1 Announcements

1. Master students: Spoken Exam, 30 minutes: official date **18 February 2011**.
2. Master students can register for the exam after 17 November, but no later than one week before the exam date! If you miss the deadline you are out.
3. Register via the Flexnow system.
4. Information about the MC course can be found here:
   `http://www.theorie.physik.uni-goettingen.de/~vink/mc_lecture/course/`
   (or simply google "richard vink goettingen" and you will also find it.

2 Summary of Lecture 1

1. MC method of performing a 1D integral:
   $$I = \int_a^b f(x)dx \approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i),$$
   with the sum over N (uniform) randomly selected numbers $a < x_i < b$.
2. The root-mean-square error σ decays as $\sigma \propto 1/\sqrt{N}$, irrespective of the dimension of the integral → consequence of the central limit theorem.
3. This becomes competitive over non-random numerical methods (eg: midpoint rule, Simpson integration) when the number of integration variables exceeds $d > 10$ or so (ie: for highly dimensional integrals → please do not confuse the dimension of the integral with the dimension of space).
4. In Statistical Physics : $d \gg 10$ in practice → hence MC is valuable.

3 Lecture 2

OUTLINE

1. Apply last weeks MC integral method to a simple many-body problem
   → **1D Ising model** = "fruit fly" of Statistical Physics.
2. Illustrate the failure of last weeks method → require one more ingredient
   → **Importance Sampling**.
3. **End of Lecture** : you will have learned a general MC method which can be applied to any (classical) many-body problem in equilibrium.
4. **Literature** : "Monte Carlo Methods in Statistical Physics, Newman and Barkema", Chapters 2-3.

3.1 1D Ising model

1. Set of $i = 1, \ldots, N$ spins $\sigma(i)$ on a line. The spin variables can assume two values : $\sigma(i) = 1$ (up) or $\sigma(i) = -1$ (down).

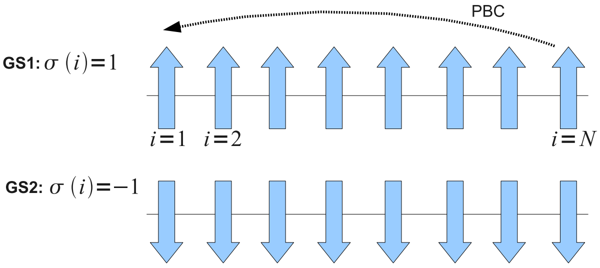
2. Only neighboring spins interact. The energy reads as:

$$E = -J \sum_{i=1}^{i=N} \sigma(i-1)\sigma(i),$$

   with **periodic boundary condition** $\sigma(0) = \sigma(N)$. J is called the exchange or coupling constant.

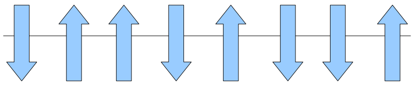
3. We consider the **ferromagnetic** case $J > 0$: alignment of neighboring spins lowers the energy.

4. There are two (equivalent) ground states (GS) which minimize the energy:



   In the **GS** : all spins point the same way, ie: they are **ordered** states
   → the ground state energy equals $E_{\rm GS} = -JN$

5. There are, of course, also many other states, for example:



   → All non-ground states have energy $E > E_{\rm GS}$.
   → Finite total number of states : 2^N.

3.2 Statistical Physics

1. Ising model can be in many different states, eg: two ground states, and many other states (including completely disordered ones).

2. For each state μ, one can calculate the Ising energy E_μ.

3. **Gibbs, 1902** : crucial result of Statistical Physics : in **thermal equilibrium**, the probability of state μ occurring decays exponentially with the energy E_μ of that state:

$$P_\mu \propto \exp\left(-\beta E_\mu\right),$$

   with inverse temperature $\beta = 1/k_B T$.

4. Ensemble (thermal) averages become weighted sums:

$$\langle Q \rangle = \frac{\sum_\mu Q_\mu \exp\left(-\beta E_\mu\right)}{\sum_\mu \exp\left(-\beta E_\mu\right)},$$

   where the **sum** is **over all states** μ that the system can possibly be in. Q can be **any** quantity of interest (for example, the energy E).

5. Normalization $Z \equiv \sum_\mu \exp\left(-\beta E_\mu\right)$ is called the partition sum.

6. 1D Ising model : discrete states : number of states is finite $\rightarrow 2^N$.

7. Continuous systems : fluids, membranes, most interesting systems :
   number of states $\rightarrow \infty$ meaning that $\sum \rightarrow \int$.

3.3 The average energy $\langle E \rangle$ of the 1D Ising model

1. 1D Ising model : the partition sum Z can be evaluated exactly:

$$Z = 2^N \cosh^N(\beta J) + 2^N \sinh^N(\beta J).$$

2. Hence $\langle E \rangle = -\frac{\partial \ln Z}{\partial \beta}$ is also known exactly.

3. We can also use MC “integration” (summation) to determine the average energy :

$$\langle E \rangle = \frac{\sum_\mu E_\mu \exp(-\beta E_\mu)}{\sum_\mu \exp(-\beta E_\mu)},$$

by summing over **many randomly chosen spin configurations** μ.

4. **Algorithm** : (1) generate random spin configuration μ. (2) calculate its energy E_μ. (3) update weighted average. (4) Repeat many times.

5. **Results** : do test for small system ($N = 10$ spins) and large system ($N = 50$) at several values of the inverse temperature β. Monitor the convergence of:

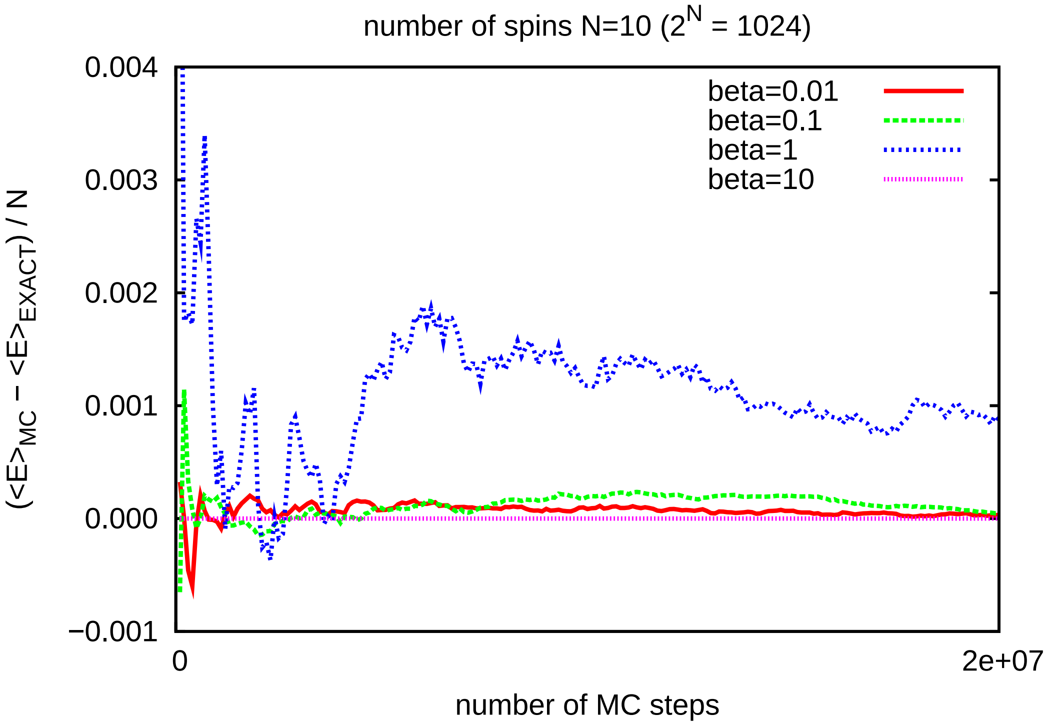
$$\frac{\langle E \rangle_{\rm MC} - \langle E \rangle_{\rm EXACT}}{N},$$

which should approach **zero** with increasing number of MC steps.

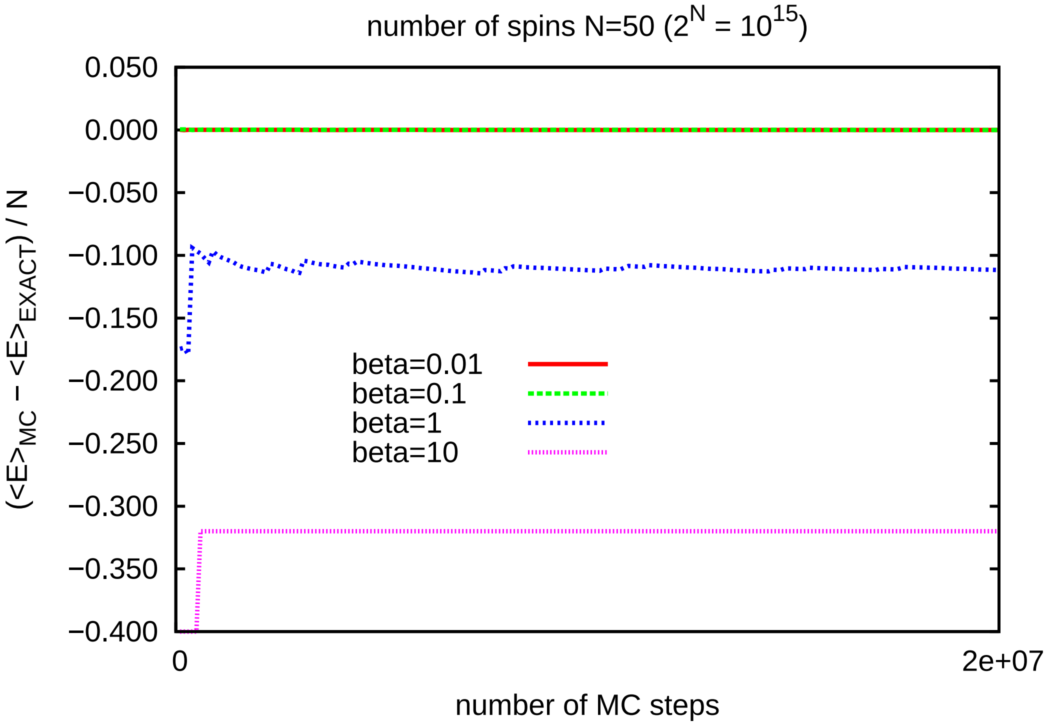
6. $N = 10$ system : convergence for all values of β → OK!
note 1 : convergence is extremely fast for high β, relatively fast for low β, and slow for “intermediate” β.
note 2 : letting the computer do an exact sum over all states would be faster here, since there are only $2^{10} = 1024$ states, ie, much **less** than the number of MC steps used.

7. $N = 50$ system : → only convergence for high β, otherwise complete failure.
note : number of states 2^N is very high so an exact summation over all states is **not** computationally feasible.

Simulation 1



Simulation 2

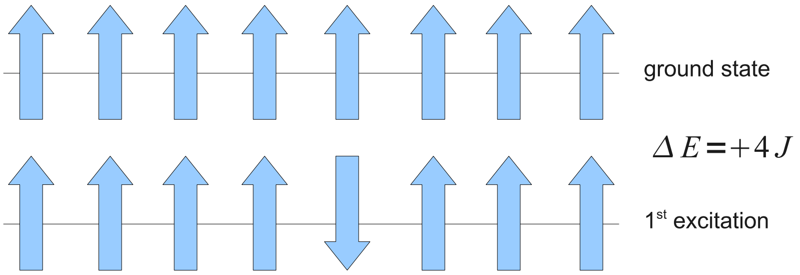


3.4 Understanding the result

1. Consider again the state probability : $P_\mu \propto \exp\left(-\beta E_\mu\right)$.

2. **high temperature limit** : $\lim_{\beta \to 0} P_\mu \to \text{constant}$.
   → All states contribute with **equal weight** to the thermal average.
   → good convergence at high T because every randomly selected state μ is a "good" one, in the sense that it contributes to the thermal average!

3. **low temperature limit** : thermal average is heavily **dominated** by **low energy states**. Extreme limit $\beta \to \infty$: only the two ground states contribute!



4. Energy cost of flipping one spin in the GS equals $4J$. Probability ratio:

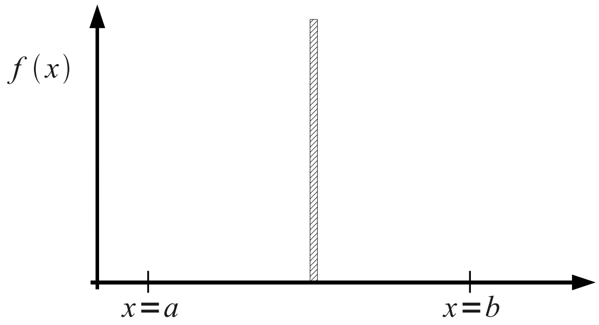
$$\frac{P_{\text{1st excitation}}}{P_{\text{groundstate}}} = e^{-4\beta J},$$

   which becomes negligible as $\beta \to \infty$.

5. We can only expect good convergence at low T if the randomly selected states μ contain the two ground states:
   → The probability of picking a ground state equals $2/2^N$.
   → Note that the test "simulation" used 20 million random states.
   → The ground states are likely to be seen for $N = 10$ ($2^{10} = 1024$) : **good convergence**.
   → Very unlikely for $N = 50$ ($2^{50} \sim 10^{15}$) : **bad convergence**.

6. **Conclusion** : evaluating thermal averages by summing over randomly selected spin states is **not** efficient at finite temperature. The problem is that the thermal average is dominated by only a rather small number of "special" states. The probability that a randomly selected state happens to be one of these "special" states is negligible in large systems.

7. Compare this to the integral $I = \int_a^b f(x)\,\mathrm{d}x$ of a function $f(x)$ that is zero everywhere on the interval $a < x < b$, except for a small peak:

   Clearly, the "standard" MC approach of evaluating this integral:

   $$I \approx \frac{b-a}{N} \sum_{i=1}^{N} f(x_i),$$

   with the sum over N randomly selected points $a < x_i < b$ is inefficient, since most of these points are likely to be **outside** the peak region, and so do not contribute to the integral (even though you do waste computer time on these points).

8. In some sense, the partition sum Z at finite temperature is also an integral over a sharply peaked function (!)
   $\rightarrow$ Maybe even several peaks (phase coexistence).

9. Compare this to the "circle-in-square" algorithm of last week, in the limit where the square is much larger than the circle.

3.5 Importance Sampling

1. Since only a **subset** of the total number of spin states contributes significantly to the thermal average, it would be nice to have an improved Monte Carlo scheme which precisely picks those states (as opposed to picking them completely random as we did before).
   $\rightarrow$ This is called **Importance Sampling**.

2. **define** M : total number of MC steps taken during the simulation. Note that, at each step, a spin state μ is selected "in some way".
   $\rightarrow$ M should be as large as possible : set in practice by available computer power.

3. **define** g_μ : the probability that state μ is selected during any one MC move.
   $\rightarrow$ in the previous **inefficient** 1D Ising simulation, states were chosen with equal probability $g_\mu = 1/2^N$.

4. **Importance Sampling** : for better efficiency, try to use a different g_μ, such that low energy states are picked more frequently than high energy ones.
   $\rightarrow$ how this can be done in practice will be explained shortly; just assume for now this is possible.

5. **Consequence** : since some states are now selected more often than pure chance, a "correction" in the thermal average is needed:

   $$\langle Q \rangle = \frac{\sum_{\mu=1}^{M} Q_\mu (1/g_\mu) \exp(-\beta E_\mu)}{\sum_{\mu=1}^{M} (1/g_\mu) \exp(-\beta E_\mu)}.$$

   motivation : if g_μ is "high", state μ is selected too often, and hence should be counted with a reduced weight $1/g_\mu$.
   note : using $g_\mu = \mathrm{constant}$ yields the previous (inefficient) form again.

6. The **optimal** choice is:

   $$g_{\mu,\mathrm{opt}} \propto \exp(-\beta E_\mu),$$

   since then it holds that:

   $$\langle Q \rangle = \frac{1}{M} \sum_{\mu=1}^{M} Q_\mu,$$

   ie: all the states contribute with equal weight.
   note 1 : the **inefficient** 1D Ising simulation also has this property, but only in the high temperature limit $\beta \rightarrow 0$.
   note 2 : the advantage of **Importance Sampling** is that the above property holds at any temperature.

7. **Remaining question** : how to efficiently generate a set of states $\{\mu = 1, \ldots, M\}$ such that the probability of state μ occurring is proportional to $e^{-\beta E_\mu}$.

3.6 The Metropolis algorithm

1. The **Metropolis algorithm** is designed to efficiently generate many states μ, such that the probability of state μ occurring is proportional to $\exp\left(-\beta E_\mu\right)$.
   $\rightarrow$ the Metropolis algorithm it is a **Markov process**.

2. **Markov process** : states are not generated from scratch each time, but instead the current state μ is modified in some stochastic manner to yield a new state ν.
   (1) Assume we are in state μ.
   (2) Propose a new state ν by making a (random) change to state μ.
   (3) Accept or reject the new state ν with some probability.
   (4) Repeat using the state selected in (3) as starting point.

3. Examples of common **Monte Carlo moves** for step (2):
   $\rightarrow$ Ising model : randomly select a spin and flip it.
   $\rightarrow$ Fluid : randomly select a molecule and displace it over a random vector.

4. Note that states μ and ν will generally be correlated.

5. Let $T(\mu \rightarrow \nu)$ be the probability that, being in state μ, the next step in the Markov chain yields state ν.

6. $T(\mu \rightarrow \nu)$ is the **product** of the proposal (2) and accept (3) probabilities:

   $$T(\mu \rightarrow \nu) = s(\mu \rightarrow \nu) a(\mu \rightarrow \nu),$$

   with s the probability that, being in state μ, state ν is proposed, and a that of being accepted.
   $\rightarrow$ Ising spin flip : $s(\mu \rightarrow \nu) = 1/N$.

7. **Theorem**: the Markov chain will generate state μ with the desired probability $\propto e^{-\beta E_\mu}$ provided it is **ergodic**, obeys **detailed balance**, and has been run long enough (**equilibrated**).

8. **Ergodicity** : the Markov chain must in principle be able to reach all states.

9. **Detailed balance** (very important):

   $$g_\mu T(\mu \rightarrow \nu) = g_\nu T(\nu \rightarrow \mu).$$

   $\rightarrow$ $g_\mu \propto e^{-\beta E_\mu}$ is the desired probability of occurrence.
   $\rightarrow$ proportionality constant in g_μ cancels in the detailed balance condition.

4 Metropolis spin flip algorithm for the Ising model

1. **Algorithm** : to simulate an Ising model of N spins as follows:
   (1) Choose a spin at random and flip it.
   (2) Accept or reject the flip with probability $a(\mu \to \nu)$.

2. **Question** : how to choose $a(\mu \to \nu)$ optimally?
   → detailed balance condition reads as:

   $$\frac{a(\mu \to \nu)}{a(\nu \to \mu)} = \frac{s(\nu \to \mu)}{s(\mu \to \nu)} e^{-\beta(E_\nu - E_\mu)}.$$

   → proposition probabilities $s(\mu \to \nu) = s(\nu \to \mu) = 1/N$ cancel out:

   $$\frac{a(\mu \to \nu)}{a(\nu \to \mu)} = e^{-\beta(E_\nu - E_\mu)},$$

   yielding a **first constraint** for the ratio between the accept probabilities.
   → being probabilities, there is also the **second constraint** : $0 \leq a \leq 1$.

3. **Aim** : maximize $a(\mu \to \nu)$ within the bounds permitted by the **two** constraints.
   (1) Suppose a move $\mu \to \nu$ **increases** the energy : $E_\nu > E_\mu$.
   (2) Detailed balance constraint then implies: $a(\nu \to \mu) > a(\mu \to \nu)$.
   (3) The optimal choice is to maximize $a(\nu \to \mu)$ as far as allowed by the second constraint:

   $$a(\nu \to \mu) = 1.$$

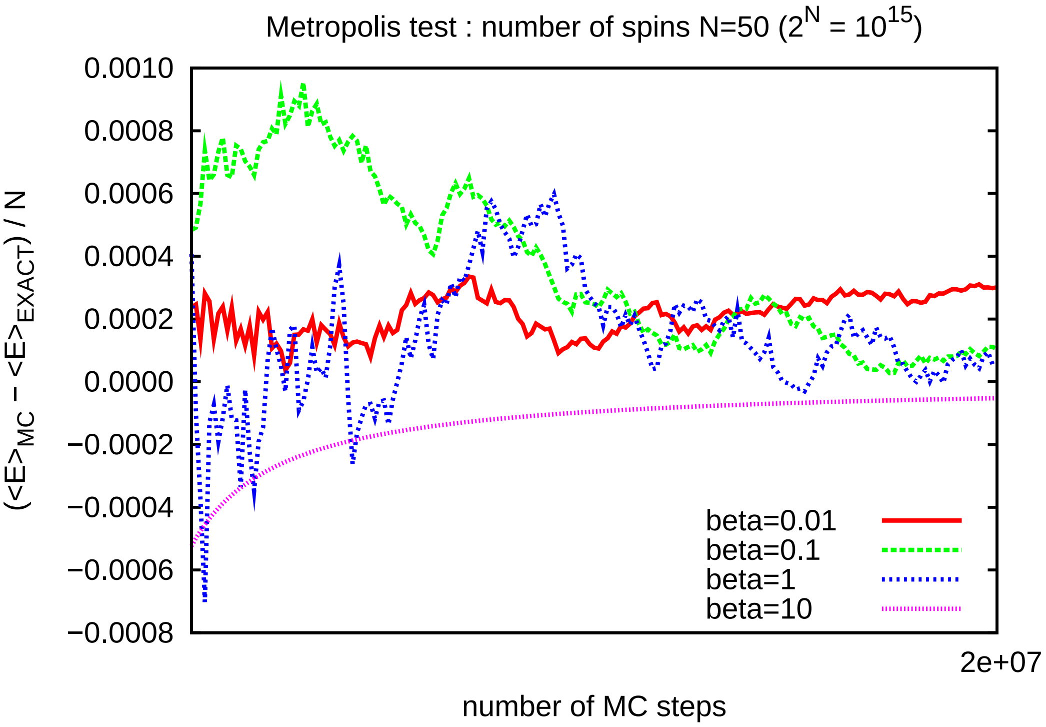
   Note that the move from $\nu \to \mu$ **lowers** the energy, and so these moves are always accepted.
   (4) Consequently: $a(\mu \to \nu) = e^{-\beta(E_\nu - E_\mu)}$.

4. The above acceptance criterion is the famous **Metropolis algorithm**, and can be conveniently summarized as:

   $$a(\mu \to \nu) = \min\left[1, e^{-\beta(E_\nu - E_\mu)}\right].$$

   → can be used for any system where $s(\mu \to \nu) = s(\nu \to \mu)$.

Demo : Metropolis sampling of the 1D Ising model



1 Summary of previous lecture

1. **Importance Sampling**
   $\rightarrow$ do not generate state μ randomly but according to:
   $$p_\mu \propto e^{-\beta E_\mu},$$
   with $\beta = 1/k_B T$ the inverse temperature, and E_μ the energy of state μ.

2. If states appear proportional to $p_\mu \propto e^{-\beta E_\mu}$, then the ensemble average takes the simple form:
   $$\langle Q \rangle = \frac{1}{M} \sum_{\mu=1}^{M} Q_\mu,$$
   with M the total number of states generated during the simulation.
   $\rightarrow$ ie: all states contribute with equal weight and so one does not “waste time” with states that contribute hardly anything to the thermal average.

3. An efficient method to generate states μ according to a desired target distribution p_μ is the **Metropolis algorithm**, which is an example of a **Markov process**.

OUTLINE FOR TODAY

1. **Revisit** the Metropolis algorithm (last week I ran out of time).

2. Derive a Metropolis Monte Carlo scheme to study an important physical phenomena, namely **phase separation in simple fluids**.

2 The Metropolis algorithm revisited

1. In the **Metropolis algorithm**, one uses the **current** state μ to generate the **next** state ν, by applying a **Monte Carlo move** to state μ.
   $\rightarrow$ MC moves are **stochastic**, ie, they involve **random numbers**.
   $\rightarrow$ the applied changes are **usually small**.

2. Famous example : **spin flip** in the Ising model.
   $\rightarrow$ choose one of the N spins randomly and change its orientation.

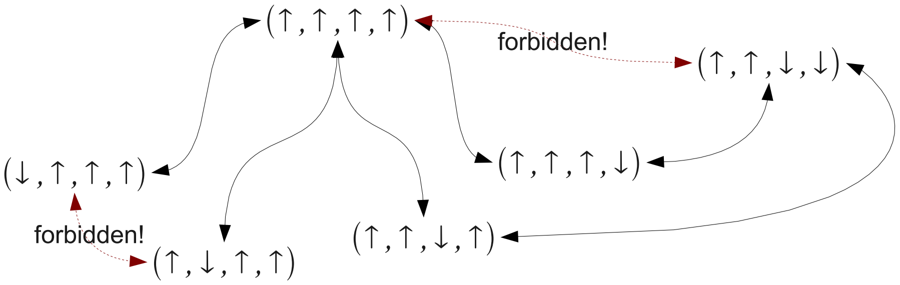
3. Define the **transition probability** $T(\mu \rightarrow \nu)$:
   this is the probability that, being in state μ, the next step of the algorithm brings you to state ν.
   $\rightarrow$ Since MC moves typically involve **small changes**, not all states ν can be reached from μ with just one move. Hence, many of the transition probabilities are zero. For the validity of a MC scheme this poses no problem.
   $\rightarrow$ **However**, it is important that at least **some** states $\nu \neq \mu$ can be reached. Hence, we do require that $T(\mu \rightarrow \nu) > 0$ for at least some states ν.

4. The starting state μ and next state ν are allowed to be identical : $T(\mu \rightarrow \mu) \neq 0$.

5. $T(\mu \rightarrow \nu)$ should only depend on the **current** state μ and not on any states that were visited before, ie: there is no **history dependence** in the transition probability.

6. One could "visualize" the **Metropolis algorithm** as being some kind of network (graph). Points (nodes) in the network correspond to states, and arrows between points indicate that a transition between these states is **in principle possible** using a **single** Monte Carlo move.

Example : Ising model with N=4 spins using single spin flip moves


$(\uparrow,\uparrow,\uparrow,\uparrow)$
forbidden!
$(\uparrow,\uparrow,\downarrow,\downarrow)$
$(\downarrow,\uparrow,\uparrow,\uparrow)$
$(\uparrow,\uparrow,\uparrow,\downarrow)$
forbidden!
$(\uparrow,\uparrow,\downarrow,\uparrow)$
$(\uparrow,\downarrow,\uparrow,\uparrow)$

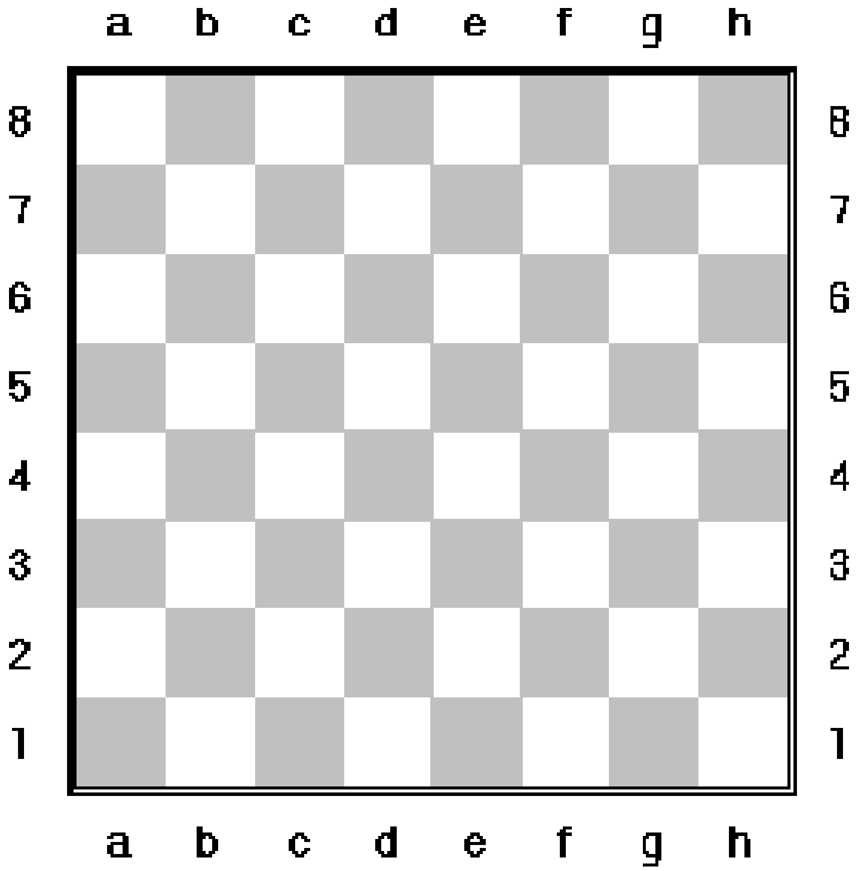
note: not all forbidden transitions are shown!

→ **note** : using spin flip moves, it is always possible to return from $\nu \to \mu$, and so **double arrows** are used. This is called **reversibility**, and is actually an important requirement (see later).

7. What conditions are required such that the states in the "network" are visited with a desired target probability p_μ?
→ **ergodicity** : it must be possible to reach all states of the system in a finite number of Monte Carlo moves (spin flips are obviously ergodic).
→ **detailed balance** : $p_\mu T(\mu \to \nu) = p_\nu T(\nu \to \mu)$.

8. Important **consequence of detailed balance** : reversibility
→ If a transition $\mu \to \nu$ is possible, ie: $T(\mu \to \nu) > 0$, **and** with sensible target probabilities $p_\mu > 0$ and $p_\nu > 0$, detailed balance implies that also the reverse transition probability $T(\nu \to \mu) > 0$. To see this, simply rewrite the detailed balance equation as:
$$T(\nu \to \mu) = \frac{p_\mu}{p_\nu} T(\mu \to \nu).$$
→ **In words**:
If at MC step i you jumped from state $\mu \to \nu$, then in the next step $i+1$ it must be possible to make the reverse transition $\nu \to \mu$ with finite probability (hence the use of double arrows in the network representation). This **reversibility property** is a **minimal but not sufficient** criterion a valid MC scheme!

9. The requirements of **ergodicity** and **detailed balance** thus put a number of conditions (constraints) on the MC moves that we can use.
   → **pedagogical illustration** : consider the game of chess, and let each square represent one state. Assume a board with just one piece on it.



→ **Question** : which "pieces" can **visit all states** (ie: are ergodic) and which pieces do so **reversibly** (ie: fulfill the minimal detailed balance requirement).

- King/Queen/Rook : ergodic and reversible.
- Bishop : NOT ergodic but reversible.
- Pawn : neither ergodic nor reversible.
- Knight : reversible, but also ergodic? (yes → the Knight's Tour problem).

10. Provided one has a MC move which meets the minimum requirements of **ergodicity** and **reversibility**, one can proceed to derive the **acceptance probability** needed to sample states proportional to the desired target probability p_μ.

11. To this end, write the transition probability as a two-stage process:

$$T(\mu \to \nu) = g(\mu \to \nu) \times a(\mu \to \nu).$$

$\to$ $g(\mu \to \nu)$ is the probability that, being in state μ, a move to state ν is **proposed**.
$\to$ $a(\mu \to \nu)$ is the probability that, if state ν has been **proposed**, how likely is it to be **accepted**.

12. Substitution into the detailed balance equation, and using the condition that $0 \leq a(\mu \to \nu) \leq 1$, the optimal choice for the accept probability in many physical situations becomes:

$$a(\mu \to \nu) = \min \left[1, \frac{g(\nu \to \mu)}{g(\mu \to \nu)} \frac{p_\nu}{p_\mu} \right],$$

which defines the **Metropolis algorithm**.
$\to$ Monte Carlo moves accepted with the above probability will lead to a sequence of states $\{\mu_1, \mu_2, \ldots, \mu_M\}$, with the probability of state μ_i appearing proportional to p_{μ_i}.
$\to$ An **equilibration** period is needed, ie: the sequence must be long enough.

13. **Application** : for an Ising model containing N spins, using spin flips as Monte Carlo move, one has $g(\mu \to \nu) = g(\nu \to \mu) = 1/N$ and $p_\mu \propto e^{-\beta E_\mu}$, leading to:

$$a(\mu \to \nu) = \min \left[1, e^{-\beta(E_\nu - E_\mu)} \right].$$

$\to$ **In words** : Monte Carlo moves which **lower** the energy are always **accepted**, whereas moves which **increase** the energy are accepted with a probability that **becomes smaller** the higher the new energy is.

14. Monte Carlo moves for which $g(\mu \to \nu) = g(\nu \to \mu)$ are called **symmetric**. In this case, the proposition probabilities **cancel** from the acceptance probability.

$\to$ We will at some point also consider more complicated Monte Carlo schemes using **asymmetric** moves for which $g(\mu \to \nu) \neq g(\nu \to \mu)$.

3 Simulating simple fluids via Monte Carlo

1. Simple fluids consist (for most practical purposes) of **spherical molecules** that interact via a **pair potential**. The total energy is thus given by a sum over all pairs:

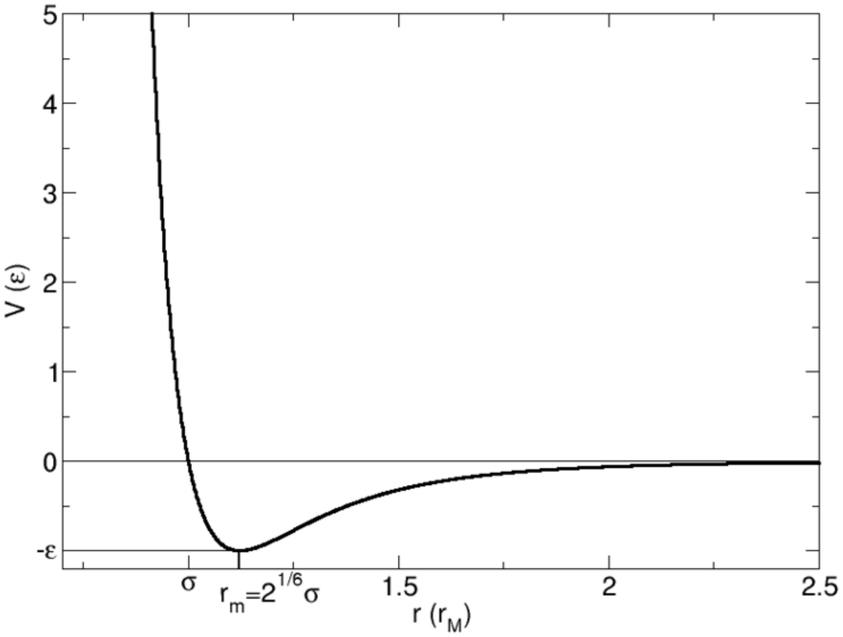
$$E = \sum_{i=1}^{N} \sum_{j=i+1}^{N} V(r_{ij}),$$

with $r_{ij} = |\vec{r}_j - \vec{r}_i|$ the distance between two molecules i and j, $\vec{r}_i$ the coordinate (position) of particle i, and N the total number of particles in the system.

2. In simple fluids, the pair potential typically resembles the Lennard-Jones form:

$$V(r) = 4\epsilon \left\{ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right\},$$

which looks like this:



3. The **essential features** are:
(1) In the limit $r \to 0$, the potential diverges $V(r) \to \infty$: the molecules **strongly repel** at short distances, ie, they do not overlap.
(2) At intermediate range, the molecules attract each other.
(3) At long distances $r \to \infty$, the potential quickly decays to zero.

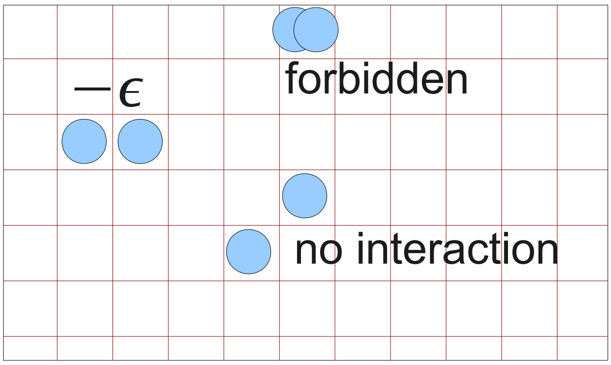
4. These essential features are already captured by the (very simple) **lattice gas model**.

4 The two-dimensional lattice gas

1. Particles live on the sites of a $L \times M$ rectangular lattice. There are two rules.

2. **Rule 1** : lattice sites may at most contain one particle.
   $\rightarrow$ this approximates the divergence of the LJ potential as $r \rightarrow 0$.
   $\rightarrow$ consequence : there can at most be $L \times M$ particles in the system.

3. **Rule 2** : when two particles occupy **nearest neighboring** sites, the energy is **lowered** by an amount ϵ.
   $\rightarrow$ this approximates the **attraction** between molecules at intermediate range.
   $\rightarrow$ on the square lattice, each site has four nearest neighbors (ie: we do not count diagonal neighbors).



4. The pair potential in the lattice gas thus assumes three values:

$$V_{ij} = \begin{cases} \infty & \text{if particles } i \text{ and } j \text{ occupy the same lattice site} \\ -\epsilon & \text{if particles } i \text{ and } j \text{ are nearest neighbors} \\ 0 & \text{otherwise} \end{cases}$$

   which (very crudely) approximates the LJ potential.

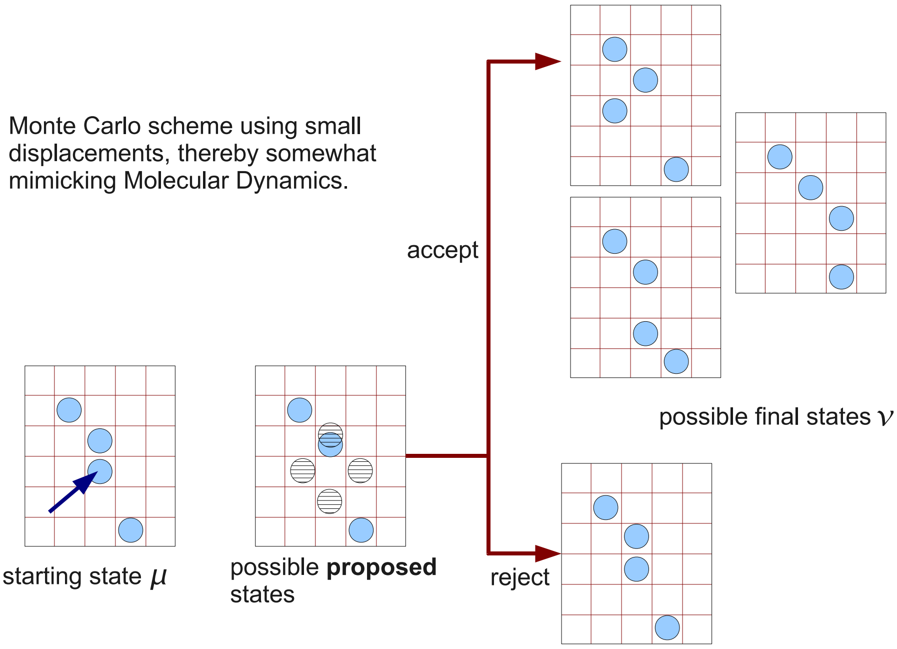
5. The total energy of the lattice gas is thus a sum over pairs as before:

$$E = \sum_{i=1}^{N} \sum_{j=i+1}^{N} V_{ij},$$

   with N the total number of particles.

5 (Lecture 4) (NVT)-Monte Carlo scheme for the 2D lattice gas

- **Aim** : to simulate the lattice gas on a $V = L \times M$ lattice (with periodic boundary conditions) at **fixed** particle density $\rho = N/V$ and temperature T.
- This is called the (NVT)-ensemble, because the number of particles N, the system "volume" V, and the temperature T are **constant**.
- What would be a valid Monte Carlo move for this problem?
- **Move 1** : try to "mimic" a **Molecular Dynamics** simulation, whereby particles make only small displacements per move.
- **On a lattice** : smallest possible displacement is a "hop" of a particle to one of its (four) nearest neighboring sites (sort of like a crippled rook in the game of chess). This naturally leads to the following Monte Carlo scheme:


Monte Carlo scheme using small displacements, thereby somewhat mimicking Molecular Dynamics.
accept
possible final states ν
starting state μ
possible **proposed** states
reject

1. Select one of the particles randomly in the starting configuration μ.
2. Tentatively put the selected particle at one of its four nearest neighboring sites chosen completely at random.
3. Decide to accept or reject the new particle configuration.

- The algorithm is **ergodic**, because each particle can in principle reach any lattice site.
- The proposition probabilities are **symmetric**:

$$g(\mu \to \nu) = g(\nu \to \mu) = \frac{1}{N} \times \frac{1}{4},$$

and so the moves can be accepted with the standard Metropolis acceptance probability:

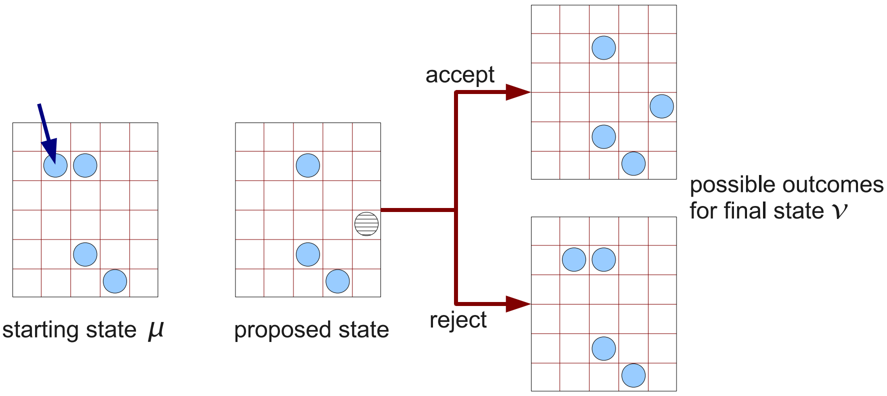
$$a(\mu \to \nu) = \min\left[1, e^{-\beta(E_\nu - E_\mu)}\right],$$

which enforces the **detailed balance** condition.

- **Note** : if a particle is proposed to be displaced onto an already occupied site, the resulting state ν would have an infinite energy $E_\nu \to \infty$, and so is automatically rejected (filtered out) by the Metropolis acceptance probability.

- **However**, it is not essential that we mimic Molecular Dynamics. All that is required are ergodicity and detailed balance. The following Monte Carlo move would be just as valid:

  Move 2 : random displacements of **arbitrary size** of randomly selected particles!



  1. Select one particle randomly.
  2. Place the selected particle at a randomly chosen new site.
  3. Decide to accept or reject the new configuration.

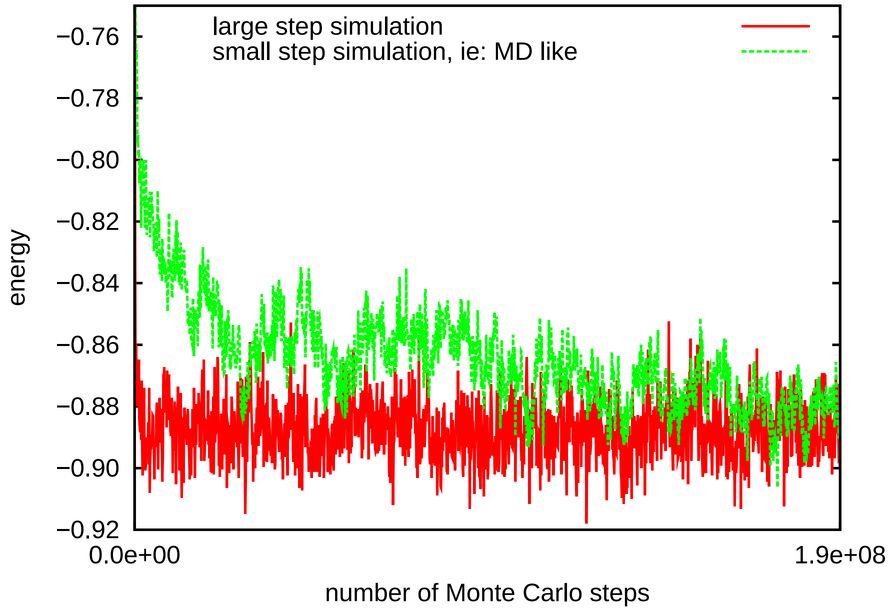
- This algorithm is clearly also **ergodic**. The proposition probabilities are again symmetric:

$$g(\mu \to \nu) = g(\nu \to \mu) = \frac{1}{N} \times \frac{1}{V},$$

  and so these moves are also accepted with the standard Metropolis acceptance probability.

- **Both algorithms** should yield identical results, but one may be faster (more efficient) than the other.

- **Comparing** both algorithms at $\beta\epsilon = 1.9$, $\rho = 1/2$, on a 50×50 lattice yields:



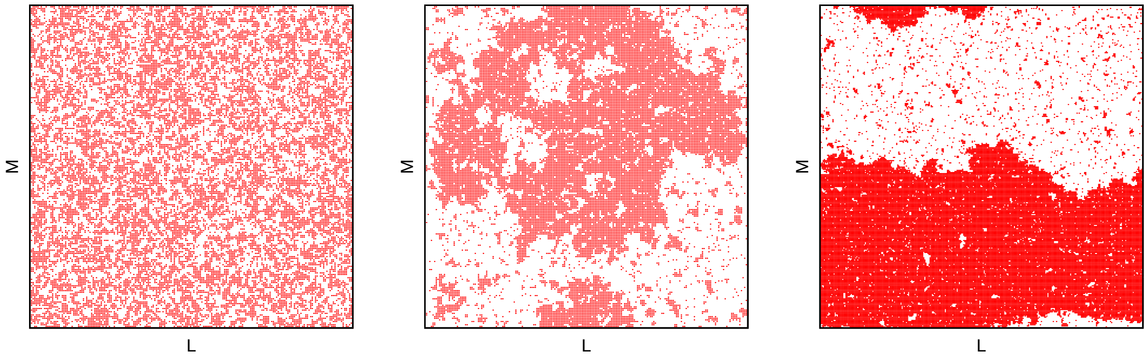
(the starting configuration of step 0 was a completely random one).

- The large-step algorithm equilibrates much faster!
  → **But**, be aware : the efficiency of Monte Carlo moves generally depends on factors such as temperature, density and system size so there is **no guarantee** that, if at one set of parameters one move outperforms another, it will always do so!

- **Important advantage** of Monte Carlo : one is allowed to move particles around however one likes, including in completely "unphysical" ways, as long as ergodicity and detailed balance are obeyed!
  → one could even envision moves whereby particles are created and destroyed, ie, simulate in the **grand canonical ensemble**.

6 More (NVT)-simulation results for the 2D lattice gas

- Whichever of the two moves we use, both should generate states μ (particle configurations) with probability $p_\mu \propto e^{-\beta E_\mu}$, after an **equilibration period**.
- What do such states look like?
- Typical simulation snapshot at density $\rho = 1/2$. The temperature **decreases** in the snapshots from left to right.

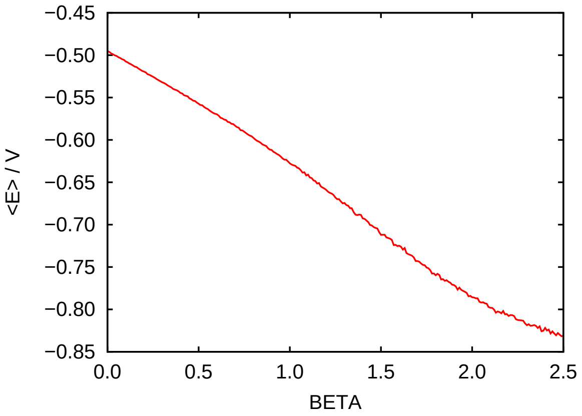


1. High temperature : homogeneous phase.
2. Intermediate temperature : many clusters of different sizes.
3. Low temperature : phase separation with sharp interfaces.

- Hence, there appears to be a phase transition:
  → high temperature : single homogeneous phase.
  → low temperature : two coexisting phases : one with a **high** particle density (liquid), and one with a **low** density (gas).
- Despite its simplicity, the lattice gas features a **liquid-gas transition**.
  → just as real fluids do!
- Could we also generate more quantitative results with our algorithm, other than to say that some transition takes place at some temperature?
  → Yes, although the (NVT)-ensemble is not the most optimal.

Simulation data for 10×10 lattice using $\rho = 1/2$.

- **Result 1** : $\langle E \rangle / V$ versus inverse temperature β:

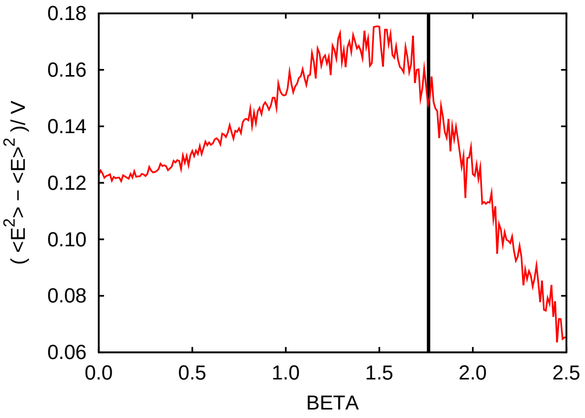


$\rightarrow$ The average energy **decreases monotonically** with β.
$\rightarrow$ The **slope** attains a maximum.
$\rightarrow$ Note that the slope corresponds to the **specific heat**!

- **Result 2** : measure the specific heat:

$$C = \frac{\langle E^2 \rangle - \langle E \rangle^2}{V},$$

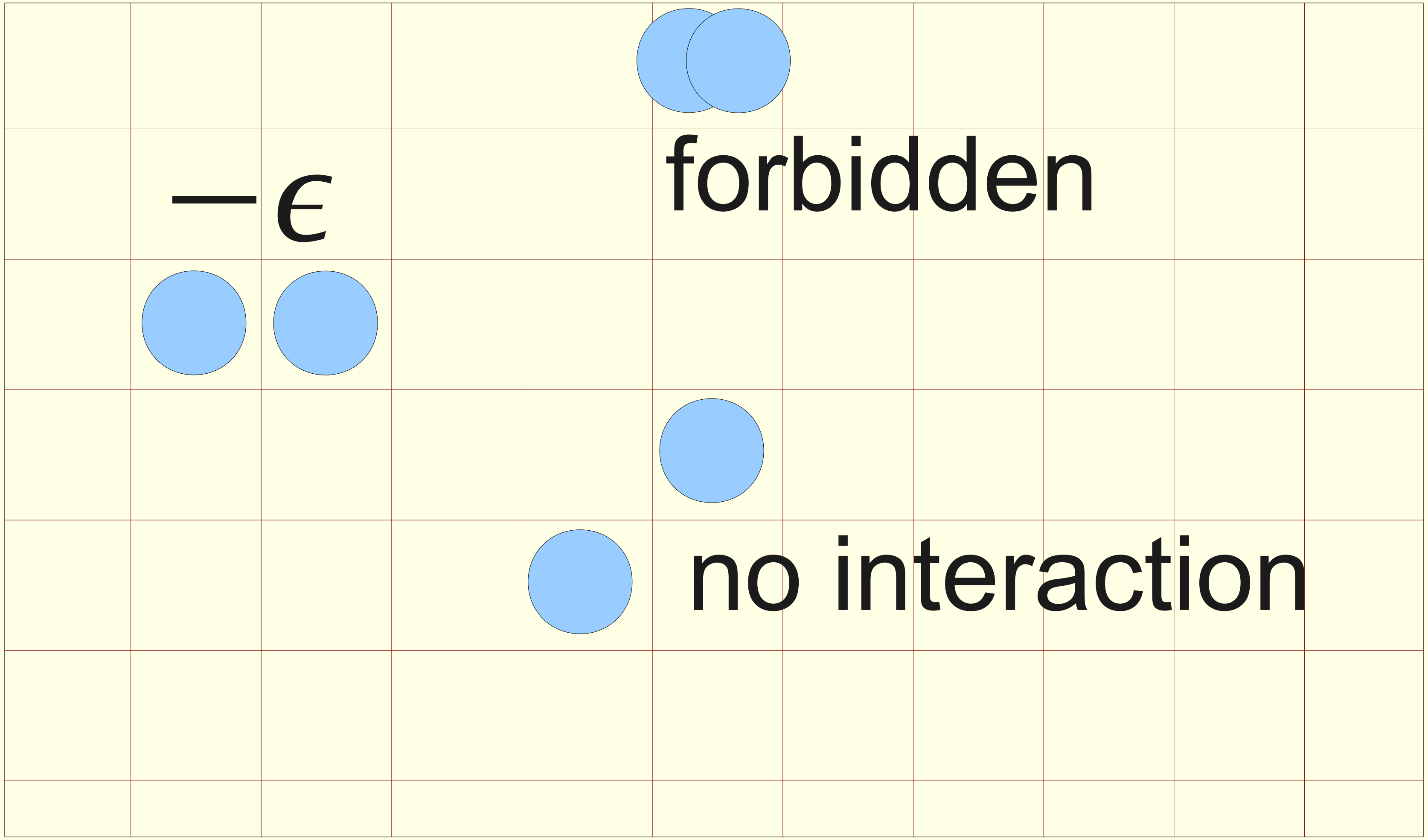
versus inverse temperature β:



- The specific heat **peaks** at a value close to the known exact result $\beta_{\text{exact}} = 2 \ln(1 + \sqrt{2}) \approx 1.763$ (vertical line).
$\rightarrow$ Agreement is **not** perfect though : **why?**

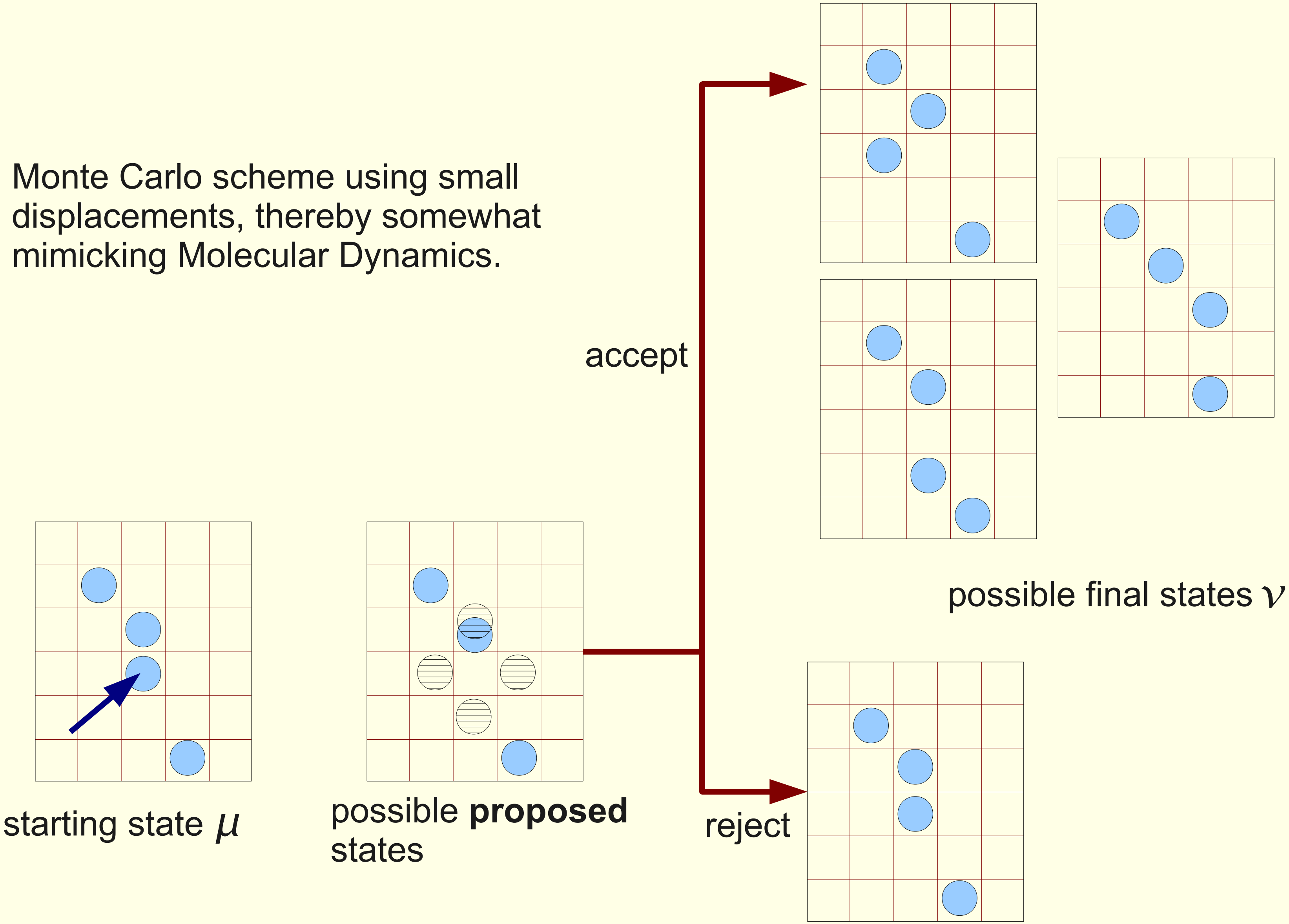
  (answer: finite size effects)

The 2D lattice gas



MC algorithm I : small steps

Monte Carlo scheme using small displacements, thereby somewhat mimicking Molecular Dynamics.



accept

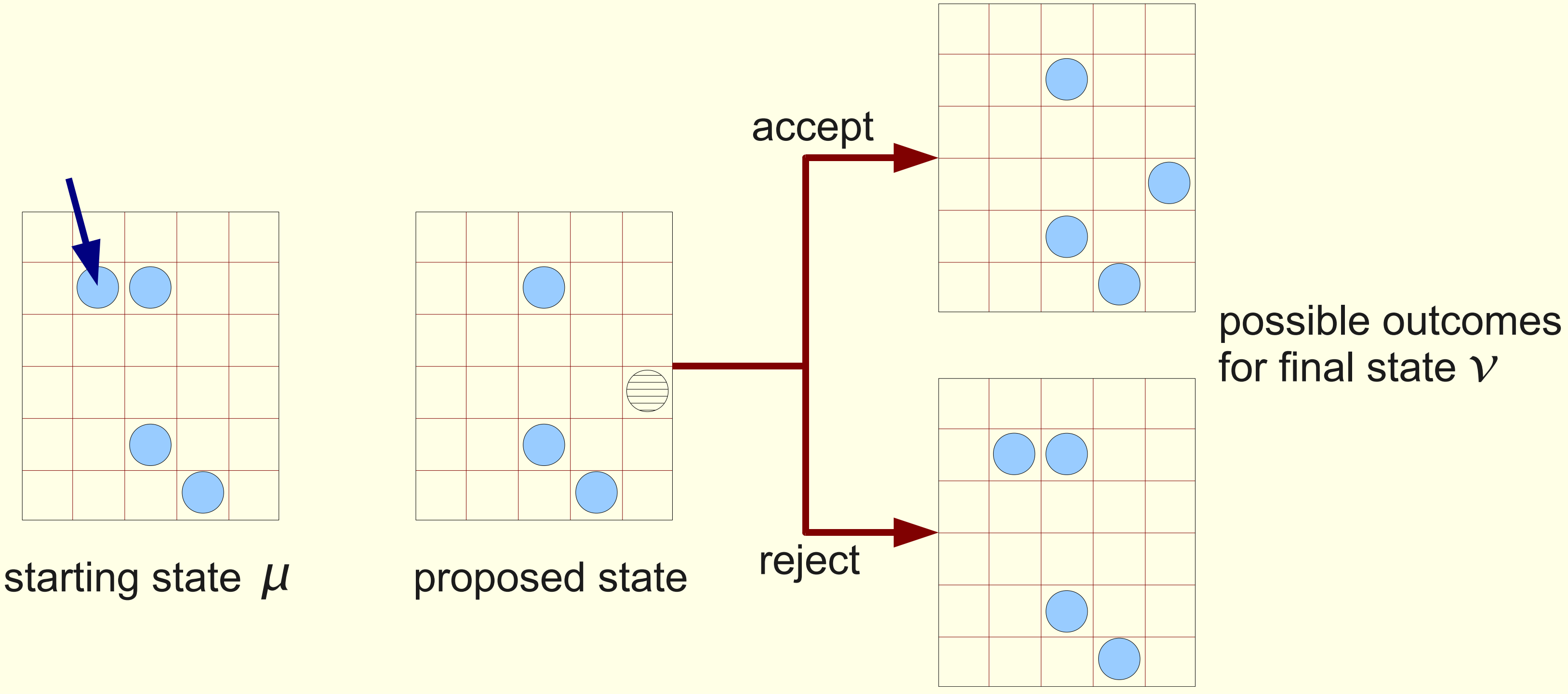
possible final states ν

starting state μ

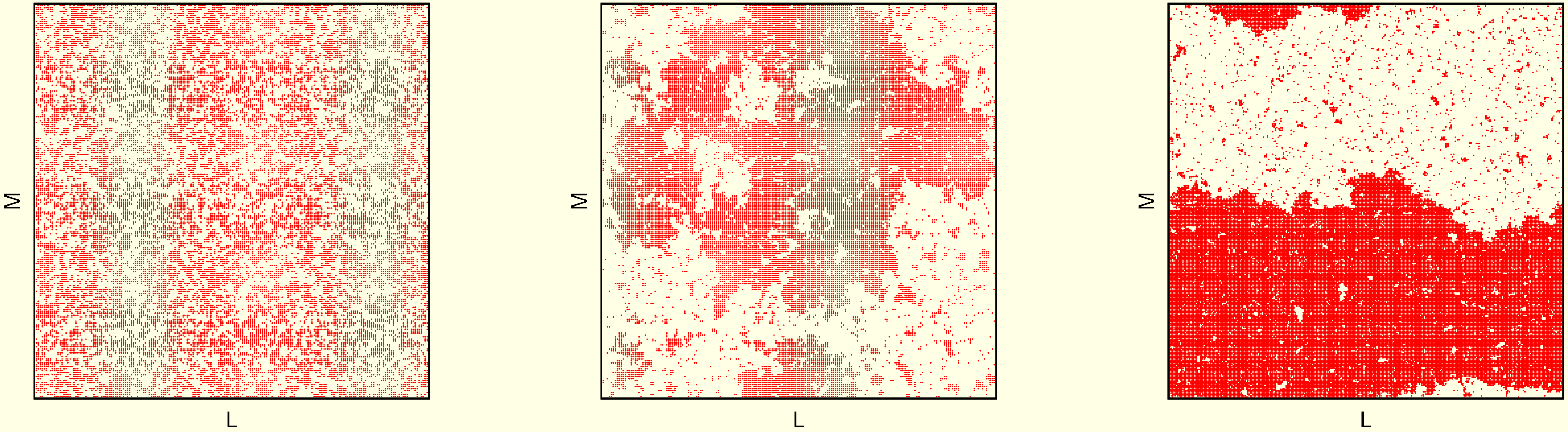
possible **proposed** states

reject

MC algorithm II : large steps

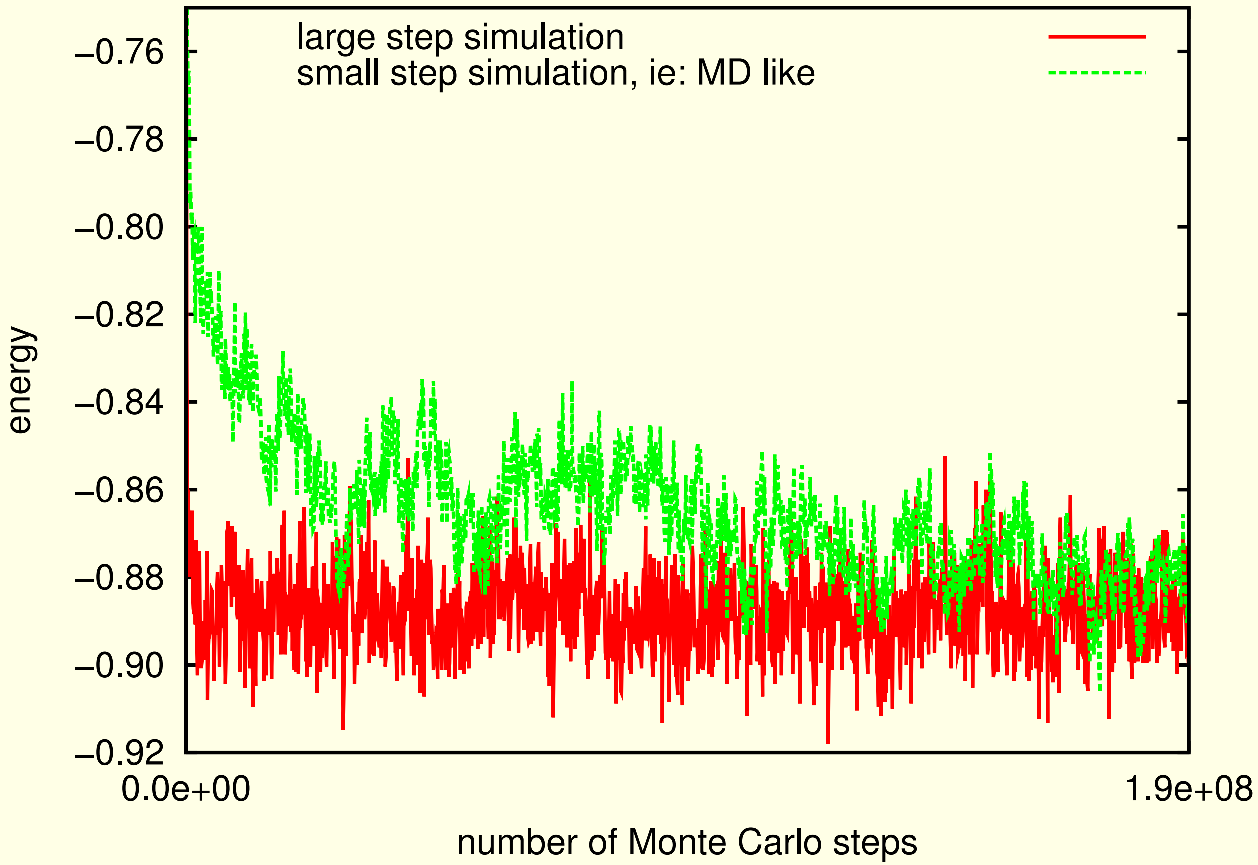


2D lattice gas : qualitative behavior



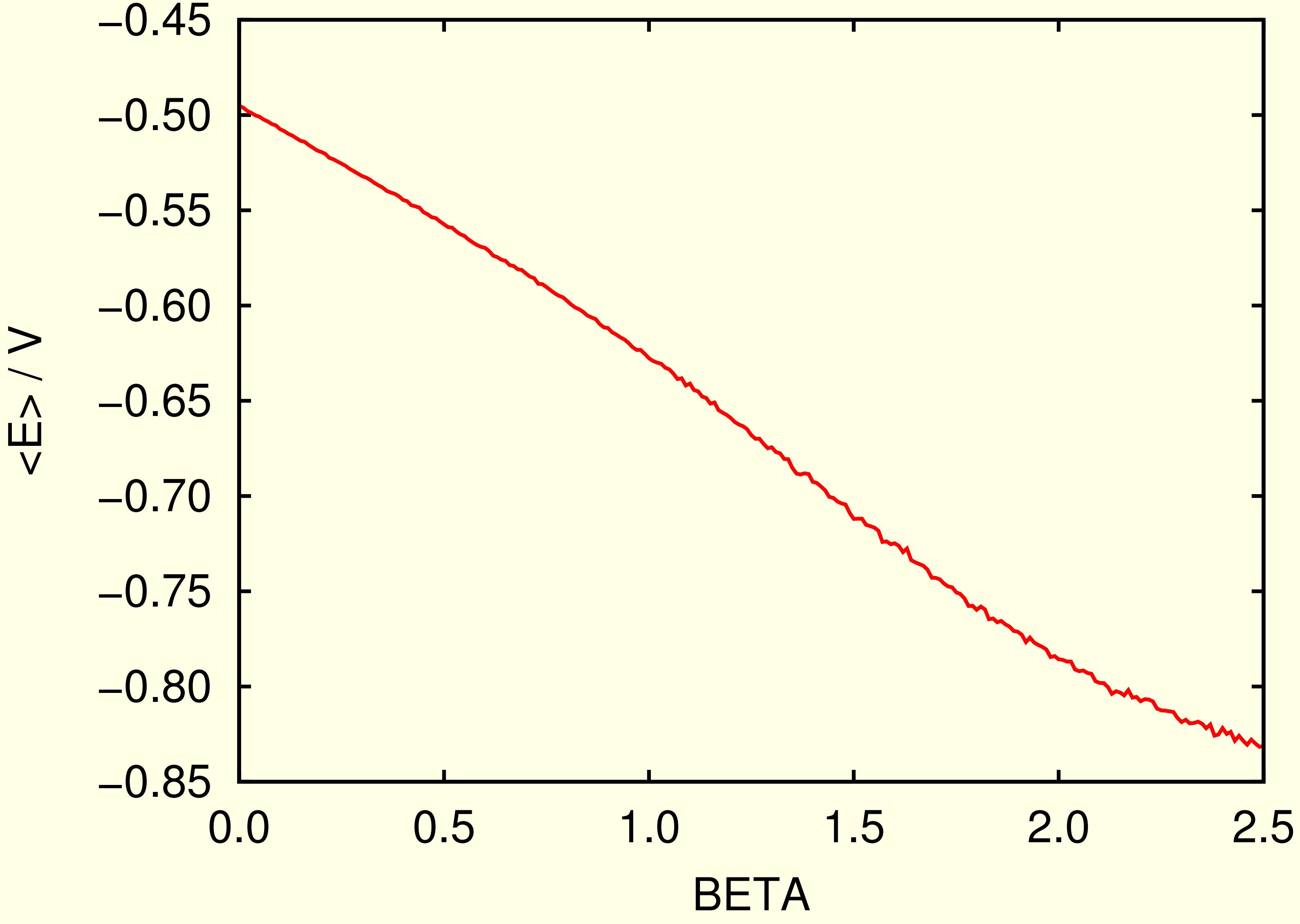
(left to right: decreasing temperature)

Comparing the two algorithms



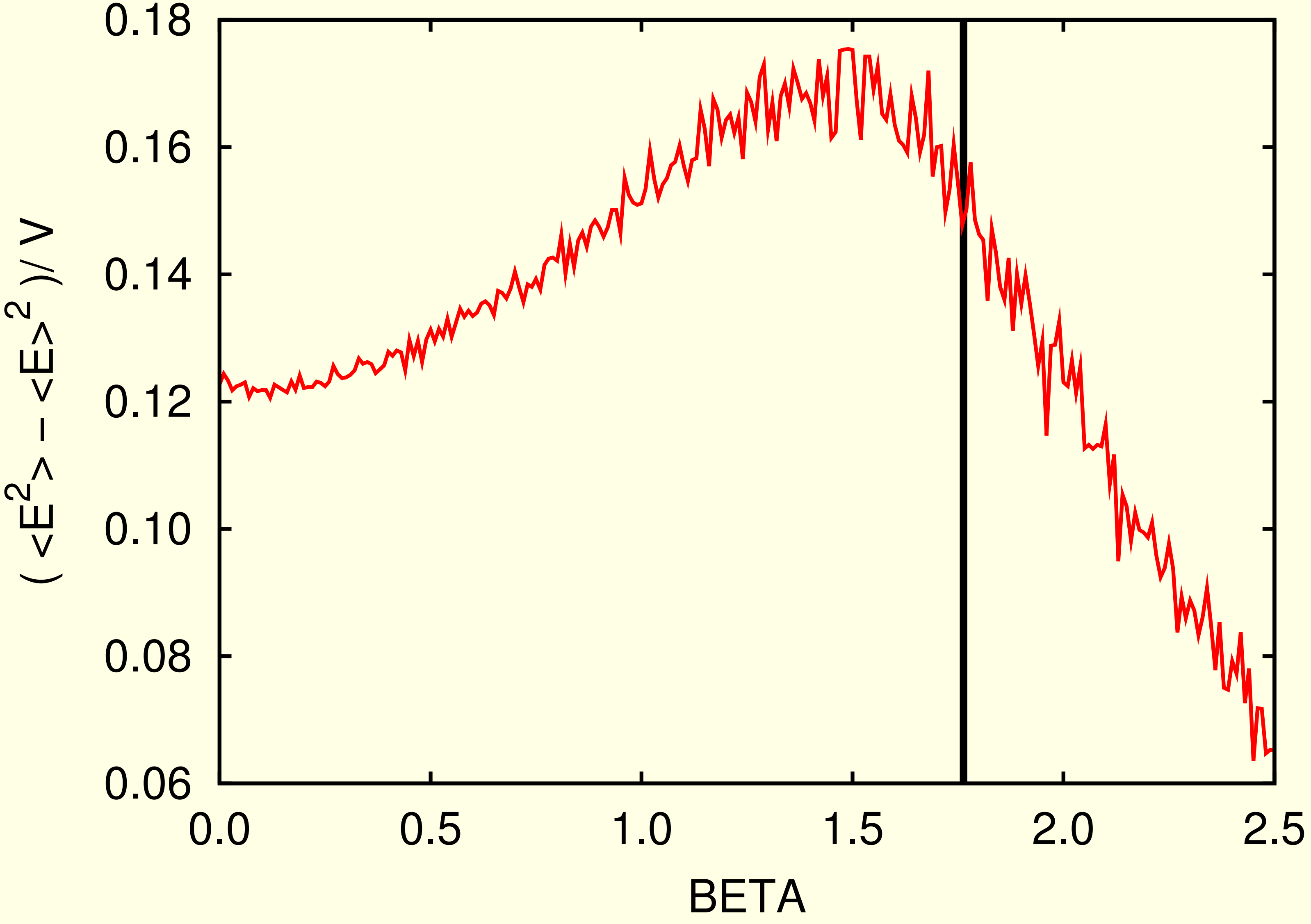
$\beta\epsilon = 1.9,\ \rho = 1/2,\ L = 50$

Average energy vs. temperature



$L = 10, \; \rho = 1/2$

Specific heat vs. temperature



$L = 10, \ \rho = 1/2$

1 Summary of previous lecture

- Introduced the 2D **lattice gas** model.
- Presented two Metropolis Monte Carlo algorithms to simulate the lattice gas.
    1. “Molecular Dynamics” variant : small displacements of particles.
    2. Large-step variant : displacements of particles of arbitrary size.
- Both algorithms were shown to be ergodic and to obey detailed balance (and therefore yield identical results), but the **large-step variant** was more **efficient** in the sense that it **equilibrated** much faster.
- **Result** : the lattice gas undergoes **phase separation**, into a liquid and a gas phase, when the temperature T becomes low. Below the transition temperature, our simple Monte Carlo program nicely shows phase coexistence and interface formation.

OUTLINE FOR TODAY

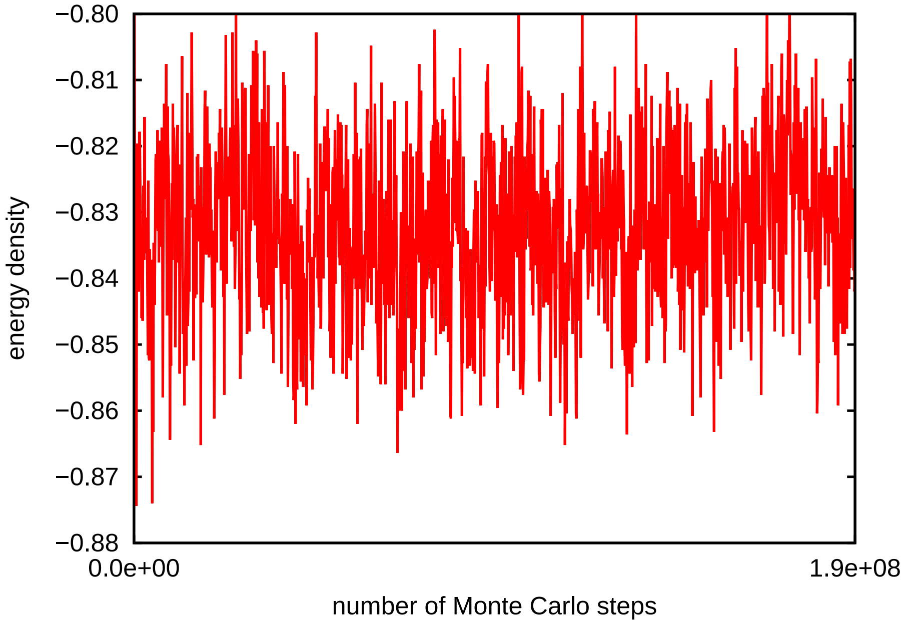
1. **Histogram reweighting** : the idea is to perform a single simulation at one value of the temperature T, and to use the information obtained from that simulation to also obtain results for different values of T *without doing additional simulations*! This clearly should increase our efficiency.
2. **Literature**: Chapter 8 of Newman & Barkema book.

2 Fluctuations

- Consider a (NVT)-ensemble Monte Carlo simulation of the lattice gas at some inverse temperature $\beta = 1/T$.
  $\rightarrow$ using, for example, one of last weeks algorithms.

- After **equilibration**, the probability of state μ occurring equals $P_\mu \propto e^{-\beta E_\mu}$, with E_μ the energy of the state.

- The **energy** thus **fluctuates** during the course of the simulation, as can be seen from a time series:



Graph of the energy of the observed states during a MC simulation of the 2D lattice gas on a $L \times L$ lattice, $L = 50$, with periodic boundaries, overall particle density $\rho = 0.5$, and $\beta = 1.763$, which is close to the phase transition temperature.

- Since the energy E in the (NVT)-ensemble fluctuates, it might be a nice idea to record the distribution $P(E)$, defined as the probability to observe the energy E (also saves disk space!).

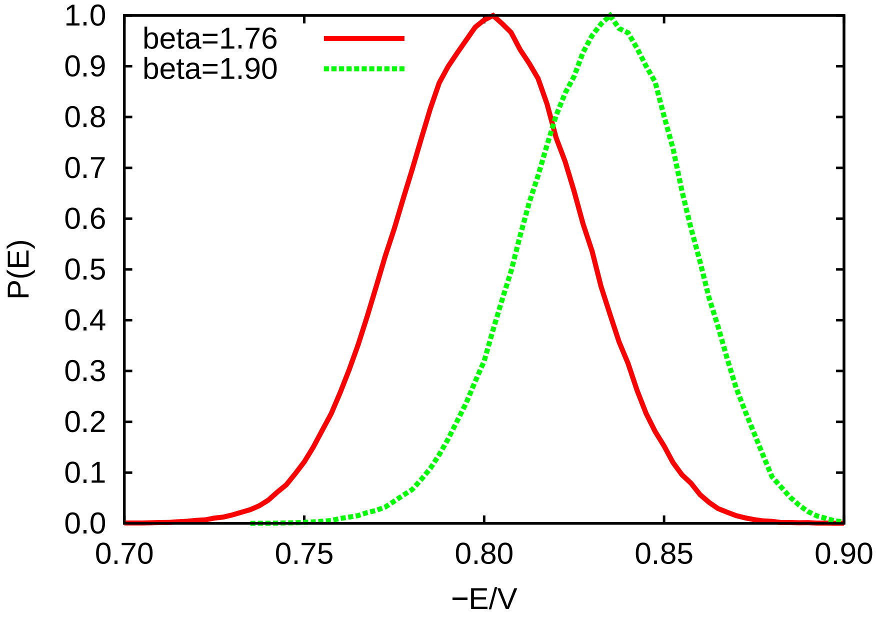
- Such a distribution is trivially measured in a MC simulation : simply update a histogram of the observed energy after each MC move. See the course website to download an example program to achieve this.

- Knowledge of the distribution $P(E)$ allows you to calculate any moment of the energy that you could possibly want:

$$\langle E^\alpha \rangle = \frac{\int E^\alpha P(E)\,\mathrm{d}E}{\int P(E)\,\mathrm{d}E}. \qquad (1)$$

For example the specific heat: $C = (\langle E^2 \rangle - \langle E \rangle^2)/V$, with V the volume of the system (or area if in 2D).
→ For the **lattice gas**, the integrals become sums, of course.

- What does $P(E)$ look like? Below are some **example** distributions for the **2D lattice gas** taken at two different inverse temperatures β, using a square lattice of $L = 20$ and $\rho = 0.5$. You could easily reproduce these graphs using the program on the website.



- What do we see?

  1. For the lattice gas $P(E)$ is just one **single peak**, roughly **Gaussian**. This need not always be the case: other systems might have a double peaked energy distribution!
  2. $P(E)$ depends on β : with increasing β, the peak position (i.e. the average energy) shifts to **lower** values, as expected (note that the energy scale is negative).
  3. For the two values of β considered here, the distributions strongly **overlap**.

- To obtain the above graph, **two independent simulations** were performed, one for each value of β.

- **Question** : since the distributions look so similar, would it be possible to perform just **one** simulation at one β, and to **extrapolate** to different β? This would obviously safe a lot of time!
→ Yes, this is often possible!

- Recall from Statistical Mechanics the expression for the **ensemble average** of the energy moments:
$$\langle E^\alpha \rangle = \frac{\sum_\mu E_\mu^\alpha \exp\left(-\beta E_\mu\right)}{\sum_\mu \exp\left(-\beta E_\mu\right)},$$
where the **sum** is **over all states** μ that the system can possibly be in.

- The sum over states μ can also be written as an integral over energy, but then we must be compensate for **degeneracy**!

- **Degeneracy** : by this is meant that it is possible for **different** states to have the **same energy**. For example, the Ising model has two ground states, both of which have the same energy.

- For this reason, one introduces the **density of states** $g(E)$ (DOS) which "counts" how many states of energy E there are in a small interval $\mathrm{d}E$ around E.

- **Important** : the DOS does not depend on temperature!

- The **sum over states** μ may thus be converted into an **integral over energy** provided one uses the **DOS** to give "extra weight" to those energy values which appear often:
$$\langle E^\alpha \rangle = \frac{\int_E E^\alpha \, g(E) \, \exp\left(-\beta E\right) \, \mathrm{d}E}{\int_E g(E) \, \exp\left(-\beta E\right) \, \mathrm{d}E}. \qquad (2)$$

- Note that the only **temperature dependence** is contained in the **Boltzmann factor**. Hence, if we knew $g(E)$, we could measure **any** energy observable of interest, at **any** temperature!

- If only we knew $g(E)$!

- But, this is **easy** : simply compare the **structure** of Eq.(2), to the "Monte Carlo formula" Eq.(1), and it follows that:
$$P(E) \propto g(E) e^{-\beta E}.$$

- Hence, the DOS is in principle known after doing just a single simulation, which leads to the idea of **histogram reweighting**.

3 Histogram reweighting

- The idea is to perform one MC simulation at inverse temperature β_0 : the output is the energy distribution $P(E)$.

- This yields the following estimate of the density of states:

$$g(E) \propto P(E) e^{\beta_0 E}. \tag{3}$$

- **Next**, we **substitute** the above expression for $g(E)$ into Eq.(2) which yields:

$$\langle E^\alpha \rangle = \frac{\int_E E^\alpha \, P(E) \, e^{-(\beta - \beta_0)E} \, \mathrm{d}E}{\int_E P(E) \, e^{-(\beta - \beta_0)E} \, \mathrm{d}E},$$

  which is a very simple expression indeed.
  → Note that the proportionality constant in Eq.(3) cancels!
  → **Remember** : $P(E)$ is the energy distribution obtained in the simulation performed at inverse temperature β_0.

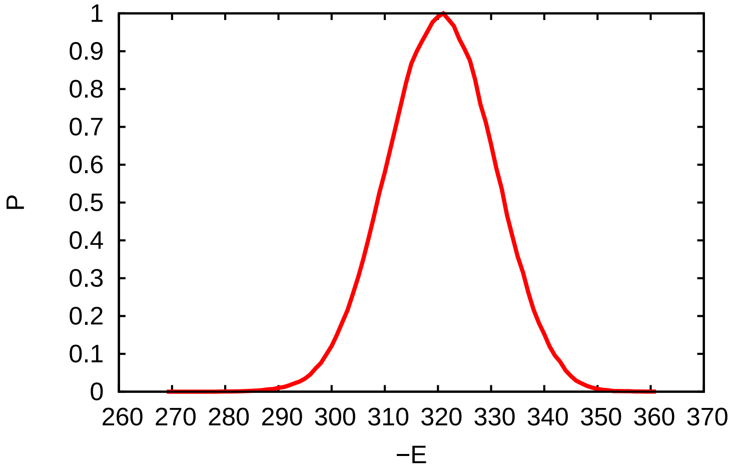
- The above equation immediately implies that the energy distribution at inverse temperature β can be obtained from the distribution measured at β_0 via

$$P_\beta(E) \propto P(E) e^{-(\beta - \beta_0)E}, \tag{4}$$

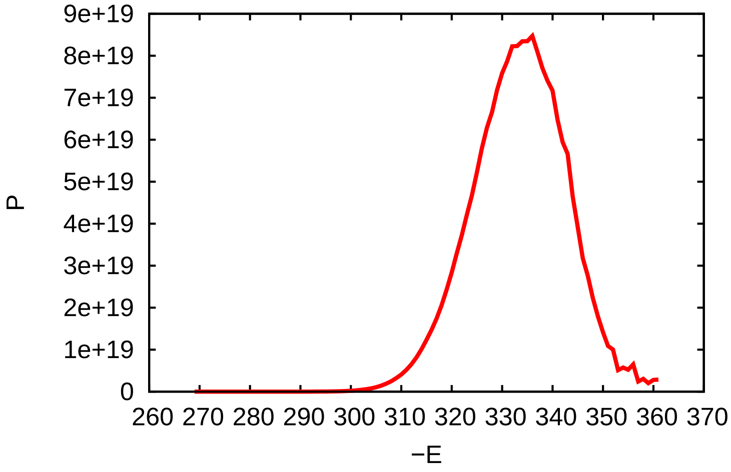
  with $P_\beta(E)$ the energy distribution at inverse temperature β, and $P(E)$ the one measured at β_0.

- **Demonstration of histogram reweighting** : we take the distribution obtained at $\beta_0 = 1.76$, and extrapolate it to $\beta = 1.90$, ie: $\beta - \beta_0 = 0.14$.

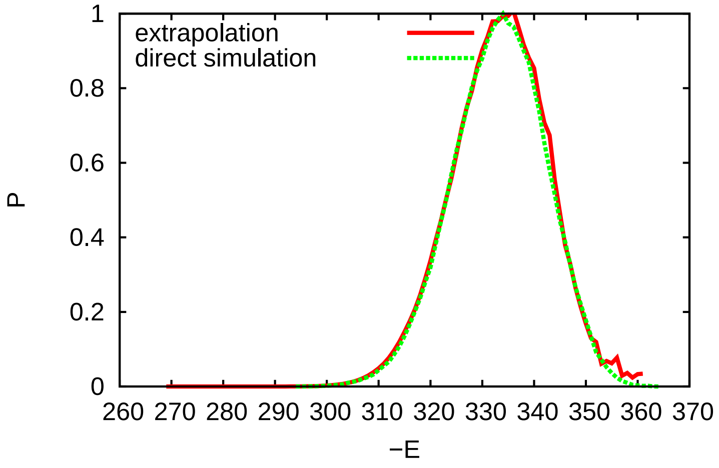
energy distribution at beta=1.76



extrapolated distribution : P(E) * e$^{0.14\ E}$



extrapolated distribution normalized and compared to direct simulation result



- Agreement is clearly excellent!
- **But**, watch out for common pitfalls:

  1. Beware to extrapolate on the **energy** E, and **not** on the **energy density** E/V.
  2. Since E is extensive, multiplying by e^E might become numerically unstable! The previous example already hints at this because the numbers on the vertical axes of the middle graph are rather large.

- **In fact**, this numerical instability commonly occurs, especially in large systems, and so a proper “fix” is needed, if histogram reweighting is to remain practical.
- **Idea** : do not perform the extrapolation on the probability $P(E)$ itself, but rather on its (natural) logarithm. That is, replace:

  $$P_\beta(E) = \frac{1}{C} P(E) e^{-(\beta-\beta_0)E},$$

  by

  $$\ln P_\beta(E) = \ln P(E) - (\beta - \beta_0)E - \ln(C),$$

  where C is the normalization constant.

- **In words**: by using the logarithm, the extrapolation formula Eq.(4) corresponds to the addition of a straight line $f(E) = aE + b$, with slope $a = -(\beta - \beta_0)$ and intercept $b = -\ln(C)$.
- The normalization constant C is given by the integral:

  $$C = \int P(E) e^{-(\beta-\beta_0)E} \, \mathrm{d}E.$$

- On a computer, the above integral becomes a sum:

  $$C = P(E_1)e^{-(\beta-\beta_0)E_1} + P(E_2)e^{-(\beta-\beta_0)E_2} + P(E_3)e^{-(\beta-\beta_0)E_3} + \ldots + P(E_k)e^{-(\beta-\beta_0)E_k},$$

  where $E_1, \ldots, E_k$ denotes the energy range that was visited (sampled) by the simulation. **In words**: the normalization constant C is a sum of many numbers, and these numbers can get very large. So large, in fact, that overflows can easily occur.

- However, we don’t need to know C itself, we only need to know $\ln(C)$.
- **Problem** : given only the logarithms $a = \ln A$ and $b = \ln B$, of two extremely large numbers A and B, how does one compute $\ln(A + B)$?

  1. Assume that $A > B$, which implies that also $a > b$.
  2. Next write: $\ln(A + B) = \ln(e^a + e^b) = \ln\left(e^a(1 + e^{b-a})\right) = a + \ln(1 + e^{b-a})$.
  3. Since $a > b$, it holds that $0 < e^{b-a} \leq 1$ so no danger of overflow.

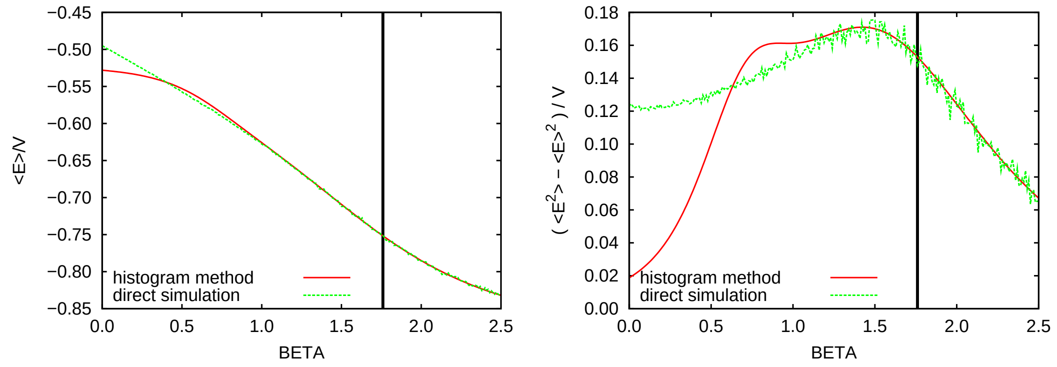
4. Note: if $a \gg b$, the term $e^{b-a} \to 0$, but since we only require $\ln(1+e^{b-a})$ there is no danger of underflow either.
5. Note that many programming languages provide a library function `log1p(x) = ln(1+x)`, so use that when available!
6. What if $B > A \to$ swap them around.
7. **Literature**: see Section 8.1.1 of Newman & Barkema.

4 Application to the lattice gas

- Recall that last week we measured the **average energy** $\langle E \rangle$ and the **specific heat** $C = (\langle E^2 \rangle - \langle E \rangle^2)/V$ of the **2D lattice gas** as a function of β.

- This was done the **direct way**, ie: for each β a separate simulation was carried out.

- Today we **revisit** the problem using **histogram reweighting** : we perform just **one** simulation at $\beta_0 = 1.76$, and extrapolate the results to different β.



System parameters: $L = 10$ and $\rho = 1/2$. The vertical bars indicate $\beta_0 = 1.76$ where the simulation used for the histogram extrapolation was performed.

- Some observations:

  1. Around β_0 agreement is clearly very good! And we can extrapolate quite far in both directions, especially to high β.
  2. Agreement deteriorates at low β, in particular regarding the specific heat.
  3. Histogram extrapolation gives very smooth curves, also in regions where the disagreement is actually pronounced! It may even give rise to artificial features in the data. These are potential pitfalls of the method that you must be aware of.

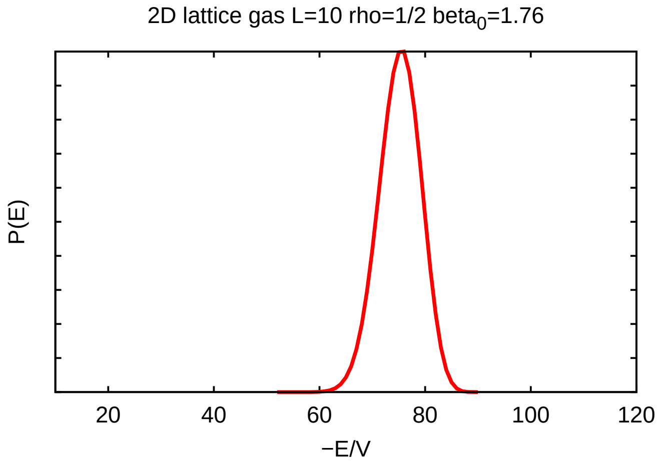
- See course website for example code!

5 Limitations of histogram reweighting

- The simulation done at β_0 does not sample all energies. It typically spends most time around the average energy $\langle E\rangle_0$, with deviations from the average being of order:

$$\sigma_0 = \sqrt{\langle E^2\rangle - \langle E\rangle^2},$$

see the example below: only energies in the range $50 < E < 90$ or so were visited (sampled).



- Consequently, we do not know the density of states $g(E)$ for all energies, only for those energies that were actually sampled.
- In addition, energies in the “tails” of the distribution, ie, those energies **not** sampled very often, will probably give a rather poor estimate of $g(E)$.
- Histogram reweighting is no longer reliable when the **average energy** $\langle E\rangle_\beta$ of the **extrapolated** distribution is inside the “tail” of the measured distribution.
- Hence, we propose the following **safety criterion** for histogram reweighting:

$$|\langle E\rangle_\beta - \langle E\rangle_0| < \sigma_0. \qquad (5)$$

$\rightarrow$ **In words**: provided $\langle E\rangle_\beta$ remains within one standard deviation of $\langle E\rangle_0$, histogram extrapolation should be safe.

- Note that σ_0 is related to the specific heat at inverse temperature $\beta_0 \rightarrow C_0 = \sigma_0^2/V$, with V the volume of the system.
- We can work this out some more. To this end, write the following Taylor expansion:

$$\langle E\rangle_\beta = \langle E\rangle_0 + (\beta - \beta_0)\left.\frac{\partial\langle E\rangle_0}{\partial\beta'}\right|_{\beta'=\beta_0} = \langle E\rangle_0 + (\beta-\beta_0)\sigma_0^2,$$

where I used the well-known result from Statistical Mechanics that the β-derivative of $\langle E\rangle$ equals σ_0^2.

- Hence, Eq.(5) takes the form:

  $$\sigma_0^2 \Delta\beta < \sigma_0 \to \Delta\beta \sim \frac{1}{\sqrt{VC_0}},$$

  with $\Delta\beta = |\beta - \beta_0|$.

- Away from any phase transition C_0 is constant: $\Delta\beta \sim 1/\sqrt{V}$. The temperature range over which you can safely extrapolate decreases with the square-root of the system volume.

- All in all, the performance of histogram reweighting becomes progressively worse in larger systems.

- The problem of Metropolis importance sampling is that we generate the same states as nature would do. That is, each state μ appears with the Boltzmann probability $P_\mu \propto e^{-\beta E_\mu}$. This restricts the range of energy that the simulation samples. Ironically, we worked so hard to get an algorithm with importance sampling during the last lectures, only to find out this lecture this has limitations also.

- Possible fixes:

  1. Do simulations for a series of closely-spaced inverse temperatures β_i, such that the corresponding distributions $P_i(E)$ overlap. Then combine the histograms sensibly → **multiple histogram method** (which is substantially more complex to implement than the single histogram method discussed today, see Section 8.2 of Newman & Barkema).
  2. Perhaps give-up Metropolis Boltzmann sampling altogether, and **artificially force** the simulation to visit all energies of interest → **biased simulations**. The "bias" must be "controlled" such that it can be correctly removed at the end of the simulation. This will be discussed in later lectures.

1 Summary of previous lecture

1. Introduced the density of states $g(E)$ (DOS), defined as the number of states with energy E. For a continuous energy, one would say that $g(E)$ counts the number of states whose energy is inside an (infinitesimally small) interval from E to $E + dE$.

2. Note that the DOS does NOT depend on temperature!

3. The DOS is extremely powerful because once it is known, it allows you to calculate any energy observable of interest at any inverse temperature $\beta = 1/k_B T$:

$$\langle E^\alpha \rangle = \frac{\int E^\alpha \, g(E) \, e^{-\beta E} \, dE}{\int g(E) \, e^{-\beta E} \, dE}. \qquad (1)$$

   $\rightarrow$ Again, note that the β-dependence is entirely contained in the Boltzmann factor.
   $\rightarrow$ Ensemble average does not depend on any proportionality constant in the DOS.

4. In a standard Metropolis MC algorithm, where a state μ appears with a probability $p_\mu \propto e^{-\beta E_\mu}$, the energy histogram $P(E)$ is related to the DOS via:

$$g(E) \propto P(E) e^{\beta E}. \qquad (2)$$

   $\rightarrow$ $P(E)$ is defined as the probability to observe energy E during the simulation.
   $\rightarrow$ **Note** : $P(E)$ depends on temperature, but $g(E)$ does not!
   $\rightarrow$ **Special case** $\beta = 0$: $P(E) \propto g(E)$.

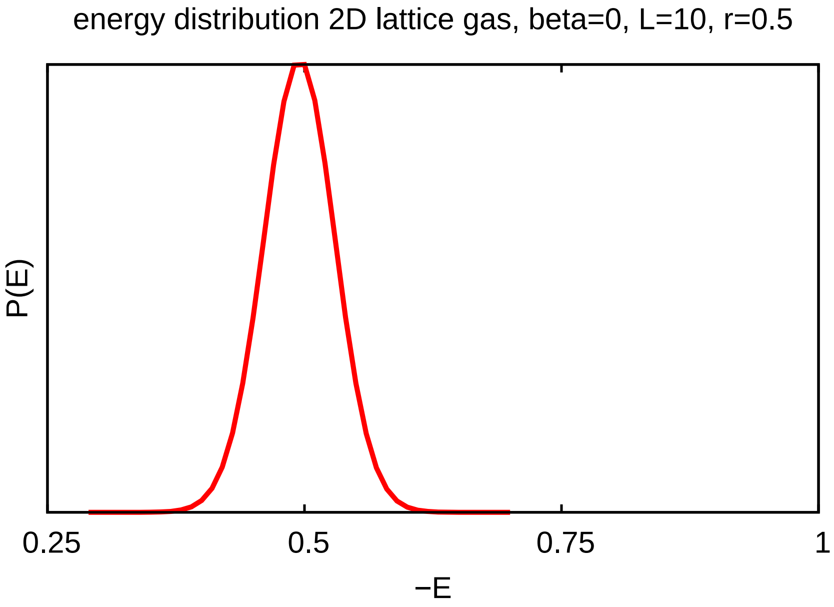
5. We can use Eq.(2) to convert the measured $P(E)$ into the DOS, and then use Eq.(1) to obtain averages as a function of temperature, without having to do simulations at those temperatures.
   $\rightarrow$ histogram extrapolation or reweighting.

6. **But**, there was a **catch**! Since the energy in a standard simulation fluctuates over only a limited range, also the range over which the DOS gets sampled is restricted.
   $\rightarrow$ **Problem** : the range decreases with system size!

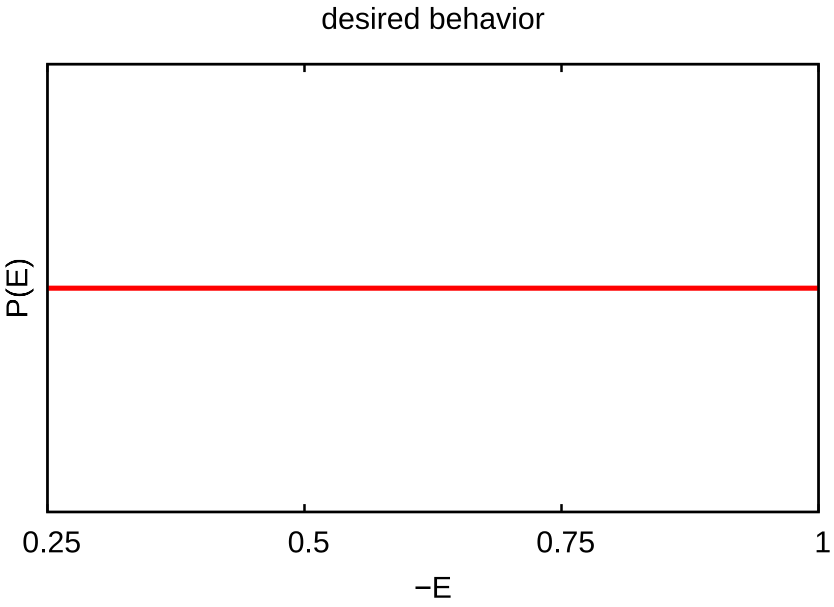
7. Hence, the temperature range over which one can reliably extrapolate is finite, and actually quite small in practice.

8. **Today** : How to extend the energy range that a simulation samples such that $g(E)$ is known over a larger interval. Ideally, we seek an algorithm that visits all energies over some desired range equally often!
   $\rightarrow$ the Wang-Landau algorithm.

1. This is the standard behavior of the energy in a Metropolis MC algorithm where states appear in proportion to their Boltzmann weight.



2. And this is the desired behavior we would like to have:



2 How to extend the energy range

1. Consider a completely hypothetical system which can be in 12 different states

$$\mu = 1, \ldots, 12.$$

2. To help you visualize this, you may think of the system being a vector which can point in 12 directions, i.e. the hour dial of a clock.

3. We assume that each state μ has its own energy given by $E_\mu = \mu$.
$\rightarrow$ so $E_1 = 1, E_2 = 2, \ldots$ etc.

4. What does the DOS $g(E)$ for this system look like?
$\rightarrow$ Recall that $g(E)$ counts the number of states with energy E.
$\rightarrow$ Since each state has its own unique energy, there is no degeneracy, and so $G(E) = 1$.

5. Ensemble averages for this model are trivial:

$$\langle E \rangle = \frac{\sum_{E=1}^{12} E g(E) e^{-\beta E}}{\sum_{E=1}^{12} g(E) e^{-\beta E}}.$$

$\rightarrow$ There is no immediate need for a MC approach to this problem.

6. Nevertheless, assume performing a standard MC importance sampling simulation of this system with $p_\mu \propto e^{-\beta E_\mu}$.
$\rightarrow$ That is, we introduce transitions between the states, via some hypothetical MC move.
$\rightarrow$ To help you visualize things, assume a MC move consisting of moving the “dial” up or down with 50% probability. This is clearly ergodic.
$\rightarrow$ Accept the moves with:

$$P_{\text{acc}}(\mu \rightarrow \nu) = \min\left[1, e^{-\beta(E_\nu - E_\mu)}\right].$$

7. What would the energy distribution $P(E)$ look like in such a simulation as a function of β?

8. Low temperature limit $\beta \rightarrow \infty$: all time is spend in the ground-state $E = E_1$, and hence $P(E)$ has just one peak. This is bad from the point of view of histogram reweighting (draw $P(E)$ on blackboard).

9. High temperature limit $\beta \rightarrow 0$: all moves are now accepted, and since our “clock algorithm” visits all states with equal probability, and since each state has its own energy, we obtain a random walk in energy. Hence $P(E)$ is a flat. This is clearly excellent for histogram reweighting (draw $P(E)$ on blackboard).

10. Hence, to get optimal performance, we must run our simulation at infinite temperature $\beta = 0$. In this limit one also has:

$$P(E) \propto g(E).$$

11. **Unfortunately**, this behavior is not true in general. It only works here because the DOS in our model is constant. In general, the DOS is not constant, and life would be very boring if it were (no phase transitions for example).

12. Hence, we reconsider our clock model at $\beta = 0$, but this time we make it slightly more realistic by including degeneracy. We still assume 12 states $\mu = 1, \ldots, 12$, but now the energies are chosen such that:

$$\begin{cases} E = 1 & \mu = 1, 2 \\ E = 2 & \text{otherwise} \end{cases}$$

The DOS thus consists of two peaks: one at $E = 1$ with height 2, and one at $E = 2$ with height 10 (draw on blackboard).

13. Simulating at $\beta = 0$ where $P(E) \propto g(E)$, also the energy distribution $P(E)$ that we measure will have two peaks. The peak at $E = 1$ will be 5 times lower than the peak at $E = 2$, which is bad for histogram reweighting.

14. Hence, we do not obtain a flat histogram. The state with $E = 2$ appears 5 times "too often". Or, more generally, a state with energy E appears $P(E)$ times too often.

15. One could formally compensate for this in a different (ideal) algorithm which accepts states μ with probability $p_\mu \propto 1/P(E_\mu)$. This way, states that were observed "too often" in the original algorithm, become suppressed.

16. At $\beta = 0$, we have $P(E) \propto g(E)$, and so the ideal algorithm is one in which states are accepted according to:

$$P_{\text{acc}}(\mu \to \nu) = \min\left[1, g(E_\mu)/g(E_\nu)\right].$$

17. This algorithm would yield a flat energy distribution.

18. There is a **catch** : you don't know the DOS $g(E)$ beforehand, so implementing the above ideal algorithm will be difficult. But we ignore this point for the moment.

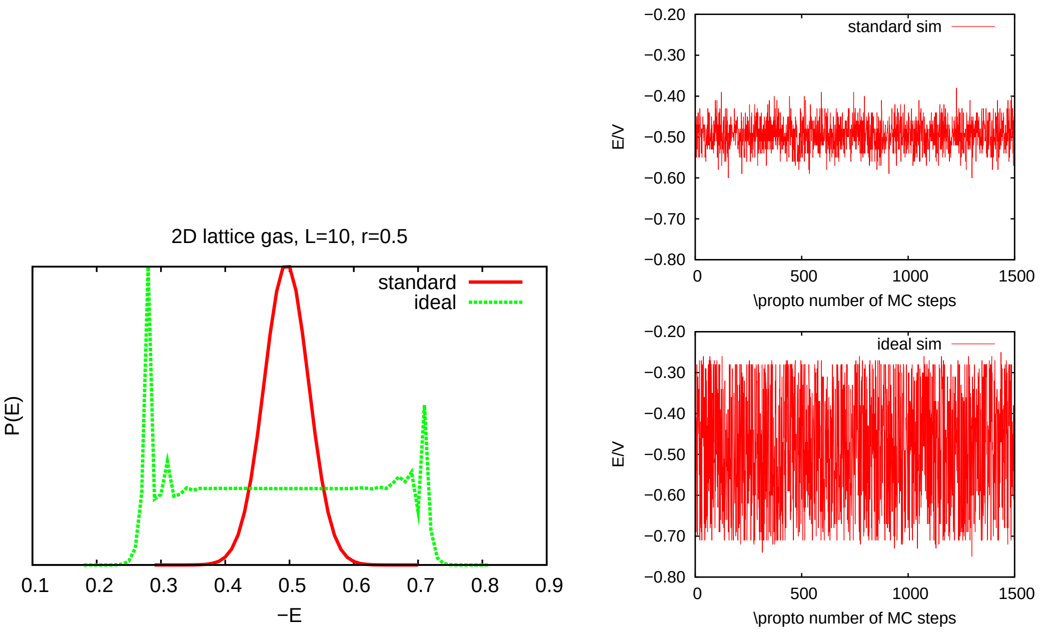
3 A simple test

1. Let's first see if the idea works at all.

2. Perform a simulation of the 2D lattice gas at $\beta = 0$. Record the energy distribution $P(E)$, which is proportional to the DOS $g(E)$ in this case because $\beta = 0$.

3. Now use the estimated $g(E)$ to perform the **ideal** simulation, accepting moves conform:
$$P_{\text{acc}}(\mu \to \nu) = \min \left[1, g(E_\mu)/g(E_\nu)\right].$$

4. This is what we get:



5. Over the range of energy visited by the "original" simulation, the ideal simulation clearly yields a flat distribution. So the idea seems to work!

6. The "timeseries" of the ideal simulation is indeed compatible with a random walk in energy, and covers a much wider range in energy in far less time!

7. Of course, in the "tail regions" things break down.

8. Hence, we have **formally** solved the problem to obtain a flat energy distribution.

9. But, it is not practical, as it requires prior knowledge of $g(E)$. In addition, we still need to extend the energy range over which the simulation samples.

4 A practical solution : Wang-Landau sampling

- **Problem** : to generate states μ whose energies E_μ are inside a specified range $E_{\min} < E_\mu < E_{\max}$, and with the probability of state μ occurring:

$$p_\mu \propto 1/g(E_\mu). \tag{3}$$

- The WL algorithm provides a solution to this problem. It initially assumes that the DOS $g(E) = 1$, and then iteratively improves the result.

The WL algorithm

step 0 : put $g(E) = 1$ and prepare an initial state μ whose energy is inside the desired range $E_{\min} < E_\mu < E_{\max}$.

step 1 : Perform MC moves (eg: spin flips, particle displacements) and accept each move with probability:

$$P_{\mathrm{acc}}(\mu \to \nu) = \min\left[1, g(E_\mu)/g(E_\nu)\right],$$

where it is assumed that the proposition probabilities are symmetric. After each MC step, multiply the DOS of the resulting energy E by a factor $f = 2$:

$$g(E) \to f \times g(E).$$

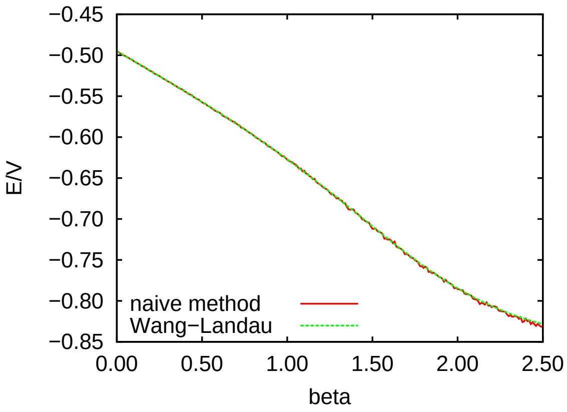
The effect of this modification is to make it less likely to visit energy E again, see Eq.(3), and to instead explore energies that have not been visited yet.
Note 1 : since you are dynamically changing the distribution from which you sample, detailed balance is violated! You will never reach any kind of equilibrium!
Note 2 : in an actual implementation, one updates $\ln g(E)$ to prevent overflow.

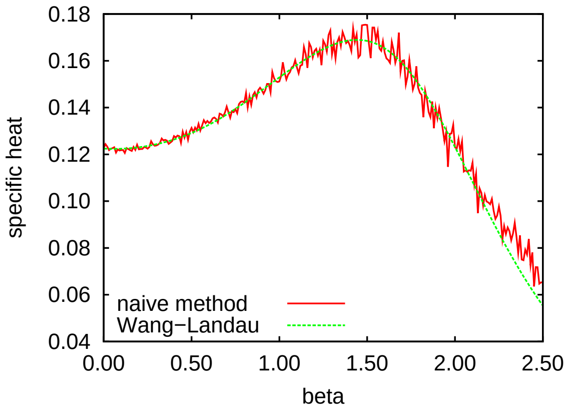
step 2 : Repeat step 1 until all energies over the desired range have been visited “sufficiently often”. One criterion is to require that you have “tunneled” back and forth n times over the interval. This completes one WL iteration.

step 3 : Reduce the factor $f \to \sqrt{f}$ and goto **step 1**. When f becomes very small, changes to the DOS become negligible, detailed balance is recovered, and the algorithm has converged.

5 Application of WL sampling to the 2D lattice gas

1. 2D lattice gas, $L = 10$, $\rho = 1/2$.
2. Use WL sampling to obtain the DOS, with $n = 200$ "tunnelings" per WL iteration.
3. Life demo on laptop!
4. Once converged, use DOS to measure $\langle E \rangle$ and specific heat versus β.





Results are clearly very good!

6 Toward high-resolution simulations of phase transitions

1. With the WL algorithm we have a powerful tool to study phase transitions. We gain direct access to the DOS, from which properties at all temperatures follow.
2. **Next** : application to the liquid-gas transition of the 2D lattice gas.
3. **Till now** : (NVT) simulations only, whereby particles are displaced randomly, but the number of particles is constant.
4. This ensemble is too restrictive $\rightarrow$ no density fluctuations.
5. **Solution** : grand canonical Monte Carlo simulations.
   $\rightarrow$ fluctuating numbers of particles!
   $\rightarrow$ clearly requires a new Monte Carlo move!

7 The GC ensemble

1. (NVT)-ensemble : energy E fluctuates, $\beta = 1/k_BT$ constant.
   $\rightarrow$ Boltzmann probability $\propto e^{-\beta E}$ (note the combination $\beta \times E$).
2. GC-ensemble : in addition to energy fluctuations, number of particles N fluctuates.
   $\rightarrow$ controlled by constant chemical potential κ.
   $\rightarrow$ GC Boltzmann probability $\propto e^{-\beta E+\kappa N}$.
   $\rightarrow$ Note the combination $\kappa \times N$.
3. **Aim** : to design a MC scheme in which particles diffuse, but also where the number of particles N fluctuates. The probability of a state μ occurring should be:
   $$p_\mu \propto e^{-\beta E_\mu + \kappa N_\mu}.$$
4. Anything is allowed, as long as **ergodicity** and **detailed balance** are obeyed!

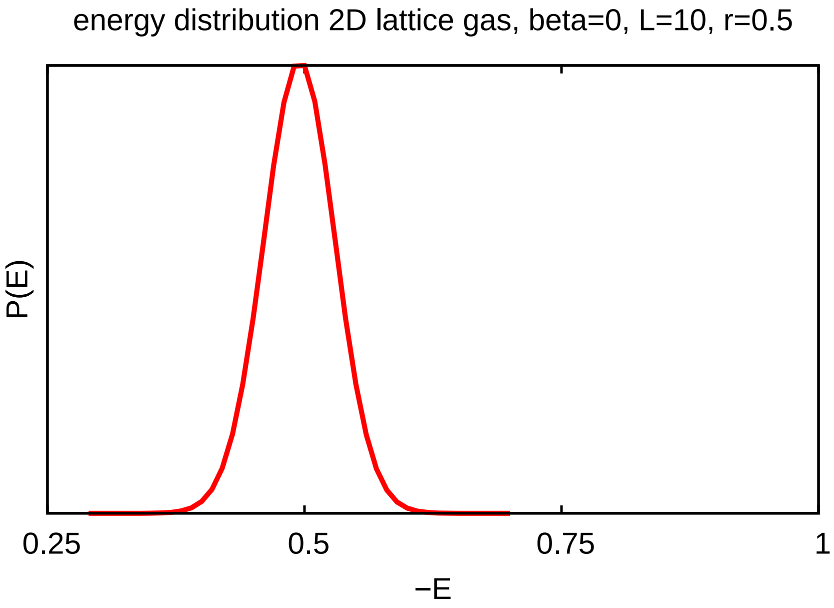
8 A simple GCMC algorithm

1. For the lattice gas, a valid MC move would be:
   (1) choose a random lattice site.
   (2) if the site is empty, put a particle there; if the site is occupied, remove the particle.
2. This move is clearly ergodic; it is also symmetric, and so:
   $$P_{\text{acc}}(\mu \rightarrow \nu) = \min\left[1, e^{-\beta(E_\nu - E_\mu) + \kappa(N_\nu - N_\mu)}\right].$$
3. This algorithm allows for a GC simulation at fixed β and κ. However, it explicitly uses the particle-hole symmetry of the lattice, and so does not generalize to *off-lattice* fluids.

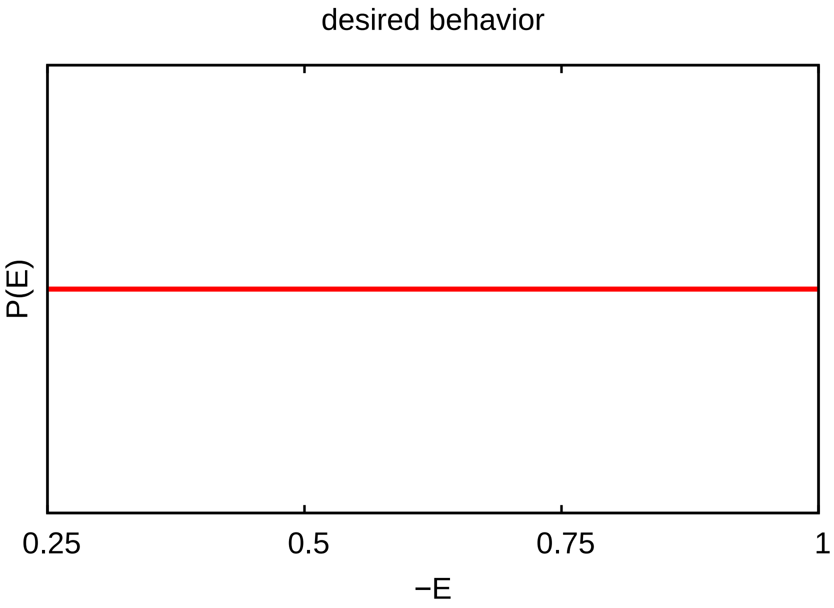
9 An *off-lattice* GCMC algorithm

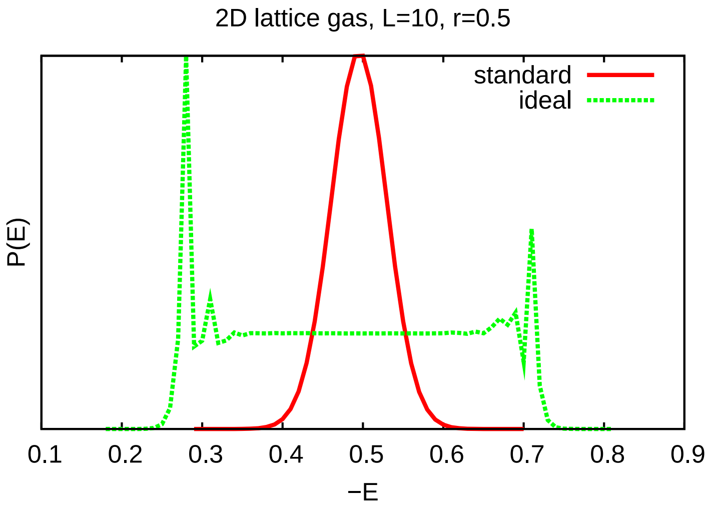
1. The more common way to simulate in the GC-ensemble is to introduce two MC moves: **insertion** and **deletion**.
2. At each step, insertion or deletion is chosen with 1/2 probability.
3. **insertion** : choose a random location in the system (site), and try to put a particle there.
4. **deletion** : choose at random one particle, and remove it.
5. Derivation of the acceptance probability using the **detailed balance** equation in Metropolis form:
$$P_{\rm acc}(\mu \to \nu) = \min\left[1, \frac{g(\nu \to \mu)}{g(\mu \to \nu)} \frac{p_\nu}{p_\mu}\right],$$
where g is the proposition probability, and p is now the GC Boltzmann weight.
6. In this algorithm, the proposition probabilities g do not cancel, i.e. the moves are **asymmetric**.
7. **Insertion moves** : Writing down the probabilities:
$$p_\mu \propto e^{-\beta E_\mu + \kappa N}, \qquad p_\nu \propto e^{-\beta E_\nu + \kappa(N+1)}, \tag{4}$$
$$g(\mu \to \nu) \propto \frac{1}{2} \times \frac{1}{V}, \qquad g(\nu \to \mu) = \frac{1}{2} \times \frac{1}{N+1}. \tag{5}$$
8. Working out terms, and using the Metropolis choice, one gets:
$$P_{\rm acc}(N \to N+1) = \min\left[1, e^{-\beta(E_\nu - E_\mu)} \frac{V e^\kappa}{N+1}\right].$$
9. **Deletion moves** : Writing out the probabilities:
$$p_\mu \propto e^{-\beta E_\mu + \kappa N}, \qquad p_\nu \propto e^{-\beta E_\nu + \kappa(N-1)}, \tag{6}$$
$$g(\mu \to \nu) \propto \frac{1}{2} \times \frac{1}{N}, \qquad g(\nu \to \mu) = \frac{1}{2} \times \frac{1}{V}. \tag{7}$$
10. Working out terms, and using the Metropolis choice, one gets:
$$P_{\rm acc}(N \to N-1) = \min\left[1, e^{-\beta(E_\nu - E_\mu)} \frac{N}{V e^\kappa}\right].$$

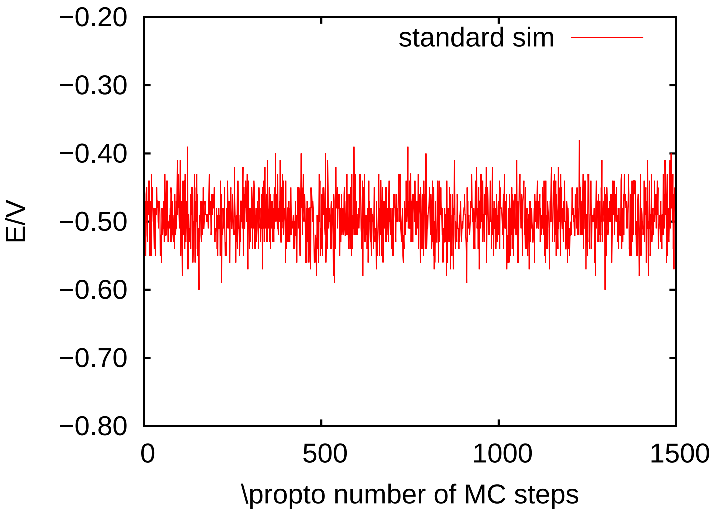
1. This is the standard behavior of the energy in a Metropolis MC algorithm where states appear in proportion to their Boltzmann weight.

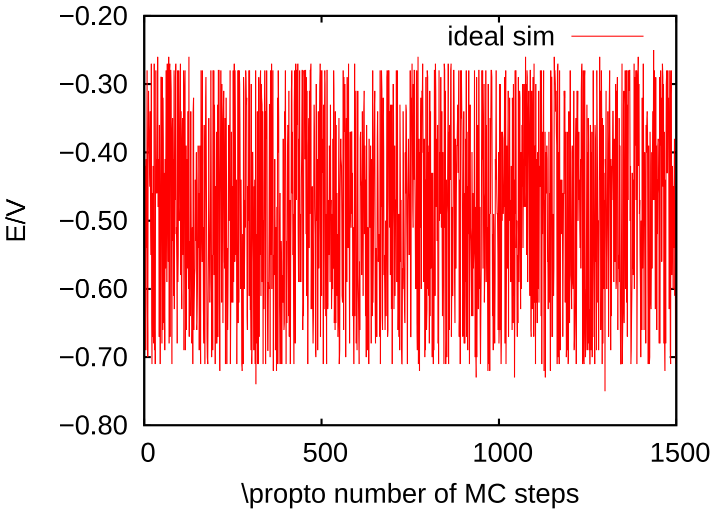


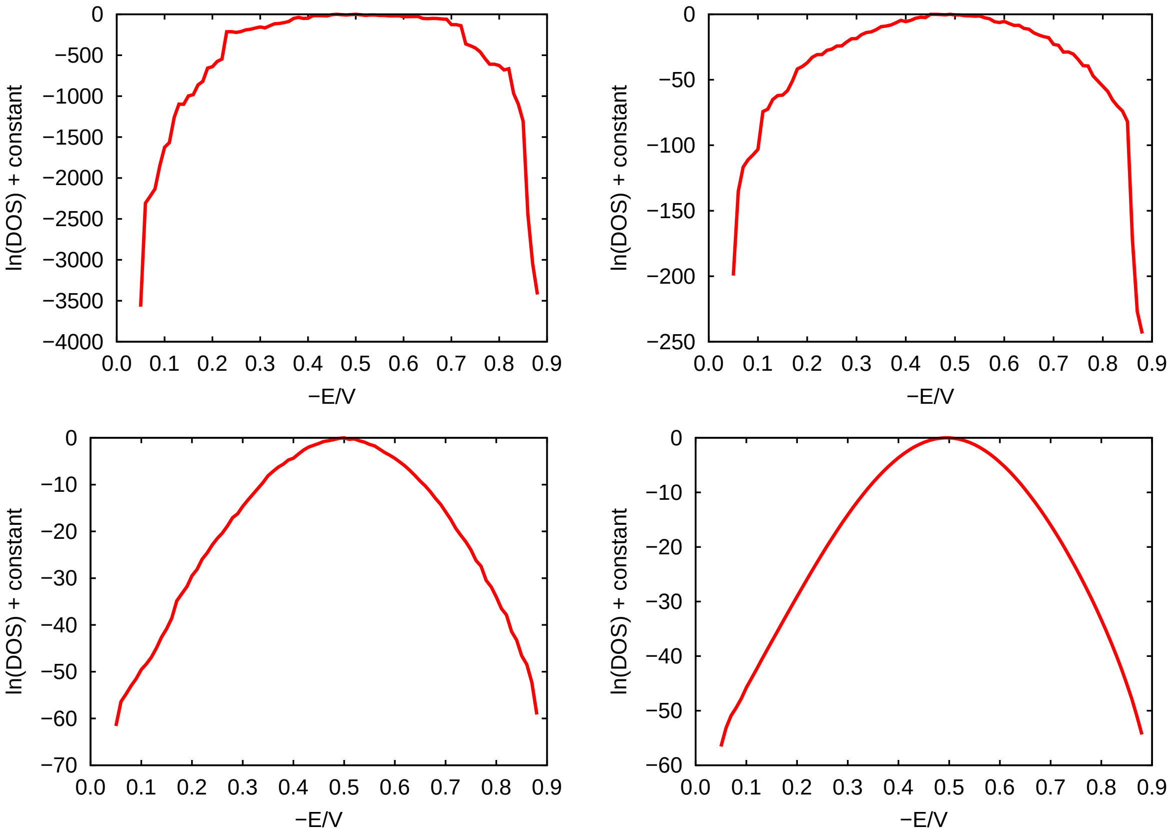
2. And this is the desired behavior we would like to have:

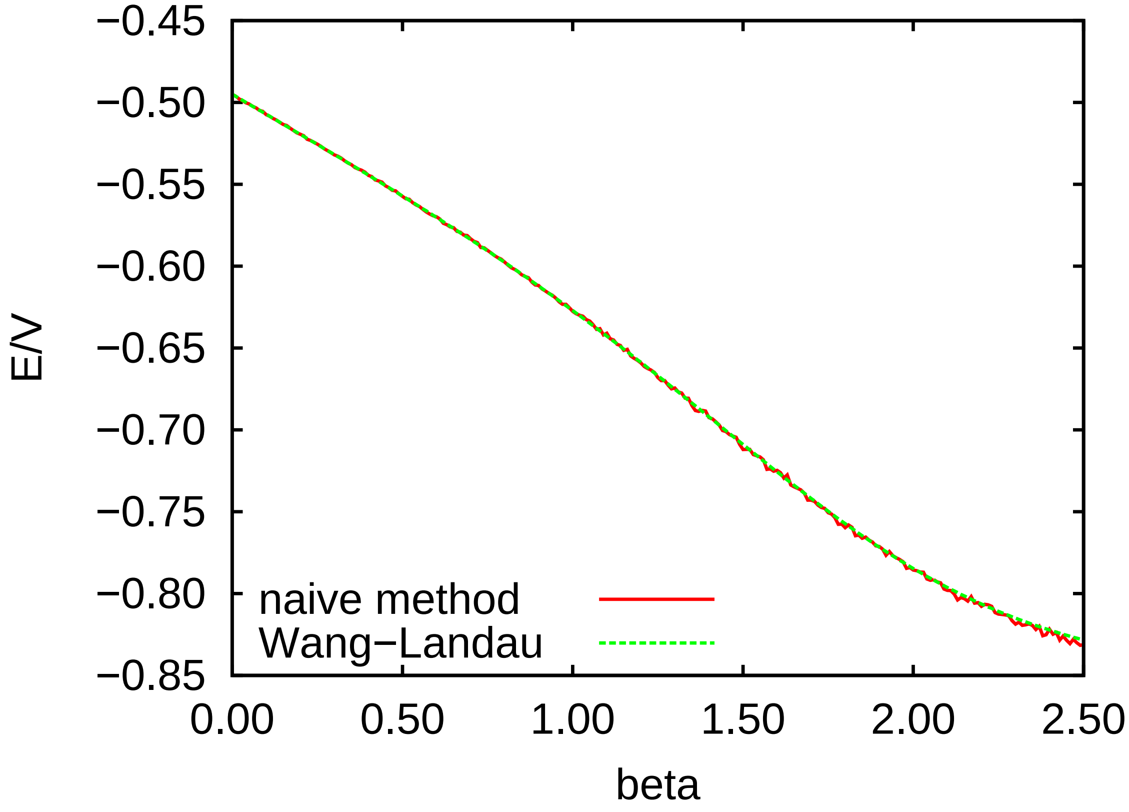


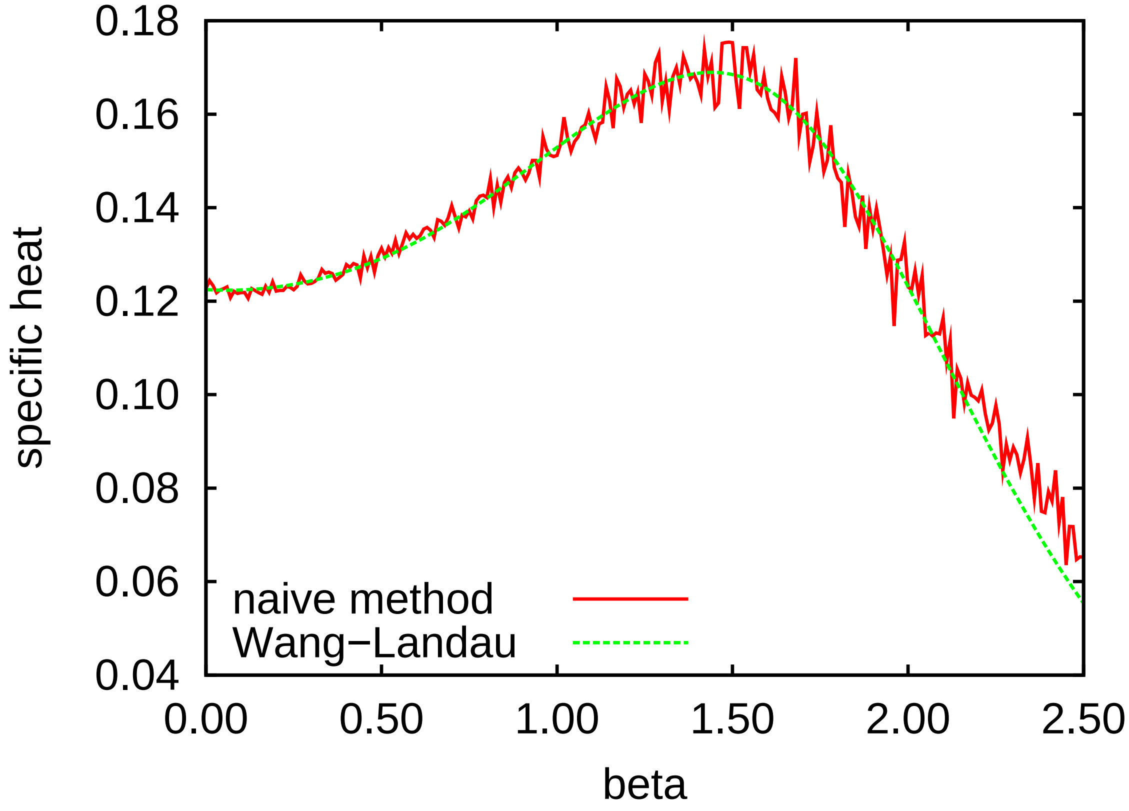

2D lattice gas, L=10, r=0.5
standard
ideal
P(E)
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
−E


standard sim
E/V
−0.20
−0.30
−0.40
−0.50
−0.60
−0.70
−0.80
0
500
1000
1500
\propto number of MC steps


ideal sim
E/V
−0.20
−0.30
−0.40
−0.50
−0.60
−0.70
−0.80
0
500
1000
1500
\propto number of MC steps


ln(DOS) + constant
0
−500
−1000
−1500
−2000
−2500
−3000
−3500
−4000
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
−E/V
ln(DOS) + constant
0
−50
−100
−150
−200
−250
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
−E/V
ln(DOS) + constant
0
−10
−20
−30
−40
−50
−60
−70
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
−E/V
ln(DOS) + constant
0
−10
−20
−30
−40
−50
−60
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
−E/V


E/V
−0.45
−0.50
−0.55
−0.60
−0.65
−0.70
−0.75
−0.80
−0.85
0.00
0.50
1.00
1.50
2.00
2.50
beta
naive method
Wang−Landau


specific heat
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.00
0.50
1.00
1.50
2.00
2.50
beta
naive method
Wang−Landau

1 Summary of previous lecture

1 Introduced the **Wang-Landau algorithm**.

- This enabled us to obtain the DOS $g(E)$ over a specified energy range:

$$E_{\rm min} < E < E_{\rm max}.$$

- Ensemble averages can be accurately obtained as function of $\beta = 1/k_BT$ from just a single WL simulation:

$$\langle E^\alpha \rangle = \frac{\int E^\alpha \, g(E) \, e^{-\beta E} \, dE}{\int g(E) \, e^{-\beta E} \, dE}.$$

2 Introduced the **grand canonical** (GC) ensemble : particle number N and energy E both **fluctuate**, while the inverse temperature β and the chemical potential κ are **constant**.

- In the GC ensemble, the probability of state μ is given by:

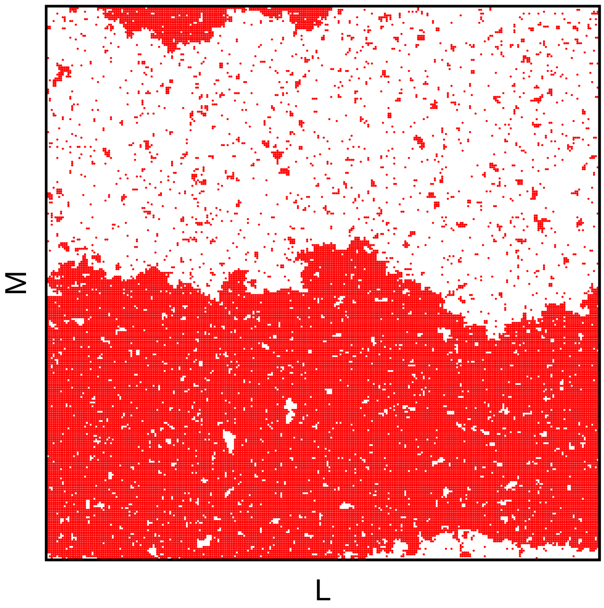
$$p_\mu \propto e^{-\beta E_\mu + \beta \kappa N_\mu},$$

where $\beta = 1/k_BT$ is the inverse temperature, κ the chemical potential, E_μ the energy of state μ, and N_μ the number of particles state μ contains.

- Presented two MC algorithms which can be used to simulate in the GC ensemble. $\rightarrow$ In particular the general variant, consisting of **insertion/deletion** moves.

- **Today** : to combine GCMC moves with WL sampling to obtain the **line tension** of the liquid-gas interface of the 2D lattice gas.

- **Recall** that the 2D lattice gas **phase-separates** at low temperatures:



2 GC simulations of the 2D lattice gas

- Implement the GC Monte Carlo algorithm for the 2D lattice gas using insertion/deletion moves. At the start of each move, choose either insertion or deletion with equal probability (flip a coin). The accept probabilities of the moves are:

$$\text{insertion :} \qquad P_{\text{acc}}(\mu \to \nu) = \min\left[1, e^{-\beta(E_\nu - E_\mu)} \frac{V e^{\beta\kappa}}{N_\nu}\right],$$

$$\text{deletion :} \qquad P_{\text{acc}}(\mu \to \nu) = \min\left[1, e^{-\beta(E_\nu - E_\mu)} \frac{N_\mu}{V e^{\beta\kappa}}\right].$$

- Since both E and N fluctuate, it is possible to collect the two-dimensional histogram: $\to H_2(E,N)$: number of times the simulation has visited a state with energy E and particle number N.

- 2D histograms are difficult to handle numerically, so one often just considers the one-dimensional histograms:

  1. $H(E)$: number of times the simulation has visited a state with energy E.
  2. $H(N)$: number of times the simulation has visited a state with particle number N.

- The 1D histograms may be regarded as "projections" of the full 2D histogram:

$$H(E) = \sum_N H_2(E,N), \qquad H(N) = \sum_E H_2(E,N).$$

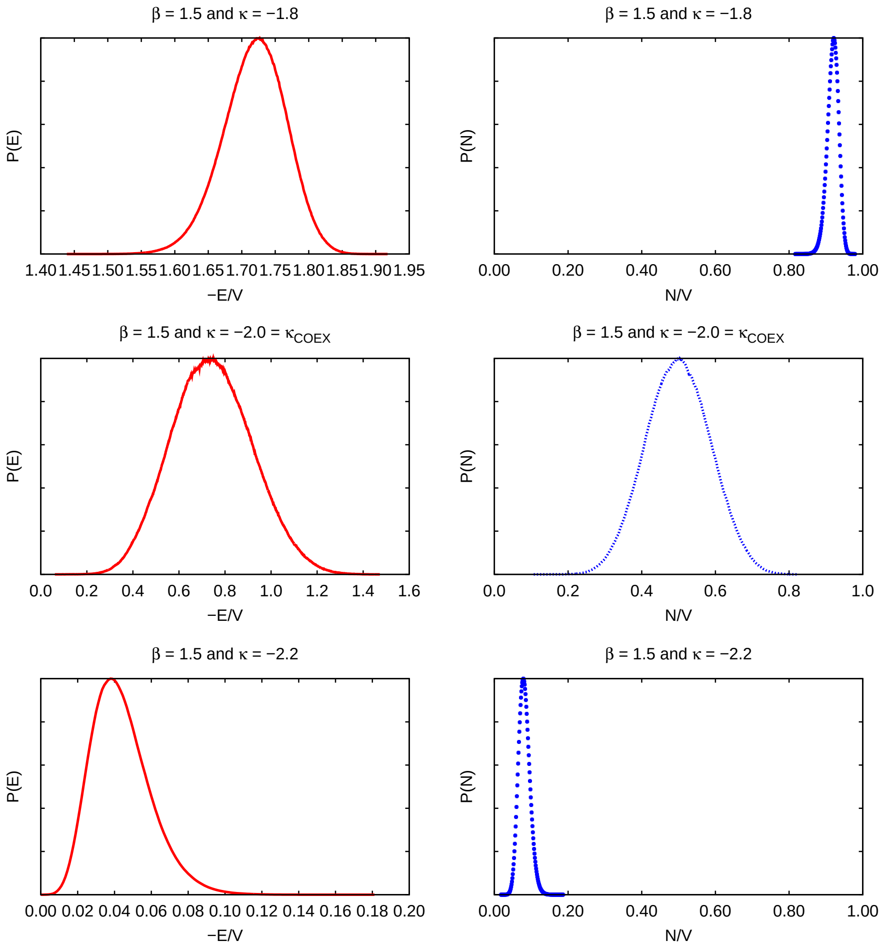
- Obviously, all histograms depend on β and κ (as well as on the size L of the lattice).

- **To do** : measure some of these distributions to get an idea of their shape.

3 GC simulation of the 2D lattice gas : high T features

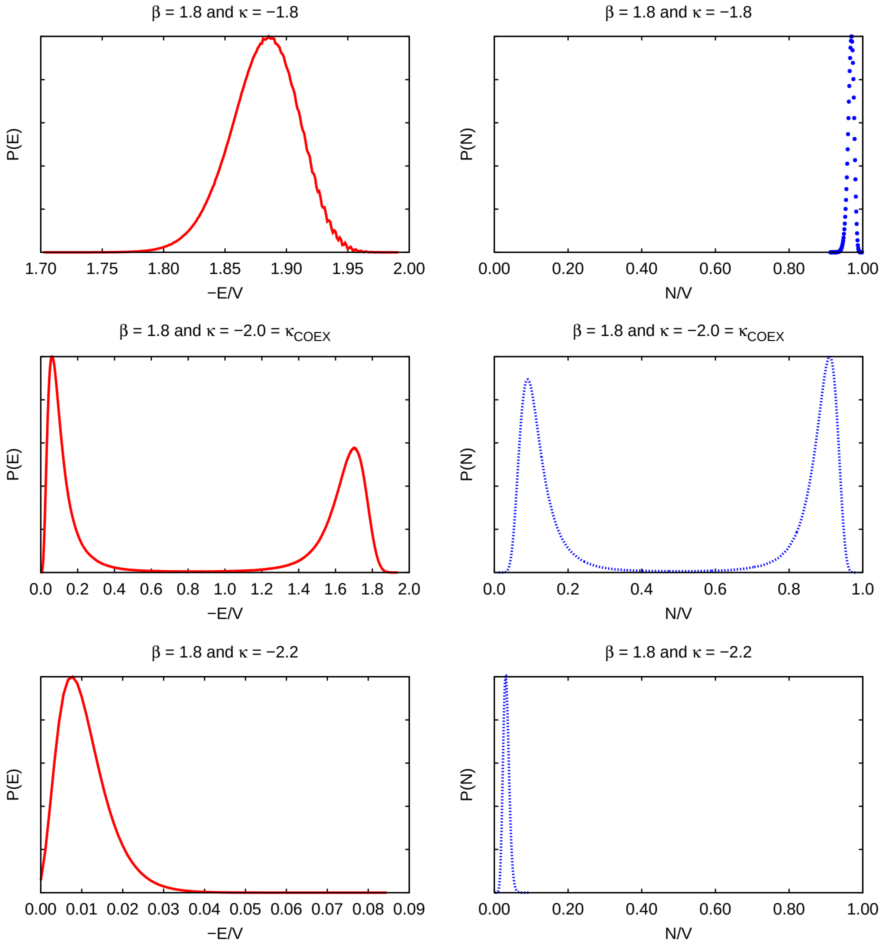
- Typical results for high temperature $\beta = 1.5$, $L = 30$, and various values of the chemical potential κ (high temperature means above the liquid-gas transition temperature).
- All simulations began with an empty system!

1. Note that E/V and κ are negative!
2. Note that $0 < N/V < 1 \rightarrow$ artifact of the lattice gas!
3. The higher κ, the higher the particle density.
   $\rightarrow$ This makes sense because in the GC ensemble $p_{\mathrm{GC}} \propto e^{\beta \kappa N}$.
4. The higher κ, the lower the energy.
   $\rightarrow$ This makes sense because with more particles in the system, there will also be more particle pairs that are nearest neighbors, and these lower the energy by an amount ϵ.
5. There is a special $\kappa \equiv \kappa_{\mathrm{COEX}}$ where the average particle density $N/V = 1/2$.
6. **Important** : At high temperature, we only see **single** peaks.



4 GC simulation of the 2D lattice gas : low T features

- Same features as before, except that at $\kappa = \kappa_{\rm COEX}$, bimodal energy and particle number distributions appear!
- Hence, at $\kappa_{\rm COEX}$, the system prefers to spend most time in **two** distinct regions, and not so much time elsewhere.
- At $\kappa_{\rm COEX}$, the peaks in $P(N)$ are distributed **symmetrically** around $N/V = 1/2$.



5 What do we see?

- There is a special $\kappa = \kappa_{\rm COEX}$.
- At high temperature and $\kappa_{\rm COEX}$, the particle number distribution $P(N)$ is a single peak centered around $\rho \equiv N/V = 1/2$.
- At low temperature and $\kappa_{\rm COEX}$, the particle number distribution $P(N)$ is double-peaked (bimodal).
- **But**, since the peaks are distributed **symmetrically** around $\rho = 1/2$, also at low temperature the **average** particle density equals $1/2$.
- Hence, $\kappa_{\rm COEX}$ reflects a hidden symmetry of the system, at which the average particle density always equals $\rho = 1/2$.

6 Understanding $\kappa_{\rm COEX}$

- For the 2D lattice gas it holds that $\kappa_{\rm COEX} = -2$ exactly. Why?
- The lattice gas is isomorphic (identical) to the Ising model.
- Recall that the GC energy of the lattice gas reads as:

  $$E_{\rm GC} = -\epsilon \times P - \kappa \times N,$$

  with P the number of particle pairs that are nearest neighbors, and N the total number of particles.
- Introduce the occupation variables:

  $$\sigma_i = \begin{cases} 1 & \text{if lattice site i is occupied,} \\ 0 & \text{otherwise.} \end{cases}$$

- One can then write:

  $$E_{\rm GC} = -\frac{\epsilon}{2} \sum_{i=1}^{V} \sum_{j \in nn(i)} \sigma_i \sigma_j - \kappa \sum_{i=1}^{V} \sigma_i,$$

  with V the total number of lattice sites, and where the factor of $1/2$ removes the double-counting of bonds.
- Introduce $s_i = 2\sigma_i - 1$. Since $\sigma_i = 0$ or 1, it holds that $s_i = \pm 1$, i.e. an Ising spin variable.

- Using $\sigma_i = (s_i + 1)/2$, one can rewrite $E_{\rm GC}$ as:

$$E_{\rm GC} = -\frac{\epsilon}{8} \sum_{i=1}^{V} \sum_{j \in nn(i)} (s_i s_j + s_i + s_j + 1) - \frac{\kappa}{2} \sum_{i=1}^{V} (s_i + 1).$$

- Assume that each lattice site has z nearest neighboring sites (for the 2D square lattice it holds that $z = 4$). One can then simplify the linear terms as:

$$\sum_{i=1}^{V} \sum_{j \in nn(i)} s_i = z \sum_{i=1}^{V} s_i, \qquad \sum_{i=1}^{V} \sum_{j \in nn(i)} s_j = z \sum_{i=1}^{V} s_i.$$

- This leads to:

$$E_{\rm GC} = -\frac{\epsilon}{8} \sum_{i=1}^{V} \sum_{j \in nn(i)} s_i s_j - \left(\frac{\epsilon z}{4} + \frac{\kappa}{2} \right) \sum_{i=1}^{V} s_i + \text{constant},$$

  which is just the Ising model of the second lecture in an external field:

$$H = \frac{\epsilon z}{4} + \frac{\kappa}{2}.$$

- Since the Ising model has spin reversal symmetry, it holds that $H = 0$ at the phase transition. For the lattice gas this implies:

$$\kappa_{\rm COEX} = -z\epsilon/2.$$

- On a 2D square lattice $z = 4$ hence $\kappa_{\rm COEX} = -2\epsilon$, and since in my simulations $\epsilon \equiv 1$, it explains the values used in the graphs.

- **Important** : this simple result only holds for the lattice gas. For more realistic fluids, such as Lennard-Jones, $\kappa_{\rm COEX}$ is typically not known beforehand.
  $\rightarrow$ We will solve this problem shortly!

7 Interpretation of the result

- Simulating at $\kappa_{\rm COEX}$, we find that $P(N)$ is single-peaked at high temperature, and double-peaked at low temperature.

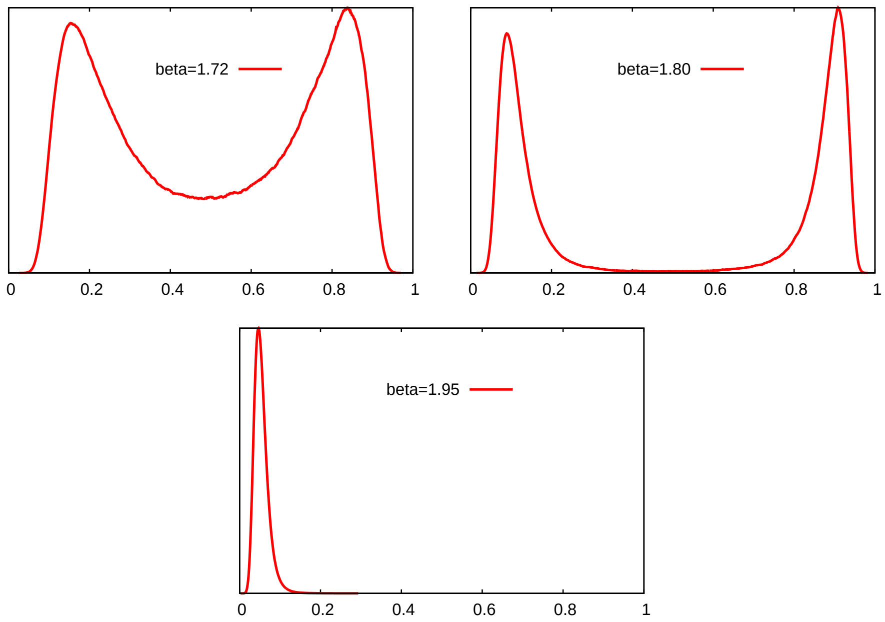
- Peaks in distributions may be regarded as phases.

- Hence, at high T there is only one phase, but at low T we have two-phase coexistence.

- In particular, the peak positions at low T yield the particle densities of the two coexisting phases:
  $\rightarrow$ The peak at low density corresponds to the gas phase.
  $\rightarrow$ The peak at high density corresponds to the liquid phase.

- Hence, one can construct a phase diagram, by measuring the coexisting densities as a function of temperature. But, as it turns out, there is still one problem.

- Variation of $P(N)$ with β, obtained for 2D lattice gas at $\kappa_{\rm COEX}$ and $L = 30$:



- **Problem** : in order to go from one peak to the other, the simulation has to traverse through a region of low-probability. And this probability **decreases** with increasing β. When β is too high, we “get stuck”, and only see one peak.

- For the 2D lattice gas, one workaround is to use the **symmetry** of the system. If one observes a peak at $\rho_{\rm gas} = x$, then the density of the liquid phase by symmetry equals $\rho_{\rm liq} = 1 - x$. **But**, most systems do not have this symmetry, and so this “fix” will not work in general.

- More useful is to ask **why** the simulation “gets stuck”, and how to **solve** it.

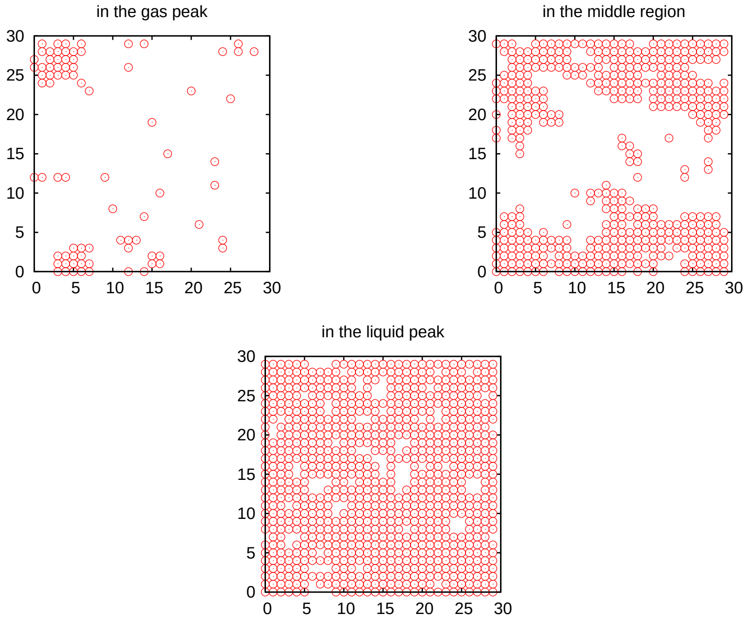
8 The origin of “getting stuck” : free energy barriers

- The probability to “traverse” from one peak to the other becomes smaller with increasing β.
- Apparently, there exists a “barrier” ΔF that needs to be crossed in order to traverse, and this barrier becomes larger the higher β gets:

$$p_{\text{traverse}} \propto e^{-\Delta F}.$$

- What is the origin of the barrier in case of the 2D lattice gas? Or, more generally, of any first-order transition?
- Reconsider the simulation at $\beta = 1.8$ and $L = 30$, where things “still worked”. Take pictures (snapshots) of particle configurations at three distinct regions:

  1. In the region of the gas peak.
  2. In the region “between the peaks”, i.e. at $\rho = 1/2$.
  3. In the region of the liquid peak.



- In the peak regions, we observe pure phases: one lean phase (gas) and one dense phase (liquid).
- In the middle, however, we see phase separation! Gas and liquid coexist. Hence, there is a substantial amount of interface. On a 2D periodic square, the total amount of interface has a length of order $2 \times L$.

- But interfaces have a tension σ (interface tension, or line tension in 2D), and so interface formation costs (free) energy. Hence, the barrier that prevents the simulation from traversing is due to interface formation:

$$\Delta F = 2\sigma L, \qquad (2D). \tag{1}$$

- At low temperature, σ is large, implying that the barrier is large. Precisely at the transition, σ vanishes, implying that the barrier is zero. This indeed explains why the simulations "got stuck" at low temperature, but not at high temperature.
- **Important** : note that Eq.(1) also implies that the problem of "getting stuck" is even more severe in larger systems!

9 How to get "unstuck"

- The problem is that our GC simulation "gets stuck" in one of the peaks of the distribution $P(N)$, whereas we would ideally like to sample the entire density range $0 < N/V < 1$.
- This is exactly what we solved last week : Wang-Landau sampling!
- Last week, WL sampling was used to sample over a specified **energy** range, but there is nothing that prevents us from generalizing this to a specified **particle density** range.
- Things are slightly more complicated, however, since in the GC ensemble, both energy E and particle number N fluctuate (governed by the respective control parameters β and κ).
- As a result, the DOS $g(E, N)$ is two-dimensional, depending on both E and N.
  $\rightarrow$ Recall that the DOS now "counts" the number of states with energy E and particle number N.
  $\rightarrow$ Note also that the DOS does **not** depend on β, **nor** on the chemical potential κ.
- Ensemble averages for any β and κ can be calculated once $g(E, N)$ is known, for example:

$$\langle N^2 E \rangle = \frac{\sum_E \sum_N N^2 E g(E, N) e^{-\beta E + \beta\kappa N}}{\sum_E \sum_N g(E, N) e^{-\beta E + \beta\kappa N}},$$

  where we now have double-sums. Again, the β, κ dependence is contained solely in the GC Boltzmann factor.

- We can easily formulate a 2D Wang-Landau algorithm to obtain $g(E, N)$. Using insertion/deletion moves attempted at each step with equal probability, and accepting as follows:

$$\begin{aligned} P_{\text{insert}}(\mu \to \nu) &= \min\left[1, \frac{V}{N_\nu} \frac{g(E_\mu, N_\mu)}{g(E_\nu, N_\nu)}\right], \qquad (2) \\ P_{\text{delete}}(\mu \to \nu) &= \min\left[1, \frac{N_\mu}{V} \frac{g(E_\mu, N_\mu)}{g(E_\nu, N_\nu)}\right], \qquad (3) \end{aligned}$$

  the DOS can be obtained following the usual WL iteration scheme.
  → **Note** that in this algorithm β and κ do not appear!
  → **Recall** from last week that insertion/deletion moves are **asymmetric**, hence the accept probabilities contain a "correction" prefactor involving N_μ and V.

Alternative approach

- However, as said before, 2D histograms are not the most convenient to deal with, so an alternative method will be proposed.

- Introduce the "integrated" DOS:

$$g_\beta(N) = \sum_E g(E, N) e^{-\beta E}.$$

  → For fixed N, we thus integrate over all energies.
  → Note that the result depends on β, but not on κ.
  → The "integrated" DOS is, of course, just the **canonical partition sum**.

- We can use the "integrated" DOS to obtain ensemble averages as function of any κ, but for **fixed** β only, for example:

$$\langle N \rangle = \frac{\sum_N N g_\beta(N) e^{\beta\kappa N}}{\sum_N g_\beta(N) e^{\beta\kappa N}}.$$

- Inspecting the structure of the above equation, one also finds that:

$$P(N) \propto g_\beta(N) e^{\beta\kappa N}. \qquad (4)$$

  → This equation is very important because it allows us to convert $g_\beta(N)$ into the particle number probability at any chemical potential, in particular $\kappa = \kappa_{\text{COEX}}$.

- Since the "integrated" DOS is a one-dimensional function, the corresponding WL algorithm is easy to implement.

- We again use insertion/deletion moves, attempted with equal probability, but now accept them according to:

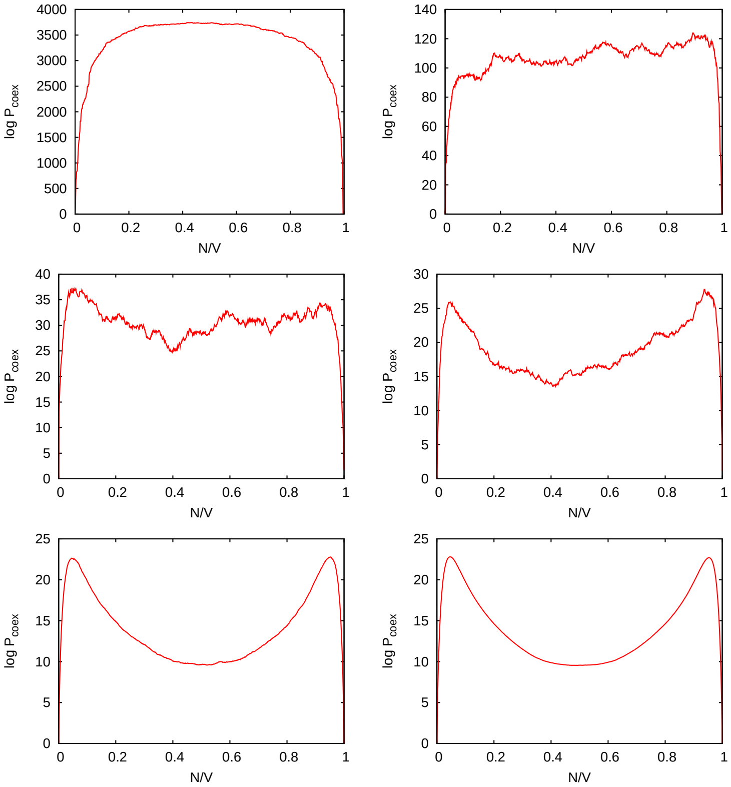
$$P_{\text{insert}}(\mu \to \nu) = \min\left[1, \frac{V}{N_\nu} \times \frac{g_\beta(N_\mu)}{g_\beta(N_\nu)} \times e^{-\beta(E_\nu - E_\mu)}\right], \tag{5}$$

$$P_{\text{delete}}(\mu \to \nu) = \min\left[1, \frac{N_\mu}{V} \times \frac{g_\beta(N_\mu)}{g_\beta(N_\nu)} \times e^{-\beta(E_\nu - E_\mu)}\right]. \tag{6}$$

- In this approach, we do standard Metropolis importance sampling on the energy E, but WL sampling on the particle number N.
  → **Note** that this algorithm depends on β, but not on κ.

- Below is an application of this method to the 2D lattice gas at $\beta = 1.95$ and $L = 30$. Shown is $\ln P(N)$ at the coexistence chemical potential, which is obtained from $g_\beta(N)$ via Eq.(4).
  → The previous problem of traversing has clearly been solved, and we obtain a double-peaked distribution, including **excellent** data in the middle region!

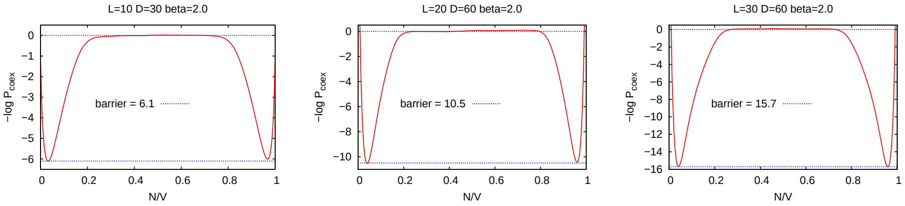
- **Note** : if κ_{COEX} is not known beforehand, it can be located with Eq.(4) using trial-and-error values of κ until the two peaks appear! (for example: equal-area or equal-height).



10 The line tension

- Naively, the probability of traversing $p_{\rm traverse} \propto e^{-\Delta F}$, where ΔF is the barrier of interface formation.

- Hence, the barrier is "encoded" in the **natural logarithm** of $P(N)$, which is just the free energy:
$$F \propto -k_B T \ln P(N),$$
from which ΔF can be directly read-off, see below:



- The large system yields a **huge** barrier, clearly showing that WL sampling is essential:
$$p_{\rm traverse} \sim e^{-15.7} \sim 10^{-7}.$$

- In a rectangular $L \times D$ box, with $D \gg L$, we expect the interfaces to form **perpendicular** to the elongated edge D. $\rightarrow$ **draw picture on blackboard.**

- The total amount of line interface thus equals $2L$.

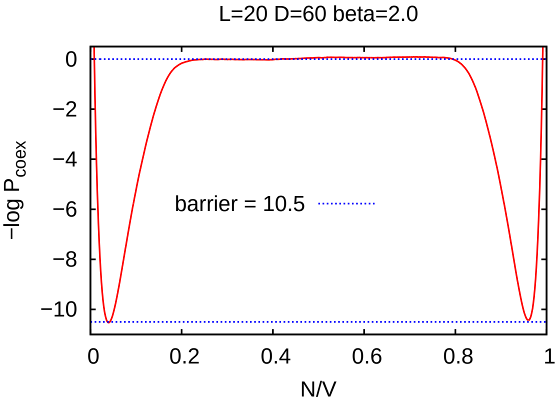
- The interface cost per unit of length, i.e. the **interfacial tension**, thus equals to:
$$\sigma = \Delta F/(2L),$$
which offers a **precise** way to measure the interfacial tension. In a 3D $L \times L \times D$ periodic box one would get:
$$\sigma = \Delta F/(2L^2).$$

- Good agreement for the largest systems, while the smallest system is somewhat off:
$$\sigma_{L=10} = 0.305, \qquad \sigma_{L=20} = 0.263, \qquad \sigma_{L=30} = 0.262,$$
(in units of $k_B T$ per lattice spacing).

- Another important aspect of these graphs is that they illustrate **finite-size effects**! $\rightarrow$ What you get from a simulation typically depends on how big the system is that you simulate! This may be considered a nuisance, although in practice it actually enhances our understanding of things (such as in the graphs above).

The flat region between the peaks in $\ln P(N)$

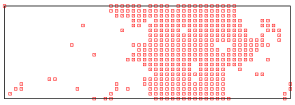
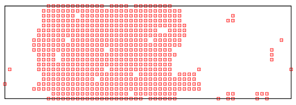
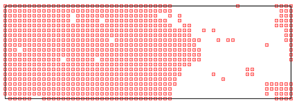
- Consider again the coexistence particle number distribution $\ln P(N)$ at $\beta = 2$, $L = 20$, $D = 60$ for the 2D lattice gas:



- There is a flat region between the peaks. What is the origin of this?

- The physical interpretation of $\ln P(N)$ is that of **minus** the free energy. Over some density range, we can apparently vary the particle density at zero cost in free energy because $\ln P(N)$ is flat.

- So what is going on here? Consider the following "snapshots" taken at $\rho = N/V = 1/3, 1/2, 2/3$.

- All snapshots reveal coexistence, with the interfaces perpendicular to the elongated edge; only the relative amount of the phases changes.

- Put differently: as you increase ρ, you are shifting the interfaces. Since the free energy is flat, this apparently does not cost any effort. Why?

- Coexistence means equal pressure in both phases. Hence, the work that needs to be done to shift the interfaces is zero! **Provided the interfaces do not interact!**

- Apparently, this is the case here, and hence the flat region. Note that, by using a rectangular box, this effect can be enhanced because the distance between the interfaces is larger then.

- If you want to use the barrier ΔF to measure the line tension accurately, the presence of a flat region is essential.

11 Ways of measuring efficiency

- In research → aim for efficient simulation algorithms!
- How "efficient" are the GC Monte Carlo algorithms that were presented for the lattice gas in previous lectures?
- How does one measure the **efficiency** of a simulation? This is not so trivial as it may seem.

The accept rate

- To address the issue of **efficiency** we will consider the two GC algorithms that were introduced two lectures ago.
- Algorithm one : **symmetric algorithm** (only works on lattice)

  1. Randomly select lattice site.
  2. If site is empty → insert particle; otherwise remove particle.
  3. Accept the resulting state with probability:

$$P_{\text{acc}}(\mu \to \nu) = \min \left[1, e^{-\beta(E_\nu - E_\mu) + \kappa(N_\nu - N_\mu)}\right] .$$

- Algorithm two : **general algorithm** (also works off-lattice)

  1. With probability of $p = 1/2$ decide whether to **insert** or **delete** a particle.
  2. In case of insertion, choose a lattice site at random and put a particle there. Accept the resulting state with probability:

$$\text{insertion :} \qquad P_{\text{acc}}(\mu \to \nu) = \min \left[1, e^{-\beta(E_\nu - E_\mu)} \frac{V e^{\beta\kappa}}{N_\mu + 1}\right] .$$

  3. In case of deletion, choose a particle at random, and remove this particle. Accept the resulting state with probability:

$$\text{deletion :} \qquad P_{\text{acc}}(\mu \to \nu) = \min \left[1, e^{-\beta(E_\nu - E_\mu)} \frac{N_\mu}{V e^{\beta\kappa}}\right] .$$

- As usual, E and N denote energy and particle number; β and κ are the inverse temperature and chemical potential.
- What the algorithms have in common is that moves are sometimes **accepted**, but also **rejected**. Rejected moves in standard sampling schemes do not lead to any new information, and are thus be considered **bad**, in the sense that they waste time.
- **Idea 1** : measure the accept rate; surely the algorithm with the highest accept rate is the best!

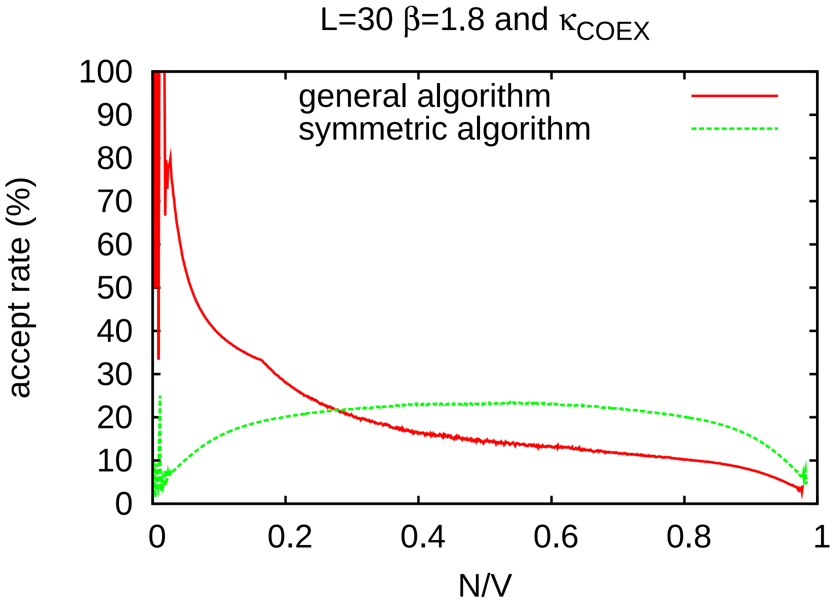
- **Application to 2D lattice gas** : system size $L = 30$, at the coexistence chemical potential $\kappa_{\rm COEX}$, for two values of β:

$$\beta = 1.0 \quad : \quad \text{symmetric algorithm} \sim 60\%, \quad \text{general algorithm} \sim 38\%, \tag{7}$$
$$\beta = 1.8 \quad : \quad \text{symmetric algorithm} \sim 16\%, \quad \text{general algorithm} \sim 25\%. \tag{8}$$

- In general, accept rates get smaller at low temperature. At high temperature, the symmetric algorithm yields the highest accept rate, while at low temperature the general algorithm "wins". So if you want to to study phase separation, which occurs at low temperature, the general algorithm appears to be the method of choice.

- But does this make sense? At high density, the general algorithm will often try to insert particles onto sites that are already occupied. These moves will always be rejected. Surely, this cannot be efficient. So let's do a more refined analysis.

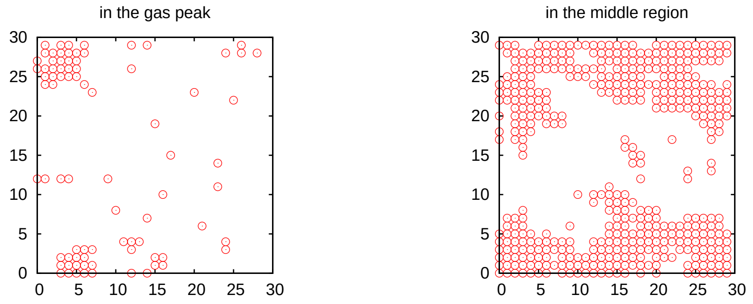
- **Idea 2** : monitor the accept rate as function of the particle density N/V. Result for $L = 30$, $\beta = 1.8$, and at $\kappa_{\rm COEX}$:



- The **general algorithm** indeed **fails** at high density! This is dangerous because it implies that the liquid phase is not sampled well, only the gas phase is. Even though the accept rate may on average be high, some regions in phase space are sampled very poorly!

- In contrast, the **symmetric algorithm** performs equally well in both phases.

- **Note** : the general algorithm has a "funny kink" at $\rho \sim 0.15$: this is where the transition from the homogeneous gas phase to the liquid phase occurs! Hence, sudden changes in the accept rate can sometimes signify interesting physics taking place.



- But, the most important message to take from this is that the accept rate is generally not a good criterion to determine the efficiency of a simulation!

- Of course, an accept rate of zero is always bad!

A better measure of efficiency: the auto-correlation function $\chi(t)$

- The problem is that the accept rate does not tell you anything about how much the **particle configuration actually changes**.

- **Extreme example** : assume you are always inserting/removing a particle at the same lattice site. The accept rate might well be close to 100 %, but it is clear the configuration as a whole does not change much.

- A more stringent measure of how much the configuration changes during the course of t Monte Carlo steps is the **auto-correlation**:

$$\chi(t) \propto \langle X(t)X(0)\rangle - \langle X(t)\rangle\langle X(0)\rangle.$$

  $\rightarrow X(t)$ is the value of some observable X at Monte Carlo step t; obvious choices for X are the energy E or particle number N.

- If $X(0)$ and $X(t)$ are correlated $\chi(t)$ will be finite.

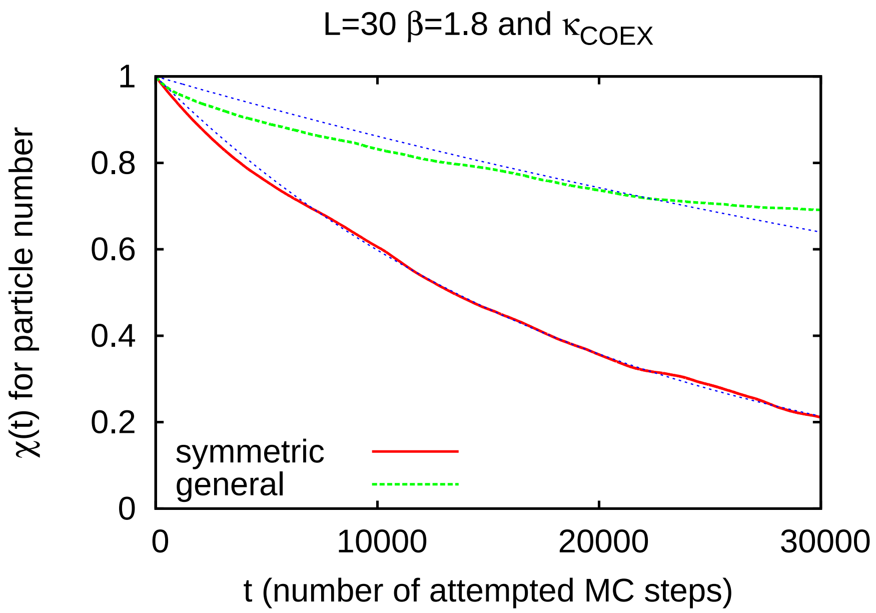
- If $X(0)$ and $X(t)$ are uncorrelated $\chi(t) \rightarrow 0$.

- A good algorithm is one in which $\chi(t)$ decays to zero quickly with t.

- In practice, the correlation function is rather more complex to calculate. Given a Monte Carlo time series of observables $X(0), X(1), \ldots, X(n)$, the auto correlation function is given by:

$$\chi(t) = \left(\frac{1}{n-t} \sum_{t'=0}^{n-t} X(t')X(t'+t) \right) - \left(\frac{1}{n-t} \sum_{t'=0}^{n-t} X(t') \right) \times \left(\frac{1}{n-t} \sum_{t'=0}^{n-t} X(t'+t) \right).$$

- **Application to the 2D lattice gas**, $L = 30$, $\beta = 1.8$, at coexistence:



- The **symmetric** algorithm clearly outperforms the **general** variant.
- Note that typically $\chi(t) \propto e^{-t/\tau}$. By fitting one finds that:

$$\tau_{\text{symmetric}} \sim 19000, \qquad \tau_{\text{general}} \sim 67000.$$

$\rightarrow$ At these values of β, L, and κ, the symmetric algorithm thus decorrelates about three times faster.

Today: the transition matrix Monte Carlo method

1 Rejected moves

- Based on the decay of the auto-correlation of the particle number, the symmetric algorithm is thus superior. So I will use that for the rest of the lecture.
- However, the symmetric algorithm is still not very good, because the accept rate is at most ≈ 25 %.
- The vast majority of MC moves is thus rejected. More precisely, 75% of CPU time is plainly wasted.
- Do rejected moves not carry any information at all?
- yes, they do: transition matrix!

2 Transition matrix Monte Carlo

- The standard approach in Monte Carlo simulations is to:

  (0) Starting in state μ (say with energy E_μ and particle number N_μ).

  (1) Propose (generate) a new state ν (with corresponding E_ν and N_ν).

  (2) Accept the new state ν with some probability; otherwise reject it.
- Most CPU time is consumed in step (1), which is essentially wasted if in step (2) you decide to reject the move.
- However, the proposed state ν already contains some information by itself, even if you reject it in step (2).
- To see this, consider the density of states, which is two-dimensional in the present grand-canonical context: $g(E, N)$.
- Recall the "naive" definition of $g(E, N)$: it counts how many states there are with energy E and particle number N. In other words, it measures **degeneracy**.
- **Example**: In the 2D lattice gas, there is only one state where $N = V = L^2$ (all sites occupied); the energy of this state equals $E = -2\epsilon V$. And if $N = 1$, there are V possible locations for this single particle, all of which have zero energy. In terms of the DOS:
$$g(-2\epsilon V, V) = 1, \quad g(0, 1) = V.$$
- **Key observation** : if you randomly generate a grand-canonical lattice gas configuration, it is V times more likely that you select a state with $N = 1$ particle compared to $N = V$ particles.

- Recall now the above MC algorithm. The state ν that you generate in step (1) of the scheme is also generated randomly. Hence, the DOS at the corresponding energy E_ν and particle number N_ν of state ν is likely to be large. This observation holds regardless of what you do with state ν in step (2).
- So it makes sense to also record (count) the transitions from $\mu \to \nu$ that you **propose** in step (1), **irrespective of whether you accept** state ν.
  $\to$ transition matrix Monte Carlo method.

3 Definition

- Recall that the DOS $g(E, N)$ is two-dimensional. In other words, we characterize the states with their energy E and particle number N.
- We now introduce the transition matrix:

$$C\left[(E_\mu, N_\mu), (E_\nu, N_\nu)\right].$$

  It is defined as follows: being in a state μ, characterized by its energy E_μ and particle number N_μ, how often has the MC scheme **proposed** state ν, with respective energy E_ν and particle number N_ν?
- Hence, one needs to include one “bookkeeping” line in the MC algorithm at step (1) where the C matrix is updated by one.
- To make life easier, I will use the “shorthand” notation:

$$\vec{\mu} \equiv (E_\mu, E_\nu),$$

  such that the transition matrix can be written as: $C(\vec{\mu}, \vec{\nu})$.
- The transition matrix is generally memory intensive!
- For the 2D lattice gas, using the symmetric algorithm, it holds that:
  1. For each proposed move, N changes by ± 1.
  2. For each proposed MC move, E changes by at most $\pm 4\epsilon$.
- Hence, given a state $\vec{\mu}$, there are at most 18 possible proposed states $\vec{\nu}$. Since $0 \le N \le V$ and $-2V \le E \le 0$, this implies a memory load of $M = 18(V+1)(2V+1)$ integers, i.e. $M \propto V^2$.
- The scaling of the memory consumption with system size may well become a problem in practice! Fortunately, as I will show later, one does not usually need to store all transitions. But for now assume that we do.

4 Relation between transition matrix and DOS

- From the transition matrix one can calculate the **transition probability**:

  $$T(\vec{\mu},\vec{\nu}) \equiv \frac{C(\vec{\mu},\vec{\nu})}{\sum_{\vec{\nu}} C(\vec{\mu},\vec{\nu})}.$$

  This is the probability that, being in state $\vec{\mu}$ the algorithm **proposes** a move to state $\vec{\nu}$. Of course, the longer you run the MC algorithm, the more accurate P gets.

- If $T(\vec{\mu},\vec{\nu})$ is large, then the DOS $g(\vec{\nu})$ of the **proposed** state $\vec{\nu}$ must be large. In fact, it holds that:

  $$T(\vec{\mu},\vec{\nu}) \propto g(\vec{\nu}),$$

  provided the MC moves are **symmetric**.

- Hence, we can write:

  $$\frac{g(\vec{\mu})}{g(\vec{\nu})} = \frac{T(\vec{\nu},\vec{\mu})}{T(\vec{\mu},\vec{\nu})}.$$

- In principle, we only need to know the DOS up to a prefactor. So if you put $g(\vec{\mu}_0) \equiv 1$, with $\vec{\mu}_0$ some arbitrarily chosen reference state, the above equation allows you to construct the DOS at other values.

- Note that this problem is generally over-specified! You have more equations than needed → optimization procedure!

- However, I prefer to illustrate the method using a more simple example where this issue does not occur → liquid crystals!

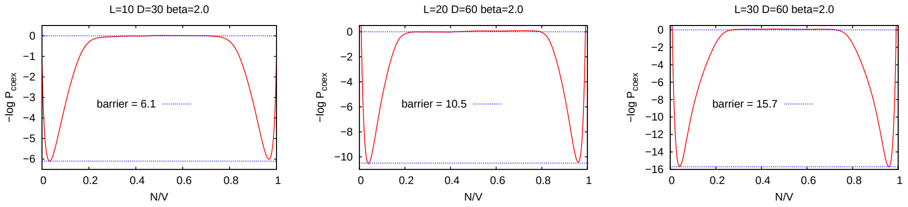
1 Summary of previous lecture

1 Liquid Crystals.

- Introduced the isotropic-to-nematic transition.
- Application to the LL model (thermotropic system).
- Bimodal energy distribution $P(E)$.
- Huge gain in using the transition matrix.

2 Today: Finite-size effects

- In any computer simulation, due to limited memory and CPU time, we can only handle a finite number of particles or system size.
- The simulations up to now generally used square or cubic lattices of edge L, i.e. lattices of finite size.
- A natural question is : do results that we obtain in simulations depend on L?
- From our previous simulations of the 2D lattice gas, we already know that the answer can be yes!
- Recall the logarithm of the GC distribution $P(N)$ at coexistence for the 2D lattice gas:



- The barrier clearly depends on the size of the system!
- The reason is that the barrier corresponds to the free energy cost of interface formation, and so in 2D we expect that $\Delta F \propto L$, which can be used to extract the line tension.
- In general, one should expect that results obtained in a simulation depend on the size L of the simulated system, especially near phase transitions.
- Since we are typically interested in **thermodynamic limit** behavior $L \to \infty$, there is obviously a gap to bridge!
- In this lecture I want to illustrate some of these points using forest fires as an example.

3 A simple model of forest fires

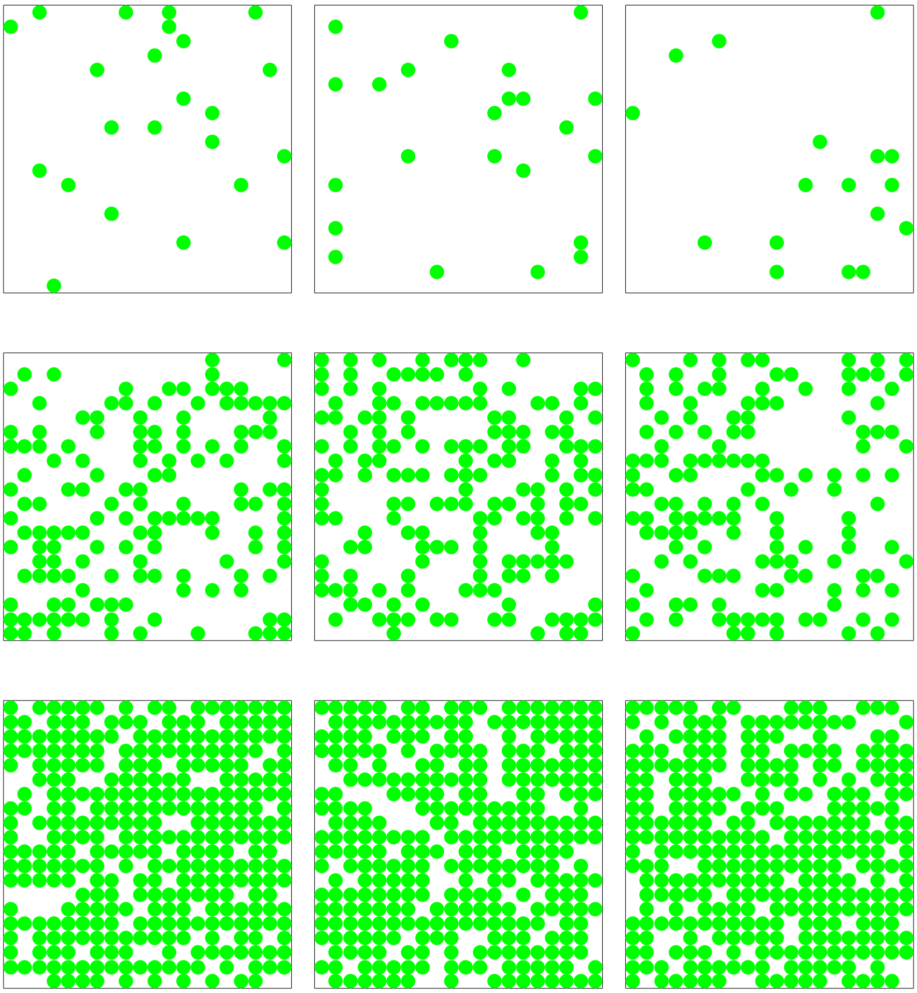
- Consider a farmer growing x-mas trees.
- He owns a peculiar piece of land. It is completely square, and sub-divided into $L \times L$ sub-plots.
- Each sub-plot can hold one x-mas tree.
- The trees are genetically manipulated and extremely fast growing. They can be planted in spring, and still be harvested in the same year, ready to be sold for x-mas.
- Unfortunately, during the summer it gets hot and dry, and the farmer is plagued by forest fires.
- When a tree is on fire, it will also put the trees on nearest neighboring plots on fire. Recall that each plot has 4 nearest neighbors, except the ones on the edge.
- Clearly, if all plots have a tree growing on them, and one tree catches on fire, all trees burn down. This is bad.
- To prevent this, the farmer decides to not plant a tree on every plot, but to plant a tree on each plot with a probability $0 \leq p \leq 1$.
- That is, in spring, the farmer visits each plot once, selects a random number $0 \leq r < 1$, and if $r < p$ he plants a tree there.
- The case $p = 1$ corresponds to all plots having a tree, which is bad if one of the trees catches fire. If $p = 0$ no tree is planted at all, which is good from the point of view of fire, but not good for profit.
- **Question** : is there an “optimal” value of p, whereby a forest fire does not spread all over?

4 Monte Carlo approach to solving the problem

- It is very easy to simulate the problem of the farmer.
- To simulate the “production process” of one year, we prepare a $L \times L$ grid, visit each grid point once, and with a probability p mark the point as occupied (1) or not (0).

  ```
  for(i=0;i<L*L;i++) {
    if(random_number<p) plot[i]=1;
    else plot[i]=0;
  }
  ```

  This process is then repeated many times, to get an idea of what a “typical” harvest averaged over many years looks like.
- Below are some “typical” x-mas tree fields for $p = 0.05$, $p = 0.4$, and $p = 0.8$, from top to bottom, on a 20×20 grid:



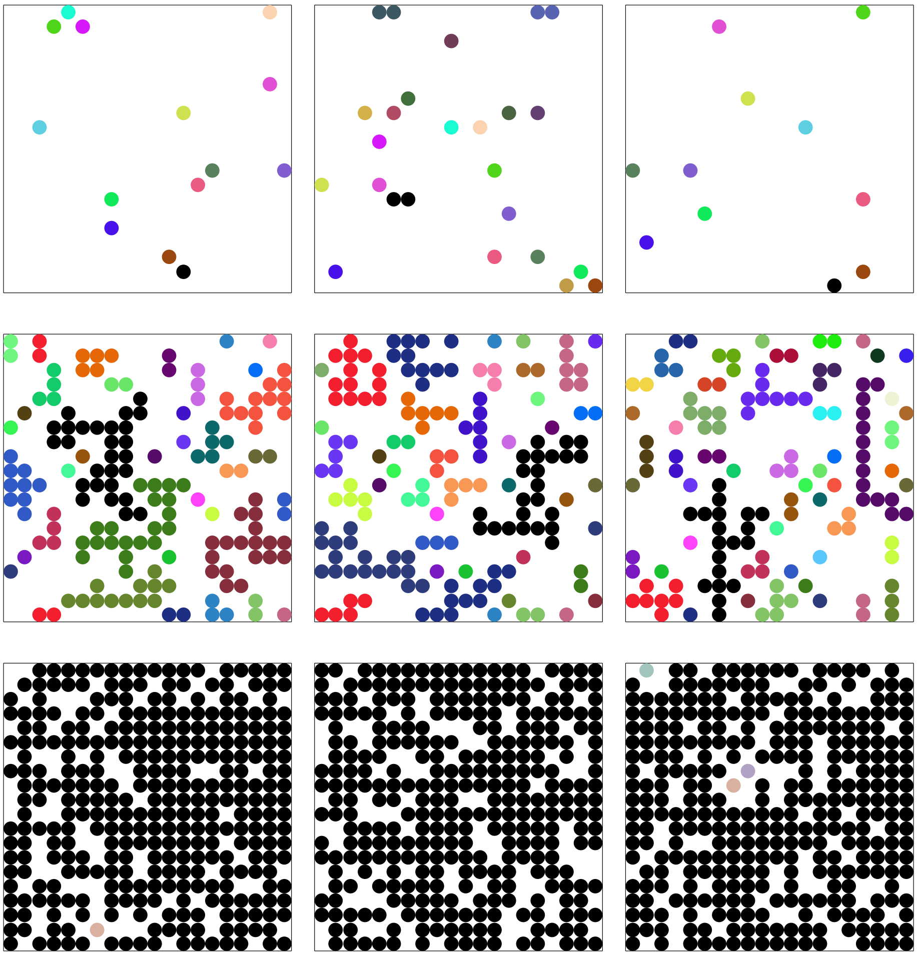
- For small p, mostly isolated trees; while for larger p we get increasingly larger clusters.

- Recall the **cluster definition** : trees that are nearest neighbors belong to the same cluster. If one tree in a cluster catches fire, all trees in the cluster burn down.
- Note that for $p = 0.8$ we essentially have a single cluster so large it “spans” the entire field. This is called a **percolating** cluster.
- From the point of view of fire, a percolating cluster is bad because it gives the potential for an extremely large fire forming.

5 Color coding the clusters

- It is helpful to visualize the clusters using some color coding. Rather than just marking each occupied plot with the same color, we mark trees that belong to the same cluster with a distinct color. The snapshots below are similar to the ones above, but with the clusters color-coded. Recall that if one tree in a cluster catches on fire, all trees in that cluster burn down.

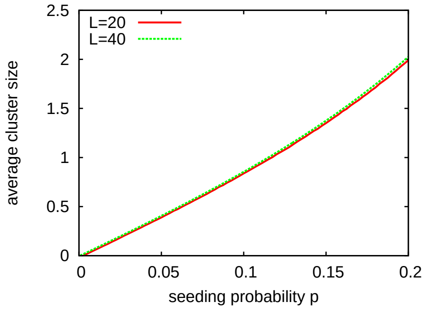
  1. White : empty space (no tree)
  2. Black is reserved to denote the largest cluster.

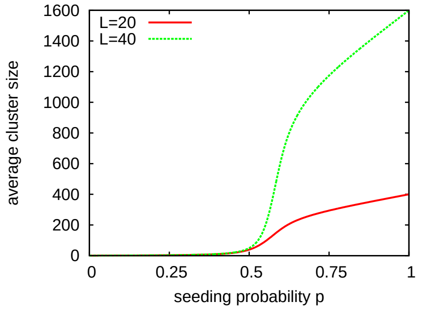


- Again, when p is small, we observe many small clusters. For intermediate p we still observe many clusters, but they are larger. For large p, we essentially see one system-spanning (percolating) cluster.

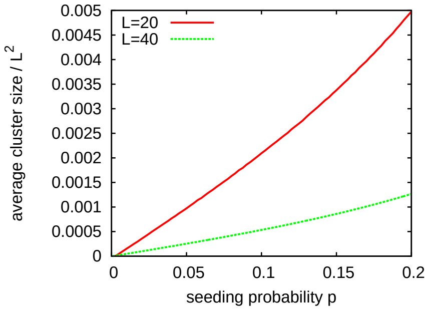
6 Measuring the average cluster size

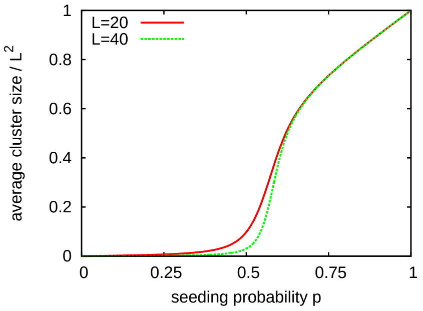
- For fire control, it is important to know the average cluster size as a function of p. Here, cluster size is defined as the number of trees that belong to the cluster.
- I have used a MC scheme to measure this. For a grid of size L, and some value of p of interest, many x-mas tree fields (samples, configurations) were generated. For each sample the clusters were identified, and the average cluster size was recorded. This process was then repeated for different p values and plotted in a graph. Note that the only difference between the below graphs is the scale of the horizontal axes!





- When p is small, the average cluster size does not depend strongly on L.
- But when p is larger, the rather striking finding is that the average cluster size depends strongly on L!
- This should not come as a surprise. When $p = 1$, all plots are occupied. That is, there is a single large cluster of size L^2. This explains the huge difference in cluster size at large p.
- Hence, we should not plot the average cluster size, but rather the average cluster size divided by L^2.

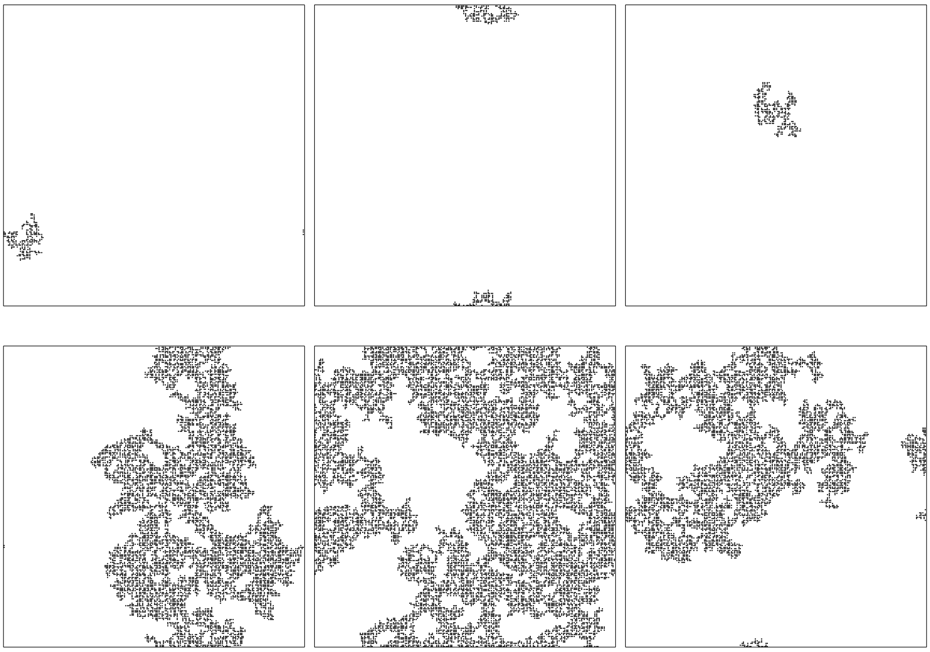




- Obviously, we now get good agreement in the limit $p \to 1$.
- However, the agreement in the limit of low p is gone! Here the curves are close to zero. So what is going on here?

7 The percolation transition

- When p is small, also the clusters are small (finite), and their size does not depend on L (see previous plot). Hence, dividing a finite number by L^2 will yield zero in the **thermodynamic limit** $L \to \infty$.
- We thus appear to have a phase transition driven by the occupation probability p, and with order parameter:
$$\Delta = \langle N \rangle / L^2,$$
where $\langle N \rangle$ denotes the average cluster size.
- The average is obtained as follows: (1) generate a x-mas tree field i. (2) Identify the clusters in that field and calculate the average size N_i. (3) Repeat over many M x-mas field realizations and finally calculate:
$$\langle N \rangle = (1/M) \sum_{i=1}^{M} N_i.$$
- An alternative order parameter would be to take the average size of the largest cluster: $\Delta = \langle N_{\rm max} \rangle$.
- The transition is as follows:
  1. When p is small, we have $\Delta = 0$.
  2. When p is large, we have $\Delta \to 1$.
  3. Above some special value $p = p_{\rm cr}$, Δ starts to become non-zero! This is where the transition occurs. Precisely at the transition, the percolating cluster starts to form → **Percolation Transition**.
  4. From the above graph we conclude that $p_{\rm cr} = 0.5 - 0.6$ or so. We cannot say much more from this graph due to strong finite-size effects around the transition.
- Interestingly, the transition is remarkably sharp. Below are some “typical” snapshots using $p = 0.50$ and $p = 0.59$ on a 200×200 system. In these snapshots, only the largest cluster is shown (as black). PBC are used.



- For $p = 0.5$ the largest cluster is still very small compared to the size of the system (and hence poses no serious fire risk). For $p = 0.59$, however, percolating clusters start to appear.
- Note that they have an interesting fractal structure. Indeed, if you plot the largest cluster at $p = p_{\rm cr}$ you get a fractal. However, you don't know $p_{\rm cr}$ yet!

8 Critical behavior

- Percolation is the standard example of a critical phenomenon.
- Critical phenomena are characterized by a diverging correlation length ξ.
- **Percolation**: For the percolation problem, you could take the size of the second-largest cluster as measure for the correlation length. When $p < p_{\rm cr}$, all clusters are small, and hence also ξ is small. When $p > p_{\rm cr}$, there is one huge (percolating) cluster, and so the second largest cluster will be small again (and hence also ξ). However, precisely at $p = p_{\rm cr}$, the correlation length diverges.
- In the vicinity of the transition, the correlation length diverges as a power law:

  $$\xi \propto |p - p_{\rm cr}|^{-\nu},$$

  with $\nu > 0$ a critical exponent.
- The order parameter also shows power law behavior:

  $$\Delta \propto (p - p_{\rm cr})^{\beta}, \qquad p > p_{\rm cr},$$

  with $\beta > 0$ another critical exponent.

- Also of interest are the fluctuations in the order parameter or susceptibility:

  $$\chi = \left(\langle N^2 \rangle - \langle N \rangle^2 \right) / L^d,$$

  which diverges as:

  $$\chi \propto |p - p_{\rm cr}|^{-\gamma},$$

  defining yet another critical exponent $\gamma > 0$.

- **Always remember** : power laws only hold in the vicinity of the transition, and in large systems:

  $$|p - p_{\rm cr}|/p_{\rm cr} \ll 1, \quad L \to \infty.$$

- I also mention the hyperscaling relation between critical exponents:

  $$2\beta + \gamma = d\nu,$$

  where $d = 2$ is the spatial dimension.

9 Critical behavior in finite systems

- Since the correlation length diverges $\xi \to \infty$ at the transition, it is clear that simulations "loose" the correct physics at some point because L is finite.

- **DANGER** : When $\xi > L$: finite-size effects! You will find that the results of your simulation depend systematically on L.

- Directly fitting finite-size simulation data to critical power laws is meaningless.
  → Rumor has it that this once lead a famous physicist to remark "The only good simulation is a dead simulation".

- Fortunately, finite-size effects can be understood. Hence, the $L \to \infty$ behavior is still accessible, albeit indirectly.
  → Finite-size scaling (FSS).

- FSS is an extremely rich topic. For today I want to present just one simple (but very powerful) example.

- Central in FSS is the "Ansatz" that the correlation length that you measure in a finite system **at criticality** scales proportional to the system size L, i.e.

  $$L \propto \xi.$$

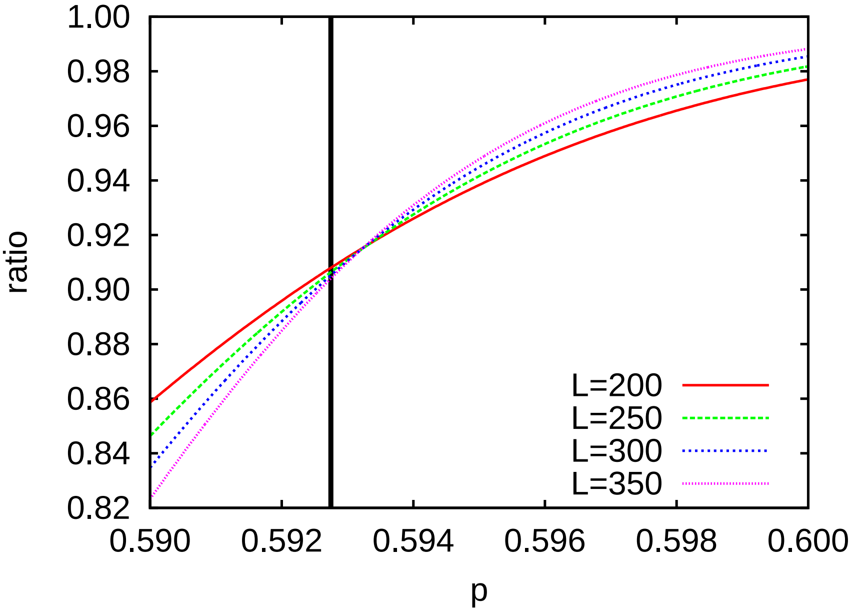
- With this simple idea, we can estimate $p_{\rm cr}$ simply as follows, following an idea originally put forward by Kurt Binder. To this end, I first introduce:

  $$t \equiv p - p_{\rm cr}.$$

- Hence: $\xi \propto t^{-\nu} \propto L \to t \propto L^{-1/\nu}$.
- Also: $\Delta \propto t^{\beta} \propto L^{-\beta/\nu}$.
- Also: $\chi \propto t^{-\gamma} \propto L^{\gamma/\nu}$.
- Consider now the ratio:

  $$\frac{\chi}{L^d \Delta^2} \propto \frac{L^{\gamma/\nu}}{L^d L^{-2\beta/\nu}} = L^{\gamma/\nu+2\beta/\nu-d} = \text{constant},$$

  by virtue of hyperscaling: $2\beta + \gamma = d\nu$.
- Hence, plotting the above ratio versus p, for several values of L, we expect to see an intersection point at $p = p_{\rm cr}$.
- This is the famous **Binder cumulant intersection method**.
- **Application** to percolation:



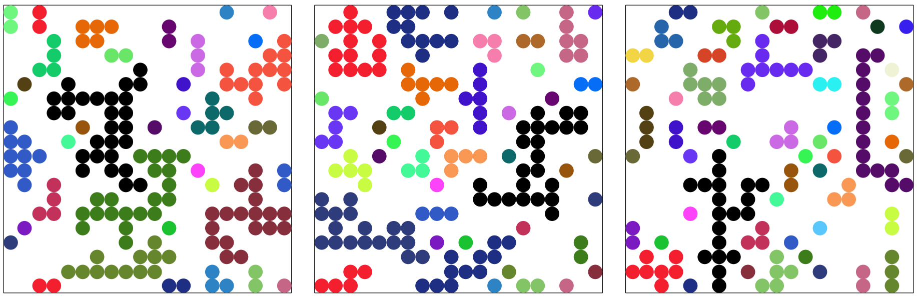
- From the intersection, $p_{\rm cr}$ can be quite accurately read-off. The vertical line marks the result of large-scale simulation.

1 Previous lecture

- Percolation, critical phenomena, and finite-size scaling.

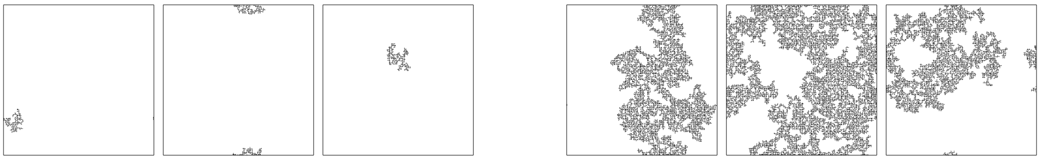
2 Summary of percolation

- Site percolation (of x-mas trees) $\rightarrow$ relevant for forest fires.
- **Model** : $L \times L$ lattice; each site contains a tree with probability p.
- Cluster formation : trees on nearest-neighboring sites belong to the same cluster.
- When $p = 0$ no trees at all $\rightarrow$ average cluster-size is zero.
- When $p = 1$ all sites occupied $\rightarrow$ average cluster-size is L^2.
- For intermediate p : average cluster size assumes some "in-between" value.



$p = 0.4$

- To obtain the average cluster size, one iterates over all clusters C in one configuration, followed by a sum over many different configurations M.
- At $p = p_{\rm cr}$ a system spanning cluster first appears $\rightarrow$ this marks the percolation transition!



$p = 0.50$ $p = 0.59$

- In the vicinity of $p_{\rm cr}$ the system exhibits critical behavior $\rightarrow$ power law scaling:

$$\Delta \propto t^{\beta} \; (t > 0, \text{and zero otherwise}), \qquad \chi \propto |t|^{-\gamma}, \qquad \xi \propto |t|^{-\nu},$$

with positive critical exponents obeying hyperscaling $2\beta + \gamma = d\nu$, with $d = 2$ the spatial dimension.

- **Note** : t is the relative distance from the critical point:

$$t \equiv \frac{p - p_{\rm cr}}{p_{\rm cr}}.$$

- Recall that the percolation order parameter Δ can be defined as the **average cluster size divided by the system area**...

- ...and χ as the order parameter fluctuation:

$$\chi \equiv L^d \left(\langle \Delta^2 \rangle - \langle \Delta \rangle^2 \right).$$

- **Problem** : critical behavior is only observed in the thermodynamic limit $L \to \infty$, while simulations deal with finite L.
  → need **finite-size scaling** (FSS) to bridge the gap.

- **FSS Ansatz** : $\xi \propto L$ → this yields a relation between t and L:

$$\xi \propto t^{-\nu} \propto L \to t \propto L^{-1/\nu}.$$

- This can be used to express Δ and χ in terms of L as well:

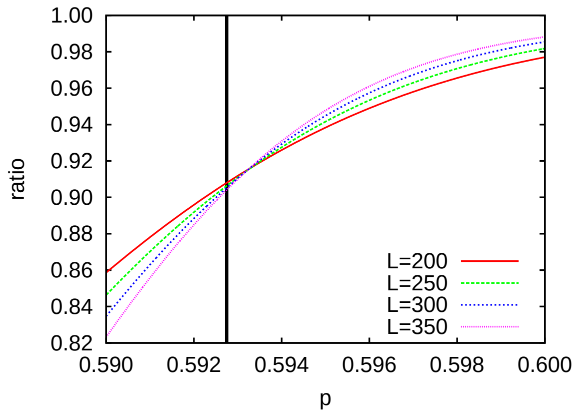
$$\Delta \propto L^{-\beta/\nu}, \qquad \chi \propto L^{\gamma/\nu}.$$

- Consider now the ratio:

$$\frac{\chi}{L^d \Delta^2} \propto \frac{L^{\gamma/\nu}}{L^d L^{-2\beta/\nu}} = L^{\gamma/\nu + 2\beta/\nu - d} = \text{constant},$$

  i.e. independent of the system size L, by virtue of hyperscaling.

- **Binder** : to locate $p_{\rm cr}$ one simply plots the above ratio versus p, for various system sizes L. The curves for different L should reveal an intersection point, from which $p_{\rm cr}$ can be read-off:



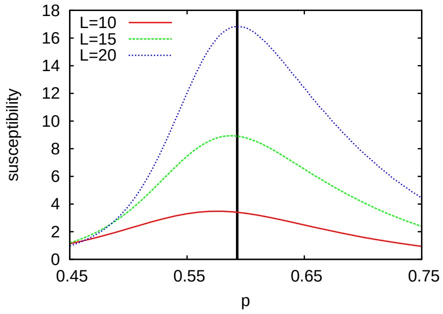
- The vertical bar marks $p_{\rm cr} \approx 0.59274621$ obtained in the high-resolution simulation: Phys. Rev. Lett. **85**, 4104 (2000). The data in the above plot were obtained overnight. The discrepancy is 0.04%.

Today

- What about the critical exponents? Can these be measured also?
- Yes, they can be!

3 Finite Size Scaling, Part II

- Using the Binder cumulant you can estimate the critical percolation threshold $p_{\rm cr}$. The method only assumes hyperscaling, but you don't need to actually know the critical exponents themselves.
- But what if you also want to measure the exponents, for example γ characterizing the divergence of the susceptibility: $\chi \propto |t|^{-\gamma}$.
- Since the simulation data are taken from finite systems, you do not observe the above power law directly.
  $\rightarrow$ So, what do you see instead in finite systems?
- **Application to percolation** : plotted is χ versus p for several values of L.



- In finite systems, χ does not diverge, but reaches a maximum. We say that the transition in the finite system is rounded.
- There are **two main effects** :
  1. The peak height grows profoundly with L.
  2. The peaks also shift to the right, but this is harder to see. Note that the peak positions are close to $p_{\rm cr}$ of the thermodynamic limit (vertical bar), but systematically below it! As L increases, the shift $|p_{\rm cr} - p_L|$ becomes smaller.
- **Note** : p_L is defined as the value of p where χ in a finite system of size L reaches its maximum.

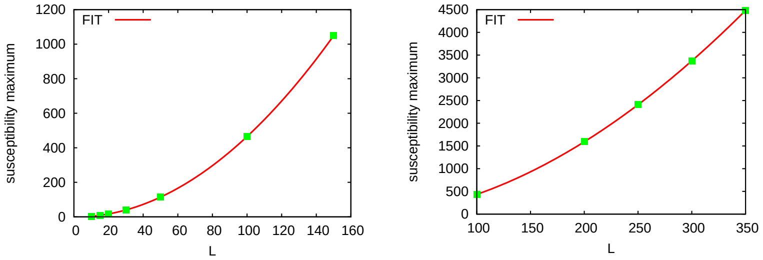
4 Understanding the finite-size effect : peak height

- The increase of the peak height can be understood from the finite-size scaling Ansatz $\xi \propto L$ which lead to:
$$\chi \propto L^{\gamma/\nu}. \qquad (1)$$

- **In words** : the maximum value of the susceptibility $\chi_{\max}$ should increase as a power law in L, with exponent γ/ν. This can be used to measure the exponent ratio γ/ν, by plotting $\chi_{\max}$ versus L, and then fitting to Eq.(1).

- For 2D percolation, the outcome is not very accurate, and quite sensitive to the range in L over which the fit is performed.

- **Results** : for small $L \rightarrow \gamma/\nu \approx 1.98$; for large $L \rightarrow \gamma/\nu \approx 1.82$.



- **Exact values** :
$$\nu = \frac{4}{3} \approx 1.33, \qquad \gamma = \frac{43}{18} \approx 2.39, \qquad \gamma/\nu \approx 1.79.$$

- So, in order for this method to yield reasonable results, you need pretty large systems. And even then the discrepancy can be quite large.

- The reason is that $L^{\gamma/\nu}$ is the leading scaling term. There will generally be **corrections to scaling**, which die out in large enough systems. Apparently, for percolation, the corrections to scaling are strong, and so we don't get overly good agreement in the system sizes considered here, but at least the trend is correct.

5 Understanding the finite-size effect : peak position

- We also found that the probability p_L where χ attains its maximum depends on L. Can we also understand this? Yes, but the answer is slightly more elaborate.

- Consider again the critical power laws:
$$\xi \propto t^{-\nu}, \quad \chi \propto t^{-\gamma} \rightarrow \chi \propto \xi^{\gamma/\nu}. \qquad (2)$$

- Remember that the above only holds in the thermodynamic limit $L \rightarrow \infty$ (infinite system size).

- What happens in a finite system of size L?
- As long as $\xi \ll L$, we are safe. The system is then "big enough" to capture the relevant physics, and we should recover Eq.(2). In other words, there is no explicit L-dependence in this case.
- In contrast, when $\xi \gg L$, we are in danger. The system is now "too small" to capture the relevant physics, and Eq.(2) gets replaced by:

  $$\chi \propto L^{\gamma/\nu},$$

  i.e. we pick up an explicit L-dependence.
- Hence, in a finite system, the susceptibility behaves as:

  $$\chi = \begin{cases} \xi^{\gamma/\nu} & \xi \ll L \\ ? & \text{intermediate range} \\ L^{\gamma/\nu} & \xi \gg L \end{cases} \tag{3}$$

- Mathematically, we can re-write this as:

  $$\chi = \xi^{\gamma/\nu} f(L/\xi), \tag{4}$$

  with f a so-called scaling function.
- The functional form of $f(x)$, with $x = L/\xi$, is not known. We only know its limiting behavior:

  1. When $x \gg 1$ ($\xi \ll L$), we must recover $\xi^{\gamma/\nu}$. Hence: $\lim_{x \gg 1} f(x) = \text{constant}$.
  2. When $x \to 0$ ($\xi \gg L$), we must recover $L^{\gamma/\nu}$. Hence: $\lim_{x \to 0} f(x) \propto x^{\gamma/\nu}$.
- Even though the scaling function f for intermediate values is not known, Eq.(4) contains all the information we need. But it still depends on ξ which is not convenient.
- Introduce $g(x) = x^{-\gamma} f(x^{\nu}) \to f(x) = x^{\gamma/\nu} g(x^{1/\nu})$. Substitute into Eq.(4) and use $x = L/\xi$ gives $\chi = L^{\gamma/\nu} g(x^{1/\nu})$. The argument of g can be further simplified using $\xi \propto t^{-\nu} \to x^{1/\nu} = (L/\xi)^{1/\nu} = L^{1/\nu} t$, leading to:

  $$\chi = L^{\gamma/\nu} g(L^{1/\nu} t),$$

  which is the fundamental scaling equation.

6 Using the scaling relation

- The scaling relation is extremely powerful, because it states that data from different system sizes are **identical**, provided they are scaled appropriately.

- To see how this rescaling needs to be done, rewrite the equation as:

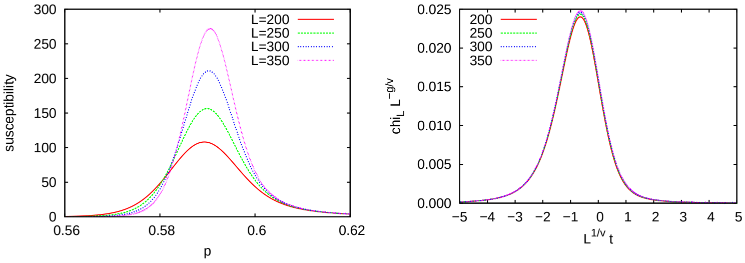
  $$\chi_L \times L^{-\gamma/\nu} = g(L^{1/\nu} t),$$

  where $t = (p - p_{\rm cr})/p_{\rm cr}$, and where the subscript is to remind you that χ refers to data obtained in a finite system.

- **In words** : take your raw data of χ versus p. On the y-axes, plot χ multiplied by $L^{-\gamma/\nu}$. On the x-axes, plot $L^{1/\nu} t$.

- If the data are plotted like this, the curves from different L should collapse. However, in order to make such a plot, the exponents and $p_{\rm cr}$ must be known.

- Results for 2D percolation, using the exact known exponents, and the estimate of $p_{\rm cr}$ reported earlier. Excellent collapse!



- If the exponents and $p_{\rm cr}$ are not known beforehand, they can be measured using trial-and-error until a data collapse occurs. An initial estimate of $p_{\rm cr}$ could be obtained from cumulant intersections.

- As an extra bonus, this method also enables us to measure the exponent ν.

7 Corollary : why does the peak shift?

- In the “raw” data, the susceptibility peak occurs at an L-dependent value p_L.

- In the scaling plot, the peaks have collapsed. This means that the susceptibility maxima at p_L occur at the same value of the scaling function argument.

- The scaling function argument equals $tL^{1/\nu}$. Hence, at the susceptibility maximum, it must hold that $tL^{1/\nu} = \text{constant}$.

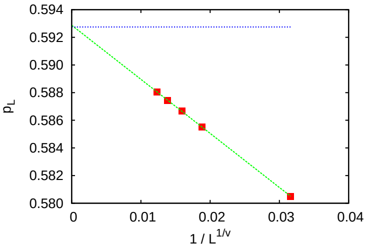
- Using that $t \propto p - p_{\rm cr}$, it follows that p_L is shifted from $p_{\rm cr}$ as:

  $$p_L = p_{\rm cr} + \frac{a}{L^{1/\nu}},$$

  with constant a.

- This offers an alternative method of measuring $p_{\rm cr}$. Simply plot p_L versus $1/L^{1/\nu}$, perform a linear fit → the intercept gives $p_{\rm cr}$.

- Result for 2D percolation, where the exact known value of ν was used.



- As before, if ν is not known beforehand, it can be fitted also, although this does require high-quality data.

8 Obtaining the order parameter exponent

- The derivation of the susceptibility scaling function carries over to the order parameter as well. Recall that $\Delta \propto t^\beta$, and so we simply need to replace $\gamma \to -\beta$.

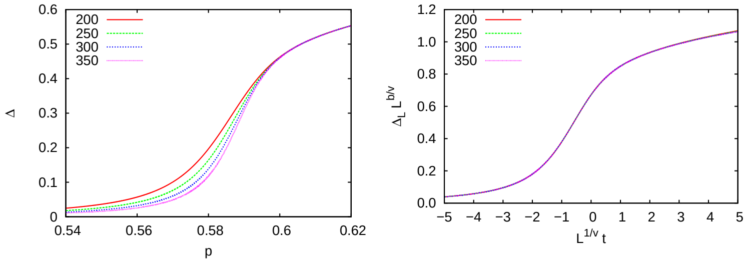
- This leads to:

$$\Delta_L \times L^{\beta/\nu} = h(L^{1/\nu} t),$$

  with another scaling function h.

- As before, $t = (p - p_{\rm cr})/p_{\rm cr}$, and the subscript is to remind you that Δ refers to the order parameter obtained in a finite system.

- This enables us to also obtain the order parameter exponent, see below for the 2D percolation result:



1 Summary of previous lecture

- Finite-size scaling (FSS) applied to the percolation transition.
- Introduced the method of **scaling plots**: at the critical point, data from different system sizes L "collapse", provided they are scaled appropriately.
$\rightarrow$ Application to the percolation susceptibility χ and order parameter Δ.
$\rightarrow$ This method also enables one to measure critical exponents, in addition to the critical percolation threshold p_{cr}.
- **Note** : scaling plots are general! All that is assumed are the usual power laws $\xi \propto t^{-\nu}$, $\Delta \propto t^{\beta}$, $\chi \propto t^{-\gamma}$, and the scaling "Ansatz" $\xi \propto L$.
- **In contrast** : the Binder cumulant method relies on **hyperscaling**.
- Examples of hyperscaling violation: mean-field systems, quenched random-field disorder! But this becomes rather technical.

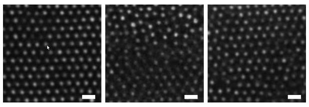
2 Today

- MC simulations of colloid-polymer (CP) mixtures. This will combine many of the concepts discussed in previous lectures.

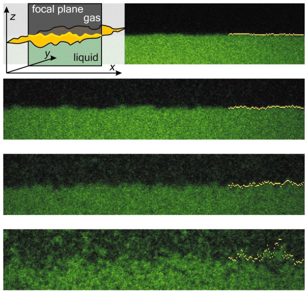
3 What are colloids?

- A colloid is a "blob" of matter with a typical size in the order of one micrometer, floating (suspended) in some solvent.
- **In physics** : so-called model colloids of silica are often used. These can be prepared in all kinds of shapes.
$\rightarrow$ typically spherical, rather monodisperse, and since they are made of silica, they closely resemble hard spheres!
- Colloids are used in physics because they are small enough to be "thermal", yet big enough to be visualized.
$\rightarrow$ by thermal is meant that colloidal systems are described by the Boltzmann distribution.

- Why are colloids interesting?
  → colloids behave somewhat like atoms, but on expanded length scales. Detailed information in experiments can be obtained.
  → single particle tracking using confocal microscopy.



Melting of a colloidal crystal (scale bar = 2μm).



Liquid-gas phase separation in a CP mixture.

- If you want to study the above phenomena in an atomic system, it will be very difficult to get such detailed information.

- **Bonus** : since colloidal systems are thermal, they are also well suited to be simulated with the MC methods described in this lecture.
  → This will be illustrated today using a mixture of colloids and polymers, undergoing a liquid-gas transition (ie, the second picture shown above).

4 Mixtures of colloids and polymers

- Before we can simulate a CP mixture, we need to understand how colloids and polymers approximately interact with each other.

- Since we are dealing with a binary mixture, there are three pair potentials that need to be specified:

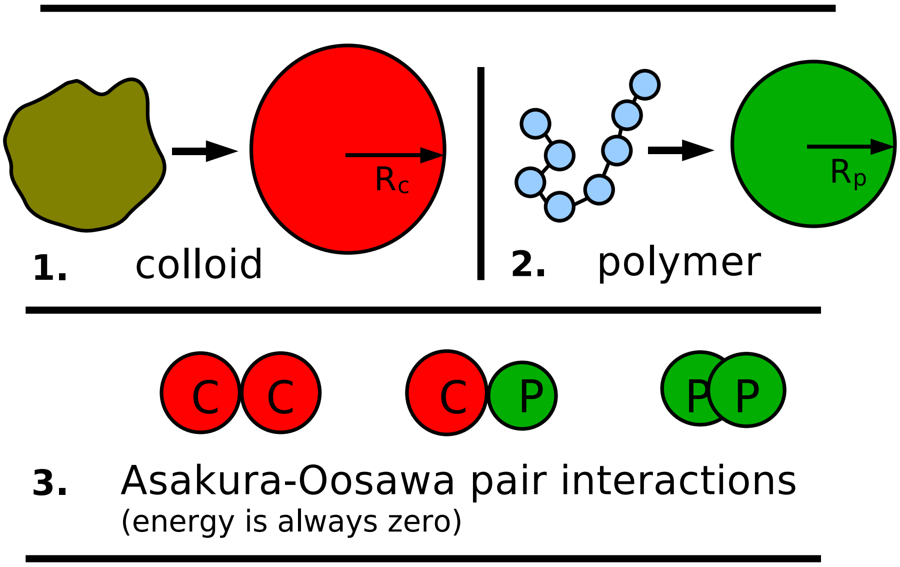
  1. colloid-colloid interaction.
  2. polymer-polymer interaction.
  3. colloid-polymer interaction.

- Colloids are approximately hard spheres, of radius R_C. Hence, the colloid-colloid interaction is simply the hard-core potential.

- But what about the polymers? Note that a polymer is a chain of monomers.

- Let us, for simplicity, assume good solvent conditions!

- **Good solvent** : the polymer swells, because it wants to maximize the contact area of the monomers with the solvent. In contrast, in bad solvent conditions, all the monomers would collapse onto a compact spherical object.
- A single polymer is thus essentially a rotating and diffusing line segment. The end segments can be conceived as "tracing out" a sphere in space. The radius of this sphere is called the radius of gyration R_{P}.
- In 3D, the cross-section between line segments is essentially zero, ie lines cannot overlap. Hence, in good solvent-conditions, the polymers hardly interact with each other. That is, they behave like an ideal gas.
  → The polymer-polymer interaction is thus always zero; the radii of gyration can overlap (interpenetrate) with no cost in energy.
- But what about the colloid-polymer interaction? Since the colloids are hard, the polymers cannot penetrate them. To a first approximation, this can be described by assuming that the colloid-polymer interaction is also a hard-core one, but with radius $R_{\mathrm{C}} + R_{\mathrm{P}}$.
- Using a picture:



- This set of interactions is known as the AO model, after Asakura and Oosawa. The only parameter is the colloid-to-polymer size ratio $q = R_P/R_C$.
- Note that the interactions are *a-thermal*, ie, temperature does not play a role. The equilibrium physics of the AO model is set by the colloid and polymer density.

5 Why a liquid-gas transition?

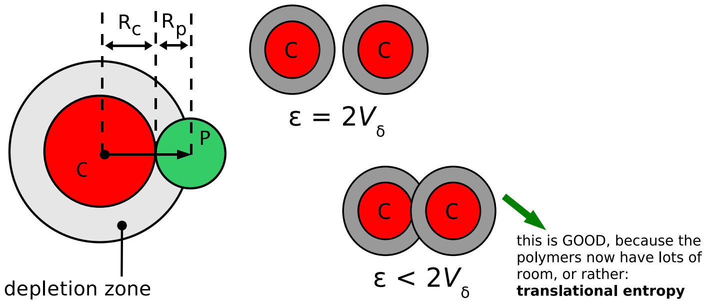
- In the lattice gas, we observed liquid-gas coexistence, provided we used the coexistence chemical potential μ_{COEX}, and the temperature was low enough.

- This can intuitively be understood from the pair potential, which features an attractive part. The dense liquid has a lower (translational) entropy than the gas, but at low T this is compensated by the lower energy of the liquid. Hence, liquid-gas coexistence below the critical temperature is possible.

- However, the energy in the AO model is always zero. What is the physical origin of the transition in this case?

- Answer : **translational entropy** $\rightarrow$ the depletion effect!

  1. Consider again the AO pair potentials.
  2. Each colloid is surrounded by a zone of radius $R_\delta = R_\text{C} + R_\text{P}$ and volume $V_\delta = 4\pi R_\delta^3/3$ which cannot contain any polymer centers $\rightarrow$ the depletion zone.


R_C
R_P
C
P
depletion zone
C
C
ε = 2V_δ
C
C
ε < 2V_δ
this is GOOD, because the polymers now have lots of room, or rather: **translational entropy**

  3. One isolated colloid excludes a volume $\epsilon = V_\delta$ for polymers. (by *exclude* I mean there is a space with volume ϵ where there cannot be any polymer centers). Two isolated colloids exclude twice this volume $\epsilon = 2V_\delta$. However, two colloids close together exclude less $\epsilon < 2V_\delta$ because their depletion zones overlap.
  4. By grouping together, the colloids themselves loose entropy, but they also create more entropy for the polymers. When the polymer density (or polymer chemical potential in a GC treatment) is high, this becomes favorable. Hence, as for the lattice gas, there is also an attraction, this time arising from entropy.
  5. This leads to a very similar phase diagram as for the lattice gas!
  6. Connection to the lattice gas: $\beta = 1/T \leftrightarrow \mu_P$ and $\mu_\text{COEX} \leftrightarrow \mu_\text{COEX,colloid}$.
  7. Since the interactions are short-ranged, we even expect it to be in the same universality class.

- Note that for the AO model, there is no obvious underlying symmetry, and so the coexistence chemical potential is *a priori* not known.

- **Aim** : to obtain coexisting phase densities using grand canonical MC.

6 GC Monte Carlo simulations of the AO model

- The standard GC simulation approach easily extends to the AO model, with perhaps some subtleties, namely: (1) we now have a mixture, and (2) the system is 3D and *off-lattice*.
- Use a cubic $V = L \times L \times L$ simulation box with PBC.
- The basic idea is to "populate" the box with colloids and polymers, in accordance with the GC Boltzmann probability:

  $$P_{\mathrm{GC}} \propto e^{\mu_C N_C + \mu_P N_P - \beta E},$$

  where the imposed chemical potentials are constant, but where the number of particles N_X fluctuates. Note that we have a binary mixture, and therefore two chemical potentials.
- We can construct MC moves in the usual way, using the concepts of ergodicity and detailed balance (DB).
- The easiest choice is single insertion and removal of particles. Since we now have a binary mixture, this approach requires **four** types of MC moves (for the single component lattice gas, only two moves were needed):

  1. Insert colloid at random location in the simulation box.
  2. Remove randomly selected colloid.
  3. Insert polymer at random location in the simulation box.
  4. Remove randomly selected polymer.
- This procedure is clearly ergodic.
- DB can be used to derive the accept probabilities:

  $$a(\mu \to \nu) = \min\left[1, \frac{g(\nu \to \mu)}{g(\mu \to \nu)} \frac{p_\nu}{p_\mu}\right].$$
- Recall that μ denotes the initial state, ν the final state, g the proposition probability, and p the desired target distribution that we wish to sample.
- Assume each move is attempted with equal probability.
- For **Colloid insertion** $\mu(N_C, N_P) \to \nu(N_C + 1, N_P)$ this yields:

  $$\begin{aligned} p_\mu &\propto e^{\mu_C N_C + \mu_P N_P - \beta E_\mu}, \quad p_\nu \propto e^{\mu_C (N_C+1) + \mu_P N_P - \beta E_\nu}, & (1) \\ g(\mu \to \nu) &\propto (1/4) \times (1/V), \quad g(\nu \to \mu) = (1/4) \times 1/(N_C + 1), & (2) \end{aligned}$$

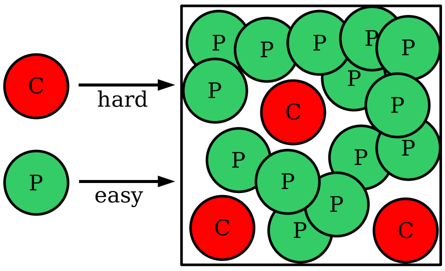
  leading to:

  $$a(\mu \to \nu) = \min\left[1, \frac{Ve^{\mu_C}}{N_C + 1} e^{-\beta(E_\nu + E_\mu)}\right].$$

  Note that if you put the colloid on top of another colloid or polymer $E_\nu \to \infty$, in which case you must reject the move.

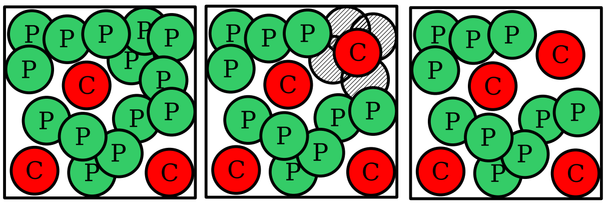
7 Toward cluster moves

- The derivation of the other accept probabilities is analogous, and will not be explicitly repeated here. The reason is that single particle insertions are expected to be inefficient for the AO model, so they are not recommended.
- To see this, consider a dense polymer phase. The polymers may overlap freely with each other, so it is easy to insert even more polymers. However, it is very difficult to insert a colloid in this system.

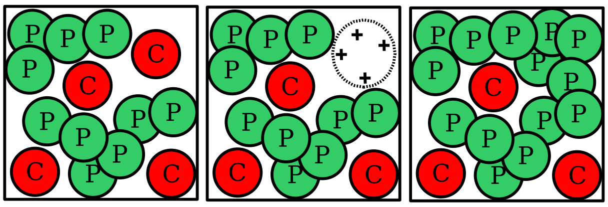


- Ideally, when you insert a colloid, you would like to remove the underlying polymers first. That way, you create some room, and are more likely to generate a state without overlaps. Of course, there could still be overlap with other colloids (this you must live with).
- So the cluster move that we envision should resemble:

1. **Colloid insertion** : $(N_C, N_P, E_\mu) \to (N_C + 1, N_P - n, E_\nu)$
   $\to$ Choose random location for colloid, remove all overlapping n polymers.



2. **Colloid removal** : $(N_C, N_P, E_\mu) \to (N_C - 1, N_P + n, E_\nu)$
   $\to$ Choose colloid for removal randomly.
   $\to$ Now comes the tricky bit. You must fill the depletion zone of the colloid with some number of polymers n. But how to choose n?
   $\to$ In order to enforce DB, you should choose n stochastically. Here, I choose n uniformly between $0 \le n < \Delta$. This number of polymers is then put randomly into the depletion zone of the removed colloid.
   $\to$ **note**: you must put the polymers randomly in the depletion zone of the colloid, i.e. in a sphere of radius $R_C + R_P$, which is bigger than the colloid itself!



- Of course, such a “swap” move is highly artificial, as nature would never move polymers in this way. But, in the framework of equilibrium MC this is perfectly acceptable, as long as we enforce ergodicity and DB.

8 Deriving the accept probability for the cluster move

- Using the cluster swap move, we only have **two** MC moves, namely colloid insertion and colloid removal. The insertion and removal of polymers is done automatically during the swap.

- We must still derive the accept probability. Recall the DB equation:

$$a(\mu \to \nu) = \min \left[1, \frac{g(\nu \to \mu)}{g(\mu \to \nu)} \frac{p_\nu}{p_\mu} \right].$$

- Assume that insertion and deletion swap moves are attempted with equal probability.

- Consider first **colloid insertion**, ie, a transition as follows:

$$\mu(N_C, N_P, E_\mu) \to \nu(N_C + 1, N_P - n, E_\nu).$$

- The forward selection probability is easy:

$$g(\mu \to \nu) = (1/2) \times (1/V).$$

- Also the target probabilities are easy:

$$p_\mu \propto e^{\mu_C N_C + \mu_P N_P - \beta E_\mu}, p_\nu \propto e^{\mu_C (N_C+1) + \mu_P (N_P - n) - \beta E_\nu}.$$

- Now the reverse selection probability $g(\nu \to \mu)$:

  1. Probability of choosing the colloid removal swap move: $1/2$.
  2. Probability of choosing the same colloid: $1/(N_C + 1)$.
  3. Since you removed n polymers, you should put the same number back. The probability that the reverse swap move does so is $1/\Delta$, since n is chosen uniformly randomly (**provided** $n < \Delta$, otherwise it is zero).
  4. Finally, you must choose the same n locations for those polymers in the depletion volume. This yields a term $n! \times (1/V_\delta)^n$, where the factorial arises because the particles are indistinguishable.

5. $g(\nu \to \mu)$ is the product of the above.

- substitution into DB gives:

$$P_{\mathrm{acc}}(N_C \to N_C + 1) = \begin{cases} 0 & \text{if } n \geq \Delta, \\ \min\left[1, \frac{e^{\mu_c} V}{\Delta(N_C+1)} \frac{n!}{(e^{\mu_p} V_\delta)^n} e^{-\beta(E_\nu - E_\mu)}\right] & \text{otherwise.} \end{cases}$$

- The derivation for the removal cluster move goes the same way. The result is: $P_{\mathrm{acc}}(N_C \to N_C - 1) = \min\left[1, \frac{\Delta N_C}{e^{\mu_c} V} \frac{(e^{\mu_p} V_\delta)^n}{n!} e^{-\beta(E_\nu - E_\mu)}\right]$

- **Implementation hint** : Note that the accept rate for colloid removal can already be calculated at the start of the move at the cost of just one random number n. That is, you could already do the accept/reject before looking at any of the particles.

9 Some results

- In the lattice gas, we fixed the chemical potential to the coexistence value, and for T low enough, we noticed that the particle number probability distribution became double-peaked.

- In a CP mixture, we must choose the colloid chemical potential to its coexistence value, and then for high enough polymer chemical potential, the colloid particle number distribution $P(N_C)$ becomes double-peaked.

10 Life demo on laptop

1. Program the cluster move for the AO model.

2. Set the colloid and polymer chemical potential μ_C and μ_P.

3. Measure the distribution $P(N_C)$ defined as the probability to observe a system containing N_C colloids.

- **Demo** using 3D box with PBC, $L = 8$, $q = 0.8$, $e^{\mu_p} = 0.766$, $e^{\mu_c} = 3.066$.

- We indeed observe a double-peaked distribution. As usual, the peak positions yield the colloid density of the gas and liquid phase.

11 Toward unbiased simulations

- The above demo was somewhat biased. The polymer chemical potential was chosen close to the critical point, and I already knew what the coexistence colloid chemical potential was. Close to the critical point, there is no free energy barrier due to interfaces, so standard Boltzmann sampling works just fine.

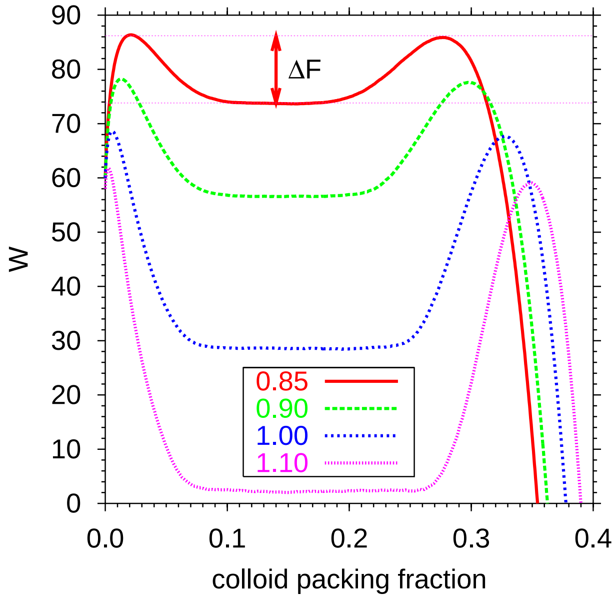
- **But** what to do if nothing is known beforehand? In that case, we need to “scan” the (N_C, μ_P) plane of the phase diagram in some clever way.
- For example with **Wang-Landau sampling**!
- To this end, we set $\mu_C = 0$, and we modify the accept rates with the DOS $g(N_C)$.
- **Colloid insertion** :

$$P_{\rm acc}^{WL}(N_C \to N_C + 1) = \begin{cases} 0 & \text{if } n \geq \Delta, \\ \min\left[1, \frac{V}{\Delta(N_C+1)} \frac{n!}{(e^{\mu_P} V_\delta)^n} e^{-\beta(E_\nu - E_\mu)} \times \frac{g(N_C)}{g(N_C+1)}\right] & \text{otherwise.} \end{cases}$$

- **Colloid removal** : $P_{\rm acc}^{WL}(N_C \to N_C - 1) = \min\left[1, \frac{\Delta N_C}{V} \frac{(e^{\mu_P} V_\delta)^n}{n!} e^{-\beta(E_\nu - E_\mu)} \times \frac{g(N_C)}{g(N_C-1)}\right]$
- Note that $g(N_C|\mu_P)$ is actually the “integrated” DOS introduced in previous lectures. That is, it does not depend on μ_C, but it does depend on μ_P.
- Initially, $g(N_C)$ is set to unity. WL sampling is used to iteratively improve the result.
- Once $g(N_C)$ is known, the particle number distribution follows:

$$P(N_C) \propto g(N_C) e^{\beta \mu_C N_C}.$$

- You then numerically tune μ_C until you see the double-peaks. Typically, you tune μ_C until the peaks have equal area under them. This can be done automatically using some root-finding scheme.
- Below are some typical example, where the $W = \ln P$ is shown:



(each distribution required one WL simulation)

- Note the flat region between the peaks! In this regime, we have two-phase coexistence, with interfaces. ΔF is related to the surface tension.

1 Announcements

- Today is the last lecture!

2 Summary of last lecture

- Colloid-polymer mixtures in **3D**.
- Phase separation provided the polymer density is high enough.
- Entropy effect!
- Coexistence between colloid-rich and colloid-poor phases.
- **Today** : interfaces!
  $\rightarrow$ How "wide" is the interface between the phases?
  $\rightarrow$ You can already see from the pictures of the experiment/simulation that the interface is not flat, but shows spatial fluctuations; these are called **capillary waves**

3 But first : ergodicity of the cluster move

- Colloids are inserted/removed one-at-a-time. But for each colloid you take out, you put back n polymers, drawn uniformly between $0 \leq n < \Delta$.
- Accept rate colloid insertion:

$$P_{\rm acc}(N_C \to N_C+1) = \begin{cases} 0 & \text{if } n \geq \Delta, \\ \min\left[1, \frac{e^{\mu_C}V}{\Delta(N_C+1)} \frac{n!}{(e^{\mu_P}V_\delta)^n} e^{-\beta(E_\nu - E_\mu)}\right] & \text{otherwise.} \end{cases}$$

- Colloid removal: $P_{\rm acc}(N_C \to N_C - 1) = \min\left[1, \frac{\Delta N_C}{e^{\mu_C}V} \frac{(e^{\mu_P}V_\delta)^n}{n!} e^{-\beta(E_\nu - E_\mu)}\right]$
- **Implementation hint**: Note that the accept rate for colloid removal can already be calculated at the start of the move at the cost of just one random number n. That is, you could already do the accept/reject before looking at any of the particles.
- Does the fact that Δ is finite not violate ergodicity?
- Say you run the program with $\Delta = 2$, so you insert at most one polymer for each colloid. Is the maximum polymer density then not restricted to $1/V_\delta$ as a consequence?
- No, because you could insert a colloid close to another polymer, and when this colloid is removed, you could put one polymer on top of the polymer already present one. Hence, the density is not restricted!

- However, for the purists, you could also select n from another distribution, with an extent from zero to infinity. Such as the Poisson distribution:

  $$P(n) = e^{-a} \frac{a^n}{n!},$$

  so you must replace $1/\Delta \to P(n)$ in the accept rates.

- A clever choice seems to be $a = e^{\mu_p} V_\delta$, because then a number of terms cancel.

- Yet another example of the flexibility of MC algorithms.

4 Colloid-polymer (liquid-gas) interfaces

- The interface separates a region of low colloid density, from one where the colloid density is high.

- Typically, if you approach the interface along a line perpendicular to the interface plane, you find that the density drops at the interface; we shall call this the z direction, and the density along it the density profile $\rho(z)$.

- The density drop is usually gradual, characterized by an effective width w.

- A mean-field calculation gives a hyperbolic tangent density profile:

  $$\rho(z) = \frac{\rho_L + \rho_G}{2} + \frac{\rho_L - \rho_G}{2} \tanh\left(\frac{z - z_0}{w}\right),$$

  with w the width, and z_0 the interface position.

- At z_0 the Gibbs construction is obeyed!

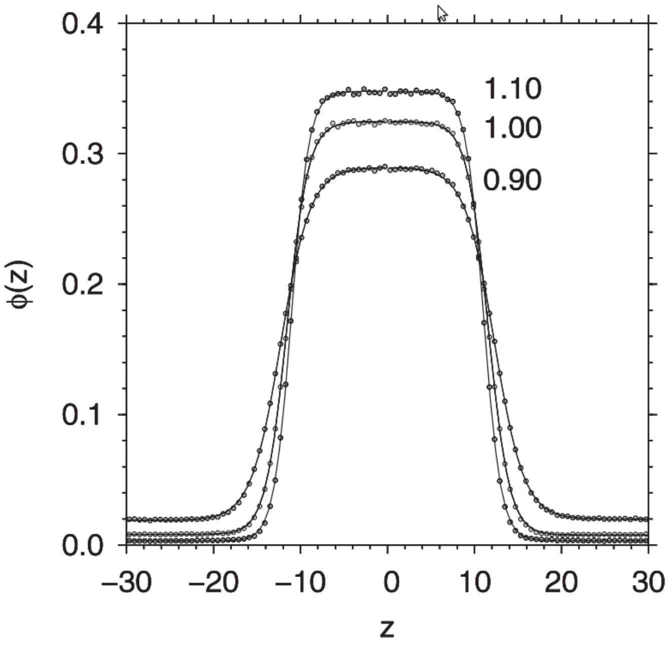
- Naively, we expect that w depends on the temperature. Close to the critical point, the interface is very broad.

5 Measuring the interface width in a simulation

- In a simulation, interfaces are readily generated. Take an elongated $L \times L \times D$ box, $D \gg L$ with PBC, high polymer fugacity. In that case, the system phase separates, and the interfaces form perpendicular to the longest edge of the box.

- To obtain the colloid density profile, divide the box into "slices". The number of colloids in each slice, divided by the volume of the slice, gives an estimate of the density in that slice. Doing this for all slices gives the profile.

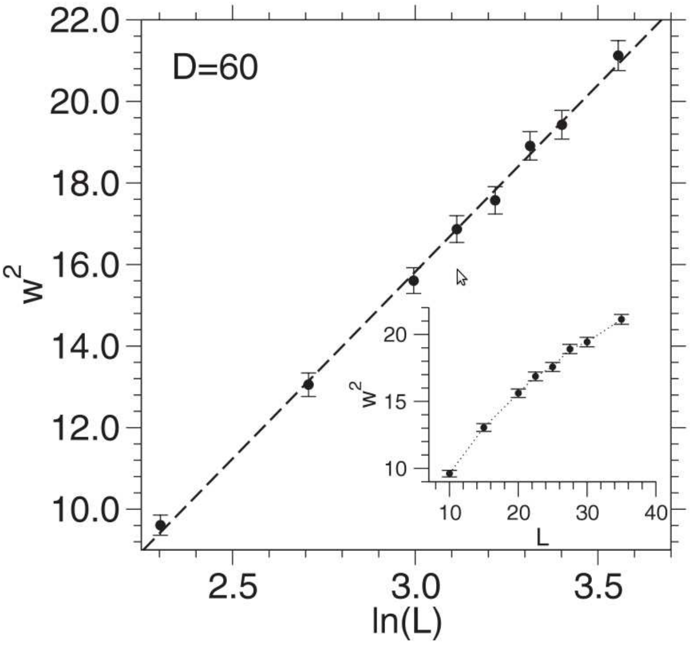
- **Application to AO model of CP mixtures** : graph of colloid packing fraction profiles, obtained in $L = 20$ and $D = 60$ boxes, at various polymer fugacities; recall that the polymer fugacity $\sim$ inverse temperature. Note that data such as these can easily be extracted with the code of the last lecture.



- Note that the profile at low polymer fugacity is indeed broader than at higher values, as expected.

- By fitting the profiles to the hyperbolic tangent equation, the width w can be obtained numerically.

- As it turns out, the width is ill-defined. Here is a graph of w versus the lateral system size L; w was obtained by fitting to the tanh function. The data apply to $z_p = 1.1$, i.e. the highest fugacity used above.



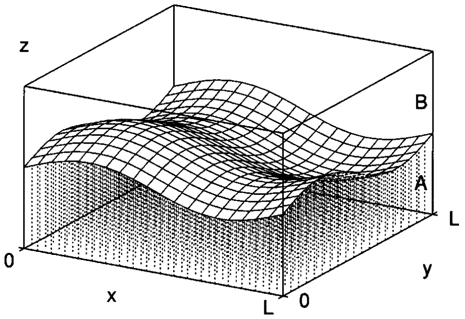
- The width depends on the lateral size L of the simulation box! It seems that $w^2 \propto \ln L$.
- **Hence**, the simple question:

  How wide is the CP interface?

  does not have a definite answer. Rather, the answer depends on L!
- Can we understand this result? Yes, we can!

6 Capillary Wave theory

- In CWT, is it is assumed that the interface position h is a smooth function of the lateral x, y coordinates.



- This obviously fails when you go to small length scales (of the order of single particles) but should be fine for large wavelengths.

- The energetically lowest state is to have a flat interface: $h(x,y) = c$. This is what would happen at zero temperature.

- However, at finite temperature, thermal excitations can cause the interface to be non-flat. It then has an area that exceeds that of the flat interface.

- In CWT, it is assumed that the free energy cost of having a non-flat interface is the **area increase × surface tension**.

- The area of a flat interface is L^2, that of a non-flat function $h(x,y)$ is:

$$A = \int_0^L \int_0^L \sqrt{1 + \nabla h \cdot \nabla h}\, dxdy,$$

  where $\nabla = (\partial_x, \partial_y)$ is the gradient operator.

- The CWT Hamiltonian is therefore:

$$H_{\rm cw} = \gamma \left(\int_0^L \int_0^L \sqrt{1 + \nabla h \cdot \nabla h}\, dxdy - L^2 \right).$$

- Assume small fluctuations $|\nabla h| \ll q$ and Taylor expand:

$$H_{\rm cw} = \frac{\gamma}{2} \int_0^L \int_0^L \left(\frac{\partial h}{\partial x}\right)^2 + \left(\frac{\partial h}{\partial y}\right)^2 dxdy.$$

  This is the CWT Hamiltonian.

- Now write $h(x,y)$ as a Fourier series:

$$h(x,y) = \frac{1}{L} \sum_{\vec{q}} h(\vec{q}) \exp\left(i\vec{x} \cdot \vec{q}\right),$$

  with $\vec{x} = (x,y)$ and $\vec{q} = \frac{2\pi}{L}(n,m)$.

- Substitute into Hamiltonian and after some algebra:

$$H_{\rm cw} = \frac{\gamma}{4} \sum_{\vec{q}} |h(\vec{q})|^2 q^2.$$

- This is a set of decoupled Harmonic oscillators, so we may use the equipartition theorem to calculate the thermal expectation values of the amplitudes:

$$\langle |h(\vec{q})|^2 \rangle = \frac{1}{\gamma \beta q^2}.$$

- The interface is a superposition of many waves. Each wave is characterized by a wave vector $\vec{q}$. Long wavelength excitations have a large amplitude; and the amplitude decays as a power law in q.

- The typical squared interface width can now be written as a 2D integral:

$$w^2 = \int d\vec{q} \langle |h(\vec{q})|^2 \rangle = \frac{1}{\gamma\beta} \int_{q_{\min}}^{q_{\max}} \frac{1}{q^2} \, 2\pi q \, dq \propto \ln\left(\frac{q_{\max}}{q_{\min}}\right).$$

- The smallest q-vector you can have is set by the size of your box; the maximum is set by the limit where CWT breaks down, say, the size a of a particle:

$$q_{\min} = 2\pi/L, \; q_{\max} = 2\pi/a \to w^2 \propto \ln(L/a).$$

- The squared width of the interface grows logarithmically with L, precisely what we saw in our simulation data.
- Consequently, it does not make sense to speak of “the” width of the interface. This is still a big controversy. How can you compare interface profiles of (mean-field) theories to simulation data?
- Note that, in 2D, the effect is even more severe:

$$w^2 = \frac{1}{\gamma\beta} \int_{q_{\min}}^{q_{\max}} \frac{1}{q^2} \, dq = \frac{1}{q_{\min}} - \frac{1}{q_{\max}}$$

  Here, we can safely take $q_{\max} \to \infty$, while for $q_{\min} = 2\pi/L$, leading to:

$$w^2 \propto L.$$

- That is: linearly with L; much “fuzzier” interfaces in 2D.

7 Final remarks

general principle

- MC method is a versatile method to study equilibrium statistical physics.
- As long as you have DB and ergodicity, it is allowed!

great flexibility on moves

- This allows many kinds of “dynamics”, such as: spin flips, particle displacements, particle deletion/insertion moves, cluster moves.
- This is just the tip of the iceberg. There are many more (and beautiful) MC moves out there.

smart data collection

- Use histogram reweighting to also get information about state-points (temperatures) where you did not simulate.
- Rejected moves can also give information (transition matrix)

biased sampling

- MC with standard Metropolis samples the Boltzmann distribution, but in some cases sampling from other distributions is better. For example, when you need to cross a free-energy barrier arising from interface formation.
- Wang-Landau algorithm; but again there are other choices here as well (umbrella sampling).

finite-size effects

- Since simulation boxes are small, the results you get might depend on L.
- Near phase phase transitions, interfaces.
- Rather than be discouraging, size-effects actually give useful information! As shown again today for interfaces.
- Finite-size effects may also become experimentally relevant.