Accelerating Mesoscale Molecular Simulation using CUDA and MPI

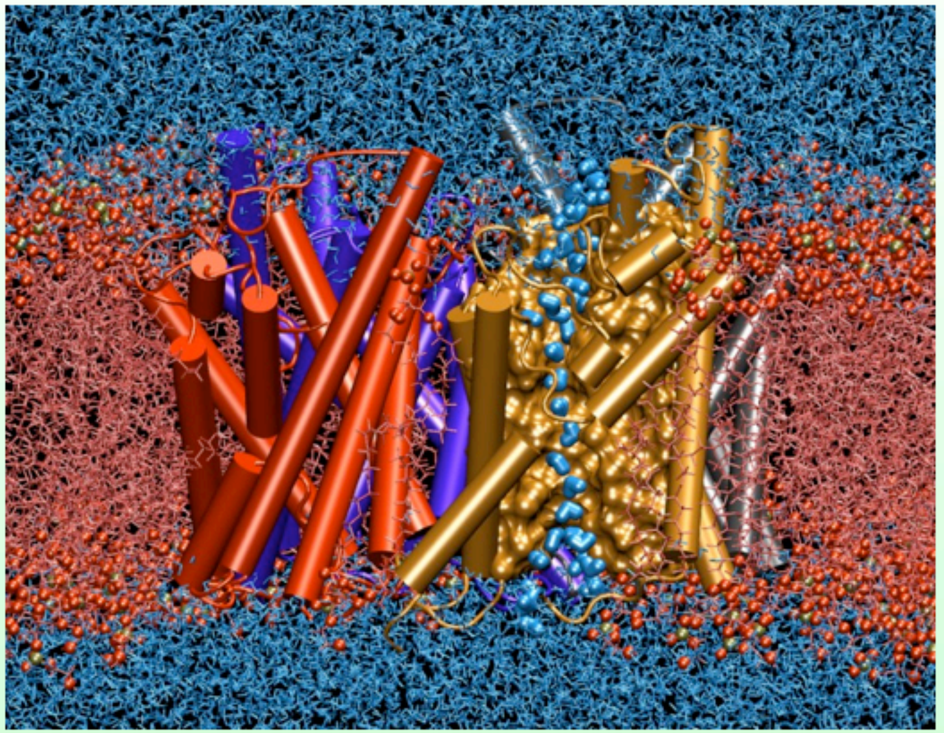
Christopher Ryan
Biophysics Graduate Group

CS267 poster / talk
May 11th 2009

Molecular dynamics (MD): Intrinsic strengths and weaknesses

Strengths
- All-atom representation gives high chemical detail.
- Quantitative information (in physical units) can be gathered about a system.
- Many standard tools (including parallel ones) have been refined over the course of many years.

Weaknesses
- Despite much progress, this method is still **prohibitively slow** to address many outstanding research questions. (e.g., large scale processes that happen on the order of milliseconds and involve millions of atoms).



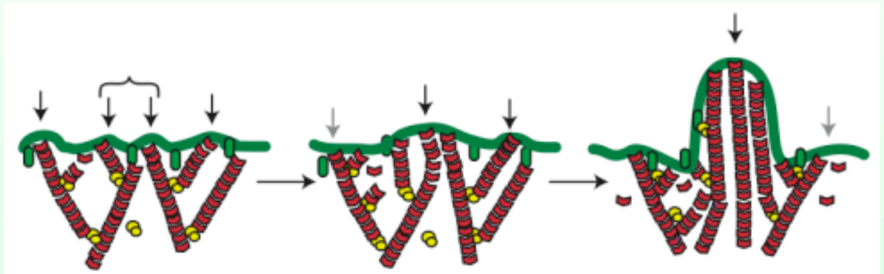
Membrane-bound aquaporin (i.e., water channel)
timescale: 10^{-9} seconds

Jensen MO, Tajkhorshid E, Schulten K.
Biophysical Journal, 85:2884-2899, 2003.

Many important processes occur at long time- and length-scales

actin-based motility

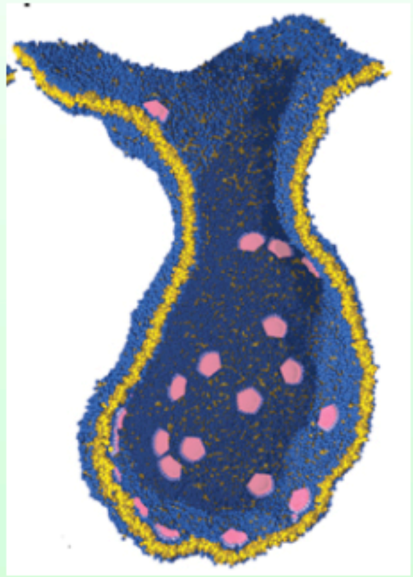
10^{-3} - 10^{-1} seconds



Liu A, Richmond D, Maibaum L, Pronk S, Geissler PL, Fletcher D. *Nature Physics*, 4 (2008)

endocytosis

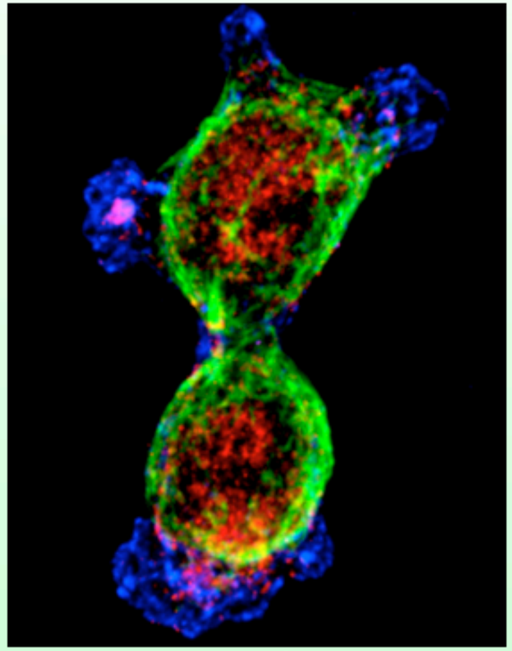
10^{-3} - 10^{0} seconds



Reynwar BJ, Illya G, Harmandaris VA, Müller MM, Kremer K, Deserno M. *Nature*. 2007 May 24.

cytokinesis

> 10^{0} seconds



Pellinen T *et al*. *Dev Cell*. 2008 Sep 15(3).

Mesoscale molecular simulation with Dissipative particle dynamics (DPD)

• First devised in the early 1990s by Hoogerbrugge and Koelman [1], later revised by Espanol to ensure proper thermodynamics [2].

• It is a method of "coarse grained" simulation -- atomistic details of the solvent and other molecules are abstracted to yield larger particles (see the next slide for an example).

• Based on the laws of stochastic fluid dynamics (e.g., Brownian motion) to give correct hydrodynamic behavior at long time- and length-scales.

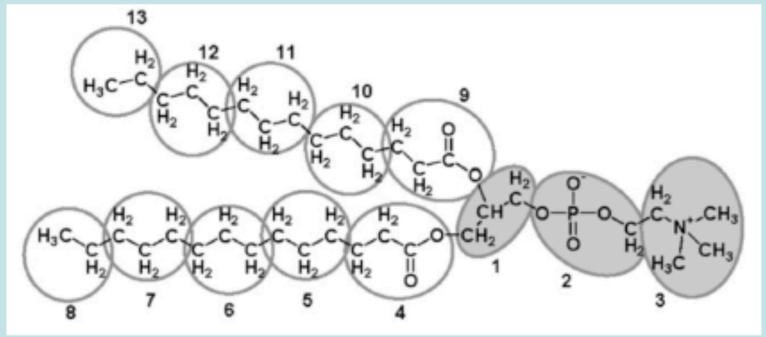
• The technique is becoming increasingly popular in materials and biological sciences.

$$\mathbf{F}_i = \sum_{j \neq i} \left[\mathbf{f}^C(\mathbf{r}_{ij}) + \mathbf{f}^D(\mathbf{r}_{ij}, \mathbf{v}_{ij}) + \mathbf{f}^R(\mathbf{r}_{ij}) \right]$$

pair potential friction force random force

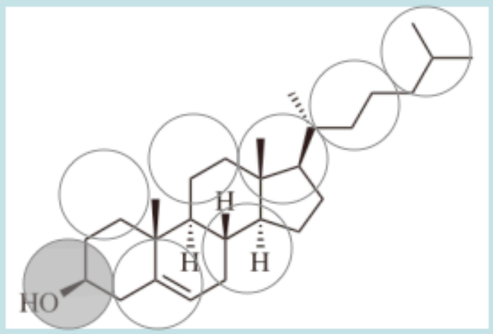
[1] Hoogerbrugge PJ and Koelman MVA. Europhysics Letters, 19(3):155–160, 1992.
[2] Espanol P and Warren PB. Europhysics Letters, 30(4):191–196, 1995.

An example of DPD coarse-graining: biological membranes

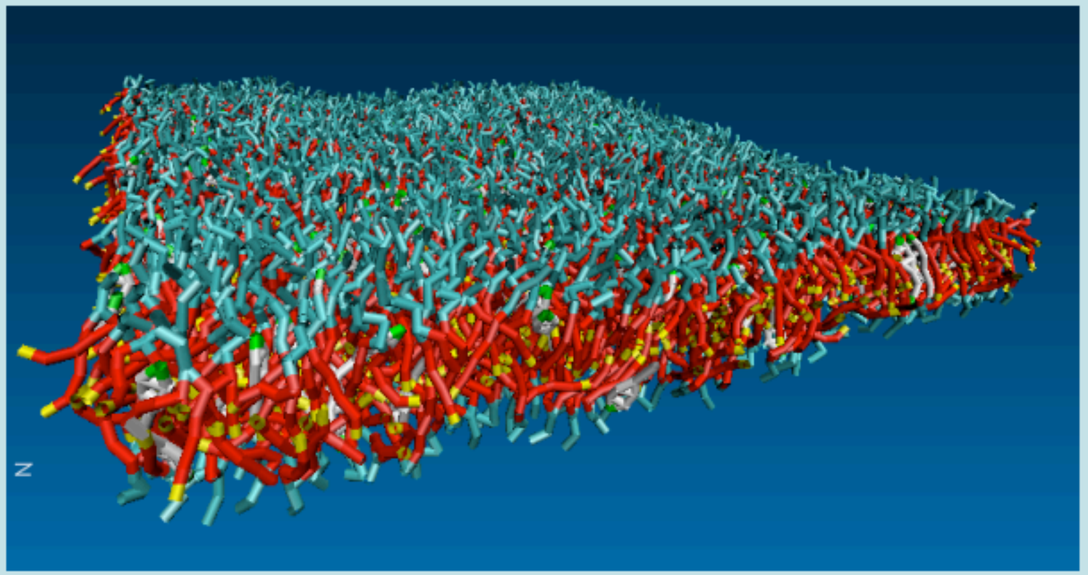


phospholipid

+



cholesterol



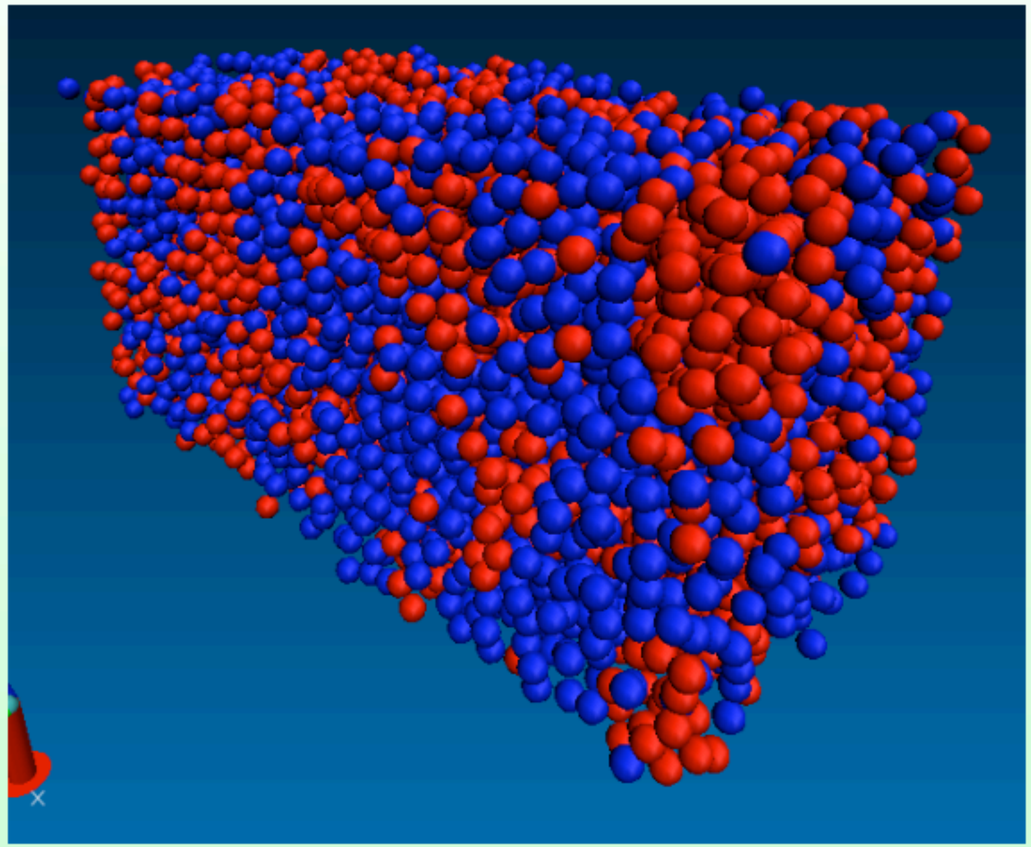
- 3600 phospholipid / cholesterol molecules
- 90000 water particles (3 water molecules per particle)

The model system: mixing and phase separation in a 2-component liquid

- Adapted from *Understanding Molecular Simulation* by Daan Frenkel & Berend Smit [1].

- Particles interact with each other by a soft repulsive force.

- Numerical physical quantities can be obtained (e.g., density profile along the z-axis, interfacial tension, pressure, temperature, etc).

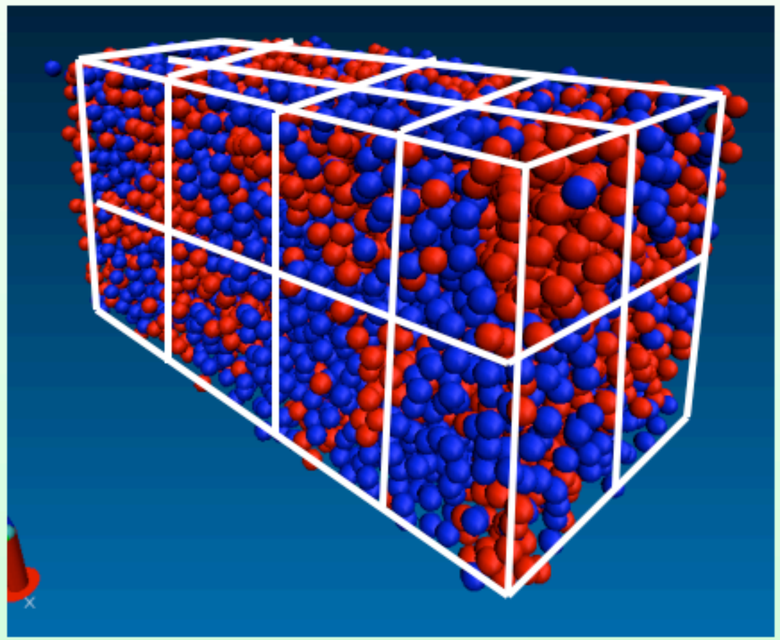


[1] Frenkel D, Smit B. Understanding molecular simulation: from algorithms to applications. Academic Press, 2002. ISBN 0122673514

General strategies for parallelization

• **Decompose the simulation box into equidistributed domain sizes.** Since the system is in a condensed phase (unlike in HW 2), work can be distributed approximately evenly without a quad tree. The size of the dimension of each box will approximately equal the cutoff distance.

• **Let each core be responsible for calculating the force on each particle within it's designated volume.** This is the most computationally expensive part of the algorithm by which the system evolves.



evenly-sized domains
(note: this scale is much smaller than what will be implemented in the final version.)

CUDA implementation

• **Graphics Card**. Nvidia 9500 GT (fairly modest). 8-way SIMD on 4 “multiprocessors,” 8 kb of dedicated memory (I think). Running on Ubuntu Linux.

• **Memory concerns will likely dominate performance tuning**. It will may challenging to store and cheaply update the particle position array on the dedicated constant cache.

• **Multidimensional matrices should not be an issue**. Though a sparse matrix of particle interactions *could* be made, only the sum of these forces needs to be stored (so no need for concern about matrix padding).

MPI implementation

• Available parallel computing resources include the NERSC computers and an 8-core Mac Pro.

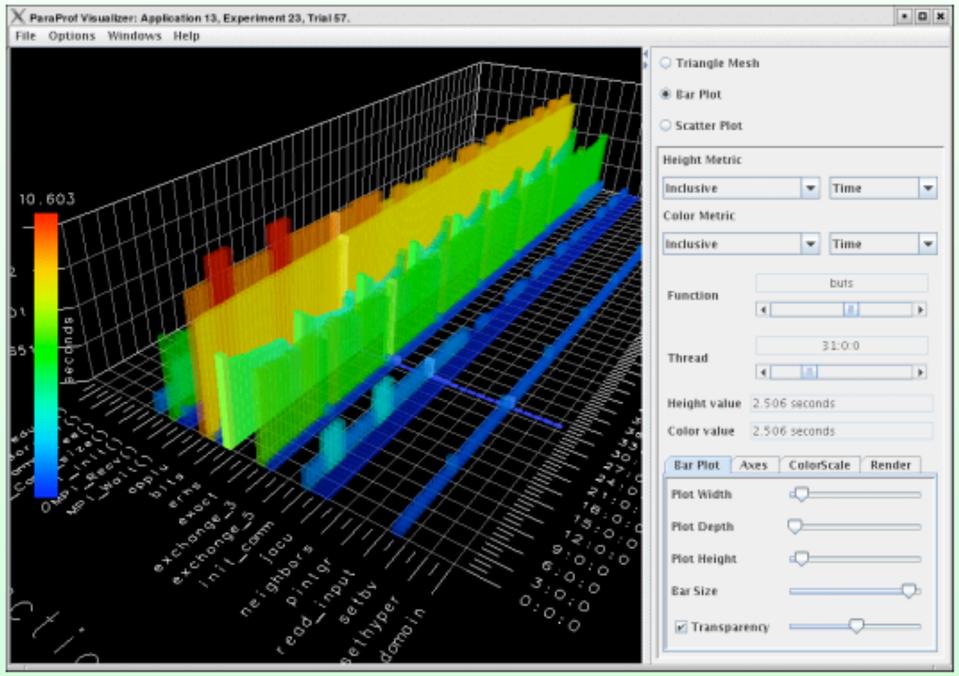
• In comparison to the CUDA implementation, the MPI implementation should afford larger domains. I expect that one domain size will be more “natural” than the other, since particles feel pair-potential and dissipative forces of each other as given by the energy function.

• Some similar techniques to those used in HW2 could be used, however since particle impose forces on each other (even over domain boundaries) the problem is particularly different.

Other tools that I hope to use

- Performance analysis tools (Karl Fuerlinger, 3/18/2009):
  - IPM -- for MPI version
  - TAU -- for both (?)

- An OpenMP version (if I *really* have extra time).



3D performance data from TAU

Thanks!

- Jocelyn Rodgers
- The membrane simulation subgroup
- Professor Smit

- UC Berkeley biophysics graduate group