibration and statistical noise
- Factors of 2 are easy to miss!

Shear viscosity for rigid-bodies in SRD fluid

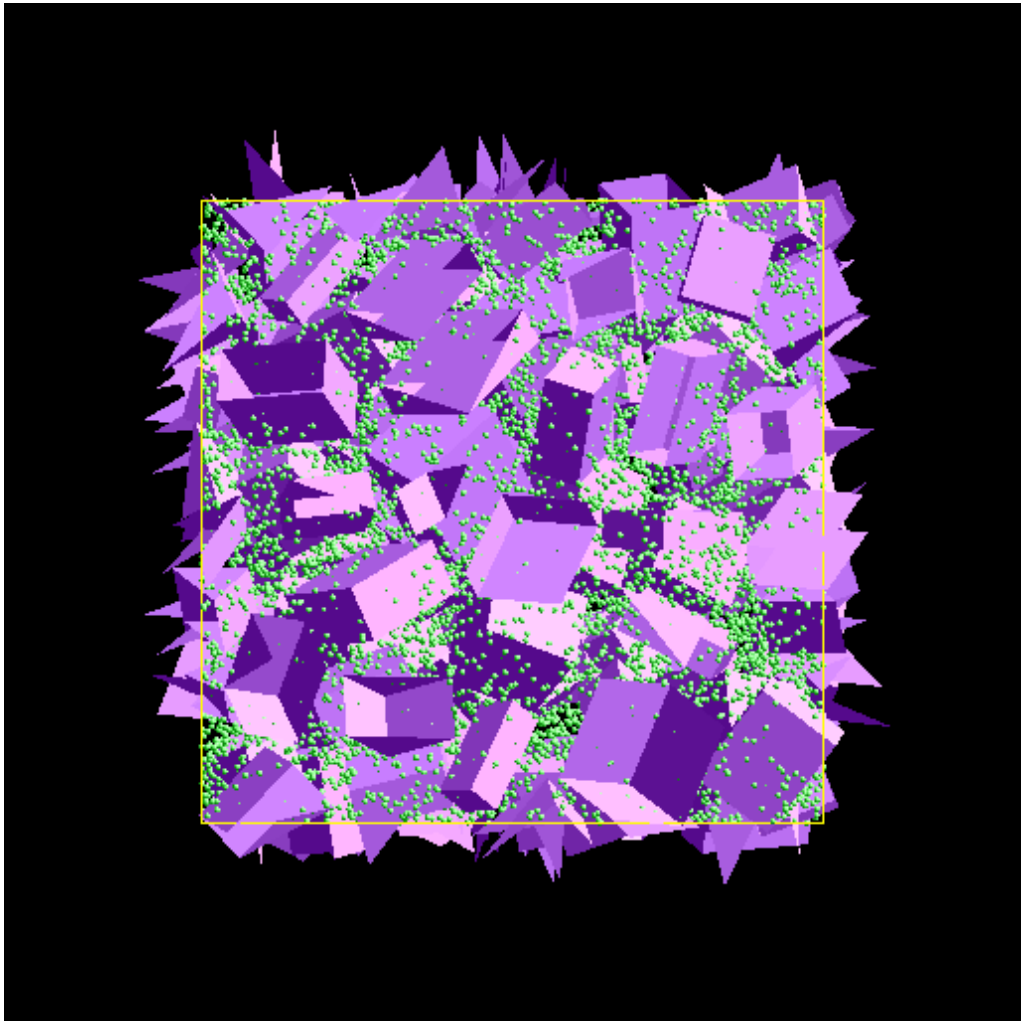




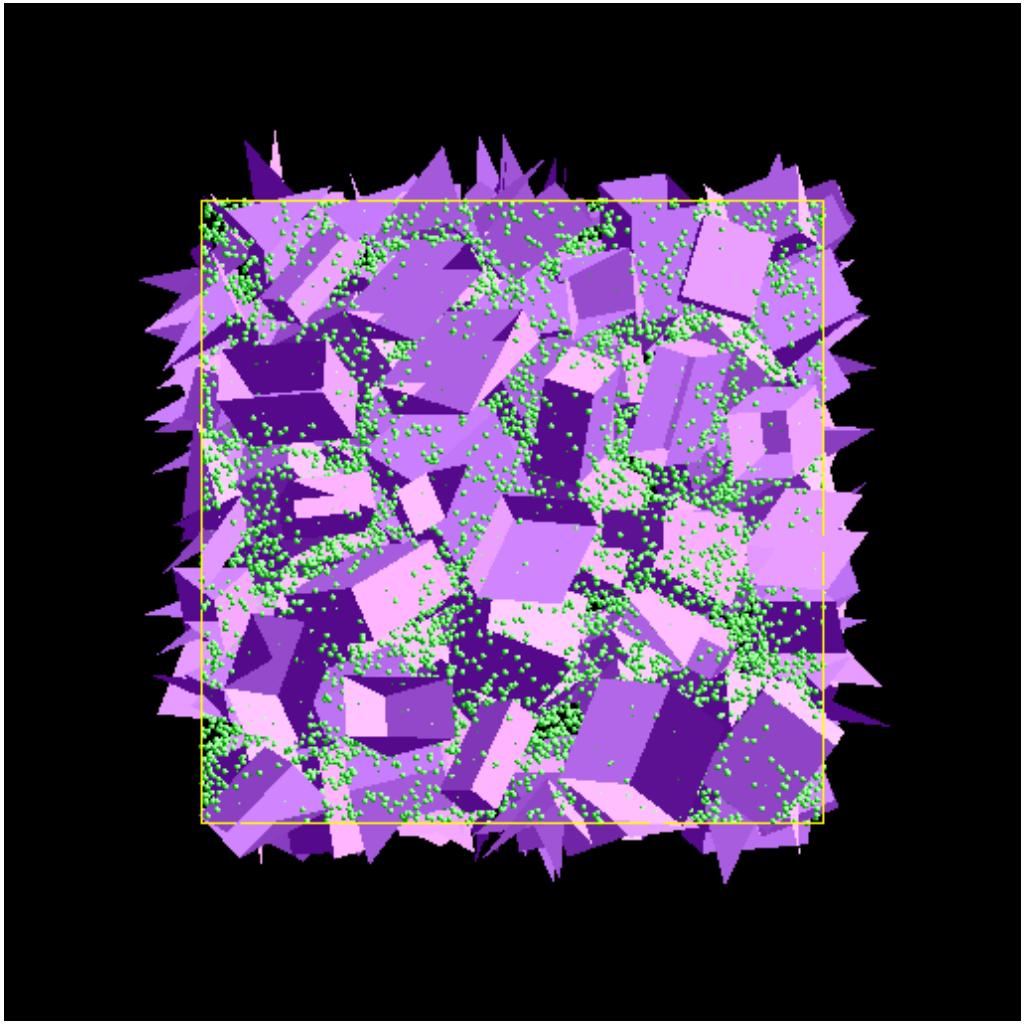




Shear viscosity for aspherical bodies in SRD fluid



Shear viscosity for aspherical bodies in SRD fluid



Any of these examples could use short-chain polymer solvents

Trade-offs between methods

- NEMD methods pros:
  - intuitive to understand
  - quick to converge
- NEMD methods cons:
  - unphysically large temperature gradients and heat fluxes
  - bigger systems to allow for gradient

Trade-offs between methods

- NEMD methods pros:
  - intuitive to understand
  - quick to converge
- NEMD methods cons:
  - unphysically large temperature gradients and heat fluxes
  - bigger systems to allow for gradient

- Green-Kubo method pros:
  - equilibrium simulation
  - can use smaller system
- Green-Kubo method cons:
  - slow to converge
  - hard to tell when correlation integral has converged

Hands-on exercise #1

- Focus on viscosity (or thermal conductivity) (or both!)
  - viscosity simulations are more visual to animate
- Study scripts in examples/VISCOSITY (or examples/KAPPA)
  - 4 scripts, for each of 4 methods
  - understand what each command and parameter represents
- Figure out how to analyze output to get η (or κ)

Hands-on exercise #1

- Focus on viscosity (or thermal conductivity) (or both!)
  - viscosity simulations are more visual to animate
- Study scripts in examples/VISCOSITY (or examples/KAPPA)
  - 4 scripts, for each of 4 methods
  - understand what each command and parameter represents
- Figure out how to analyze output to get η (or κ)

- Reproduce 4 values in examples/VISCOSITY/README
- Do scripts run faster in parallel?
- Do they produce the same answers in parallel?

Hands-on exercise #2

- Change parameters in input scripts:
  - size of system, density, temperature
  - shear rate, cutoff of potential
- IMPORTANT - When you change script and do a new run:
  - visualize to insure system dynamics are normal
  - monitor velocity (or temperature) profile
  - check convergence of G-K integrations
  - are you running long enough?
- Otherwise your η or κ values may be bogus

Hands-on exercise #2

- Change parameters in input scripts:
  - size of system, density, temperature
  - shear rate, cutoff of potential
- IMPORTANT - When you change script and do a new run:
  - visualize to insure system dynamics are normal
  - monitor velocity (or temperature) profile
  - check convergence of G-K integrations
  - are you running long enough?
- Otherwise your η or κ values may be bogus
- Choose one larger/smaller value of a parameter
  - how much larger or smaller?
- Does η (or κ) change with that parameter?
- Do all methods still agree?
- Does variation make physical sense?

Hands-on exercise #3

- Make a plot as vary a parameter over a wide range
  - size of system, density, temperature
  - shear rate, cutoff of potential
- E.g. η versus shear-rate for shear-thinning
- What other parameters should remain constant?
  - e.g. temperature, pressure
- How much can parameter vary before dynamics break down?
  - e.g. liquid crystallizes at too high a density

Hands-on exercise #3

- Make a plot as vary a parameter over a wide range
  - size of system, density, temperature
  - shear rate, cutoff of potential
- E.g. η versus shear-rate for shear-thinning
- What other parameters should remain constant?
  - e.g. temperature, pressure
- How much can parameter vary before dynamics break down?
  - e.g. liquid crystallizes at too high a density

- Bonus: modify script to run series of simulations as parameter varies
  - see Section_howto.html 6.4 and variable command
- Bonus: run/viz M-P viscosity scripts in examples/ASPHERICAL