PHYS-E0412, Computational Physics, Lecture 4, 29 January 2019

Ilja Makkonen

Aalto University
School of Science

Learning objectives for week 4

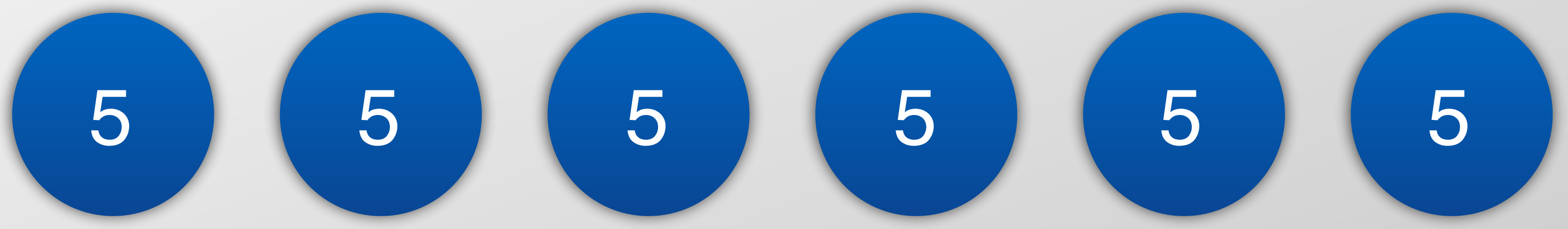
- A case study of Monte Carlo simulation in statistical physics: the Ising model

Homework 4:

Simulated annealing of random-bond Ising model with Metropolis sampling

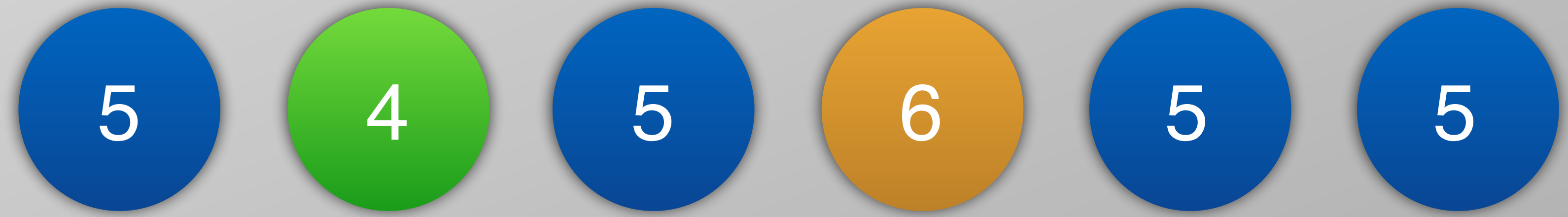
Intro

Take N particles



Each particle has energy that we assume to be an integer E_i (above, all have 5)

Particle pairs can exchange energy $E_i \to E_i + 1 \text{ and } E_j \to E_j - 1$



Total number of particles and total energy is conserved ("conservative energy exchange model")

Any single-particle energy is equally probable for exchange, except zero, assume $E_i \geq 0$

What is the most likely distribution of the single-particle energies?

Simulation

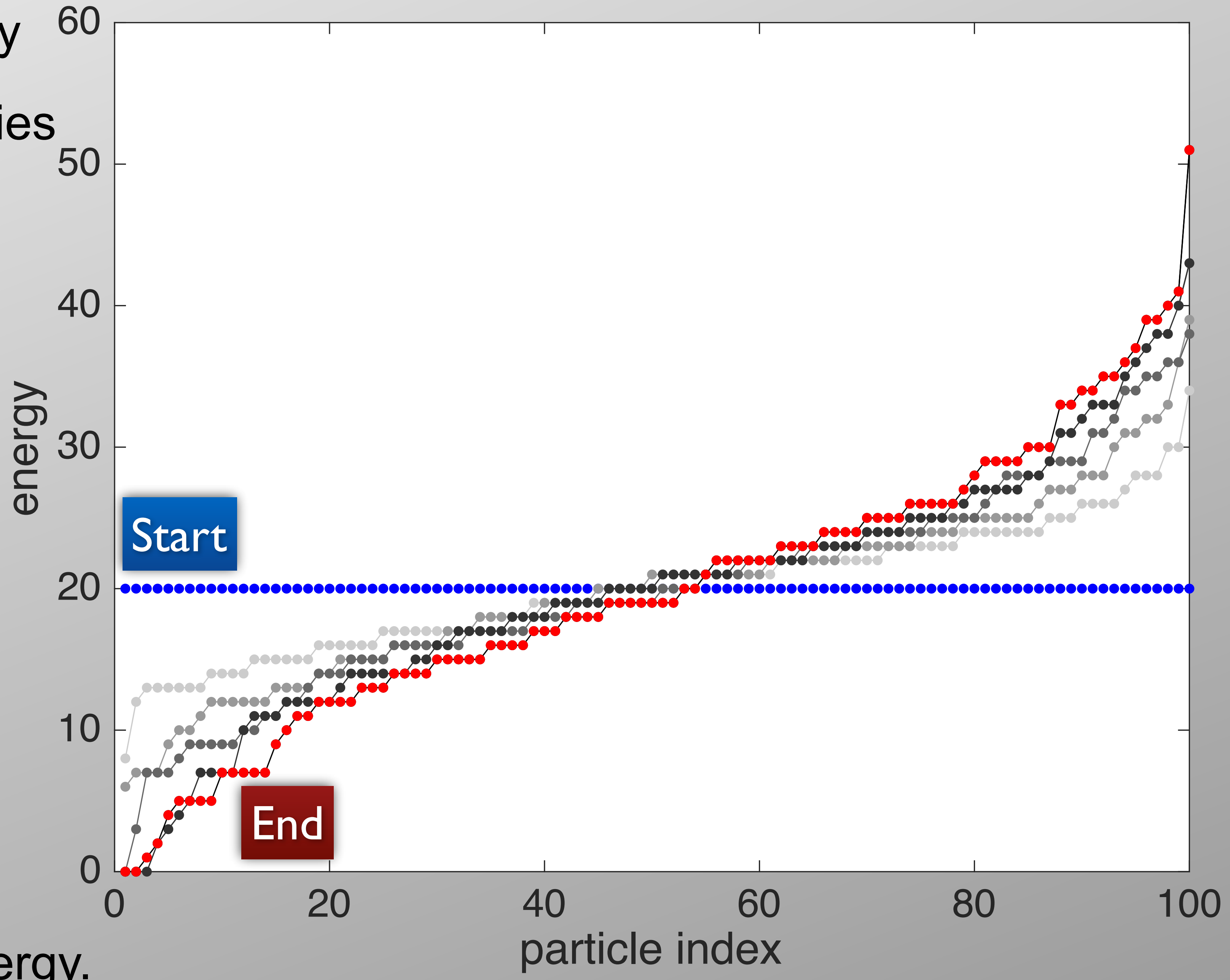
Simple process:

Each particle starts with same energy

Take random pair and change energies

$$E_i \rightarrow E_i + 1 \text{ and } E_j \rightarrow E_j - 1$$

```
for j=1:N, % loop over particles
   particle1=randi(N);
   if E(particle1)==0, continue; end
   particle2=randi(N);
   E(particle1)=E(particle1)-1;
   E(particle2)=E(particle2)+1;
end
```

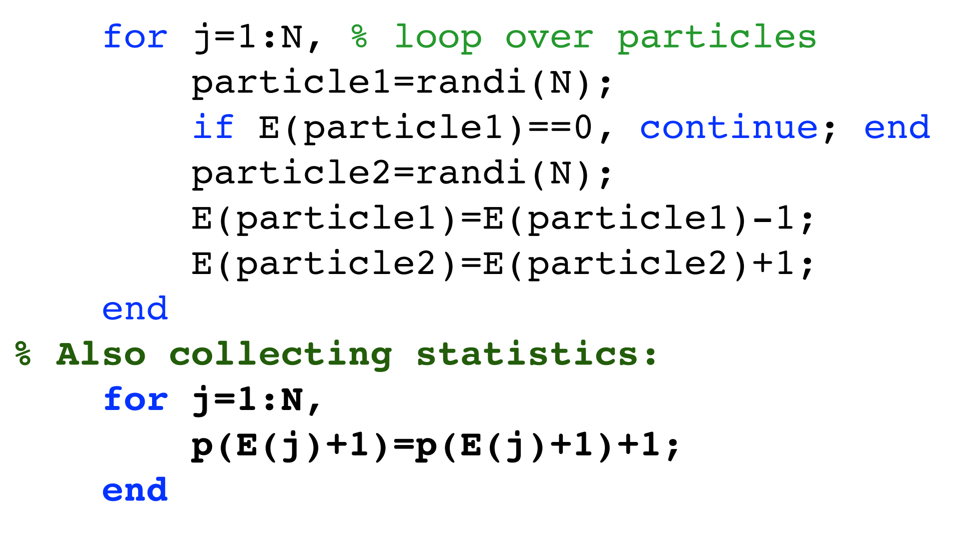


Continue avoids negative energies

In the plot, particles are sorted by energy.

This is only a short simulation. What is the probability distribution if we simulate longer?

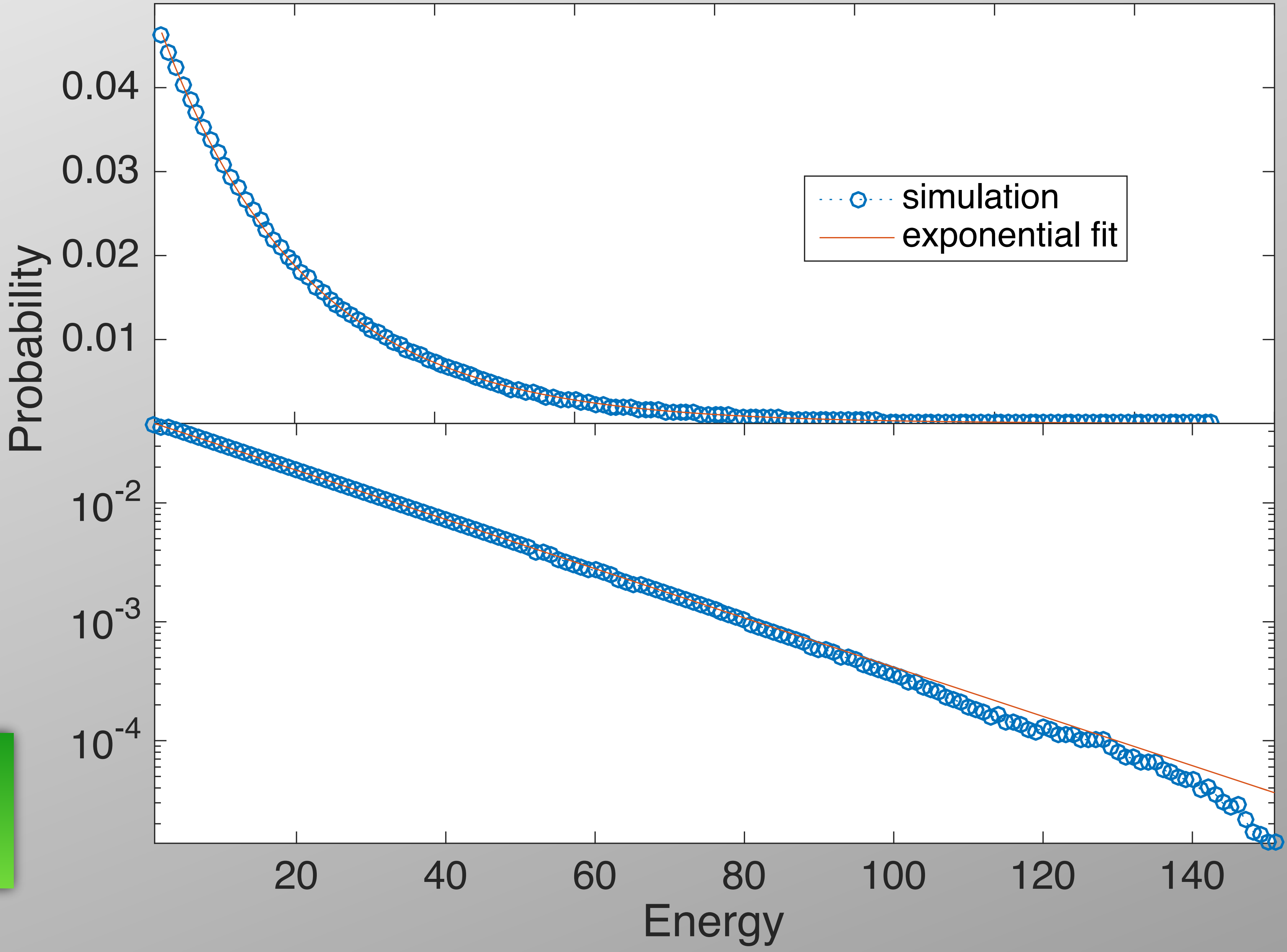
Results



```
    for j=1:N, % loop over particles
        particle1=randi(N);
        if E(particle1)==0, continue; end
        particle2=randi(N);
        E(particle1)=E(particle1)-1;
        E(particle2)=E(particle2)+1;
    end
% Also collecting statistics:
    for j=1:N,
        p(E(j)+1)=p(E(j)+1)+1;
    end
```

Now also collecting statistics

Results on the plot right:



Probability distribution seems to follow an exponential law

$$p_i \propto \exp(-\beta E_i)$$

This is the Boltzmann factor you should have seen in statistical physics!

We had a special case, but similar things can be found in a more general case also.

Statistical physics

Consider N particles.

Possible energy states $E_1, E_2, E_3, \ldots$.

Number of particles at state i is n_i.

Total number of particles is found by summing n_i's:

$$\sum_i n_i = N \ ,$$

and average total energy is:

$$\sum_i n_i E_i = UN$$

What is the most probable distribution of n_i's for a given N and U?

(Assuming every state is equally probable.)

Most probable distribution

Number of ways Ω a certain set $\{n_i\}$ can be found is

$$\Omega[\{n_i\}] = \frac{N!}{n_1!n_2!n_3!\ldots}.$$

In equilibrium Ω (or the entropy $S \propto \ln\Omega$) is maximized with the constraints of a constant N and U (using Lagrange multipliers), resulting in

$$P(E_i) = \frac{n_i^*}{N} = \frac{\exp(-\beta E_i)}{\sum_i \exp(-\beta E_i)}.$$

Thermodynamics tells us that $\beta = 1/(kT)$ (k is the Boltzmann constant).

$\exp(-\beta E_i)$ is called the Bolzmann factor.

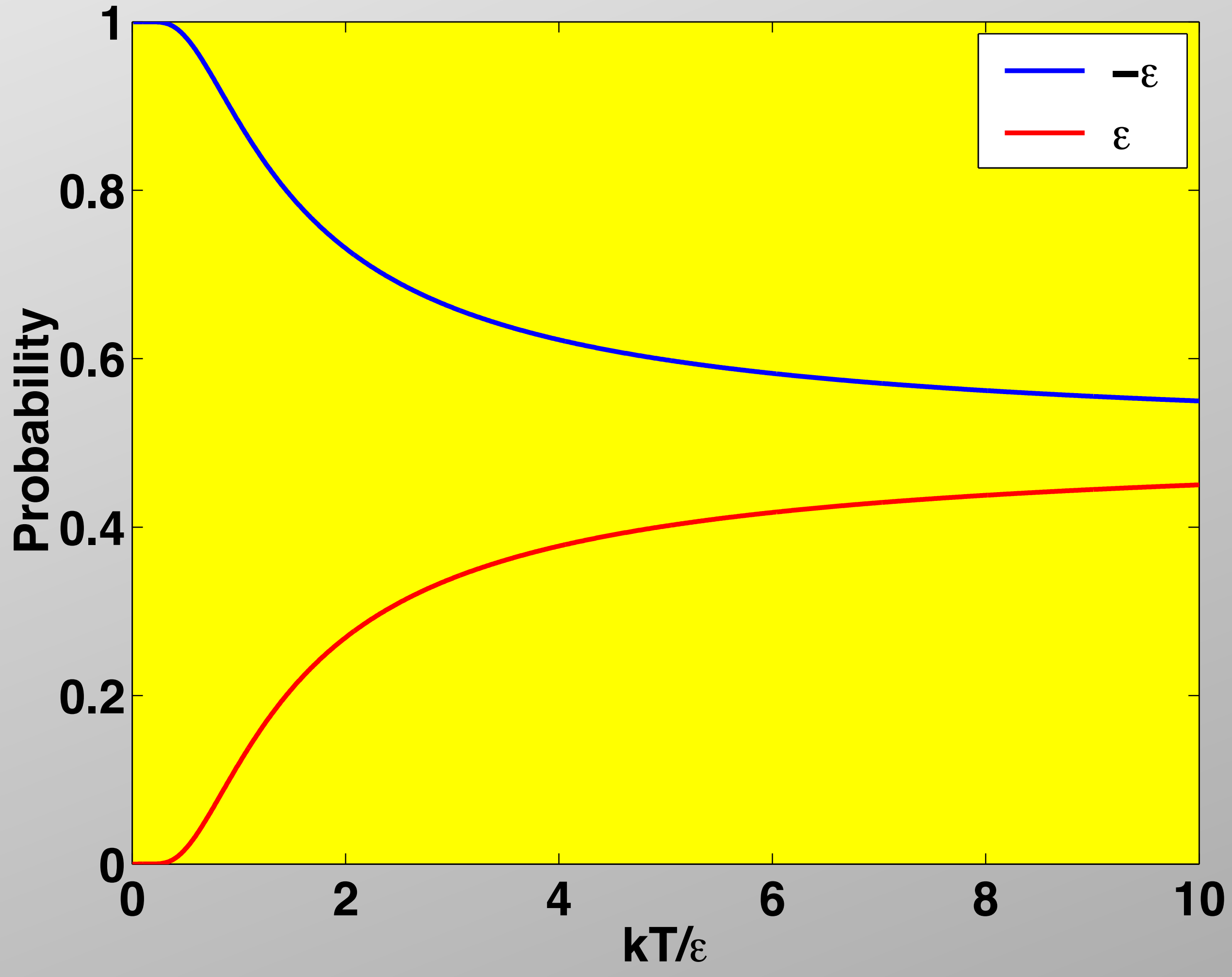
For details on how this works for the type of energy exchange models discussed above, see e.g. Scafetta and West, J. Phys.: Condens. Matter 19, 065138 (2007).

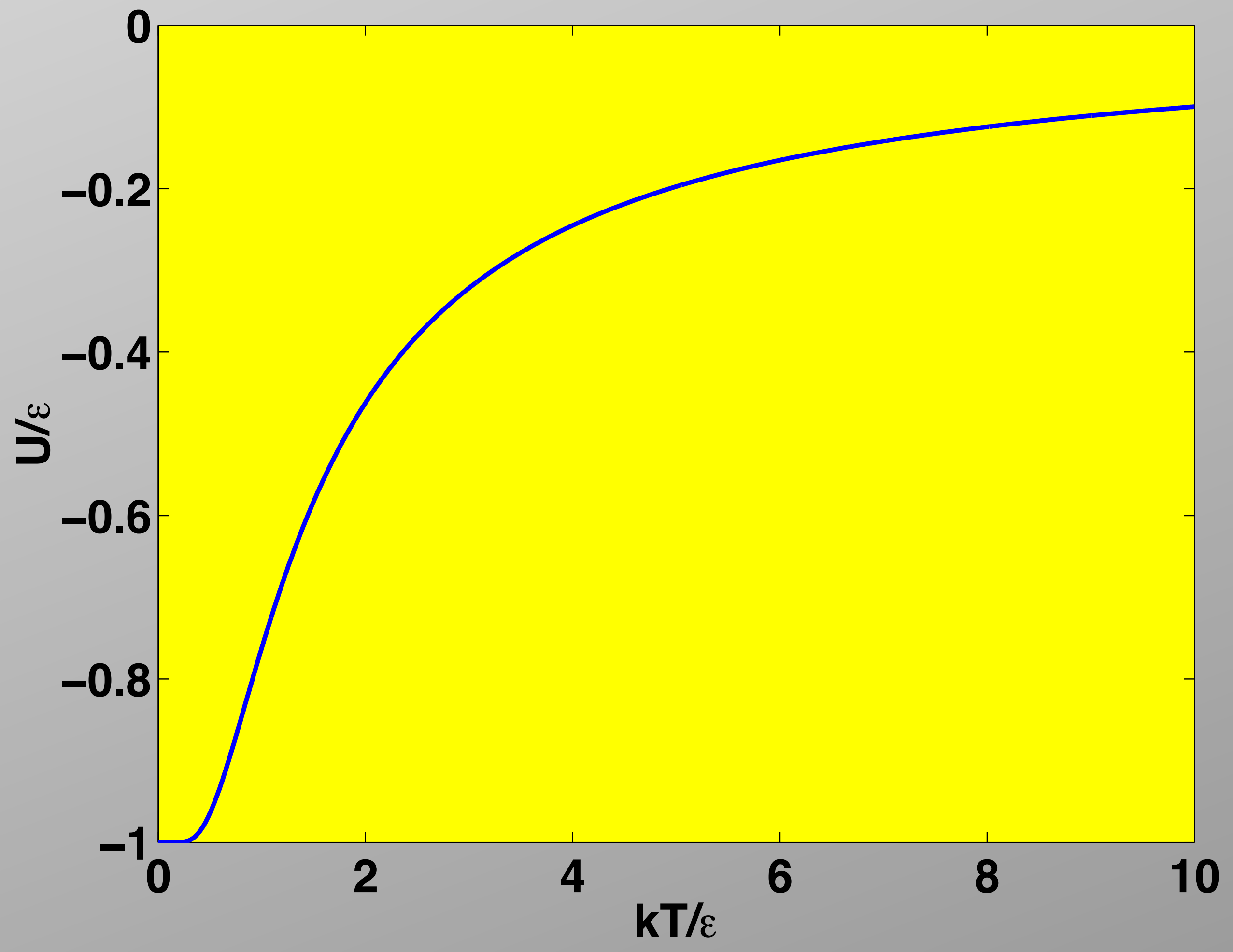
Plugging in numbers (needed later)

Example: One particle, two states, energies ϵ and $-\epsilon$.

Probabilities: $\frac{\exp(-\beta\epsilon)}{2\cosh(\beta\epsilon)}$ and $\frac{\exp(\beta\epsilon)}{2\cosh(\beta\epsilon)}$.

Energy: $U = -\epsilon \tanh(\beta\epsilon)$.





Definitions

Partition function is defined as: $Z = \sum_i \exp(-\beta E_i)$.

So, e.g., probability for a state i is $\exp(-\beta E_i)/Z$.

Partition function of previous example:

$$Z = \exp(-\beta\epsilon) + \exp(\beta\epsilon) = 2\cosh\left(\beta\epsilon\right) .$$

If number of states with energy U is $\Omega(N, V, U)$, we can define

Entropy: $S(N, V, U) = k \ln \Omega(N, V, U)$.

Note also that

$$Z = \sum_U e^{-\beta U} \Omega = \sum_U e^{-\beta(U - TS)}$$

so states with minimum $F = U - TS$ dominate.

F is called free energy.

Linking statistical physics to thermodynamics.

Observables

Expectation value of observable A is

$$\langle A\rangle = \sum_i A_i \exp(-\beta E_i)/Z \ .$$

The average energy as derivative $U = -\frac{\partial \ln Z}{\partial \beta}$.

Response functions: Specific heat is

$$C_V = \left(\frac{\partial U}{\partial T}\right)_V$$

and thus it is also

$$C_V = \frac{1}{kT^2}\frac{\partial^2 \ln Z}{\partial \beta^2}$$

or **fluctuation** of energy

$$C_V = \frac{1}{kT^2}\left(\langle E^2\rangle - \langle E\rangle^2\right)$$

We have had N, V, and T fixed, this is called canonical ensemble.

Microstates and all that

Everything looks roughly the same if we replace single-particle states with configurations of all the (non-interacting) particles. These configurations are commonly called “microstates”.

Example: Take the one particle and two states model discussed above, and generalise this to N non-interacting particles with two possible energies.

N=3, 1-body states “u” and “d”, now microstates are: (uuu), (uud), (udu), ..., (ddd)

Energy of an microstate is (Nu-Nd) ε, where Nx is number of particles at state x (u or d).

Nu+Nd=N, and the total number of microstates is 2^N.

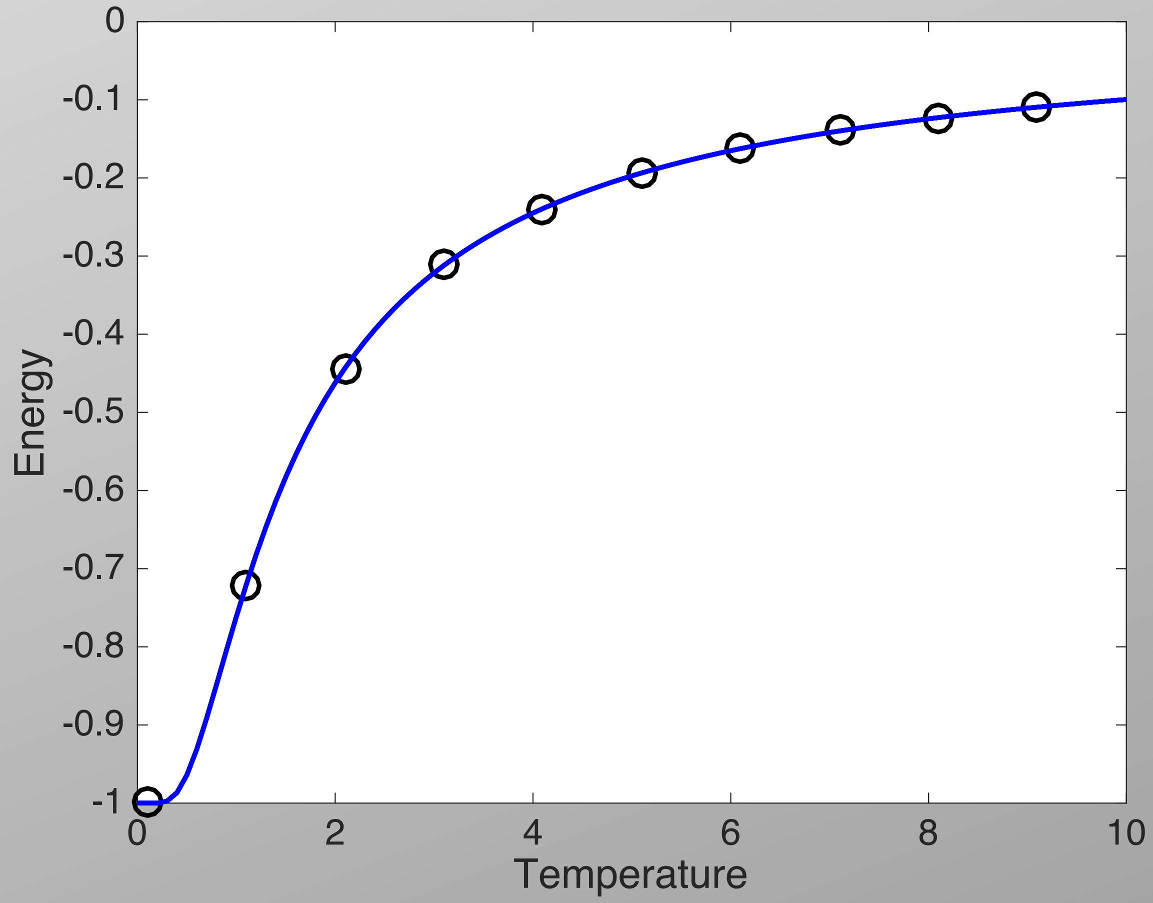
Expectation values as sums over microstates: $U = \frac{\sum_i E_i \exp(-\beta E_i)}{\sum_i \exp(-\beta E_i)}$

Loop over microstates

We can use bits of an integer to code if the particle is “u” or “d”

Take integers from zero to 2^N -1 and bit “1” is “u” and bit “0” is “d”:

```
N=15; % Number of particles
T=1; % Temperature
sum_e=0; % <energy>
partition_f=0; % partition function
for i=1:2^N,
    Nu=sum(bitget(uint32(i-1),1:N));
    Nd=N-Nu;
    E=Nu-Nd;
    boltzmann=exp(-E/T);
    sum_e=sum_e+E*boltzmann;
    partition_f=partition_f+boltzmann;
end
```



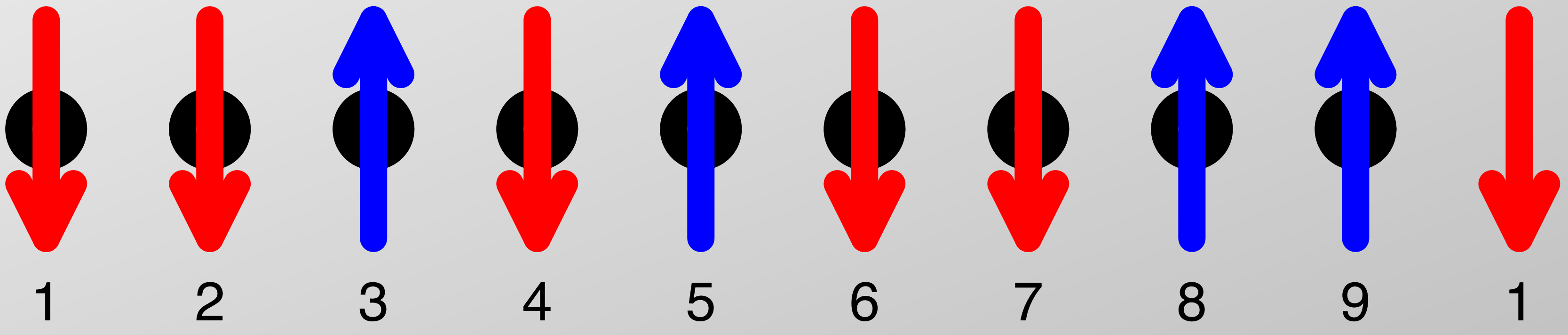
Here energy is scaled so that ε=1. Energy per spin follows the -tanh(1/T) formula!

Matlab has different types of variables...hidden inside it, try class(uint32(1)) and class(1).

But this was stupid, as we could separate the sum over microstates to N times sum over two states. Yes, but this can NOT be done if the particles start to interact (**NEXT**)!

Ising model, 1D

N spins on a 1D lattice:



interaction between neighbors:

$$\mathcal{H} = -J \sum_{i=1}^{N} \sigma_i \sigma_{i+1} \ ,$$

where $\sigma_i = \pm 1$ is spin at site i.

$J > 0$, and PBC have been used.

$T = 0$, ground state with all spins up or down.

Like before: Expectation values as sums over microstates:

$$\langle A \rangle = \frac{\sum_{j=1}^{2^N} A(X_j) \exp(-\beta \mathcal{H}(X_j))}{\sum_j \exp(-\beta \mathcal{H}(X_j))}$$

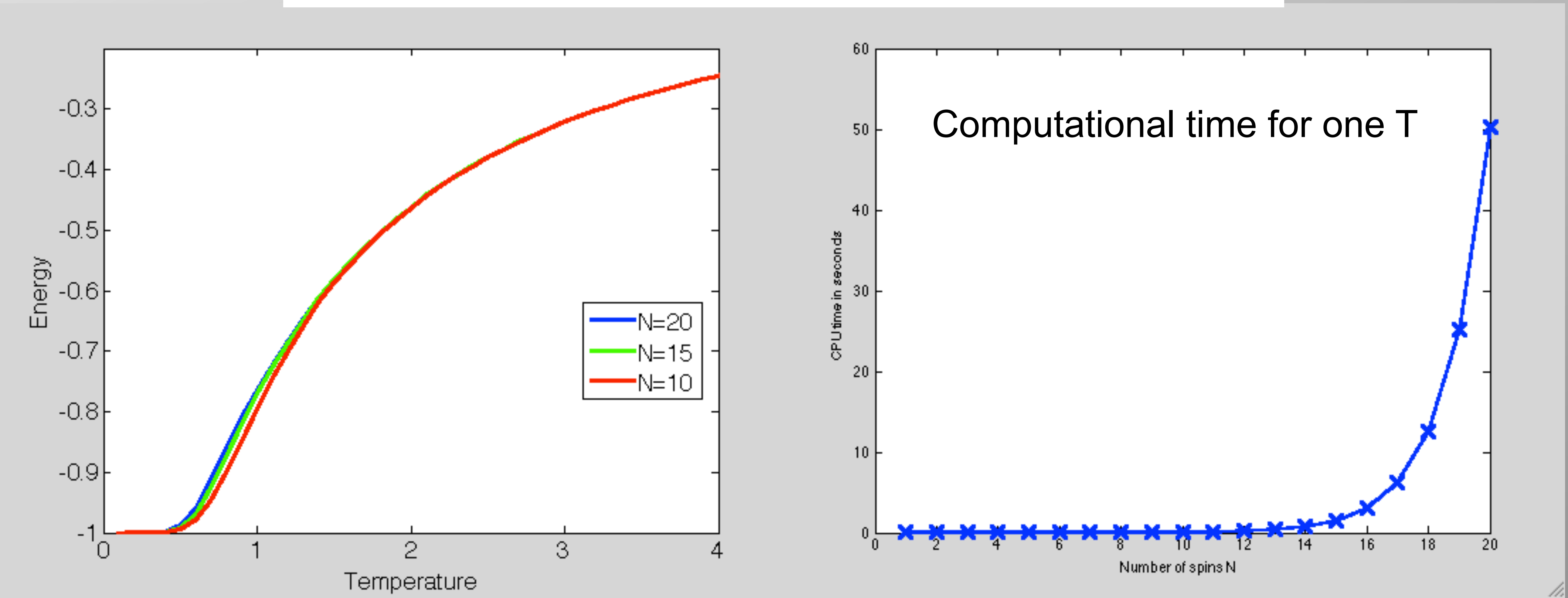
X contains all spins $\{\sigma_i\}_{i=1}^{N}$

Ising model, 1D

Matlab summing over microstates:

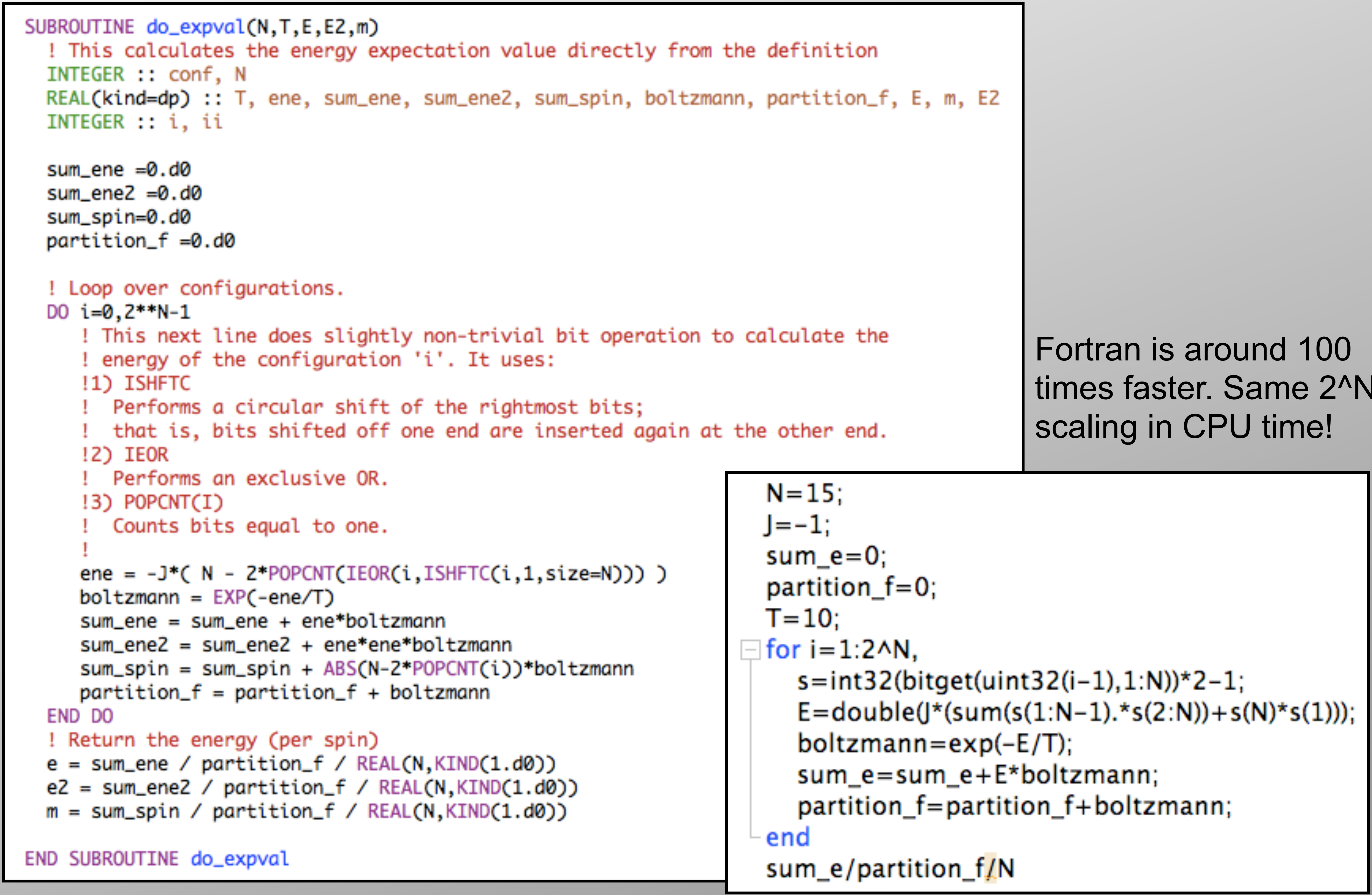
```
sum_e=0;
partition_f=0;
for i=1:2^N,
    s=int32(bitget(uint32(i-1),1:N))*2-1;
    E=double(J*(sum(s(1:N-1).*s(2:N))+s(N)*s(1)));
    boltzmann=exp(-E/T);
    sum_e=sum_e+E*boltzmann;
    partition_f=partition_f+boltzmann;
end
```

Interaction



Computational time for one T

Fortran and Matlab working on bits

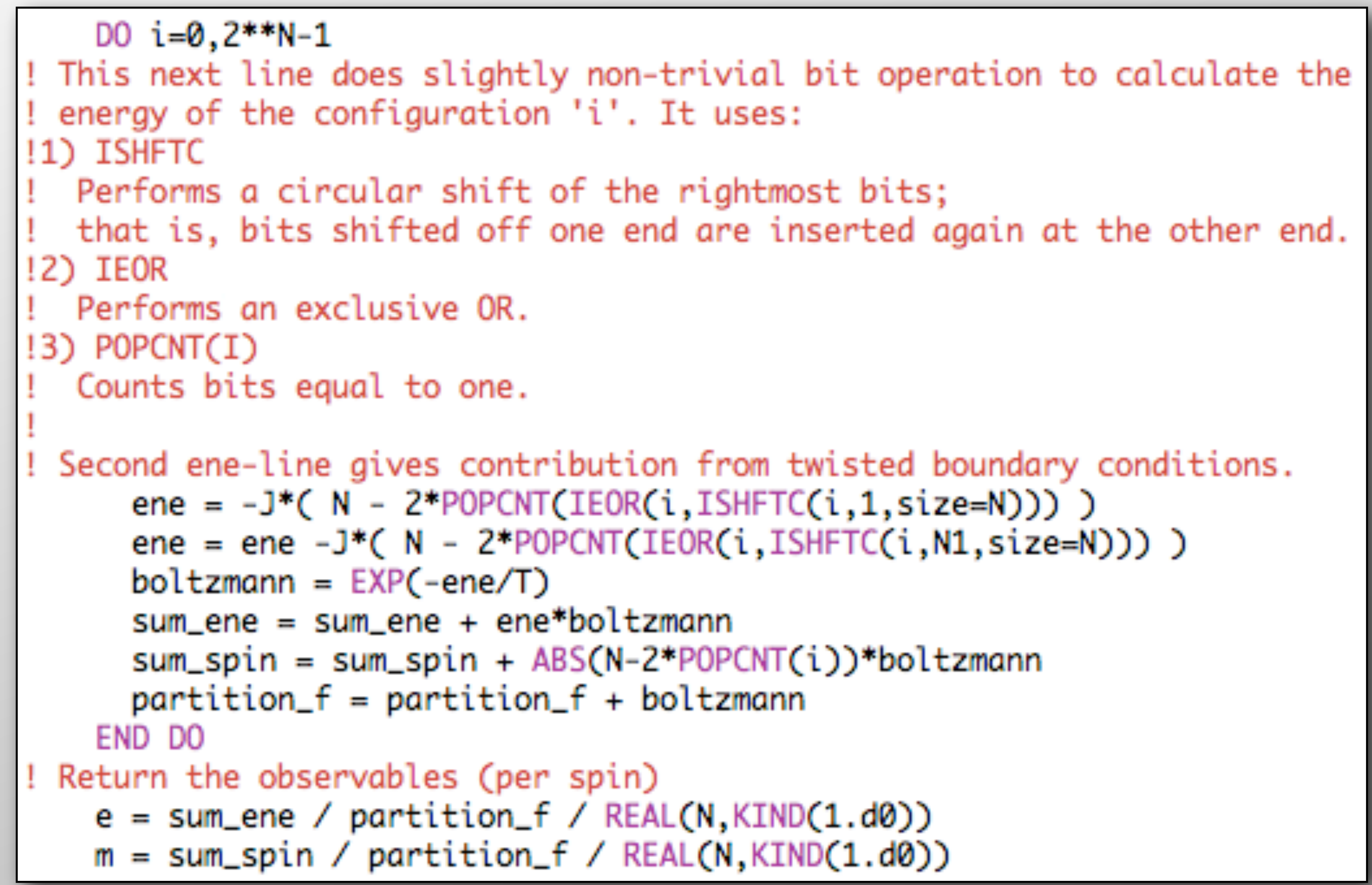


```fortran
SUBROUTINE do_expval(N,T,E,E2,m)
  ! This calculates the energy expectation value directly from the definition
  INTEGER :: conf, N
  REAL(kind=dp) :: T, ene, sum_ene, sum_ene2, sum_spin, boltzmann, partition_f, E, m, E2
  INTEGER :: i, ii

  sum_ene =0.d0
  sum_ene2 =0.d0
  sum_spin=0.d0
  partition_f =0.d0

  ! Loop over configurations.
  DO i=0,2**N-1
     ! This next line does slightly non-trivial bit operation to calculate the
     ! energy of the configuration 'i'. It uses:
     !1) ISHFTC
     !  Performs a circular shift of the rightmost bits;
     !  that is, bits shifted off one end are inserted again at the other end.
     !2) IEOR
     !  Performs an exclusive OR.
     !3) POPCNT(I)
     !  Counts bits equal to one.
     !
     ene = -J*( N - 2*POPCNT(IEOR(i,ISHFTC(i,1,size=N))) )
     boltzmann = EXP(-ene/T)
     sum_ene = sum_ene + ene*boltzmann
     sum_ene2 = sum_ene2 + ene*ene*boltzmann
     sum_spin = sum_spin + ABS(N-2*POPCNT(i))*boltzmann
     partition_f = partition_f + boltzmann
  END DO
  ! Return the energy (per spin)
  e = sum_ene / partition_f / REAL(N,KIND(1.d0))
  e2 = sum_ene2 / partition_f / REAL(N,KIND(1.d0))
  m = sum_spin / partition_f / REAL(N,KIND(1.d0))

END SUBROUTINE do_expval
```

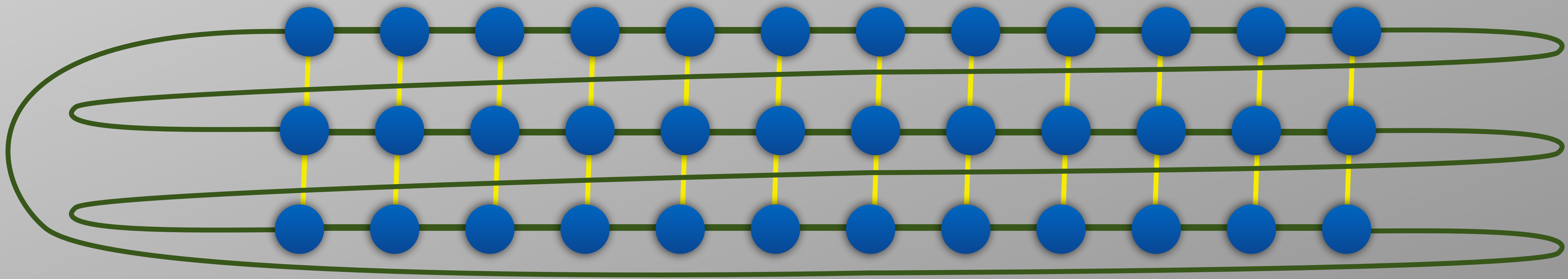
```matlab
N=15;
J=-1;
sum_e=0;
partition_f=0;
T=10;
for i=1:2^N,
    s=int32(bitget(uint32(i-1),1:N))*2-1;
    E=double(J*(sum(s(1:N-1).*s(2:N))+s(N)*s(1)));
    boltzmann=exp(-E/T);
    sum_e=sum_e+E*boltzmann;
    partition_f=partition_f+boltzmann;
end
sum_e/partition_f/N
```

Fortran is around 100 times faster. Same 2^N scaling in CPU time!

Brute force summation, 2D case



```
    DO i=0,2**N-1
! This next line does slightly non-trivial bit operation to calculate the
! energy of the configuration 'i'. It uses:
!1) ISHFTC
!  Performs a circular shift of the rightmost bits;
!  that is, bits shifted off one end are inserted again at the other end.
!2) IEOR
!  Performs an exclusive OR.
!3) POPCNT(I)
!  Counts bits equal to one.
!
! Second ene-line gives contribution from twisted boundary conditions.
      ene = -J*( N - 2*POPCNT(IEOR(i,ISHFTC(i,1,size=N))) )
      ene = ene -J*( N - 2*POPCNT(IEOR(i,ISHFTC(i,N1,size=N))) )
      boltzmann = EXP(-ene/T)
      sum_ene = sum_ene + ene*boltzmann
      sum_spin = sum_spin + ABS(N-2*POPCNT(i))*boltzmann
      partition_f = partition_f + boltzmann
    END DO
! Return the observables (per spin)
    e = sum_ene / partition_f / REAL(N,KIND(1.d0))
    m = sum_spin / partition_f / REAL(N,KIND(1.d0))
```



Horizontal couplings along the green line, vertically yellow (over boundary not shown)

Dead end?

The 1D Ising model seems to be too hard to calculate by direct summation.

Any method **doubles** computer time as we add one more spin to the system.

In addition, analytic results for 1D Ising tells that it is not very interesting.

However, 2D version turns out to have a phase transition!

J couples nearest-neighbor spins.

Notation $\langle i, j \rangle$ counts all n-n pairs once.

Again $\sigma = \pm 1$

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - H \sum_i \sigma_i$$

Sites i on lattice, various boundary conditions possible.

H is external magnetic field.

But we need a more clever algorithm for the simulations.

Monte Carlo in statistical physics

Our example: N spins pointing up or down. Expectation value of observable A:

$$\langle A \rangle = \frac{\sum_{j=1}^{2^N} A(X_j) \exp(-\beta \mathcal{H}(X_j))}{\sum_j \exp(-\beta \mathcal{H}(X_j))}$$

X contains all spins $\{\sigma_i\}_{i=1}^{N}$

We can <u>not</u> sum all terms for large N, bad scaling.

Find the important terms by Metropolis, sampling X from correct distribution:

$$\langle A \rangle = \frac{1}{M} \sum_{i}^{M} A(X_i), \;\; X_i \text{ from } \; \exp(-\beta \mathcal{H}(X_i))/Z$$

Here beta is the inverse temperature and H is the Hamiltonian (energy) of the system

Generalisation to continuum variables, e.g.

$$Z = \frac{1}{N! h^{3N}} \int \exp[-\beta H(p_1 \cdots p_N, x_1 \cdots x_N)] \, d^3p_1 \cdots d^3p_N \, d^3x_1 \cdots d^3x_N$$

Metropolis

0. Choose an initial state X.

1. Pick a random site i.

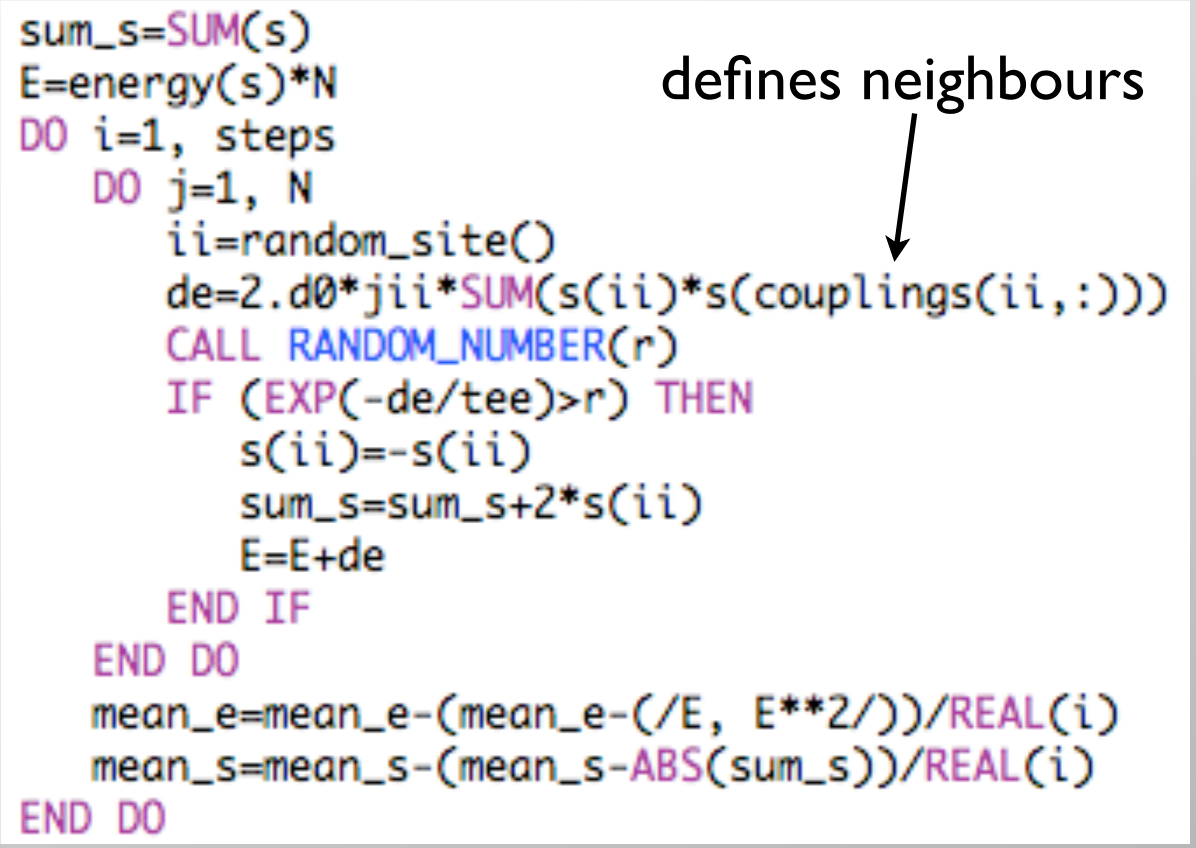
2. Calculate the energy change ΔE if the spin σ_i is flipped.

3. Generate a random number r (between zero and one).

4. If $r < \exp\left(-\Delta E/kT\right)$, flip the spin.

5. Measure observables.

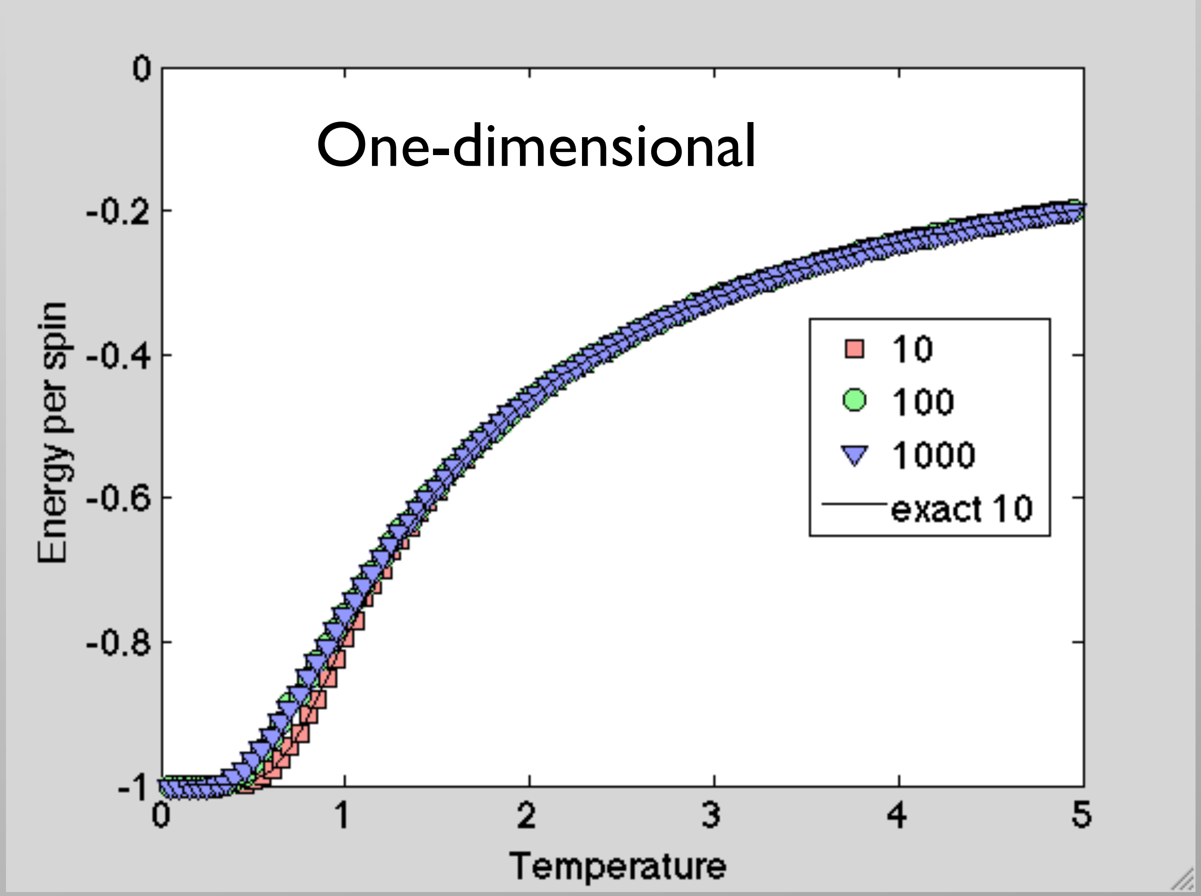
6. Go to 1.

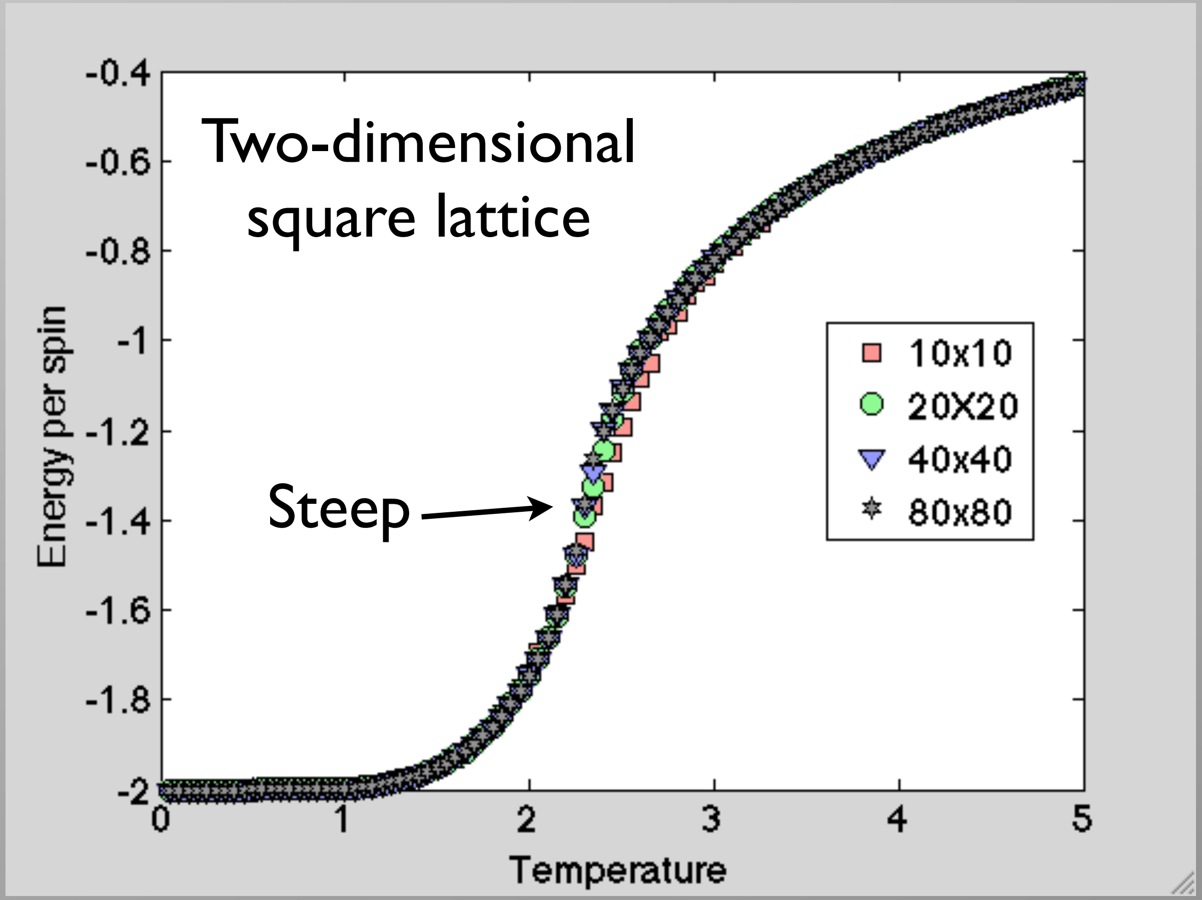


```
sum_s=SUM(s)
E=energy(s)*N
DO i=1, steps
   DO j=1, N
      ii=random_site()
      de=2.d0*jii*SUM(s(ii)*s(couplings(ii,:)))
      CALL RANDOM_NUMBER(r)
      IF (EXP(-de/tee)>r) THEN
         s(ii)=-s(ii)
         sum_s=sum_s+2*s(ii)
         E=E+de
      END IF
   END DO
   mean_e=mean_e-(mean_e-(/E, E**2/))/REAL(i)
   mean_s=mean_s-(mean_s-ABS(sum_s))/REAL(i)
END DO
```

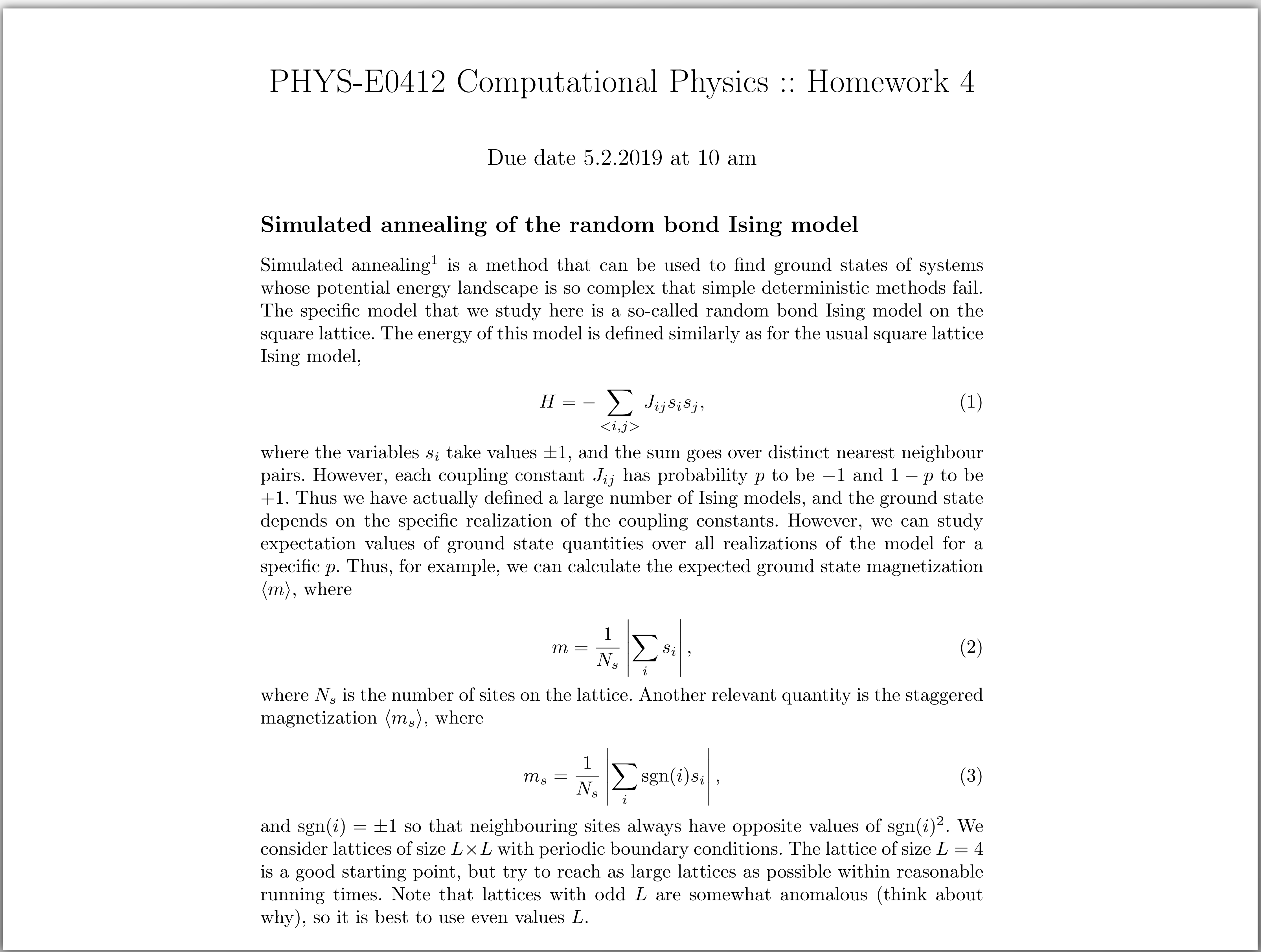
defines neighbours

Any lattice with the same code, just modify couplings





Homework 4



PHYS-E0412 Computational Physics :: Homework 4

Due date 5.2.2019 at 10 am

Simulated annealing of the random bond Ising model

Simulated annealing[1] is a method that can be used to find ground states of systems whose potential energy landscape is so complex that simple deterministic methods fail. The specific model that we study here is a so-called random bond Ising model on the square lattice. The energy of this model is defined similarly as for the usual square lattice Ising model,

$$H = -\sum_{<i,j>} J_{ij} s_i s_j, \tag{1}$$

where the variables s_i take values ± 1, and the sum goes over distinct nearest neighbour pairs. However, each coupling constant J_{ij} has probability p to be -1 and $1-p$ to be $+1$. Thus we have actually defined a large number of Ising models, and the ground state depends on the specific realization of the coupling constants. However, we can study expectation values of ground state quantities over all realizations of the model for a specific p. Thus, for example, we can calculate the expected ground state magnetization $\langle m \rangle$, where

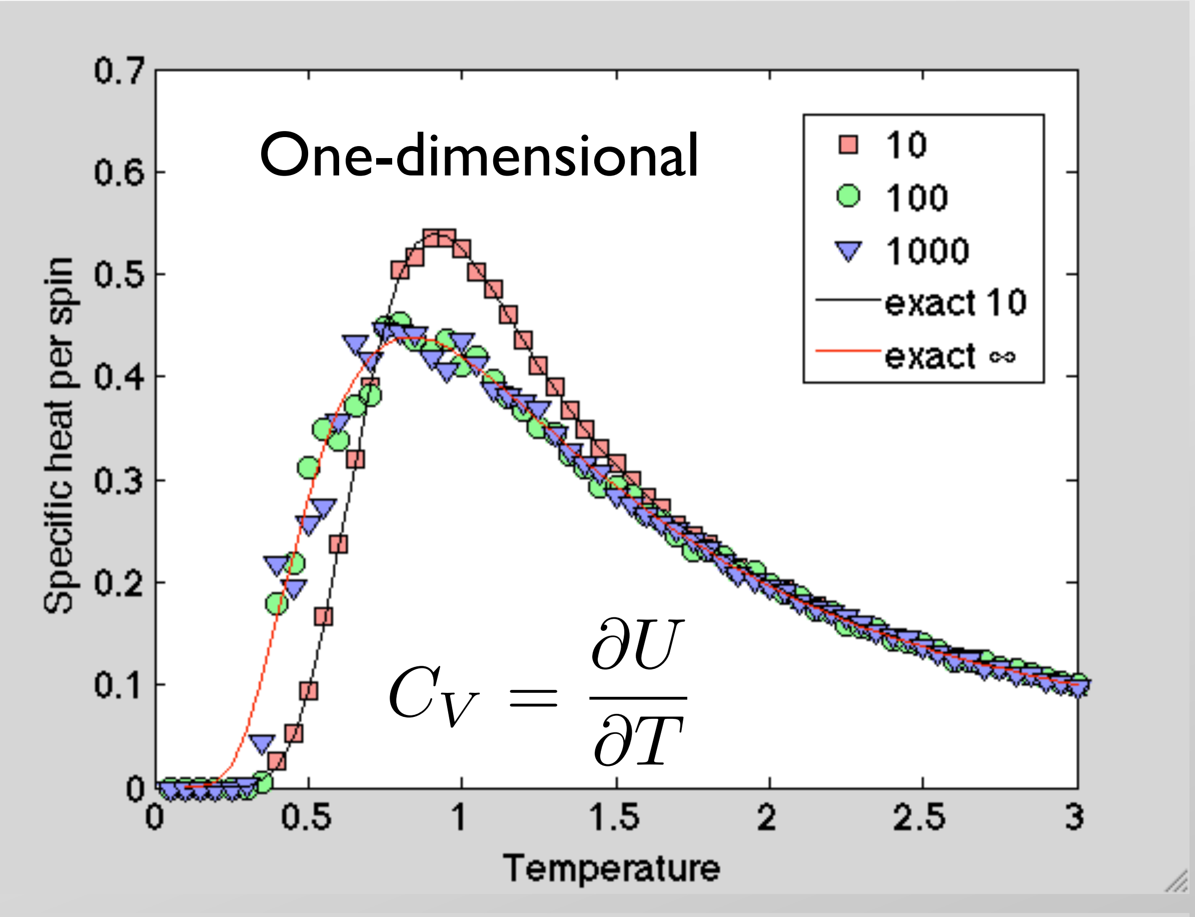
$$m = \frac{1}{N_s} \left| \sum_i s_i \right|, \tag{2}$$

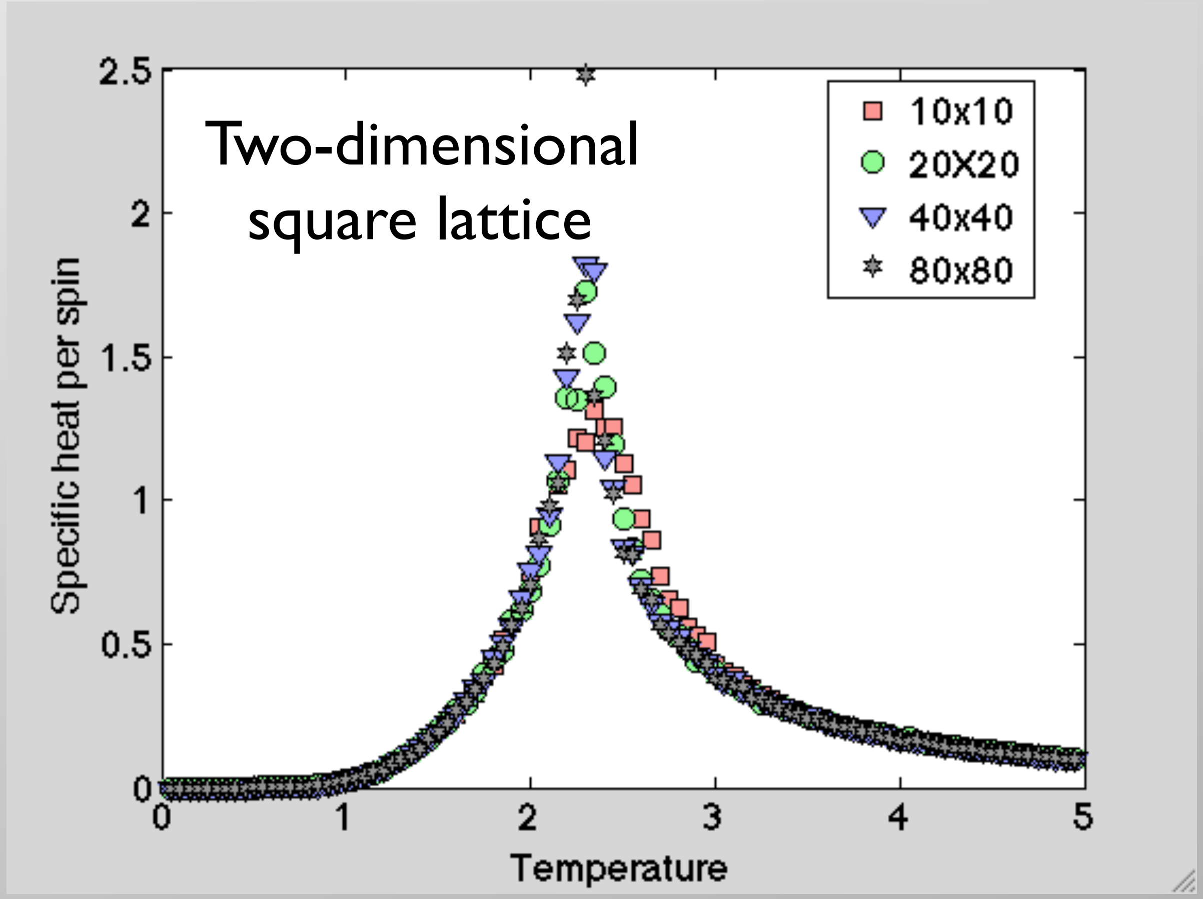
where N_s is the number of sites on the lattice. Another relevant quantity is the staggered magnetization $\langle m_s \rangle$, where

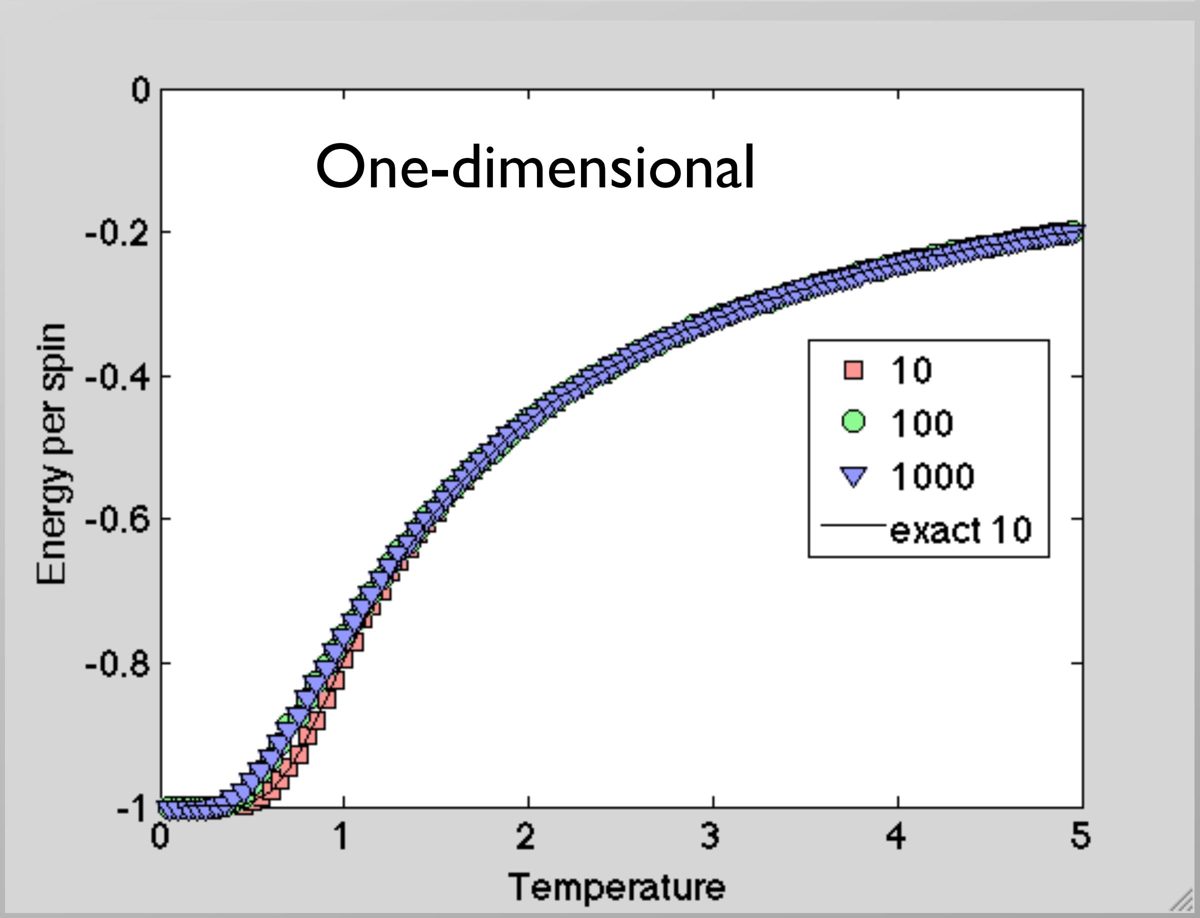
$$m_s = \frac{1}{N_s} \left| \sum_i \mathrm{sgn}(i) s_i \right|, \tag{3}$$

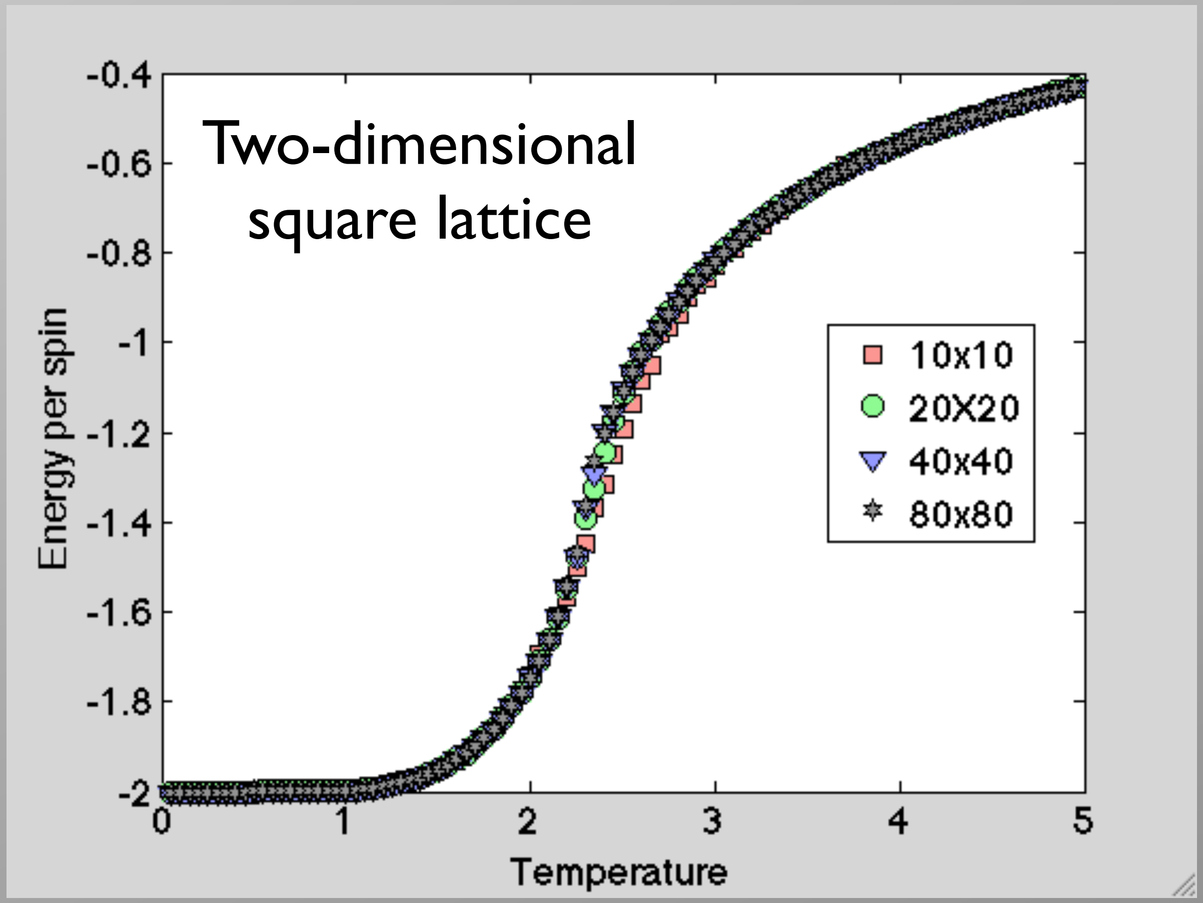
and $\mathrm{sgn}(i) = \pm 1$ so that neighbouring sites always have opposite values of $\mathrm{sgn}(i)$[2]. We consider lattices of size $L \times L$ with periodic boundary conditions. The lattice of size $L = 4$ is a good starting point, but try to reach as large lattices as possible within reasonable running times. Note that lattices with odd L are somewhat anomalous (think about why), so it is best to use even values L.

Difference between one and two dimensions









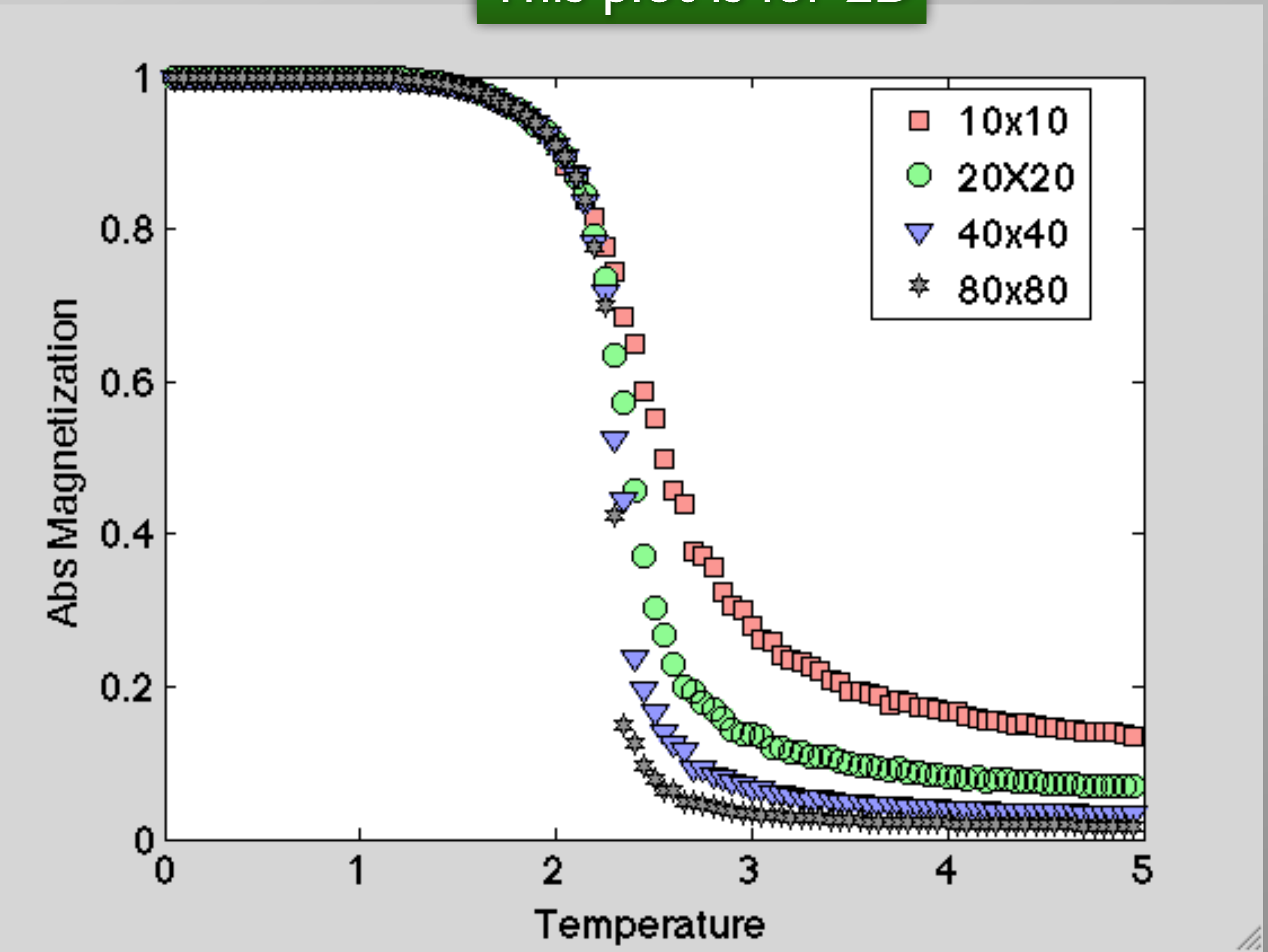
Phases of the Ising model

1. If $T = 0$, all spins are either up or down, and no spin flips possible as $\exp\left(-\Delta E/kT\right) \to 0$. This state has magnetization $m = \frac{1}{N}\sum_i \sigma_i = \pm 1$.
2. If $\beta = 0$ $(T = \infty)$, all states have same weight, and spins are random. No magnetization ($m = 0$).

What happens between these limits?

1D: Smooth transition to lower magnetization.

2D: Sudden transition, phase transition?



This plot is for 2D

Mean-field analysis

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j \approx \left\langle -J \sum_{\langle i,j \rangle} \sigma_i \right\rangle \sigma_j = -Jzm\, \sigma_j$$

Approximate neighbors with mean of the spin m (magnetization).

Number of nearest neighbors is z.

Problem of one spin, two states with energies $\pm Jzm$ (solved in intro to stat. phys. above).

Expectation value of spin σ_j is $\langle \sigma_j \rangle = \tanh(zJm/kT)$.

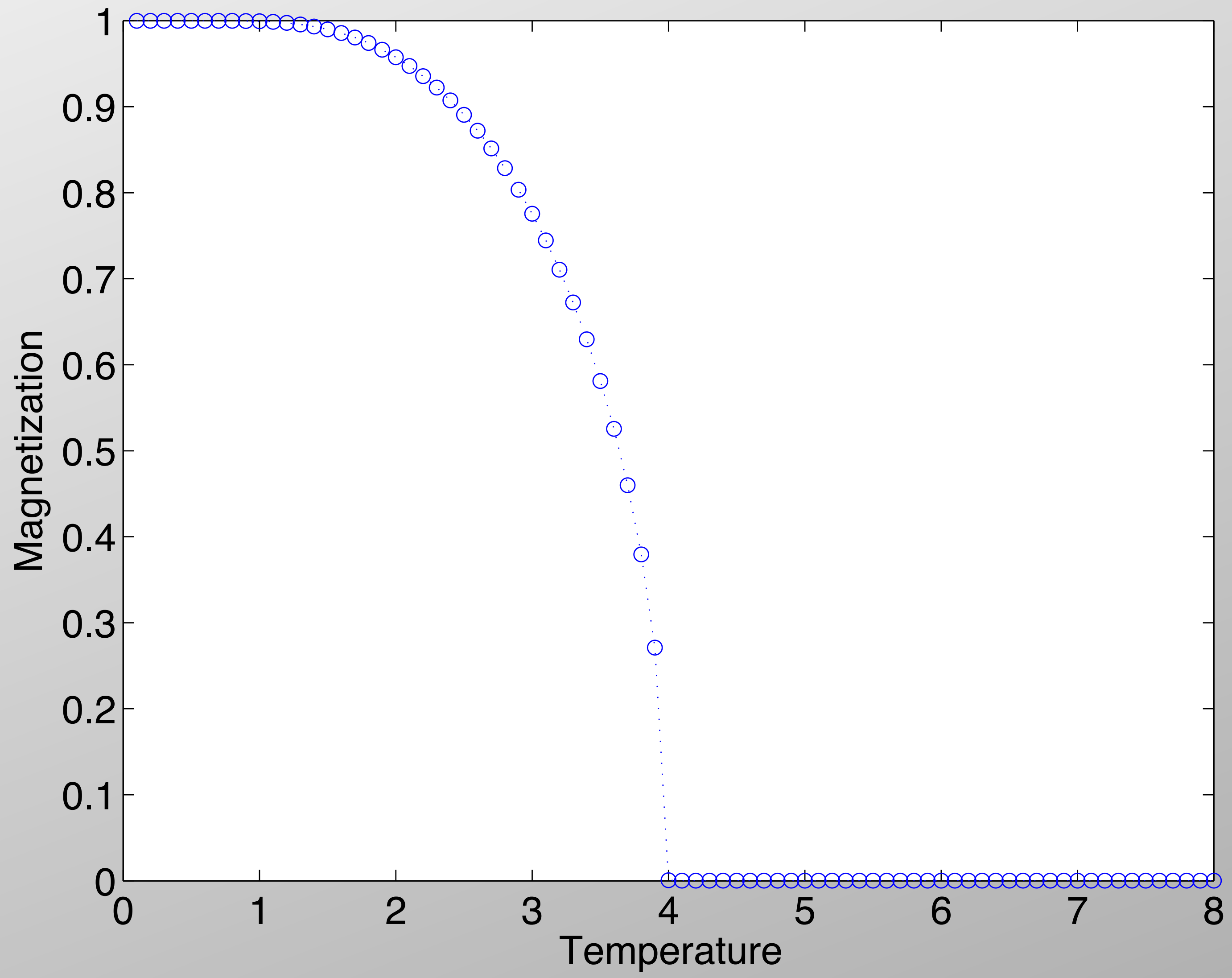
But $\langle \sigma_j \rangle$ should be m!

We get $m = \tanh(zJm/kT)$.

Need of self-consistent solution, plug m in on RHS, get new m on LHS.

Continue until converged.

Mean-field for 2D (z=4)



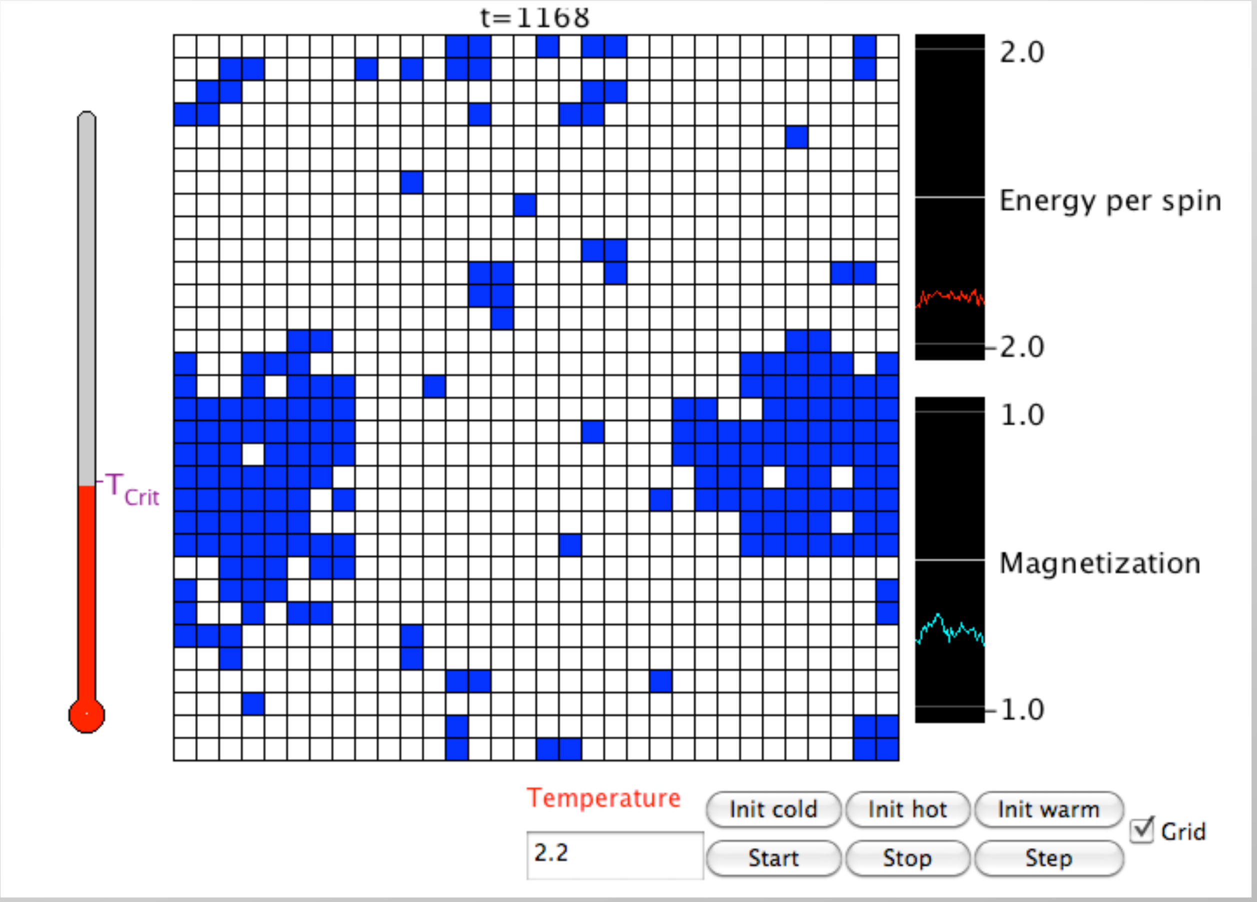
At small x, $\tanh(x) \approx x - x^3/3$, so that

$$m = \frac{zJm}{kT} - \frac{1}{3}\left(\frac{zJm}{kT}\right)^3$$

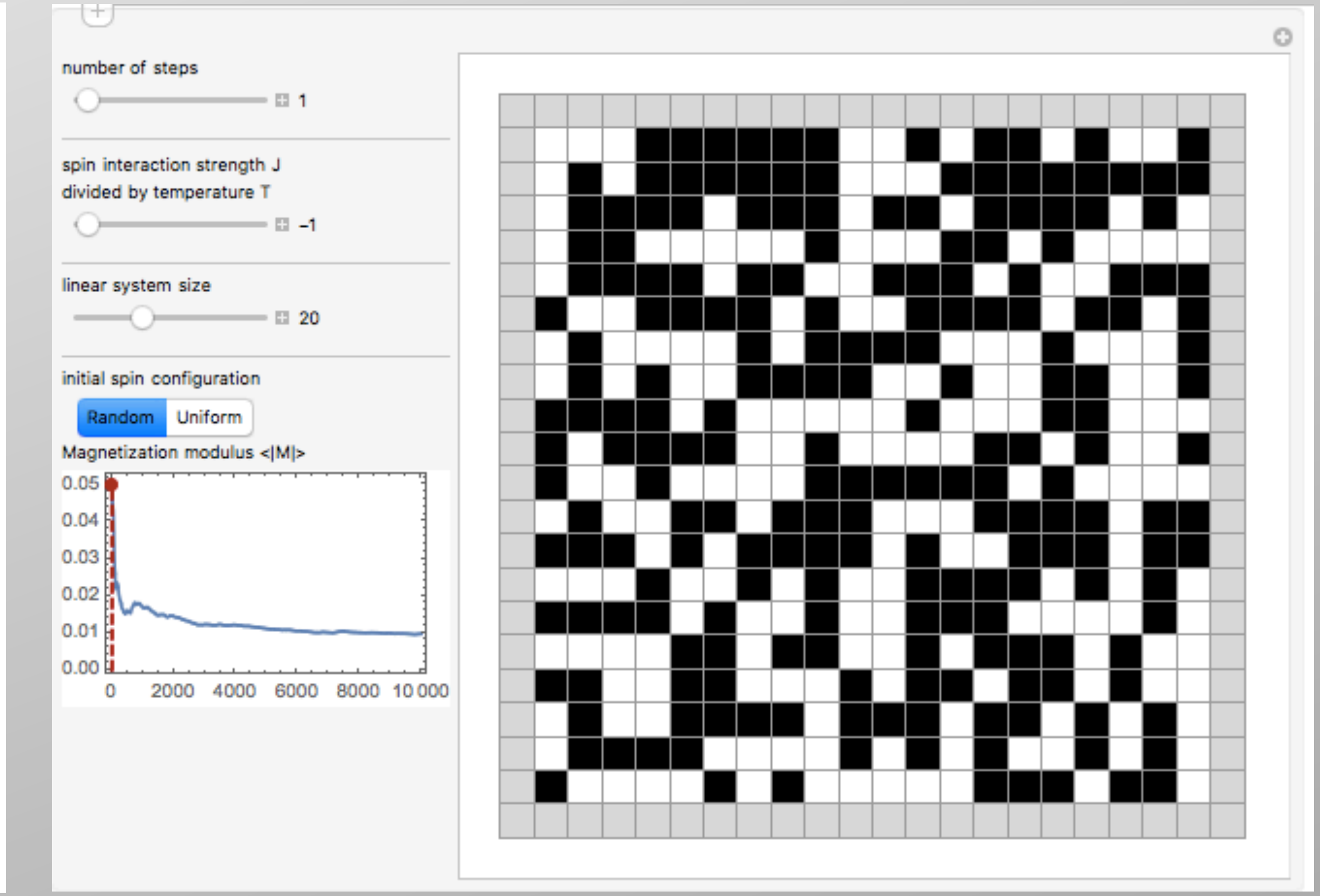
Solutions $m = 0$ and $m \propto (zJ/k - T)^{\frac{1}{2}(=\beta)}$, $T_C = zJ/k$, β is critical exponent.

Metropolis, 2D Ising model

You can get visual understanding from the Java/Javascript/HTML5/
Mathematica applets available, like:



Temperature
2.2



number of steps 1

spin interaction strength J divided by temperature T −1

linear system size 20

initial spin configuration: Random | Uniform

http://physics.ucsc.edu/~peter/ising/ising.html
(Java, but up-to-date browsers don't support it anymore)
https://physics.weber.edu/schroeder/software/demos/IsingModel.html (Javascript + HTML5)
http://demonstrations.wolfram.com/The2DIsingModelMonteCarloSimulationUsingTheMetropolisAlgorit/
(Mathematica)

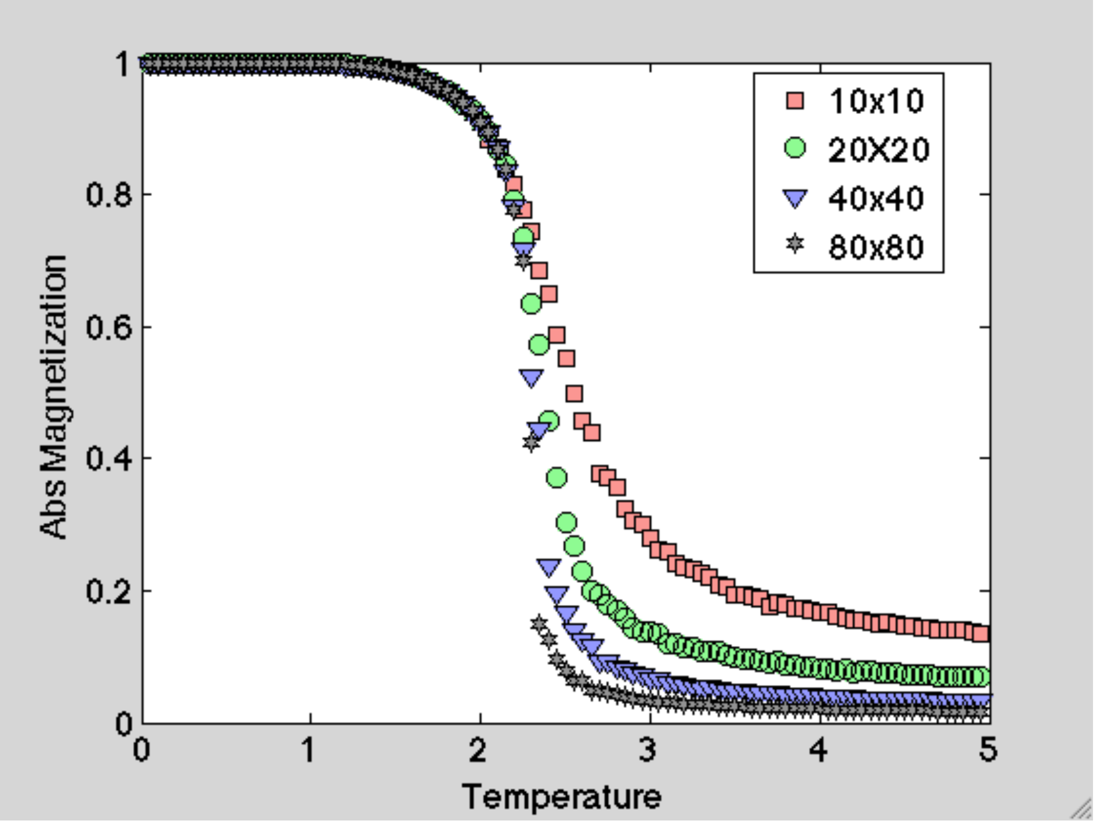
Average value of spin is magnetization.

Abs of this as a function of temperature:

(Averaged over lattice sites and simulation steps)

This is an example of something that we call

Phase transition



How to notice a phase transition? Liquid-gas system, change in density.

Most transitions have an **order parameter** that is finite in ordered phase and zero in disordered phase.

Order parameter can be, e.g., density (like above) or magnetization.

Finite system can not have a phase transition, as all physical variables are analytic functions of the system parameters.

Possible in thermodynamic limit (TDL): $N, V \to \infty$ (N/V=constant).

Consequence:

Finite systems can have smooth transitions that correspond to phase transition in TDL.

So TDL is needed for ising, also. Broken symmetry (and no ergodicity) in TDL.

Phase transition, 2D Ising model

At transition point energy and entropy in balance.

“Peierls argument”

Domain wall of length N highers energy by $2JN$.

Possible domain wall configurations between 2^N and 3^N. (Wall can turn to two or three directions each site.)

Entropy:

$$k\ln(2^N) < S < k\ln(3^N)\ .$$

When $TS = U$,

$$kTN\ln(2) < 2NJ < kTN\ln(3)\ ,$$

or

$$1.8 < kT/J < 2.9\ .$$

Exact in TDL: $kT_c = \dfrac{2J}{\ln(1+\sqrt{2})} \approx 2.269\,J\ .$

Critical exponents, 2D Ising model

m vanishes when $T \to T_c$ (from below) as $m \propto (T - T_c)^{\beta}$.

Defines exponent β.

Similarly,

$$C_V \propto |T - T_c|^{-\alpha}\,, \quad (T > T_c)$$

$$C_V \propto |T - T_c|^{-\alpha'}\,, \quad (T < T_c)$$

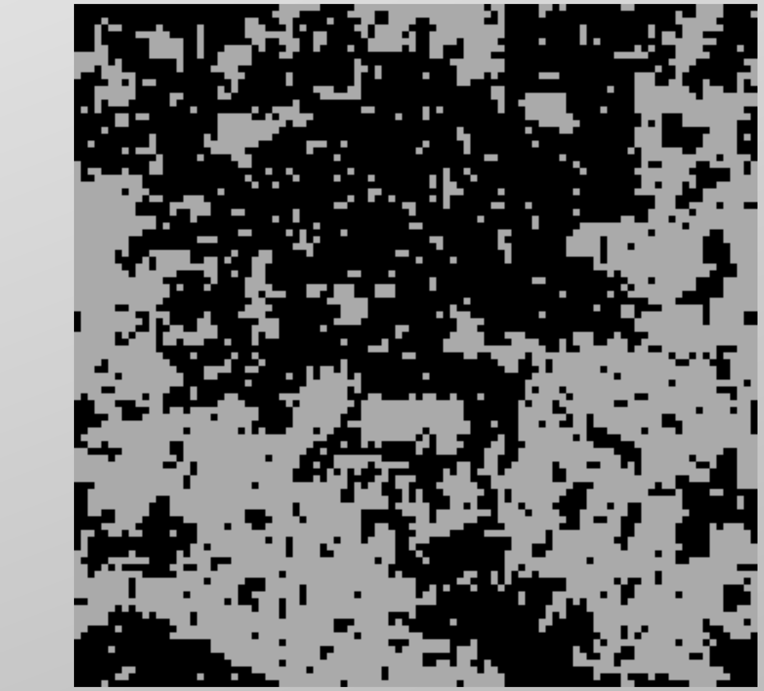
Exact results: $\alpha = \alpha' = 0$ (log) and $\beta = \frac{1}{8}$.

And log means that actually

$$C_V \propto -\ln\left|1 - \frac{T}{T_c}\right|.$$

Critical slowing down of Metropolis

In the simple Metropolis, only one spin is flipped. Information travels only one lattice spacing per step. Spatial correlation length of spins is ξ. To reach a new configuration, information must travel length ξ. When T approaches Tc, ξ reaches the system size. Computational efficiency of Metropolis goes to zero. This is called critical slowing down. Flipping cluster can cure this.



Flipping more than one spin.

1. Randomly choose a site (called i below).
2. Draw bond to all nearest neighbors j with probability $1 - \exp(-\frac{J}{kT}\delta_{\sigma_i,\sigma_j})$, so only same spins can have a bond.
3. If bond formed, repeat previous step to all nearest neighbors of j. Continue until no new bonds formed.
4. Flip all spins bonded together (cluster).

This algorithm reduces critical slowing down.

More complex than simple Metropolis.

This is the Wolff algorithm, see Phys. Rev. Lett. 62, 361 (1989).