Take-Home ~~message~~ questions

- Why do simulations work?
- When don’t they work?
- What can you get from a simulation?

What is a simulation?

A simulation is not:

- analytically solving a differential equation or pde - e.g., ballistics versus weather
- quadratures - e.g., calculating a Fourier transform of electron density vs. P.F. $Z(\{\mathbf{x}\})$

What is a simulation?

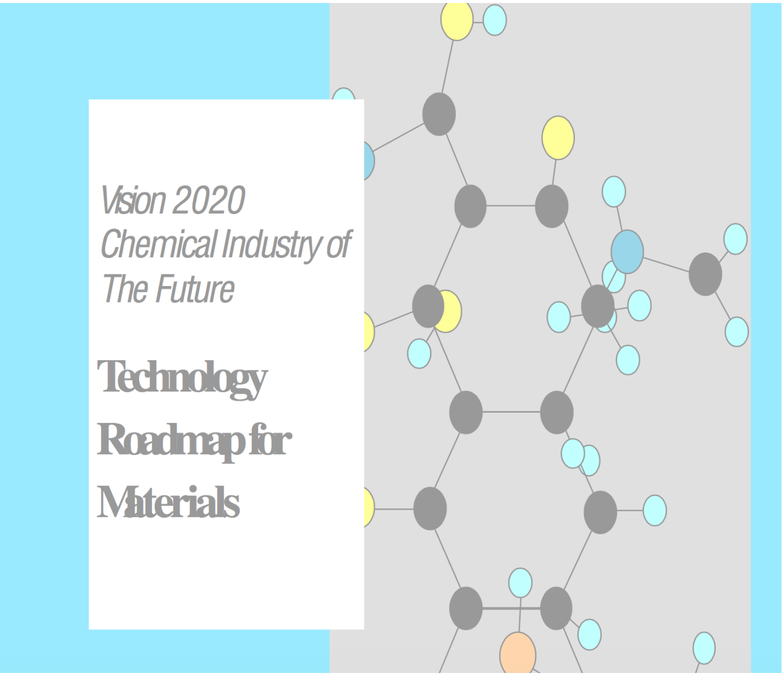
"a computer *experiment* of the behaviour of a *model* of a physical system in which matter is replaced by mathematical constructs that interact in ways that mimic the interactions in the physical system, and where the model's evolution generates states corresponding to those of the real system."

Caveat

We never simulate a real system, but only a model of a real system; we first have to construct a model and second adapt it for calculation on a computer.

Why do we do simulations?

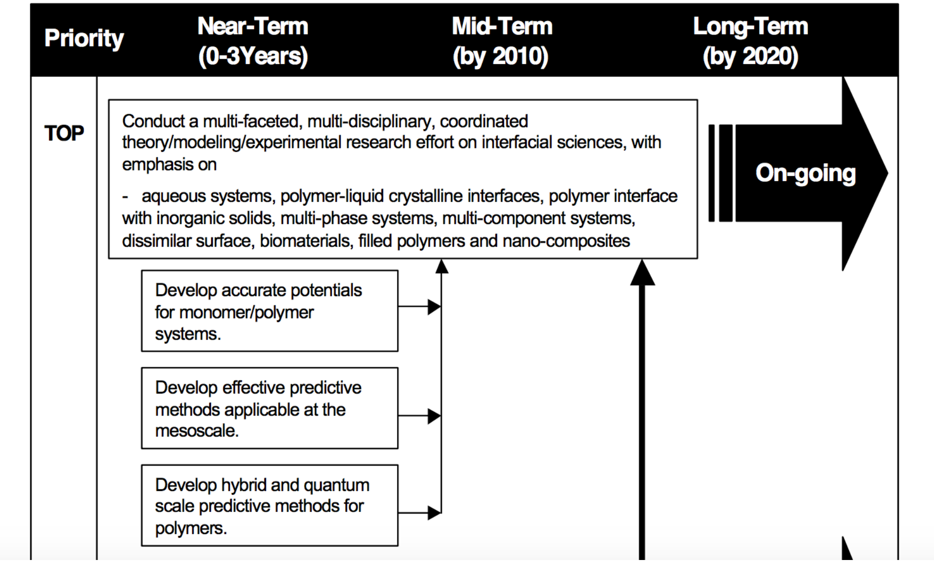
- Experiments are too complicated and theories are too simple
- A model can capture what we think are the important properties of an experiment, and allow us to ignore irrelevant aspects: if we later find the model is wrong, we can look for the missing important property
- We have almost complete control over all aspects of the simulation; so we can perform thought experiments like changing atmospheric pressure, turn electrostatic interactions on or off, etc
- Simulations are relatively cheap and quick compared to experiments
- We can visualize aspects of a simulation that are hard to do in an experiment



Vision 2020
Chemical Industry of
The Future

Technology
Roadmap for
Materials

Figure 4-3. Priority R&D for Modeling and Predictions: Methods, Theory and Validation



Priority | **Near-Term (0-3Years)** | **Mid-Term (by 2010)** | **Long-Term (by 2020)**

TOP

Conduct a multi-faceted, multi-disciplinary, coordinated theory/modeling/experimental research effort on interfacial sciences, with emphasis on

- aqueous systems, polymer-liquid crystalline interfaces, polymer interface with inorganic solids, multi-phase systems, multi-component systems, dissimilar surface, biomaterials, filled polymers and nano-composites

Develop accurate potentials for monomer/polymer systems.

Develop effective predictive methods applicable at the mesoscale.

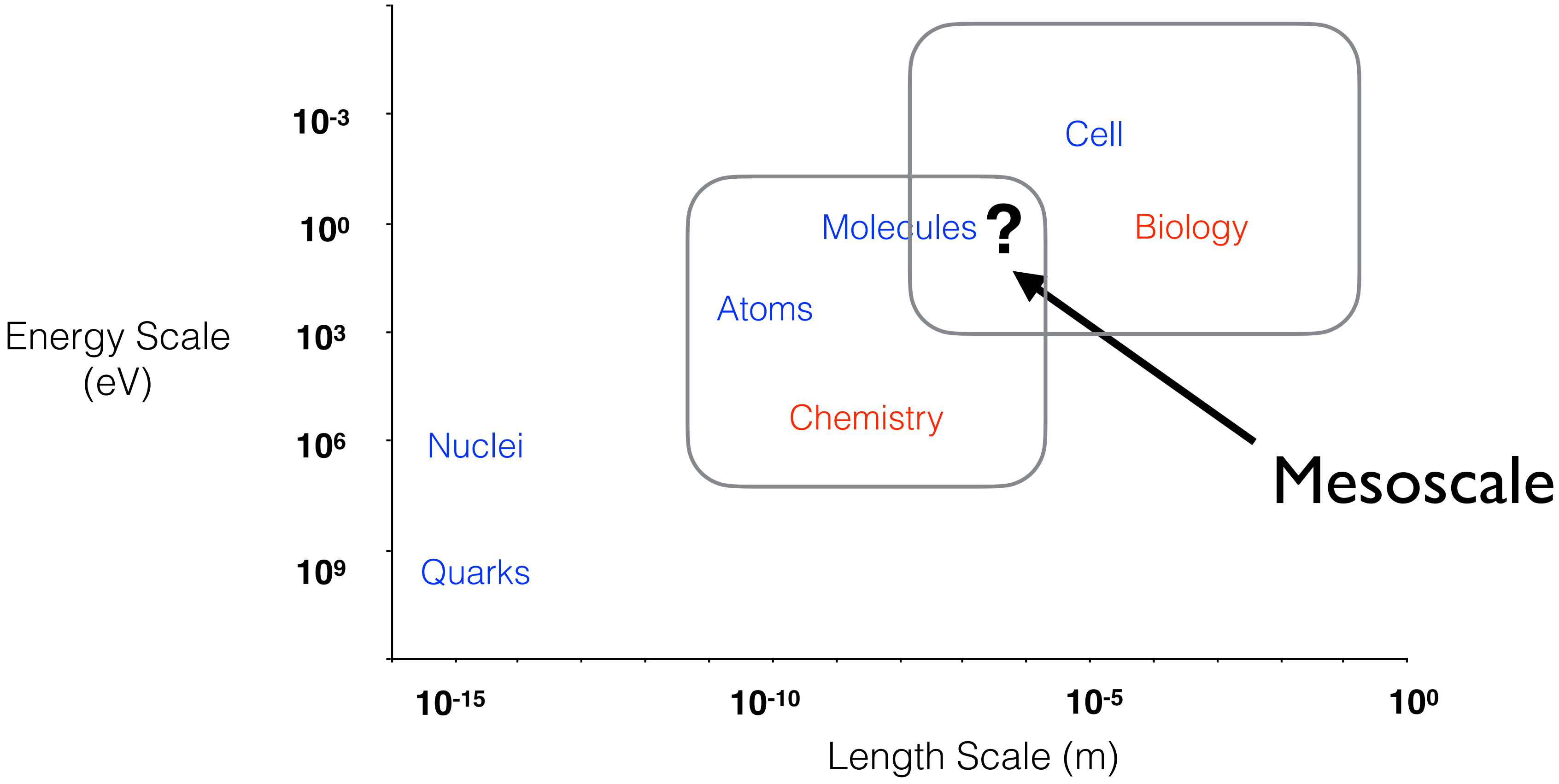
Develop hybrid and quantum scale predictive methods for polymers.

http://www1.eere.energy.gov/manufacturing/resources/chemicals/pdfs/materials_tech_roadmap.pdf

Why do simulations work?

Simulations work because the world appears to be able to be divided into different layers - defined by relevant length and energy scales - in which entities move and interact according to forces that only apply, and refer, to entities in that layer.

Is there something new in the mesoscale between molecules and cells?



R. B. Laughlin et al. The Middle Way, PNAS 97:32-37 (2000)

What types of simulation are there?

There are essentially two types of simulation in biophysics (but with sub-divisions and cross-over):

Mechanical types - that integrate *more-or-less* accurate equations of motion for interacting particles, .e.g., Newton’s laws for a rocket, Molecular Dynamics, Dissipative particle dynamics, Brownian dynamics, …

Statistical mechanical types - that calculate observable averages in specific thermodynamic ensembles: $\langle A \rangle = 1/Z \sum A(\{x_i, v_i\}) e^{-\beta H(\{\mathbf{x_i}, \mathbf{v_i}\})}$, e.g., Monte Carlo.

These are mathematically distinct but physically equivalent (where they can both be applied) ways of calculating properties of a system. Which is more useful depends on the problem at hand.

agent-based modelling?

What kinds of simulation can we do?

1) Those based on integrating some form of Newtonian equations of motion for interacting particles:

$m.dv/dt = F$	MD	↑
$m.dv/dt = F^C + F^D + F^R$	DPD	Finer
$m.dv/dt = F^C - m\gamma.v + \sqrt{(2m\gamma k_B T)}.\zeta(t)$	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 or molecules. Once the particles are defined (mass, radius), and the forces given (bonds, non-bonded, electrostatics), we integrate Newton's 2nd law.

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)

2) Those based on defining the Hamiltonian of a system, and performing phase space averages using the Metropolis Monte Carlo method. Lattice models are a sub-set of MC where the d.o.f are confined to a lattice for simplicity and speed (see Lecture 10)

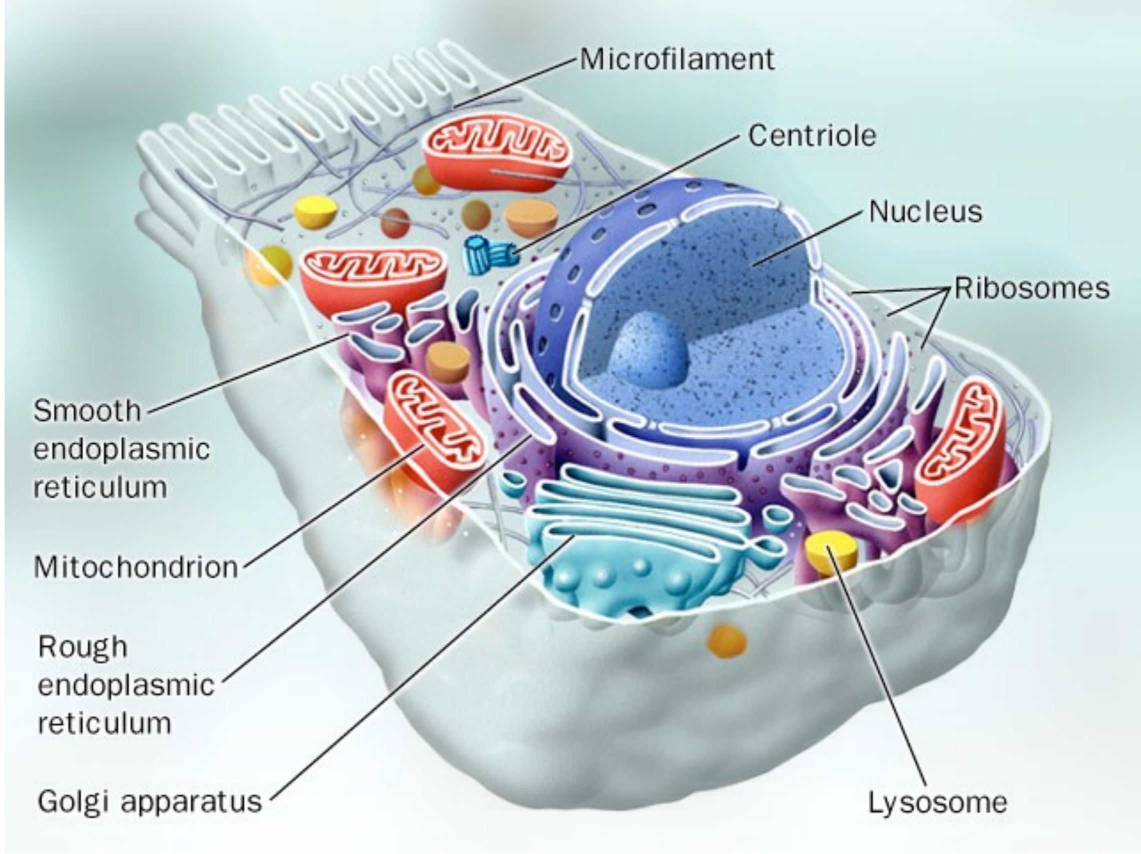
Binder, K (ed.) Monte Carlo Methods in Statistical Physics,

Topics in Current Physics Vol. 7, Springer Verlag, Berlin, 1986

Simulations in the cell

What kinds of cellular dynamics can we simulate?

Protein-protein binding
Protein diffusing in cytosol
Ion channel dynamics, pumps
Membrane potential dynamics
Filament self-assembly and collapse
Actin cytoskeleton dynamics
Membrane fluctuations
Vesicle transport
Vesicle fusion
Endo- and exocytosis
Golgi, ER self-assembly and transport
Cell crawling



www.daviddarling.info

Usually we are interested in the *thermodynamic* properties of a system, so we create a mathematical model and relate it to the physical system via *thermodynamic coordinates*: temperature, pressure, volume, work and heat, electric field, charge, force, area, length, etc.

How to choose a simulation type for a given soft matter problem?

Ask yourself:

- What are the length and time scales? ns and nm or microns or metres?
  Local or global properties?
- Am I interested in trends or absolute values?
- Do I need to study a *particular* molecule? e.g., DOPC vs DOPG or a generic "lipid"
- Are H-bonds (or C=C, or aromatic, etc) important?
- Do I have accurate values for interaction parameters between atoms/molecules?
- Am I interested only in equilibrium states?
- If not, what part of the dynamics do I need to get right?
- What computer resources do I have?

Often, there is not much choice about which technique to use as it is forced on you by the system you want to study.

Calculation:
How many water molecules are in a neuronal synapse, whose volume is ~ $(1/2\ \mu m)^3$

How many x, y, z coordinates would you have to integrate in 1 second with 1 femtosecond timestep?

Physical System

The **System** is a physical experiment that we want to reproduce in a computer:

Argon gas in a box
Lipid molecules in a membrane
Ferromagnetic atoms on a lattice
Rodlike liquid crystal molecules in solvent

It must be in a measurable **State** with specified **Initial Conditions** and **Boundary Conditions:**

0.05 Mole of Argon gas in a closed 1 litre glass bottle at 300 K
0.6 gm DMPC in 1 ml of water at STP

The physical entities must **Interact** in some way with each other and with the container; typically, the system has constant mass that is ensured by a closed container**:**

lipids in water can diffuse around, aggregate and separate, but are constrained by the walls of the container so that their number is constant

We need an **Equation of Motion** for the entities, usually Newton's laws or some artificially-chosen EOM. And we must be able to measure something, viz, **Observables**.

Mathematical System

We must represent the physical entities as mathematical objects in precisely-defined *states*

argon atoms in a box ⟹ point particles with mass, position, velocity and force field

lipid molecules in membrane ⟹ ball-and-spring model, Lennard-Jones potential (6-12)

ferromagnetic atoms on a lattice ⟹ Ising spins with 2 states: up or down

rodlike liquid crystal molecules ⟹ rigid ellipsoids with non-spherically symmetric potential

Finally: what accuracy is required. Is it enough that atoms are billiard balls? Do we need charge? How many atoms? Do we want to see a phase transition?

NB. more accurate = slow and hard, less accurate = fast and easy.

Summary

A simulation = a physical system + a model + a mathematical algorithm for generating states of the model + observables that correspond to physical properties that can be measured.

Units

Physical quantities have units (Mass, Length, Time) that define scales of interest in a system.

$k_BT = 4.14\ 10^{-21}$ J ~ 0.026 eV ~ 1/40 eV

4 pN.nm ~ 1 k_BT

1 Mole / litre ~ 0.6 molecules/nm^3

mass of e^- ~ 0.511 MeV

mass of CH_4 ~ 16 gm/mol ~ 2.6 e^{-23} gm

Computers know nothing of units; all quantities are dimensionless; all equations are discrete; all numbers are integers even when they're real, the same program run on different platforms (Windows, Linux, Mac) *will* produce different results.

This means that in a simulation we are explicitly (or implicitly) converting all dimensional quantities into dimensionless ones by multiplying/dividing by some standard M, L, T scales.

Implicit because if you forget the units, or get them wrong, the simulation will often happily continue, and produce rubbish, but it won't tell you.

Reduced units

In MD (and, in fact, all particle-based simulations), once we have values for a mass m_0 (e.g., one atom), length r_0 (e.g., diameter of one atom) and energy (or temperature k_BT), we can make all other physical quantities dimensionless:

Reduced quantity = function of physical quantities

Mass $m = M/m_0$
Time $t = T/t_0$
Length $l = L/r_0$
Area $a = A/r_0^2$
Volume $v = (L/r_0)^3$
Density $\rho = \text{Density}.r_0^3/m_0 = N / l^3$
Diffusion constant $D' = (D.\, t_0/r_0^2)$
Area per lipid $a_{Lipid} = A/(N.\, r_0^2)$

The benefit of this is that we reduce the range of physical quantities (imagine simulating H with a mass $1.67\ 10^{-27}$ Kg) and can represent many physical systems by one simulation.

(Blackboard: Time ≡ Energy, what is the meaning of the time-scale $t_0 = \sqrt{(m.r_0^2 / k_BT)}$)

Anatomy of a simulation

Setting up a simulation requires specifying precisely the **system** and its **surroundings**

	What physical system do I want to simulate
System	What do I want to learn about it?
State	What length and time scales are important? (nm or mm? km for weather sims)
Boundary conditions	What are the boundary conditions?
Initial conditions	
	What are the entities of interest? atoms, molecules, etc.
Interactions	How do they interact?
Equations of motion	How does the simulation evolve?
Observables	What accuracy do I want? (larger system/longer run may be better, but more expensive)

Usually we are interested in the *thermodynamic* properties of a system, so we connect the simulation to experiments via *thermodynamic coordinates*: temperature, pressure, volume, work, heat, electric field, charge, force, area, length, etc.

We do not simulate a real system, but only a model of a real system; we first construct the model (particles + forces) and second adapt it (discretize Newton's laws) for calculation on a computer.

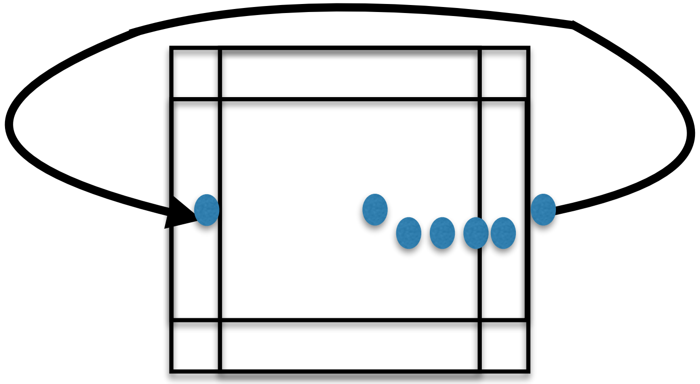
System, State, Boundary Conditions

Suppose our *system* is a fluid, a *state* is defined by giving each particle a mass, x(t), v(t), F(t), …
and we need a box to contain them: but what happens at the walls? We need *boundary conditions*

We would like to simulate a macroscopic system (10^{22} particles) so we can do thermodynamics, but we only have ~ 10^6 - 10^8 particles in an MD simulation. There are two choices:

Hard boundaries - isolated system, large influence of walls on bulk

Periodic boundaries - infinitely periodic system, no walls at all!



If $x(t + dt) > L$ then $X(t + dt)$ -= L

else if $x(t + dt) < 0$ then $X(t + dt)$ += L

and for y(t), z(t)

Typically, PBCs are used to reduce edge effects: if a particle leaves the simulation box across a face/edge/corner it is translated to the opposite face/edge/corner with same velocity. This can lead to spurious forces due to multiple images of the simulation box.

Newton's laws have translational invariance for central forces (with minimum image convention):

$$F(x+L) = m\, d^2(x+L)/dt^2 = m\, d^2x/dt^2 = F(x)$$

Initial conditions

We must specify the *initial conditions* of the system.

The thermodynamic ensemble of our simulation determines the initial conditions:

Microcanonical - N, V, E constant

Canonical - N, V, T constant

Grand Canonical - μ, V, T const

Typically this means placing the particles in suitable positions, giving each a velocity drawn from a Maxwell distribution, and specifying their initial forces subject to physical constraints (i.e., atoms in a molecule should not be separated by more than their average bond length nor be overlapping)

In most cases, a simulation needs time to *forget* the influence of the non-equilibrium initial state before we can start to sample observables in equilibrium.

Sometimes, the initial phase of relaxation to equilibrium is also of interest.

Impact of initial conditions

A system may not relax to equilibrium, or it relaxes very slowly, because of:

A) **Conservation of some quantity:** If the particles in the initial state have a net momentum, and the MD integrator conserves the total momentum, the whole system will translate for ever. Typically, the initial velocities of all particles are chosen so that the CM momentum is zero.

B) **Energy barriers/Metastability:** Self-assembly of lipid molecules into a membrane is very slow as the molecules initially form micelles that have an energy barrier against merging to form a single bilayer that spans the simulation box.

C) **Boundary conditions:** If we are investigating a phase transition or the evolution of a symmetric phase (lamella, hexagonal, etc), the shape of the simulation box may artificially stabilise one or other phase and not allow the system to sample correctly its phase space.

Instead of a random initial condition that is expected to give rise to an ordered state, we can alternatively construct an ordered initial state and use the initial phase of the simulation to relax this state to its equilibrium state.

Interactions

Interactions are defined by a *force field* that specifies the force between any two particles as a function of their position (and, sometimes, velocity): always approximate!

Force fields can have many different terms depending on the interactions between the particles and the length and time scale of the simulation. In terms of their complexity, we have:

all-atom MD > coarse-grained MD >> DPD ~ Brownian Dynamics > MC

Hard spheres = billiard balls

Ising spin model of ferromagnet

Ball and spring model of a membrane

Lennard-Jones particles

Bond forces within polymers

Bending potential "

Torsion potential "

Hydrogen bonds, polarization, dipolar forces, ..

Interactions

Lennard-Jones potential
(non-bonded particles)

$$V_{\mathrm{LJ}} = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Electrostatic force between
charged particles

$$F = k_e \frac{q_1 q_2}{r^2},$$

where k_e is Coulomb's constant ($k_e = 8.99 \times 10^9\ \mathrm{N\ m^2\ C^{-2}}$),

General Amber MD force field
(GAFF)
J. Comput. Chem. 25:1157-1174 (2004)

$$E_{\mathrm{pair}} = \sum_{\text{bonds}} k_r(r - r_{\mathrm{eq}})^2 + \sum_{\text{angles}} k_\theta(\theta - \theta_{\mathrm{eq}})^2 + \sum_{\text{dihedrals}} \frac{v_n}{2}$$

$$\times [1 + \cos(n\phi - \gamma)] + \sum_{i<j}\left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\varepsilon R_{ij}}\right]$$

Martini coarse-grained MD force field
J. Phys. Chem. B 111:7812-7824 (2007)

$$U_{\mathrm{LJ}}(r) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^{6}\right] \qquad U_{\mathrm{el}}(r) = \frac{q_i q_j}{4\pi\epsilon_0\epsilon_r r}$$

$$V_{\mathrm{bond}}(R) = \frac{1}{2}K_{\mathrm{bond}}(R - R_{\mathrm{bond}})^2 \qquad V_{\mathrm{id}}(\theta) = K_{\mathrm{id}}(\theta - \theta_{\mathrm{id}})^2$$

Considerations for integration schemes

- How many times must we evaluate F(x)? Computational cost
- At how many time points must x(t), v(t) be known? Memory cost
- How big can Δt be? Stability
- How accurate is the integration scheme? Truncation and round-off errors
- Are the forces truncated in space (e.g., electrostatics)?

- Δt must be small enough so that neglected terms in the Taylor series of x(t) are small compared to the terms kept: for Euler the first neglected term is $O(\Delta t^2)$ so method is 2nd order; f or Verlet, method is 4th order.

- Many other methods exist of higher order or different types (Haile, Sect. 4.4, pp157 ff)

- Euler/Verlet require only one calculation of force for each particle - this is the best we can do

- Some forces in the "force field" may require much smaller dt than others, making it inefficient to calculate all forces every time step.

Euler algorithm

Simplest method of solving F = ma: a first-order approximation.

Consider the Taylor series for the position x(t) and velocity v(t) of a particle in 1d:

$$x(t + dt) \sim x(t) + dx/dt * dt + d^2x/dt^2 * dt^2/2 + \ldots$$

$$v(t + dt) \sim v(t) + dv/dt * dt + d^2v/dt^2 * dt^2/2 + \ldots$$

Now dx/dt = v(t) and dv/dt = a(t) = F(t)/m, and assume the dt^2 term is so small compared to the x, dx/dt terms that we can ignore it during the time step dt (equivalent to velocity being constant during dt):

$$\boxed{\begin{aligned} x(t + dt) &\sim x(t) + v(t) * dt + O(dt^2) + \ldots \\ v(t + dt) &\sim v(t) + (F(t)/m) * dt + O(dt^2) + \ldots \end{aligned}}$$

Algorithm: given x(t), v(t) at one time use these equations to iterate forwards in time.

Verlet algorithm

Euler's algorithm has an error of order dt^2, because we neglected this term in the Taylor series. If we keep more terms, we get a more accurate trajectory.

Consider the same Taylor series for $x(t)$ but expand it for $x(t + dt)$ and $x(t - dt)$:

$$x(t + dt) \sim x(t) + dx/dt * dt + d^2x/dt^2 * dt^2/2 + d^3x/dt^3 * dt^3/6 + O(dt^4)$$

$$x(t - dt) \sim x(t) - dx/dt * dt + d^2x/dt^2 * dt^2/2 - d^3x/dt^3 * dt^3/6 + O(dt^4)$$

now add them together (and write $d^2x/dt^2 = a(t) = F(t)/m$):

$$x(t + dt) \sim 2 * x(t) - x(t - dt) + (F(t)/m) * dt^2 + O(dt^4) + \ldots$$

$$v(t) \sim (x(t + dt) - x(t - dt))/2*dt$$

Algorithm: this is more accurate than Euler as the truncation error is $O(dt^4)$, and given $x(t)$ at two times we use these equations to iterate forwards in time. Small problem is that $v(t)$ has error $O(dt^2)$, but method does not need $v(t)$ to calculate trajectory.

Observables

Given the set $\{\mathbf{x}_i(t), \mathbf{v}_i(t)\}$ for all the particles, we can calculate any observable property of the system by integrating over the phase space trajectory of the particles.

e.g. Temperature = $k_B T \sim < 1/2m.v_i^2>$

But adjacent states are typically highly correlated so we have to simulate for a long time to get good statistics, and allow long gaps between samples to have independent measurements.

How do we know if our samples are independent?

We can calculate the auto-correlation function (= 2-point correlation function) of the observable of interest

$$C_2(\tau) = (\langle O(t + \tau).O(t)\rangle - \langle O(t)\rangle^2) \;/\; (\langle O(t)^2\rangle - \langle O(t)\rangle^2)$$

The number of time steps between samples must be at least as large as the *longest* correlation time of the observables of interest. Note that different observables can have different correlation times, so we cannot measure just one time period and use it for all observables.

Collective properties (CM of a membrane, thickness fluctuations, surface tension) have longer correlation times than single particle properties (lipid tail length, velocity)

Precision, error and truth

Experiments have external influences, e.g., temperature fluctuations, dirt, admixtures, etc.

A simulation has **approximations**, **systematic** errors, and **statistical** errors

Systematic Errors

Initial state
Finite system size
Truncation error (missing terms in Taylor series)
Round-off error (machine precision, sqrt, order of calculations)
Random number generator isn't
Bugs in the code

Statistical Errors

Too few samples
Samples too close together
Correlations
Stuck in metastable state

Any simulation that does not quantify and discuss the errors due to finite system size, finite run length, the influence of the initial state, boundary conditions, etc, is useless.

Reproducible ≠ Right.

Truth?

Simulations are *approximations* to the truth.

It may be hard to define precisely what is a **State** or to quantify the effects of the **BCs**, and once they are defined it may be hard to represent them accurately on a computer.

The degree to which our *Model* corresponds to reality (or, how accurate our simulation is) follows from how accurately the mathematical properties of our model represent the behaviour of the real entities; or, how much of the physics of an experiment is captured in the simulation:

e.g., a RW captures a coin tossing experiment very well

an MD simulation of a drug molecule binding to a protein is not so accurate.

Pros and Cons of Simulations

Advantages

Keeps only “relevant” properties

Exact knowledge of microstates

Equations of motion can be chosen

Effects of each force term can be isolated and studied

Measurements don’t perturb the system

We can always repeat a simulation with exactly the same, or carefully-different conditions

Usually cheaper than experiments

Disadvantages

What is a relevant property?

We lose direct connection to experiments

We see only what we expect to see

One simulation gives one data point, and we may need many points

Force fields can be hard to choose and not *transferable* between similar systems

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

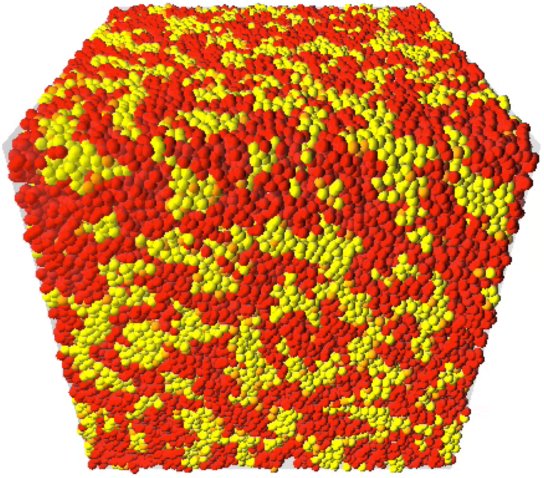
Summary

- Why do simulations work? - the world is divided into quasi-independent layers where entities interact/move according to their own rules
- When don't they work? - errors are systematic/statistical, we cannot always remove them, especially if we leave something out or put it in wrongly
- What can you get from a simulation? - insight and sometimes numbers

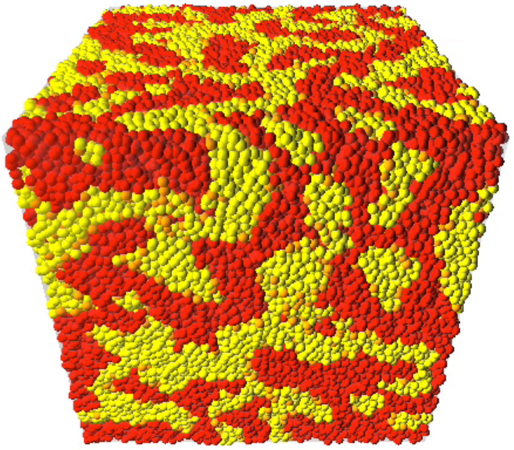
Break

- Qualitative and quantitative DPD applications
- Exercise

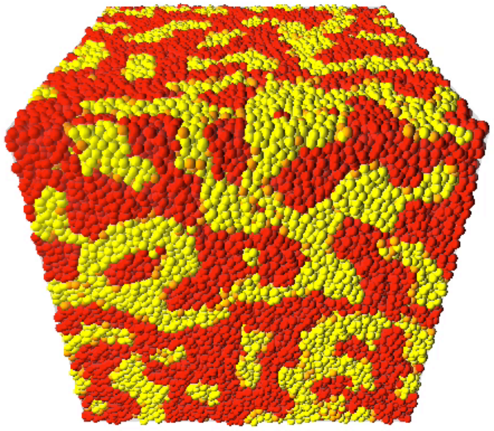
Visualization of simulations - lamella self-assembly



15322k



15350

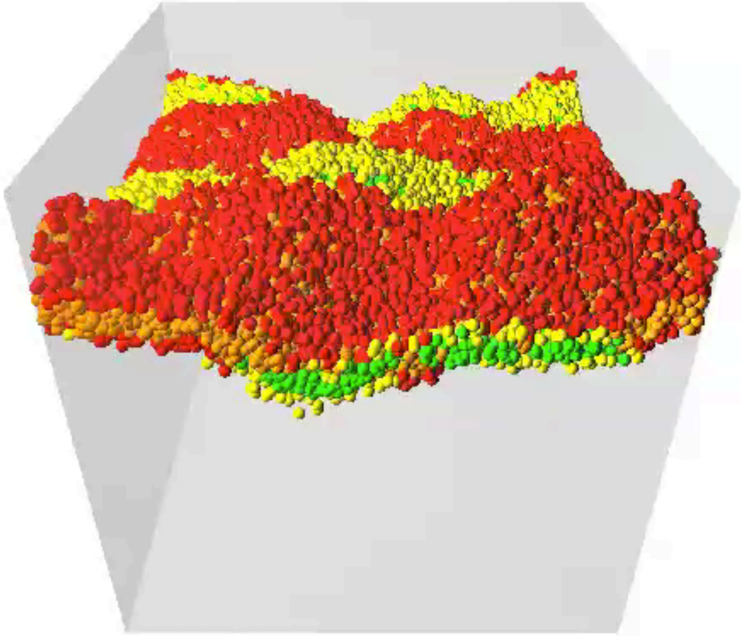


15351

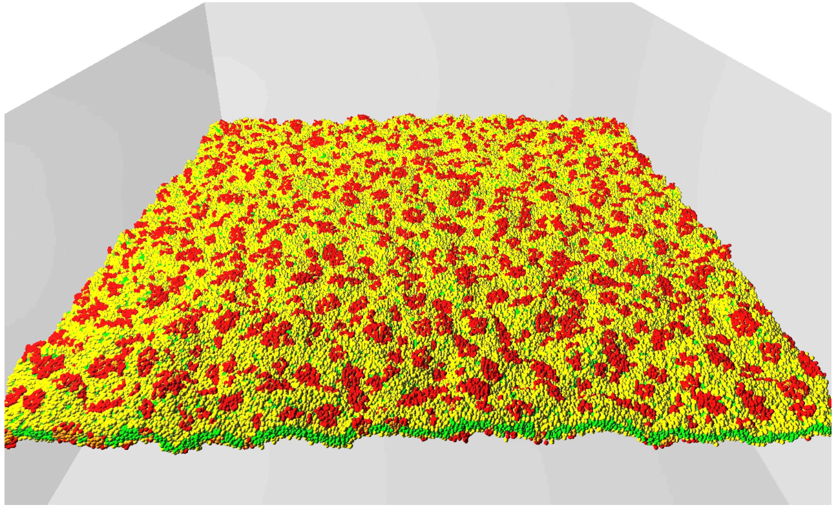
What determines if lamella form at all?

What determines the lamella spacing?

Domain formation in membranes



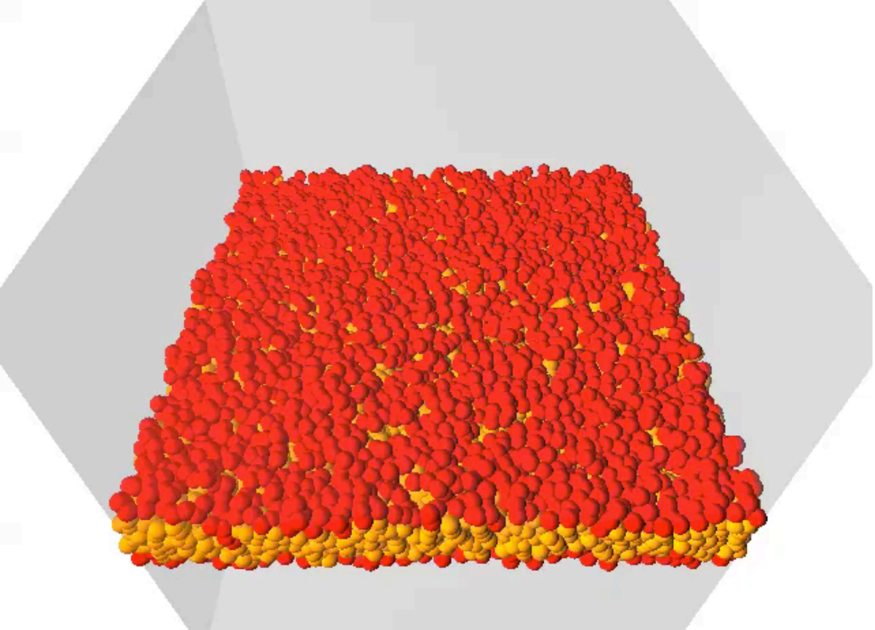
1088



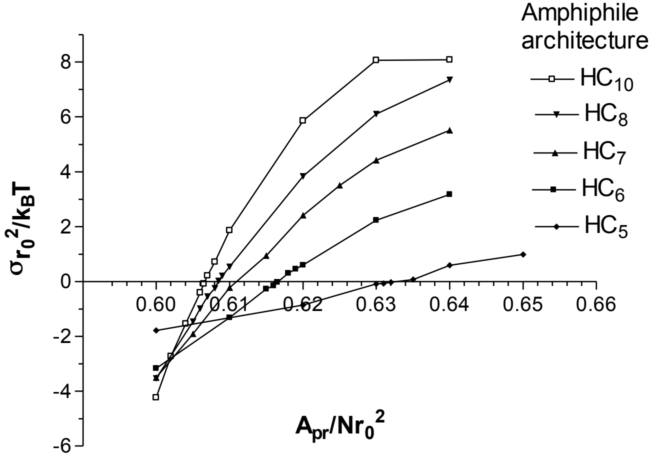
This is run dmpci.1094n
lipids in a box $(50r_0)^3$

Can we go beyond movies?

Quantitative results can be obtained from coarse-grained simulations if it can calibrated appropriately, e.g., domains are controlled by size, line tension, and diffusion.



50d10



Surface Tension = K (A - A$_0$)/A$_0$

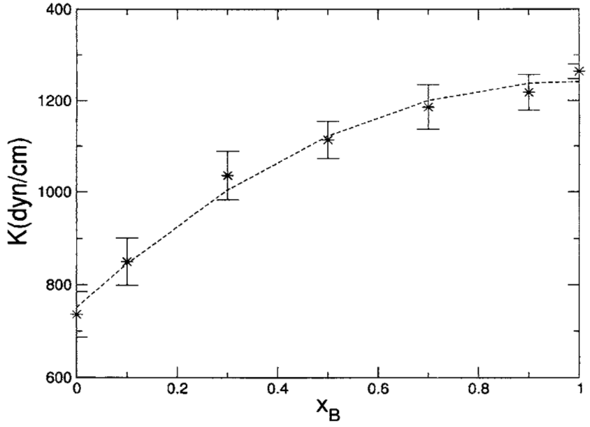
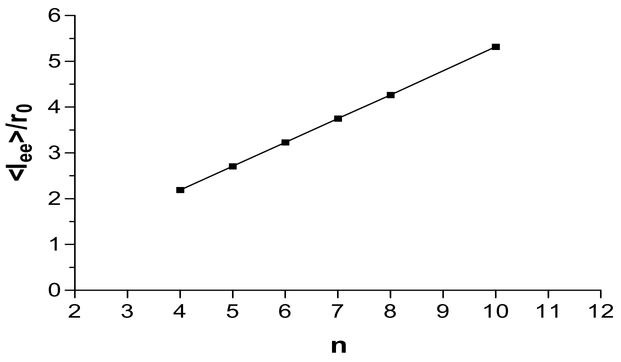


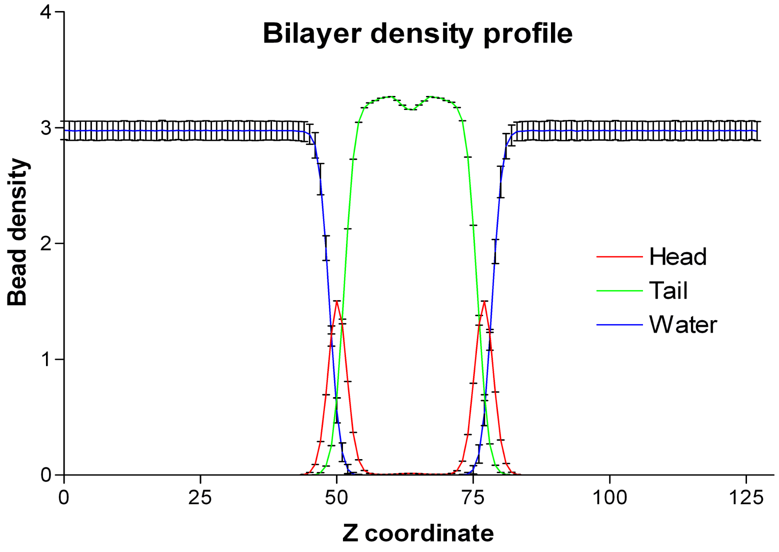
FIG. 4. Bilayer area stretch modulus as a function of amphiphile *B* concentration, mixed with amphiphile *A*. The dashed line is a quadratic fit of the area stretch modulus.

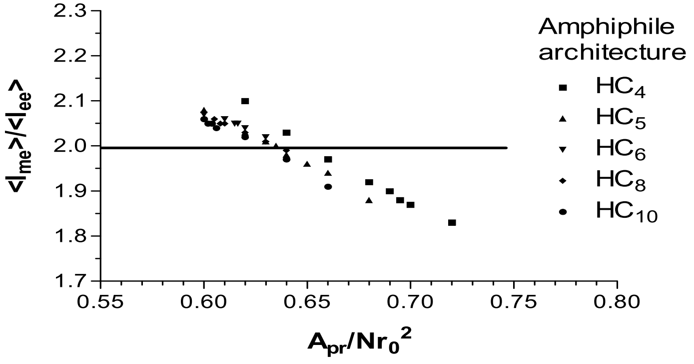
What determines the membrane bending and stretching moduli?

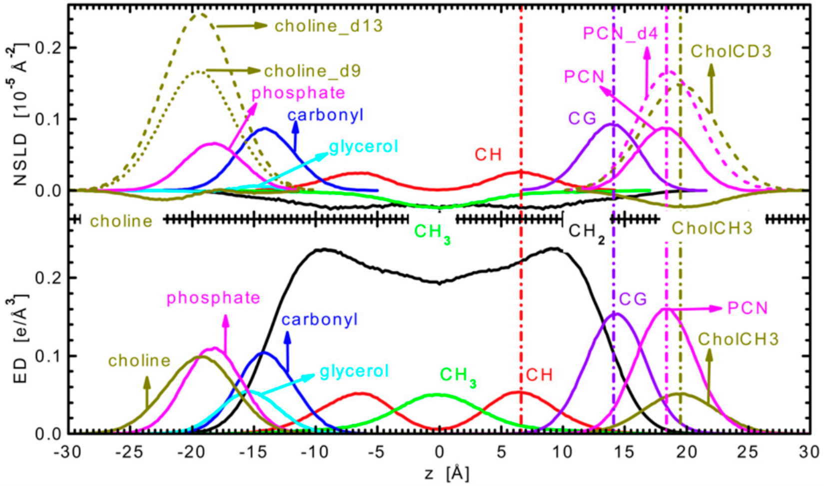
Membrane structure

l_{ee} is linear in tail length







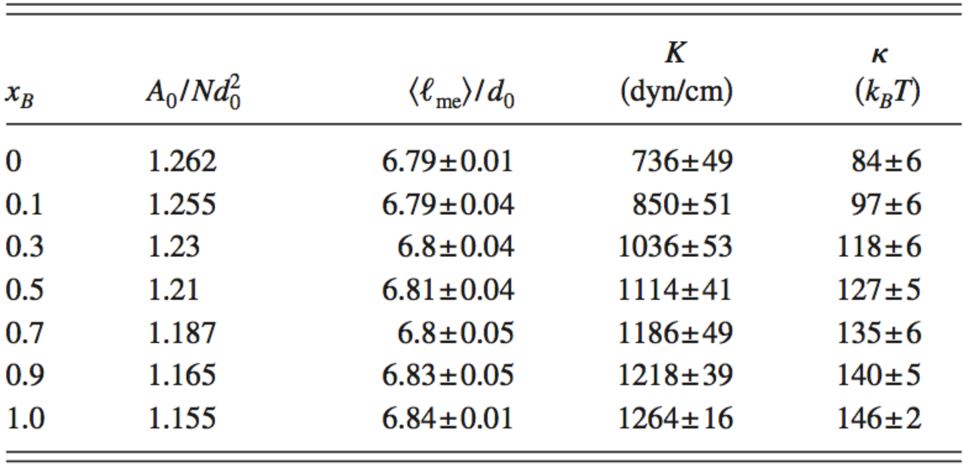


Kucerka, N et al. Biophys. J. 95:2356 (2008)

Membrane thickness is $l_{me} \sim 2. l_{ee}$ for small area expansions

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

Domain material properties



x_B	A_0/Nd_0^2	$\langle \ell_{me} \rangle / d_0$	K (dyn/cm)	κ (k_BT)
0	1.262	6.79±0.01	736±49	84±6
0.1	1.255	6.79±0.04	850±51	97±6
0.3	1.23	6.8±0.04	1036±53	118±6
0.5	1.21	6.81±0.04	1114±41	127±5
0.7	1.187	6.8±0.05	1186±49	135±6
0.9	1.165	6.83±0.05	1218±39	140±5
1.0	1.155	6.84±0.01	1264±16	146±2

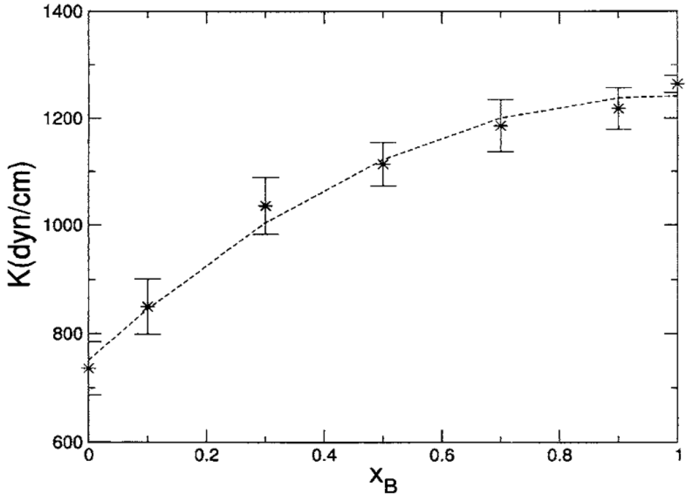


FIG. 4. Bilayer area stretch modulus as a function of amphiphile *B* concentration, mixed with amphiphile *A*. The dashed line is a quadratic fit of the area stretch modulus.

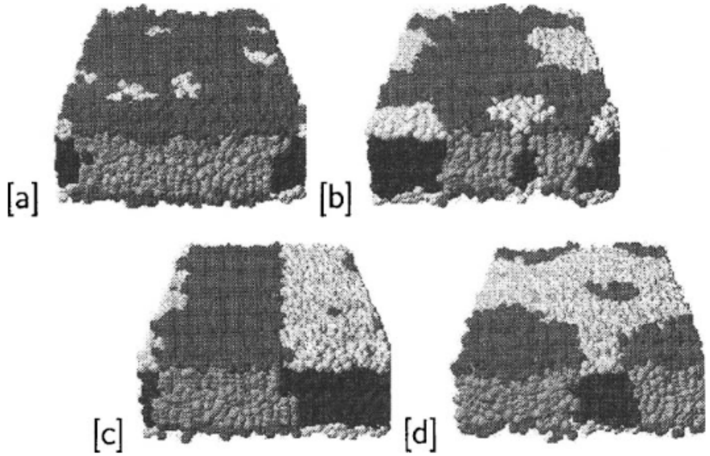


FIG. 3. Phase separation of amphiphiles *A* (denoted by red heads) and *B* (denoted by yellow heads) for various mixture compositions in the tensionless state, at 400 000 DPD steps [(a) x_B=0.1, (b) x_B=0.3, (c) x_B=0.5, and (d) x_B=0.7]. The simulation box and water beads are invisible for clarity.

Illya et al., J Chem. Phys. 125:114710 (2006)