Molecular Simulation Background

Why Simulation?

1. Predicting properties of (new) materials
2. Understanding phenomena on a molecular scale
3. Simulating known phenomena **?**

Example: computing the melting point of ice

Why bother?

This works better.

Why Simulation?

1. Predicting properties of (new) materials
2. Understanding phenomena on a molecular scale.
3. ~~Simulating known phenomena.~~

The Monte Carlo Method

Aim: to compute thermal averages of equilibrium systems.

$$\langle A \rangle = \frac{\sum_i \exp(-\epsilon_i / k_B T) A_i}{\sum_i \exp(-\epsilon_i / k_B T)}$$

Where *i* labels all eigenstates of the system, and

$$A_i = \langle i | A | i \rangle$$

Classical limit: replace the SUM over quantum states by an INTEGRAL of phase space

$$\langle A \rangle = \frac{\int \mathrm{d}\mathbf{p}^N \mathrm{d}\mathbf{r}^N \; A(\mathbf{p}^N, \mathbf{r}^N) \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}{\int \mathrm{d}\mathbf{p}^N \mathrm{d}\mathbf{r}^N \; \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}$$

Where *H* is the Hamiltonian of the system and β=1/kT

In replacing the sum by an integral, we have attributed a "volume" h^{3N} to every quantum state

Problem:

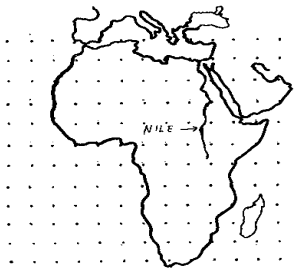
We cannot compute the sum over all quantum states (because there are so many)

And we cannot compute the classical integral either (except the integration over momenta).

Consider "normal" numerical integration

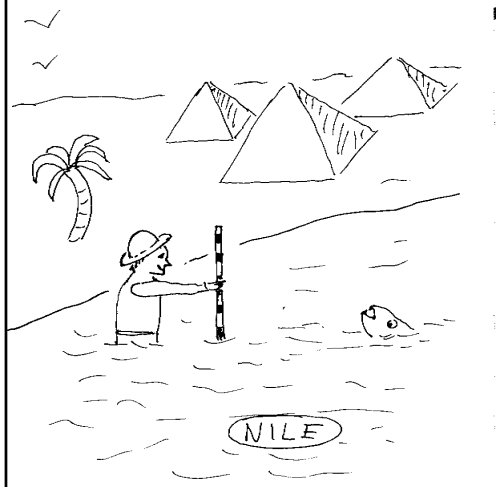
100 particles, 3 dimensions, 10 points in every direction.

Requires 10^{300} points for a very poor estimate of the integral…



Similar problem (but much less serious):

Measure the depth of the Nile by quadrature…



BETTER STRATEGY:

IMPORTANCE SAMPLING

We wish to perform a RANDOM WALK in configuration space, such that

The number of times that each point is visited, is proportional to its Boltzmann weight.

$$n(\mathbf{r}^N) = c \exp[-\beta \mathcal{U}(\mathbf{r}^N)]$$

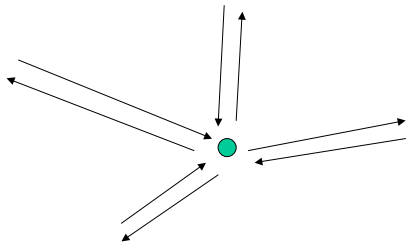
Then

$$\langle A \rangle \approx \frac{1}{L} \sum_{i=1}^{L} n_i \, A(\mathbf{r}_i^N).$$

How do we achieve that?

Whatever our rule is for moving from one point to another, it should not destroy the equilibrium distribution.

That is: in equilibrium we must have



$$\mathcal{N}(o)\sum_n \pi(o\to n) = \sum_n \mathcal{N}(n)\pi(n\to o)$$

Stronger condition:

$$\mathcal{N}(o)\pi(o\to n) = \mathcal{N}(n)\pi(n\to o).$$

For every pair {*n*,*o*}.

Detailed Balance

Now we construct the transition probabilities

$$\pi(o\to n) = \alpha(o\to n)\times \mathrm{acc}(o\to n)$$

Then, detailed balance implies that:

$$\mathcal{N}(o)\alpha(o\to n)\times \mathrm{acc}(o\to n)$$

$$=$$

$$\mathcal{N}(n)\alpha(n\to o)\times \mathrm{acc}(n\to o)$$

Often, we choose

$$\alpha(o\to n) = \alpha(n\to o)$$

Then it follows that

$$\mathcal{N}(o)\times \mathrm{acc}(o\to n)$$

$$=$$

$$\mathcal{N}(n)\times \mathrm{acc}(n\to o)$$

$$\frac{\mathrm{acc}(o\to n)}{\mathrm{acc}(n\to o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp\{-\beta[\mathcal{U}(n)-\mathcal{U}(o)]\}$$

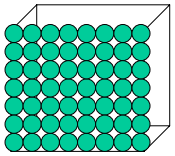
Metropolis, Rosenbluth,Rosenbluth,

Teller and Teller choice:

$$\mathrm{acc}(o\to n) = \min\left(1, \exp\{-\beta[\mathcal{U}(\mathbf{r}'^N)-\mathcal{U}(\mathbf{r}^N)]\}\right)$$

Practical issues:

1. Boundary conditions
2. Reduced units
3. Time-saving devices

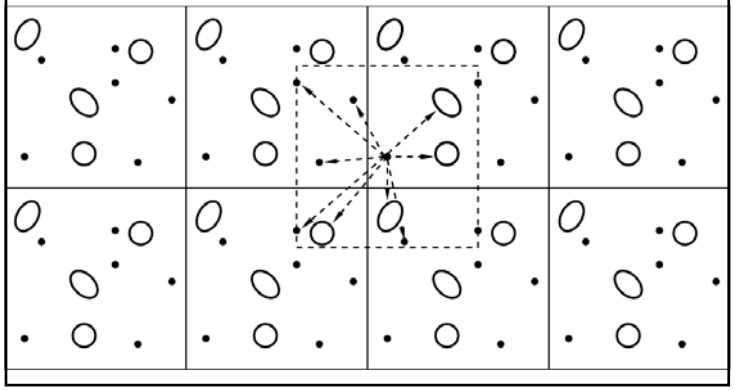


In small systems, boundary effects are always large.

1000 atoms in a simple cubic crystal – 488 boundary atoms.

1000000 atoms in a simple cubic crystal – still 6% boundary atoms…

"Solution" : Periodic boundary conditions



Reduced units

Example: Particles with mass **m** and pair potential:

$$v(r) = \epsilon f(r/\sigma)$$

Unit of length: σ

Unit of energy: ε

Unit of time: $\sigma\sqrt{m/\epsilon}$

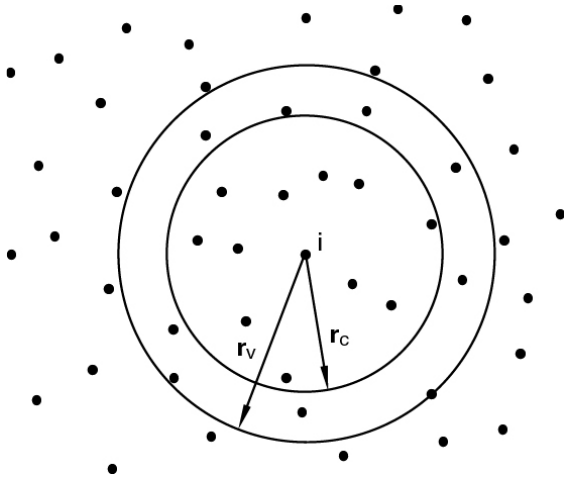
The most time-consuming part of any simulation is the evaulation of all the interactions between the molecules.

In general: $N(N-1)/2 = O(N^2)$

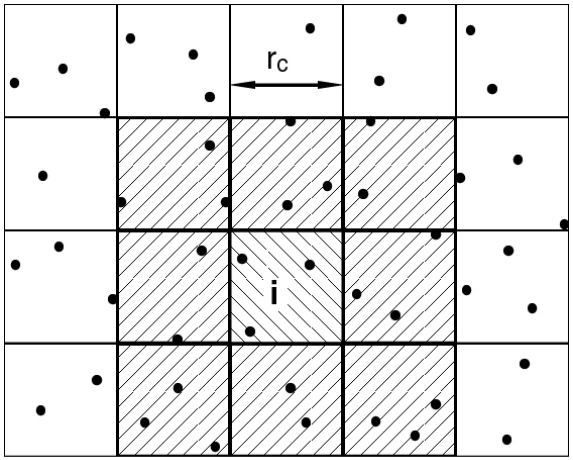
But often, intermolecular forces have a short range:

Therefore, we do not have to consider interactions with far-away atoms…

Verlet list



Link list



NOTE:

Long-ranged forces require special techniques.

1. Coulomb interaction (1/r in 3D)
2. Dipolar interaction ($1/r^3$ in 3D)

…and, in a different context:

1. Interactions through elastic stresses (1/r in 3D)
2. Hydrodynamic interactions (1/r in 3D)
3. …

The shaky foundations of…

Molecular Dynamics

Molecular Dynamics

The Basis:

$F_i = m_i a_i$

i=1,2,...,N

N-body problem. Can only be solved numerically

(except in very special cases)

How?

$$X(t+\Delta t) = X(t) + \dot{X}(t)\Delta t + \frac{1}{2!}\ddot{X}(t)\Delta t^2 + \frac{1}{3!}\dddot{X}(t)\Delta t^3 + \ldots$$

...at least, in principle.

Naive approach: truncate Taylor expansion.

$$X(t+\Delta t) \approx X(t) + \dot{X}(t)\Delta t + \frac{1}{2!}\ddot{X}(t)\Delta t^2$$

ABSOLUTELY FORBIDDEN!

The naive “forward Euler” algorithm

- is not time reversible
- does not conserve volume in phase space
- suffers from energy drift

Better approach: “Verlet” algorithm

$$X(t+\Delta t) = X(t) + \dot{X}(t)\Delta t + \frac{1}{2!}\ddot{X}(t)\Delta t^2 + \frac{1}{3!}\dddot{X}(t)\Delta t^3 + \frac{1}{4!}\ddot{\dot{X}}(t)\Delta t^4 + \ldots$$

$$X(t-\Delta t) = X(t) - \dot{X}(t)\Delta t + \frac{1}{2!}\ddot{X}(t)\Delta t^2 - \frac{1}{3!}\dddot{X}(t)\Delta t^3 + \frac{1}{4!}\ddot{\dot{X}}(t)\Delta t^4 + \ldots$$

+

$$X(t+\Delta t) + X(t-\Delta t) = 2X(t) + \ddot{X}(t)\Delta t^2 + O(\Delta t^4)$$

or

$$\boxed{X(t+\Delta t) \approx 2X(t) - X(t-\Delta t) + \ddot{X}(t)\Delta t^2}$$

Verlet algorithm

Verlet algorithm

- is time reversible
- does conserve volume in phase space
- (is “symplectic”)
- does not suffer from energy drift

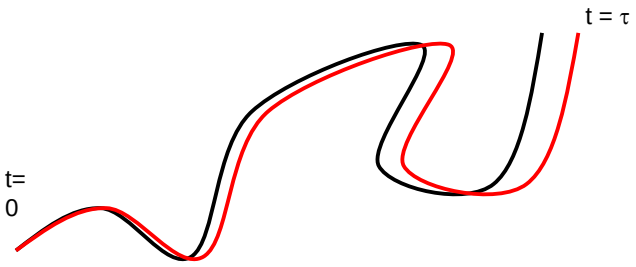
...but is it a good algorithm?

i.e. does it predict the time evolution of the system correctly???

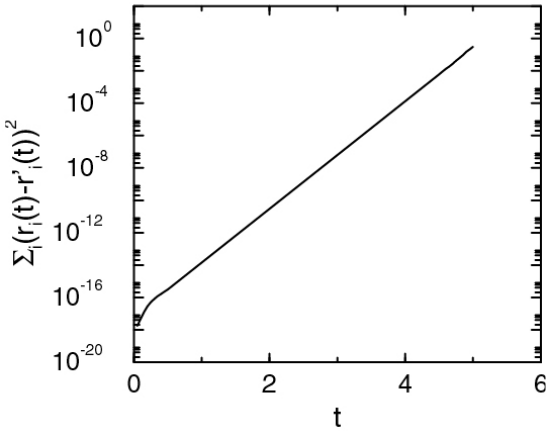
Dynamics of “well-behaved” classical many-body system is chaotic.

Consequence:

Trajectories that differ very slightly in their initial conditions DIVERGE EXPONENTIALLY (“Lyapunov instability”)



The Lyapunov disaster in action...



Any small error in the numerical integration of the equations of motion, will blow up exponentially....

always...

...and for any algorithm!!

SO:

Why should anyone believe Molecular Dynamics simulations ???

Answers:

1. In fact, one should not...

exit Molecular Dynamics...

Answers:

1. In fact, one should not...
2. Good MD algorithms (e.g. Verlet) can also be considered as good Monte Carlo algorithms – they therefore yield reliable **STATIC** properties ("Hybrid Monte Carlo")

What is the point of simulating dynamics, if we cannot trust the resulting time-evolution???

Answers:

1. In fact, one should not...
2. Good MD algorithms (e.g. Verlet) can also be considered as good Monte Carlo algorithms – they therefore yield reliable **STATIC** properties ("Hybrid Monte Carlo")
3. All is well (probably), because of...

The Shadow Theorem....

For any realistic many-body system, the shadow theorem is merely a hypothesis.

It states that (my words):

Good algorithms generate numerical trajectories that are "close to" a REAL trajectory of the many-body system.

Question:

Does the Verlet algorithm indeed generate “shadow” trajectories?

Take a different look at the problem.

Do not discretize NEWTON’s equation of motion...

...but discretize the **Lagrangian Equations of Motion**

Intermezzo:

Classical mechanics – the Lagrangian approach.

Newton: $\mathbf{F_i} = m_i \ddot{\mathbf{r}}_i$

Lagrange:

Consider a system that is at a point $\mathbf{r}_0$ at time t=0 and at point $\mathbf{r}_t$ at time t=t, then:

The system follows a trajectory r(t) such that:

“Action”

$$S \equiv \int_0^t dt' \mathcal{L}(\mathbf{r}(t))$$ is an extremum.

“Lagrangian”

Where the Lagrangian is defined as:

$$\mathcal{L}(\mathbf{r}(t)) = T_{kinetic} - U_{pot}(\mathbf{r})$$

For example, if we use cartesian coordinates:

$$\mathcal{L}(\mathbf{r}(t)) = \sum_{i=1}^{N} \frac{1}{2} m_i \dot{\mathbf{r}}_i^2 - U(\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N)$$

What does this mean?

Consider the "true" path **R**(t), with **R**(0)=$\mathbf{r}_0$ andR(t)=$\mathbf{r}_t$.

Now, consider a path close to the true path:

$$\mathbf{r}(t') = \mathbf{R}(t') + \delta\mathbf{r}(t')$$

Then the action **S** is an extremum if

$$\frac{\delta S}{\delta \mathbf{r}(t')} = 0 \quad , (\forall t')$$

(what does this equation mean??)

$$S_{\text{continuous}} = \int_{t_0}^{t_1} dt\, L(t)$$

Discretized version

$$S_{\text{discrete}} = \Delta t \sum_{i=0}^{i_{max}} L(t_i)$$

$$L(t_i) = T(t_i) - U(t_i)$$

e.g. for one coordinate in one dimension

$$L(t_i)\Delta t = \frac{1}{2} m\Delta t \frac{(X_{i+1} - X_i)^2}{\Delta t^2} - U(X_i)\Delta t$$

and hence the discretized action is

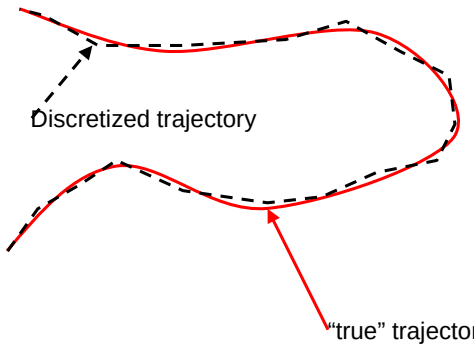
$$S_{\text{discrete}} = \sum_{i=1}^{i_{max}} \left(\frac{m(X_{i+1} - X_i)^2}{2\Delta t} - U(X_i)\Delta t \right)$$

Now do the standard thing:

Find the extremum for small variations in the path, i.e. for small variations in all X_i.

$$\frac{\partial S_{\text{discrete}}}{\partial X_i} = 0 \qquad (\forall i)$$

This will generate a discretized trajectory that starts at time t_0 at X_0 , and ends at time t_1 at X_1.



$$\frac{\partial S_{discrete}}{\partial X_i} = \frac{\partial}{\partial X_i}\sum_{i=1}^{i_{max}}\left(\frac{m(X_{i+1} - X_i)^2}{2\Delta t} - U(X_i)\Delta t\right)$$

$$\frac{\partial S_{discrete}}{\partial X_i} = \frac{-m(X_{i+1} - X_i) + m(X_i - X_{i-1})}{\Delta t} - \Delta t\frac{\partial U(X_i)}{\partial X_i}$$

And hence:

$$0 = \frac{m}{\Delta t}\left(2X_i - X_{i+1} - X_{i-1} - \frac{\Delta t^2}{m}\frac{\partial U(X_i)}{\partial X_i}\right)$$

$$0 = \left(2X_i - X_{i+1} - X_{i-1} - \frac{\Delta t^2}{m}\frac{\partial U(X_i)}{\partial X_i}\right)$$

REWRITE AS:

$$X_{i+1} = 2X_i - X_{i-1} + \frac{\Delta t^2}{m}F(X_i)$$

VERLET!!!

The Verlet algorithm generates trajectory that satisfies the boundary conditions of a REAL trajectory – both at the beginning and at the endpoint.

Hence, if we are interested in statistical information about the dynamics (e.g. time-correlation functions, transport coefficients, power spectra...)

...then a “good” MD algorithm (e.g. Verlet) is fine.