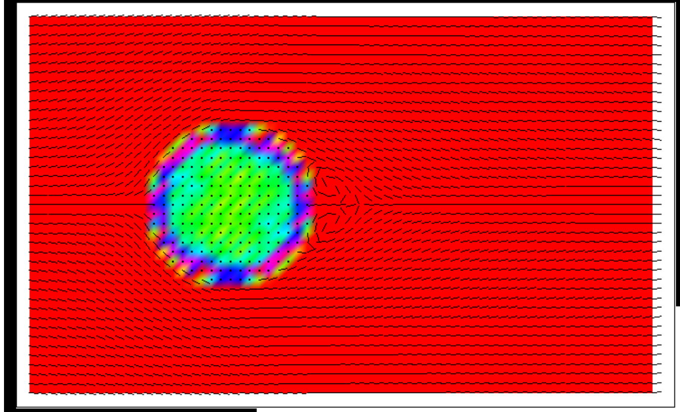
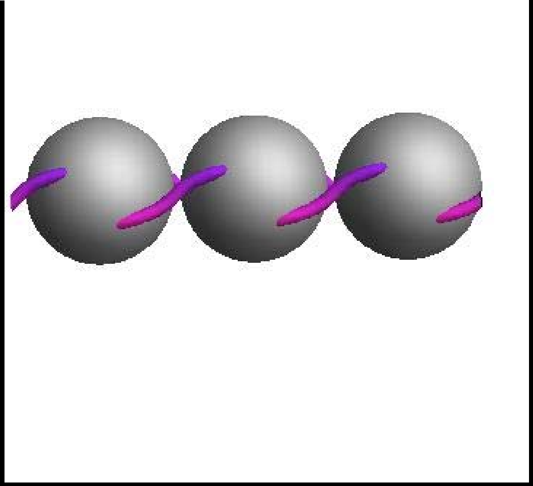
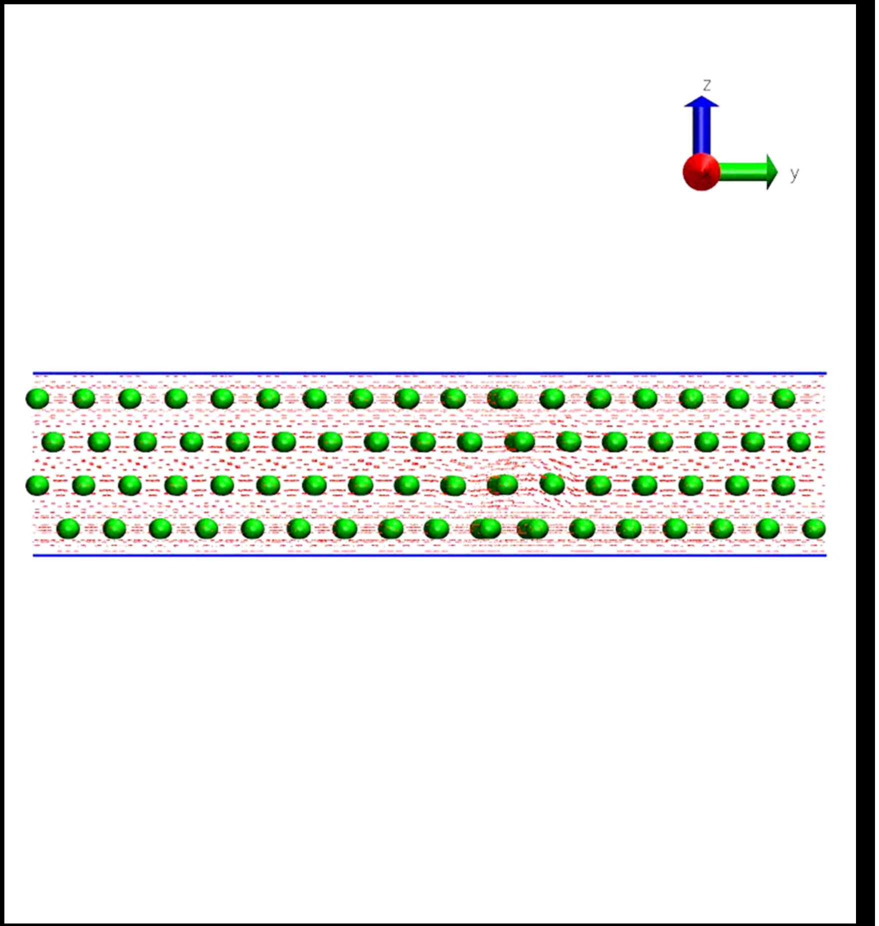
HYDRODYNAMIC FORCES IMPLEMENTED INTO LAMMPS THROUGH A LATTICE-BOLTZMANN FLUID







Colin Denniston,

Frances Mackay, Santtu Ollila, Tyson Whitehead

University of Western Ontario

Continuum fluid with thermal noise

- Navier-Stokes equations with thermal noise (Landau & Lifshitz):

$$(\partial_t + \partial_\alpha u_\alpha)\rho = 0$$

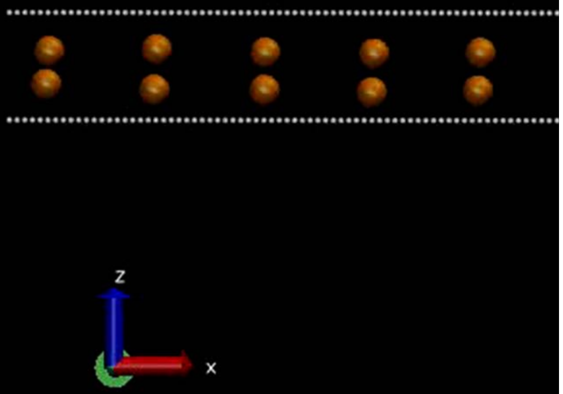
$$\rho(\partial_t + u_\alpha \partial_\alpha)u_\beta = -\partial_\alpha (P_{\beta\alpha} + s_{\beta\alpha}) + \partial_\alpha\left(\eta_{\alpha\beta\gamma\nu}\partial_\gamma u_\nu\right) + F_\beta$$

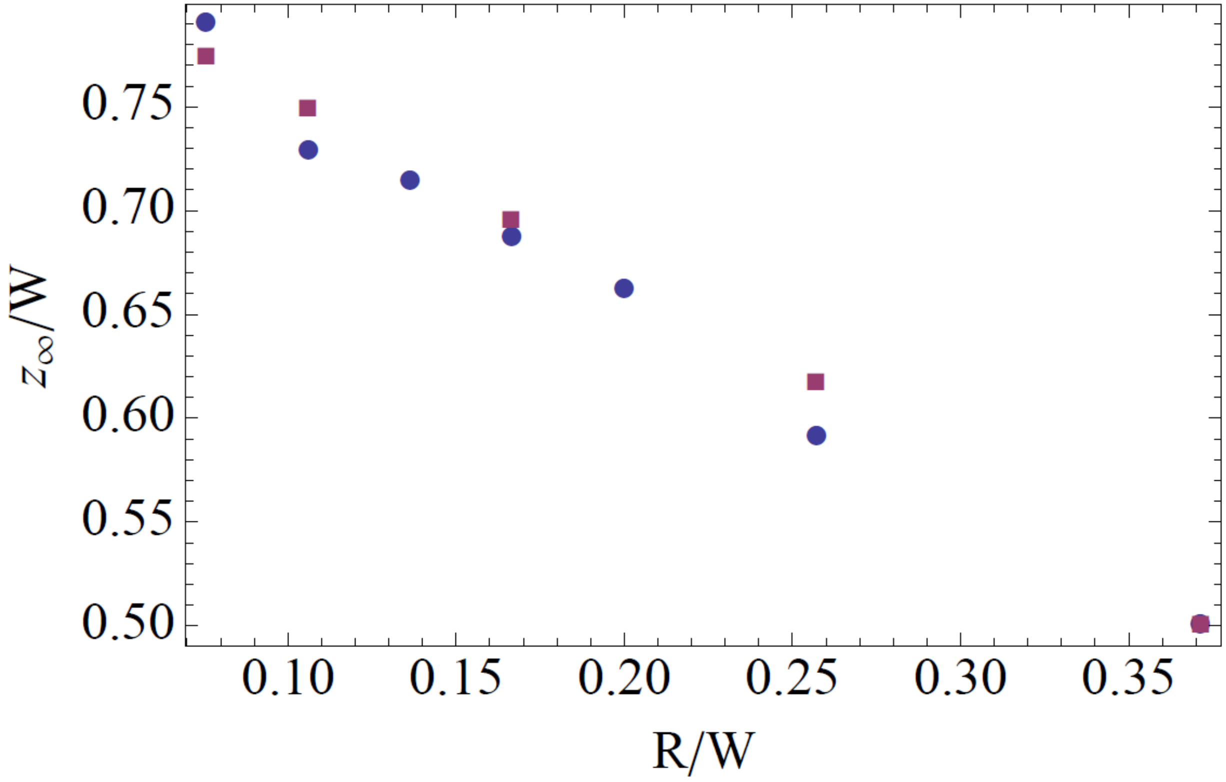
- Note that the thermal noise appears in the stress tensor so will conserve mass and momentum. It should also obey the fluctuation-dissipation theorem (Landau & Lifshitz):

$$\left\langle s_{\alpha\beta}(\mathbf{r},t)\, s_{\gamma\nu}(\mathbf{r}',t')\right\rangle = 2k_B T \eta_{\alpha\beta\gamma\nu}\,\delta(\mathbf{r}-\mathbf{r}')\,\delta(t-t')$$

Noise in LB ref: S.T.T.Ollila, CD, M. Karttunen, T. Ala-Nisslia, J.Chem. Phys. **134**, 064902 (2011).

Does inertia matter for small Re?

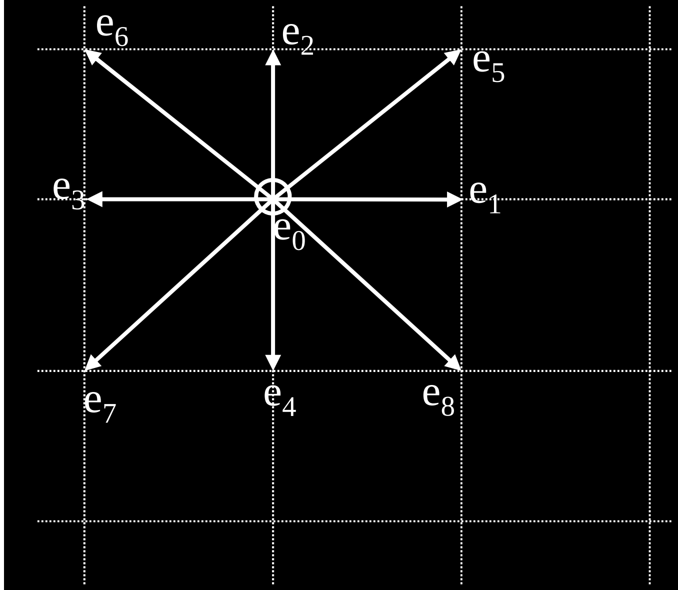




$$Re = \frac{v\,L}{\eta}$$

Point particle result:
Segre & Siilberberg (1961),
Ho & Leal (1974)

Simple Lattice Boltzmann Algorithm



- f_i = partial densities (9 in 2d, 15 in 3d)
- $\{ f_\mathrm{i} \}_\mathrm{i}$ = a discrete probability distribution
- Moments of these distributions ***are*** the physical variables of interest:

$$\rho \equiv \sum_i f_i, \quad \rho\mathbf{u} \equiv \sum_i f_i \mathbf{e}_i,$$

- They evolve via the equation:

$$(\partial_t + \mathbf{e}_{i\alpha}\partial_\alpha) f_i = -\lambda_{ij}(f_j(\mathbf{x},t) - f_j^{eq}(\mathbf{x},t,\{f_k\}))$$

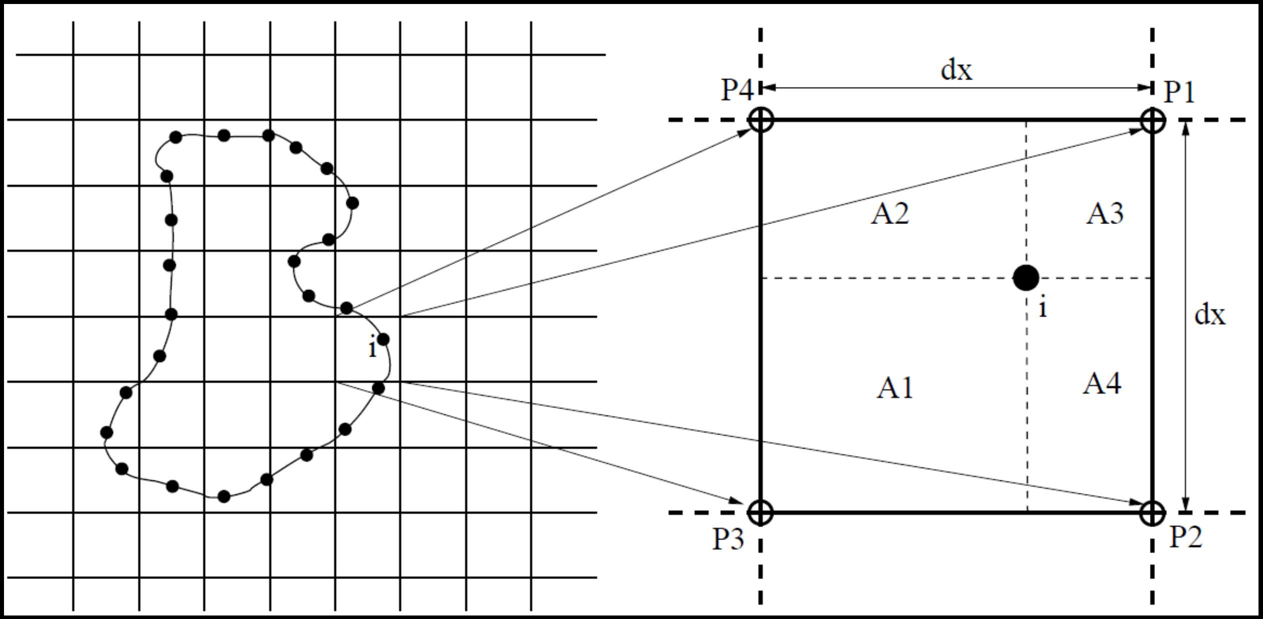
where

$$f_i^{eq} = A_i + B_i \mathbf{u}\cdot\mathbf{e}_i + C_i \mathbf{u}\cdot\mathbf{u} + D_i (\mathbf{u}\cdot\mathbf{e}_i)^2,$$

and A_i, B_i, C_i, and D_i are chosen so that

$$\sum_i f_i^{eq} = \rho, \quad \sum_i f_i^{eq}\mathbf{e}_i = \rho\mathbf{u}, \quad \sum_i f_i^{eq} \mathrm{e}_{i\alpha}\mathrm{e}_{i\beta} = -p\delta_{\alpha\beta} + \rho u_\alpha u_\beta, \cdots$$

- Each node represents a fixed area ΔA_i
- Nodes are distributed onto the lattice



weights proportional to the opposite enclosed area within the cell.

Eg. ξ_{i1}=A1/dx^2

Easily generalized to 3-D (use volume instead of area).

Peskin's Immersed boundary method is similar. With compact support spreading 2 lattice sites from nodes lattice effects can be almost eliminated.

- First done for non-point objects in LB by Duenweg & Lobaskin, NJP (2004).

Modelling:

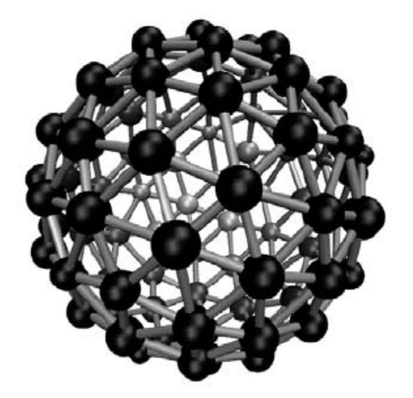
- Particles live off-lattice and evolve using molecular dynamics (written as a package for LAMMPS).
- Particles are mapped to the mesh using NDA algorithm and hydrodynamic forces on each particle computed from:

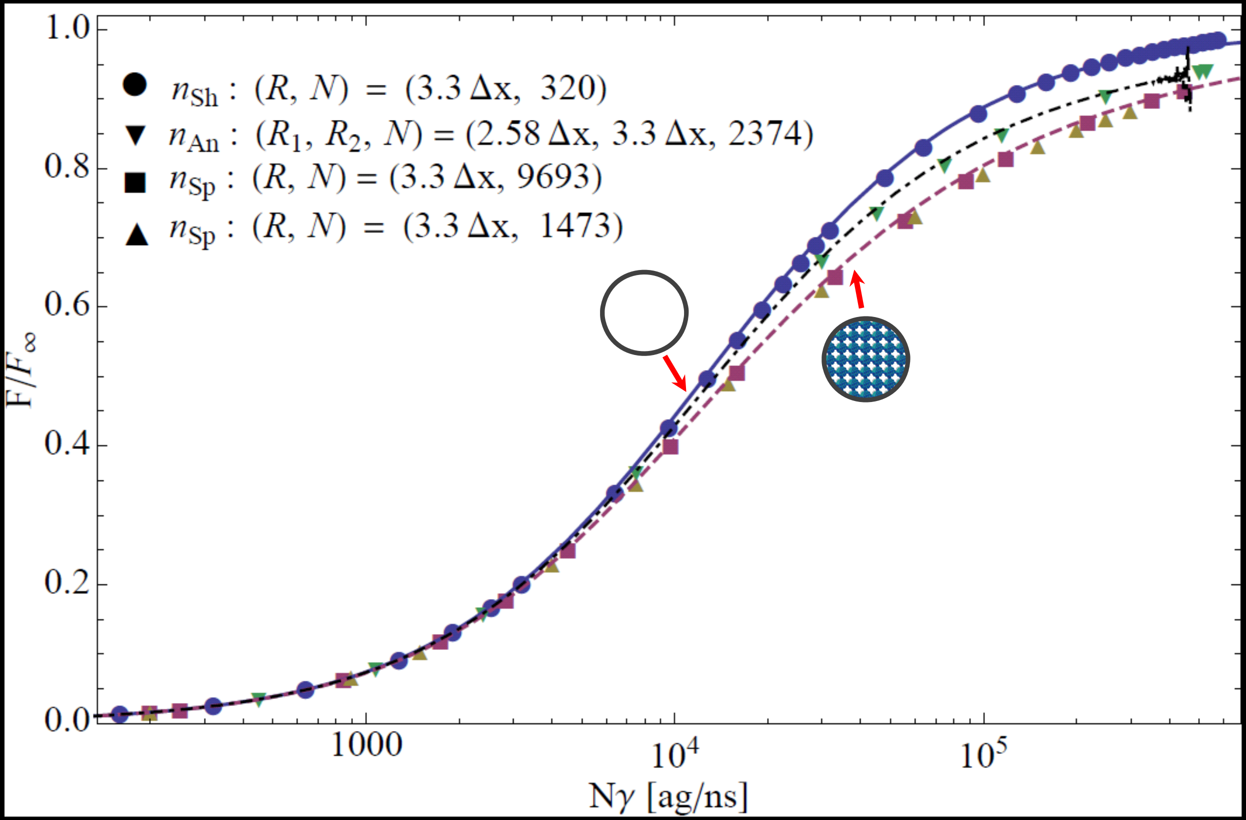
$$\mathbf{F}_{ij} = (\mathbf{v}_i - \hat{\mathbf{u}}_i)\xi_{ij}\gamma$$

$$\mathbf{F}_i = \sum_{j=1}^{n} \mathbf{F}_{ij} = (\mathbf{v}_i - \hat{\mathbf{u}}_i^{(I)})\gamma$$

- γ is “drag” coefficient (to be determined), v_p is the particle velocity, and u_i is the interpolated fluid velocity at node *i*. The resulting torque is also computed for rotational motion.
- The fluid experience an equal and opposite force distributed back onto lattice in a matched way.

Drag Force

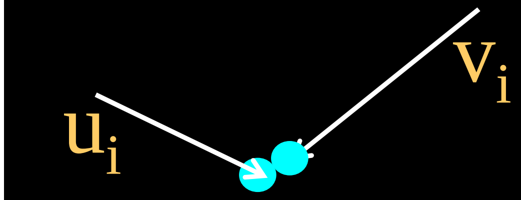



1.0
0.8
0.6
0.4
0.2
0.0
● n_{Sh} : $(R, N) = (3.3\,\Delta x,\ 320)$
▼ n_{An} : $(R_1, R_2, N) = (2.58\,\Delta x,\ 3.3\,\Delta x,\ 2374)$
■ n_{Sp} : $(R, N) = (3.3\,\Delta x,\ 9693)$
▲ n_{Sp} : $(R, N) = (3.3\,\Delta x,\ 1473)$
F/F_∞
1000
10^4
10^5
$\mathrm{N}\gamma$ [ag/ns]

Brinkman Theory: Felderhoff et al., Bhatt & Sacheti,
S.T.T. Ollila, T. Ala-Nissila, CD, JFM 709, 129 (2012).

In discrete time this can be conservative!

Consider the collision of two point particles



If the collision conserves momentum and kinetic energy then:

$$\mathbf{u}_f = \mathbf{u}_i + (\mathbf{v}_i - \mathbf{u}_i)\frac{2m_v}{m_v + m_u}$$

$$\mathbf{v}_f = \mathbf{v}_i + (\mathbf{v}_i - \mathbf{u}_i)\frac{-2m_u}{m_v + m_u}.$$

$$\mathbf{F}_{node} = \frac{\Delta p_{node}}{\Delta t_{collision}} = \frac{m_v(\mathbf{v}_f - \mathbf{v}_i)}{\Delta t_{collision}}$$

$$\mathbf{F}_{fluid} = \frac{\Delta p_{fluid}}{\Delta t_{collision}} = \frac{m_u(\mathbf{u}_f - \mathbf{u}_i)}{\Delta t_{collision}}.$$

i.e. something that looks like a discrete time drag force with:

$$\gamma = \frac{2m_u m_v}{m_u + m_v}\left(\frac{1}{\Delta t_{collision}}\right)$$

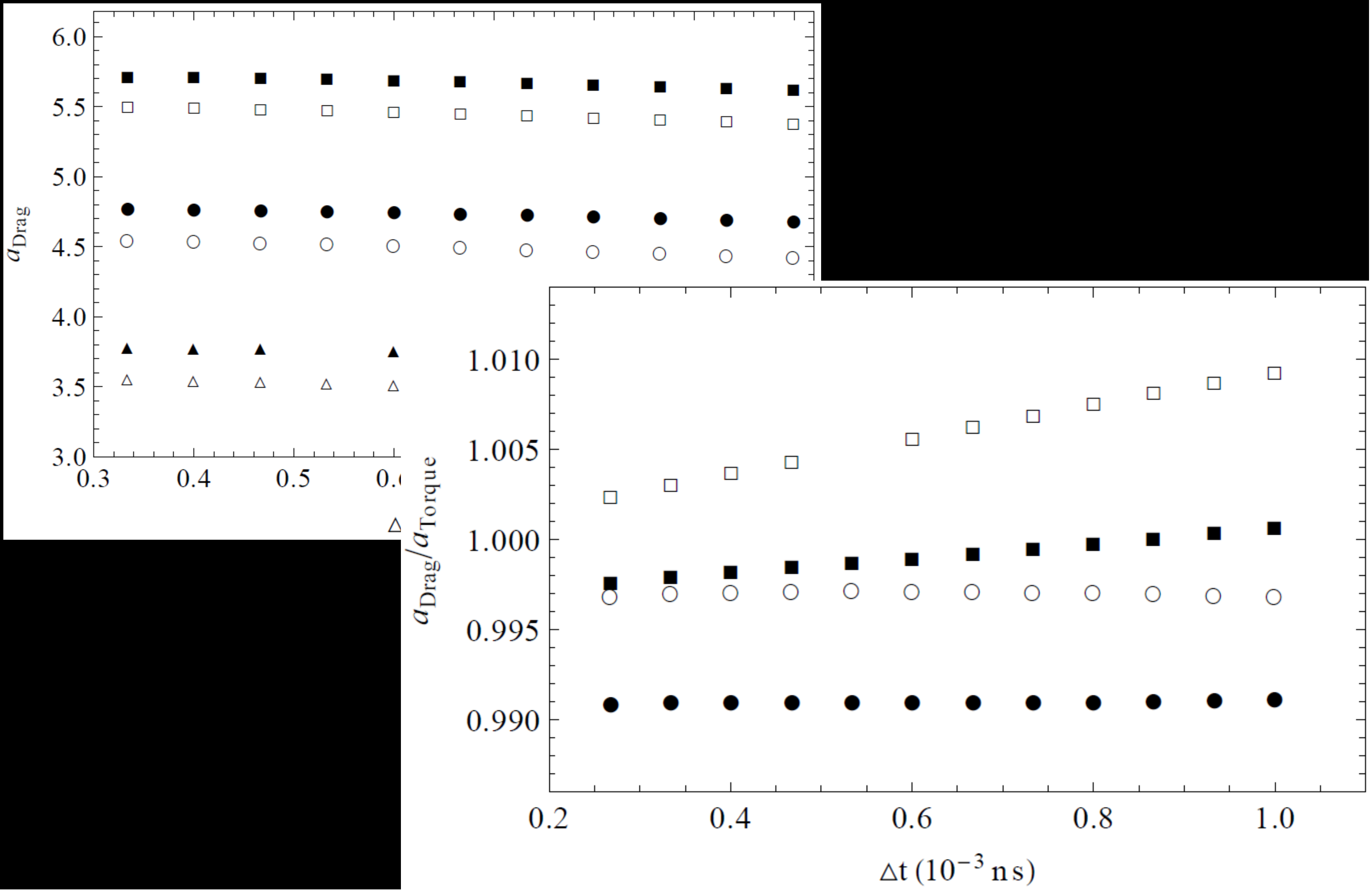
m_v=node mass

m_u=fluid mass interacting with node via interpolation stencil

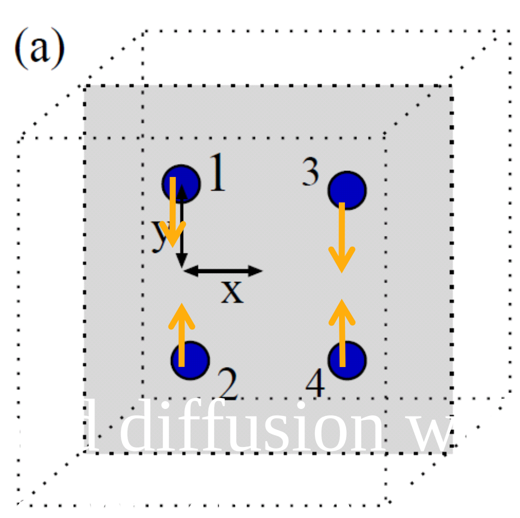
We also take: $\tau/\Delta t_{collision} = 1$

Ref: F.E. Mackay, CD, J. Comp. Phys. **237**, 289 (2013).

Hydrodynamics Radius:



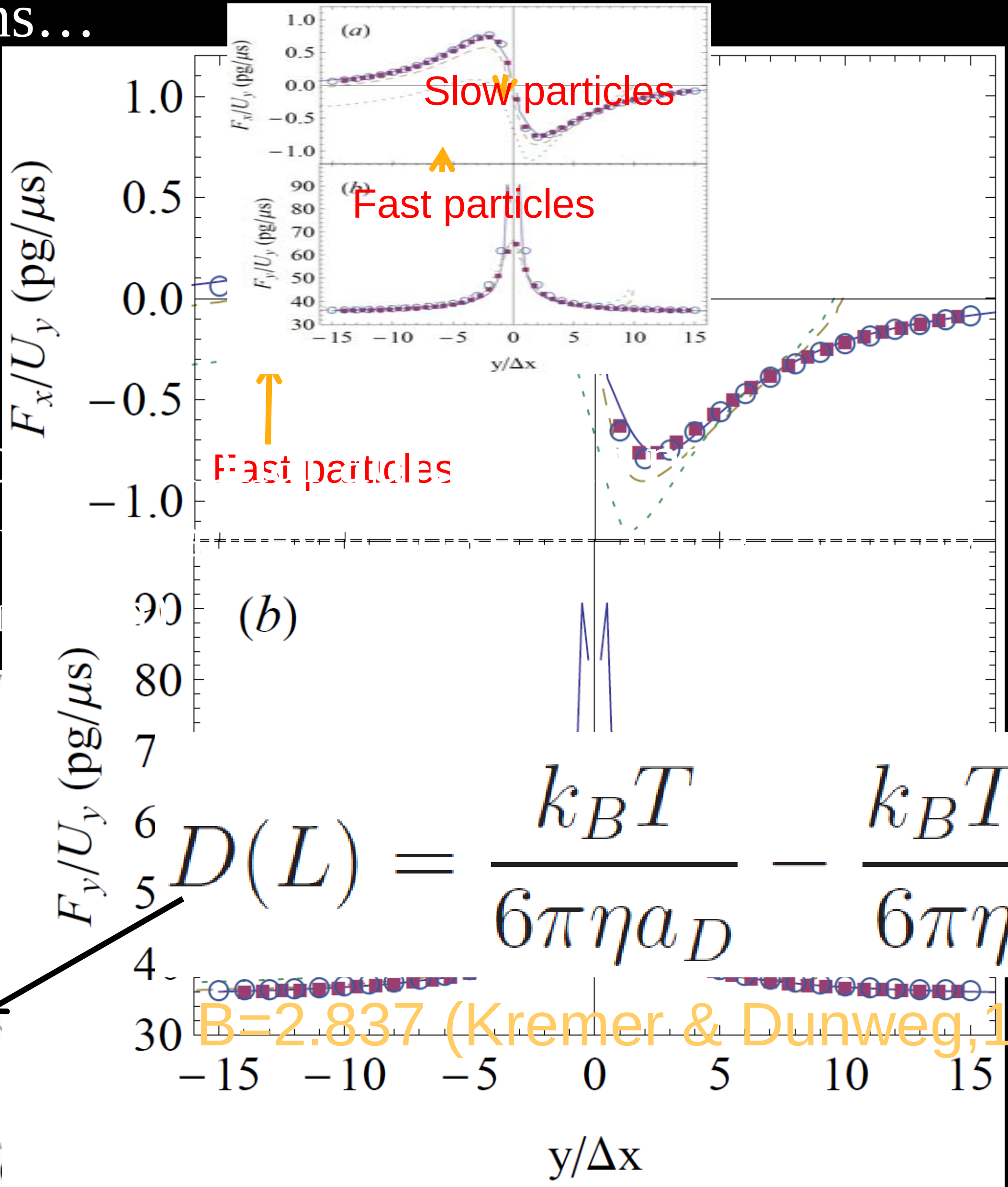
Picking γ in this way also give consistent particle-particle hydrodynamic interactions…



and diffusion with nc

(other schemes have req

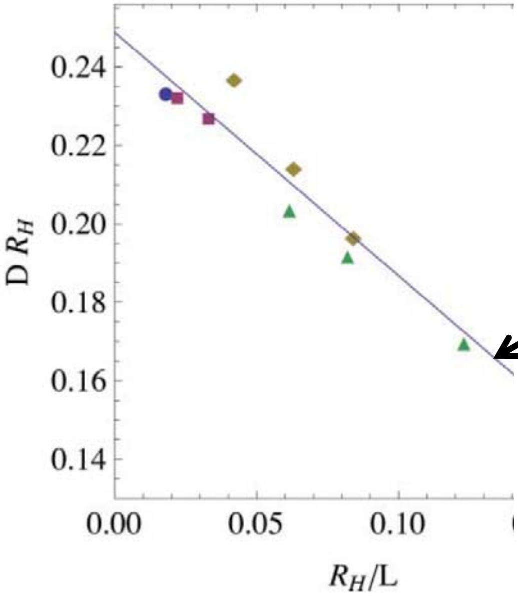
to get correct diffusive m



Slow particles

Fast particles

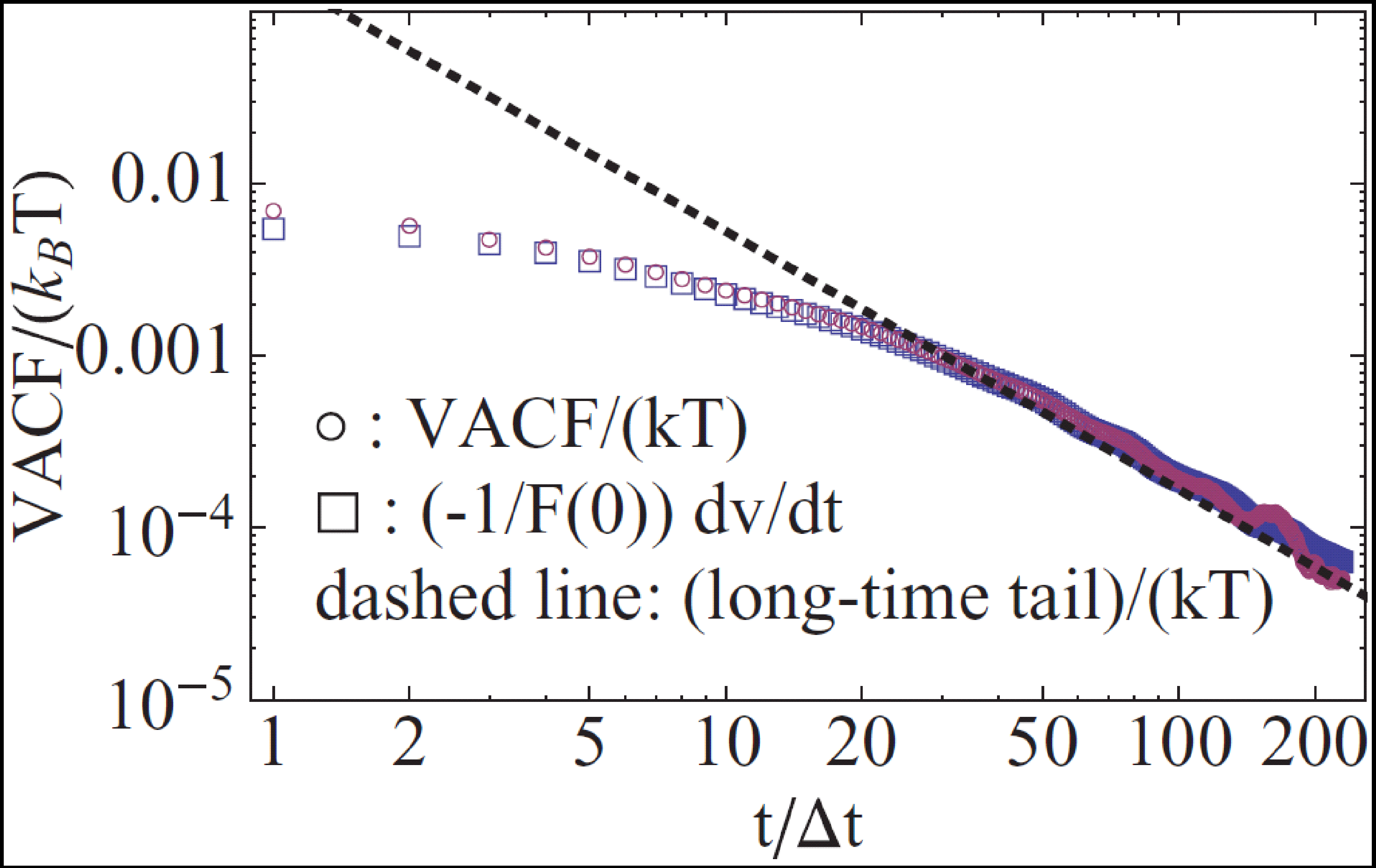
Fast particles



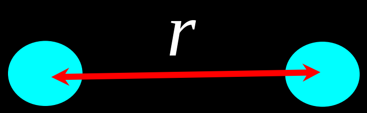
$$D(L) = \frac{k_BT}{6\pi\eta a_D} - \frac{k_BT}{6\pi\eta}\frac{B}{L}$$

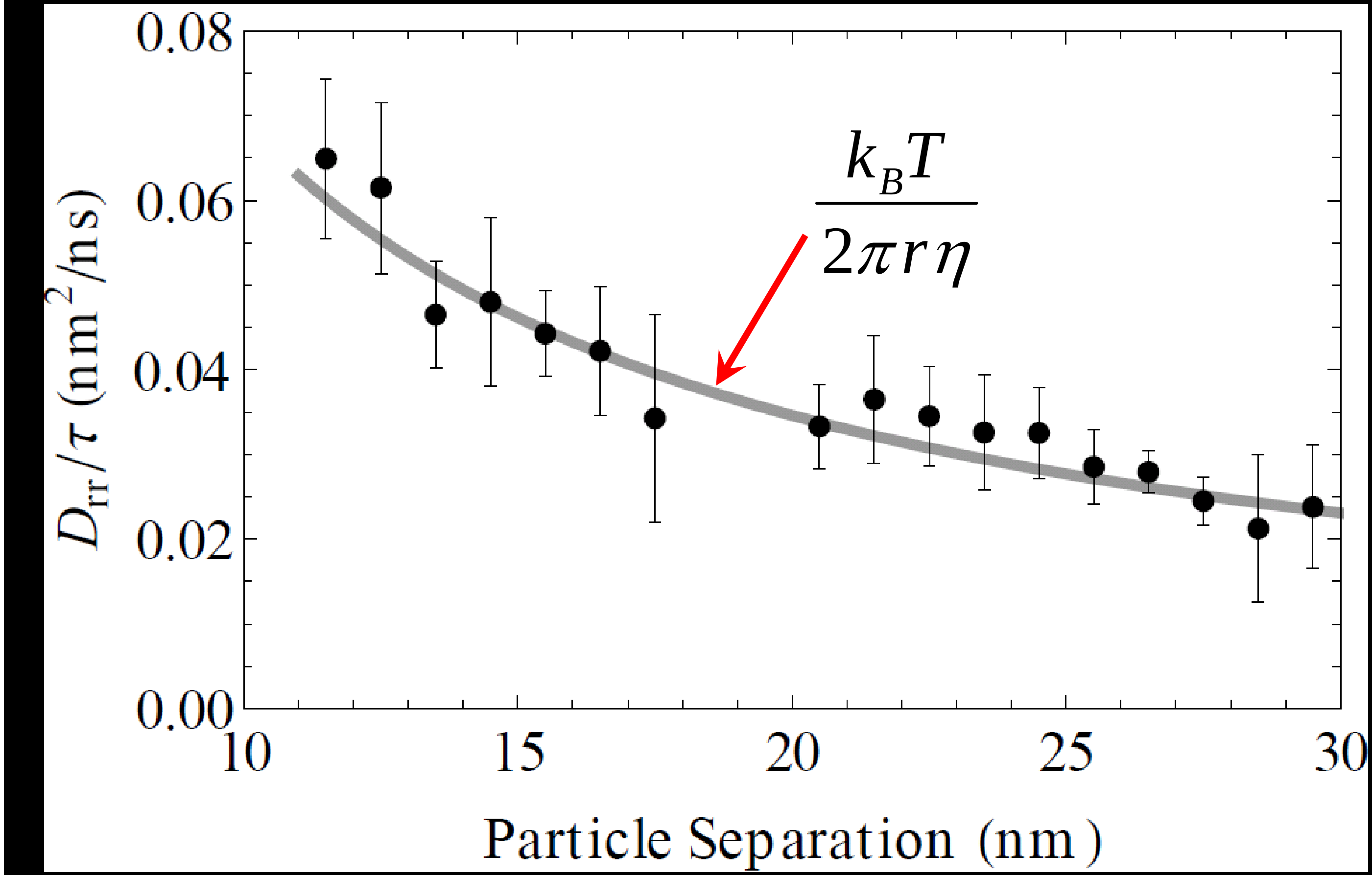
B=2.837 (Kremer & Dunweg,1993)

Velocity auto-correlation function



Two-particle diffusion:





Theory curve: Crocker et al.,PRL **85**, 888 (2000)

Implementation in LAMMPS:

- MPI domain decomposition taken same as particles in LAMMPS
- Implemented as fixes:

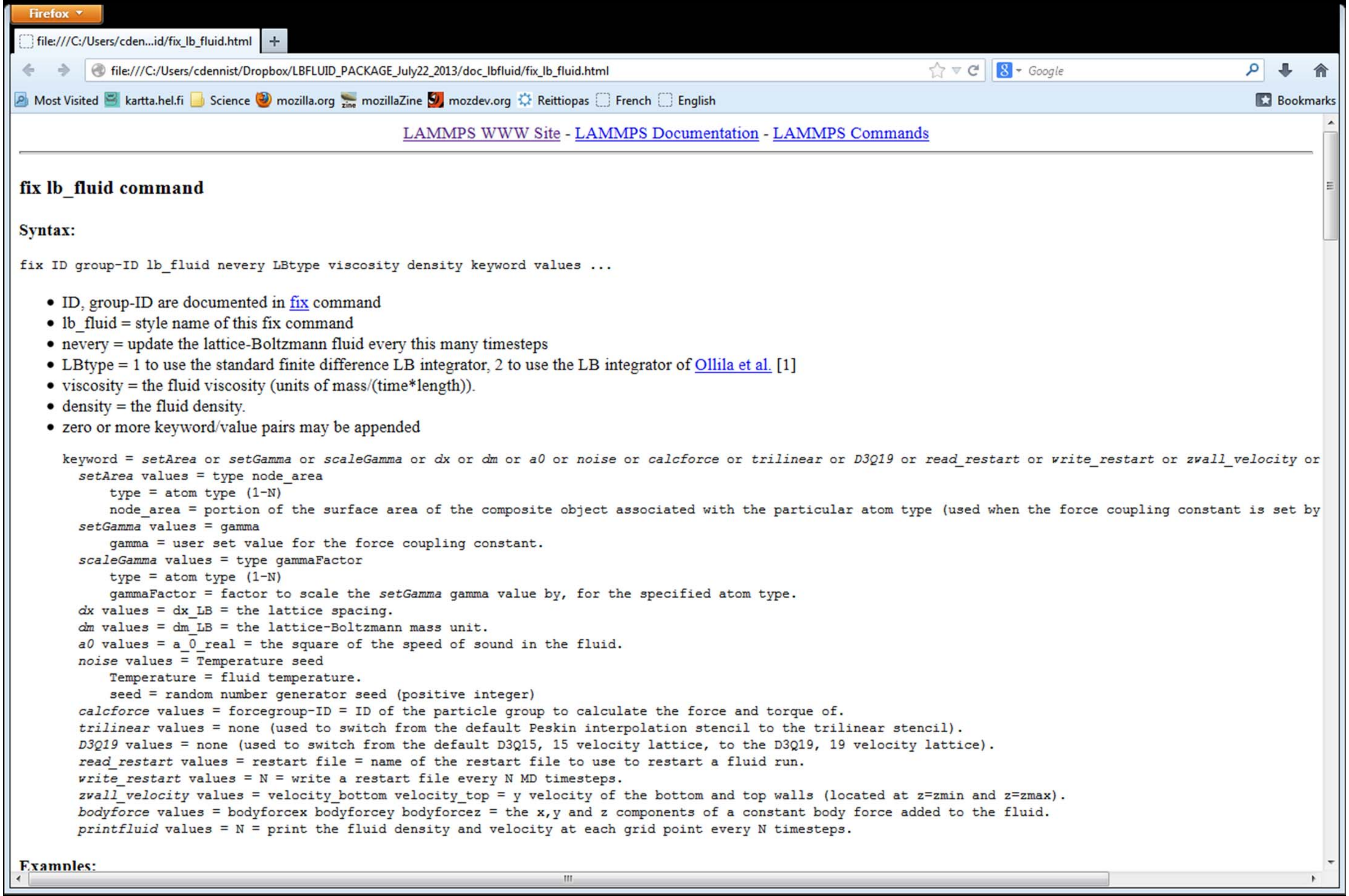
  fix lb_fluid – applies force of particle on fluid

  fix viscous_lb – applies force of fluid on particles

  along with fix_momentum_lb
- Hook into update.cpp : units

  Otherwise just a regular user package

Ref: J. Comp. Phys. **237**, 289 (2013)


Firefox
file:///C:/Users/cden...id/fix_lb_fluid.html
+
file:///C:/Users/cdennist/Dropbox/LBFLUID_PACKAGE_July22_2013/doc_lbfluid/fix_lb_fluid.html
Google
Most Visited
kartta.hel.fi
Science
mozilla.org
mozillaZine
mozdev.org
Reittiopas
French
English
Bookmarks

LAMMPS WWW Site - LAMMPS Documentation - LAMMPS Commands

fix lb_fluid command

Syntax:

```
fix ID group-ID lb_fluid nevery LBtype viscosity density keyword values ...
```

- ID, group-ID are documented in fix command
- lb_fluid = style name of this fix command
- nevery = update the lattice-Boltzmann fluid every this many timesteps
- LBtype = 1 to use the standard finite difference LB integrator, 2 to use the LB integrator of Ollila et al. [1]
- viscosity = the fluid viscosity (units of mass/(time*length)).
- density = the fluid density.
- zero or more keyword/value pairs may be appended

```
keyword = setArea or setGamma or scaleGamma or dx or dm or a0 or noise or calcforce or trilinear or D3Q19 or read_restart or write_restart or zwall_velocity or
  setArea values = type node_area
      type = atom type (1-N)
      node_area = portion of the surface area of the composite object associated with the particular atom type (used when the force coupling constant is set by
  setGamma values = gamma
      gamma = user set value for the force coupling constant.
  scaleGamma values = type gammaFactor
      type = atom type (1-N)
      gammaFactor = factor to scale the setGamma gamma value by, for the specified atom type.
  dx values = dx_LB = the lattice spacing.
  dm values = dm_LB = the lattice-Boltzmann mass unit.
  a0 values = a_0_real = the square of the speed of sound in the fluid.
  noise values = Temperature seed
      Temperature = fluid temperature.
      seed = random number generator seed (positive integer)
  calcforce values = forcegroup-ID = ID of the particle group to calculate the force and torque of.
  trilinear values = none (used to switch from the default Peskin interpolation stencil to the trilinear stencil).
  D3Q19 values = none (used to switch from the default D3Q15, 15 velocity lattice, to the D3Q19, 19 velocity lattice).
  read_restart values = restart file = name of the restart file to use to restart a fluid run.
  write_restart values = N = write a restart file every N MD timesteps.
  zwall_velocity values = velocity_bottom velocity_top = y velocity of the bottom and top walls (located at z=zmin and z=zmax).
  bodyforce values = bodyforcex bodyforcey bodyforcez = the x,y and z components of a constant body force added to the fluid.
  printfluid values = N = print the fluid density and velocity at each grid point every N timesteps.
```

Examples:

Sheared Colloids

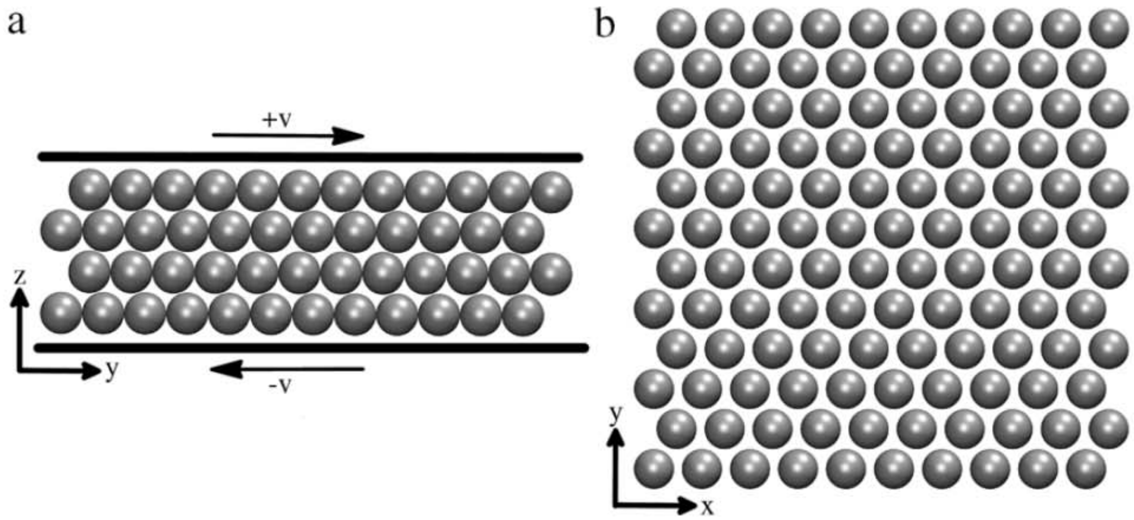


Fig. 8. Geometry used for the confined colloid simulation. Left: view of the four layers in the z–y plane. Right: view of a single layer in the x–y plane. For simplicity, the spheres shown simply indicate the location of the central atoms for each particle. (In the simulation, each colloidal particle is composed of 3613 MD particle nodes.)

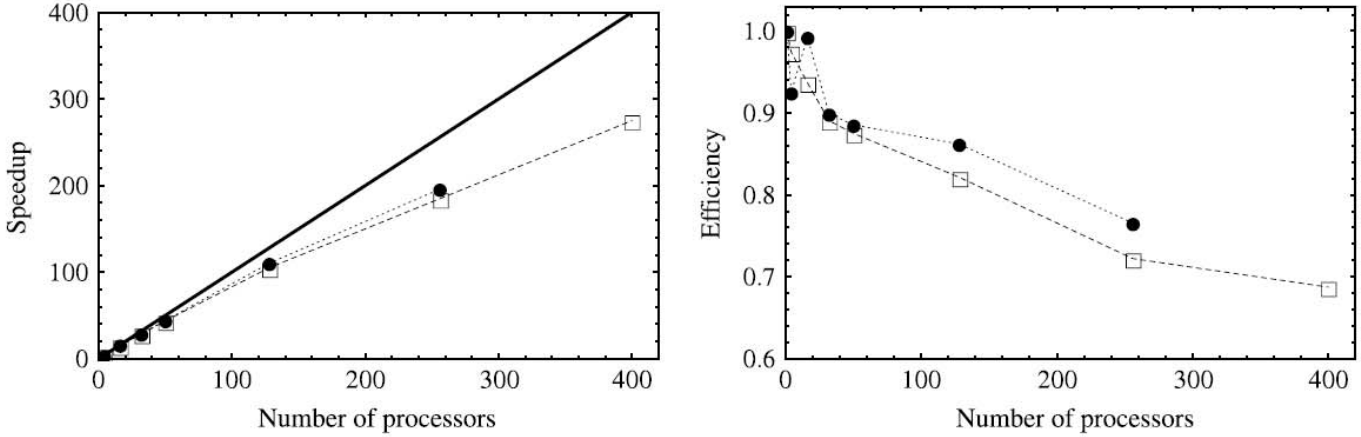
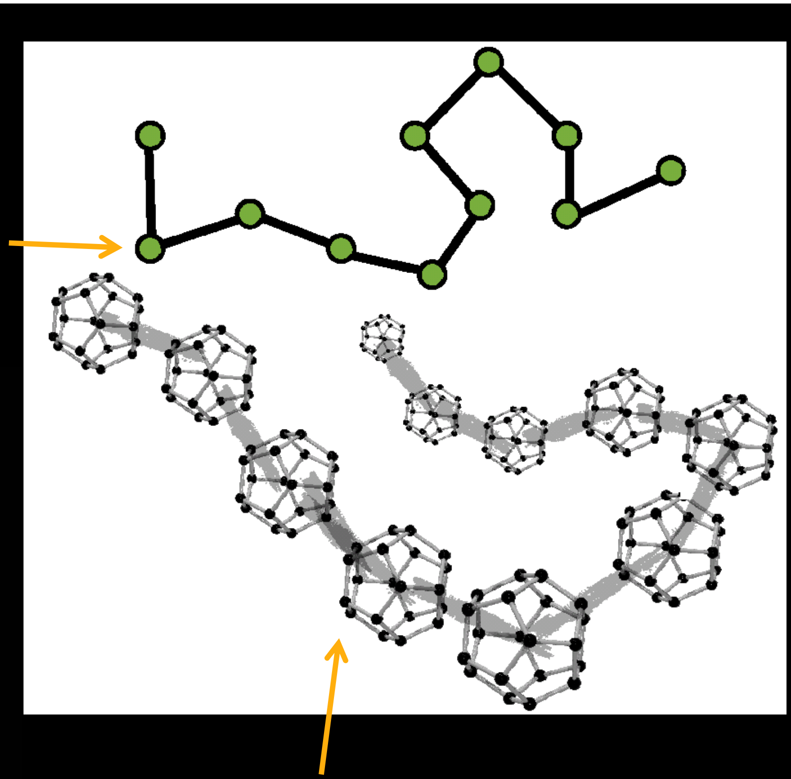


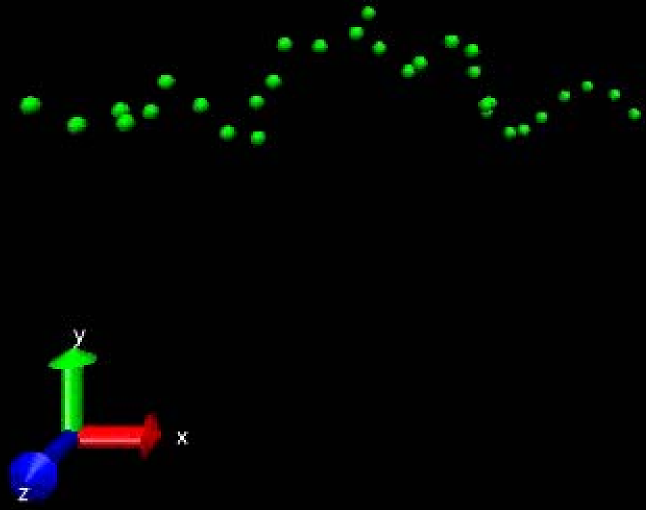
Fig. 9. Scalability results for the $280\Delta x \times 280\Delta x \times 100\Delta x$ lattice site, 1734240 MD particle node system. Solid circles correspond to simulations run on the SharcNet cluster, Requin, while hollow squares correspond to simulations run on the WestGrid cluster, Nestor. Left: speedup $= T_1/T_p$, with the solid line showing ideal, linear speedup. Right: efficiency $= T_1/(p \cdot T_p)$.

Polymer dynamics:



Point particles

Particles with extended size



Ref: JCP **134**, 064902 (2011)

Polymer Diffusion

Analytic value known = 2.837

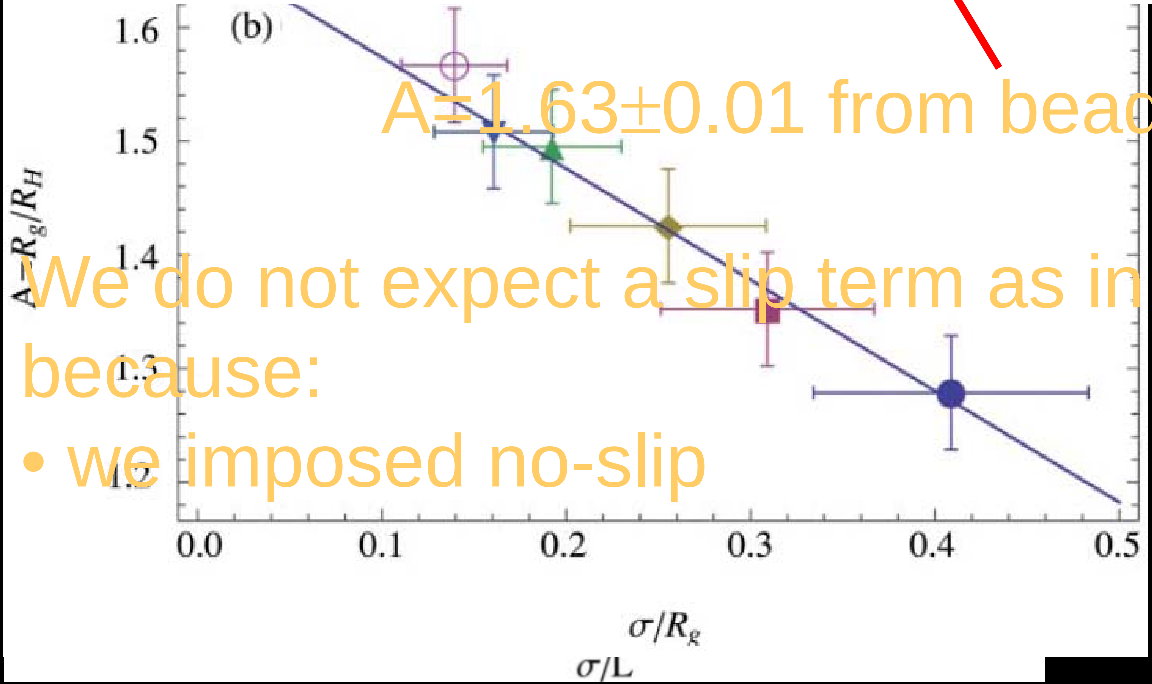
Analytic value known = 2.837

Expected:

$$D_{\mathrm{cm}} = \frac{k_{\mathrm{B}}T}{6\pi\eta}\left(\frac{A}{R_g} - \frac{B}{L}\right)$$

$$D_{\mathrm{cm}} = \frac{k_{\mathrm{B}}T}{6\pi\eta}\left(\frac{A}{R_g} - \frac{B}{L}\right)$$

A=1.63±0.01 from bead-spring calculation[1]



A=1.63±0.01 from bead-spring calculation[1]

We do not expect a slip term as in Kirkwood model because:

- we imposed no-slip

Asymptotic value

A=1.67 ±0.05

Slope gives

B=2.8±0.05

[1] Dünweg et al., JCP **117**, 914 (2002).

Ref: JCP **134**, 064902 (2011)

Polymer in a channel

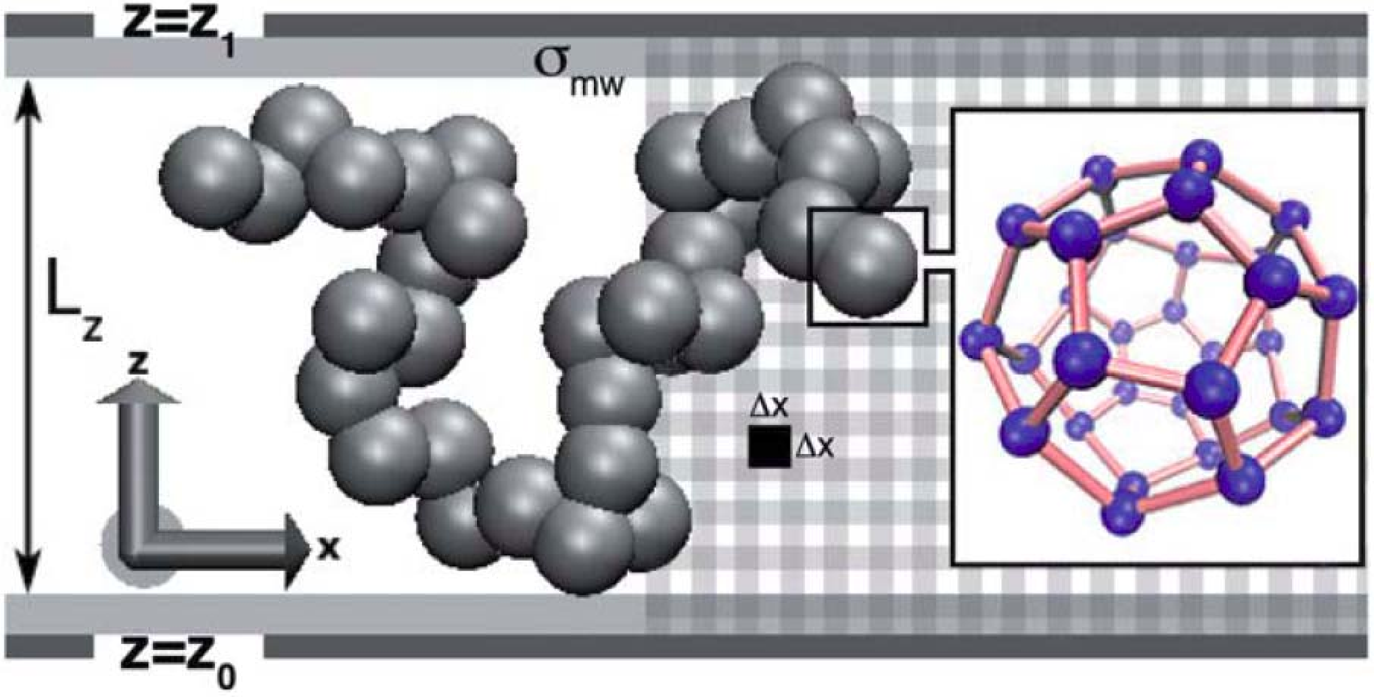


Fig. 1 An illustration (not to scale) of our polymer model and coordinate system in the present work. The frame on the right-hand side encloses a schematic of the structure of a 30-node composite monomer used in this work for hydrodynamic consistency.[30,34] The Lennard-Jones walls (see eqn (10)) are located at $z_0 = \Delta x$ and $z_1 = N_z\Delta x$, where N_z is the number of lattice layers and $\Delta x = 1.0$ nm is the resolution of the LB fluid lattice that is sketched as a grey mesh on the right-hand side. We use the D3Q15 LB lattice.[35] The picture of the polymer was generated in VMD.[36]

$$C = \frac{R_{G,\infty}}{L_z}$$

Diffusion

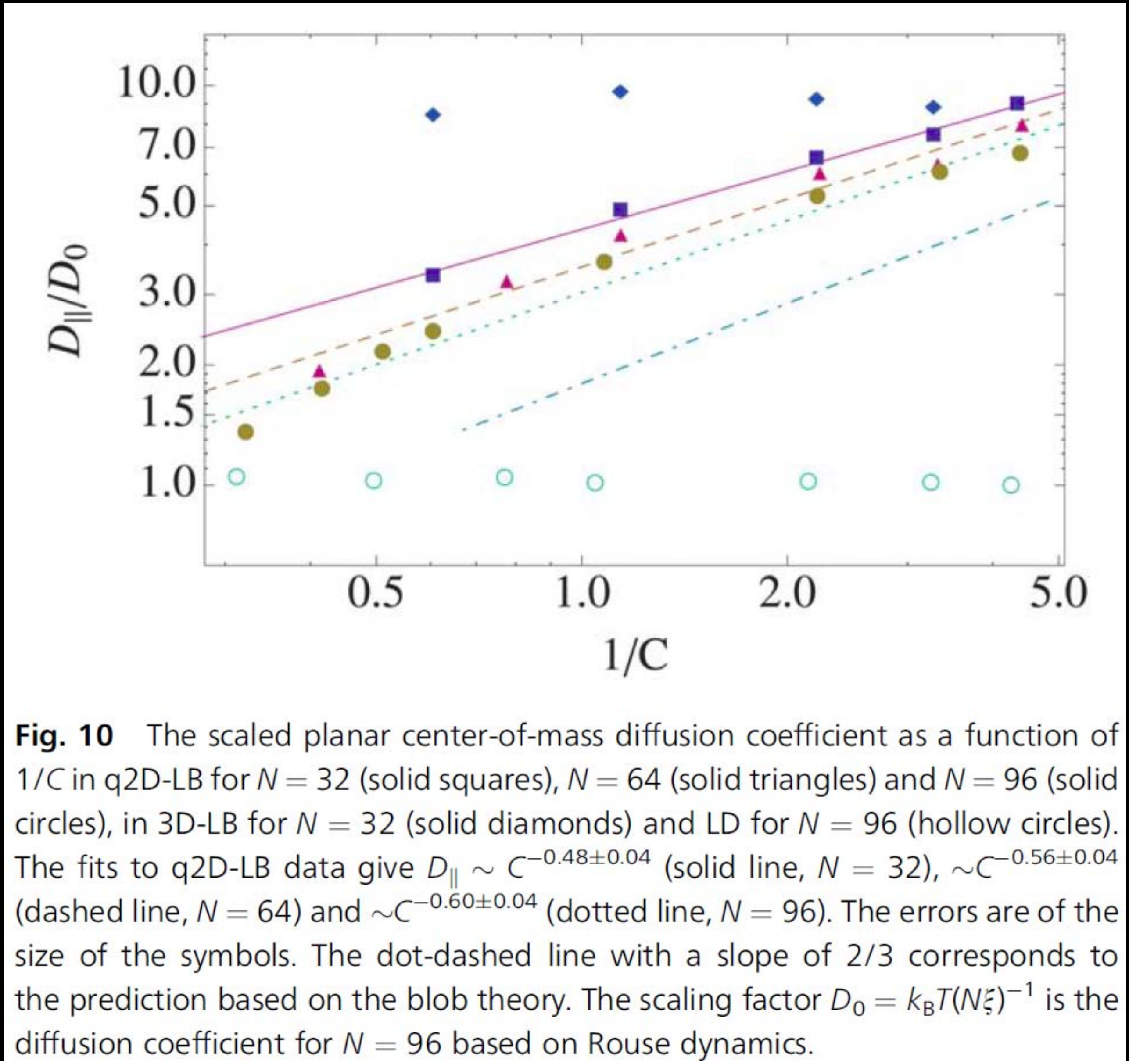
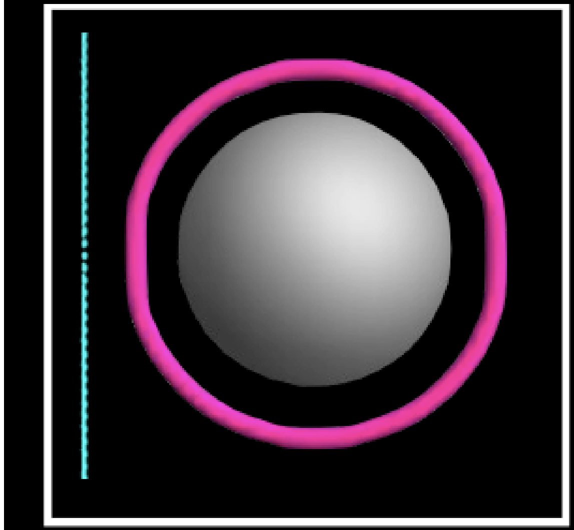
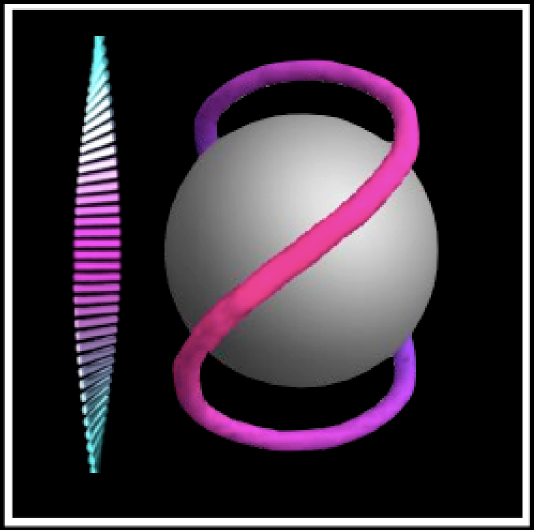


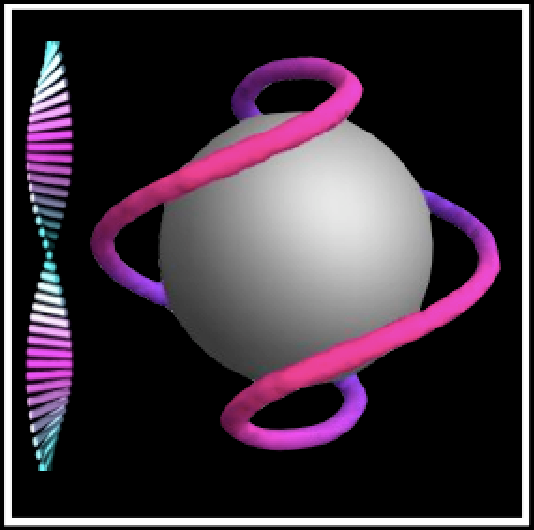
Fig. 10 The scaled planar center-of-mass diffusion coefficient as a function of $1/C$ in q2D-LB for $N = 32$ (solid squares), $N = 64$ (solid triangles) and $N = 96$ (solid circles), in 3D-LB for $N = 32$ (solid diamonds) and LD for $N = 96$ (hollow circles). The fits to q2D-LB data give $D_\parallel \sim C^{-0.48\pm0.04}$ (solid line, $N = 32$), $\sim C^{-0.56\pm0.04}$ (dashed line, $N = 64$) and $\sim C^{-0.60\pm0.04}$ (dotted line, $N = 96$). The errors are of the size of the symbols. The dot-dashed line with a slope of 2/3 corresponds to the prediction based on the blob theory. The scaling factor $D_0 = k_B T(N\xi)^{-1}$ is the diffusion coefficient for $N = 96$ based on Rouse dynamics.

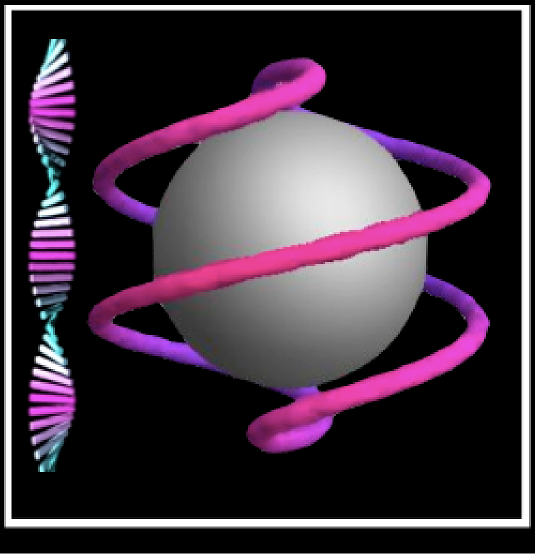
Ref: Soft Matter **9**, 3478 (2013)

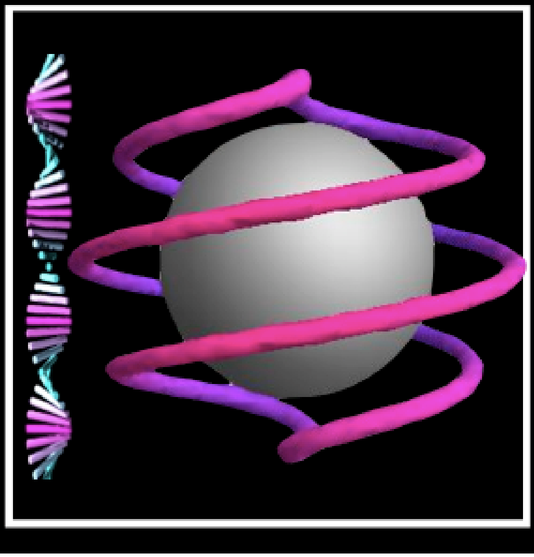
Colloids in a Liquid Crystal





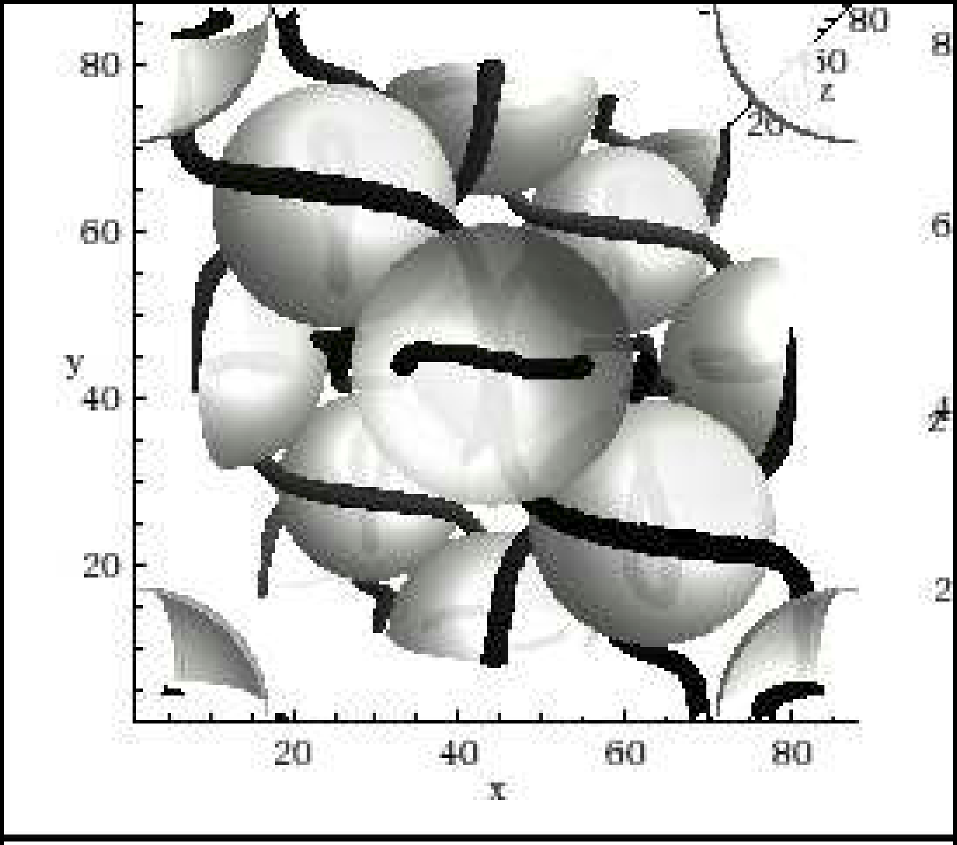






Ref: EPL **94**,
66003 (2011)

Colloids in LC in a lattice:



To Do:

- GPU package – preliminary version working (speedup ~ 75 for LB but now need to make LB domain decomposition bigger than particle domain decomposition)
- Lubrication forces at close distances (allows coarser mesh)
- Multiphase flow – have preliminary version of binary fluid and liquid crystal
- Multiscale within one simulation

Conclusions

- Fluctuations and particles were included in a lattice-Boltzmann model with a conservative coupling between MD and LB
- Inertia can matter at small Re for particles in flow
- Particles included in a way that guaranteed conservative coupling and allows the LB fluid to act as the thermostat for the particles
- Implemented in LAMMPS as fixes

Funding: NSERC, Ontario ERA, SharcNet