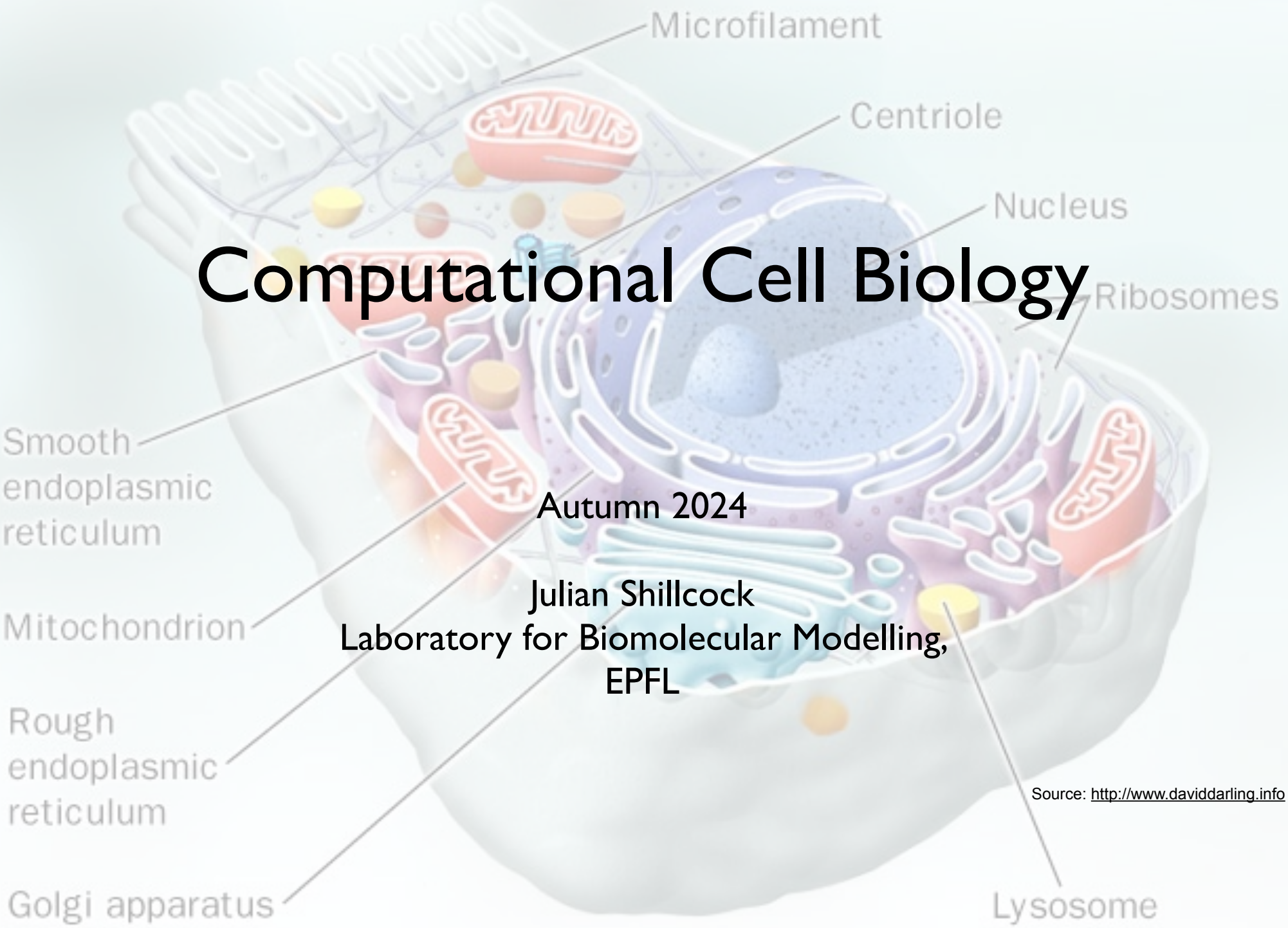


Computational Cell Biology

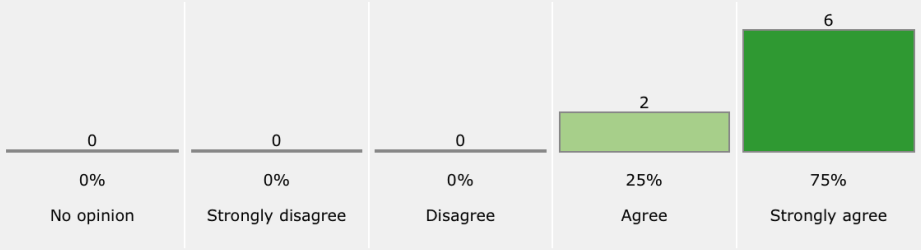
Autumn 2024

Julian Shillcock
Laboratory for Biomolecular Modelling,
EPFL

Source: http://www.daviddarling.info

Indicative feedback

The running of the course enables my learning and an appropriate class climate



Remarks

[3 remarque(s)]

- During the exercise sessions, it feels bad when the whole class is getting bottlenecked by a single person's PC. Maybe we could make more people do more simulations so we have enough data to continue and finish in time. Very pleasant class otherwise.
- Good course
- The workload for this class is not adapted to its credits weight. Having homework, tests, projects, presentations is a bit too much, even if they are supposed to be small. Apart from that, the class is very interesting and the lectures are good.

Sustainable simulations

Simulations aren't free!

How much does the electricity to run a simulation for, say, 1 hour, cost?

What about 1 day? 1 week? 1 month?

Is it possible to reduce the energy consumption?

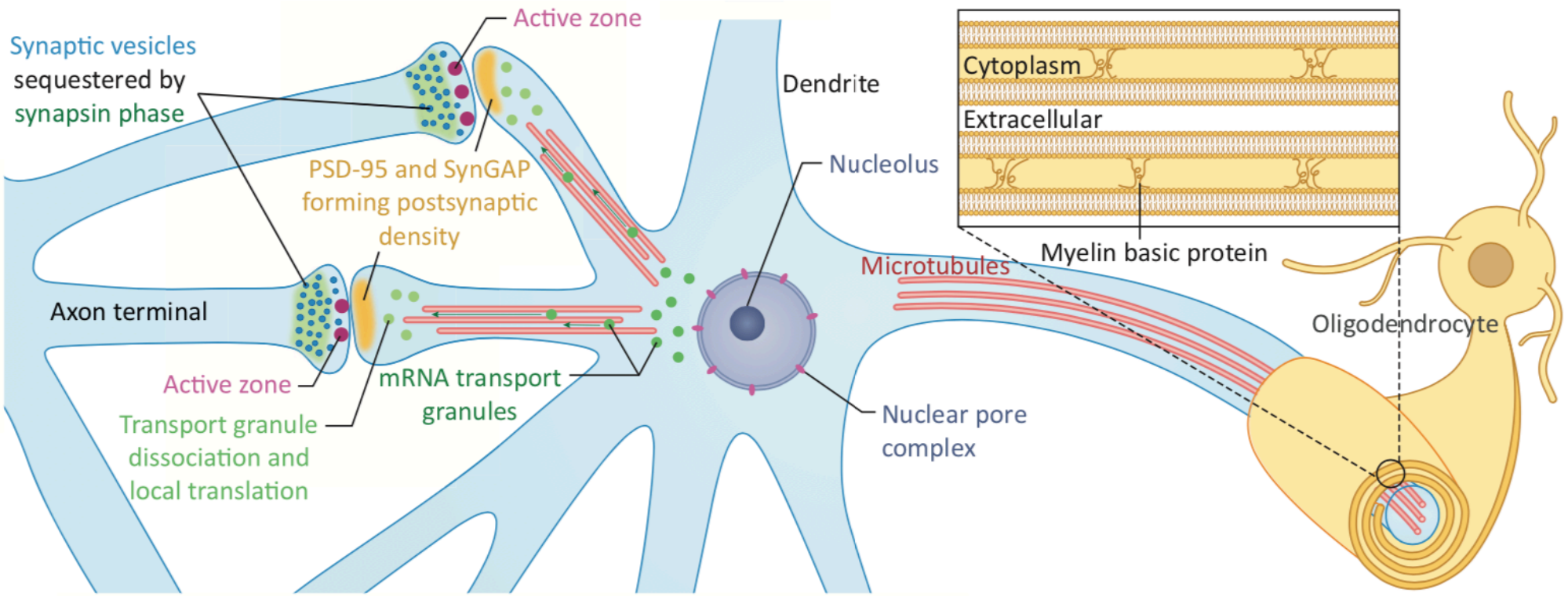
Yes, by thinking ahead about what simulations to do.

Go here  https://epfl-lmnn.github.io/index.html

(divide the time by 10^9)

Core Concepts

- Simulations share a common skeleton (*initialisation*, *interactions*, *evolution*, *boundary conditions*, *observables*)
- Different length scales need different methods for efficiency
- Different scales - different compute resources required
- Coarse-graining - faster simulation but less detail/accuracy
- Coarse-graining is art - some things you get right, but others will be wrong



VH Ryan and N Fawzi, Trends Neurosci. **42**:693 (2019) Trends in Neurosciences

Human brain has

~10^{11} neurons
~10^{4} synapses/neuron

= 10^{15} synapses

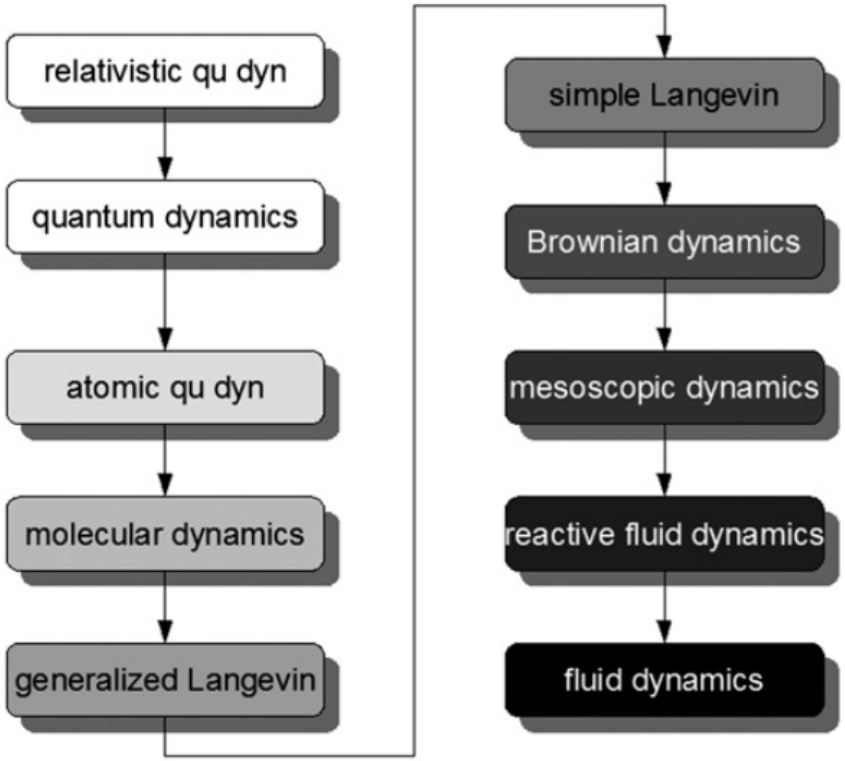
Q. How many water molecules in a synapse?

vol. ~ (0.5 micron)^3

water molecule ~ 0.3 nm

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)

Why coarse-grained simulations?

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

Physical reasons: don't know the force fields, system is too large or too slow, our experiment is not at atomic scale; we don't need picosecond accuracy, or we're interested in general chemical trends not specific chemicals

Computational reasons: it would be nice to simulate 1 billion Lennard-Jones particles, interacting via a realistic force field, for 10 minutes of real time, however…

For a given problem, we choose an accuracy we can live with and see how to attain it.

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 far beyond molecular, e.g., long wavelength membrane fluctuations, easy to visualize

Coarse-grained simulation types

All based integrating some form of Newtonian equations of motion

$m.dv/dt = F$	MD	Finer ↑
$m.dv/dt = F^C + F^D + F^R$	DPD	
$m.dv/dt = F^C - m\gamma.v + \sqrt{(2m\gamma k_B T)}.\zeta(t)$	Langevin	Coarser ↓
$0 = F^C - \gamma.v + \sigma.\zeta(t)$	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)

How to coarse-grain atoms and molecules

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

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

1) Several atoms ⇒ one bead so fewer d.o.f to integrate

2) Lennard-Jones forces ⇒ softer forces so larger Δt

This means cheaper, faster simulations!

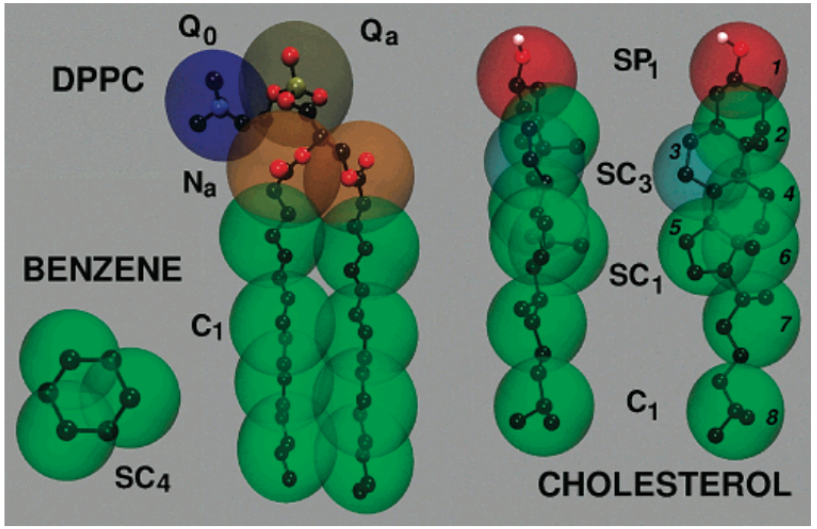


Figure 1. Mapping between the chemical structure and the coarse grained model for DPPC, cholesterol, and benzene. The coarse grained bead types which determine their relative hydrophilicity are indicated. The prefix "S" denotes a special class of CG sites introduced to model rings.

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)

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

Particles interact via 3 non-bonded forces that are: soft, short-ranged (vanish beyond r_0), central, pairwise-additive, and *conserve momentum locally*.

Conservative force gives particles an identity, e.g. hydrophobic or hydrophilic

Dissipative force destroys relative momentum between pairs of interacting particles

Random force creates relative momentum between pairs of interacting particles

Particles are connected together to form molecules (or polymers) using bond forces.

(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: Non-bonded 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 $\mathbf{F}^R_{ij}(\mathbf{r}_{ij}) = \sigma_{ij}(1 - r_{ij}/r_0)\,\Gamma_{ij}\,\mathbf{r}_{ij} / r_{ij}$

Note that $r_{ij} = |\,\mathbf{r}_i - \mathbf{r}_j\,|$, $v_{ij} = |\,\mathbf{v}_i - \mathbf{v}_j\,|$ and γ_{ij} and σ_{ij} must be related by $\sigma_{ij}^2 = 2\gamma_{ij}k_BT$ (see Espagnol and Warren Europhysics Letters **30**:191 (1995)).

The self-interaction value of 25 was first derived by Groot and Warren for water. Other choices exist for specific cases, e.g. those above for lipids (Grafmüller et al. Biophys. J. 96:2658 (2009)).

The random force *creates* relative momentum between *pairs* of interacting particles (which is how it conserves momentum: magnitude is random but the sum is zero):

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

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

where N(0, 1) is a zero mean, unit variance Gaussian RNG (but we usually use a uniform RNG).

The dissipative and random forces act as a thermostat to keep the average temperature constant (canonical ensemble). It is *independent* of the conservative force and is sometimes used with MD forces - (Soddemann et al., PRE 68:046702 (2003)).

Setting DPD conservative parameters

The dissipative and random forces form a thermostat that does not change when simulating different systems. We'll ignore them, but see Groot and Warren (1997) for details.

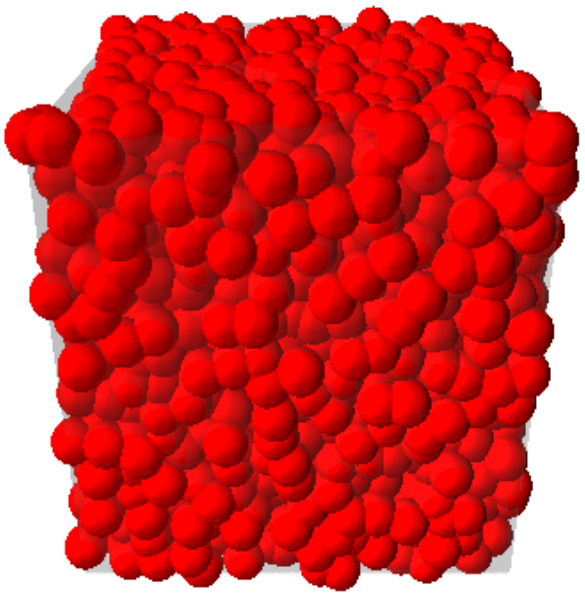
The conservative interaction parameters a_{ij} can be set from thermodynamics.

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}.



$L_X = L_Y = L_Z = 10r_0$

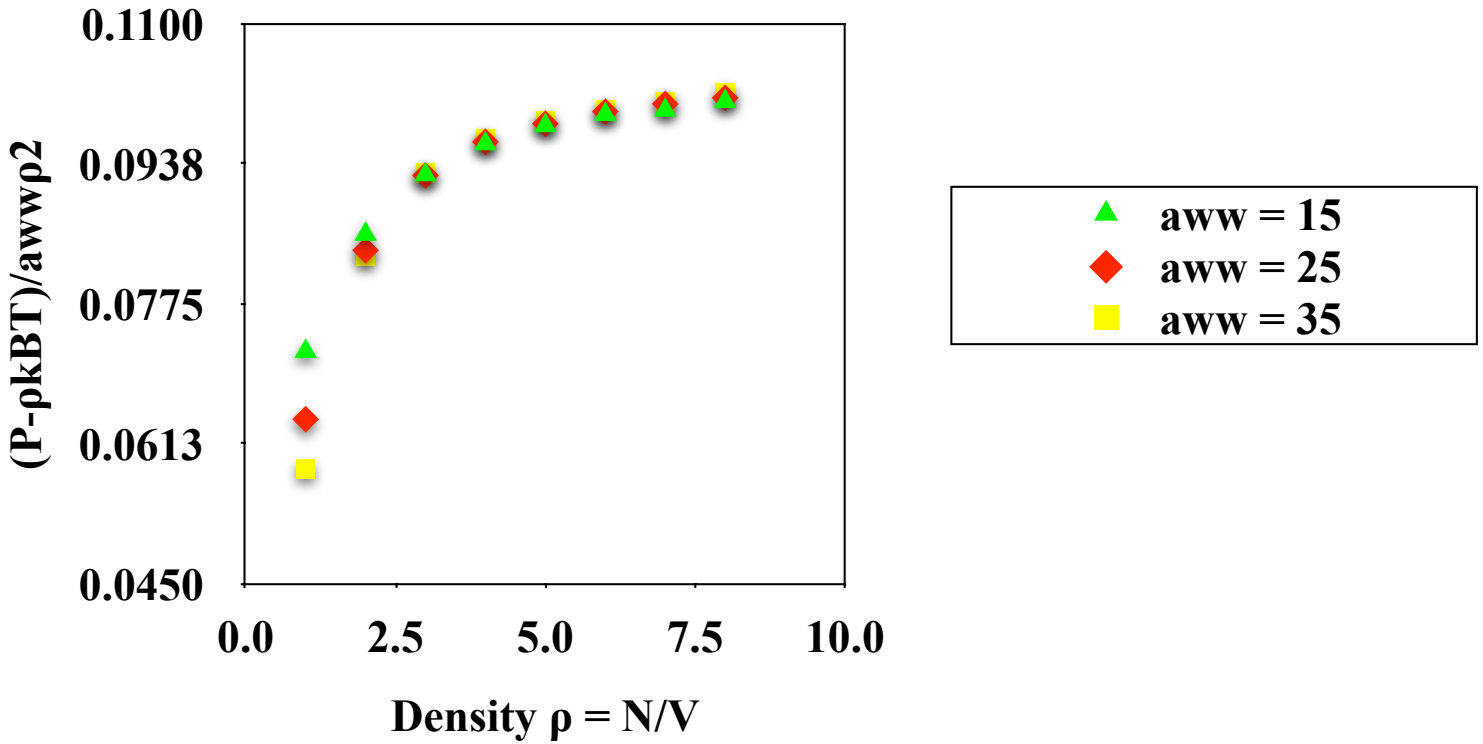
ρ = 3 - 10 beads/volume

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 squared.



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 $\alpha = 0.10 \pm 0.01$

How to choose the conservative parameter

$$K_T = -1/V \left(\partial V/\partial P \right)_T$$

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

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

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

giving $a_{WW} = 75\, k_B T / \rho$. Most DPD simulations use a (dimensionless) bead density of $\rho = 3$

so $a_{WW} = 25\, k_B T$

So, the density of a single-component DPD fluid 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 in DPD

Every bead type interacts with all others (e.g., lipids with head beads H, tail beads T, immersed in water W - see table)

The off-diagonal elements of the force matrix set the repulsion or attraction between fluid elements of different types

These elements are related to their mutual *solubility*

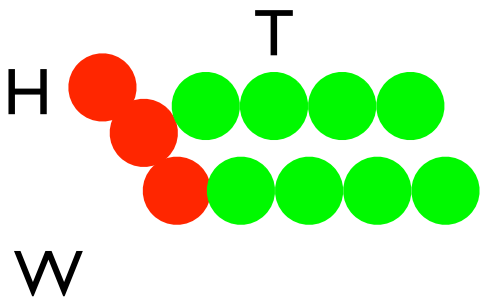
Note that all DPD forces are repulsive: the self interactions are repulsive because they represent the compressibility (resistance to being compressed) of a pure fluid, and the off-diagonal elements are repulsive because they represent the solubility of mixtures which are usually less cohesive than the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

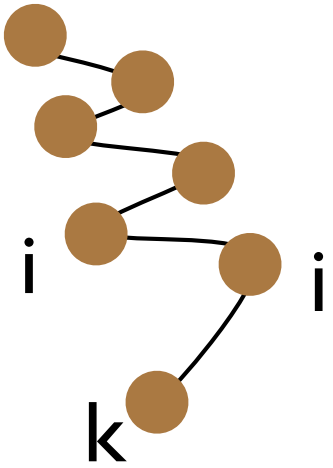
This is the price paid for having no hard core repulsion - you cannot have purely attractive forces or the fluid will collapse on itself.

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

In Lecture 13, we'll cover the theory of how to fix the off-diagonal force parameters (Flory-Huggins theory)

DPD algorithm: Bond forces





Beads are connected by bonds

$$\mathbf{F}(\mathbf{r}_{ii+1}) = -k_2(|\mathbf{r}_{ii+1}| - l_{ij0})\,\mathbf{r}_{ii+1}/|\mathbf{r}_{ii+1}|$$

Hookean spring parameters: $k_2 = 128$, $l_{ij0} = 0.5$, so $V_2(r) = 0.5.k_2(r_{ij} - l_{ij0})^2$

These parameters were chosen to keep the lipid tail length on average at the desired value: so a physical measurement was used to constraint their value. In principle, parameters can vary for all bead types, but this is not common.

Adjacent bonds can have an angle constraint

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

Chain bending stiffness parameters: $k_3 = 15$, $\phi_0 = 0$ was chosen to ensure lipids don't interdigitate.

Each parameter needs a distinct physical measurement to fix its value.

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

```
Bead   H
       0.5
       30
       4.5

Bead   T
       0.5
       35   10
       4.5  4.5

Bead   W
       0.5
       30     75   25
       4.5    4.5  4.5
```

Bead name

Radius (= 1/2 range of non-bonded forces)

Conservative force parameter

Dissipative force parameter

Beads at each end of bond (symmetric)

Spring constant / unstretched length

```
Bond       H H   128   0.5
Bond       H T   128   0.5
Bond       T T   128   0.5

BondPair   H T T 15.0     0.0
BondPair   T T T 15.0     0.0
```

Bead triple defining a bending potentital bond

Energy / preferred angle

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 : $p(t+dt) = p(t) + 0.5*dt*(F(t) + F(t+dt))$

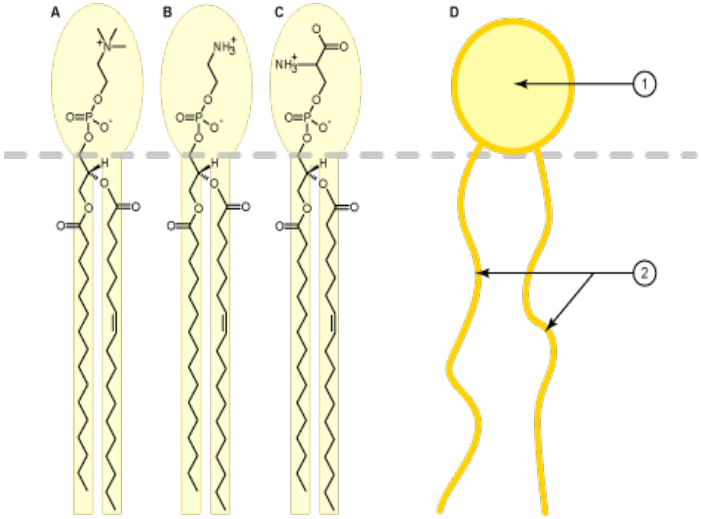
Because we set m = 1, velocity (v) = momentum (p).

Note that λ is a heuristic parameter, typically ~0.5, to take the stochastic force into account (see dmpci file). It is used to account for time-varying force over the course of a time-step. It is needed no matter how small the time-step, because the random force necessarily changes during the step, i.e., the stochastic force is not 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.

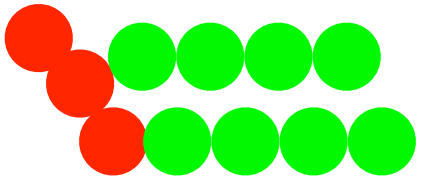
How do you coarse-grain a lipid?

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



$H_3(T_4)_2$

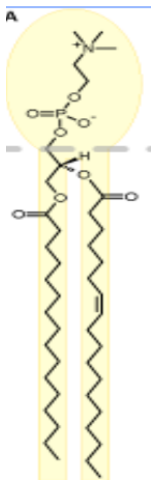
For DMPC and lipids that differ only in tail length (lauryl, myristoyl, palmitoyl, stearoyl, ...). We find the relation that each DPD tail bead represents 3-4 methyl groups. So cgDMPC has ~11 beads. Ambiguity comes from the fact that a DPD *bead* is a rather fuzzy concept, based on a volume of material, 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. The two monolayers should not inter-penetrate each other, which requires bending stiffness of tails.

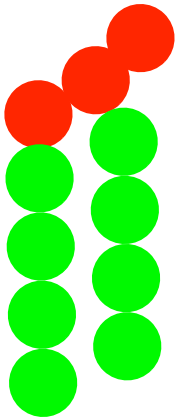
Shillcock, JC, and Lipowsky, R, J. Chem. Phys. 117:5048 (2002)

Coarse-graining a lipid membrane



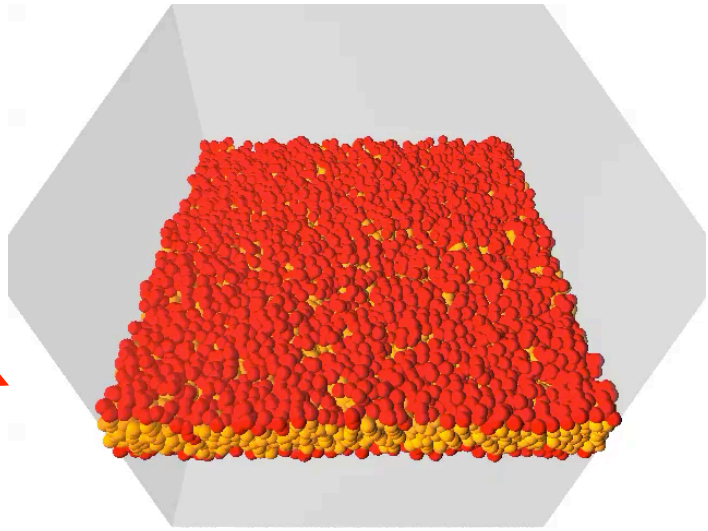
Real lipid

How?



cg lipid

How?



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

Experimental lipid bilayer properties

Fully hydrated lipid bilayer areas, thicknesses and K_C's, June 2009

Lipid	Temp. (°C)	Area ± 0.5 (Å²)	Hydrophobic Length, $2D_C$ (Å)	Bending Modulus K_C ($x10^{-20}$J)	
DPPC	50	62.9(±1.3)[a],64.0[b],64.3[c],63.1[d*]	29.2[a],28.5[b],27.9[c],28.4[d*]	6.7(±0.7)[e]	
DHPC	48	65.1[e]	27.6[e]	4.2(±0.7)[e]	
DLPE	35	51.2[b]	25.8[b]	-	
DMPC	30	59.7[f],60.6[g]	26.2[f],25.4[g]	6.9[g]	
DLPC	30	63.2[g]	20.9[g]	5.5[g]	
DOPC	30	72.2[h],72.5[b],72.1[i],72.4[j,k],67.4[d*]	27.2[h],27.1[b], 27.2[i],26.8[j,k],28.8[d*]	8.0[i,j],7.6(±0.5)[k]	
	(15)	69.1[k]	27.7[k]	8.5(±0.5)[k]	
	(45)	75.5[k]	26.2[k]	7.2(±0.5)[k]	
DOPS	30	65.3[l]	30.2[l]	-	FLUID PHASE
EggPC	30	69.4[f,b]	27.2[f],27.1[b]	-	
POPC	30	68.3(±1.5)[j]	27.1[j]	8.5[j]	
SOPC	30	67.0(±0.9)[m]	29.2(±0.4)[m]	9.0(±1.2)[m]	
diC22:1PC	30	69.3[j]	34.4[j]	12.7[j]	
18:0-22:5PC	24	68.7[n]	30.5[n]	11.0(±0.2)[n],10.7±0.8**	
18:0-22:6PC	24	68.2[n]	30.5[n]	12.0(±0.2)[n], 7.9±0.5**	
DMPC	10	47.2[o]	30.3(±0.2)[o]		
DiC16PC,18,20,22,24	20	47.5[p,q]	34.4[b],37.1[q],40.7[q],44.0[q],48.0[q]		
DMPS	20	40.8[l]	36.0[l]		
DLPE	20	41.0[b]	30.0[b]	GEL PHASE	
DHPC-Interdig.	20	77.2[e]	20.3[e]		
DHPC-gel	20	46.9[e]	34.6[e]		

[a]Biophys.J. 70:1419(1996); [b]Biochim.Biophys.Acta: Reviews on Biomembranes 1469:159(2000); [c]Biophys.J.:Biophys.Lett 90:L83(2006); [d]Biophys.J. 95:2356(2008); [e]Chem.Phys.Lipids 160:33(2009); [f]Chem.Phys.Lipids 95:83(1998); [g]Biophys.J. 88:2626(2005); [h]Biophys.J. 75:917(1998); [i]Phys.Rev.E 69:040901(2004); [j]J.Membr.Biol. 208:193(2005); [k]Biophys.J. 94:117(2008); [l]Biophys.J. 86:1574(2004); [m]Biochim.Biophys.Acta 1178:1120(2008); [n]J.Am.Chem.Soc. 125:6409(2003); [o]Biophys.J. 83:3324(2002); [p]Biophys.J. 64:1097(1993); [q]Biophys.J. 71:885(1996); *Neutron data; **Upon reanalysis(2009)

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 (a_0) in nm^2 to determine the number of lipid molecules:

$N_{lipid} = 2.((32\ r_0\ nm)^2 / a_0\ nm^2)$ molecules

Initially choose $a_0 \sim \pi (r_0/2)^2 \sim 0.785 r_0^2$ this gives N ~ 2607 (assumes all lipids are little circles!)

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 $a_{Lipid} = A/N$ from the simulation and equate it to experimental value. If $a_{Lipid} = 1.26\ r_0^2$, and the experimental value is $a_0 = 0.6\ nm^2$:

$r_0 = \sqrt{(0.6 / 1.26)} \sim 0.69$ nm and $N_{lipid} = 1621$ in equilibrium

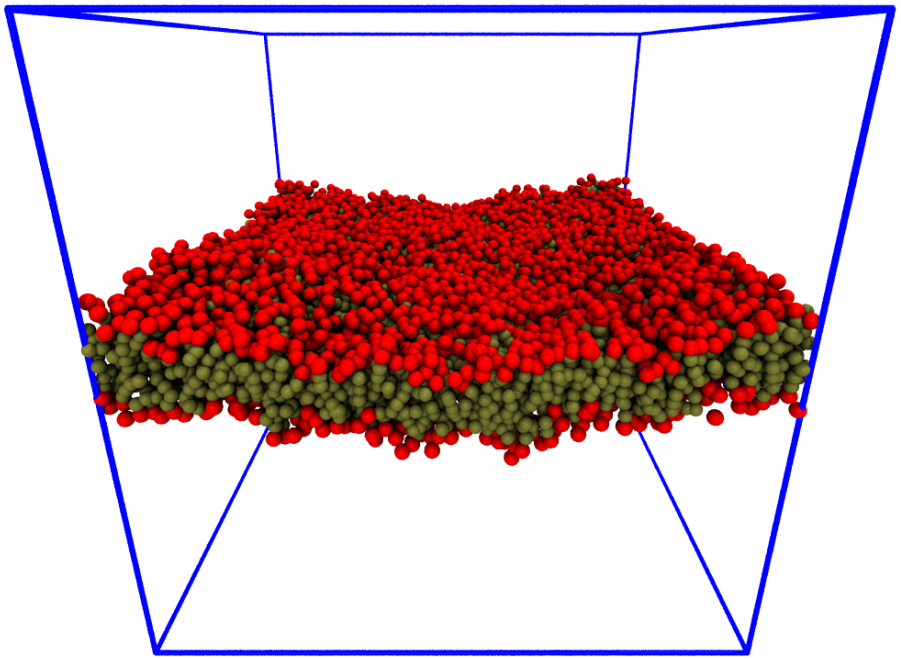
Question. Why does each lipid occupy an area ~ $1.26\ r_0^2$ instead of $\pi (r_0/2)^2 \sim 0.78\ r_0^2$?

see dmpci.m6 on moodle page for today for an equilibrated lipid bilayer simulation

```
Bead  H
      0.5
      30
      4.5

Bead  T
      0.5
      35  10
      4.5 4.5

Bead  W
      0.5
      30    75  25
      4.5   4.5 4.5


Bond      H H  128  0.5
Bond      H T  128  0.5
Bond      T T  128  0.5

BondPair  H T T 15.0    0.0
BondPair  T T T 15.0    0.0

Polymer  Water  0.9802   " (W) "
Polymer  Lipid  0.0198   " (H H (* (T T T T)) H T T T T) "
```

dmpcas.m6

```
Bilayer Trapezoidal Surface Tension
-0.032320654  1.3417115

Bilayer Surface Tension
-0.032335754  1.3427113
```



A time scale for lipid bilayers?

EPFL

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 = 0.69 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):

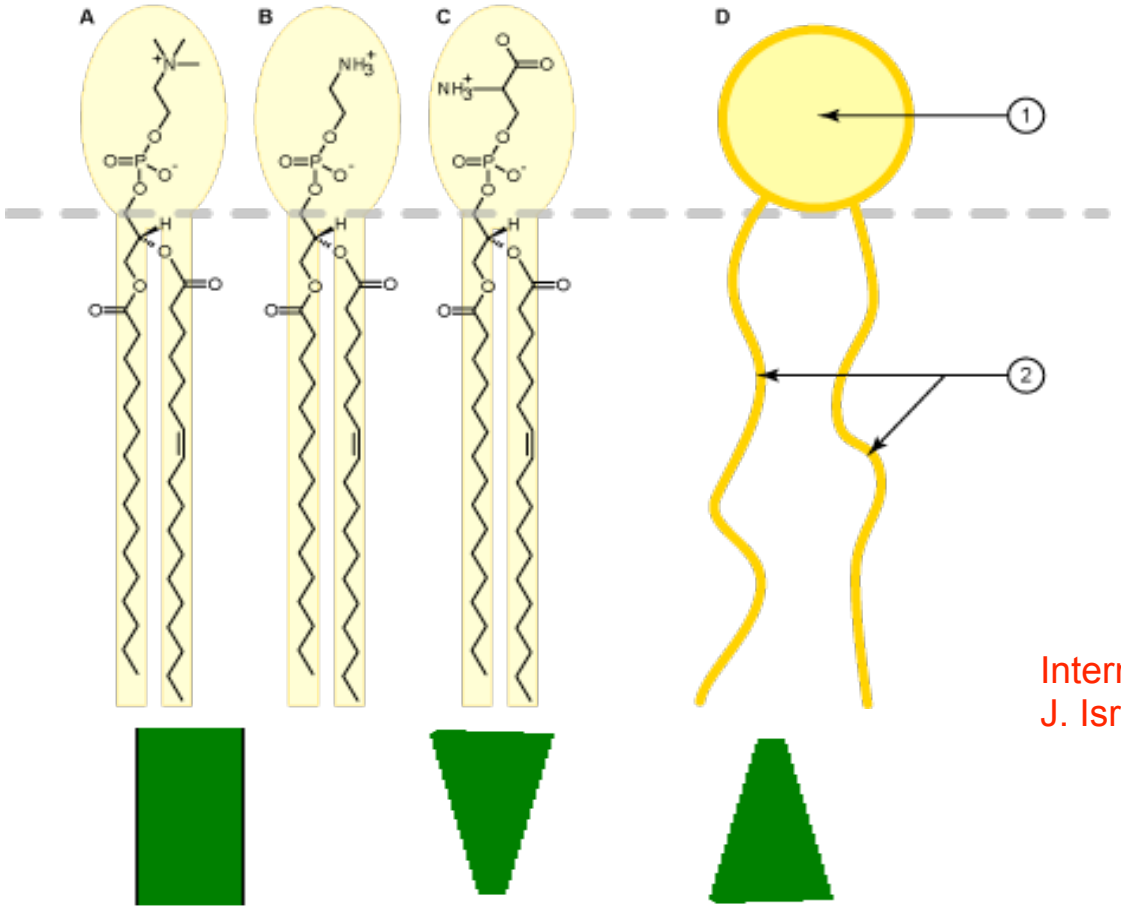
$$t_0 = 0.01.(0.69.10^{-9})^2 / 10^{-12} \sim 5 \text{ ns}$$

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, and you can estimate the real time that the simulation represents.

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.

Lipids have a shape

If the lipid headgroup has the same "size" as the tails the molecule is like a cylinder; if the head or tail has a smaller volume, the molecule is like a cone. This shape strongly influences their behaviour.



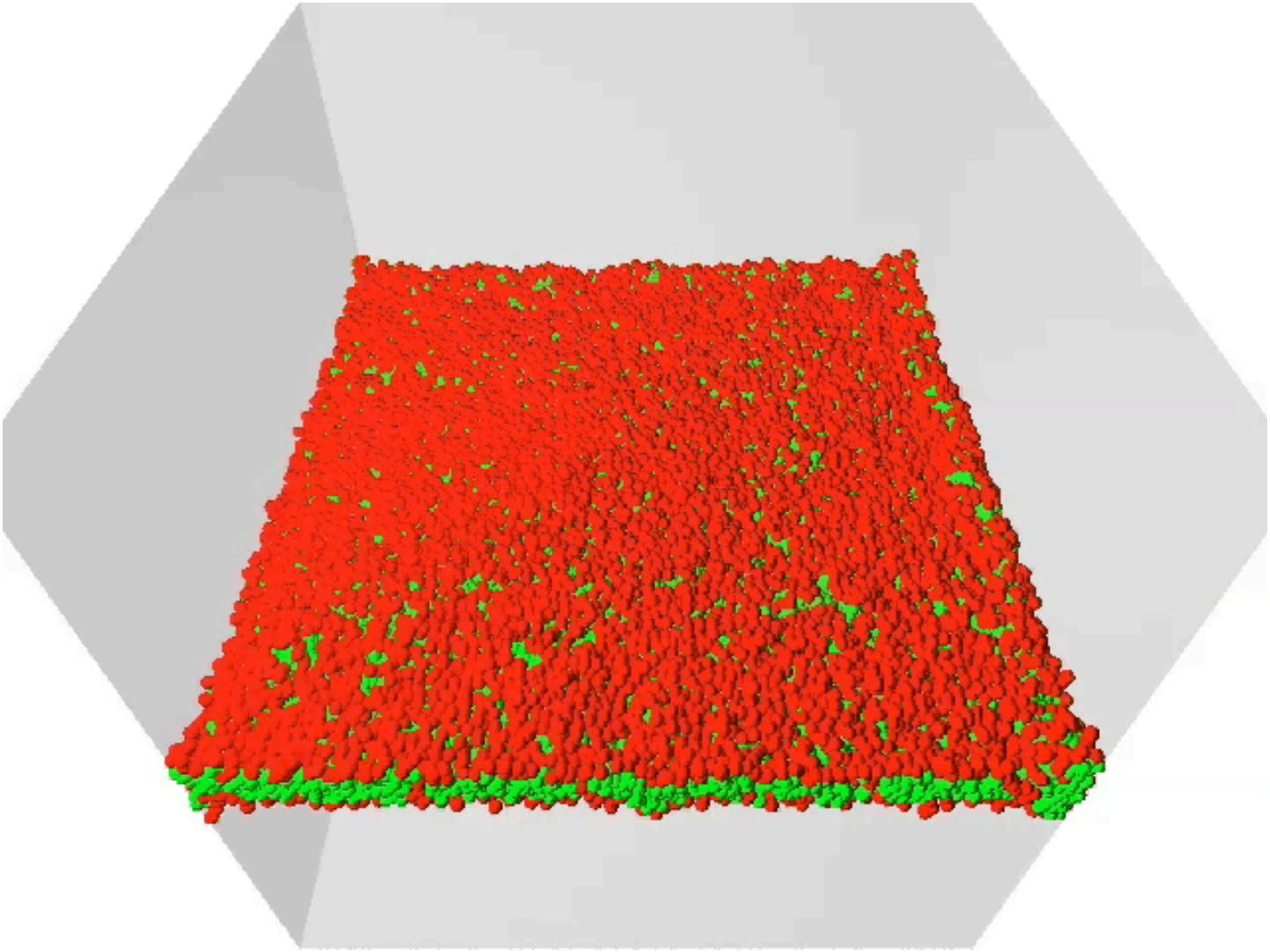
Packing parameter ~ v/a_0L_c

v = lipid volume
a_0 = headgroup area
L_c = tail length

Intermolecular and Surface Forces, 2nd edn.
J. Israelachvili, Academic Press, London, 1992.

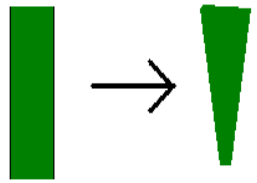
If v/a_0L_c ~ 1 planar bilayers; < 1 curved; < 1/3 micelles; > 1 inverted micelles

DPD is useful for lipid shape changes



Initially-tensionless membrane
5538 lipids
40 nm x 40 nm

$C_0 = 0 \rightarrow > 0$



NB Water is invisible

Illya, G, Lipowsky, R, Shillcock JC, J. Chem. Phys. 122:244901 (2005)

What can we use DPD for?

Recall that which simulation technique to use depends on what you want to know.

DPD is good for:

- soft matter
- complex fluids
- interactions larger than ~ atom/molecule
- averages over many molecules
- interactions that depend on entropy or steric forces not specific ligand binding or ES
- trends not detailed chemistry

In the context of cellular biophysics, potential topics include:

- self-assembly of supramolecular structures, droplets, vesicles, membranes, nanoparticles
- membranes - structure and dynamics
- nanoparticle interactions with membranes, vesicles, polymers
- phase transitions and order

Summary

How do we relate a real system (e.g., membrane) to a coarse-grained simulation?

Compare dimensionless ratios of important quantities for M, L, T

Important = Relevant to the dynamics of interest

Coarse-graining implicitly:

collapses times scales
softens atomistic force field
loses finely-detailed information

But it speeds up simulations, allows much larger systems, reveals long length and time scale motions inaccessible in atomistic simulations

Coarse-graining is most useful when the important motions are larger/slower than atomic length/time scales

Exercise period

1. No lecture or exercise period next week
2. Distribute Take home test 2, due in 2 weeks (30th October)
3. Work on homework 1
4. Can run jobs on the helvetios machine?

Data Management Plans

Funding agencies now expect applications to include a DMP. A DMP specifies what data you will produce, how you will store it and name files, and how other users can find it and use it again.

EPFL provides information on preparing DMPs:

https://www.epfl.ch/campus/library/services/services-researchers/rdm-guides-templates/

A common standard are referred to as the *FAIR* principles:

https://www.epfl.ch/campus/library/wp-content/uploads/2019/09/EPFL_Library_RDM_FastGuide_All.pdf#page=2

DMP

A DMP is a *living document* that describes the following aspects of your data:

- What data is produced? e.g., fluorescent microscopy images, simulation files, gene sequences, etc.
- What formats are used to store it? e.g., tiff, png, bmp, ascii
- What file/directory naming strategy will be used? e.g., /users/shillcock/my_data
- (What license it will be released under - not relevant in this course)

etc

Findable - how can a user find the data?
Accessible - how is it licensed and released?
Inter-operable - does user need special software to read data?
Reusable - Long-term storage of data in a repository

Living = continuously updated throughout a project

Ex. 1 Entropic spring

Simulation of an entropic spring - apply a stretching force to both ends of a single, long polymer in a DPD simulation and measure the end-end length as a function of the force (it probably has to be a very small force).

Then invert it to get F(L) and plot it including error bars of the statistical errors. How do you convert results to physical units?

Now make a fraction of the beads sticky (so that the polymer tends to stick to itself) and see how this changes the F(L) curve. You will need to vary the number of sticky beads to find an interesting regime (too few and nothing will happen, too many and the polymer will just stick together in a tight ball). Interesting means that the system shows some unusual, non-linear behaviour.

Needs commands in DPD to solve - see Section 8 of the User Guide.

```
Bead  W
      0.5
      25
      4.5

Bead  B
      0.5
      25    25
      4.5   4.5

Bead  BH
      0.5
      25   25   25
      4.5  4.5  4.5

Bead  BT
      0.5
      25   25   25   25
      4.5  4.5  4.5  4.5



Bond  BH B 128    0.5
Bond  BT B 128    0.5
Bond  B B  128    0.5

Polymer Water    0.99995   " (W) "
Polymer Spring   0.00005   " (BH (14 B) BT) "

Box         30  15  15       1  1  1
Density     3
Temp        1
RNGSeed     -999
Lambda      0.5
Step        0.01
Time        6000
SamplePeriod     10
AnalysisPeriod   2000
DensityPeriod    6000
DisplayPeriod    100
RestartPeriod    6000
Grid        1  1  1
```

Input file on moodle page for today: dmpci.ex1

Simulating an entropic spring under tension

We create command targets for the two ends of a molecule and apply equal and opposite forces to stretch it.

```
Command  SelectBeadTypeInSimBox 1  head  BH
Command  SelectBeadTypeInSimBox 1  tail  BT

Command  Comment  1000  // Apply a constant force to the first and last beads in
the +X and –X directions  //

Command ConstantForceOnTarget        1000  head  fh   1 0 0   5.0
Command ConstantForceOnTarget        1000  tail  ft   1 0 0   -5.0


Command  Comment  5000  // Delete the applied forces  //

Command RemoveCommandTargetActivity  5000   fh
Command RemoveCommandTargetActivity  5000   ft
```

Measuring the strain

To Do:

1. Pick a box size of 30 x 15 x 15; adjust the number fractions to have 1 polymer of type (BH (14 B) BT), i.e., distinct head and tail beads so they can be selected.
2. Turn force on at T = 1000 steps. How long should you keep it on?
3. How can you measure the extension?
4. Next, change the backbone to contain a new bead type that is "sticky". Try (BH B B B S S S S B B B BT), and give S the same interactions as B except for its self interaction that is reduced to make it sticky. Vary the number of S beads until you find a value that makes an observable difference.

Questions to answer

What is the stress/strain relation F(L) for the "molecular spring"?

Does it have different regimes for F(L) under different tensions? Why?

With sticky beads there are two new parameters: the number of sticky beads and their self-interaction. How can you reasonably select values for these?

dmpcas file contains time-averaged observables

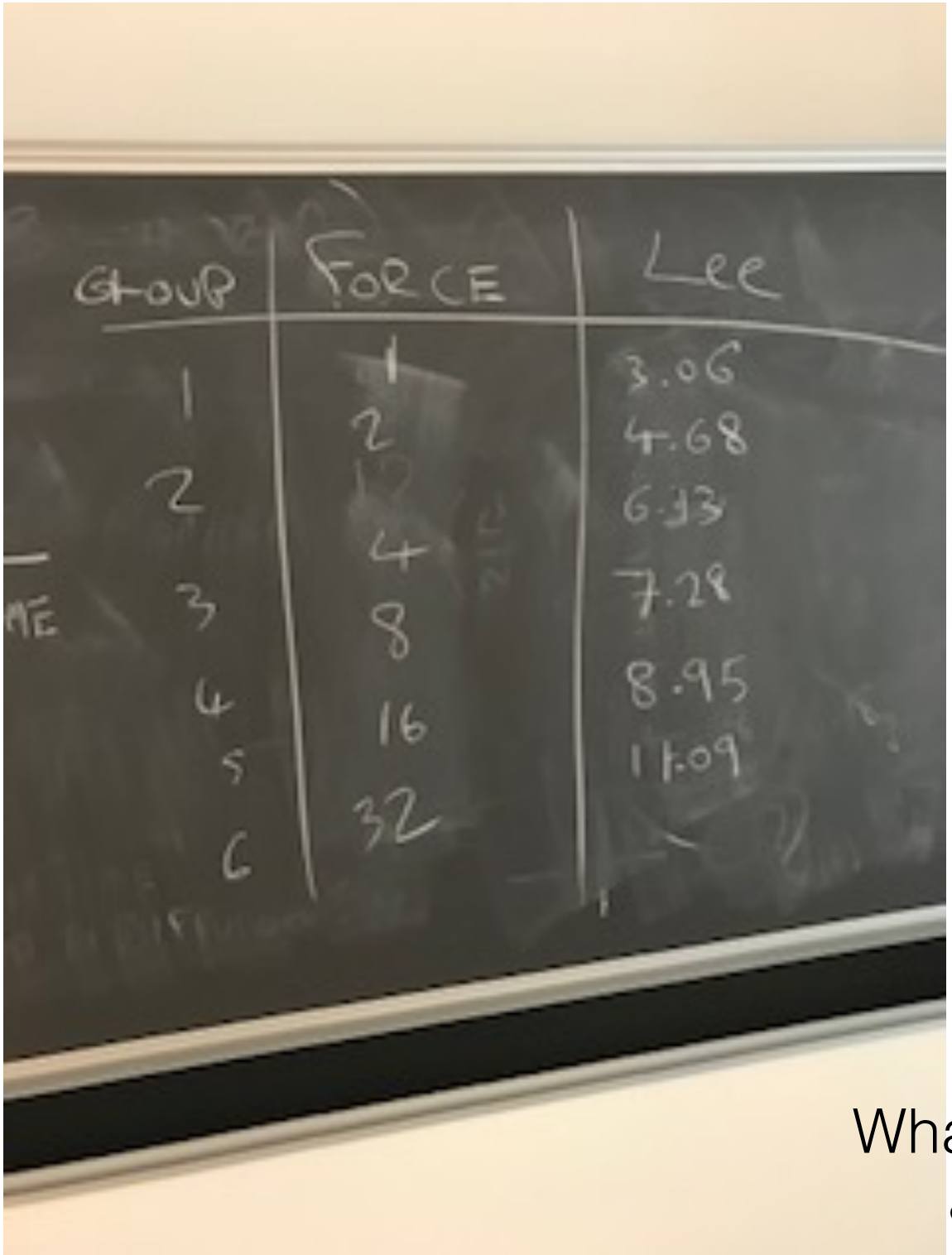
Typically there are 2 columns: mean and standard deviation

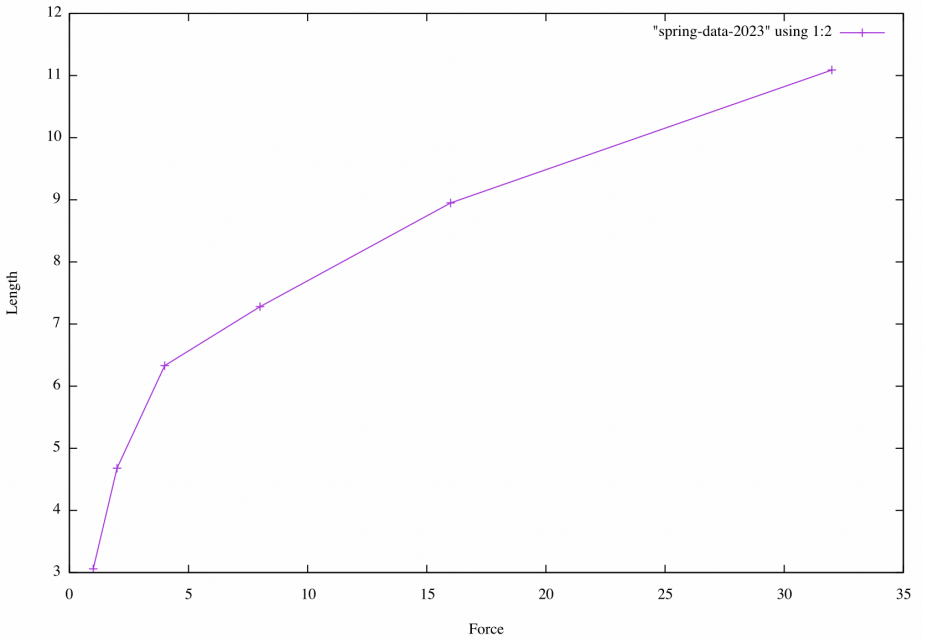
dmpcas.ex1 has <Lee> averaged between 2000 - 4000 steps

```
Time = 4000
Temperature
1.0017449      0.0051323594

Pressure
23.671181      0.081945563

...

BB bond length
0.58796179     0.020986014

Water EE distance
0              0

Spring EE distance
6.4922776      0.6952996
```

Use the time period while force is applied, which is at the end time of the period

This is the mean and std. dev. of polymer end-to-end length





Two limits:
F ~ 0, entropic spring

F >> 0, series of Hookean springs

What are the spring constants?