Take-home message

- There's no free lunch in coarse-graining a system
- Can multiscale simulations overcome L, T challenges?
- How to coarse-grain atoms and molecules

Why coarse-grained simulations?

If QM or Molecular Dynamics is so good, why do anything else?

Several reasons: don't know the force fields, system considered is too large or too slow, don't need picosecond accuracy, interested in genera chemical features not specific chemicals

Although it might be nice to simulate a billion Lennard-Jones particles, interacting via a hugely complicated force field for10 minutes of real time - *it ain't gonna happen*. We have to choose an accuracy we can live with and see how to attain it.

Advantages of coarse-grained simulations like DPD for complex fluid simulations:

Very cheap computationally

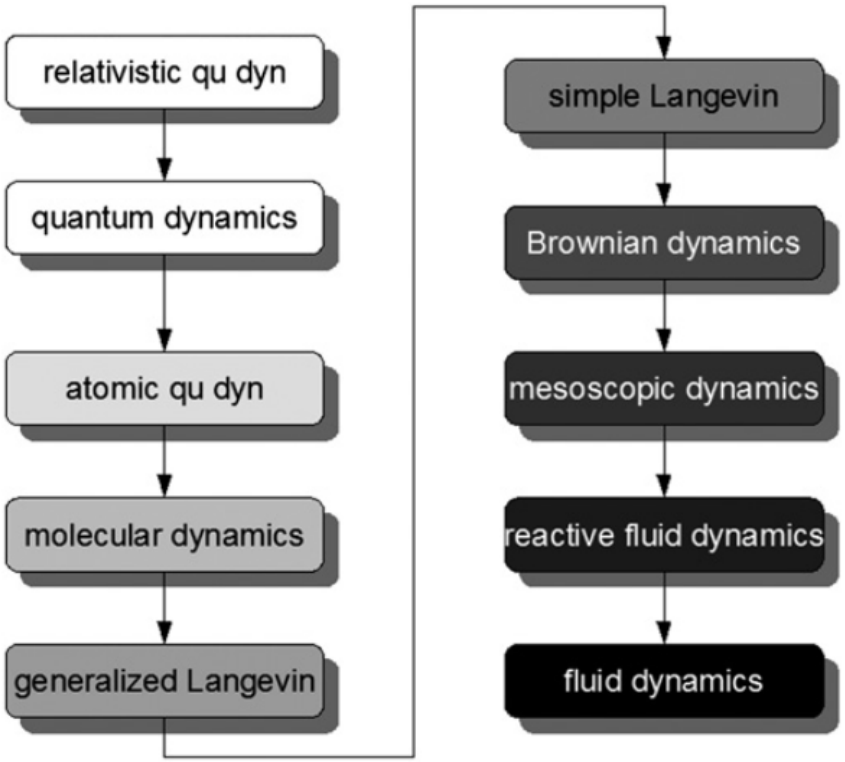
Very forgiving of non-equilibrium initial states and force law details

Large system sizes (microns) and long times (milliseconds) accessible whilst retaining near-molecular resolution

Provides insight into dynamics on scales well beyond molecular, e.g., long wavelength membrane fluctuations, easy to visualize

What are coarse-grained simulations?

Setting up a simulation requires asking questions about what exactly is "the system" we want to study, what are its fundamental entities, what do we want to learn, and how accurate do we need the results (most accurate is not always desirable):



We do not simulate a real system, but only a model of a real system; we first construct a model (particles + forces) and, second, solve it on a computer.

Fig. 4 Hierarchy of models for simulation,[17] ranging from very detailed (white background) to very coarse-grained (black background). Each level has its own description of the reduced system and its own simulation method. Each higher level loses some details of the preceding level.

H. J. C. Berendsen Faraday Discussions 144:467 (2010)

What about multiscale simulations on multicore machines?

See the paper by Yu et al. on today's moodle page for a multiscale model of the SARS-CoV-2 virus

Blue Brain Project

Characteristics of single-scale simulations

a scale is a range in length and time

where distinct physical processes dominate the behaviour of matter

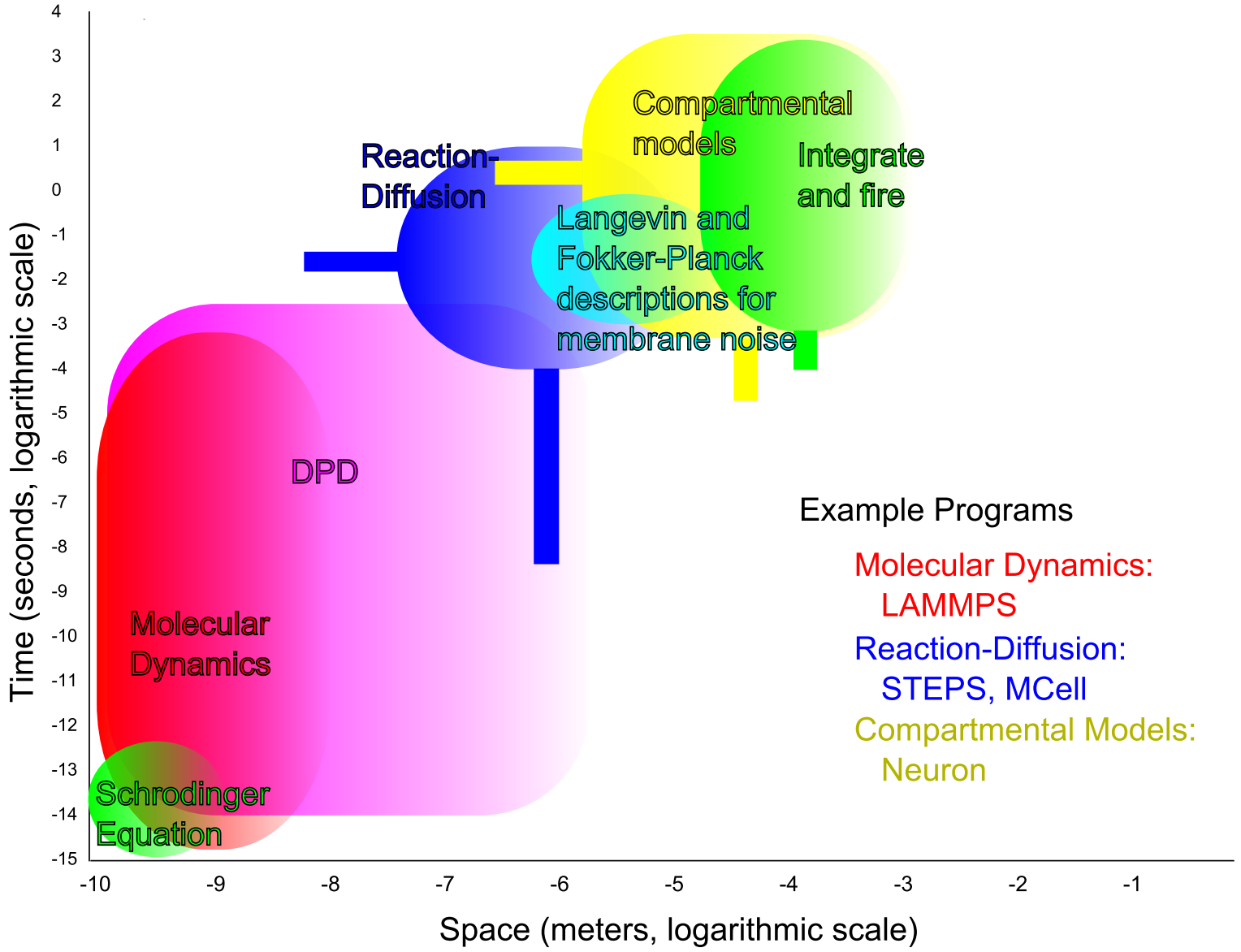
e.g., Brownian motion is important for bacteria but not for cars; inertia is irrelevant for protein interaction networks; lipid diffusion is unimportant for Action Potentials, ...

In single-scale simulations:

- all dynamics outside a limited space-time range are clamped or equilibrated; i.e., much slower or faster than the processes of interest
- dynamics involves signals whose speeds are all comparable (except MC)
- lumped parameters obscure details from finer scales
- energy dissipation is ignored: systems relax to equilibrium or have no energy cost
- cannot push them beyond a characteristic scale as execution time scales as L^d with $d \geq 3$

Blue Brain Project

Simulation techniques relevant to brain modelling



Dan Keller, EPFL

Claim: multiscaling may work for some ranges of L, T but fails at atomic scale

Remember how many water molecules are in a neuronal synaptic spine
10**9 and 10**15 time steps using 1 fs step size.

Blue Brain Project

Computational gain

Goal: to simulate 1 sec of activity in ~1 sec of real time at *all scales* in a multi-scale simulation

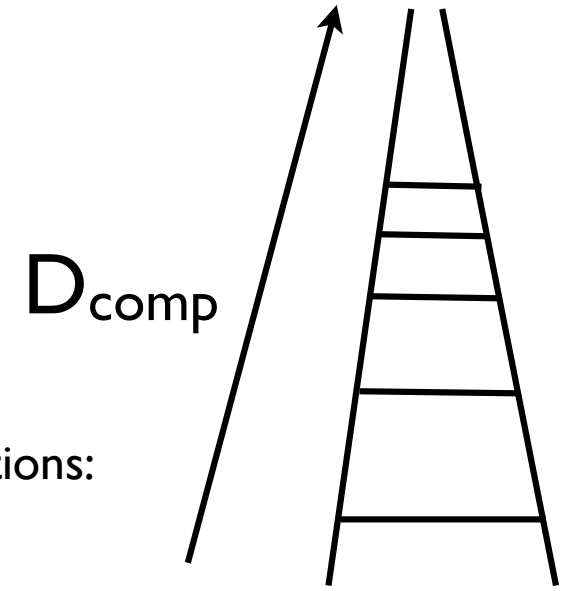
In order to simulate 1 sec per sec at any given level (L,T), we must update

D_{comp} ~ (Typical volume / vol of smallest unit) x (1 / δt)

degrees of freedom in 1 second of REAL TIME. This is a computational distance.

Define a computational velocity (cp. FLOPS) for any scale:

V_{comp} (N) = No. of dof updated per second using N procs



In order to achieve 1 sec of activity per sec, we must satisfy two conditions:

$$D_{comp} / V_{comp}(N) \leq 1 \quad \text{and} \quad V_{comp}(1) \times \delta t \geq 1$$

We can approach these conditions by reducing D_{comp} or increasing V_{comp} e.g., by increasing number of processors N or the speed of the code

Blue Brain Project

Gain mismatch between simulation scales

NEURON's differential eqns. have

D_{comp} ~ 300 x (1 ms / 0.05 ms) = 6000

V_{comp} ~ 1 GFLOP

$D_{comp}/V_{comp} \sim 6.10^{-6}$

$V_{comp} \times \delta t \sim 10^9 \times 0.05$ ms / 1 sec = 5.10^7

NCC has ~ 10^4 neurons

1 neuron has ~ 10^4 synapses

DPD has

D_{comp} ~ (1 µm³ / 1 nm³) x (1 µs / 0.01 ns) = 10^{14}

V_{comp} ~ 10^{10} bead.steps / cpu-day

$D_{comp}/V_{comp} \sim 10^4$

1 synapse has 1 µm³ of space ~ 10^9 water molecules

$V_{comp} \times \delta t \sim 10^{10} \times 0.01$ ns / 86,400 sec = 10^{-6} << 1

MD has

D_{comp} ~ (100 nm)³ / (0.1 nm)³ x (1 µs / 0.01 ps) = 10^{17}

V_{comp} ~ 10^{10} bead.steps / cpu-day

$D_{comp}/V_{comp} \sim 10^7$

$V_{comp} \times \delta t \sim 10^{10} \times 0.01$ ps / 86,400 sec = 10^{-9} << 1

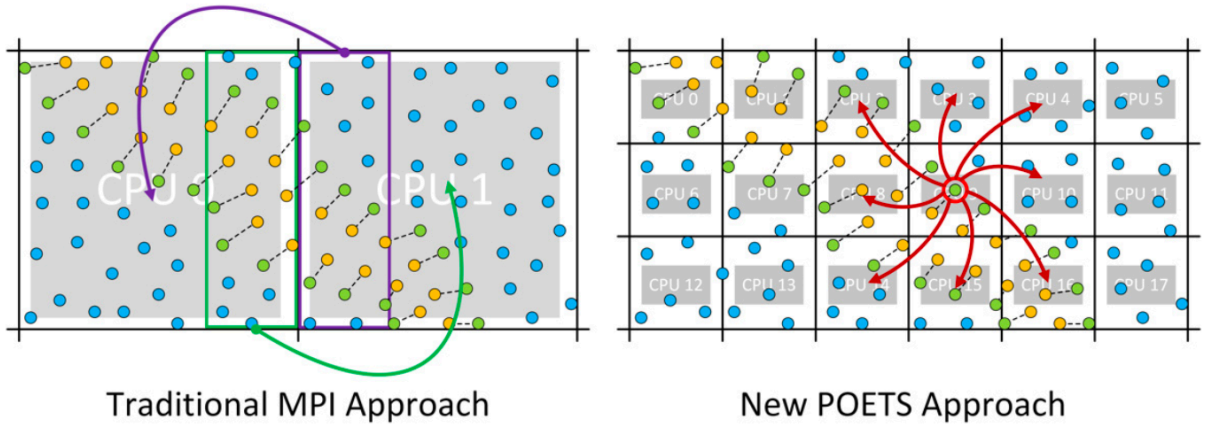


Figure 3. Contrast between the traditional MPI-based parallelization approach (left panel) in which each CPU owns a volume of space containing hundreds or thousands of beads, and the POETS approach (right panel) where space is sub-divided down to unit-volume cells containing only 3 or 4 beads, each of which is managed by one light-weight CPU.

POETS = partially-ordered event-triggered systems

https://poets-project.org/

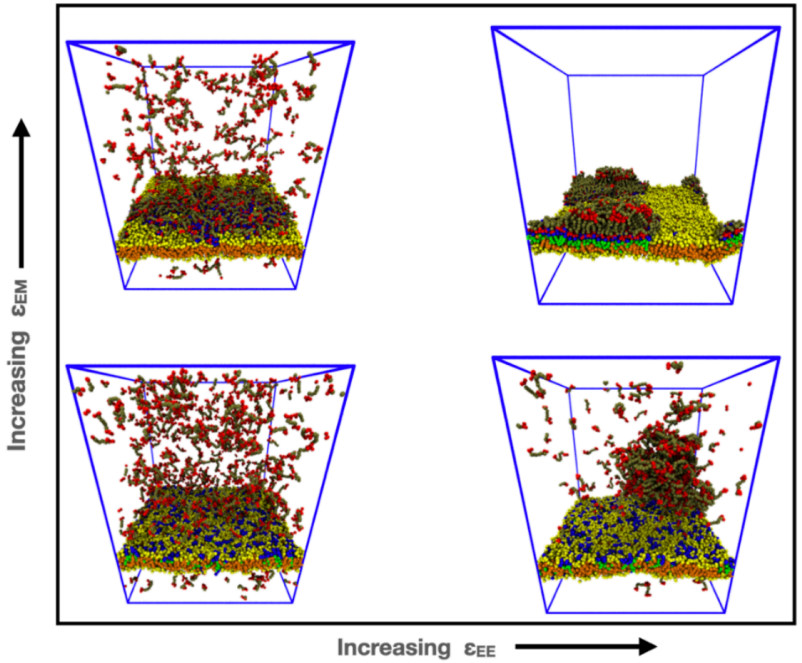


Figure 8. Morphology diagram in the (ε_{EE}, ε_{EM}) plane showing the equilibrium states of the IDPs and membrane, as their relative interactions are varied.

Shillcock et al. , Coupling Bulk Phase Separation of Disordered Proteins to Membrane Domain Formation in Molecular Simulations on a Bespoke Compute Fabric, *Membranes* **12**: 17 (2021)

Coarse-grained simulation types

All based integrating some form of Newtonian equations of motion

$$m.dv/dt = F \qquad \text{MD}$$

$$m.dv/dt = F^C + F^D + F^R \qquad \text{DPD}$$

$$m.dv/dt = F^C - m\gamma.v + \sqrt{(2m\gamma k_B T)}.\zeta(t) \qquad \text{Langevin}$$

$$0 = F^C - \gamma.v + \sigma.\zeta(t) \qquad \text{Brownian}$$

The difference lies in what constitutes a "particle" and how complex the forces are.

In MD, the particles are atoms but in coarse-grained techniques, the particles are groups of atoms, molecular groups, even molecules.

In these cases, once the particles are defined (mass, radius), and the forces are given (bonds, non-bonded, electrostatics), we integrate Newton's 2nd law as in MD.

Allen, MP, and Tildesley, DJ, Computer Simulation of Liquids, Clarendon Press, Oxford, 1987

Frenkel, D and Smit, B, Understanding Molecular Simulation, Academic Press, 2002

Berendsen, HJC, Faraday Discussions 144:467 (2010)

Coarse-graining atoms and molecules

Molecular Dynamics is accurate at atomic length scales, but sometimes we want to simulate far above this scale, e.g., membranes and vesicles.

The process of replacing atoms by groups of atoms particles is called *coarse-graining*. It has two advantages

several atoms ⇒ one bead so fewer d.o.f to integrate

Lennard-Jones forces ⇒ softer forces so larger Δt

This means cheaper, faster simulations!

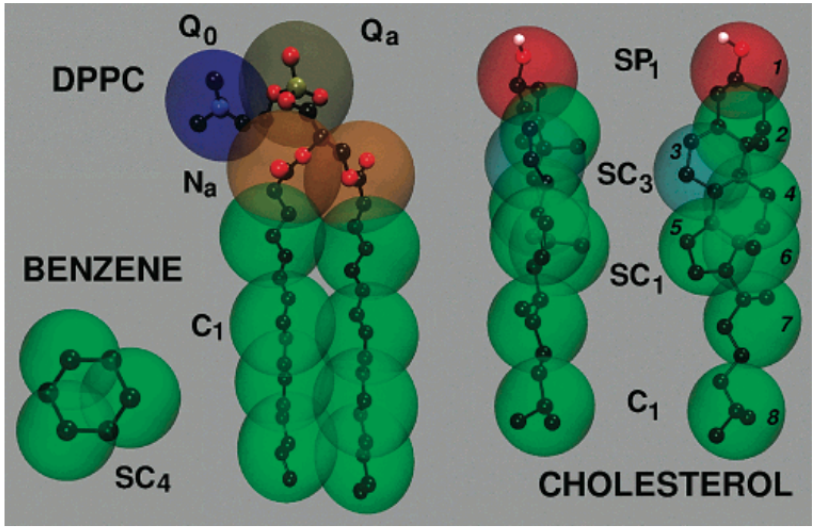


Figure 1. Mapping between the chemical structure and the coarse grained model for DPPC, cholesterol, and benzene. The coarse grained bead types which determine their relative hydrophilicity are indicated. The prefix "S" denotes a special class of CG sites introduced to model rings.

Marrink, S. J. J. Phys. Chem. B 111:7812 (2007)

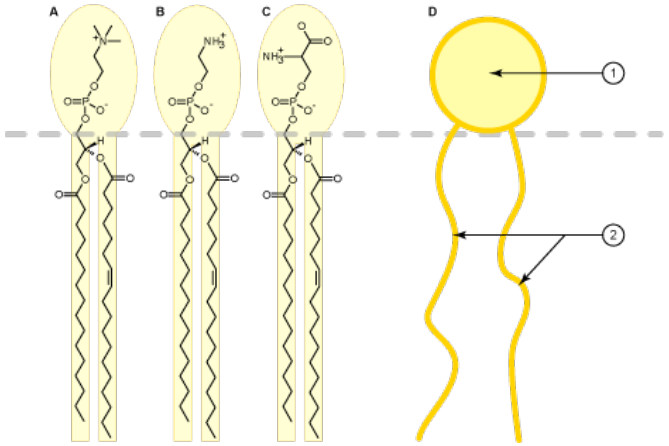
Popular coarse-graining schemes
(in order of decreasing resolution) are:

United atom - include H atoms in definition of C atoms, etc.
Coarse-grained MD - replace methyl group by a C3 particle, etc
Dissipative Particle Dynamics - lump atomic groups into fluid particles that carry momentum
Implicit solvent MD - replace water molecules by special potentials that mimic hydrophobic effect
Brownian Dynamics - particles of interest are much larger than water, so replace water molecules by an implicit representation in the force field

For coarse-graining lipids, a good review is: Bennun et al., *Chem and Physics of Lipids* **159:**59-66 (2009)

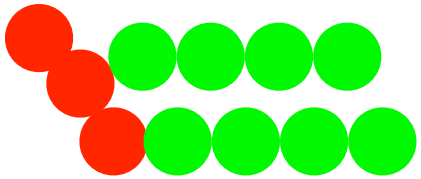
Lipids in DPD simulations

As an example: consider a dimyristoylphosphatidyl choline (DMPC) lipid bilayer and measure its material properties. This is a (very simplified) model of the plasma membrane.



1 = Head particle

2 = Chain of tail particles



$H_3(T_4)_2$

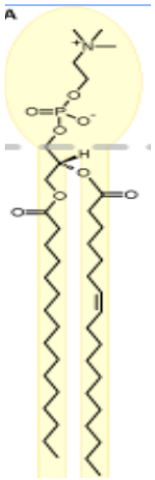
For DMPC and lipids that differ only in tail length (lauryl, myristoyl, palmitoyl, stearoyl, ...). We find the relation that each DPD tail bead represents 3-4 methyl groups. So cgDMPC has ~11 beads. Ambiguity comes from the fact that a DPD *bead* is a rather fuzzy concept, based on a volume of material, and may not divide neatly into a hydrocarbon chain's number of monomers.

Groot and Warren, J. Chem. Phys. 107:4423 (1997) and Marrink et al. J. Phys. Chem. B 111: 7812 (2007)

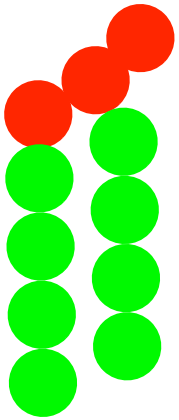
Headgroup must be large enough to balance the cross-sectional area of the tails (Israelachvili's packing param. ~ 1): 3 or 4 head beads is sufficient for a tail of length 4 - 6.

Shillcock, JC, and Lipowsky, R, J. Chem. Phys. 117:5048 (2002)

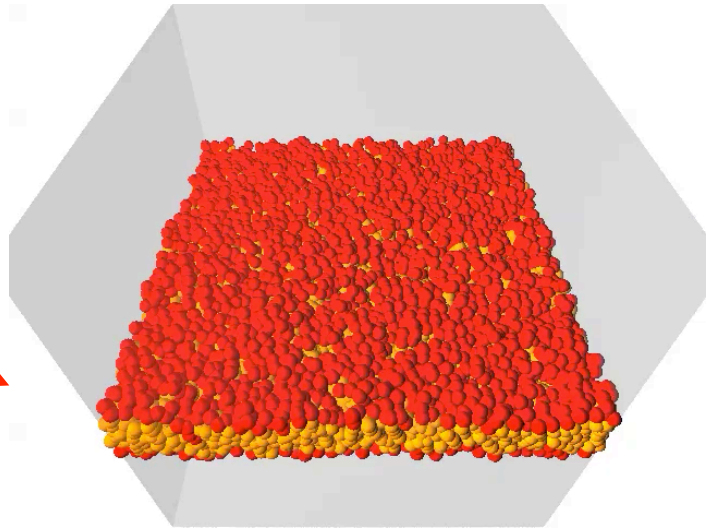
Coarse-graining a lipid membrane



How?



How?



Real lipid

cg lipid

Lipid bilayer

Headgroup area ~ 1 nm^2

Tail length ~ 0.154 + 0.126*n nm
where n = # carbons in tail

We need M, L, T

Bead size r_0 ~ 1 nm

How many CH_2 per tail bead?
- not known a priori, but we can guess ~ 2-5.

1:1 would be atomistic

All:2 would be a dimer H-T

Box size ~ 32 r_0

How many lipids?
- not known a priori

trial and error from simulations

Reduced units for lipid bilayers

Typical lipid tail length is ~ 2 nm for DMPC
Bilayer width ~ 4-5 nm
Area per molecule ~ 0.65 nm^2
Assume that the mass of all bead types is the same

So, a simulation box $(32.r_0)^3$ where r_0 is the diameter of one lipid bead, and a (dimensionless) bead density of $\rho=3$ contains $N = 3.32^3 = 98304$ beads.

For lipid bilayers, we typically use the area per lipid to determine the number of lipid molecules. Given an experimental value of 0.65 nm^2 we calculate:

$N_{lipid} = 2.((32\ r_0\ nm)^2 / 0.65\ nm^2)$ molecules

Initially choose $\pi\ (r_0/2)^2 \sim 0.65\ nm^2$, so $r_0 \sim 0.91$ nm and $N \sim 2609$ or $A/Nr_0^2 \sim 0.8$

For a lipid bilayer in equilibrium, we expect the surface tension to be zero. We adjust the box size or number of lipids until the simulation gives zero tension, and then extract the equilibrium value of a_{Lipid} for the bilayer. That is, we obtain $a_{Lipid} = A/(N.\ r_0^2)$ from the simulation and from this we can extract an accurate value to r_0. If $a_{Lipid} = 1.26$, say,

$r_0 = \sqrt{(A/N / a_{Lipid})} = \sqrt{(0.65 / 1.26)} \sim 0.72$ nm and $N_{lipid} = 1633$ in equilibrium

We now have a length scale, but what about a time scale?

An obvious process involving time is the diffusion of the lipids in the membrane. A dimensionless form of the diffusion constant is:

$$\text{Dimensionless diffusion constant: } D' = (D.\, t_0/r_0^2)$$

We measure D' in the simulation, so if we know D from experiment and r_0, we can derive a value for t_0. This gives us a natural time-scale for the motion of lipids in the membrane.

A typical lipid diffusion constant is 0.1 - 10 μm^2/sec (H. Gaede and K. Gawrisch, **Biophys. J.** 85:1734 (2003))

Suppose in a lipid bilayer simulation we find D' ~ 0.01 and we have estimated r_0 = 0.72 nm from the membrane's area/lipid.

A typical time-scale for the lipids in the membrane is then (using D ~ 1 μm^2/sec):

$$t_0 = 0.01.(0.72.10^{-9})^2 / 10^{-12} \sim 5 \text{ ns}$$

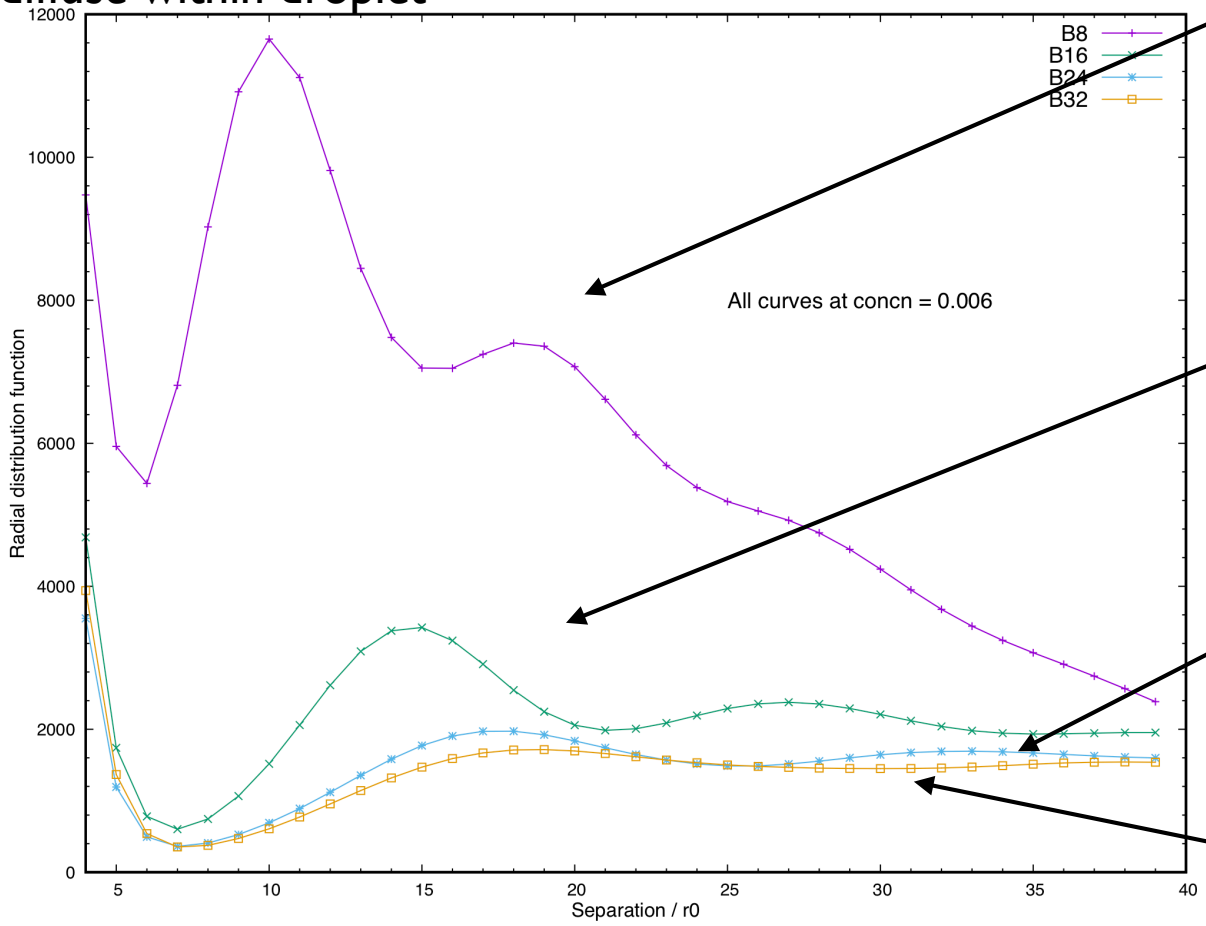
and, recalling that t_0 is the self-diffusion time, a bead will diffuse its own size in this time. A typical integration time step will then be 0.01 - 0.02.t_0

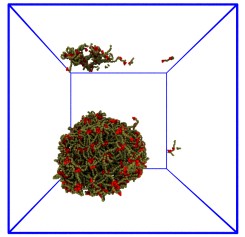
NB. There may be other time-scales in the system NOT described by this, e.g., lipid flip-flop between monolayers and solvent transport across the bilayer: need judgement here.

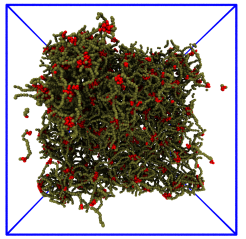
Another example

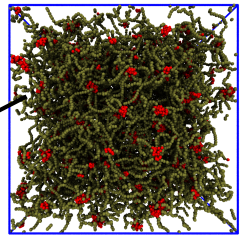
Radial distribution function reveals the geometric structure of the dense phase: put in molecules, get out spatial structure of supramolecular aggregate.

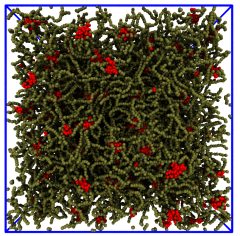
Speed of enzymatic reactions depends on porosity and ability of client proteins to diffuse within droplet











DPD algorithm: Basics

Particle based: N particles in a box, specify $\mathbf{r}_i(t)$ and $\mathbf{p}_i(t)$, i = 1…N.

Mesoscopic: Each particle is a small volume of fluid with mass, position and momentum

Newton's Laws: Particles interact with nearby particles; integrate Newton's law F = ma

Three types of force exist between all particles:

- Conservative $\mathbf{F}^C_{ij}(\mathbf{r}_{ij}) = a_{ij}(1 - |\mathbf{r}_{ij}|/r_0)\mathbf{r}_{ij} / |\mathbf{r}_{ij}|$
- Dissipative $\mathbf{F}^D_{ij}(\mathbf{r}_{ij}) = -\gamma_{ij}(1 - |\mathbf{r}_{ij}|/r_0)^2(\mathbf{r}_{ij}.\mathbf{v}_{ij})\,\mathbf{r}_{ij} / |\mathbf{r}_{ij}|^2$
- Random $\mathbf{F}^R_{ij}(\mathbf{r}_{ij}) = \sigma_{ij}(1 - |\mathbf{r}_{ij}|/r_0)\Gamma_{ij}\mathbf{r}_{ij} / |\mathbf{r}_{ij}|$

forces are soft, short-ranged (vanish beyond r_0), central, pairwise-additive, and *conserve momentum locally*. Note that γ_{ij} and σ_{ij} must be related by $\sigma_{ij}^2 = 2\gamma_{ij}k_BT$ (see Espagnol and Warren 1995)

(1853 citations) P. J. Hoogerbrugge and J. M. V. A. Koelman, Europhysics Letters **19**:155 (1992)
(1366 citations) P. Espagnol and P. B. Warren Europhysics Letters **30**:191 (1995)
(1994 citations) R. D. Groot and P. B. Warren J. Chem. Phys. **107**:4423 (1997)

DPD algorithm: Forces

Conservative $\mathbf{F}^C_{ij}(\mathbf{r}_{ij}) = a_{ij}\,(1 - r_{ij}/r_0)\,\mathbf{r}_{ij} / r_{ij}$

Dissipative $\mathbf{F}^D_{ij}(\mathbf{r}_{ij}) = -\gamma_{ij}\,(1 - r_{ij}/r_0)^2\,(\mathbf{r}_{ij}.\mathbf{v}_{ij})\,\mathbf{r}_{ij} / r_{ij}^2$

Random $\mathbf{F}^R_{ij}(\mathbf{r}_{ij}) = \sqrt{(2\gamma_{ij}\,k_BT)}\,(1 - |\mathbf{r}_{ij}|/r_0)\,\Gamma_{ij}\,\mathbf{r}_{ij} / |\mathbf{r}_{ij}|$

Conservative force, a_{ij}, gives particles an identity, e.g. hydrophobic

Dissipative force, γ_{ij}, destroys relative momentum between pairs of interacting particles

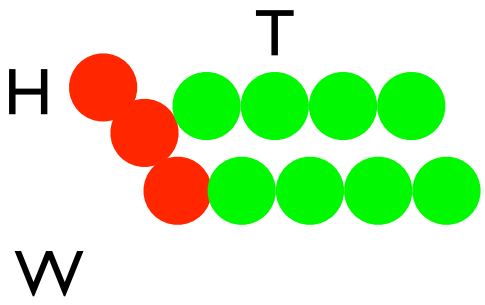
Random force, σ_{ij}, creates relative momentum between pairs of interacting particles:

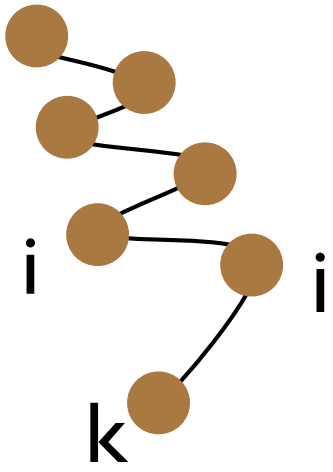
$\langle\Gamma_{ij}(t)\rangle = 0 \qquad \langle\Gamma_{ij}(t)\,\Gamma_{ij}(t')\rangle = \delta(t-t')$

$a_{ij}(t) = a_{ji}(t) \quad \gamma_{ij}(t) = \gamma_{ji}(t) \quad \Gamma_{ij}(t) = \Gamma_{ji}(t)$ which we implement as: $\Gamma_{ij} \sim N(0,1) / \sqrt{dt}$

The dissipative and random forces act as a thermostat keeping the system temperature constant on average (canonical ensemble). This thermostat is *independent* of the form of the conservative force and, in fact, the DPD thermostat is sometimes used with MD forces - Soddemann et al., PRE 68:046702 (2003). Its usefulness stems from the fact that it conserves momentum locally, so hydrodynamic modes of the fluid are preserved.

DPD algorithm: Bond forces





a_{ij}	H	T	W
H	30	35	30
T	35	10	75
W	30	75	25

Grafmüller et al. Biophys. J. 96:2658 (2009)

$$\mathbf{F}(\mathbf{r}_{ii+1}) = -k_2(|\,\mathbf{r}_{ii+1}\,| - l_{i0})\,\mathbf{r}_{ii+1}/|\,\mathbf{r}_{ii+1}\,|$$

Hookean spring parameters: $k_2 = 128\ k_BT/r_0^2$, $l_{i0} = 0.5$ These parameters are chosen to keep the lipid tail length on average at the desired value.

$$V(ijk) = k_3(1 - \cos(\phi_{ijk} - \phi_0))$$

Chain bending stiffness parameters: $k_3 = 15\ k_BT$, $\varphi_0 = 0$

Chain stiffness is chosen to ensure lipids don't interdigitate much.

DPD algorithm: Integration

Most common: velocity-Verlet scheme of Groot and Warren - J. Chem. Phys. 107:4423 (1997).

1. Update positions of all particles: $r(t+dt) = r(t) + p(t).dt + 0.5.F(t).dt^2$
2. Calculate intermediate velocities: $p'(t+dt) = p(t) + \lambda.F(t).dt$
3. Update forces on all particles : $F(t+dt) = F(r(t+dt), p'(t+dt))$
4. Update momenta of all particles : $p(t+dt) = p(t) + 0.5*dt*(F(t) + F(t+dt))$

Because we set m = 1, velocity (v) = momentum (p).

Note that λ is a heuristic parameter, typically ~0.5, to take the stochastic force into account. It is used to estimate the effect of the time-varying force over the course of a time-step. It is needed because no matter how small the time-step is, the random force ought to change during the step, i.e., the stochastic force is not smoothly-constant as the discretized equations of motion assume.

Because of the stochastic force, we have to use special integration schemes for the equations of motion, e.g., the velocity-Verlet scheme above. These are not needed for MD.

Choosing DPD interaction parameters

The dissipative and random forces form a thermostat that does not change when simulating different systems. We'll ignore it, but see Groot and Warren (1997) for details. If we have molecules like lipids or polymers, how do we set values for the Hookean springs that tie them together? or bond bending stiffness? Basically, it's Trial and Error!

But the conservative interaction parameters a_{ij} can be set from thermodynamics.

What is the equation of state of the one-component DPD fluid (= water)?

Recall an ideal gas: $PV = Nk_BT$ or $P = \rho k_BT$

Van der Waal's gas: $P = \rho k_BT/(1-\rho b) - a\rho^2$

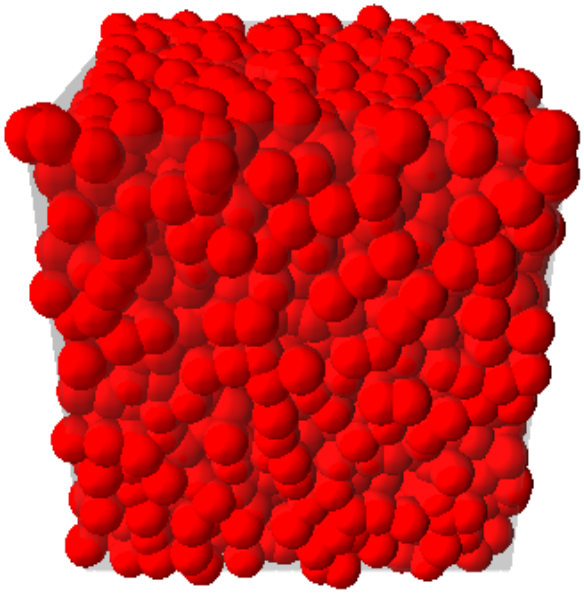
We measure the pressure of the fluid as a function of density and fix the value of the single parameter a_{ww}.

$L_X = L_Y = L_Z = 10r_0$

ρ = 3 - 10 beads/volume

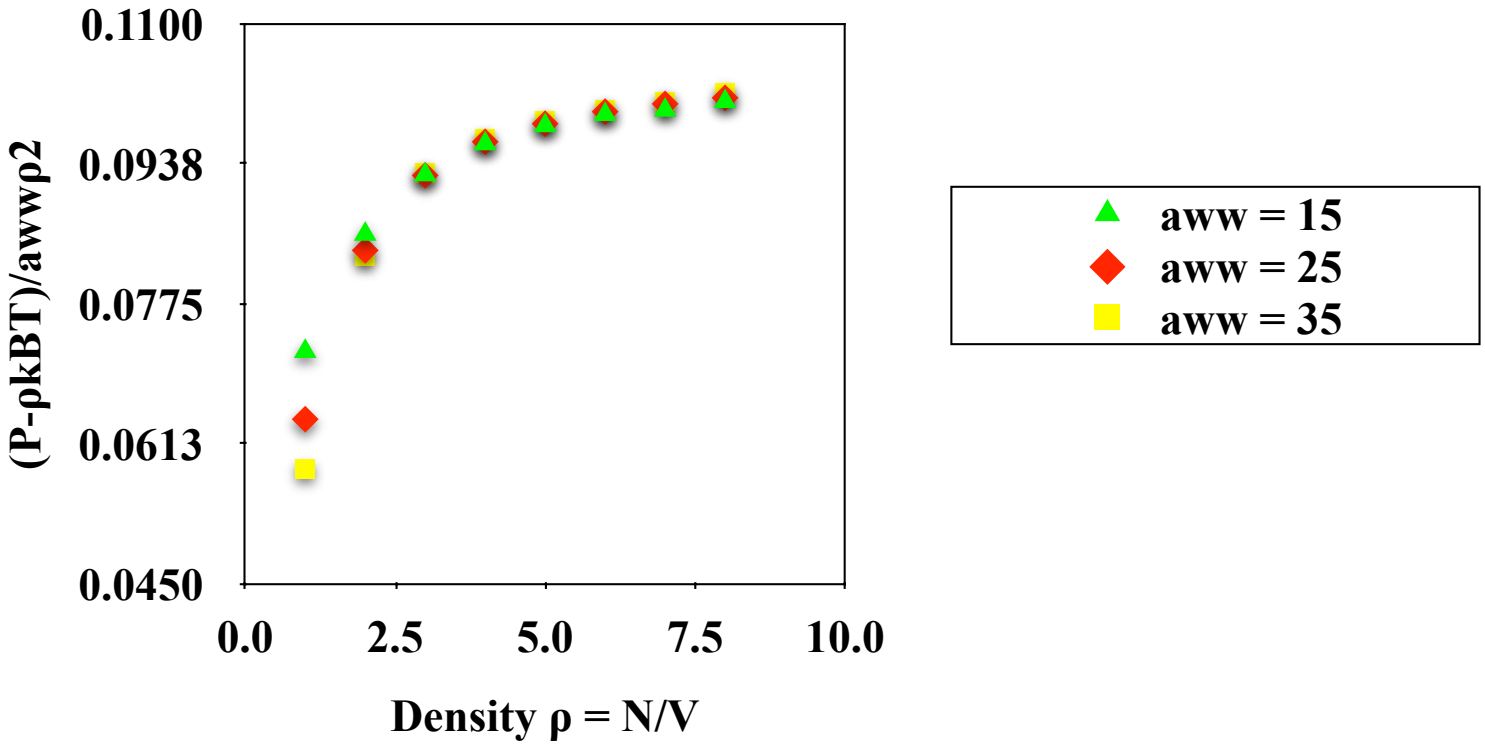
N = 3000 - 10000 beads

$a_{ww} = 25$



Equation of state for DPD fluid

Plot the excess pressure ($P - \rho k_B T$), scaled by the conservative repulsion parameter, a_{ww}, and density.



From the simulated DPD equation of state, we find numerically as the density increases:

$P = \rho k_B T + \alpha\, a_{ww}\, \rho^2$ where $\alpha = 0.10 \pm 0.01$

Compressibility of DPD "Water"

The isothermal compressibility of water is defined as: $\kappa^{-1} = (dp/d\rho)_T / k_BT \sim 15.9835$ at room temperature, and this fixes the conservative self-repulsion parameter "a_{WW}" for a single-component fluid if we want it to have the compressibility of water.

If we differentiate the EOS above for the DPD fluid, we get

$\kappa^{-1} = 1 + 2\alpha a\rho/k_BT \sim 16$

giving $a_{WW} = 75\ k_BT/\ \rho$. Most DPD simulations use a bead density of $\rho = 3$, so $a_{WW} = 25\ k_BT/r_0$

So, the single-component DPD fluid density is a free parameter as long as the beads are dense enough to interact and not have "holes" in the fluid.

Higher densities mean more interactions, so we choose the lowest value that is consistent with the assumed EOS.

But what if we have a mixture of fluids - how do we choose the off-diagonal parameters a_{ij}?

Off-diagonal conservative forces

The off-diagonal elements of the force matrix set the repulsion or attraction of fluid elements of different types when they interact which is related to their solubility.

Note that all DPD forces are repulsive: the self interactions are repulsive because they represent the compressibility (resistance to being compressed) of a pure fluid, and the off-diagonal elements are repulsive because they represent the solubility of mixtures which are usually less cohesive than the pure fluid.

This is the price paid for having no hard core repulsion - you cannot have strongly attractive forces or the fluid will collapse on itself.

a_{ij}	H	T	W
H	30	35	30
T	35	10	75
W	30	75	25

Flory-Huggins/DPD equivalence

Groot and Warren (1997) found a correspondence between the soft DPD fluid and the Flory-Huggins theory of polymer mixtures. FH theory is a mean-field theory of the free energy of a polymer mixture that predicts a phase separation for sufficiently repulsive polymers.

The free energy of a mixture of two polymer types depends on their M.wt or length N_A, N_B and volume fractions φ_A, φ_B

It has the form:

entropy

$$\beta F = (\varphi_A/N_A)\ln(\varphi_A) + (\varphi_B/N_B)\ln(\varphi_B) + \chi\,\varphi_A\,\varphi_B$$

energy

where $\beta = 1/k_BT$ and F is the free energy

N_i = No of monomers in polymers of type i = A, B

φ_i = *Volume fraction* of polymer type i and $\varphi_A + \varphi_B = 1$

χ = *Mixing* parameter or *repulsion* parameter ~ how much the polymer like/dislike each other

The free energy of a DPD fluid of two components A, B is:

$$\beta F_V = \rho_A/N_A \ln(\rho_A) + \rho_B/N_B \ln(\rho_B) - \rho_A/N_A - \rho_B/N_B + \beta\alpha(a_{AA}\,\rho_A^{\,2} + 2a_{AB}\,\rho_A\,\rho_B + a_{BB}\,\rho_B^{\,2})$$

where $\beta = 1/k_BT$, $\alpha \sim 0.1$ from simulations

ρ_i = Number density of particles of type i (= φ_A)

$a_{AA} = a_{BB}$= like-particle conservative force parameter

a_{AB} = unlike-particle conservative force parameter

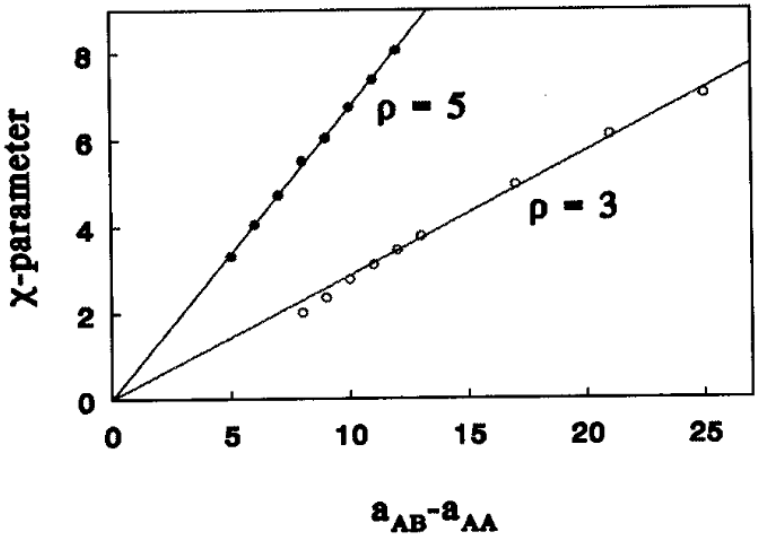


Fig. 7 in Groot and Warren, 1997

Now let $x = \rho_A/(\rho_A + \rho_B)$ and assume that $\rho_A + \rho_B \sim$ constant then:

$$\beta F_V \sim x/N_A \ln(x) + (1-x)/N_B \ln(1-x) + \chi\, x\, (1-x) + \text{const.}$$

yielding the relation: $\chi = 2\,\beta\alpha(a_{AB} - a_{AA})(\rho_A+\rho_B)$, between the Flory-Huggins parameter and the relative DPD cross interaction $a_{AB} - a_{AA}$.

As χ is known from experiment this allows DPD to be calibrated for polymer mixtures.

Polymeric Fluid Mixtures: χN Parameter

Experimental χN values are tabulated for different polymers, so we have a way of setting the unlike-bead interaction parameters a_{ij} for any pair of (immiscible) DPD polymers at a given density.

$$\chi = 2\,\beta\alpha(a_{AB} - a_{AA})(\rho_A+\rho_B)$$

Alternatively, the a_{ij} are free parameters whose values are varied until the simulated system has some correct physical property, e.g., interfacial surface tension.

For lipids, the key parameters are the tail bead / water bead repulsion (the hydrophobic effect), and the head-head, head-tail repulsion parameters as these largely determine the equilibrium A/N of a lipid bilayer.

We have many properties of lipid bilayers that we can use to calibrate these parameters.

We aim to use a few properties to calibrate the DPD parameters, and then predict other properties using the simulations. Obviously, we need more experimental properties than we have parameters!

Other methods of fixing a_{AB} are:

Travis et al. J. Chem. Phys. 127: 014019 (2007)
Sepehr and Paddison, Chem. Phys. Lett. 645:20-26 (2016)