Modeling the fluid dynamics and molecular dynamics

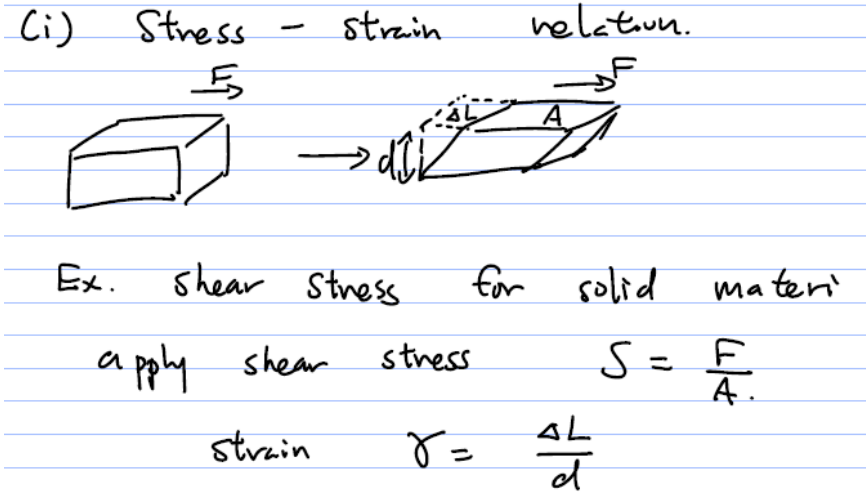
What is ‘Fluid Flow’ ?

When one applies pressure on a solid

- Structure deformation => energy conservation, elastic

When one applies pressure on a fluid

- Molecular collisions => momentum transfer, dissipation.

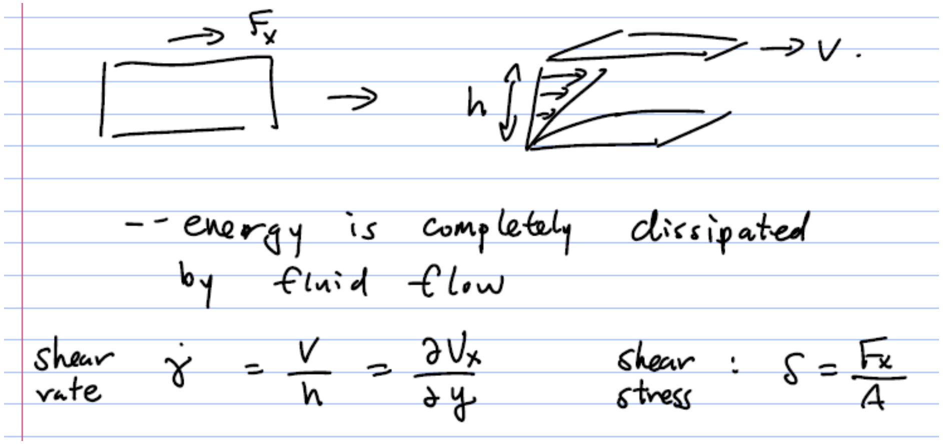


(i) Stress – strain relation.

Ex. shear stress for solid materi

apply shear stress $S = \frac{F}{A}.$

strain $\gamma = \frac{\Delta L}{d}$



-- energy is completely dissipated by fluid flow

shear rate $\dot{\gamma} = \frac{v}{h} = \frac{\partial v_x}{\partial y}$ shear stress : $S = \frac{F_x}{A}$

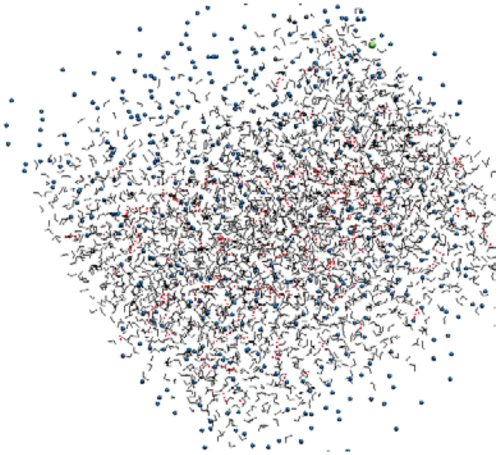
The number of water molecules in (10 nm)^3 volume ?

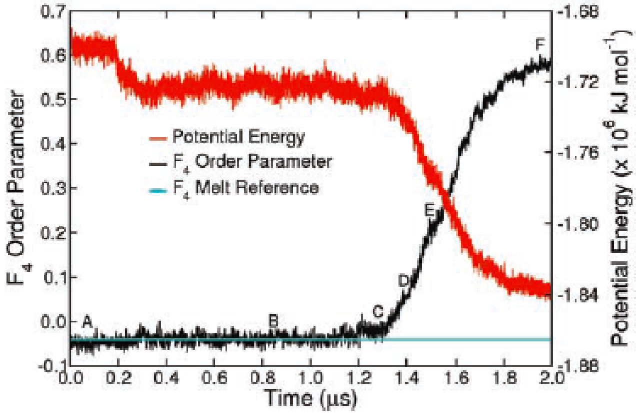
How about in (1 μm)^3 ?

Challenges of MD --

Simulation of methane in ice in 10x10x10 nm^3 volume for 2 μs

Walsh, Sum, Wu, Science (2009)





How to model the dynamics of macromolecules in a solution ?

Coarse-grain : Group many water molecules as an effective particle
Dissipative particle dynamics, Multi-particle collision dynamics

Continuum : Solve the Navier-Stokes equation by Finite Volume, Finite Element, Boundary integral

Pseudo-continuum: Lattice Boltzmann

Cost ~O(# of elements)

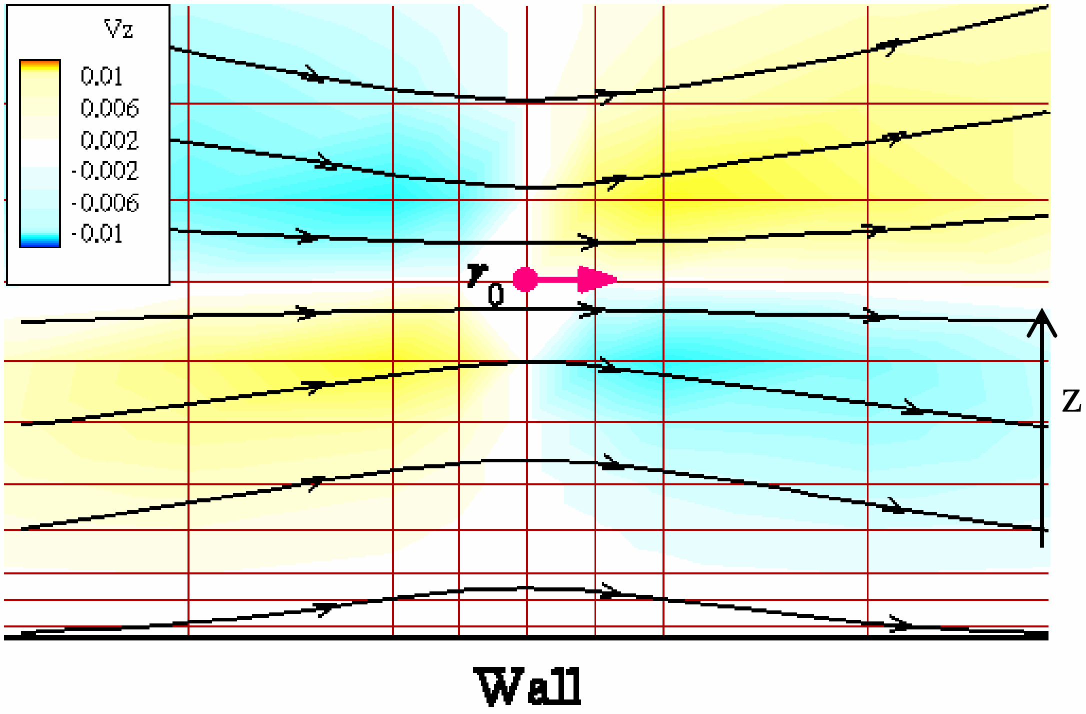
Continuum Fluid Modeling

Particle motion perturbs and contributes to the overall velocity field

$$\Delta\vec{\mathrm{v}}(\vec{r},\vec{r}_0,\vec{f}_0) = \underbrace{\Delta\vec{\mathrm{v}}_s(\vec{r}-\vec{r}_0)}_{\text{Free space}} + \underbrace{\Delta\vec{\mathrm{v}}_W(\vec{r},\vec{r}_0)}_{\text{Wall correction}} = \boxed{\boldsymbol{\Omega}(\vec{r},\vec{r}_0)}\cdot\underbrace{\vec{f}_0}_{\text{Force}}$$

Stokes Flow

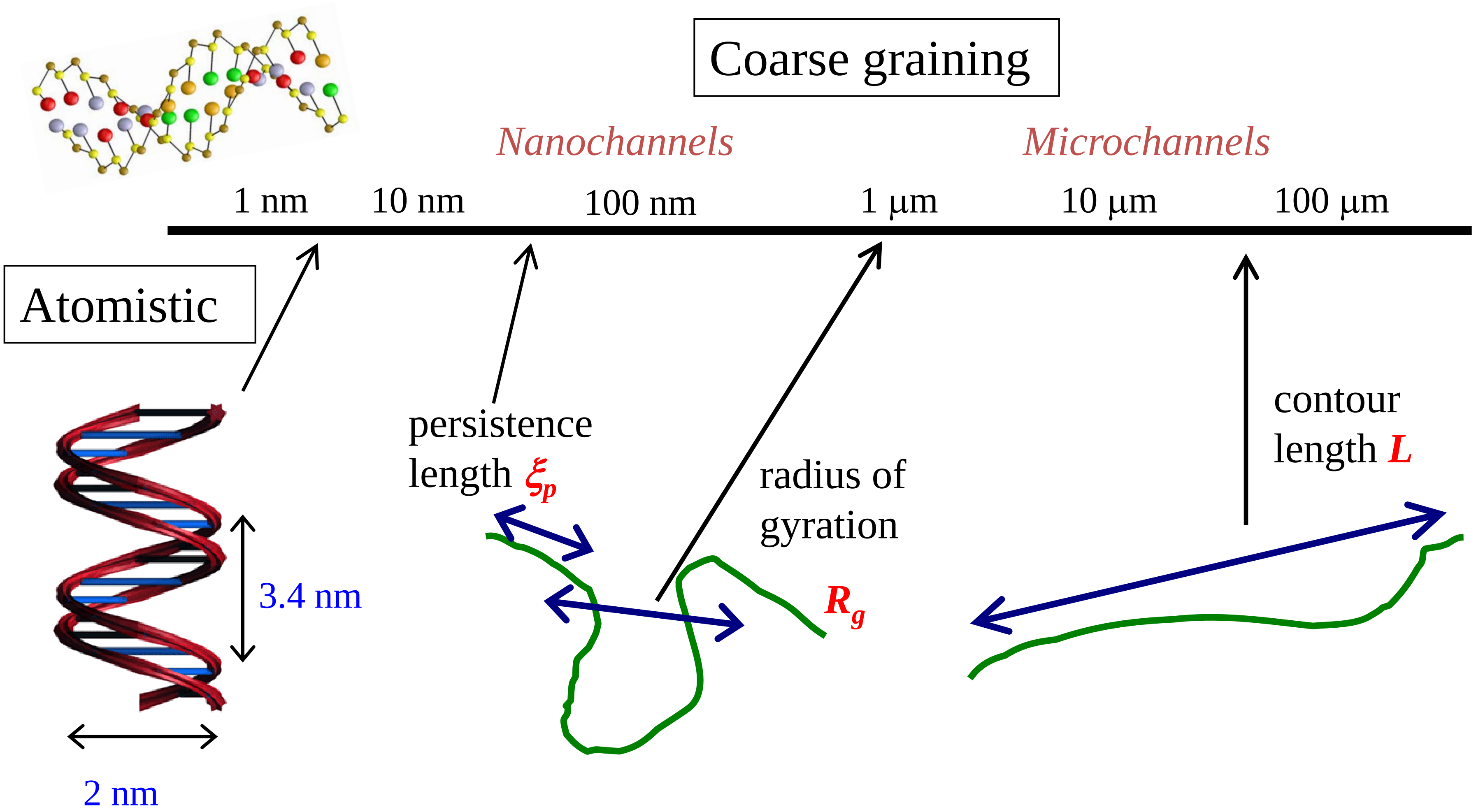
$$0 = -\nabla p + \eta\nabla^2\vec{\mathrm{v}}_W$$
$$0 = \nabla\cdot\vec{\mathrm{v}}_W$$



Solved w/
Finite Element Method

HI is dependent on particle positions as a function of time

Multi-Scale Modeling Approach



Coarse graining

Nanochannels

Microchannels

1 nm

10 nm

100 nm

1 μm

10 μm

100 μm

Atomistic

persistence length ξ_p

radius of gyration R_g

contour length L

3.4 nm

2 nm

Capture essential physics and optimize computational complexity

Flow timescale ~ minutes => coarse-grained model

Coarse-grained DNA Molecule

DNA molecule - short length scale: double helix
long length scale: flexible polymer

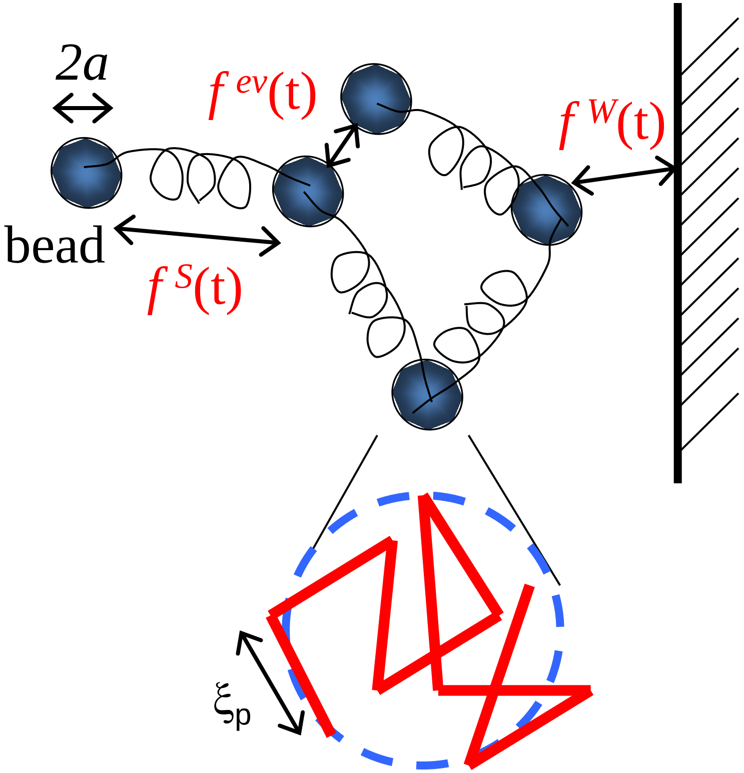
contour length ~ o(10-10^2) μm
radius of gyration ~ o(1) μm

$\xi_p \approx 50$ nm

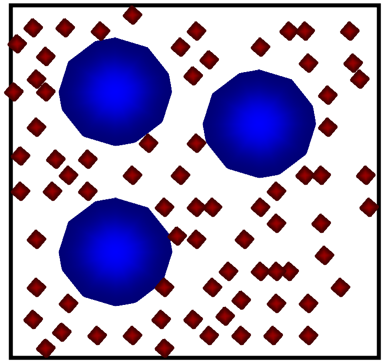
Model parameters matched to material specific properties of TOTO-1 stained λ-DNA

Approximate DNA as a *Worm-like Chain*

- Molecular scale features are coarse-grained
- Blobs interact with each other through
  - Elastic spring
  - 'Soft' excluded volume
  - Hydrodynamic interactions



Brownian Dynamics and Hydrodynamics



$$d\vec{R} = \left[\vec{U} + \frac{1}{k_B T}\mathbf{D}\cdot\vec{F} + \nabla\cdot\mathbf{D}\right]dt + \sqrt{2}\mathbf{B}\cdot d\vec{W}$$

Flow Field Force Drift Brownian

Solve Continuum Hydrodynamics

Diffusion tensor is a function of the *instantaneous molecular configuration* and *geometry* of the device

$$\mathbf{D}_{ij} = \frac{k_B T}{\zeta}(\mathbf{I}\delta_{ij} + \zeta\mathbf{\Omega}_{ij})$$

Cost ~ O($N^{2.2}$)

ζ: particle friction coef.

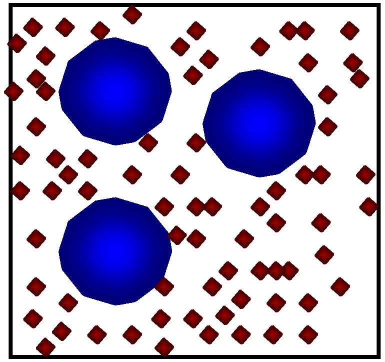
$\mathbf{\Omega}_{ij}$: hydrodynamic interaction tensor

Thermal Brownian Motion

Fluctuation-dissipation theorem couples hydrodynamics to Brownian forces

$$\sum_{j=1}^{N_b}\mathbf{B}_{ij}\mathbf{B}_{kj} = \mathbf{D}_{ik}$$

Brownian Dynamics and Hydrodynamics



$$d\vec{R} = \left[\vec{U} + \frac{1}{k_B T}\mathbf{D}\cdot\vec{F} + \nabla\cdot\mathbf{D} \right] dt + \sqrt{2}\,\mathbf{B}\cdot d\vec{W}$$

Flow Field Force Drift Brownian

Solve Continuum Hydrodynamics

Diffusion tensor is a function of the *instantaneous molecular configuration* and *geometry* of the device

$$\mathbf{D}_{ij} = \frac{k_B T}{\zeta}\left(\mathbf{I}\delta_{ij} + \zeta\mathbf{\Omega}_{ij}\right)$$

Cost ~ O($N^{2.2}$)

ζ: particle friction coef. $\mathbf{\Omega}_{ij}$: hydrodynamic interaction tensor

Thermal Brownian Motion

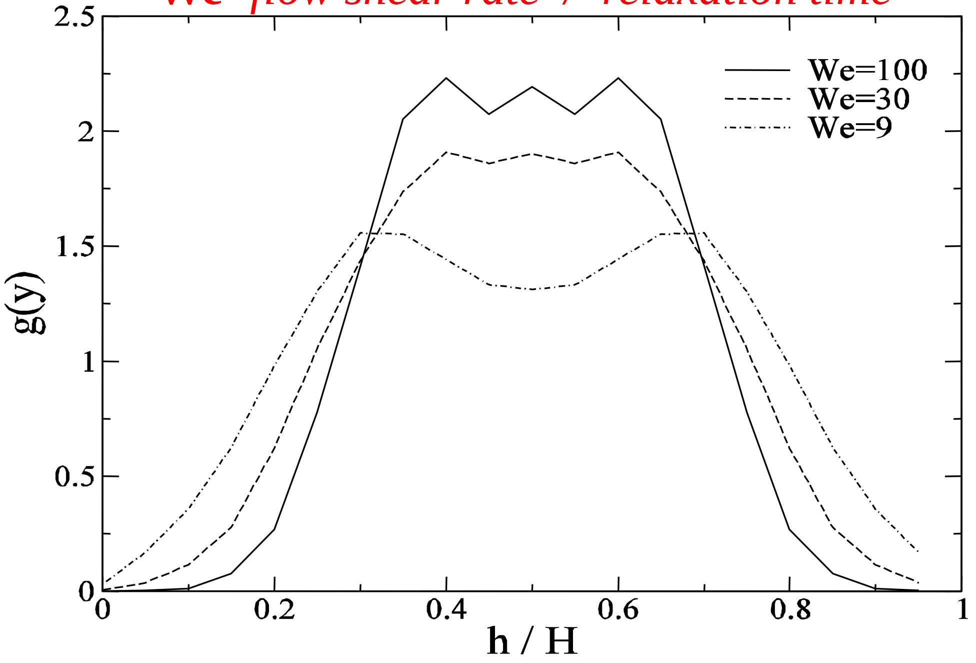
Fluctuation-dissipation theorem couples hydrodynamics to Brownian forces

$$\sum_{j=1}^{N_b} \mathbf{B}_{ij}\mathbf{B}_{kj} = \mathbf{D}_{ik}$$

DNA in microflow : Flow-induced DNA migration

λ-DNA N_c=50 c_p/c_p*=0.02

We=*flow shear rate / relaxation time*



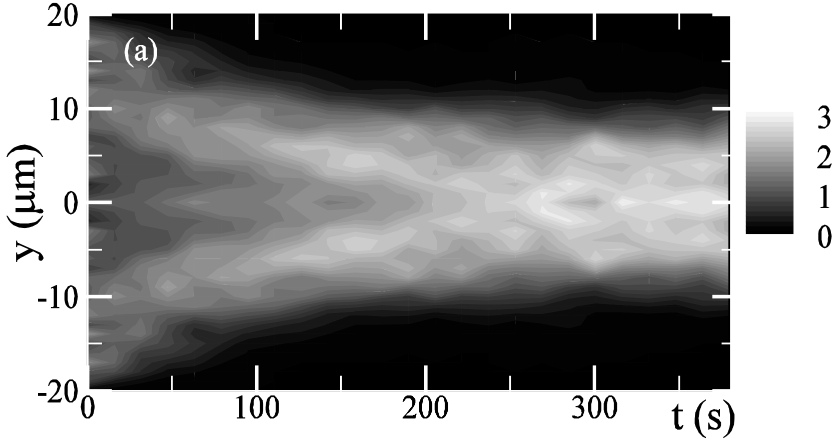
Chain stretch when

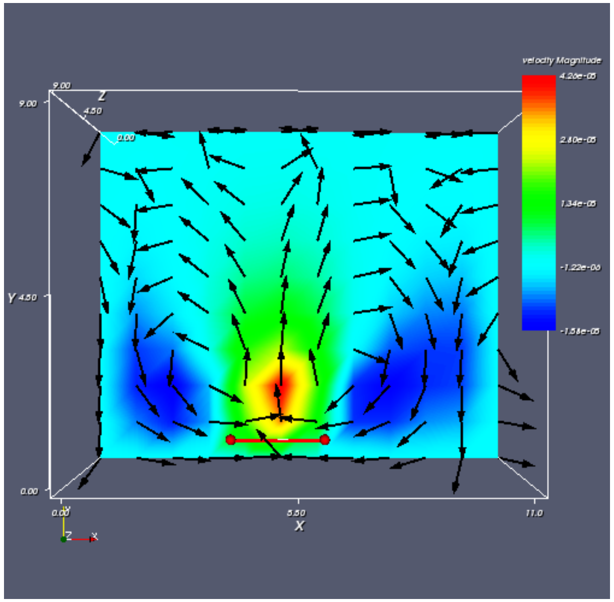
flow shear rate > (relaxation time)$^{-1}$

Steady state is reached after 100 s

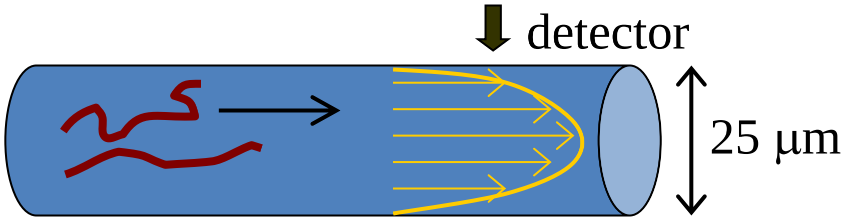
Chen et al., Macromolecules (2007)

DNA distribution in microchannel





DNA Separation in Microcapillary



λ-DNA in microcapillary flow

Sugarman & Prud'homme (1988)

Chen et al.(2005)

Detection points at 25 cm and 200 cm

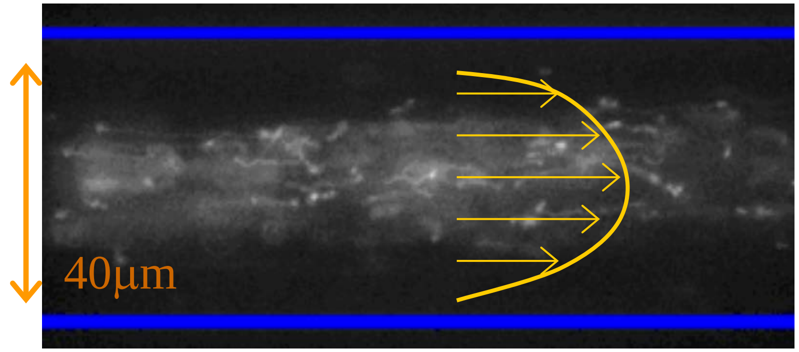
$$\boxed{R_f = \frac{\text{avg. DNA velocity}}{\text{max. fluid velocity}}}$$

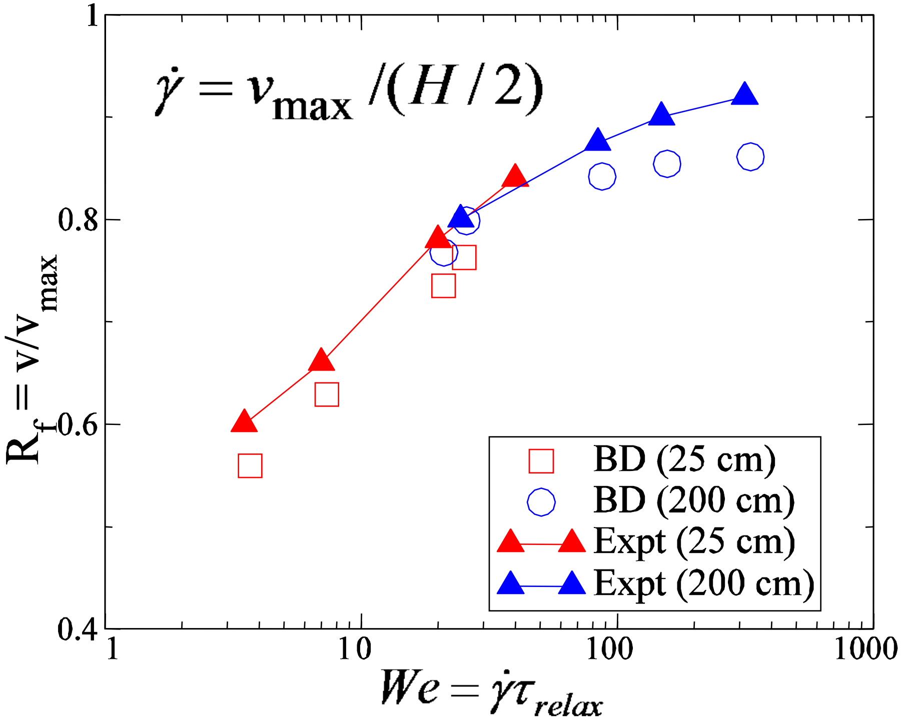
Longer DNA → higher velocity
Separation by MW is possible

Key challenges : resolution and speed

Cost ~ O($N^{2.2}$)

T2 DNA after 100 s oscillatory Poiseuille flow

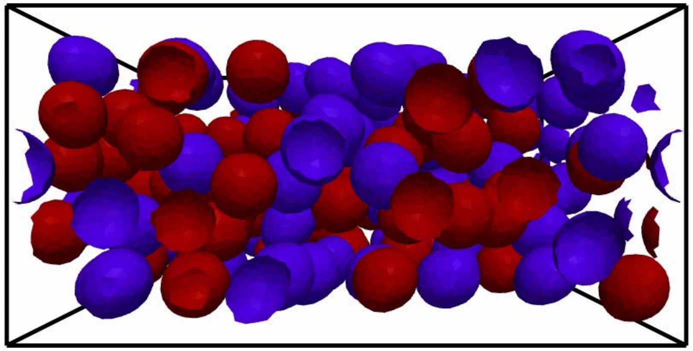




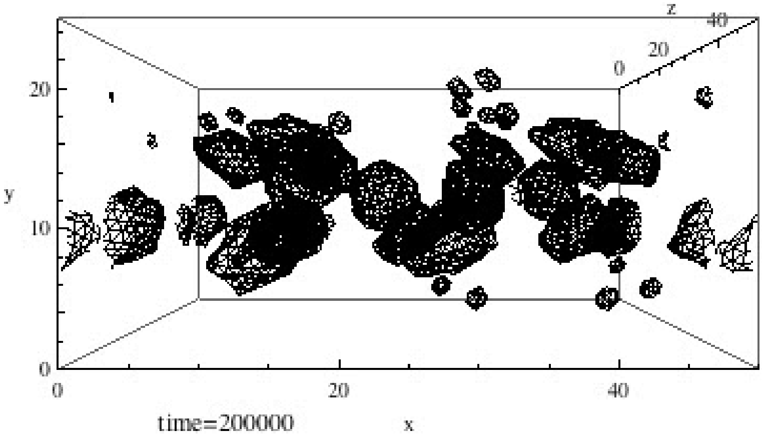
Modeling Cell Dynamics in Microflow

Flow separation of soft particles / droplets by size, elasticity, shape

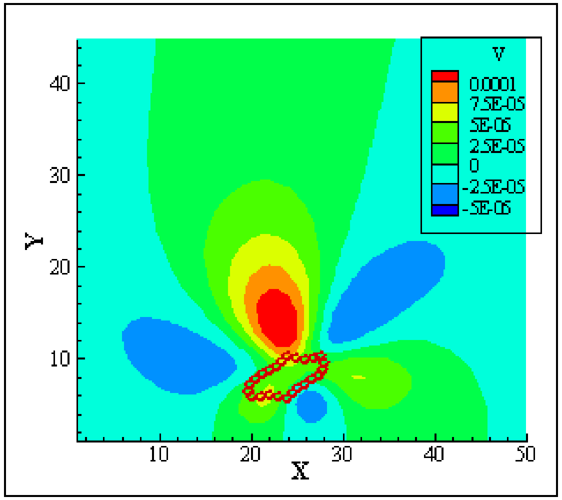
Mechanisms : Deformation – flow interactions, fluid inertia

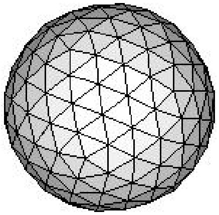


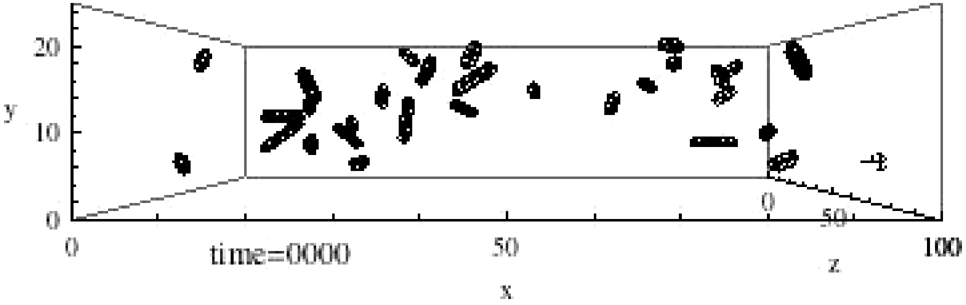
Elasticity



Size



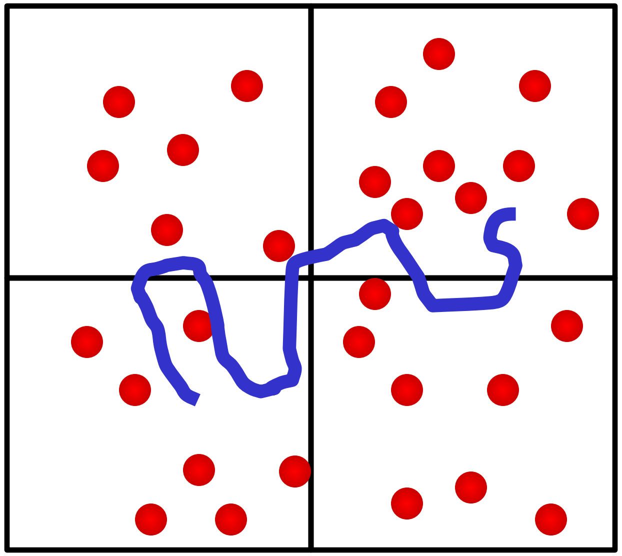




Shape

Hsu and Chen, J. Chem. Phys., (2010); YL Chen, RSC Advances (2014)

Multi-Particle Collision



Fluid hydrodynamics by simulating fluid particle dynamics in two steps:

Streaming – Newton EOM

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \Delta t\, \mathbf{v}_i(t)\,,$$

Collision – thermalize fluid velocities after collisions

$$\mathbf{v}_i(t+\Delta t) = \mathbf{u}(t) + \mathbf{v}_{i,\mathrm{ran}} - \sum_{\mathrm{cell}} \mathbf{v}_{i,\mathrm{ran}}/N_{\mathrm{c}}$$

thermostat

Angular momentum conservation

$$+\left\{ m\Pi^{-1} \sum_{j\in \mathrm{cell}} \left[\mathbf{r}_{j,\mathrm{c}} \times \left(\mathbf{v}_j - \mathbf{v}_j^{\mathrm{ran}}\right)\right] \times \mathbf{r}_{i,\mathrm{c}} \right\}$$

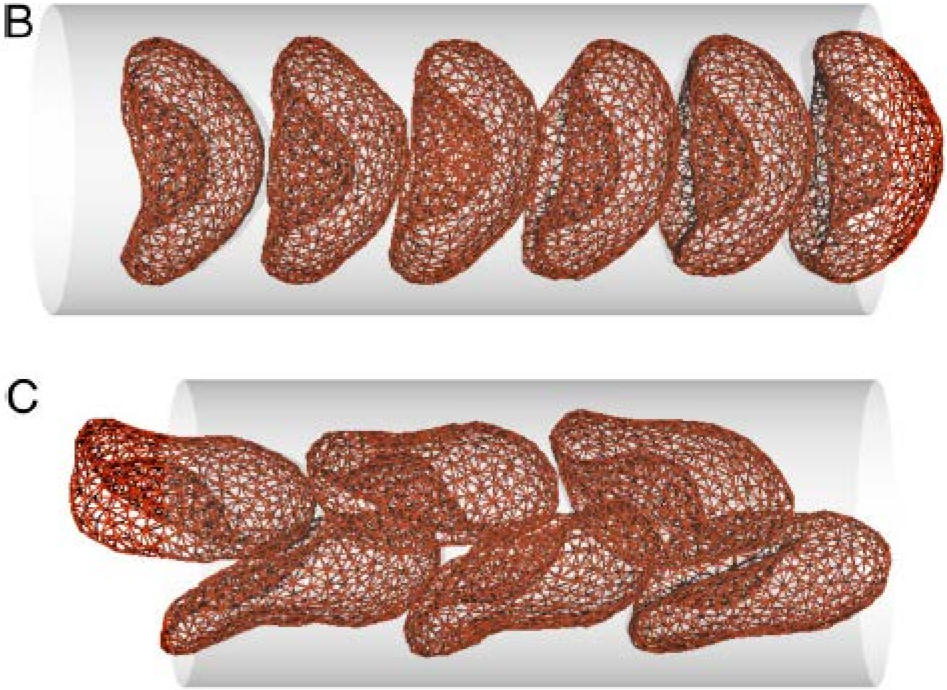
$$\mathbf{u} = \frac{m\sum_{i=1}^{N_{\mathrm{c}}} \mathbf{v}_i + m_{\mathrm{m}} \sum_{i=1}^{N_{\mathrm{m}}} \mathbf{w}_i}{N_{\mathrm{c}} m + N_{\mathrm{m}} m_{\mathrm{m}}}\,,$$

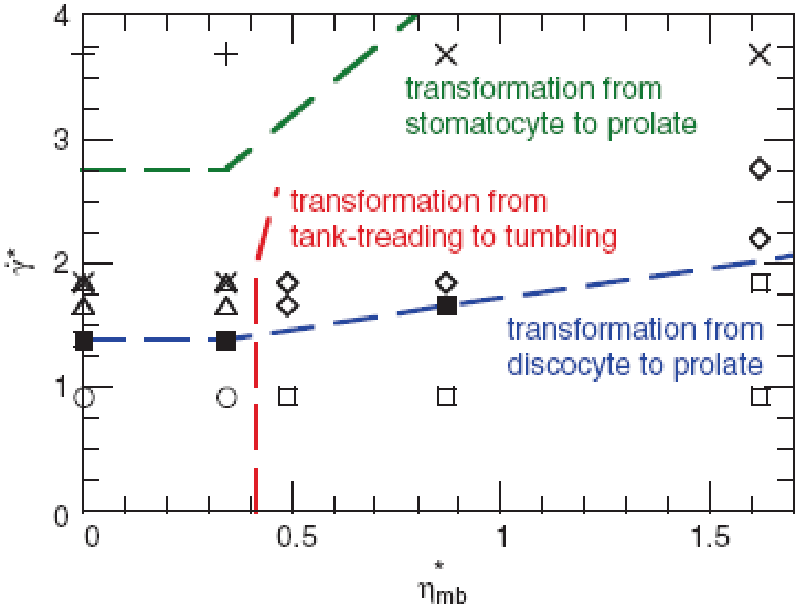
Momentum exchange between fluid particle and monomers

Gomppper et al., Adv. Polymer Sci. 221 (2009)

MPC modeling of RBC dynamics

MPC has been applied to characterized the shape change and dynamic change of elastic particles with different membrane viscosity and flow rate

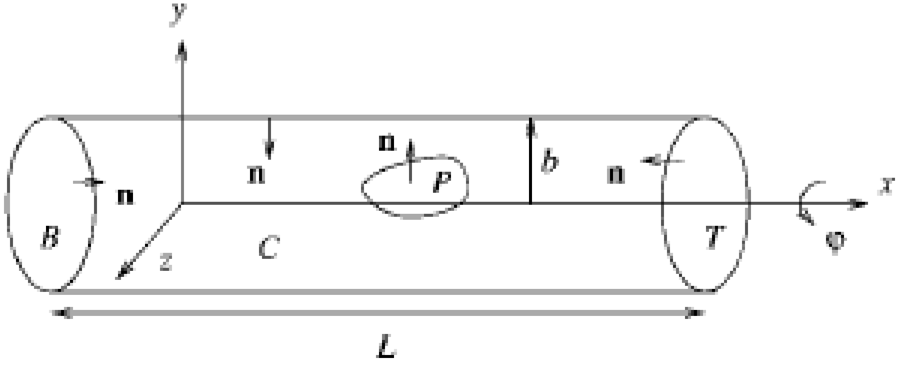




Gompper et al., Adv. Polymer Sci. 221 (2009)

Boundary Integral Method

Solving for the flow field due to a cell with a boundary integral representation



Stokes flow of point sources

$$u_j^D(\mathbf{x}_0) = -\frac{1}{8\pi\mu}\iint_{P+,C,B,T} G_{ij}(\mathbf{x},\mathbf{x}_0)\, f_i^D(\mathbf{x})\,\mathrm{d}S(\mathbf{x})$$

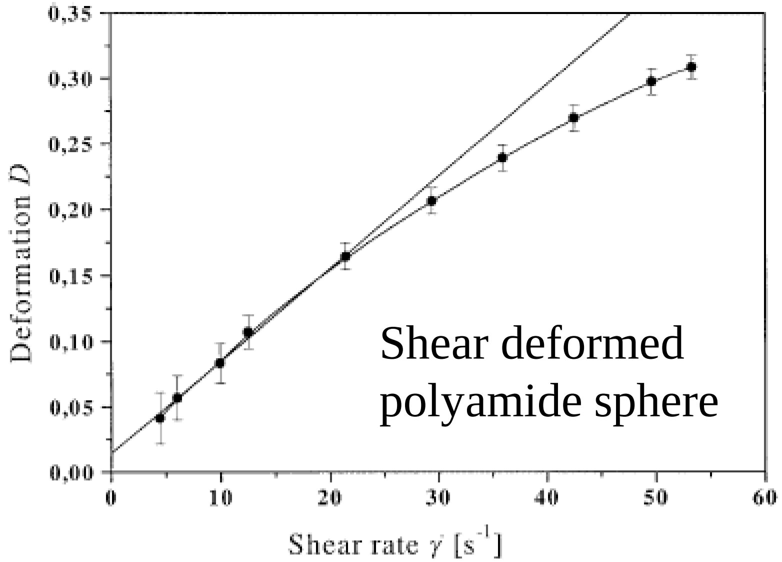
$$+\frac{1}{8\pi}\iint_{P,C,B,T} u_i^D(\mathbf{x})\, T_{ijk}(\mathbf{x},\mathbf{x}_0)\, n_k(\mathbf{x})\,\mathrm{d}S(\mathbf{x}),$$

Stokes flow of point dipoles + sources

$$G_{ij}(\mathbf{x},\mathbf{x}_0) = \frac{\delta_{ij}}{|\mathbf{x}-\mathbf{x}_0|} + \frac{(x_i - x_{0_i})(x_j - x_{0_j})}{|\mathbf{x}-\mathbf{x}_0|^3}$$

$$T_{ijk}(\mathbf{x},\mathbf{x}_0) = -6\,\frac{(x_i - x_{0_i})(x_j - x_{0_j})(x_k - x_{0_k})}{|\mathbf{x}-\mathbf{x}_0|^5}$$

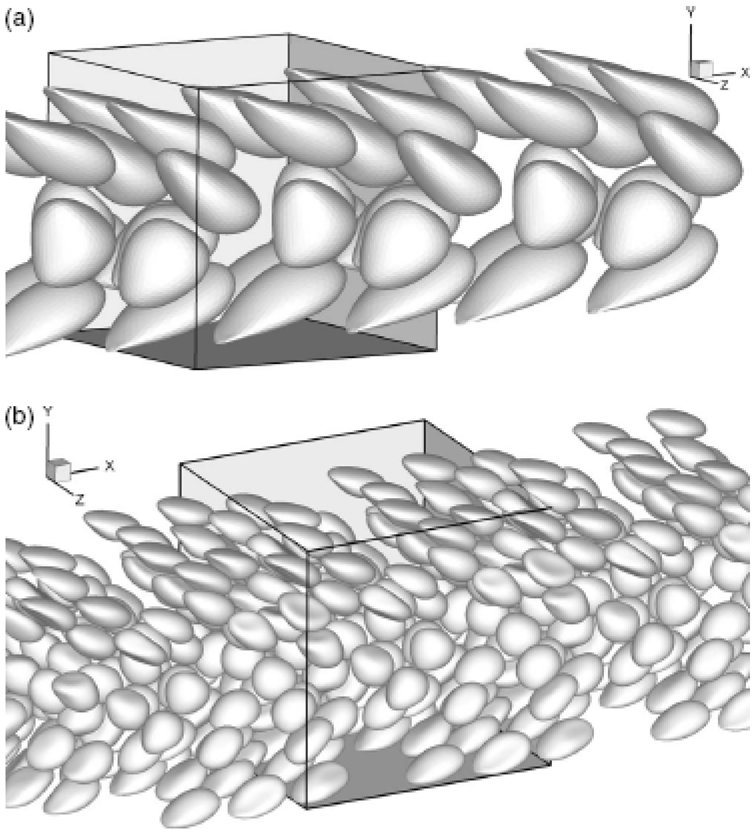
Barthes-Biesel et al., J. Comp. Phys (1989)
Pozrikidis, Modeling and Simulation of Capsules and Biological Cells (2003)



Shear deformed polyamide sphere

Walter et al., Rheol. Acta. (2002)

Application of Immersed Boundary Method



Doddi & Bagchi, (2009)
Pan, Shi, Glowski (2010)

The immersed boundary method has been applied to the flow of o(100) vesicles.

Qualitatively captures the Fahraeus-Lindqvist effect.

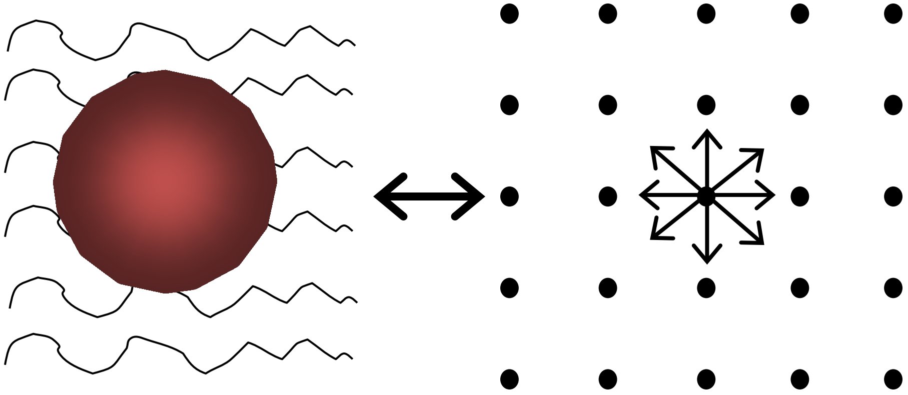
Limitation : Cells do not come too close to each other in the computation. Near-field hydrodynamics require very fine fluid grid.

The Boltzmann equation and lattice Boltzmann

Duenweg and Ladd, Adv. In Polymer Sci (2009)

Replace continuum fluid with discrete fluid positions x_i and discrete velocity c_i

$n_i(r,v,t)$ = fluid velocity distribution function



3D, 19-vector model

Hydrodynamic fields are moments of the velocity distribution function

$$\rho = \sum_i n_i \qquad j = \sum_i n_i c_i$$

$$\Pi = \sum_i n_i c_i c_i$$

Ghost Moments

Boltzmann eqn.

$$\partial_t n + v \cdot \nabla n = \left(\frac{dn}{dt} \right)_{coll}$$

$$\Delta_i[\boldsymbol{n}(r,t)] = \sum_j L_{ij}(n_j - n_j^{eq})$$

$$n_i(r + c_i \Delta t, t + \Delta t) = n_i(r,t) + \Delta_i[\boldsymbol{n}(r,t)]$$

L_{ij} = local collision operator

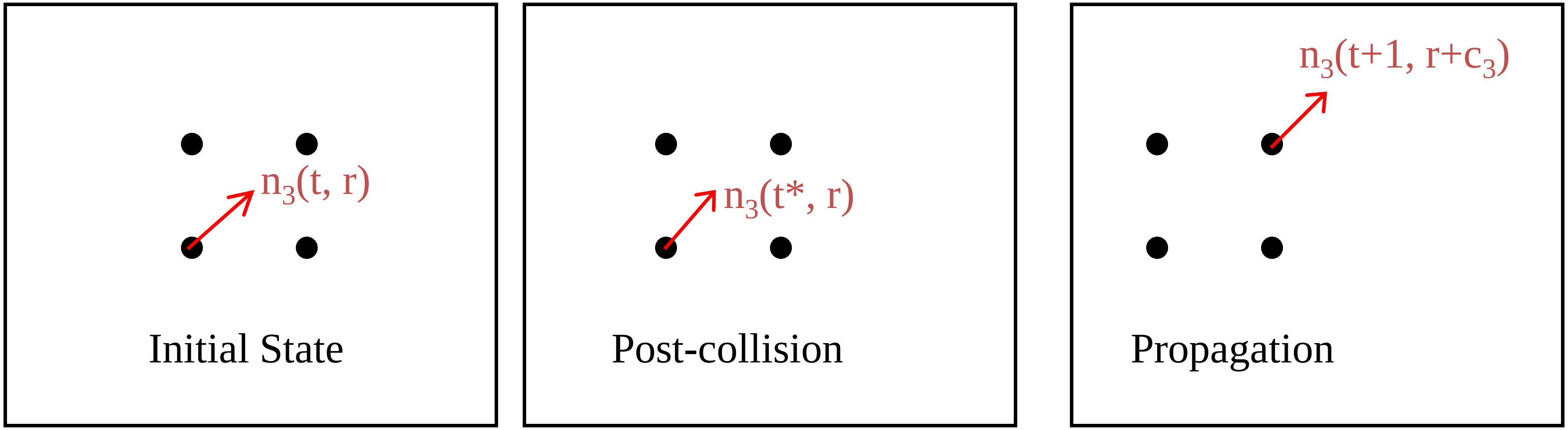
=1/τ in the simplest approx.

Local collisions and propagation

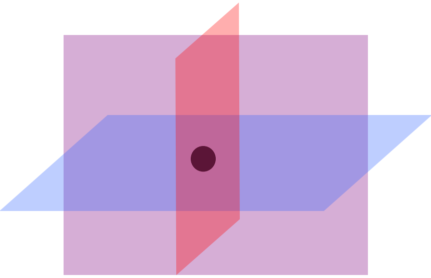
lbe.c

- Hydrodynamics comes from local collisions between fluid particles
- Collisions are uncorrelated between timesteps

$$n_i(r+c_i,t+1) = n_i(r,t) + L_{ij}[n_j(r,t) - n_j^{eq}(r,t)]$$



Common 3D model has 19 velocities c_i :
0, ±x, ±y, ±z, ±xz, ±xy, ±yz

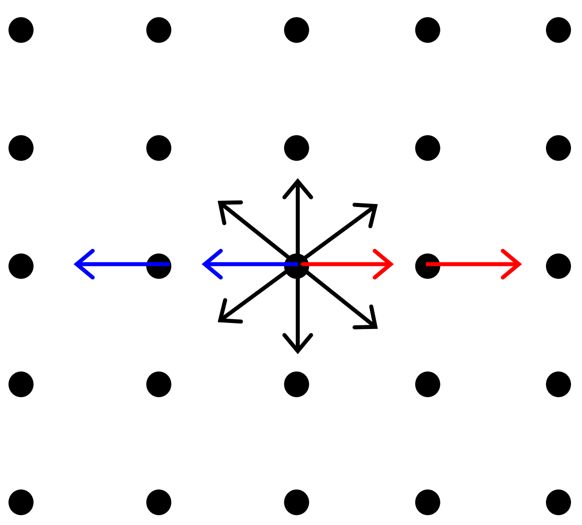


Simulation Steps

lbe.c

- Hydrodynamics comes from local collisions at solid-fluid boundaries
- Collisions are uncorrelated between timesteps

$$n_i(r+c_i,t+1)=n_i(r,t)+L_{ij}[n_j(r,t)-n_j^{eq}(r,t)]$$



1. Equilibration of fluid velocity u_i
2. Fluid particle collision – instant exchange of mass and momentum
3. Fluid-solid boundary condition
4. Propagation of fluid velocity distribution functions

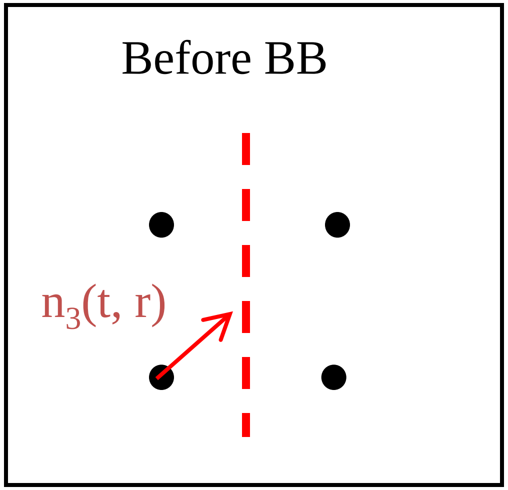
Solid-Fluid Boundary Conditions

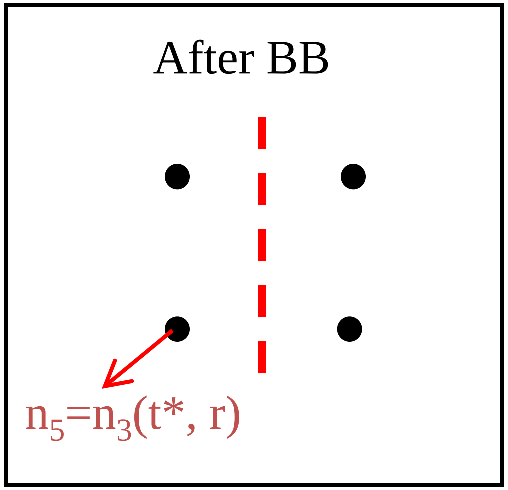
bnodes.c

Bounce Back Rule

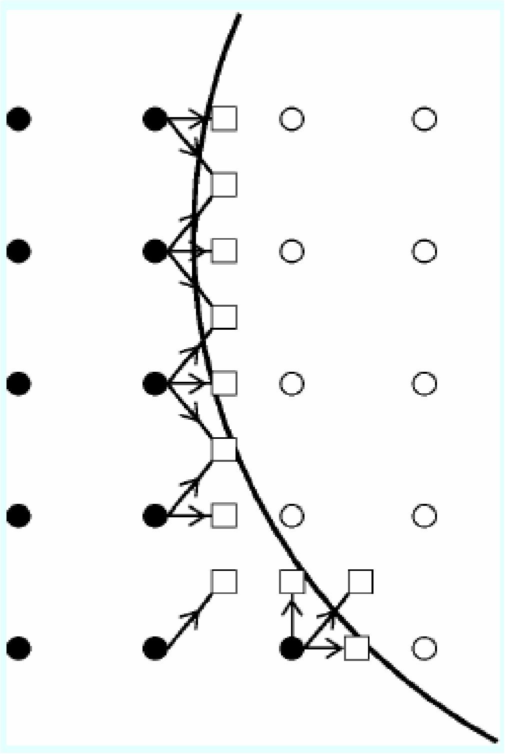
$$n_{i'}(r, t+1) = n_i(r, t^*)$$
$$n_i(r + c_i, t+1) = n_{i'}(r + c_i, t^*)$$

No slip - Fluid velocity at wall is zero





Fluid-Particle Boundary



Nguyen & Ladd (2002)

Momentum is transferred across the fluid-particle boundary

Solid-Fluid Boundaries

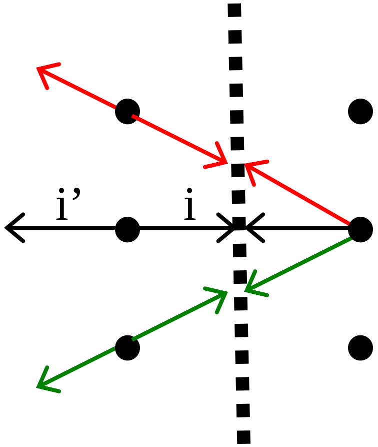
bnodes.c

The Bounce Back Rule

No slip - Fluid velocity at wall is zero

Fluid site

$$n_{i'}(r,t+1) = n_i(r,t_+)$$



Stationary Wall

Wall site

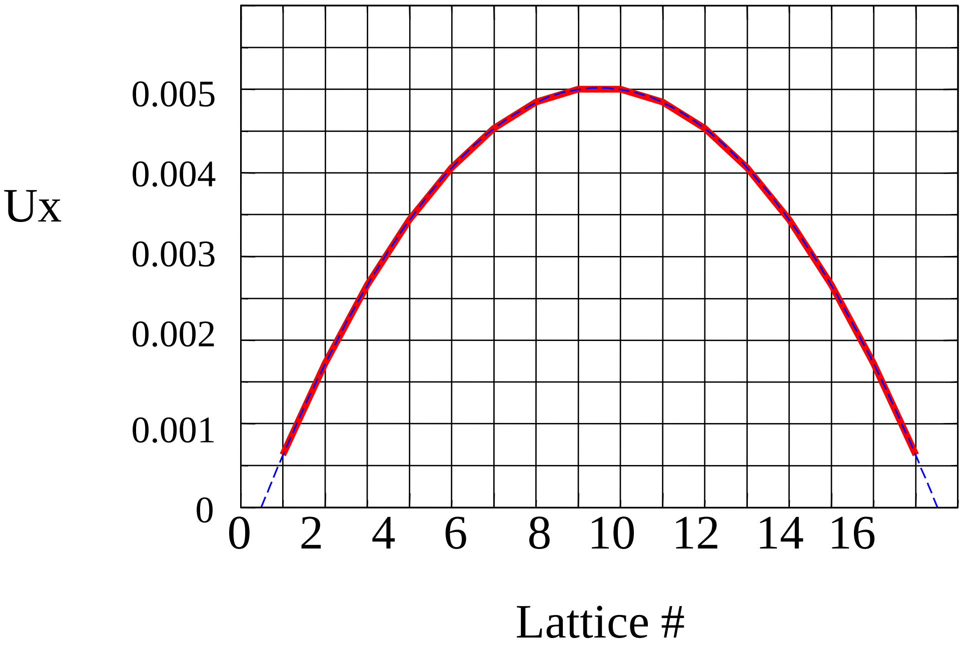
$$n_i(r+c_i,t+1) = n_{i'}(r+c_i,t_+)$$

$n_i(r,t_+)$ = postcollision distribution of velocity i at r

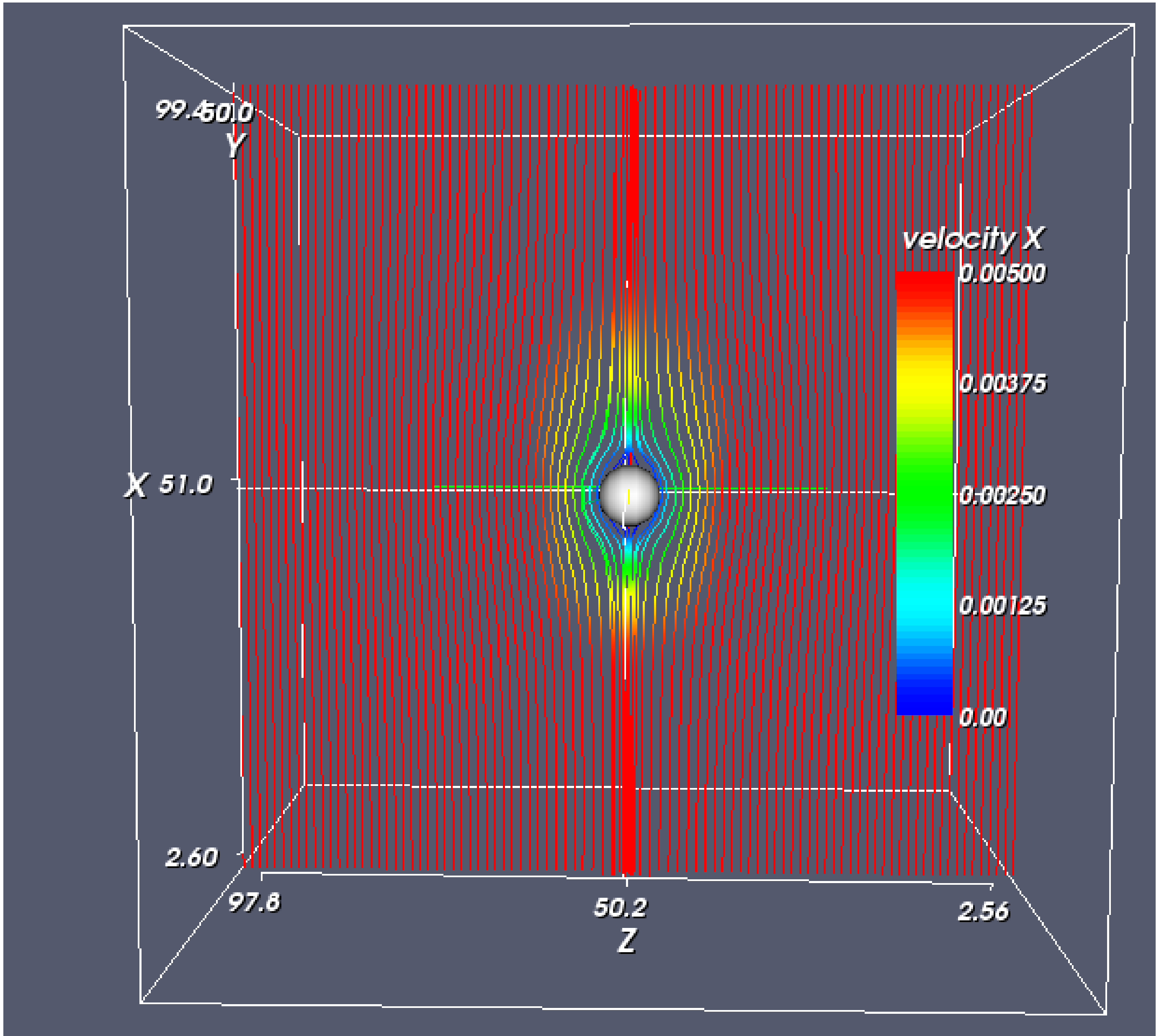
Pressure-driven flow between parallel plates

Parabolic velocity profile attained by applying no-slip and constant pressure (momentum flux) to the fluid

$$\nabla p = \eta \nabla^2 \mathbf{u} + \mathbf{f}$$



Creeping flow around a fixed particle



R = 4

V=0.005

Fluid Fluctuations

Brownian motion particularly important for submicron particles

Noises are present for all non-conserved eigenmodes of the DdQq system

How to add proper noises that satisfly the fluctuation-dissipation theorem ?

Reference: Adhikari, Stratford, Cates, Wagner (2004)

1. Find eigenvectors through Gram-Schmidt orthogonalization
2. Find variance of the non-conserved stress and ghost modes
3. Apply FDT and generate noise for the c_i

With assumptions : correlations are **k**-independent

stationarity of equal time-correlators

lbe.c

From FDT :

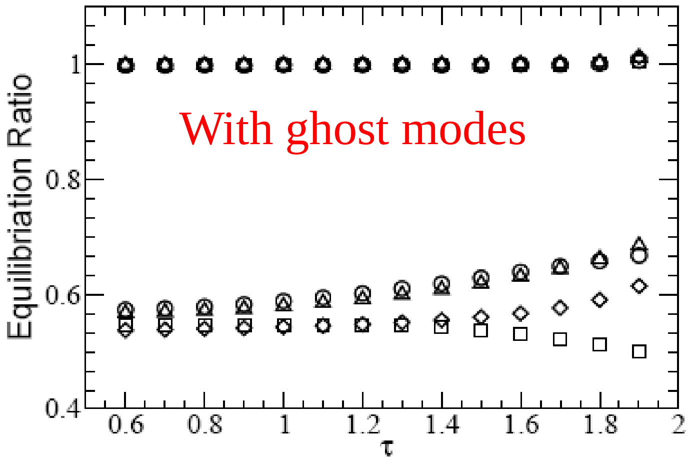
$$\left\langle n'^{a} n'^{b} \right\rangle = \frac{\tau_a + \tau_b - 1}{\tau_a \tau_b} \left\langle \delta M^{a} \delta M^{b} \right\rangle$$

τ_a, τ_b are the relaxation time of the modes a and b

Choosing the thermodynamics of an ideal gas

$$\left\langle \delta M^{a} \delta M^{b} \right\rangle = m_{aj} m_{bj} \frac{kT}{c_s^{2}} n_j^{eq} \delta_{ab}$$

Follows Poisson statistics



With ghost modes

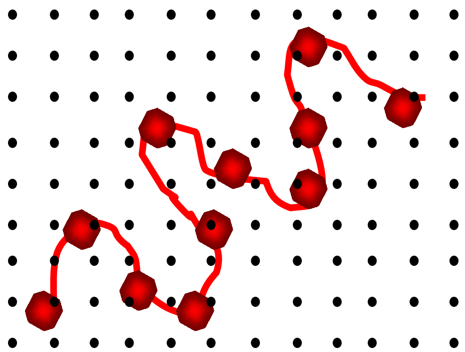
Without ghost modes

Equilibration ratio of density, velocity, and stress modes

“Immersed Boundary Method” + LBM

get_forces.c : hi_force()

Ahlrichs and Dunweg (1998)



Couple MD simulation of chains with LBM

At each time step:

(1) LBM calculates the flow field

(2) Fluid velocity at bead position is calculated by interpolation

(3) Friction force on the monomer is calculated

$$F_f = -\zeta (U_p - U_f(x))$$

$\zeta = 6 \pi \eta a$

(4) Momentum change is distributed among the nearest lattice neighbors

(5) Fluctuation is added to the monomers to balance frictional losses

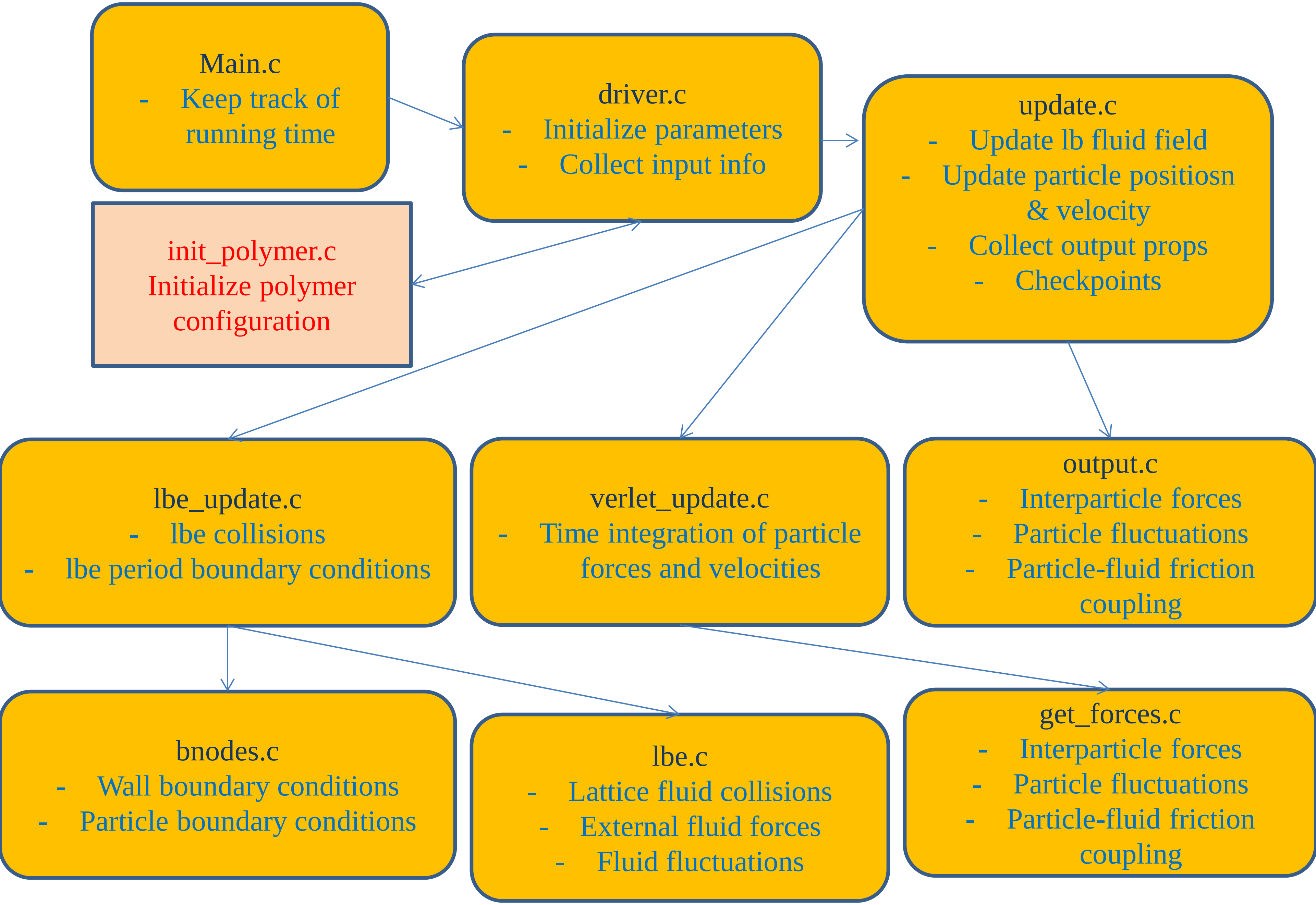
$$< \delta F(x,t) \delta F(x',t') >= 2 k_B T \zeta \delta(x - x') \delta(t - t')$$

Afternoon session 1. Using lattice Boltzmann :

A. How to choose the appropriate parameters.
B. Generate simple shear flow and Poiseuille flow.
C. Compare with the analytical results and error estimation.
D. Coupling LB to a point force monomer and validation of fluid field.

Afternoon session 2. Coupling polymer dynamics with the fluid :

A. How to choose the appropriate parameters for a polymer.
B. Incorporating thermal fluctuations to polymer and fluid.
C. Validating hydrodynamic interactions.
D. Monomers and polymers in Poiseuille flow. (will they migrate ?)



Main.c
- Keep track of running time

driver.c
- Initialize parameters
- Collect input info

update.c
- Update lb fluid field
- Update particle positiosn & velocity
- Collect output props
- Checkpoints

init_polymer.c
Initialize polymer configuration

lbe_update.c
- lbe collisions
- lbe period boundary conditions

verlet_update.c
- Time integration of particle forces and velocities

output.c
- Interparticle forces
- Particle fluctuations
- Particle-fluid friction coupling

bnodes.c
- Wall boundary conditions
- Particle boundary conditions

lbe.c
- Lattice fluid collisions
- External fluid forces
- Fluid fluctuations

get_forces.c
- Interparticle forces
- Particle fluctuations
- Particle-fluid friction coupling

Headers

header.h
Define limit parameters
Include all libraries

lbe.h
Define fluid parameters

struct.h
Define particle datatype and parameters

macro.h
Define macros

func_proto.h
Declare global functions

```
[ylchen@polyui2 LBE]$ make serial
icc   -I/usr/local/intel/mkl/10.0.3.020/include -o aslbe main.o driver.o update.o lbe_update.o bnodes.o bnodes_init.o
implicit_force.o lbe.o modes_write.o lub.o velcs_update.o hs3d.o n_list.o objects_init.o objects_map.o clusters.o
cluster_force.o cluster_update.o cj_grad.o global_sums.o utils.o output.o initlbe.o init_polymer.o ran_num.o get_forces.o
verlet_update.o msg_ser.o  -L/usr/local/intel/mkl/10.0.3.020/lib/32 -lmkl_ia32 -lpthread
ipo: remark #11000: performing multi-file optimizations
ipo: remark #11005: generating object file /tmp/ipo_iccmallY9.o
```

Intel C compiler

Intel math kernel library – vector random number generator

Inputs for the polymer parameters

# of chains
Total # of monomers
Box size

```
[ylchen@polyui2 LBE]$ more init/p_inp.dat
# Num_chain  num_beads  max_x max_y max_z
20 220 20 20 20
# spring_type(0=FENE,1=WLC)   ev_type(0=HS,1=WCA,2=gaussian)
verlet_type(0=1storder,1=2ndorder)  initconfig(1=random,2=stretched)
1  2  2  1
# H_fene  Q_fene   kuhn_length   nks
100  3.0   0.212  19.8
# evcutoff   fric  dt mon_mass kT
5.0 0.483  0.1  1.0 1.74e-3
# mon_fext
0.0 0.0 0.0
# write_props write_config write_fluid
1000 1000 1000
```

Force type: spring,
EV
Integration
Init configuration

Force parameters

Force parameters
MD integration step

Externally applied forces

Period of output

```
[ylchen@polyui2 LBE]$ more init/input_file.dat
 num_cycle      num_step      num_modes      t_lbe
   1                 10000         0            1.0
 wall_flag    backflow_flag   add_noise
   1            0                         2
  rho         vx       vy      vz
  1.0         0.0      0.0     0.0
 mass_fac        vel_fac
   1.0          1.0
  lub_cut        del_hy
   0.0          0.001
  sph_fx         sph_fy          sph_fz
   0.0          0.0           0.0
  ext_fx         ext_fy          ext_fz
   0.0          0.0           0.0
 wall_vx         wall_vy          wall_vz
   0.0          0.01            0.0
   tau          tau_v         tau_g
   1.0          1.0         1.0
  lub_N           lub_T          lub_R
   0.67           0.5            0.43
  seed
 2772301
```

# of checkpoints

# of steps / checkpoint

# of output fluid modes

lbe relaxation time

Wall type: 0 = no walls
            1 = +/- y walls
            2 = +/- yz walls

Backflow

Add_noise: 0 = no noise
            1 = particle noise
            2 = +fluid noise

Fluid density ratio,
velocity ratios

Lubrication cutoff
Hi-lubrication factor

External forces on colloids

External forces on fluid

Wall velocities

Fluid relaxation times for multi-relaxation model

Lubrication force parameters

Random number seed

Run

```
[ylchen@polyui2 LBE]$ ./aslbe > output
[ylchen@polyui2 LBE]$ more output
Running on 1 processors
Begin driver: proc # 0, task 0
Adding 20 chains, 11 monomer / chain, 220 particles
#chain 0
#monomer 0 at (1.159345e+01 4.784072e+00 1.323395e+01)
#monomer 1 at (1.036758e+01 6.397857e+00 1.233972e+01)
#monomer 2 at (9.074085e+00 6.012436e+00 1.094426e+01)
#monomer 3 at (9.058377e+00 7.348096e+00 1.158538e+01)
#monomer 4 at (9.685675e+00 8.463016e+00 1.137528e+01)
#monomer 5 at (9.408609e+00 9.148439e+00 1.237802e+01)
#monomer 6 at (8.566243e+00 1.033286e+01 1.334280e+01)
…

#monomer 216 at (2.670195e+00 2.151291e-01 2.052470e+00)
#monomer 217 at (2.376896e+00 1.898285e+01 3.773407e+00)
#monomer 218 at (1.162502e+00 2.566756e-01 5.158859e+00)
#monomer 219 at (6.259781e-01 1.303433e+00 6.058158e+00)
ncycle 0, numcycle 1
   Total mass        2.88000e+05
   Fluid mass        0.00000e+00
   Particle mass    0.00000e+00
   Map updates        1
   Front location     1
   Max cluster       0
   List updates      13
   Collision rate     0.000e+00
   Max collisions     0.000e+00
   Total momenta      0.00000e+00  0.00000e+00  0.00000e+00
Begin checkpoint       1
End checkpoint         1
Begin data output      1 on   0
End data output        1 on   0
End driver: proc #:0, task 0
Elapsed time on proc   0: 1.065000e+01 (0.000000e+00 1.065000e+01)
```

Output

```
[ylchen@polyui2 LBE]$ dir data
avg_disp2.dat       chain_props.12.dat  chain_props.17.dat  chain_props.3.dat  chain_props.8.dat  monpos.dat
avg_props.dat       chain_props.13.dat  chain_props.18.dat  chain_props.4.dat  chain_props.9.dat  p_out.000
chain_props.0.dat   chain_props.14.dat  chain_props.19.dat  chain_props.5.dat  chk_f.000          properties.dat
chain_props.10.dat  chain_props.15.dat  chain_props.1.dat   chain_props.6.dat  chk_p.000          run_time.dat
chain_props.11.dat  chain_props.16.dat  chain_props.2.dat   chain_props.7.dat  final.config   u.t0000.vtk   bond0.t0000.vtk
[ylchen@polyui2 LBE]$
```

avg_disp2.dat : average monomer MSD, avg velocity
avg_props.dat : average chain MSD, Rg2, stretch
chain_props.dat : chain COM, Rg2, stretch, MSD
chk_f.000: fluid field checkpoint
chk_p.000: chain properties checkpoint
final.config: final chain configuration
monpos.dat: monomer positions and velocities at different times
p_out.000: colloid properties
properties.000: simulation input parameters
run_time.dat: simulation run time
bond0_t????.vtk: paraview configuration files for the polymer
u_t????.vtk: paraview configuration files for the fluid field.

Appropriate Parameters for lattice Boltzmann

Lattice spacing should be bigger than the solvent molecule mean free path

$$\mathrm{Kn} = l_{\mathrm{mfp}} / \, \mathrm{dx} << 1$$

Fluid velocity shall not be faster than the speed of momentum propagation (speed of sound)

$$\mathrm{Ma} = u \, / \, c_{\mathrm{s}} << 1$$

Fluid viscosity must be positive $\eta/\rho = (t_{\mathrm{coll}} - ½) * c_{\mathrm{s}}^2$

LB code parameters: $\rho=36$, $\eta=6$, $c_s^2 = 1/3$

$$t_{\mathrm{coll}} > 1/2$$

Computational time ~ volume => choose the smallest box possible to obtain correct physics (typically L_x at least > 8)

Error ~ $O(dx^2)$

Exercise 1-2: Generate flow between two parallel plates

$$\nabla p = \eta \nabla^2 \mathbf{u} + \mathbf{f}$$

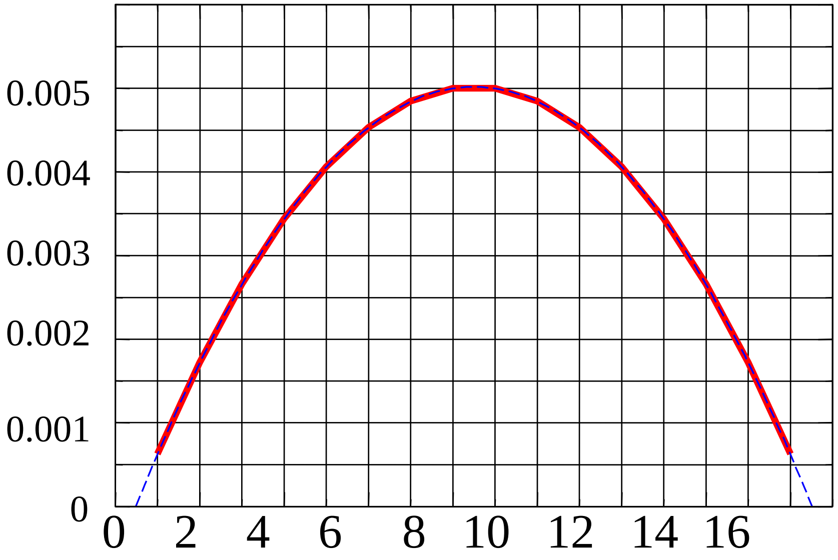
$$u(y) = u_{max}\left((\frac{H}{2})^2 - (y - y_{center})^2 \right)$$

$$u_{max} = \frac{\Delta P / L}{2\mu} = \frac{\rho f_x}{2\mu}$$

Plot the resulting fluid field using paraview or xmgrace

What is the error ?

(at the wall)



Exercise 3: A single monomer (point force) in fluid

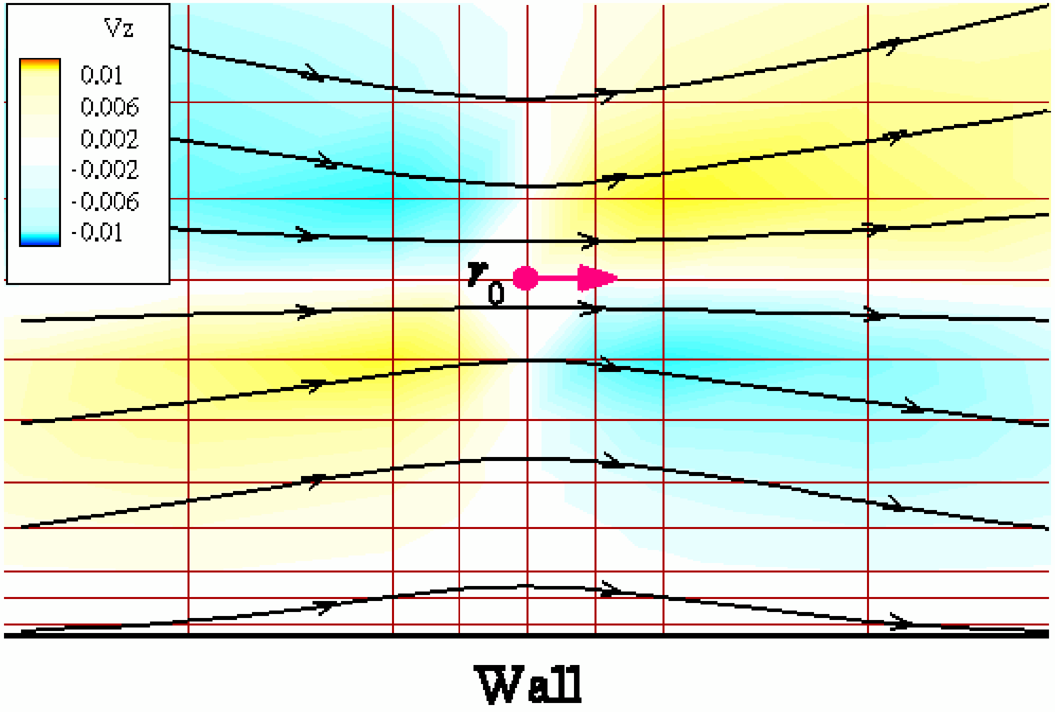
The monomer is fixed at a given position, defined in init/init.config

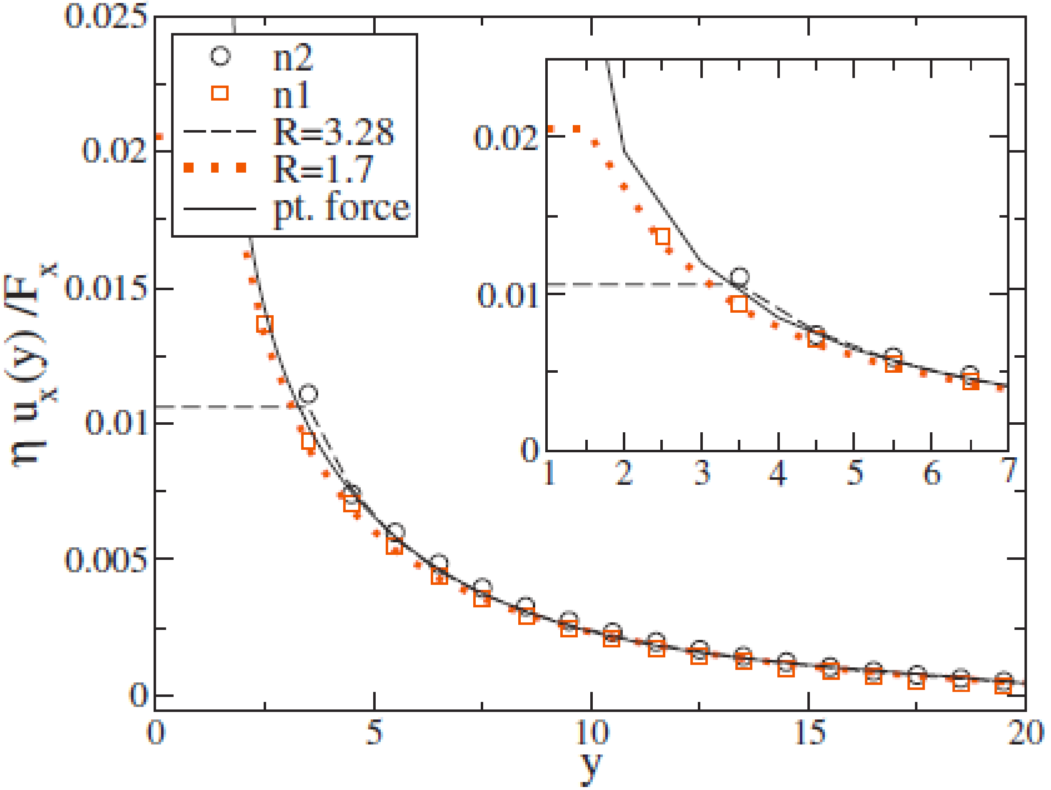
$$\mathbf{u}(\mathbf{r}) = \frac{1 + \mathbf{r}\mathbf{r}/r^2}{8\pi\eta r} \cdot \mathbf{F},$$

Plot the resulting fluid field using paraview or xmgrace

Compare with the analytical solution in Uofr.dat

What happens if the point force is moved towards the wall ?




n2
n1
R=3.28
R=1.7
pt. force
$\eta\, u_x(y)\, /F_x$
y
0.025
0.02
0.015
0.01
0.005
0
0
5
10
15
20
0.02
0.01
0
1
2
3
4
5
6
7

Appropriate Parameters for Polymer model

The distance between monomers should be around 1 lattice spacing

$$\sigma \approx \mathrm{dx}$$

Box size should be > 5 * Polymer radius of gyration

$$\mathrm{L} > 5\, R_{\mathrm{g}}$$

Run time should be much longer than the polymer relaxation time to sample equilibrium properties

Exercise 5a: A single monomer (point force) in Poiseuille flow

The monomer is fixed at a given position, defined in init/init.config

Plot the resulting particle trajectory using xmgrace

Exercise 5b: A single dumbbell (point force) in Poiseuille flow

A dumbbell is a polymer with only two beads

The dumbbell position can be defined in init/init.config

Plot the resulting dumbbell trajectory using xmgrace

Try different initial positions and orientations

Exercise 6ab: Trajectories a large number of monomers and dumbbells in Poiseuille flow

Generate random initial particle positions using

initconfig = 1 in p_inp.dat

Plot the resulting dumbbell trajectories using xmgrace

```
tail --lines=100 --quiet data/chain_props.*.dat >> combined_chainprops.dat
```