Simulation: A Bluffer's Guide.

A brief journey into the rôle, form and
limitations of computer simulation
in condensed matter physics.

*(distinctly non-exhaustive, rather subjective and
not a lot of quantum mechanics either.)*

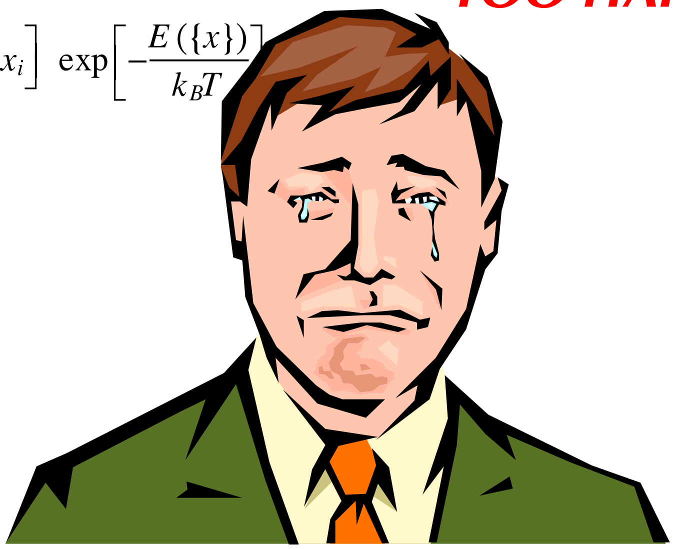
An attempt to place my work into context.

Also, an excuse for using
clip-art gratuitously.

Why Simulate?

$$
\begin{aligned}
E(\{x\}) &= \sum_{i=1}^{N} \sum_{j=i+1}^{N} V(r_{ij}) \\
Z(N, V, T) &= \prod_{i=1}^{3N}\left[\int_{-\infty}^{+\infty} \mathrm{d}\dot{x}_i\right] \prod_{i=1}^{3N}\left[\int_{-\infty}^{+\infty} \mathrm{d}x_i\right] \exp\left[-\frac{E(\{x\})}{k_B T}\right]
\end{aligned}
$$

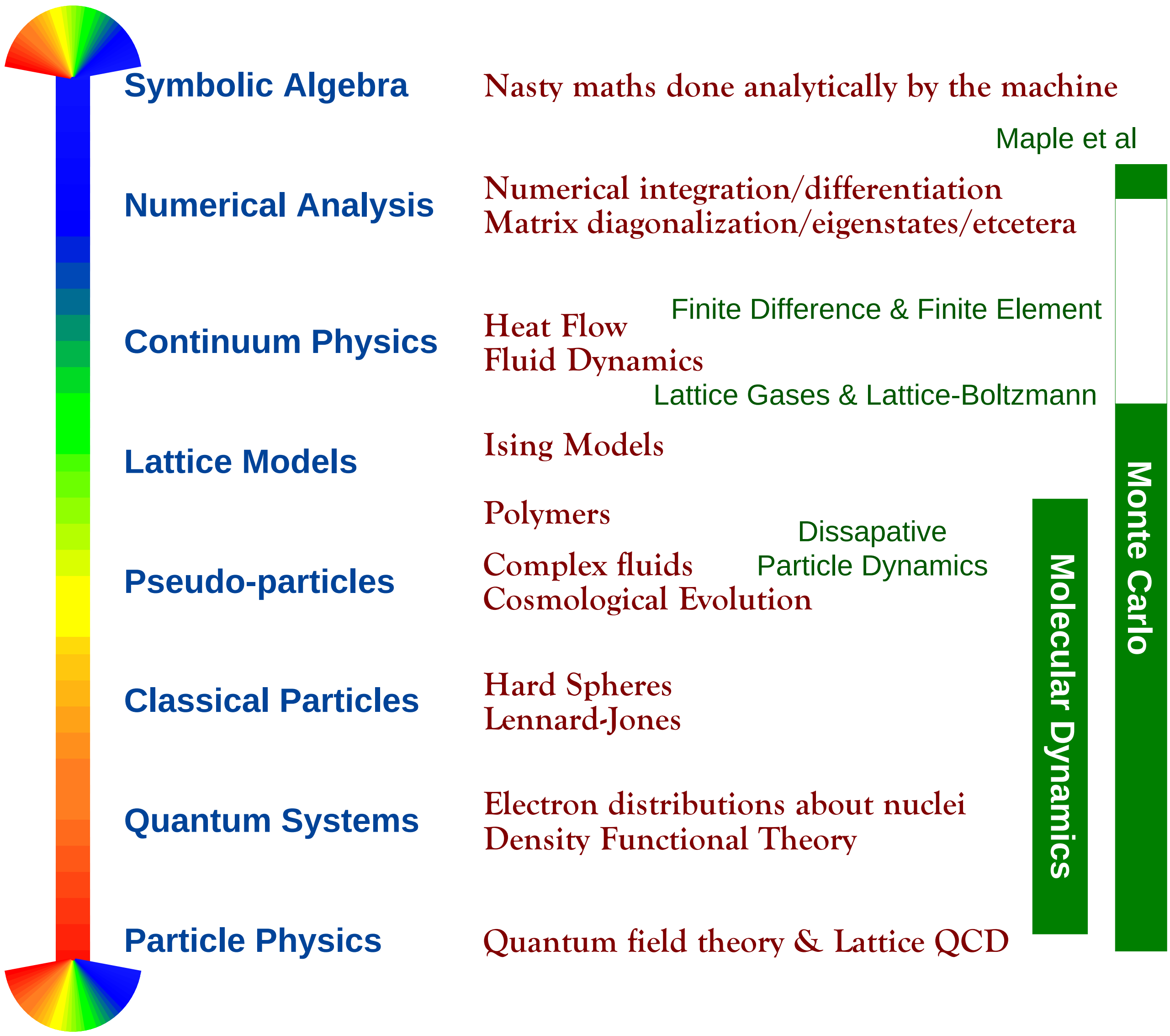
THE THEORY'S JUST TOO HARD!



Or if you want a second opinion, why not ask your friendly neighbourhood coder.

~~The~~ A Spectrum of Computational Physics

Most definitely both arbitrary and non-exhaustive.



Overall Aims

Given some model many-body system.
(A finite set of objects in some space/geometry with some rules for how they interact)

We wish to predict observable properties of a real system.
(Eg bulk properties an effectively infinite number of objects).

Equilibrium Properties
Free-energies, phase behaviour, compressibility, specific heat...
For $T = 0K$, calculating the energy is enough.
For $T > 0K$, we need lots of microstates.

Dynamic Properties
Relaxation times, diffusion times...
Needs lots of microstates.

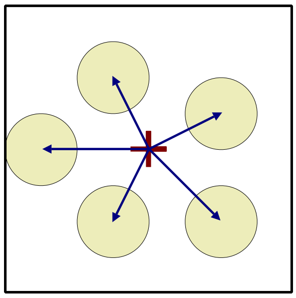
The Simulation Arena

Characterizing the System

+1	+1	+1	+1
-1	-1	+1	+1
-1	-1	+1	-1
-1	-1	-1	+1

Ising Model

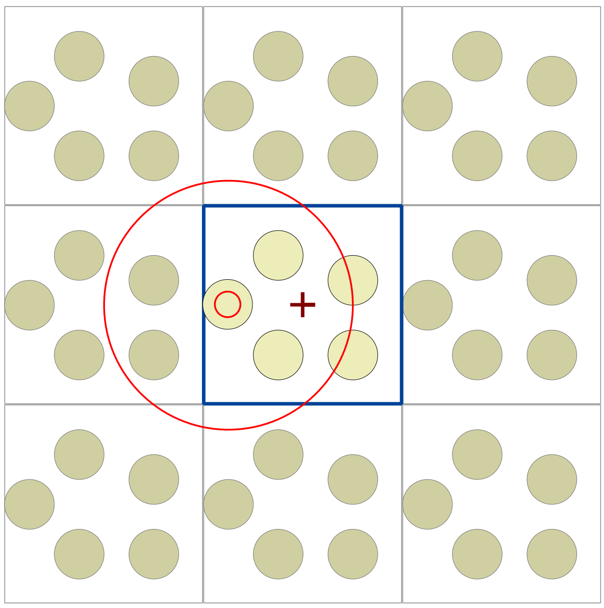
Hard Spheres



+1	+1	+1	+1	-1	-1	+1	+1	-1	-1	+1	-1	-1	-1	-1	+1

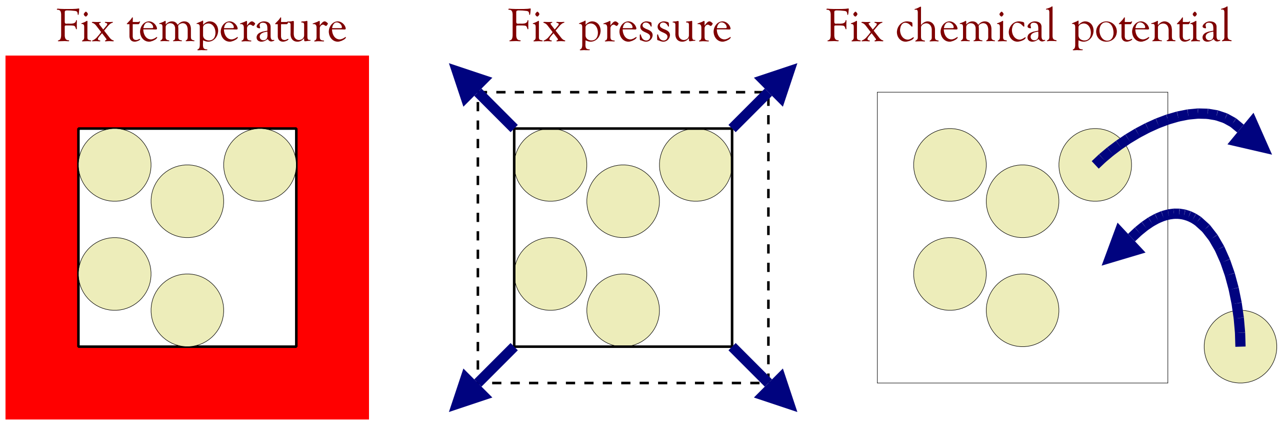
Periodic boundary conditions

+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
-1	-1	+1	+1	-1	-1	+1	+1	-1	-1	+1	+1
-1	-1	+1	-1	-1	-1	+1	-1	-1	-1	+1	-1
-1	-1	-1	+1	-1	-1	-1	+1	-1	-1	-1	+1
+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
-1	-1	+1	+1	-1	-1	+1	+1	-1	-1	+1	+1
-1	-1	+1	-1	-1	-1	+1	-1	-1	-1	+1	-1
-1	-1	-1	+1	-1	-1	-1	+1	-1	-1	-1	+1
+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
-1	-1	+1	+1	-1	-1	+1	+1	-1	-1	+1	+1
-1	-1	+1	-1	-1	-1	+1	-1	-1	-1	+1	-1
-1	-1	-1	+1	-1	-1	-1	+1	-1	-1	-1	+1



External influences

Microcanonical ensemble OR



Using appropriate statistical mechanics rules.

Dimensionless Units

Scale out "real" units, and capture the physics.

Hard Spheres:

Given N, only the ratio between radius and volume matters.

r^3/V : ie the density, ρ^*.

Ising Model:

The ratio of the interaction energy to the thermal energy matters:

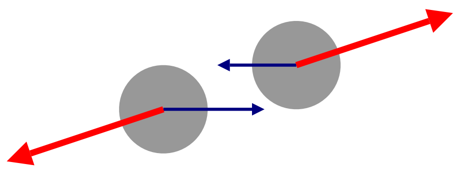
$J/k_BT = 1/T^*$.

Molecular Dynamics

Eg Evolve positions of particles under Newton's Laws.

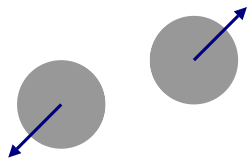
$$a = \frac{\mathrm{d}v}{\mathrm{d}t} \approx \frac{\Delta v}{\Delta t} \qquad v = \frac{\mathrm{d}x}{\mathrm{d}t} \approx \frac{\Delta x}{\Delta t}$$

1) Calculate the forces:



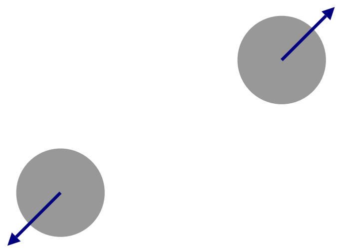
2) Update the velocities:

$$v(t + \Delta t) = v(t) + a\Delta t$$



3) Update the positions:

$$x(t + \Delta t) = x(t) + v\Delta t$$



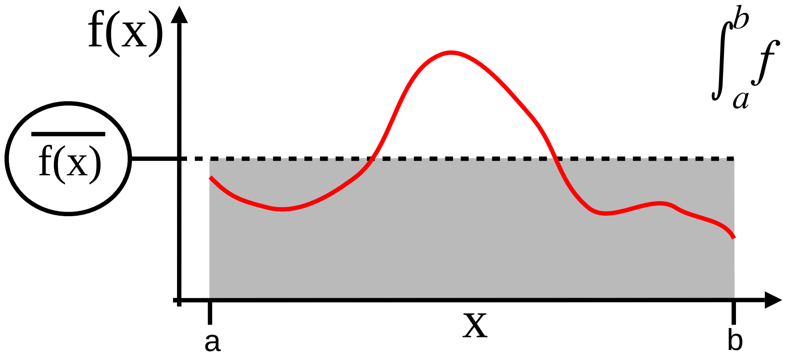
4) Repeat 1, 2 and 3.

There are more accurate ways!

Getting fixed temperature/pressure etc requires mild bludgeoning.

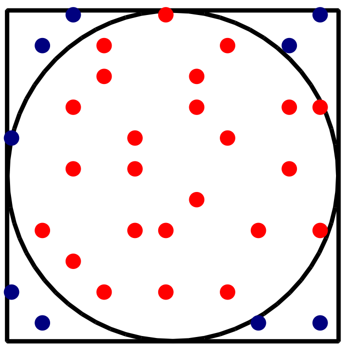
Monte Carlo Method

In 1 Dimension:



$$\int_a^b f(x)\,dx \equiv (b-a)\,\overline{f(x)} \approx \frac{1}{n}\sum_{i=1}^{N} f(\zeta_i)$$

In Two Dimensions:



$$\frac{A(circle)}{A(square)} = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4} \approx \frac{n_{inside}}{n_{total}}$$

$$\left[e.g.\ \frac{24}{33} = 2.9\right]$$

Can we do this for many-dimension integrations over phase space?

$$Z(N, V, T) = \prod_{i=1}^{3N}\left[\int_{-\infty}^{+\infty} \mathrm{d}\dot{x}_i\right] \prod_{i=1}^{3N}\left[\int_{-\infty}^{+\infty} \mathrm{d}x_i\right] \exp\left[-\frac{E(\{x\})}{k_B T}\right]$$

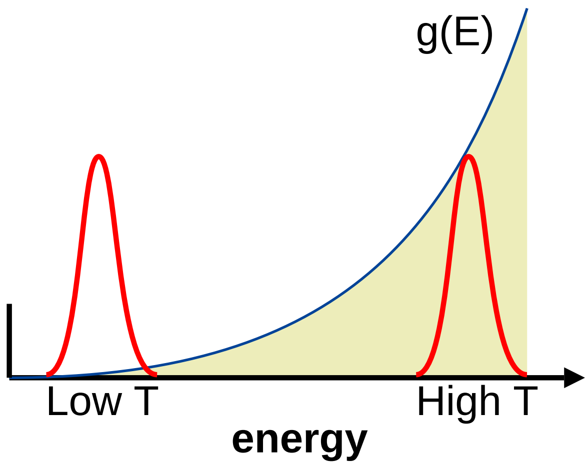
Monte Carlo In Statistical Mechanics:

Eg In the canonical ensemble.

Generate microstates at random.
Estimate energy using...

$$\overline{E} \approx \frac{1}{n} \sum_{i=1}^{n} E_i \exp\left[-\beta E_i\right]$$

This doesn't work!

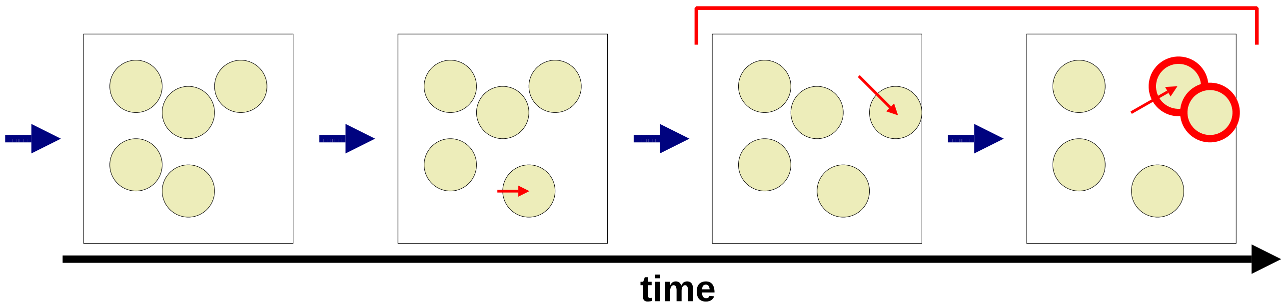


Eg Ising Model:
Tend to generate 'hot' configurations.

Eg Hard-Spheres:
Doesn't work at high densities.

Can we generate the right (high Boltzmann weight) configurations all the time, instead of waiting for them?

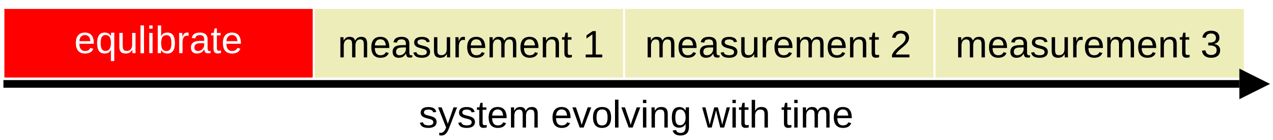
Markov Chains:



OK, but we can't accept all moves.

How do we decide whether to accept a trial move?

Convergence:

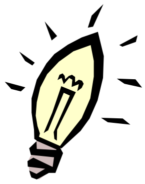


Can show that microscopic reversibility ('detailed balance') is a sufficient condition to ensure convergence.

$$P(i)\,P(i \rightarrow j) = P(j)\,P(j \rightarrow i)$$

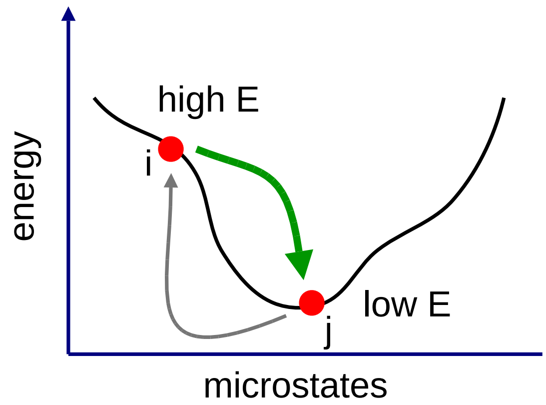
We need an algorithm that satisfies this condition and will sample with the Boltzmann distribution.

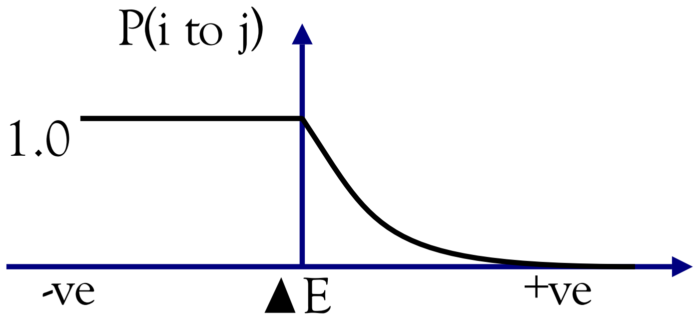
Importance Sampling:



Moves to lower energies are ALWAYS accepted.

Moves up are accepted sometimes...





$$P(i \rightarrow j) = \min[\,1.0,\ \exp(-\beta(E_j - E_i)\,]$$

$$P(i) = \frac{1}{Z}\exp(-\beta E_i)$$

Putting these into the detailed balance condition:

$$\exp(-\beta E_i) \times 1.0 = \exp(-\beta E_j) \times \exp(-\beta(E_i - E_j))$$

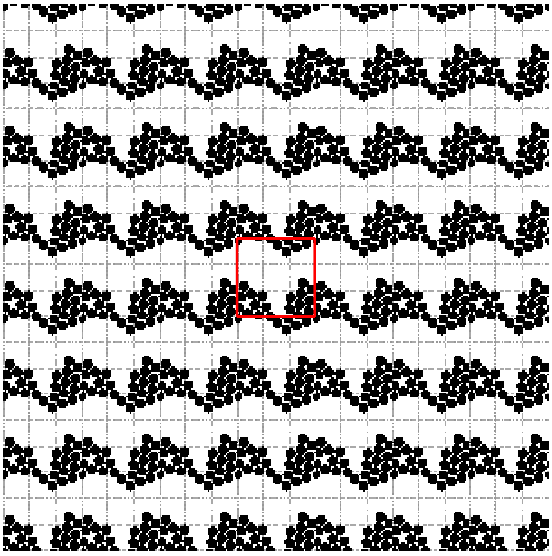
This generates a sequence of Boltzmann distributed microstates.

Estimate observables like this: $$\overline{E} \approx \frac{1}{n}\sum_{i=1}^{n} E_i$$

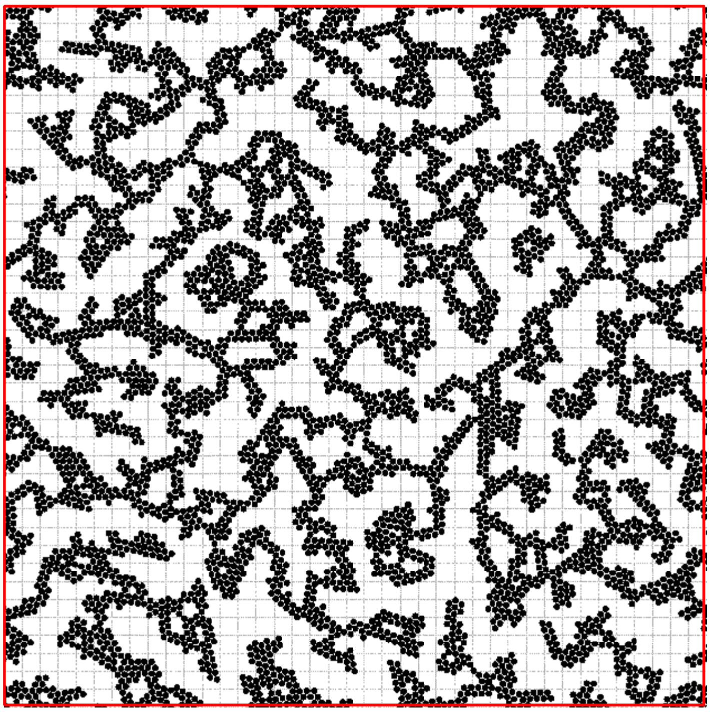
Limitations: ***Finite System Size***

Example: 2D sticky spheres:

N = 25



N = 5000

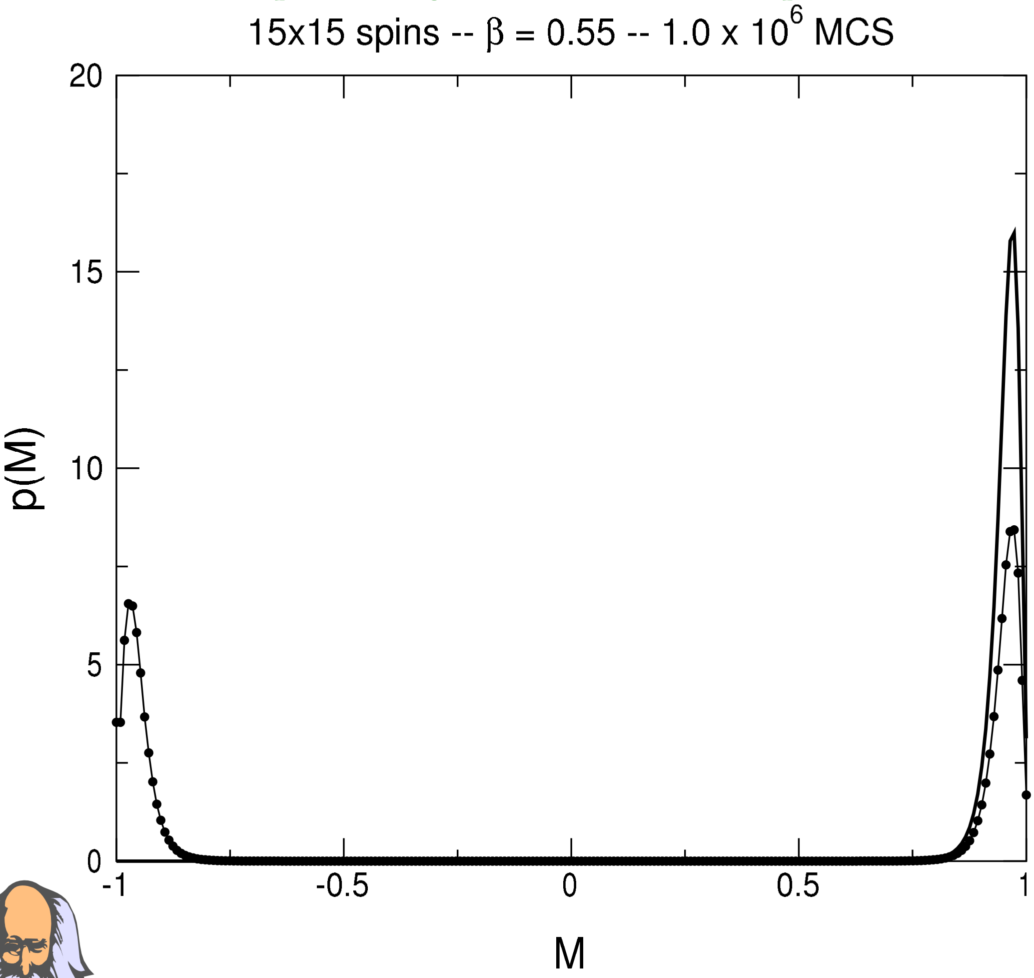


One should always examine a range of system sizes.

A wealth of literature exists on finite size effects
and theory to cope with them ('finite size scaling theory').

Limitations: *Finite Simulation Time*

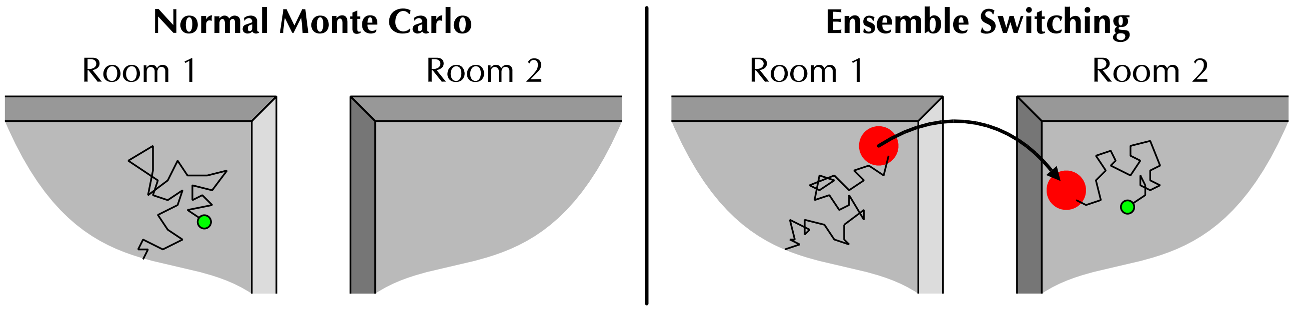
Example: Ising model at low temperature:



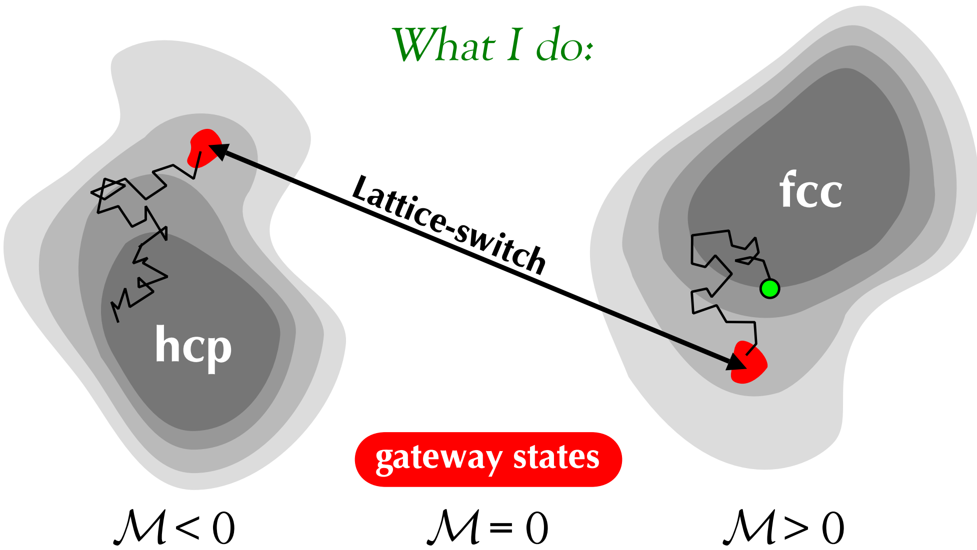
With MC is that you can fudge things legally.

Beating The System

Changing Rooms



What I do:



Final Remarks…

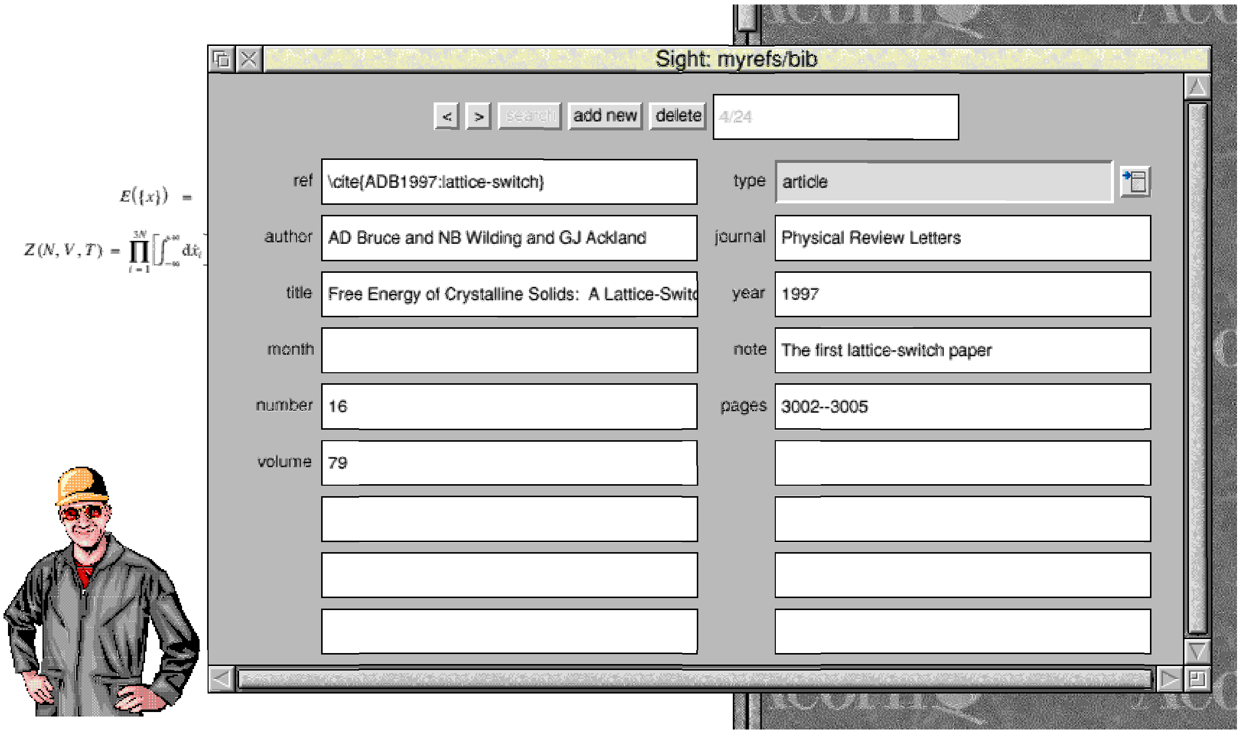
Notable Omissions:
Stability of algorithms.
Finite precision problems.
Estimating accuracy under Monte Carlo,
(and about a million other details).

Quantum Mechanics:
DFT makes calculations 10-100 times harder.

Supercomputing:
Various tricks and cunning techniques.
Makes simulation 10-100 faster.

And hopefully you don't feel like this.

Sight: BibTeX Citation Management Program



Allows you to edit, add and delete entries in
your BibTeX citation file in a "nice" way.

On Unix (at least), it makes it very easy to insert the right \cite{}.

Written in Java - can run on any kind of machine.