多体問題の計算科学

Computational Science for Many-body problems

2023.4.25

#4: 古典モンテカルロ法の応用

Applications of classical Monte Carlo method

理学系研究科　大久保毅

Graduate school of science, **Tsuyoshi Okubo**

email: t-okubo@phys.s.u-tokyo.ac.jp

- This class is from 14:55 to 16:40 (105 min.)

Today

Classical

1st: Many-body problems in physics and why they are hard to solve

2nd: Classical statistical models and numerical simulation

3rd: Classical Monte Carlo method

4th: Applications of classical Monte Carlo method

5th: Molecular dynamics simulation and its applications

Quantum

6th: Extended ensemble method for Monte Carlo methods

7th: Tensor Renormalization group

8th: Quantum lattice models and numerical simulation

9th: Quantum Monte Carlo methods

10th: Applications of quantum Monte Carlo methods

11th: Linear algebra of large and sparse matrices for quantum many-body problems

12th: Large sparse matrices and quantum statistical mechanics

13th: Advanced algorithms for quantum many-body problems

Outline

- ~~Application to classical spin systems~~
  - ~~Local update by using Metropolis sampling~~
  - Global update as the cluster update
- Computational Science using Monte Carlo method
  - Typical procedure for practical calculations
  - Important tips to obtain reliable results
- Application and analysis in the case of critical phenomena
  - Simulation on Ising model
  - Finite size scaling
- Exercises (not a report)

Application to Classical spin system

Problems in local update

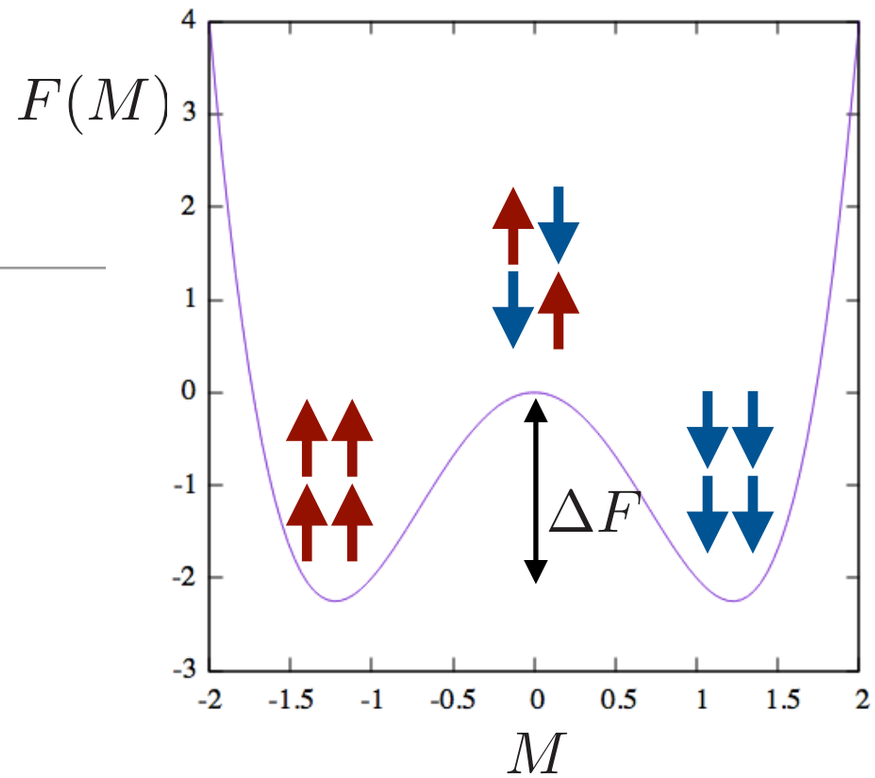
Sampling efficiency largely decreases for

1. Critical phenomena
   - Divergence of relaxation time: $\tau \propto |T - T_c|^{-z\nu}$
2. 1st order phase transition (phase coexistence)
3. Low temperature phase with discrete symmetry (e.g. Ising model)
   - Exponentially small probability to move other local minima: $\tau \propto \exp\left[\frac{\Delta F}{T}\right]$

Part of these difficulties can be reduced by using "global update".

Simultaneous change of spins in "large cluster"

Free energy landscape

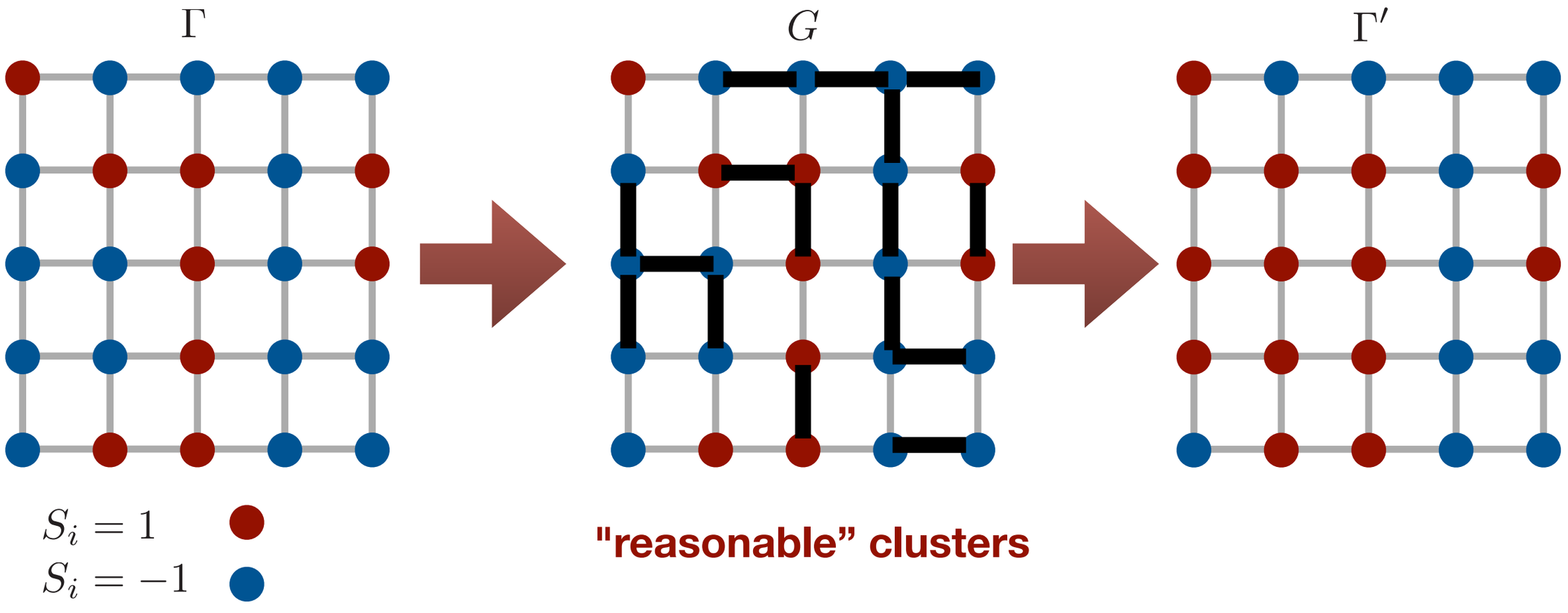


Cluster update

Cluster update method

Idea of cluster updates

- From a spin configuration Γ, we can define "reasonable" clusters G.
- When we "flip" all spins on a cluster G and make new configuration Γ', the free energy difference between Γ and Γ' is not so large.
- We can change the configuration drastically with higher probability.



How to make a cluster configuration?

Fortuin-Kasteleyn mapping (for Ising model)

Ising model

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j \ , S_i = \pm 1$$

P. W. Kasteleyn and C. M. Fortuin, J. Phys. Soc. Jpn, Suppl. **26**, 11 (1969).

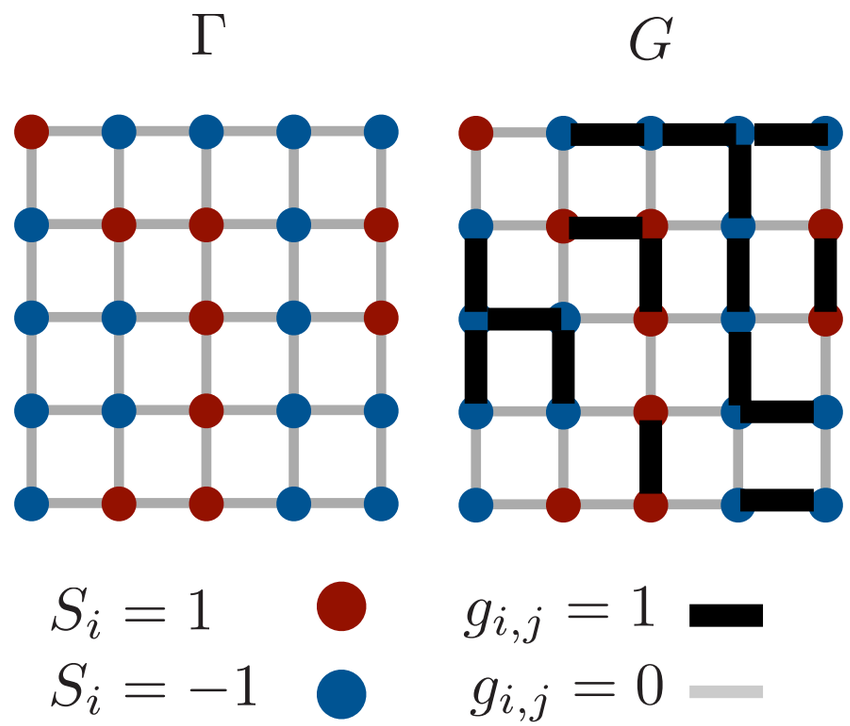
C. M. Fortuin and P. W. Kasteleyn, Physica **57**, 536 (1972).

$$e^{\beta J S_i S_j} = \underbrace{e^{-\beta J}}_{g=0} + \underbrace{\delta_{S_i,S_j}(e^{\beta J} - e^{-\beta J})}_{g=1} = \sum_{g=0,1} w(g, S_i, S_j)$$

$$Z = \sum_{\Gamma} e^{\beta J \sum_{\langle i,j \rangle} S_i S_j}$$

$$= \sum_{G} \sum_{\Gamma} \prod_{\langle i,j \rangle} w(g_{i,j}, S_i, S_j)$$

$G = \{g_{i,j}\}$ $\quad$ $\Gamma = \{S_i\}$



Markov chain in extended $(\mathbf{G}, \mathbf{\Gamma})$ space

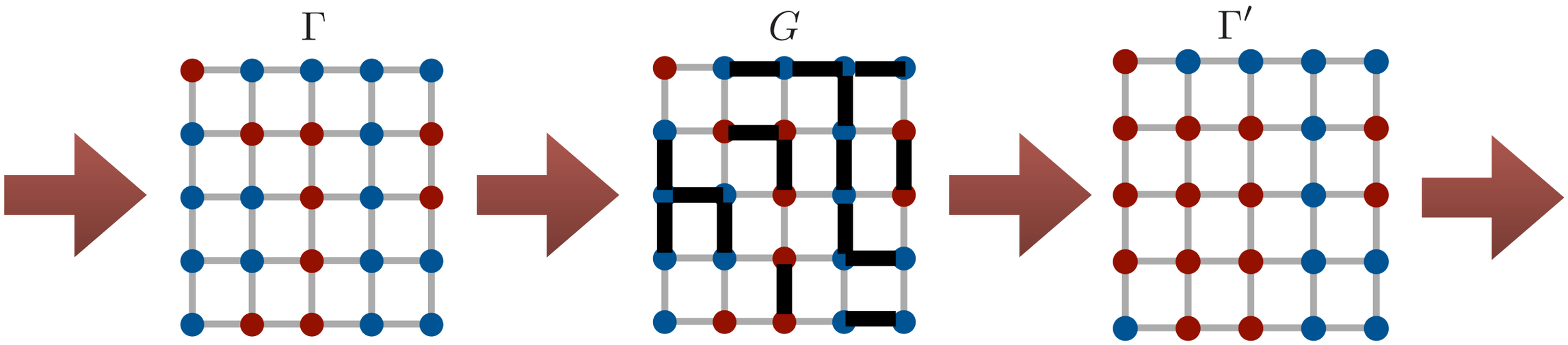
$$Z = \sum_{G} \sum_{\Gamma} \prod_{\langle i,j \rangle} w(g_{i,j}, S_i, S_j) = \sum_{G} \sum_{\Gamma} W(G, \Gamma)$$

We consider to update Γ and G alternatively:

$$\cdots \to (G_{t-1}, \Gamma_t) \to (G_t, \Gamma_t) \to (G_t, \Gamma_{t+1}) \to (G_{t+1}, \Gamma_{t+1}) \to \cdots$$

This update can be symbolically written as

$$\cdots \to \Gamma_t \to G_t \to \Gamma_{t+1} \to G_{t+1} \to \cdots$$



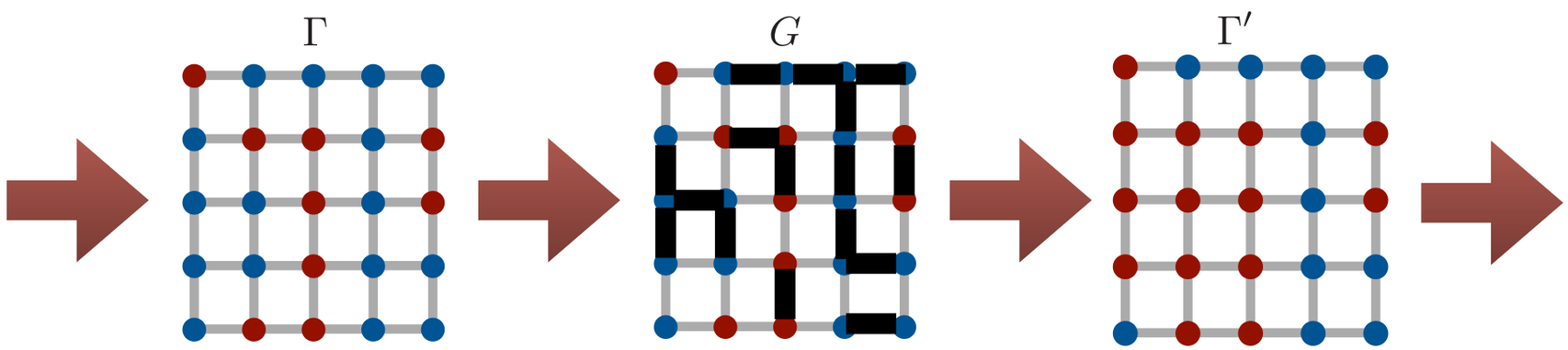
Markov chain in extended (G, Γ) space

We assign the transition probabilities as follows:

Transition probabilities (as like the heat bath)

$$\Gamma \to G' : \quad W_{(G,\Gamma)\to(G',\Gamma)} = \frac{W(G',\Gamma)}{\sum_{G''} W(G'',\Gamma)}$$

$$G \to \Gamma' : \quad W_{(G,\Gamma)\to(G,\Gamma')} = \frac{W(G,\Gamma')}{\sum_{\Gamma''} W(G,\Gamma'')}$$



The transition from spins to graph

$$\Gamma \to G' : \; W_{(G,\Gamma)\to(G',\Gamma)} = \frac{W(G',\Gamma)}{\sum_{G''} W(G'',\Gamma)}$$

$$W(G,\Gamma) = \prod_{\langle i,j\rangle} w(g_{i,j}, S_i, S_j) \quad \Rightarrow \quad \sum_{G} W(G,\Gamma) = \prod_{\langle i,j\rangle} \left[\sum_{g_{i,j}=0,1} w(g_{i,j}, S_i, S_j) \right]$$

$$W_{(G,\Gamma)\to(G',\Gamma)} = \prod_{\langle i,j\rangle} \frac{w(g_{i,j}, S_i, S_j)}{\sum_{g'_{i,j}=0,1} w(g'_{i,j}, S_i, S_j)} = \prod_{\langle i,j\rangle} w_{(S_i,S_j)\to g_{i,j}}$$

The transition probability becomes a product of the probability in each bond.

(All bonds are independent!)

$$w(g=0, S_i, S_j) = e^{-\beta J}$$
$$w(g=1, S_i, S_j) = \delta_{S_i,S_j}(e^{\beta J} - e^{-\beta J})$$

$$\sum_{g=0,1} w(g, S_i, S_j) = e^{\beta J S_i S_j}$$

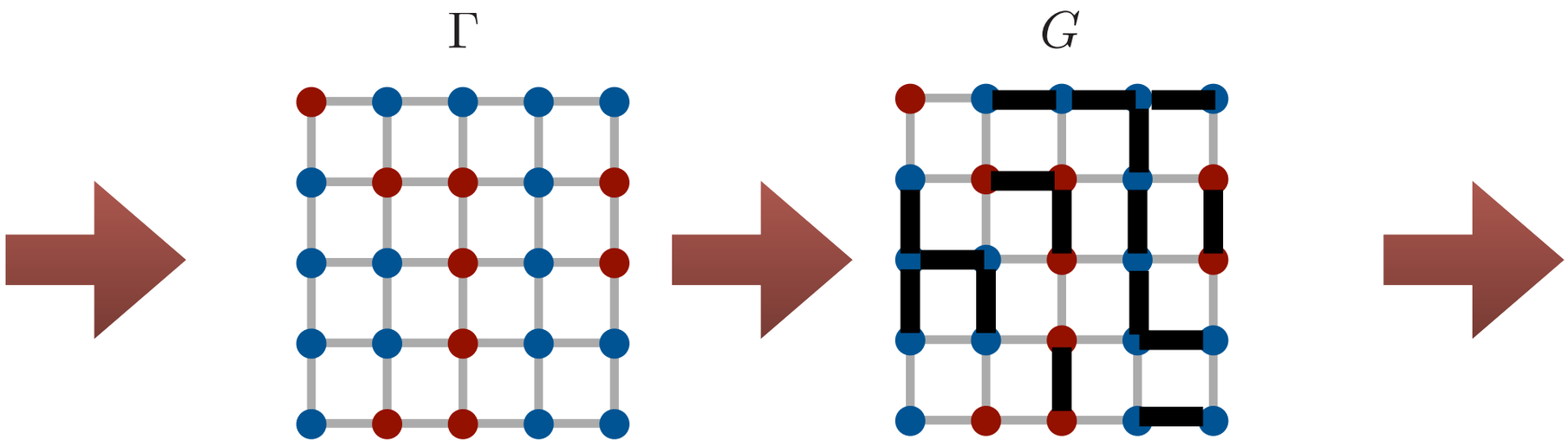
$$\Rightarrow$$

$(S_i = S_j)$	$(S_i \neq S_j)$
$w_{(S_i,S_j)\to 0} = e^{-2\beta J}$	$w_{(S_i,S_j)\to 0} = 1$
$w_{(s_i,S_j)\to 1} = 1 - e^{-2\beta J}$	$w_{(s_i,S_j)\to 1} = 0$

The transition from spins to graph

$(S_i = S_j)$	$(S_i \neq S_j)$
$w_{(S_i,S_j)\to 0} = e^{-2\beta J}$	$w_{(S_i,S_j)\to 0} = 1$
$w_{(s_i,S_j)\to 1} = 1 - e^{-2\beta J}$	$w_{(s_i,S_j)\to 1} = 0$

- Check all bonds independently.
- Assign g=0 or 1 following the above probability
  - If spins are antiparallel, always *g*=0, meaning "disconnected"
  - For parallel spins, we may assign *g*=1, meaning "connected"



The transition from graph to spins

$$W_{(G,\Gamma)\to(G,\Gamma')} = \frac{W(G,\Gamma')}{\sum_{\Gamma''} W(G,\Gamma'')}$$

$$W(G,\Gamma) = \prod_{\langle i,j\rangle} w(g_{i,j}, S_i, S_j)$$

$$w(g=0, S_i, S_j) = e^{-\beta J}$$

$$w(g=1, S_i, S_j) = \delta_{S_i,S_j}(e^{\beta J} - e^{-\beta J})$$

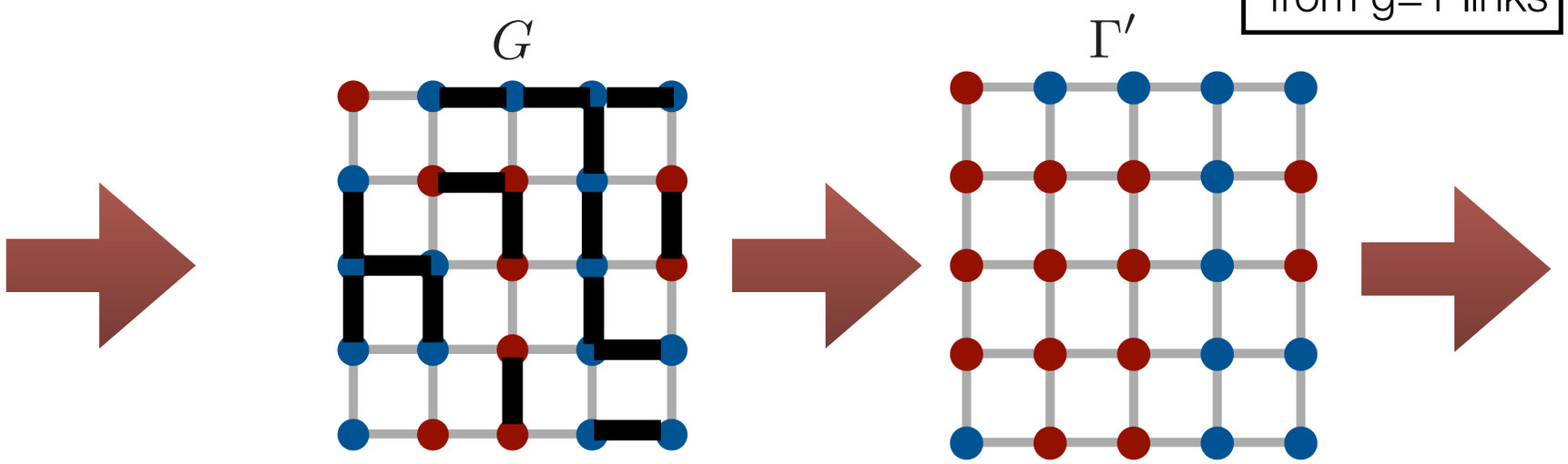
- All spins connected by g=1 will point in the same direction.
- When g=0 bonds separate spins, they will point in any direction with the same probability.

$$W_{(G,\Gamma)\to(G,\Gamma')} = \frac{W(G,\Gamma')}{\sum_{\Gamma''} W(G,\Gamma'')} = \prod_{C_j} P(\{S_i \in C_j\})$$

cluster formed from g=1 links

$$P(\{S_i \in C_j\}) = 1$$

(If all spin in cluster is pointing same direction)



Swendsen-Wang algorithm

Swendsen-Wang algorithm

R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 (1987)

Step 0: Prepare an initial state $\Gamma_0 = (S_1, S_2, \cdots, S_N)$

loop *t*

loop <i,j>

- if $S_i = S_j$, generate a random number
  - if $r \leq 1 - e^{-2\beta J}$ connects *i* and *j* ($g_{ij}=1$)

end loop <i,j>

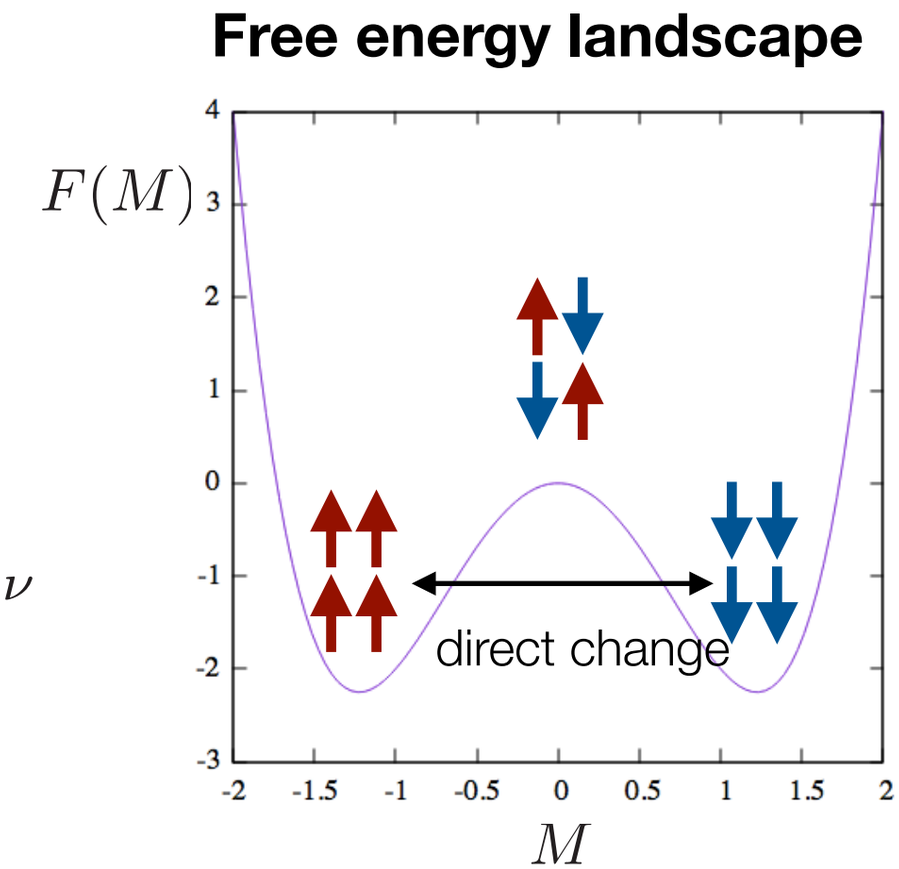
- Make clusters using algorithms (e.g. union find)
- Change spins on the same clusters simultaneously with probability 1/2 (using random number)

(Under a finite magnetic field, we need to modify the probability.)

Calculate $O(\Gamma_t)$

Merits of cluster update

1. For low temperature phase, the system easily moves to other minima.
   - Minima are related to the symmetry of the Hamiltonian.
2. For critical phenomena the dynamical critical exponent becomes much smaller.
   - Swendsen-Wang : $z \simeq 0 \quad \tau \propto |T - T_c|^{-z\nu}$
3. Graph representation is closely related to physics
   - e.g. Magnetic susceptibility in SW: $\chi = \frac{\beta}{N} \left\langle \sum_C \left(\sum_{i \in C} S_i \right)^2 \right\rangle$
   - By using observable based on graph, statistical error is largely reduced "Improved estimator"



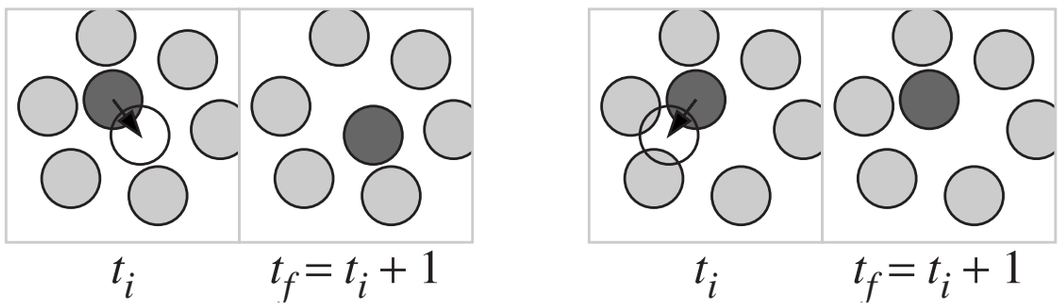
*Linear size of cluster ~ ξ

Event-chain Monte Carlo

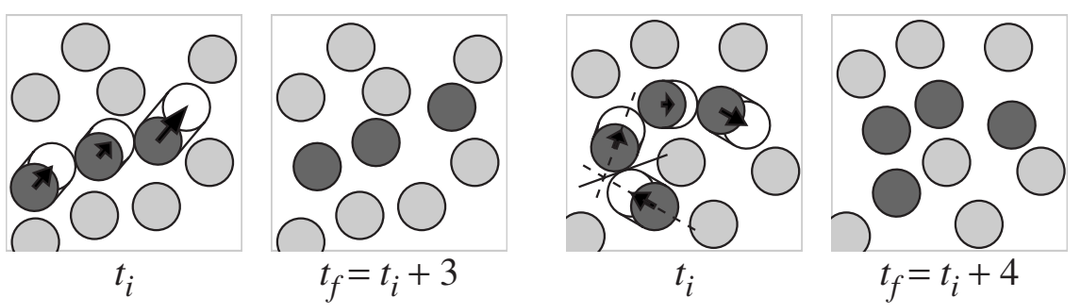
A "global" update for particle system (hard spheres)

E. P. Bernard, W. Krauth, and D. B. Wilson, Phys. Rev. E **80**, 056704 (2009)

local update

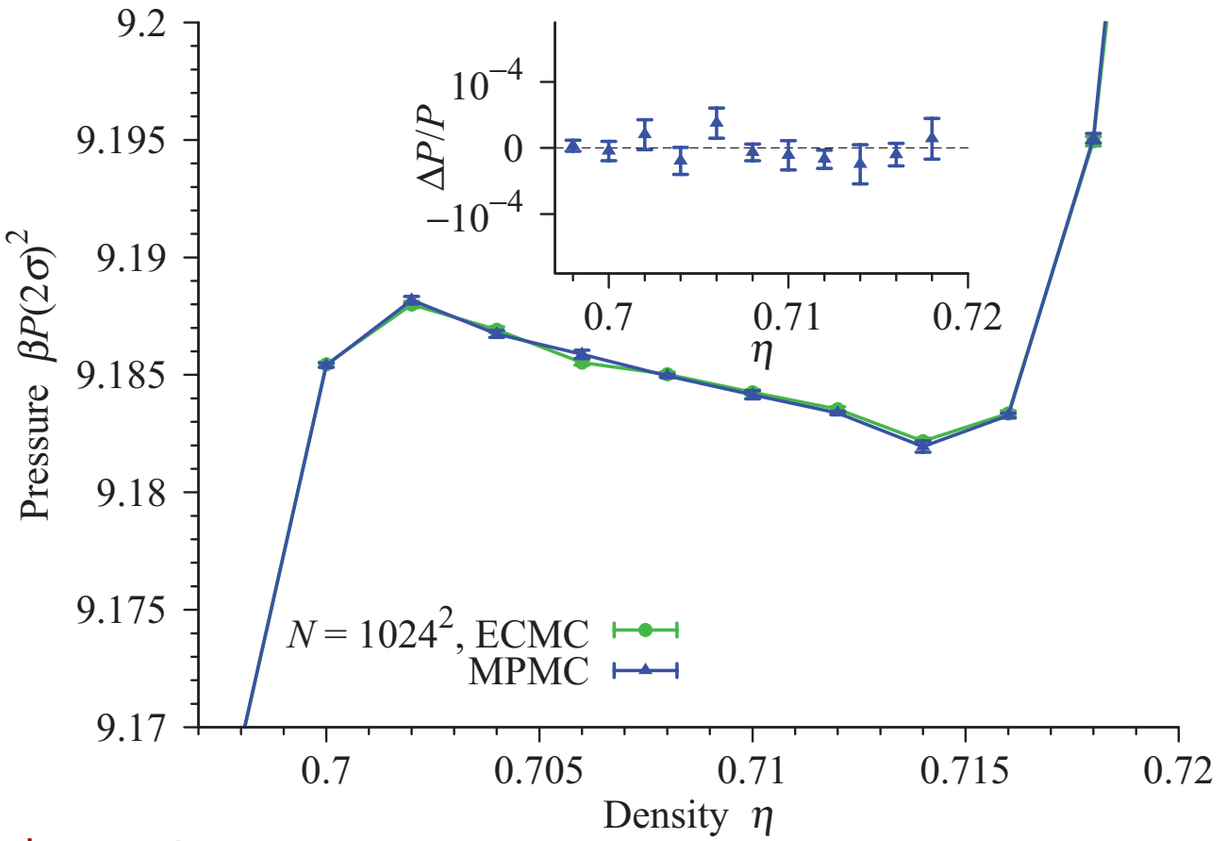


event chain update



Application to 2d melting

M. Engel *et al,* Phys. Rev. E **87**, 042134 (2013)



*Application of the event-chain MC to classical spin systems:

M. Michel, J. Mayer, and W. Krauth, Euro Phys. Lett. **112**, 20003 (2015).

Y. Nishikawa, M. Michel, W. Krauth, and K. Hukushima, Phys. Rev. E **92**, 063306 (2015).

Computational science using MCMC methods

Typical procedure for practical calculations

Remind: Purpose of Monte Carlo simulation

Typical purpose of MC simulation:

Calculate statistical averages over a probability distribution

We calculate the statistical average through the long-time average.

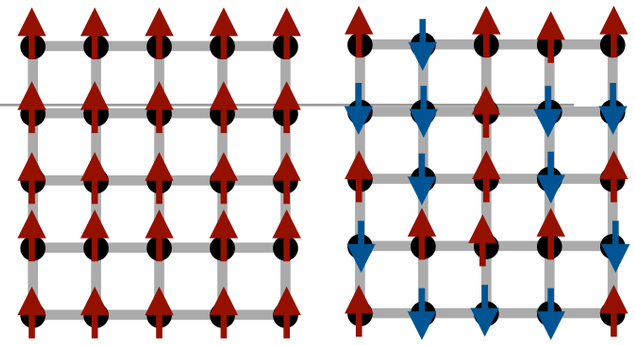
$$\langle O \rangle = \frac{1}{T} \sum_{t=t_0}^{T+t_0-1} \hat{O}(\Gamma_t)$$

Sequence of "states" generated by MCMC

$$\Gamma_0 \to \Gamma_1 \to \cdots \to \Gamma_t \to \Gamma_{t+1} \to \ldots$$

*We can also use MC simulations for cases where we first define the transition probability and do not know the steady state *a priori*.

Typical steps to obtain the final result



1. Prepare the initial state: Γ_0
   - It might be chosen randomly.
     - This usually corresponds to high temperature limit in the canonical ensemble.
   - It might be determined based on physical intuitions.
     - For example, the lowest energy state, a uniform state, ...

2. Warmup (Thermalization): Discard the first several steps t_0.
   - Usually, we are interested in the average over the steady state. Thus, to reduce the initial state dependence, we do not use the initial t_0 steps for the long-time average.

3. Time average: Take the sum along states generated along MCMC.
   - We can use the states for calculating the averages of several quantities.
   - We may use several sequences generated from different random numbers for calculating the averages.
     - They are statistically independent. However, you should consider the initial state dependence.

$$\langle O \rangle = \frac{1}{T} \sum_{t=t_0}^{T+t_0-1} \hat{O}(\Gamma_t)$$

Typical steps to obtain the final result (cont.)

4. Estimate statistical error:

- The time-averages obtained from MC simulations contain statistical errors. Thus, in order to evaluate reliability of the result, we should estimate the amplitude of statistical errors.
- The errors may depend on the observables.
  - For example, in the case of the canonical ensemble, the error of the energy is usually small, while that of the specific heat becomes larger.

5. Repeat above procedure by varying parameter:

- At least, you should check t_0 and T dependence.
  - If they are too short, the obtained results might be biased.
- You may also vary parameters of your model.
  - You can observe parameter dependence of the result.

Problem size dependence

When we increase the problems size N, the effect of a local change, *e.g.,* 1 spin flip, becomes smaller for the state of the system.

We should increase the time intervals between successive measurements so that the two state are expected to be different sufficiently.

To take this "trivial" effect into account, usually we define
1 MC step = trials over N moves (spin flips),
(in the case of local updates.)
This becomes a standard unit of MC simulation.

In this unit, the execution time for 1 MC step increases, typically, $O(N)$, in the case of "short-range" interactions.

* In the sample codes, we use this standard definition.

Important tips to obtain reliable results

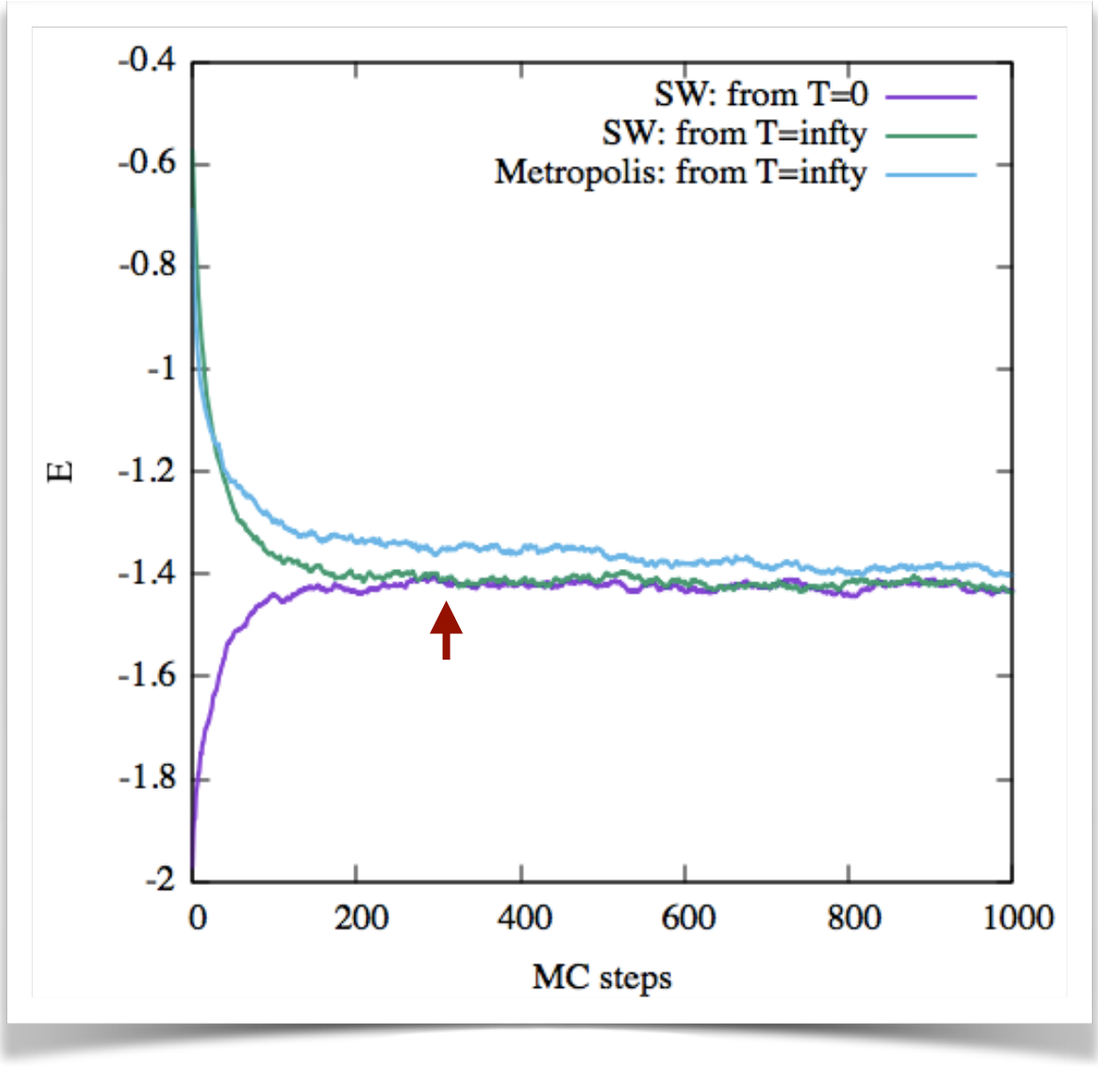
Convergence

In each calculation, we have to check the convergence.

If the correlation time is very long, obtained data (expectation values) might be biased from the initial state Γ_0.

Usual procedure:

- Discard several initial several MC steps (t_0)
- Change MC steps (T) and compare results
- Change initial state (and compare)
- Change algorithms (and compare)
- ….



Error estimation

We need to estimate the statistical errors.

$$\bar{A} \equiv \frac{1}{T}\sum_{t=1}^{T} \hat{A}(\Gamma(t))$$

Standard error: $\epsilon^2 = \langle \bar{A}^2 \rangle - \langle \bar{A} \rangle^2 \quad \epsilon \propto \sqrt{\frac{1+2\tau}{T}}$

Maximum likelihood estimation for standard error

Prepare "independent" M samples for $\bar{A}$: $\{\bar{A}_1, \bar{A}_2, \cdots, \bar{A}_M\}$

$$\sigma^2(M) = \frac{\frac{1}{M}\sum_i \bar{A}_i^2 - \left(\frac{1}{M}\sum_i \bar{A}_i\right)^2}{M-1}$$

$$\lim_{M\to\infty} \sigma^2(M) = \epsilon^2$$

Make "error bar" based on σ, and use it for data analysis.

*Independent data may obtained from

- Very long single simulation and "binning"
  - Notice the correlation time
- Samples from different random numbers

Error estimation (Cont.)

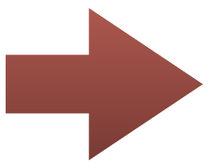
Please notice that the data might contain systematic error.

- Due to, for example, initial conditions and short observation time

Typical example: increase of the correlation time

- When we change the model parameter, the correlation time may drastically increase. In such case, if you use same observation time (t_0, T), the memory of the initial state affects the final result much more. → Biased result.

In general, the estimation of systematic errors are mote difficult.



- Please perform simulations with different conditions, and compare the result.
  - Longer simulations, different initial conditions, different algorithms, ...

Application and analysis in the case of critical phenomena

Example: Application to critical phenomena

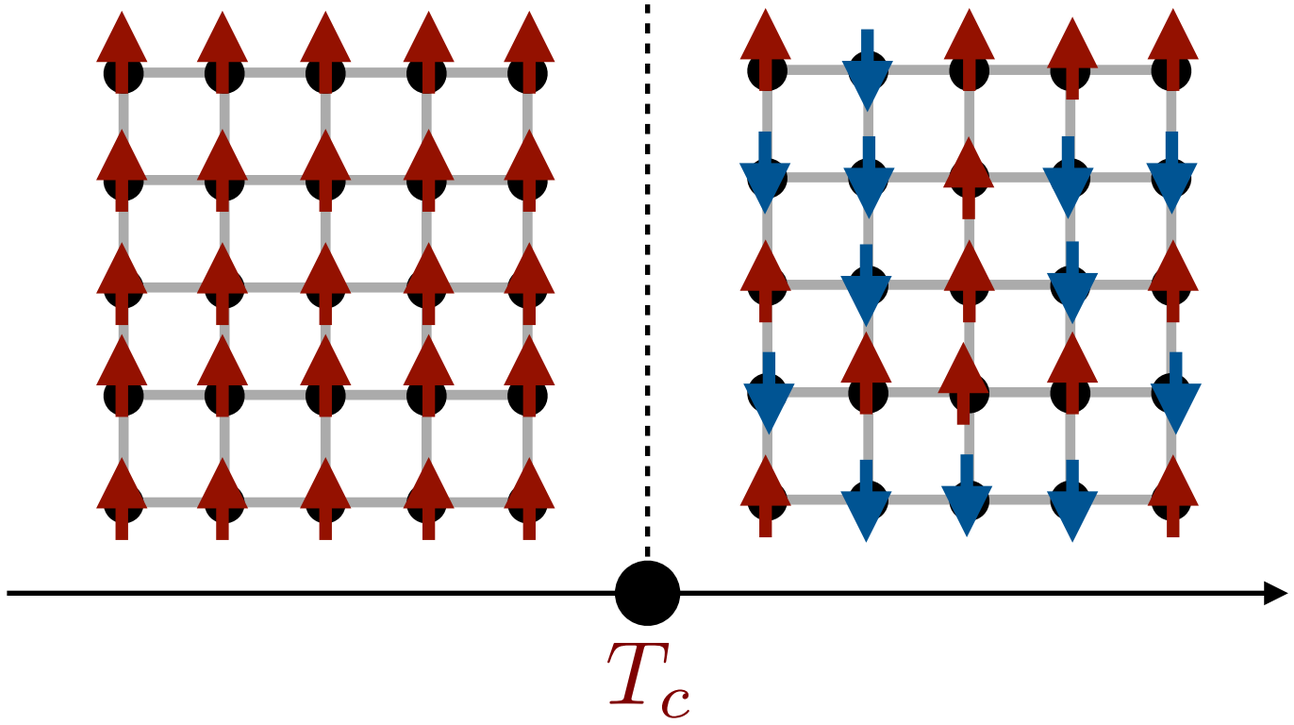
- Square lattice Ising model

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j$$

  - Continuous phase transition at $T=T_c$

$$T_c/J = \frac{2}{\ln(1+\sqrt{2})} = 2.26918531\ldots$$

  - $T > T_c$: Paramagnetic
  - $T < T_c$: Ferromagnetic

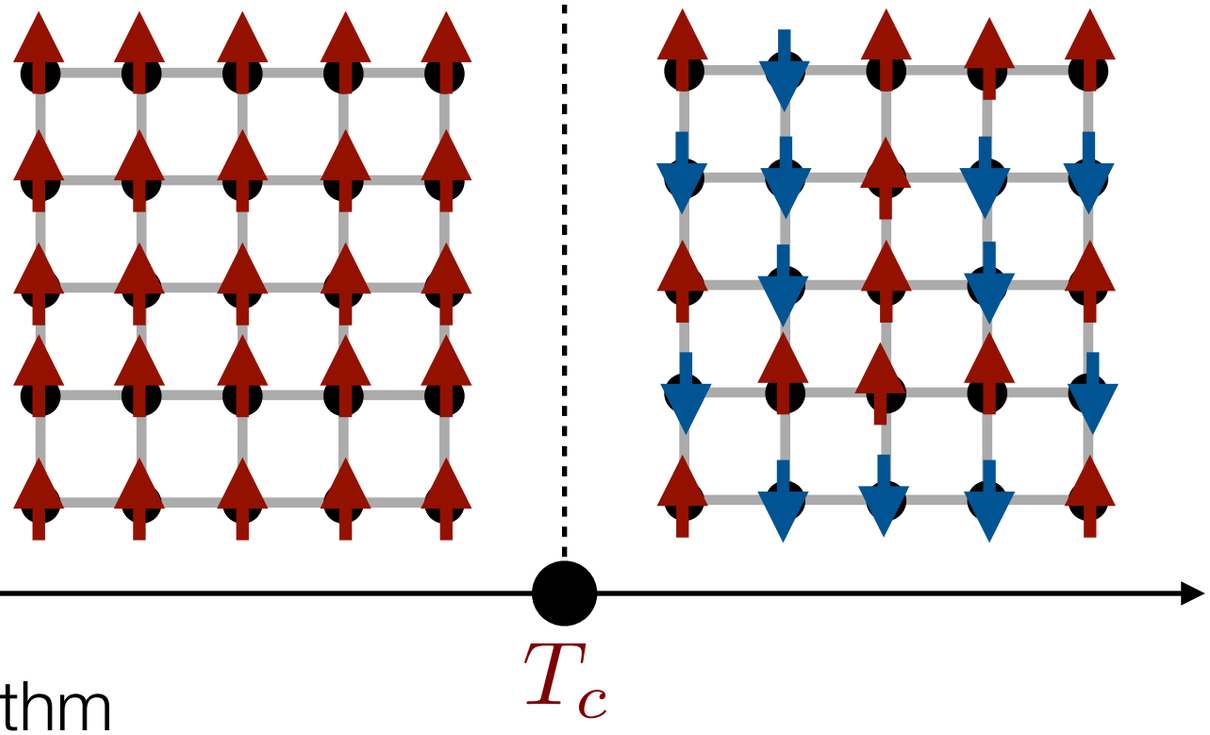


- Monte Carlo Simulations
  - Using spinmc in ALPS: Simulator for classical spin system by MCMC
    - http://alps.comp-phys.org/mediawiki/index.php/Main_Page
    - Unfortunately, the ALPS web server is down, and it is not available now.
  - Using sample code distributed in this lecture

Simulation setup

$$T_c/J = \frac{2}{\ln(1+\sqrt{2})} = 2.26918531\ldots$$

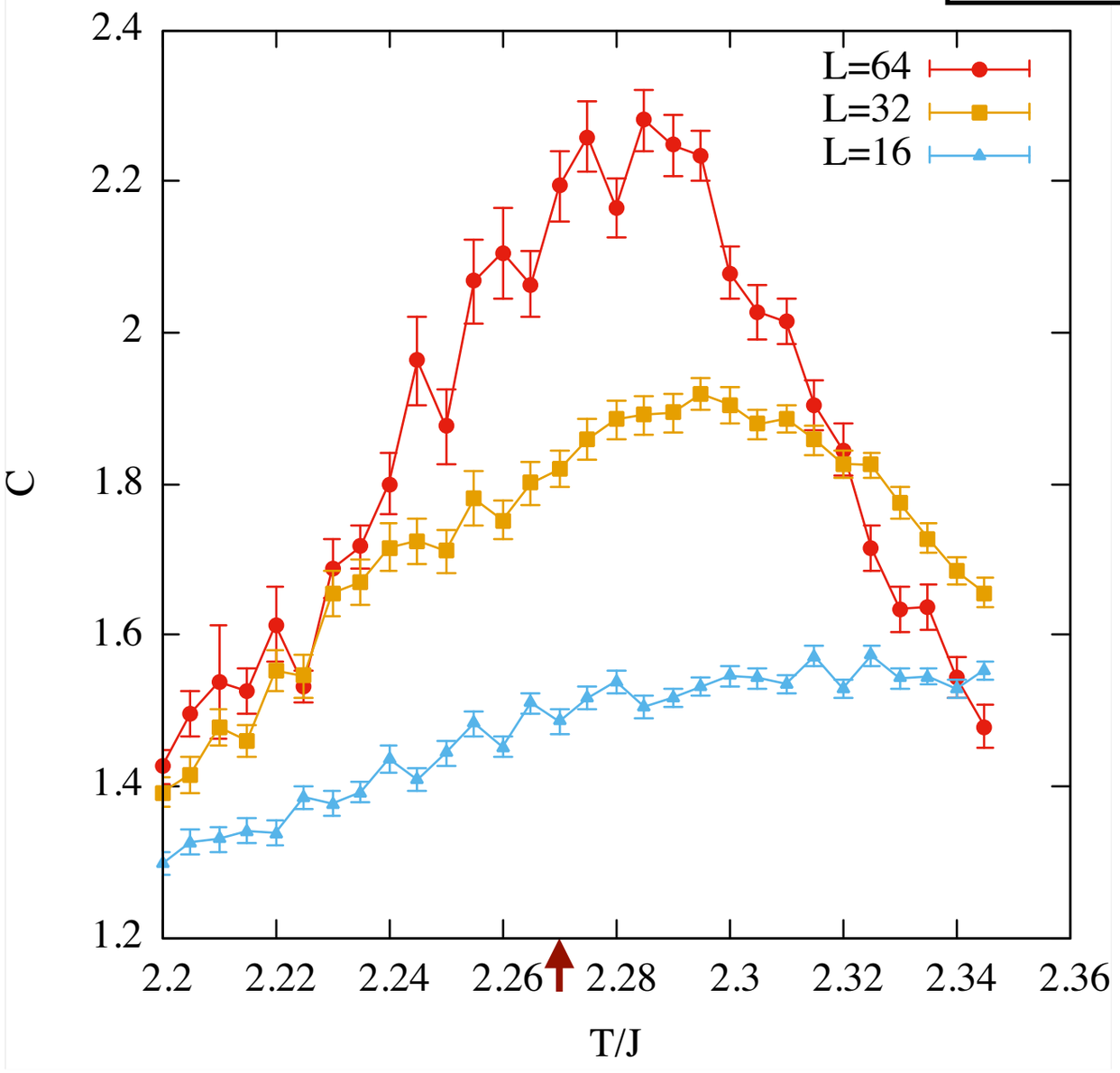
- $L \times L$ Square lattice Ising model
  - $L = 16,32,64$
  - Perform MCMC around T_c
    - $2.20 \leq T/J \leq 2.35$
  - Algorithms
    - Local update with Metropolis algorithm
    - Cluster update with Swendsen-Wang algorithm
- MC Steps
  - For thermalization: 10,000 MC steps
  - For observation: 100,000 MC steps



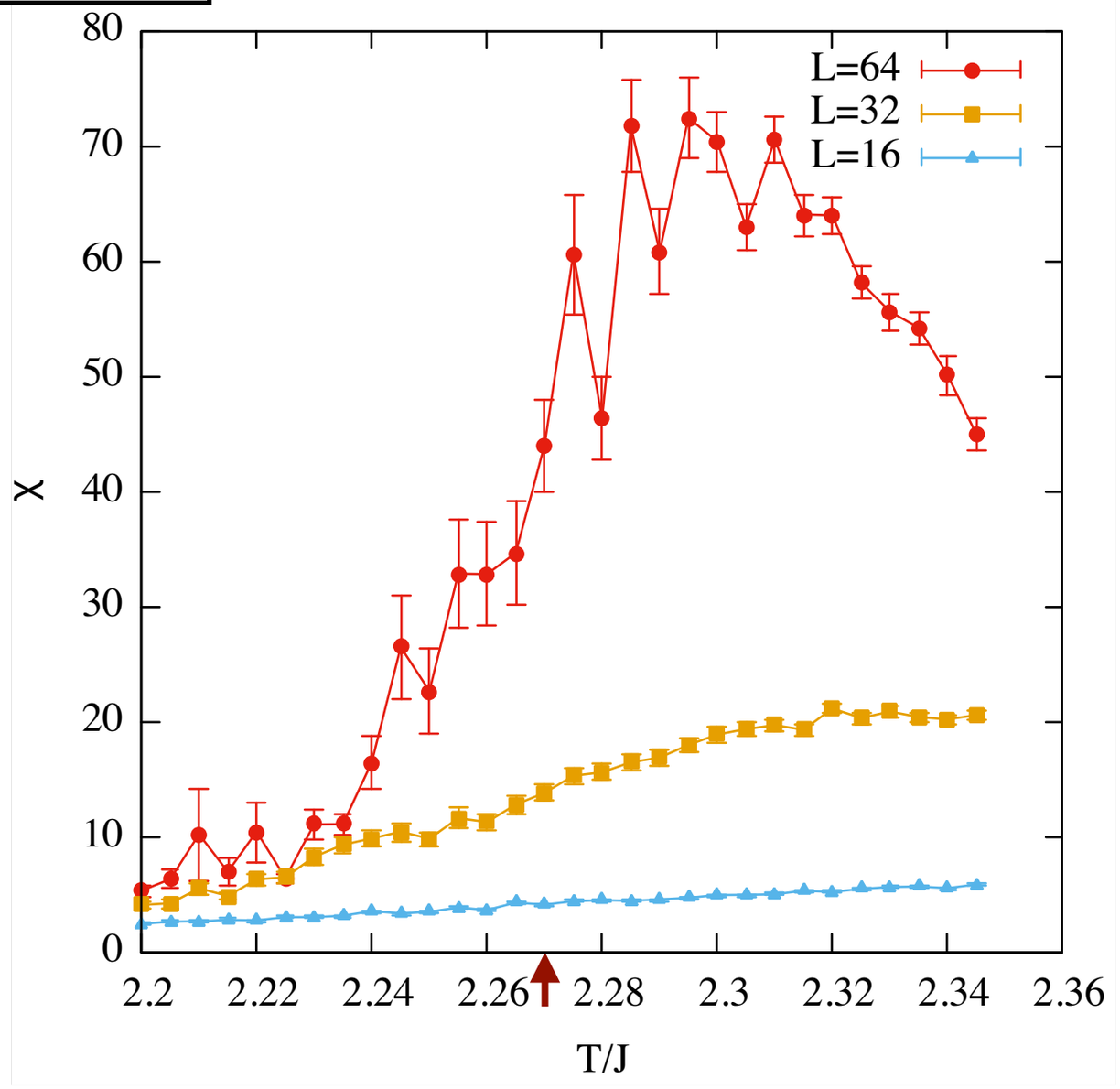
Calculated data (Metropolis)

$T_c/J \simeq 2.269$

Specific heat



Susceptibility



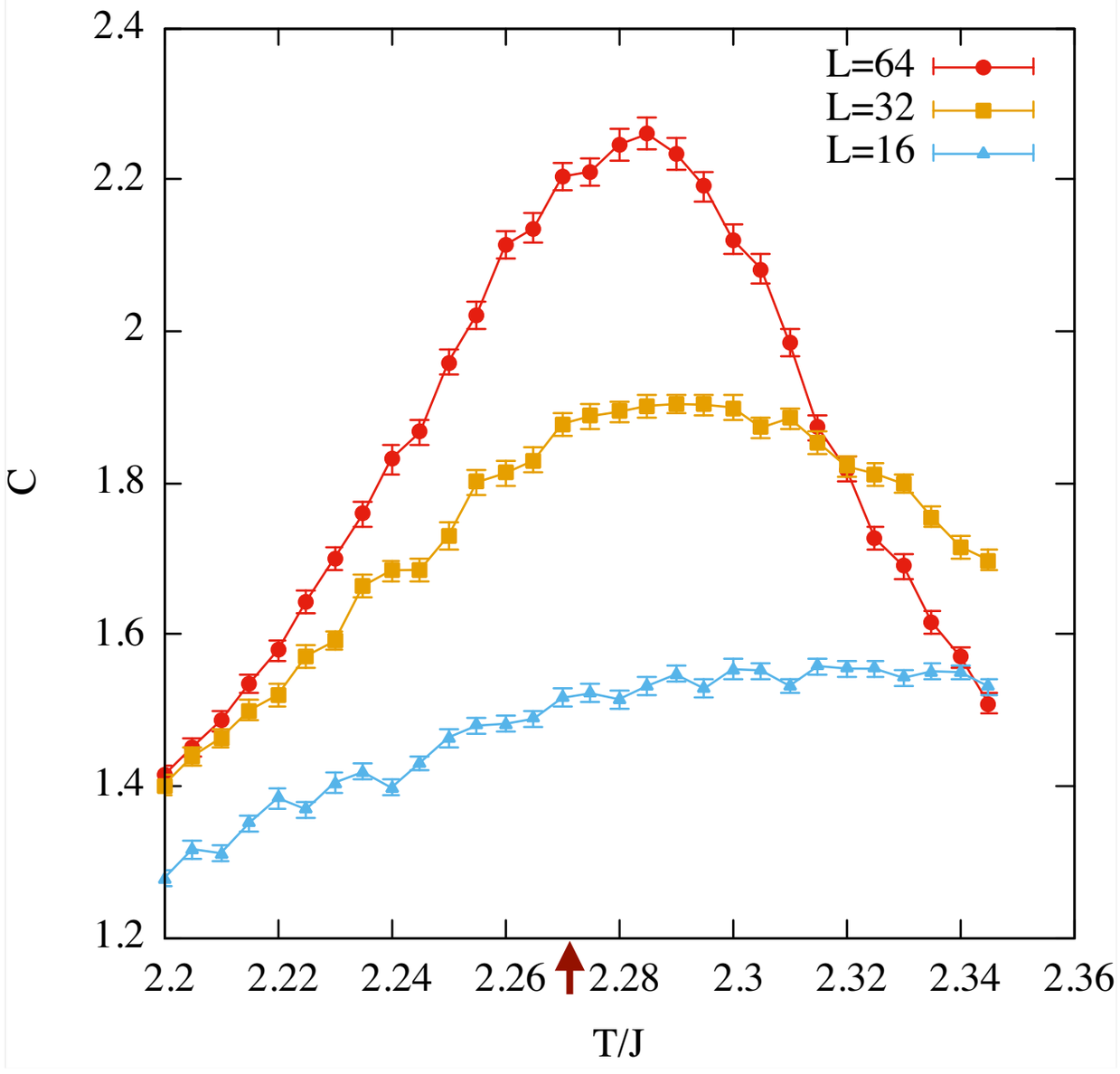
$$C = N\frac{\langle E^2\rangle - \langle E\rangle^2}{T^2}$$

$$\chi = N\frac{\langle M^2\rangle - \langle |M| \rangle^2}{T}$$

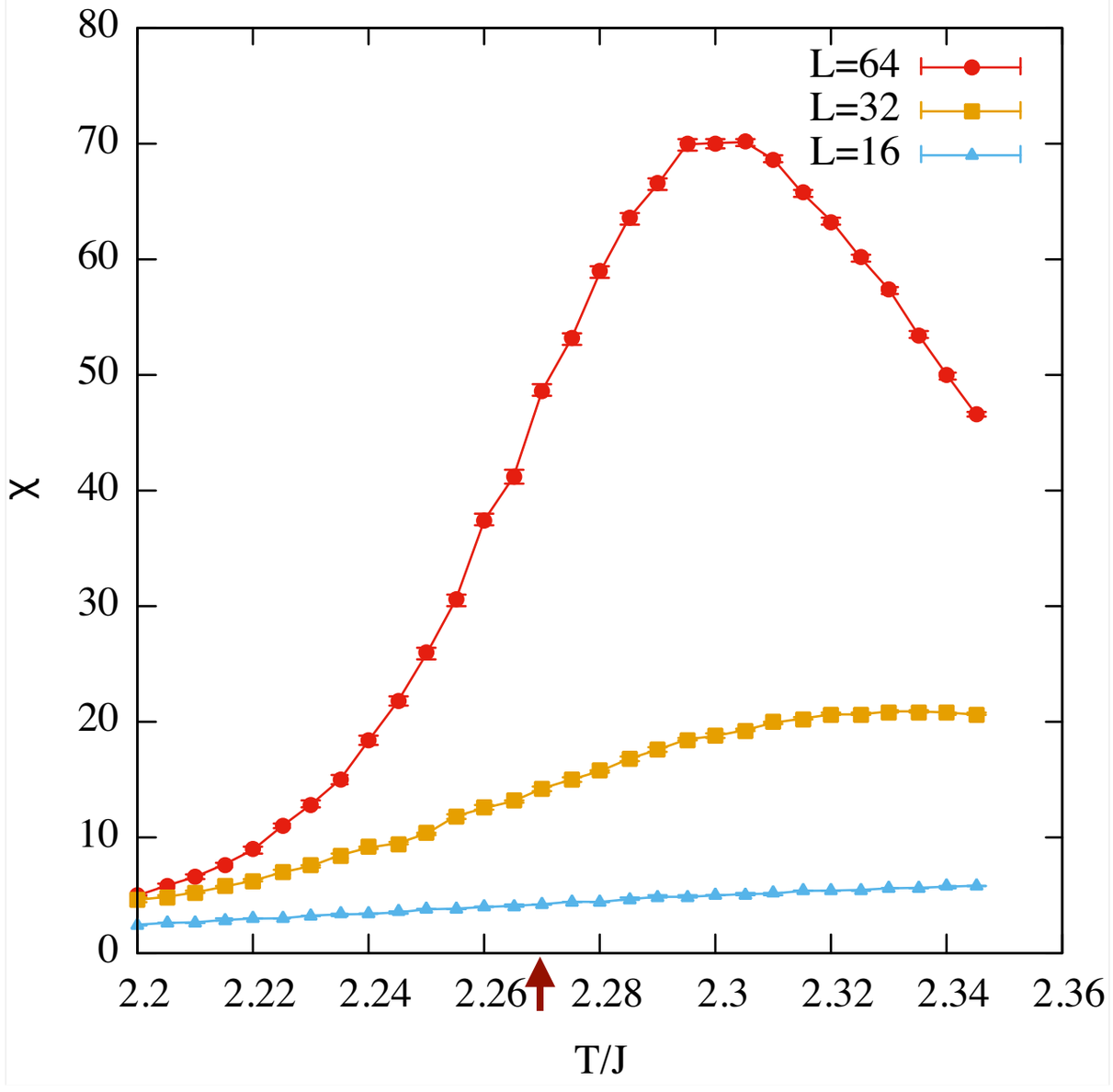
Calculated data (Swendsen-Wang)

$T_c/J \simeq 2.269$

Specific heat



Susceptibility



$$C = N\frac{\langle E^2\rangle - \langle E\rangle^2}{T^2}$$

$$\chi = N\frac{\langle M^2\rangle - \langle |M| \rangle^2}{T}$$

Data analysis: Finite size scaling (outline)

Near the critical point (transition temperature):

The singular part of the free energy density satisfies finite size scaling

$$f_s(t, h, L) = L^{-d} f_s(tL^{y_t}, hL^{y_h})$$

$$t = T - T_c$$

y_t, y_h :scaling exponent $\Longleftrightarrow$ $y_t = 1/\nu, \quad y_h = (d + \gamma/\nu)/2$

By taking derivatives, we see

$$\chi = \frac{\partial M}{\partial h} = \frac{\partial^2 f}{\partial h^2} = L^{2y_h - d} g(tL^{y_t}, 0) \qquad \text{(we set } h=0\text{)}$$

Physical quantity obeys common scaling function independent of L.

$\Rightarrow$ At the critical point, $\chi \sim L^{-x_\chi} \quad (x_\chi \equiv d - 2y_h)$

Note: $\chi = L^d \langle M^2 \rangle / T$

$$\begin{aligned} x_{M^2} &= 2(d - y_h) \\ &= \eta + d - 2 \\ &= 2x_M \end{aligned}$$

x: scaling dimension

If $x = 0$, it has no size dependence at the critical point.

Data analysis: Finite size scaling (outline)

Similarly, the energy and the specific heat obey:

$$E = \frac{\partial f}{\partial T} = L^{y_t - d} g_E(tL^{y_t}) = L^{1/\nu - d} g_E(tL^{1/\nu})$$

$$C = \frac{\partial^2 f}{\partial T^2} = L^{2y_t - d} g_C(tL^{y_t}) = L^{2/\nu - d} g_C(tL^{1/\nu}) = (L^{\alpha/\nu} g_C(tL^{1/\nu}))$$

Note: scaling relations $\nu d = 2 - \alpha, 2 - \eta = \frac{\gamma}{\nu}, \cdots$

Scaling form of general quantities are $\boxed{O = L^{-x_o} g_o(tL^{1/\nu})}$

When we plot O as $(x = tL^{1/\nu}, y = OL^{x_o})$

➡ All data are on a single curve corresponding to $y = g_o(x)$.

By using this property, we can estimate critical exponents and critical temperature.

Examples: Magnetization

(Squared) Magnetization: $\langle M^2 \rangle$

$$M = \frac{1}{N} \sum_i S_i$$

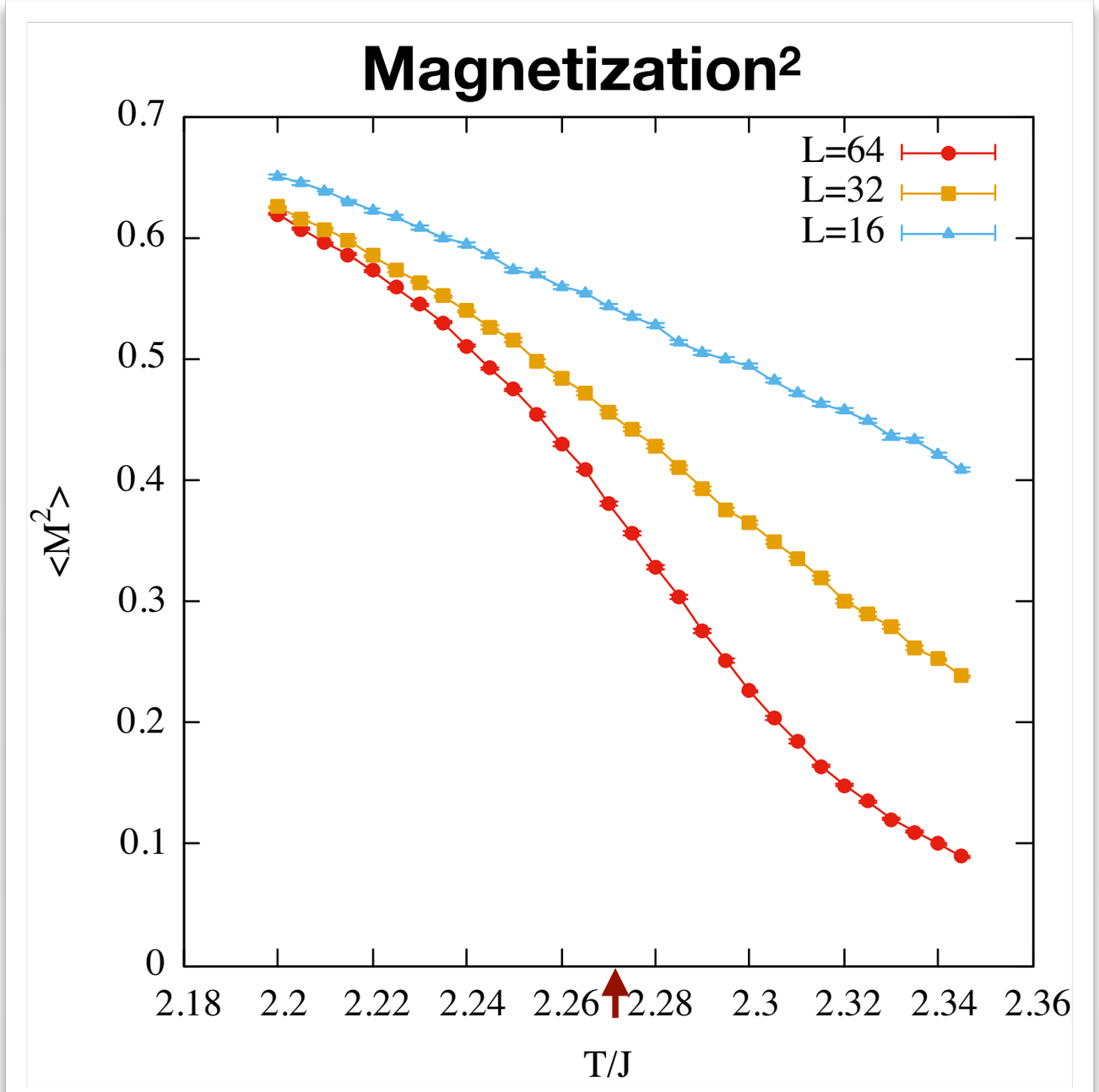
(Q. We cannot use <*M*>. Why?)

In the thermodynamic limit

$$\langle M^2 \rangle \begin{cases} = 0 & (T \geq T_c) \\ \neq 0 & (T < T_c) \end{cases}$$

So, in principle, we can estimate T_c by extrapolating the data.

Can we estimate T_c more easily?



$$T_c/J \simeq 2.269$$

Examples: Binder ratio

Binder ratio

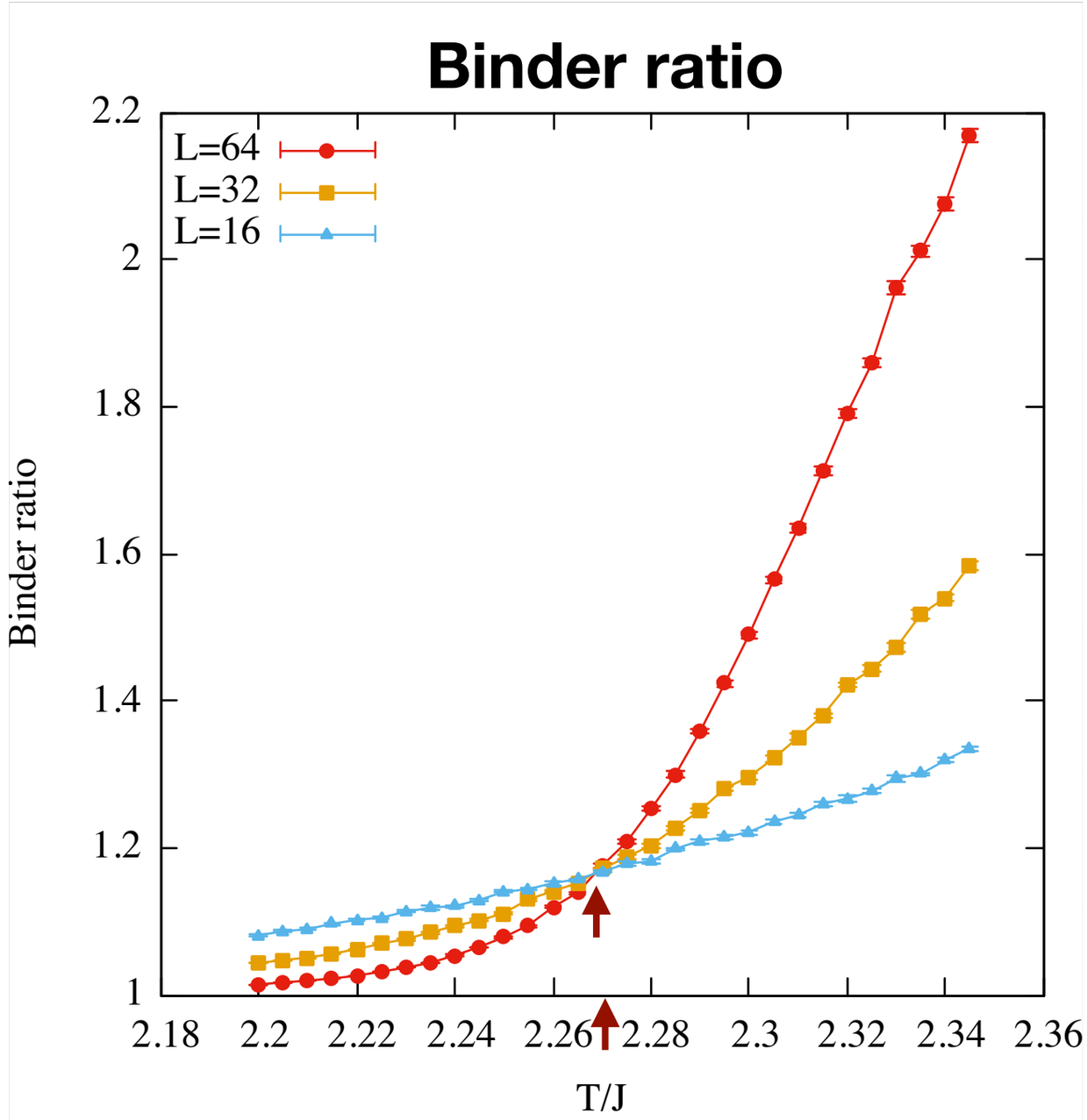
$$b = \frac{\langle M^4 \rangle}{\langle M^2 \rangle^2}$$

$$b = 3 \quad (T \to \infty)$$

$$b = 1 \quad (T \to 0)$$

The scaling dimension of b is exactly zero.

At T_c, the size dependence disappears in leading order!



$$T_c/J \simeq 2.269$$

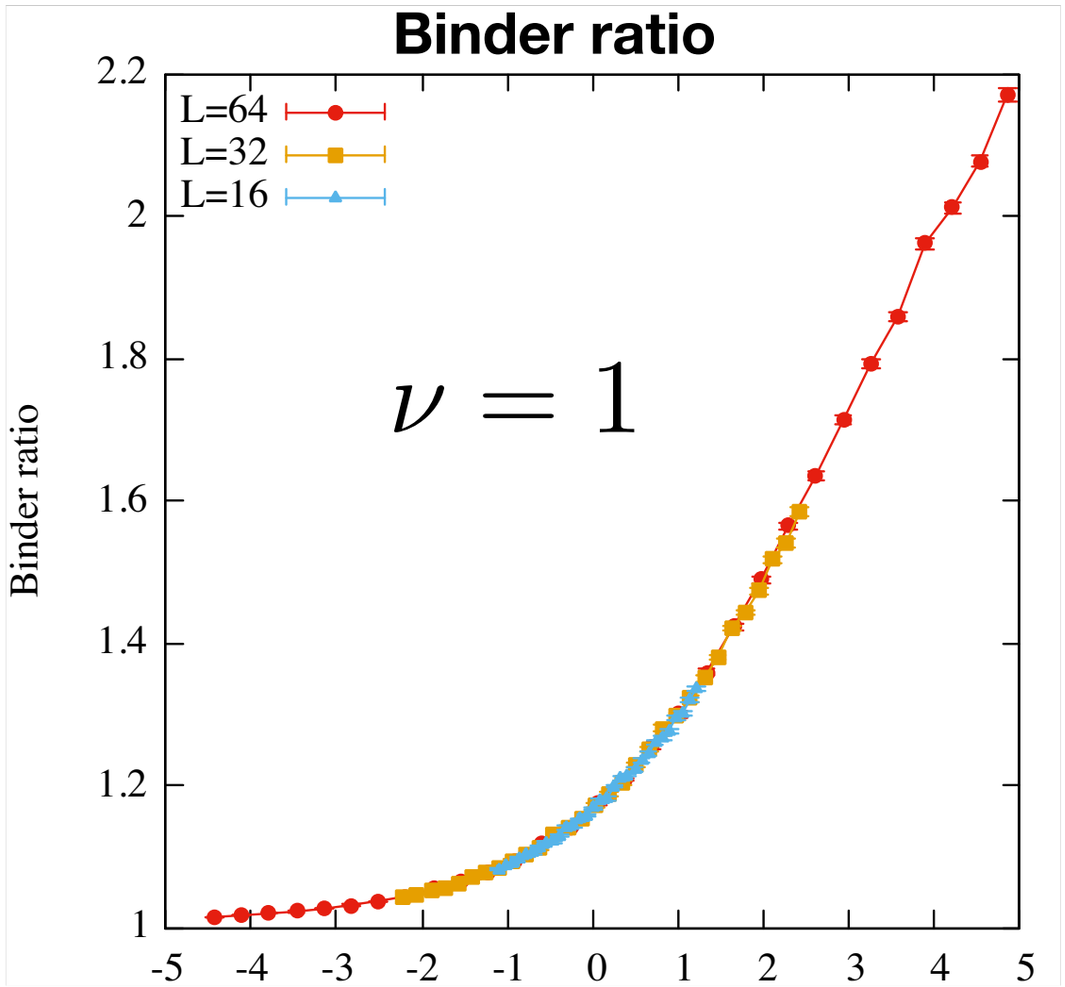
Finite size scaling: Binder ratio

Binder ratio $b = \dfrac{\langle M^4 \rangle}{\langle M^2 \rangle^2}$

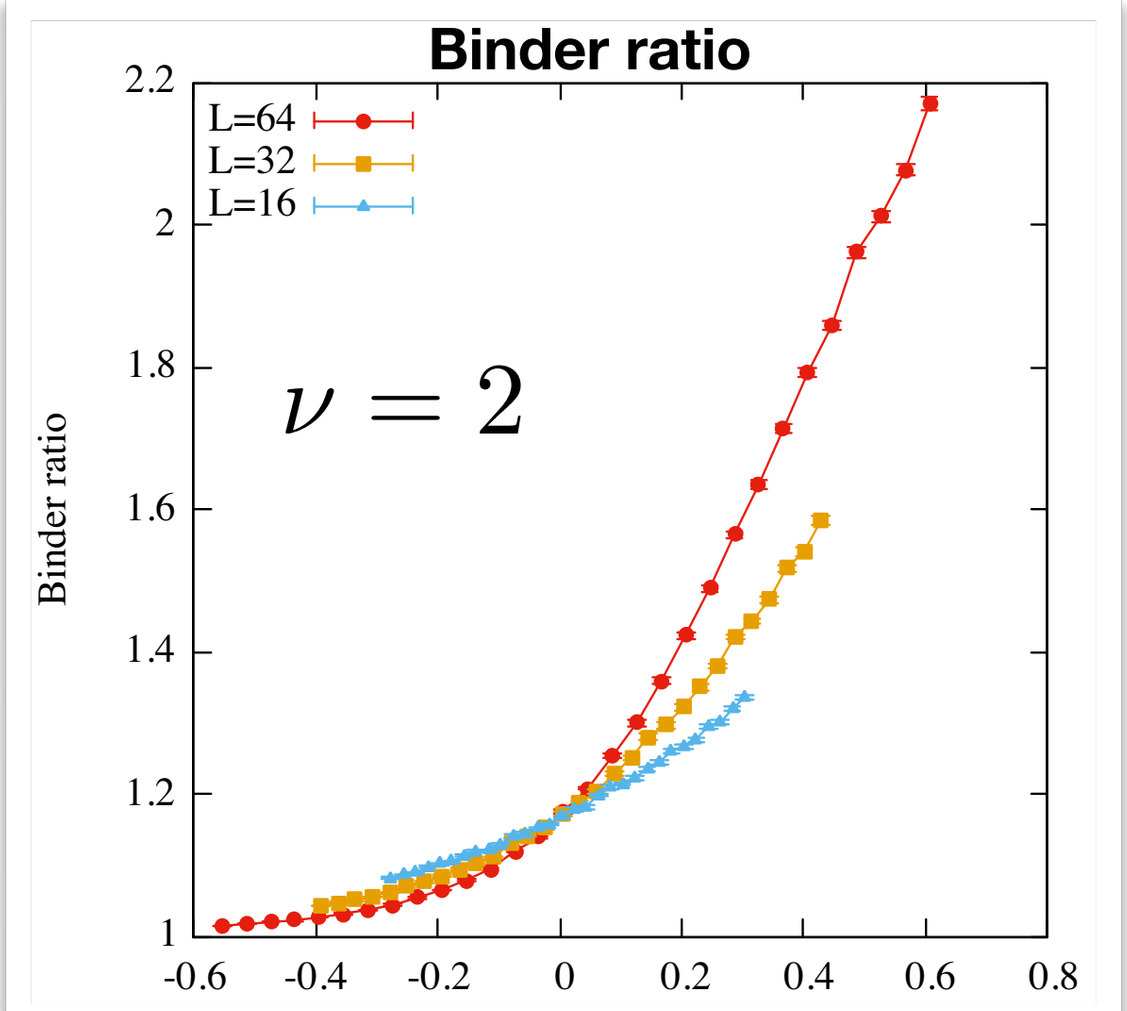
Finite size scaling around Tc

$$b = f((T - T_c)L^{1/\nu})$$

We can determine critical exponent! $\nu = 1$



$(T - T_c)L^{1/\nu}$



$(T - T_c)L^{1/\nu}$

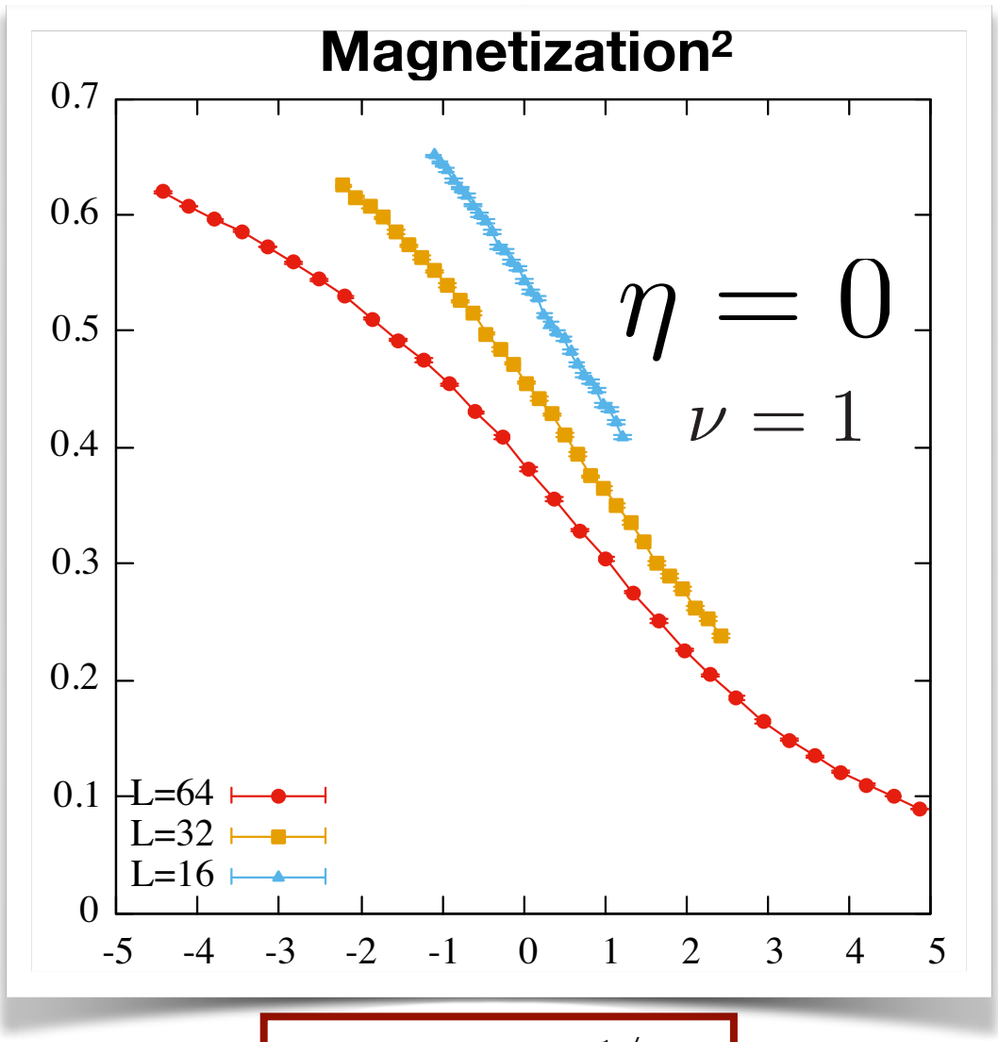
Finite size scaling: Magnetization

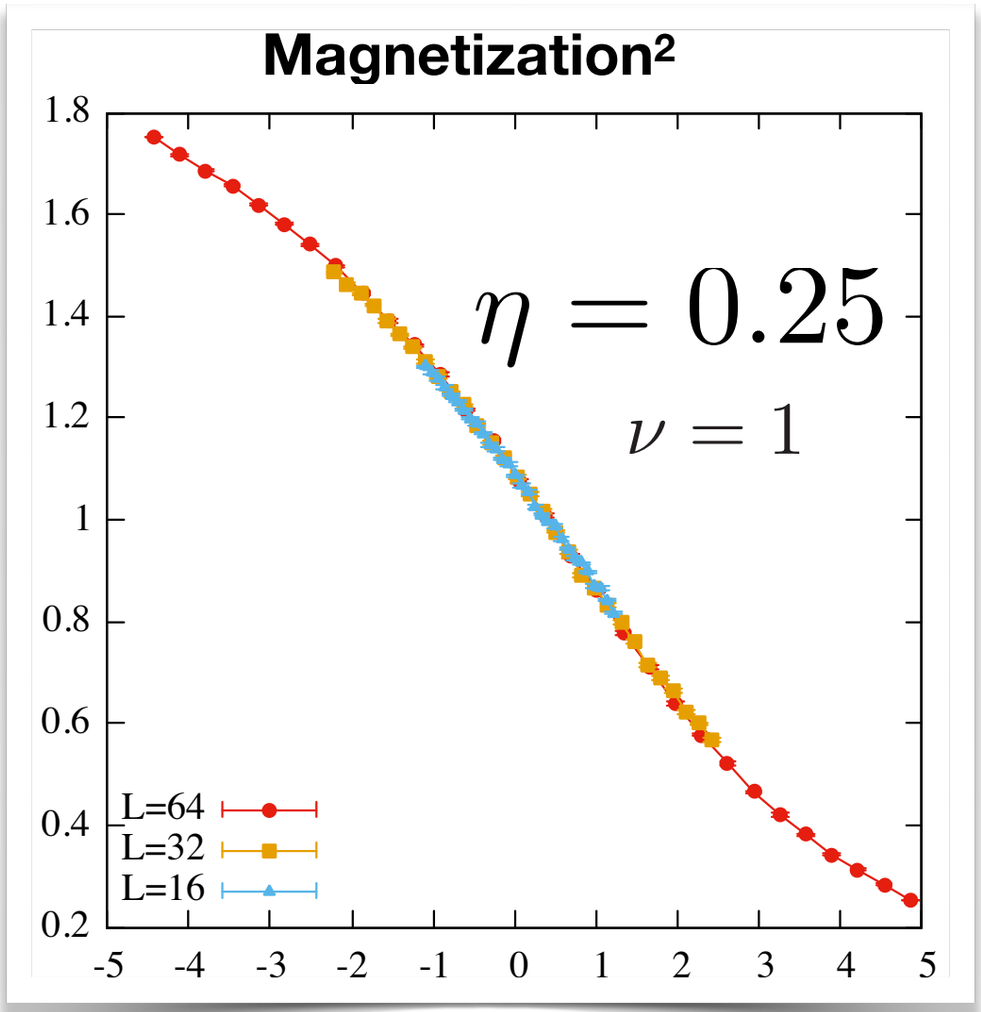
(Squared) Magnetization: $\langle M^2 \rangle$

Finite size scaling around Tc

$$\langle M^2 \rangle = L^{-\eta} g((T - T_c) L^{1/\nu})$$

By fixing $\nu = 1$ and varying η,
we can also determine another critical exponent.





Exercises and sample codes

Exercises (not a report)

Exercise1: autocorrelation of MCMC

See correlation time or autocorrelation function of Ising model calculated by Monte Carlo simulation.

- Around Tc, how does the correlation time behave by varying the temperature?
- At Tc, how about the size (L) dependence?
- Does the correlation time depend on the algorithms?

Exercise2: finite size scaling

Try the finite size scaling of, e.g. binder ratio, in the case of Ising model.

- Calculate physical quantities for various system size (L).
- Plot them without scaling, and see they are actually different.
- Try finite size scaling by assuming values of critical exponents.
  - Even if you know the exact value, it is worth trying several different values.

How to perform the exercises

To perform these exercises, you may use

- Your own code
- ~~ALPS (it is not straight forward to see the correlation time…)~~
- My sample codes for jupyter notebook (python3).
  - In order to run the sample codes you need
    - *numpy*, pickle, and *numba* modules

      (numba is used for speed up).

How to use my codes

Usage of my codes:

For jupyter notebook (Highly recommended):

jupyter notebook → select Ising-Ex1.ipynb or Ising-Ex2.ipynb

For python:

python3 Ising-Ex1.py or *python3 Ising-Ex2.py*

(In this case, you need to close figure windows to forward the analysis after you check them.)

You can see help message by "-h" option.

python3 Ising-Ex1.py -h

* --L_list of Ex2 is used, e.g.

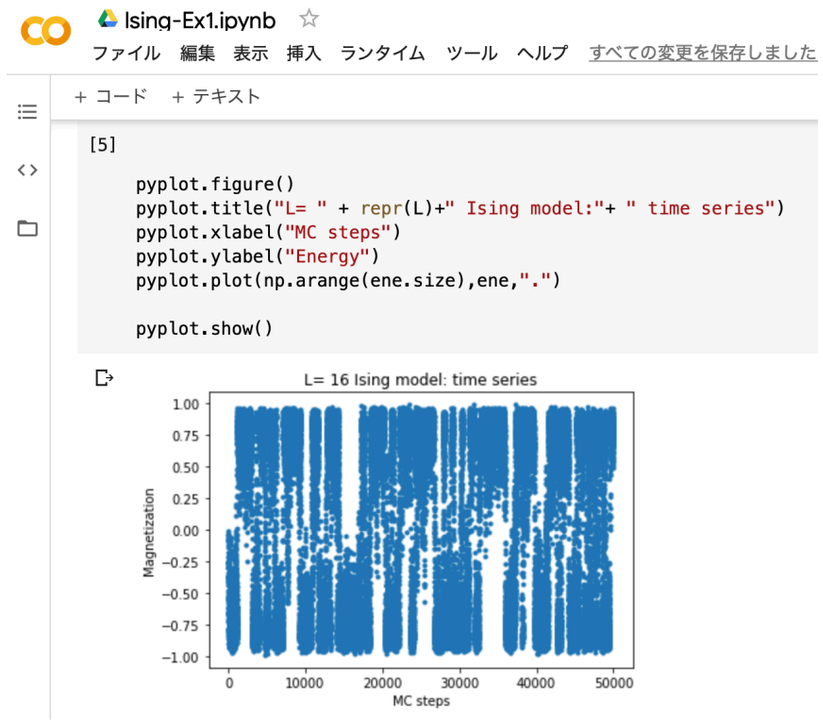
python3 Ising-Ex2.py --L_list 4 8 16 32

Do not include "[ ]" and ",".

Google Colab

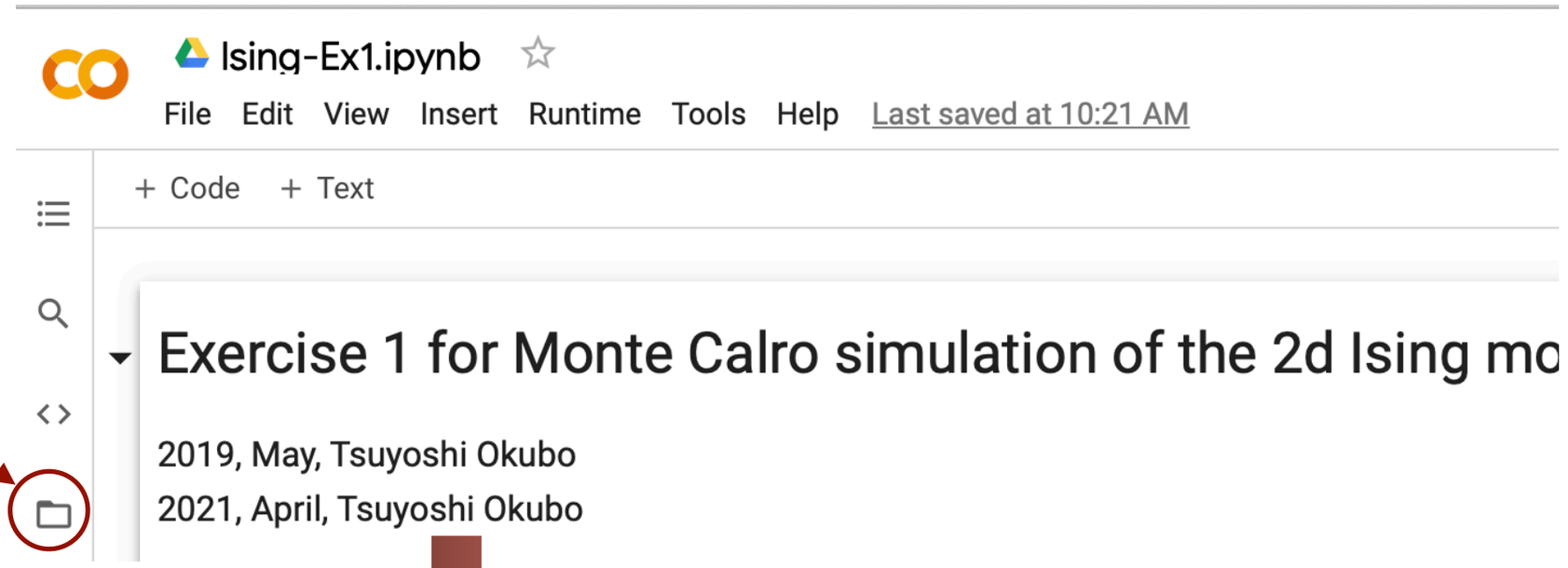
Google Colab https://colab.research.google.com/

- It is a web browser based python environment.
  - You do not need to prepare python environment on your PC.
- You can easily run the exercise codes (*.ipynb) by uploading it to the google colab.
- Before run them, you need to upload *Ising_lib.py* in addition to *.ipynb.



How to use Google Colab

1. Open, *e.g., Ising-Ex1.ipynb,* in Google colab
   - Select "**File/upload notebook**" ("ファイル/ノートブックをアップロード") and upload *Ising-Ex1.ipynb*

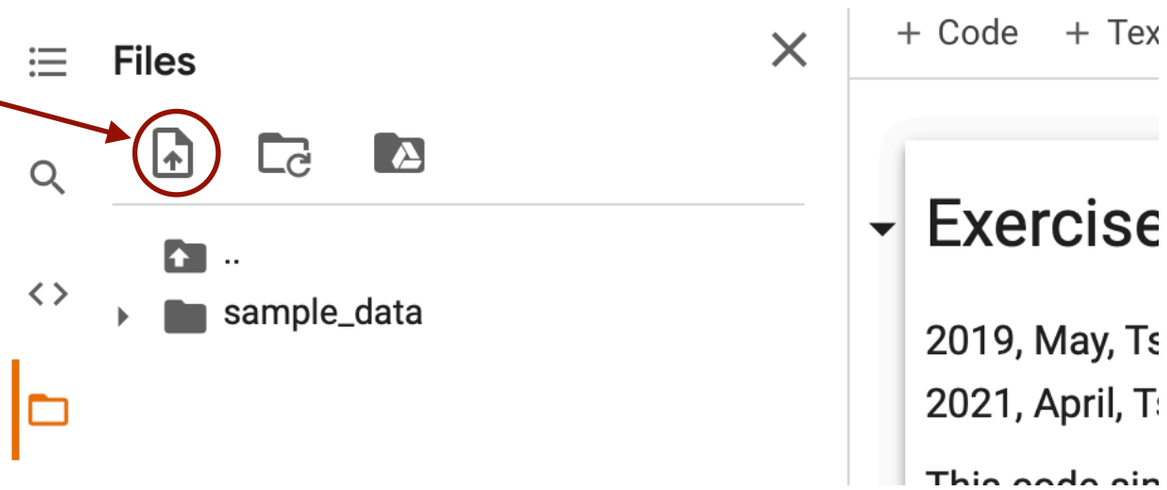


2. Click here

   (Wait a moment for the connection)

3. Click here and upload *Ising_lib.py*

   (Uploaded file will be deleted after the session finishes.)



4. Select "**Runtime/Restart and Run all**".

MateriApps Live:

MateriApps Live https://cmsi.github.io/MateriAppsLive/

- It is a collection of softwares for a variety of material simulations, together with a virtual linux environment.
- By using MA live, we can easily install and construct environment for computer simulation.
- Its virtual box version is easy to install to Win, Mac.

Examples of installed softwares:

- ALPS
- HPhi
- LAMMPS
- mVMC
- Quantum ESPRESSO
- ...

References: textbook for MCMC in the physics

- "A guide to Monte Carlo simulation in statistical physics"
  D. P. Landau and K. Binder, Cambridge university press, (2014) (4th edition).
- “Computational Physics”, J. Thijssen, Cambridge University Press.
  - Japanese translation:「計算物理学」J.M.ティッセン著、松田和典他訳、シュプリンガー・フェアラーク東京.
- “Quantum Monte Carlo methods: Algorithms for Lattice models”
  J. Gubernatis, N. Kawashima, P. Werner, Cambridge university press, (2016).
  - This is for Monte Carlo methods for quantum lattice models
- "統計科学のフロンティア12 計算統計II
  マルコフ連鎖モンテカルロ 法とその周辺"
  伊庭幸人ほか、岩波書店.
  (Unfortunately, I have not read it yet.)

Next week (5/9)

Classical

1st: Many-body problems in physics and why they are hard to solve

2nd: Classical statistical models and numerical simulation

3rd: Classical Monte Carlo method

4th: Applications of classical Monte Carlo method

5th: Molecular dynamics simulation and its applications

Quantum

6th: Extended ensemble method for Monte Carlo methods

7th: Tensor Renormalization group

8th: Quantum lattice models and numerical simulation

9th: Quantum Monte Carlo methods

10th: Applications of quantum Monte Carlo methods

11th: Linear algebra of large and sparse matrices for quantum many-body problems

12th: Large sparse matrices and quantum statistical mechanics

13th: Advanced algorithms for quantum many-body problems