拡張アンサンブル法によるモンテカルロ計算

Extended Ensemble method for Monte Carlo Methods

理学系研究科　物理学専攻　大久保　毅

Announcement: FAQ for sample python codes

I prepared FAQ page for running sample python codes:

https://exa.phys.s.u-tokyo.ac.jp/ja/members/okubo/zj47qr

If you meet troubles to use them, please visit here.

Note:

Information on this web page will be updated from time to time.

Contents

- Back ground
  - Difficulty in large relaxation time
  - Density of states and the histogram method
- Extended ensemble methods using information of the density of states
  - Multi Canonical Method
  - Wang-Landau method
- Another extended ensemble: Replica exchange method
- ALPS and report problem

Back ground

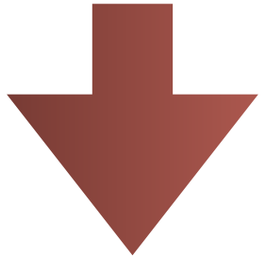
Extended ensemble = general ensemble

In conventional MC or MD simulation:

We try to estimate expectation values under

"physically relevant" ensembles.

NVE, NVT, NPT, …



Even if an ensemble is not directly connected to any physical systems, we can use is to enhance the efficiency of numerical calculations (MC, MD) for interested physical system.

Large relaxation time in standard MC and MD

- Critical phenomena
  - $\tau \sim L^z$ with standard algorithm (critical slowing down)
  - z can be significantly reduced by using "global update"
- First order phase transition, Glass transition (structural glass, spin glass), protein foldings,
  - $\tau \sim \exp(\Delta E/T)$ or $\exp(\Delta E/|T\text{-}T_c|)$; Note $\Delta E \propto L^d$ $(or\ L^{d-1})$!
  - Exponential can be reduce to polynomial by using extended ensemble methods.

Origin of exponentially long relaxation time

Partition function of the canonical ensemble

$$Z = \int d\Gamma e^{-\beta \mathcal{H}(\Gamma)} = \int dE \underline{\rho(E)} e^{-\beta E} = \int dE e^{-\underline{\beta F(E)}}$$

Density of state

"Free energy"

$$F(E) \equiv E - k_B T \log \rho(E)$$

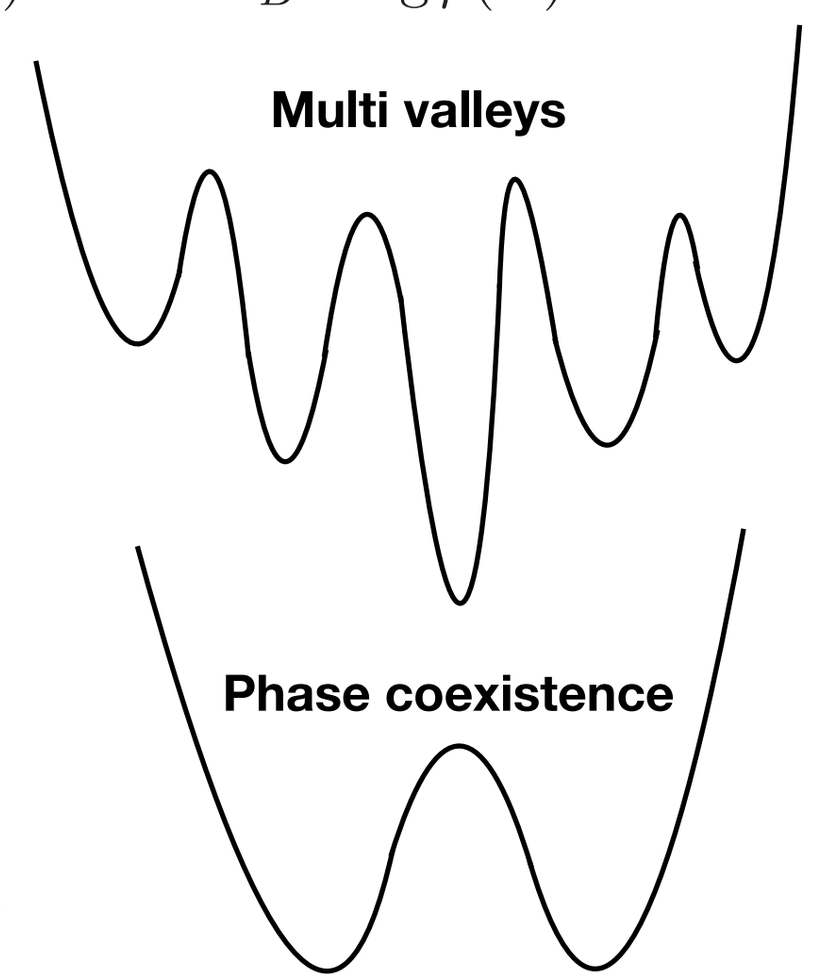
Probability distribution for energy

$$P(E) = \frac{1}{Z} e^{-\beta F(E)}$$

Note! Free energy is extensive: $F(E) \propto N$

- "Transition probability" is proportional to the exponential of Free-energy difference: exp(-ΔF/T)
- Usual algorithm of MC (and MD) changes the state (or the energy) gradually.

If there are local minima, the relaxation time could be exponentially large as the size is increased.



Density of state and histogram method

Density of state

$$Z = \int d\Gamma e^{-\beta \mathcal{H}(\Gamma)} = \int dE \rho(E) e^{-\beta E}$$
$$= \int dE \int dM \rho(E, M) e^{-\beta E}$$

$\int d\Gamma$	~O(N)-dimensional integral
$\int dE$	~1-dimensional
$\int dE \int dM$	~2-dimensional

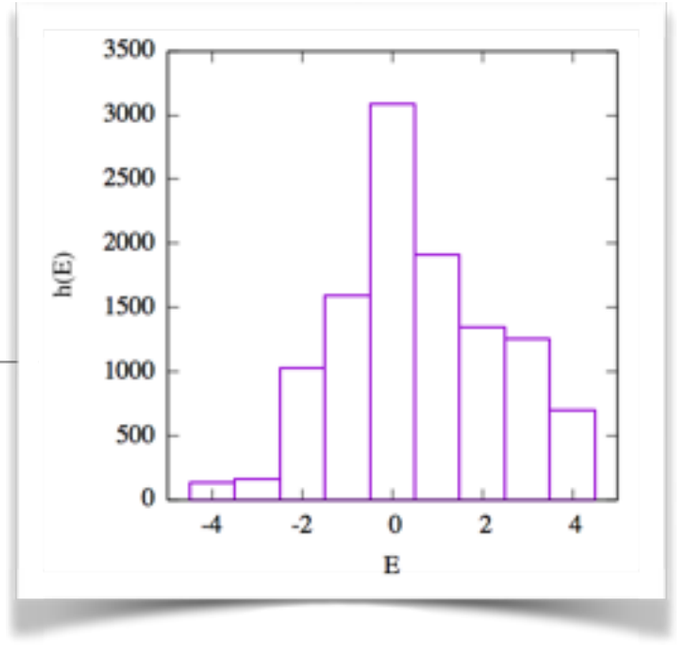
- If we know the exact $\rho(E)$(or $\rho(E, M)$), the calculation of partition function is reduced to 1 or (a few) -dimensional integral.
- Even if we only know an approximate density of state,

$$\tilde{\rho}(E) \simeq \rho(E)$$

we can improve the sampling efficiency by using its information

- Histogram method
- Multi canonical method
- Wang-Landau method

Energy Histogram



Energy histogram:

In MC or MD calculations

$h(E_i)$:# of samples (snap shots) with energy in

$$E_i - \Delta E/2 \leq E < E_i + \Delta E/2$$

$$\Rightarrow \quad P(E)\Delta E \simeq \frac{1}{N_h} h(E)$$

Total # of samples

$$N_h \equiv \sum_i h(E_i)$$

e.*g.* NVT ensemble:

$$P(E) = \frac{1}{Z(\beta)} \rho(E) e^{-\beta E} \quad \Rightarrow \quad \rho(E) \simeq \frac{Z(\beta)}{N_h \Delta E} h(E) e^{\beta E}$$

We can calculate (approximate) density of state from usual MC or MD simulations!

* Because we don't know the partition function, DOS is determined up to the proportional coefficient.

Histogram method (reweighting method)

Energy expectation value of different temperatures

$$\langle E \rangle_{\beta'} = \frac{\int dE \rho(E) E e^{-\beta' E}}{\int dE \rho(E) e^{-\beta' E}} \simeq \frac{\sum_i E_i h(E_i) e^{-(\beta'-\beta)E_i}}{\sum_i h(E_i) e^{-(\beta'-\beta)E_i}}$$

Any expectation values can also be calculated by the histogram method.

$$\langle O \rangle_{\beta'} \simeq \frac{\sum_i O(E_i) h(E_i) e^{-(\beta'-\beta)E_i}}{\sum_i h(E_i) e^{-(\beta'-\beta)E_i}}$$

$$\rho(E) \simeq \frac{Z(\beta)}{N_h \Delta E} h(E) e^{\beta E}$$

Average at energy E_i

$$O(E_i) \equiv \sum_{E(\Gamma_j) \in E_i} O(\Gamma_j)$$

Limitation of histogram method

Reweighted histogram becomes
less accurate
when T' is far from the original T.

"Tail" of the original histogram has only small # of snapshots.→large noise

Central limit theorem

Width of energy distribution: $\propto \sqrt{N}$

Average of energy: $\propto N$

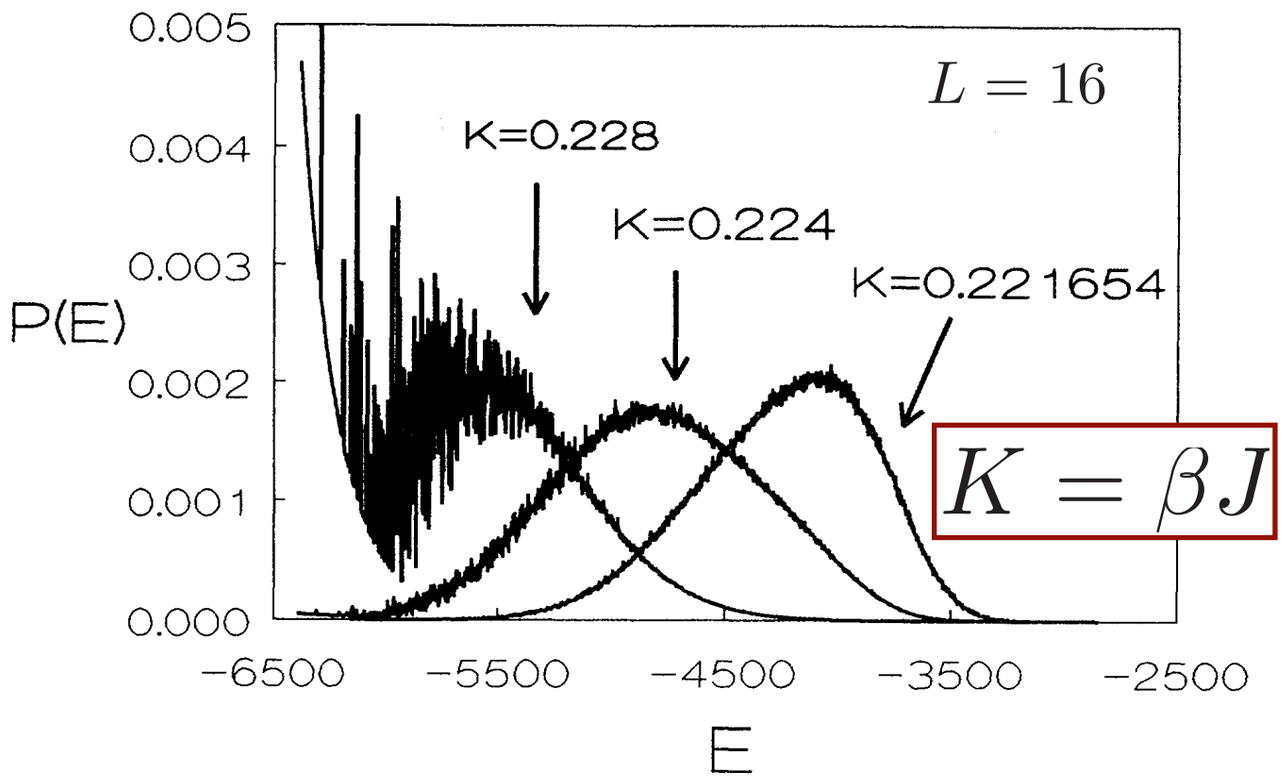
Distribution becomes narrower as *N* is increased!

Reliable temperature region for reweighting:

$$\Delta T \propto \frac{1}{\sqrt{N}}$$

Energy distribution of 3d-Ising model

A. M. Ferrenberg and D. P. Landau, Phys. Rev. B **44**, 5081 (1991)



MC simulation at K=0.221654

Reweighting to
K=0.224 and K=0.228

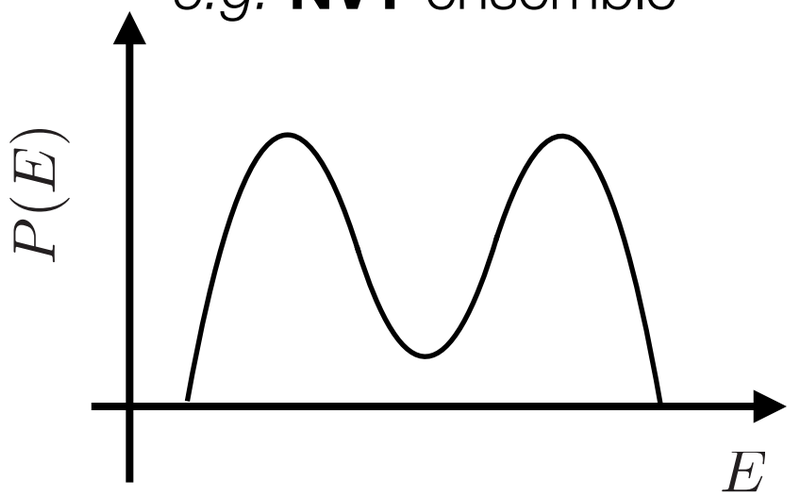
Multi Canonical methods

Idea of Multi-Canonical method

B.A. Berg and T. Neuhaus (1992)

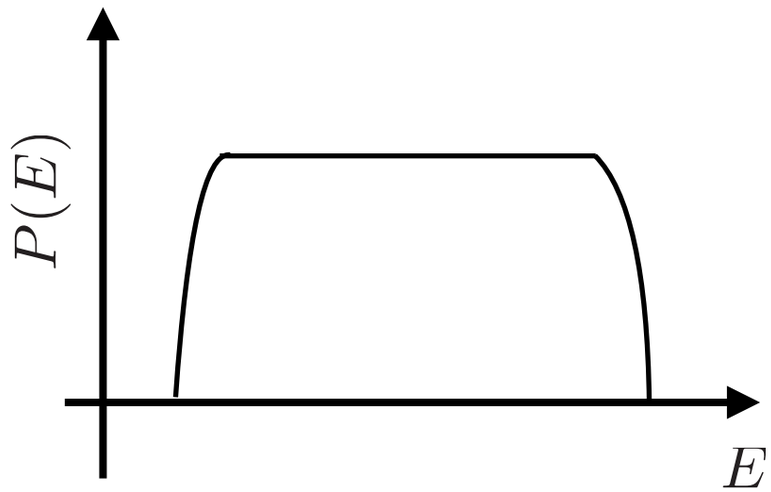
If we can prepare a special ensemble
where the energy distribution is “flat”,
we can efficiently sample all relevant states.

e.g. **NVT** ensemble



$$P(E) \propto \rho(E) e^{-\beta E}$$

“special” ensemble



$$\tilde{P}(E) \propto \rho(E) e^{-S(E)} = \text{const.}$$

Special ensemble is related to log of DOS!

$$S(E) = \log \rho(E)$$

How to obtain the special ensemble?

Special ensemble is log of DOS!

$$S(E) = \log \rho(E)$$

DOS is unknown!

We can obtain $f(E)$ approximately by iterative calculations.

“Image” of an iterative algorithm

1. Run MC simulation on a high temperature and calculate energy histogram.

$$h(E) \sim \rho(E) e^{-\beta E}$$

2. Based on the energy histogram, extract approximate $S(E)$.

$$S^0(E) = \beta E + \log h(E)$$

3. **Loop** n
    1. Run MC simulation under $S^{(n)}(E)$ and calculate histogram $h^{(n)}(E)$
    2. Calculate next $S^{(n+1)}(E)$ as

$$S^{(n+1)}(E) = S^{(n)}(E) + \log h^{(n)}(E)$$

How to obtain the special ensemble?

3. **Loop** n
   1. Run MC simulation under $S^{(n)}(E)$ and calculate histogram $h^{(n)}(E)$
   2. Calculate next $S^{(n+1)}(E)$ as

$$S^{(n+1)}(E) = S^{(n)}(E) + \log h^{(n)}(E)$$

- The histogram h(n)(E) is expected to be $h^{(n)}(E) \sim \rho(E) e^{-S^{(n)}(E)}$
- When $S^{(n)}(E)$ becomes close to $log\, \rho(E)$, the histogram becomes almost flat.

We can efficiently sample the histogram and DOS.

- By using accurate $S^{(n)}(E)$ we can calculate expectations values under the canonical ensemble by using reweiting technique.

$$\langle O \rangle_\beta = \frac{\int dE O(E) \rho(E) e^{-S(E)} e^{-\beta E + S(E)}}{\int dE \rho(E) e^{-S(E)} e^{-\beta E + S(E)}} = \frac{\langle O e^{-\beta E + S(E)} \rangle_S}{\langle e^{-\beta E + S(E)} \rangle_S}$$

Example of application

q-state Potts model on the square lattice

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \delta_{S_i, S_j} \qquad S_i = 0, 1, 2, \cdots, q-1$$

Phase transition at

$$T_c/J = \frac{1}{\log(1+\sqrt{q})}$$

$q = 2$: Equivalent to Ising model

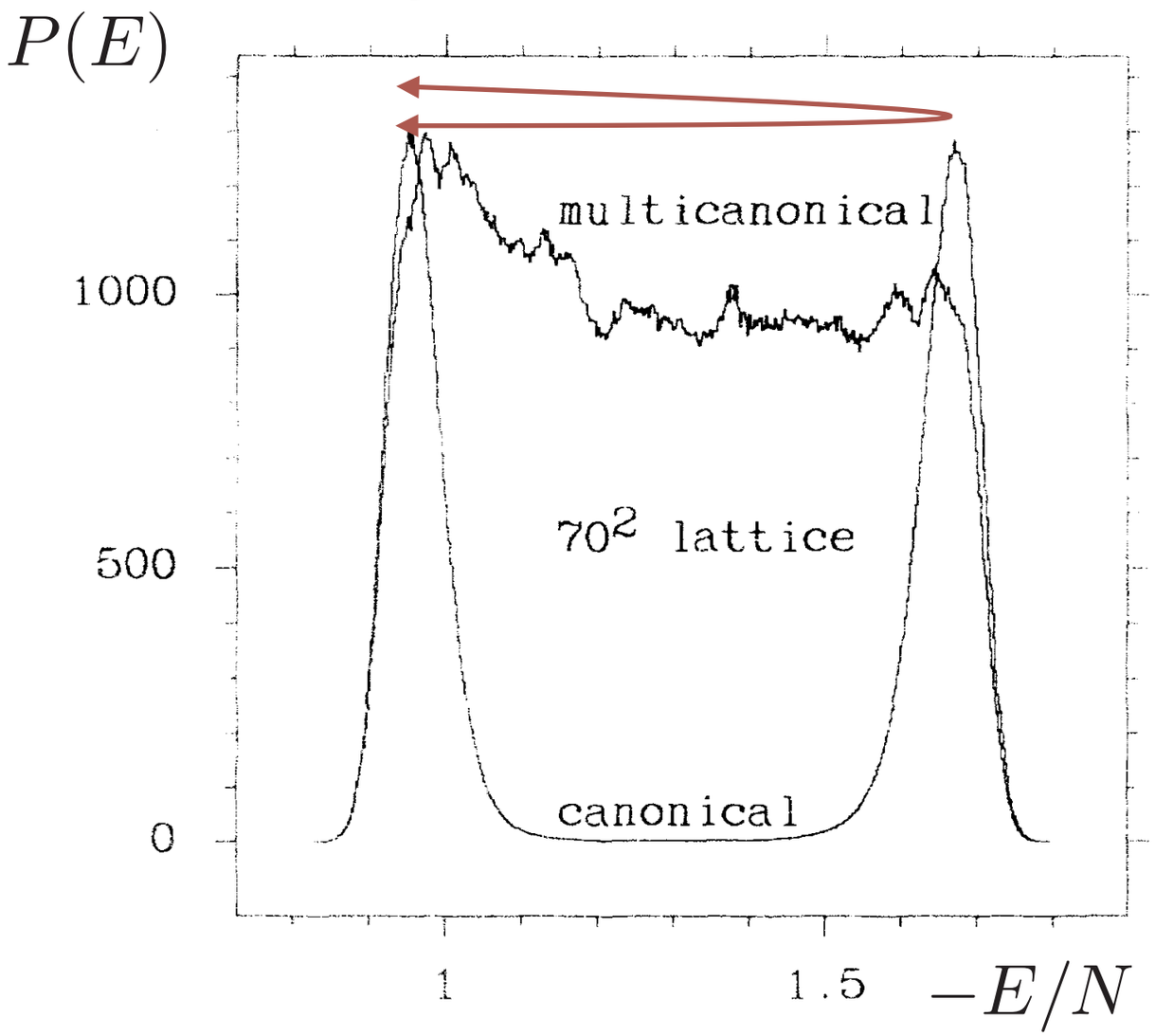
$q \leq 4$: Continuous phase transition

$q > 5$: 1st order phase transition

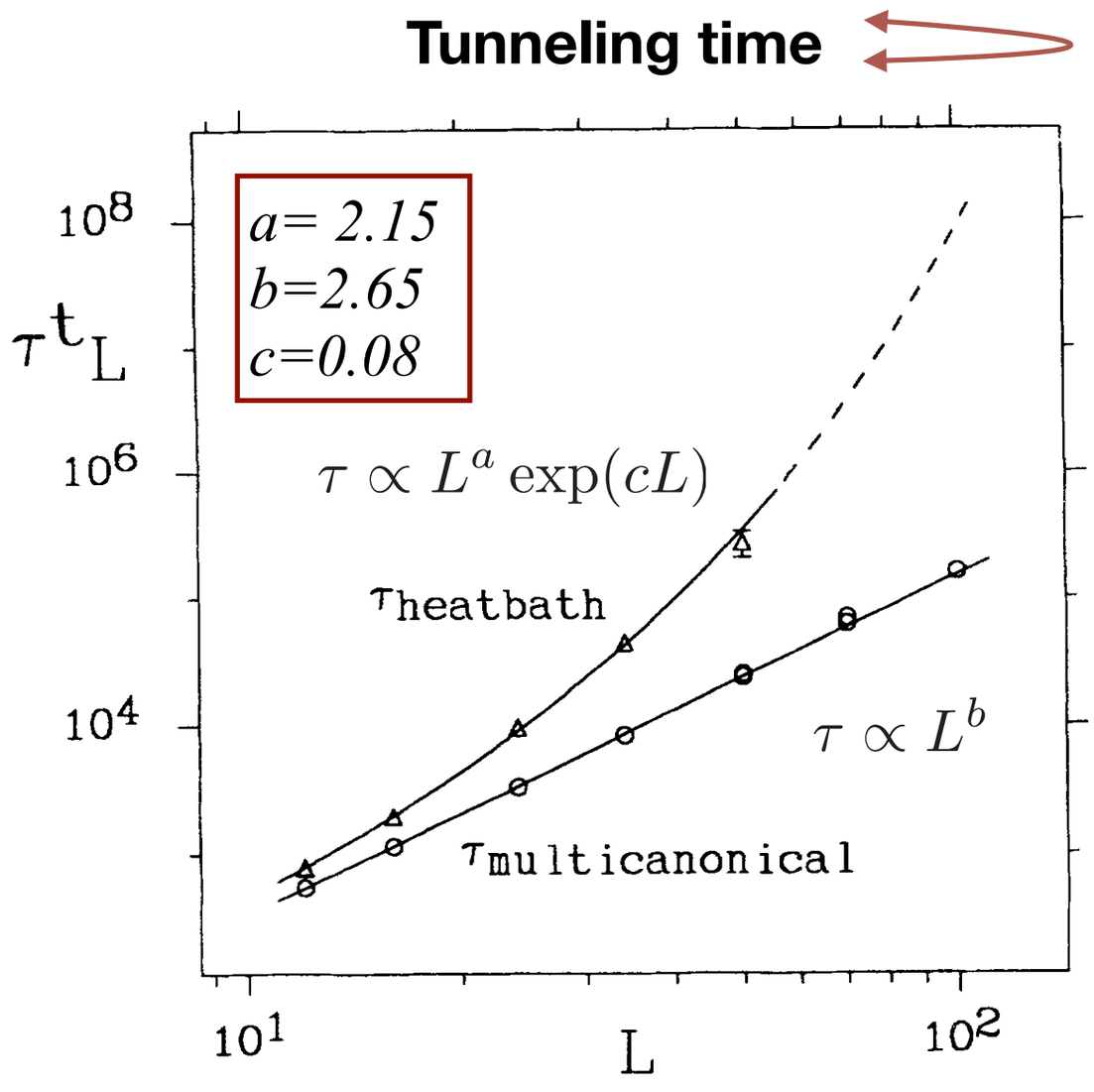
Multi Canonical method for q=10 Potts model

B.A. Berg and T. Neuhaus, Phys. Rev. Lett. **68**, 9 (1992)

Energy distribution around Tc



Tunneling time



By Multi canonical method, the tunneling time is reduced to the power of L!

Wang-Landau method

F. Wang and D. P. Landau (2001)

Another method to obtain the density of state:

Random walk on the energy space

Markov Chain Monte Carlo with the probability

$$W_{\Gamma\to\Gamma'} = \min\left(\frac{g(E(\Gamma))}{g(E(\Gamma'))}, 1\right)$$

$g(E)$:estimate of DOS

if $g(E) = \rho(E)$ → This MCMC give us completely flat histogram

Wang-Landau method:update of g(E)

F. Wang and D. P. Landau (2001)

Initially, we don't know DOS. → Set an initial guess, e.g. $g(E) = 1$

Along MCMC, we update *g(E)* of the *E(Γ)* as

$$g_{new}(E) = g(E) \times f \qquad (\log g_{new}(E) = \log g(E) + \log f)$$

If the multiplication factor is "gradually" reduced to f=1,

***g(E)* eventually converges to *ρ*(*E*).**

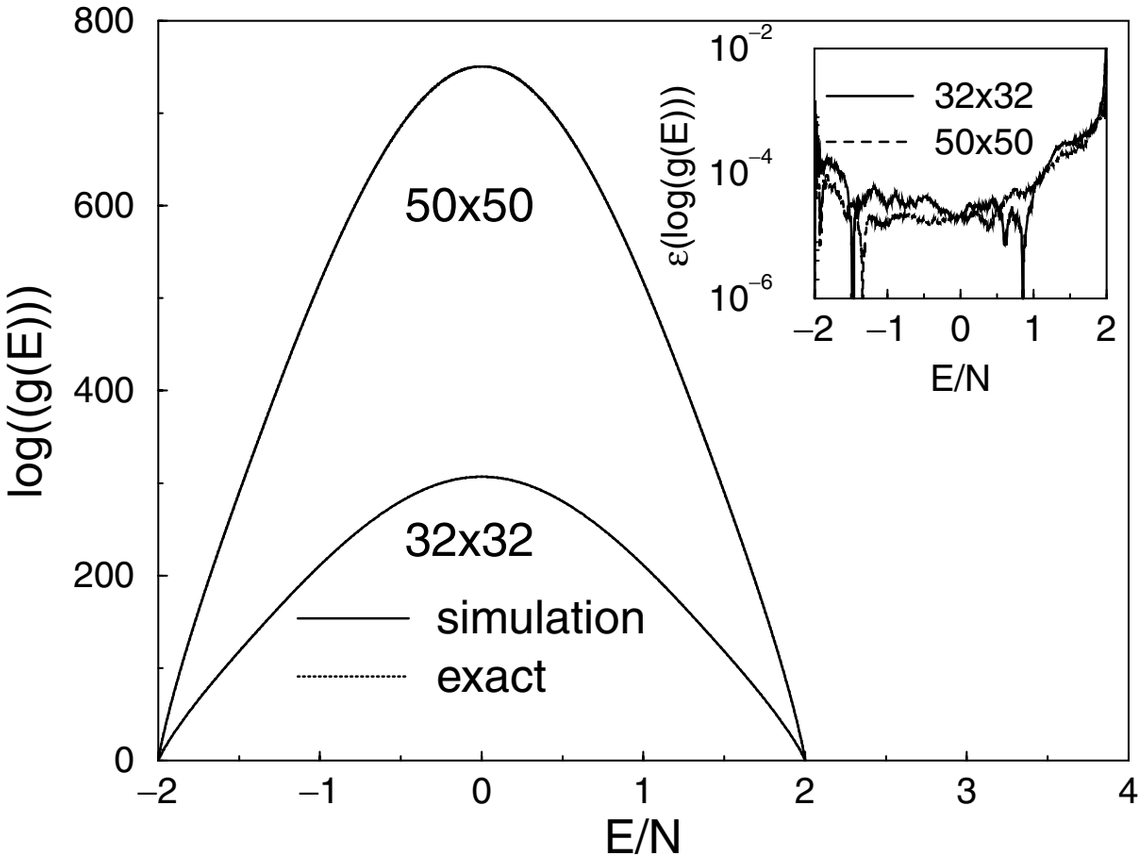
"gradual" change of f

1. Initially $f=f_0$ (e.g. $f_0 = e^1$)
2. Loop i
   - If (the histogram *h(E)* becomes "flat"?)
     - Then, we decrease f_i as
       $f_{i+1} = (f_i)^x$ (e.g. $x = 1/2$),
       and reset the histogram.
3. Repeat until f_i becomes enough small (e.g. $f \sim \exp(10^{-8})$)

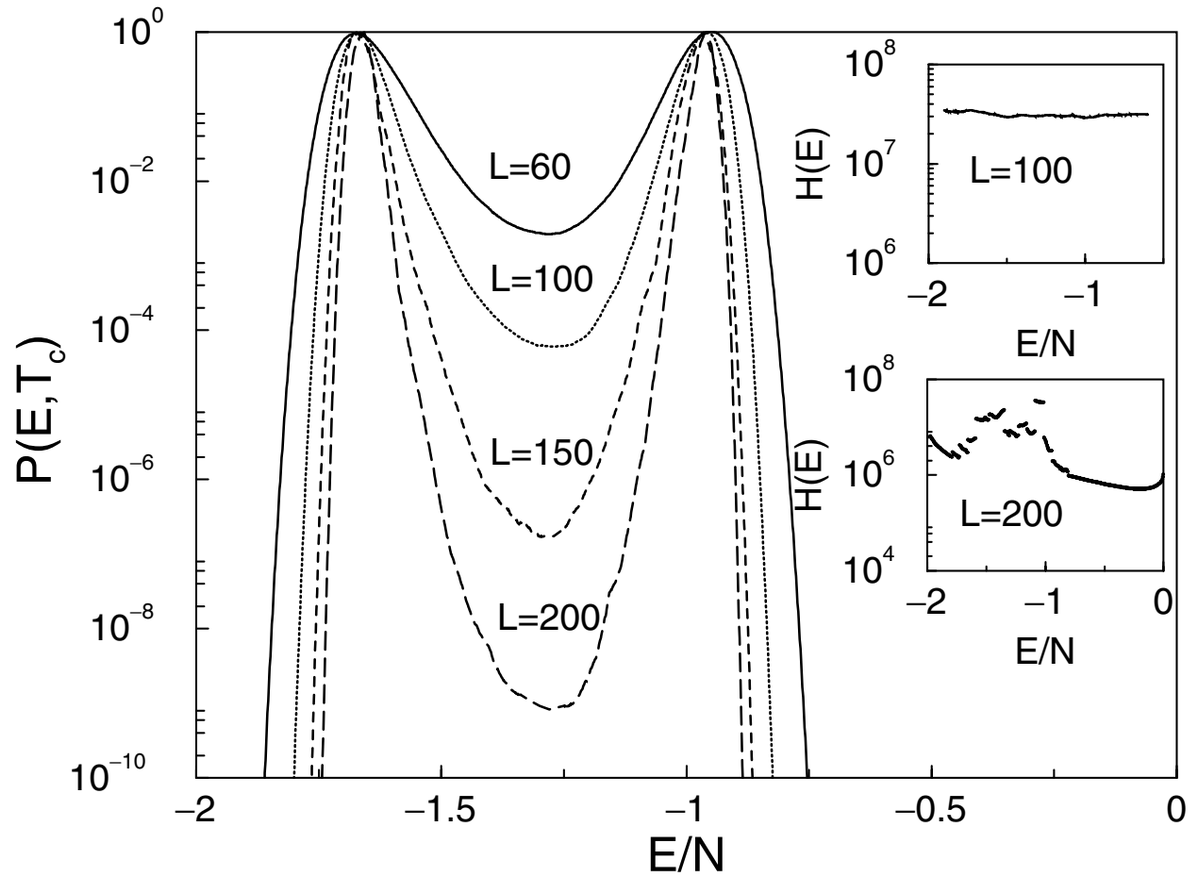
Power of Wang-Landau method

F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001)

Density of state of 2D-Ising model



Density of state of q=10 Potts model



We can obtain very accurate density of state by Wang-Landau method!

Replica Exchange method

K.Hukushim and K. Nemoto, J. Phys. Soc. Jpn. **65**, 1604 (1996).

Replica exchange (parallel tempering)

A different types of extended ensemble:

Usual MC or MD considers one parameter and one realization:

$$T,\ \Gamma = \{S_i\}, \{\boldsymbol{q}_i, \boldsymbol{p}_i\}$$

Replica exchange method considers
multiple parameters together with multiple realizations:

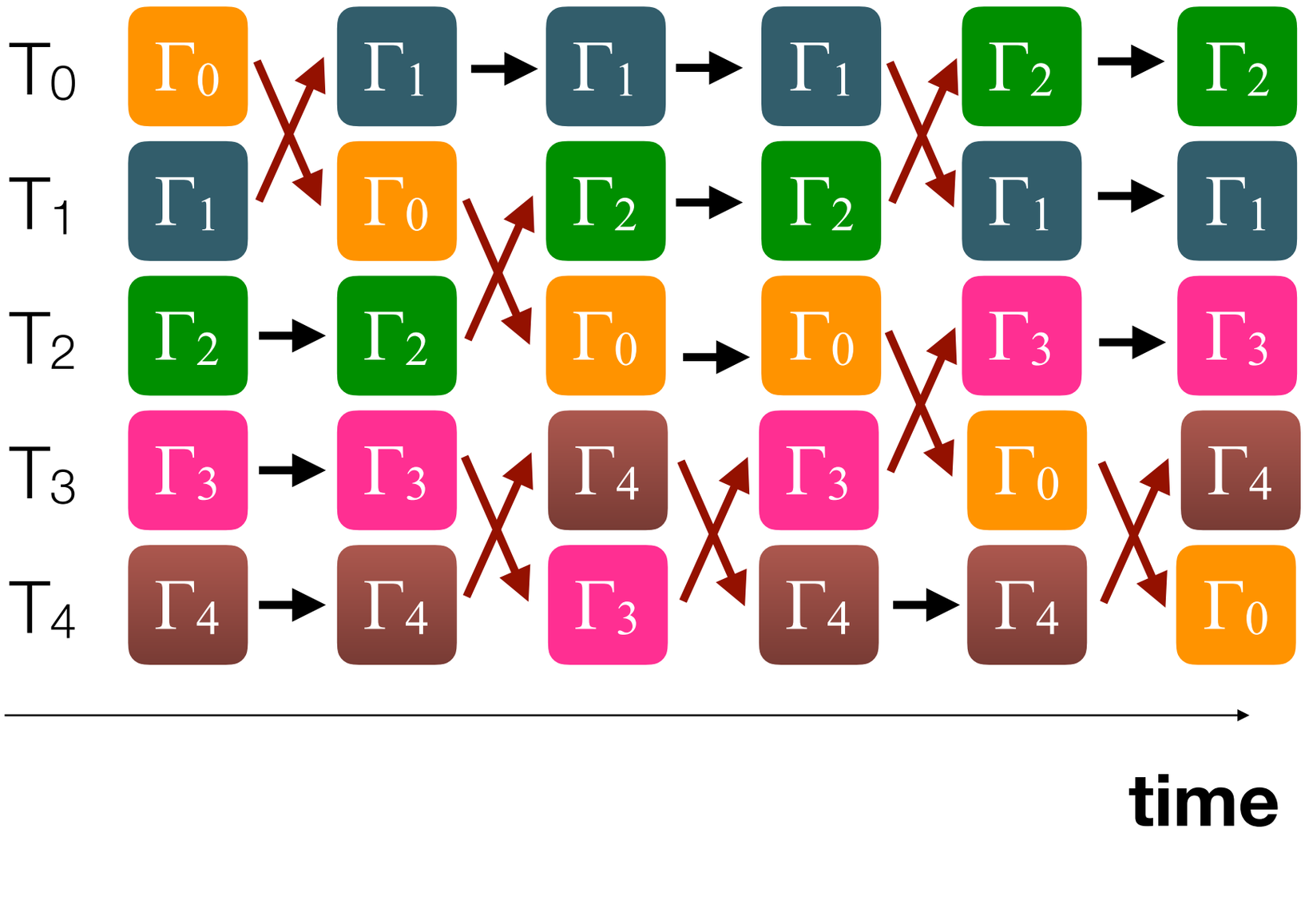
$$\{T_0, T_1, \cdots, T_M\},\ \{\Gamma_0, \Gamma_1, \cdots, \Gamma_M\},$$

Try to sample "(M+1)-dimensional" joint-distribution

$$P(\Gamma_0, \Gamma_1, \cdots, \Gamma_M; T_0, T_1, \cdots, T_M)$$

“Replica exchange”

Along simulation, we “exchange” the relationship between parameter and realization

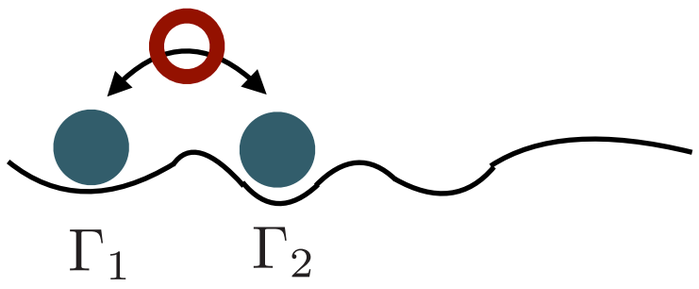


Purpose of replica exchange

Free energy landscape depends on the parameter

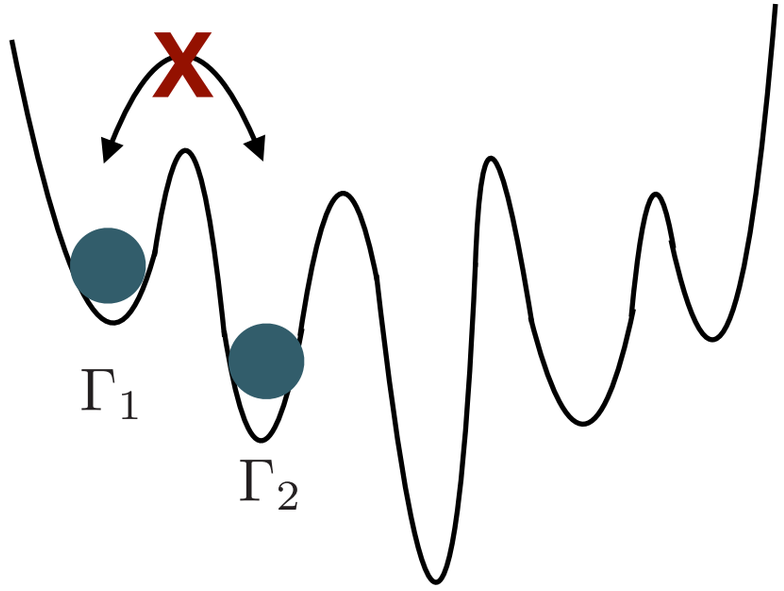
High temperature:T_h

Γ easily moves to other points!



Low temperature:T_l

Γ hardly moves to other minima!



Make a pass like:

$$\{\Gamma_1, T_l\} \rightarrow \{\Gamma_1, T_h\} \rightarrow \{\Gamma_2, T_h\} \rightarrow \{\Gamma_2, T_l\}$$

low **high** **high** **low**

* Parameter is not necessary a temperature.

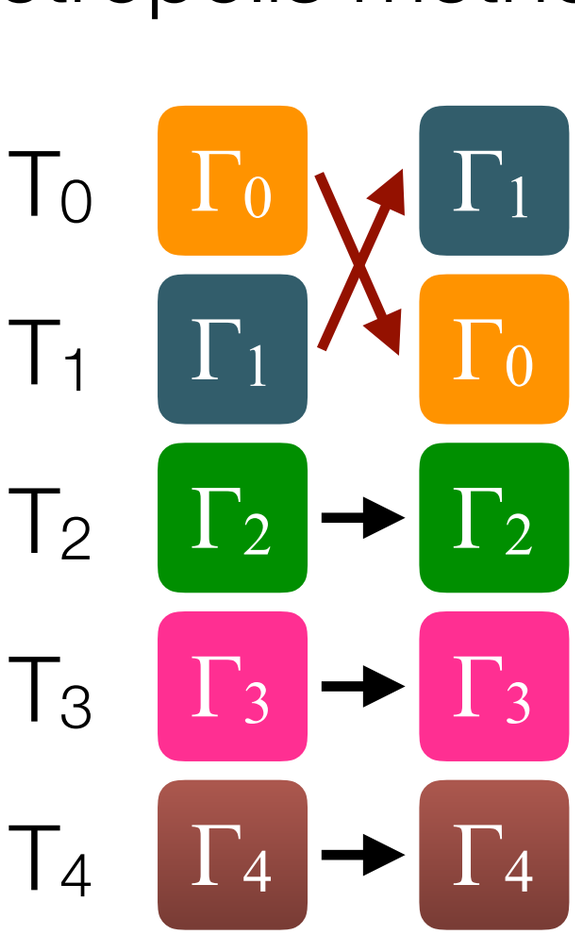
Markov Chain Monte Carlo for Replica Exchange

Target steady state distribution:

$$P(\Gamma_0, \Gamma_1, \cdots, \Gamma_M; T_0, T_1, \cdots, T_M) \propto e^{-\sum_i^M \beta_i E_i}$$

$$E_i \equiv \mathcal{H}(\Gamma_i)$$

Metropolis method:



$\mathcal{T}$ **:sequence of temperatures**

$$\mathcal{T} = \{T_1, T_0, T_2, \cdots\}$$

$$\{T_0, \Gamma_0\}, \{T_1, \Gamma_1\} \to \{T_1, \Gamma_0\}, \{T_0, \Gamma_1\}$$

$$\mathcal{T}_{01} \qquad\qquad \mathcal{T}_{10}$$

Transition probability

$$W_{\mathcal{T}_{01} \to \mathcal{T}_{10}} = \min\left(1, \frac{P(\{\Gamma_i\}; \mathcal{T}_{10})}{P(\{\Gamma_i\}; \mathcal{T}_{01})}\right)$$

$$\frac{P(\{\Gamma_i\}; \mathcal{T}_{10})}{P(\{\Gamma_i\}; \mathcal{T}_{01})} = \frac{e^{-\beta_1 E_0 - \beta_0 E_1}}{e^{-\beta_0 E_0 - \beta_1 E_1}}$$

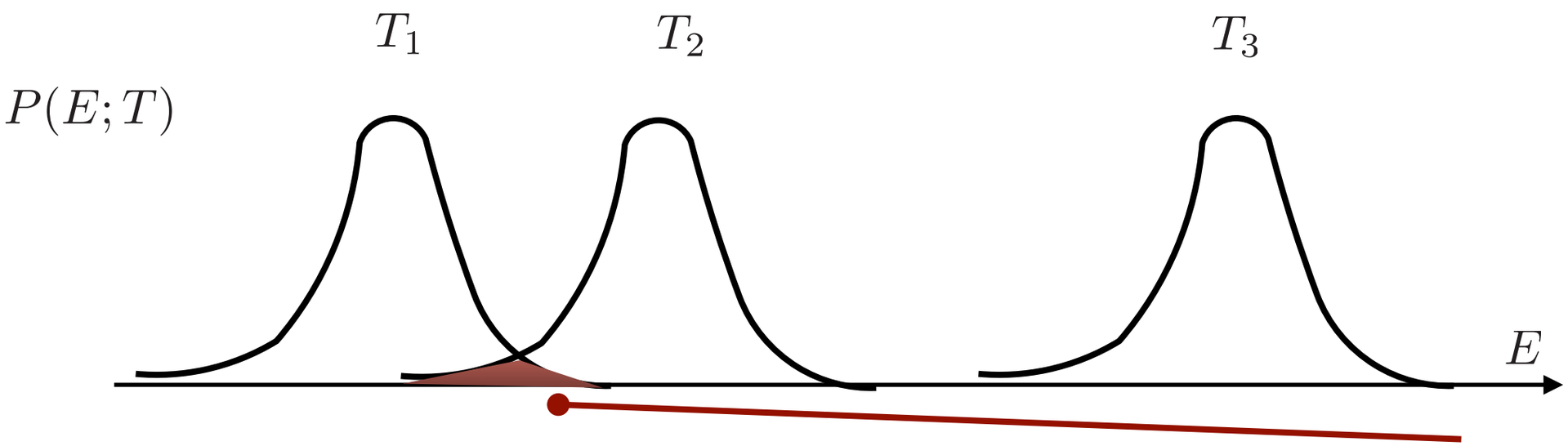
$$= e^{(\beta_0 - \beta_1)(E_0 - E_1)}$$

Select of temperature sequence

$$\frac{P(\{\Gamma_i\};\mathcal{T}_{10})}{P(\{\Gamma_i\};\mathcal{T}_{01})} = e^{(\beta_0-\beta_1)(E_0-E_1)} = \frac{P(E_1;T_0)P(E_0;T_1)}{P(E_0;T_0)P(E_1;T_1)}$$

Energy distribution at T

$$P(E;T)$$



Almost all exchange occurs the energy region of “overlap”.

$\{\Gamma_1, T_1\}, \{\Gamma_2, T_2\} \rightarrow \{\Gamma_1, T_2\}, \{\Gamma_2, T_1\}$:acceptable!

$\{\Gamma_2, T_2\}, \{\Gamma_3, T_3\} \rightarrow \{\Gamma_2, T_3\}, \{\Gamma_3, T_2\}$:almost rejected!

For efficient exchange, we have to choose a sequence of temperatures so that the energy distributions have finite overlap!

Usually we only exchange the nearest neighbor pairs of temperatures

Select of temperature sequence: Example

Suppose $C = \dfrac{dE}{dT} = \text{const.}$

$$\frac{P(\{\Gamma_i\};\mathcal{T}_{10})}{P(\{\Gamma_i\};\mathcal{T}_{01})} = e^{(\beta_0-\beta_1)(E_0-E_1)}$$

➡ Temperature sequence satisfying almost "flat" transition probability

$$(\beta_i - \beta_{i+1})(E_i - E_{i+1}) = \text{const.}$$

$$\Leftrightarrow \quad C\frac{(T_{i+1}-T_i)^2}{T_{i+1}T_i} = \text{const.}$$

$$\Rightarrow \quad T_{i+1} = \alpha T_i$$ **:Temperatures are geometric sequence!**

$$\alpha \sim 1 + O(1/\sqrt{C})$$

Important notice:

Heat capacity C is an extensive quantity: $C \sim O(N)$

➡ In order to keep finite overwrap, we need to increase temperature point M as

$$M \propto \sqrt{N} \qquad (T_{max} = T_M = \alpha^M T_{min})$$

Relaxation time of the replica exchange

In order to confirm the equilibration of the whole system,
usually we need two criterions

1. The correlation time at the highest temperature is sufficiently short,
   e.g. $\tau=O(1)$

   ➡ If a replica visits the highest temperature, it can easily change its state Γ.

2. All replicas make several (~O(10)) round trips between
   the lowest and the highest temperatures

   ➡ The ensemble at the lower temperature is in the equilibrium.

The second part determines the relaxation time of the method.

$$\tau_{\mathrm{RE}} \sim \text{round trip time}$$

* If the replica exchange is an random walk:

$$\text{round trip time} \propto M^2$$

Summary of replica exchange

Algorithm:

1. Make a temperature set $\{T_1, T_2, \ldots, T_M\}$
2. Loop n
   (1) Do MC or MD for M replicas: $\{\Gamma_1, \Gamma_2, \ldots \Gamma_M; T_1, T_2, \ldots, T_M\}$
   (2) Calculate the energies of replicas
   (3) Try replica exchange based on, e.g. Metropolis method
   - Usually we alternatively try replica exchange such as
     even n; $\{1\leftrightarrow2\},\{3\leftrightarrow4\},\{5\leftrightarrow6\},\ldots$
     odd n; $\{2\leftrightarrow3\},\{4\leftrightarrow5\},\{6\leftrightarrow7\},\ldots$
     Note: each exchange trial is independent
   (4) Observe the quantities for $\{\Gamma_1, \Gamma_2, \ldots \Gamma_M; T_1, T_2, \ldots, T_M\}$

If we already have a MC or MD programs,
it is very easy to introduce the replica exchange method!

Introduction of ALPS (for the report problem)

ALPS（Applications and Libraries for Physical Simulation)

- Set of libraries and applications for a variety of lattice models.
- Support for spin models, Hubbard model, Kondo lattice model, …
- A lot of solvers for models:
  - Classical/Quantum Monte Carlo, Exact Diagonalization, Density Matrix Renormalization Group (DMRG), Dynamical Mean Field Theory (DMFT), Time Evolving Block Decimation (TEBD), …
- We can select efficient solver for your problems.
- It can be applicable to the frontier research.

ALPS Wiki http://alps.comp-phys.org/mediawiki/index.php/Main_Page

Research topics using ALPS

(since 2015)

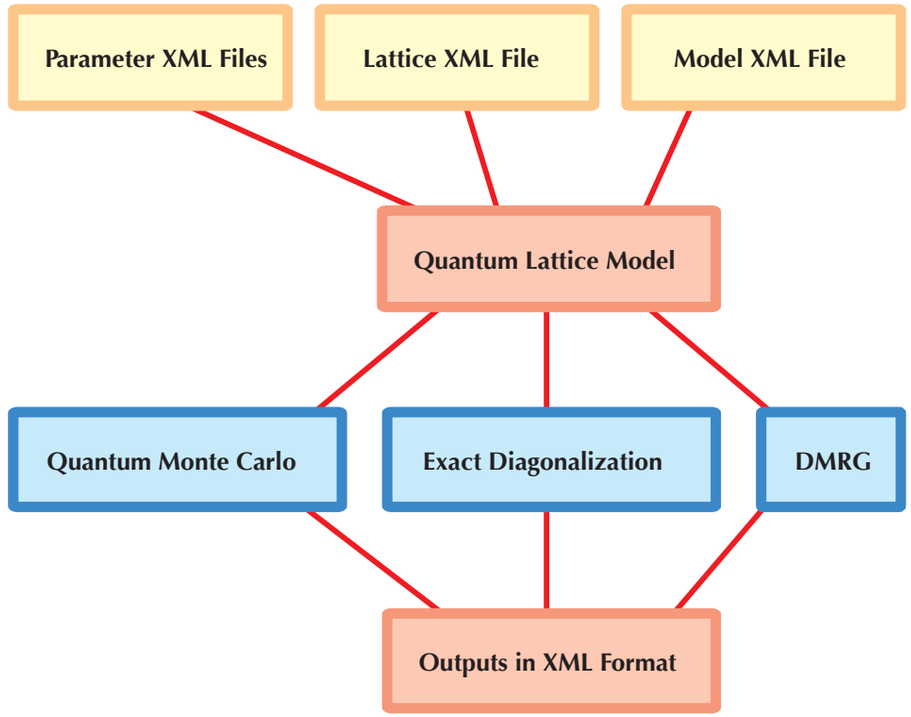
- Phase transition of ultracold atoms immersed in a BEC vortex lattice
- Entanglement entropy and topological order in resonating valence-bond quantum spin liquids
- First-order topological phase transition of the Haldane-Hubbard model
- DMFT Study for Valence Fluctuations in the Extended Periodic Anderson Model
- Static and dynamical spin correlations of the S =1/2 random-bond antiferromagnetic Heisenberg model on the triangular and kagome lattices
- Transport properties for a quantum dot coupled to normal leads with a pseudogap
- Magnetic structure and Dzyaloshinskii-Moriya interaction in the S =1/2 helical-honeycomb antiferromagnet α -$Cu_2V_2O_7$
- Mott transition in the triangular lattice Hubbard model: A dynamical cluster approximation study
- SU (N) Heisenberg model with multicolumn representations
- Superconductivity in the two-band Hubbard model
- Local Electron Correlations in a Two-Dimensional Hubbard Model on the Penrose Lattice

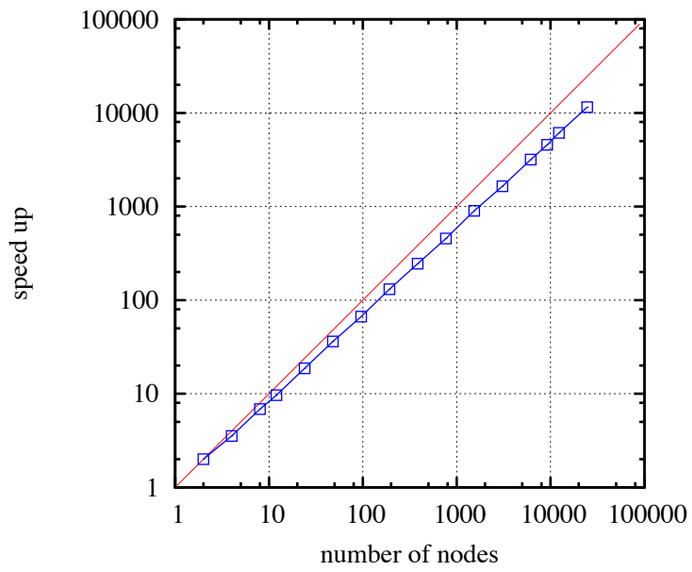
Details：http://alps.comp-phys.org/mediawiki/index.php/PapersTalks

ALPS の機能

＊MateriApps のハンズオン資料から借用

- 入出力支援
  - 格子構造, 模型は XML を用いて柔軟に指定
  - 全てのソルバーに共通した入出力形式
  - Python インターフェースを用意
    - Python から直接実行、グラフを作成
- 並列化
  - パラメータ並列のための並列化スケジューラ
  - 量子モンテカルロソルバ (looper)
    - 京で20,000ノードまで良好なスケーリング
- 競合するアプリケーション：「なし」？





MateriApps, 2013-2015. All rights reserved.

Preparation of ALPS (If you use it at ECCS)

- Login to iMac and open “Terminal” application.
- Download ALPS binaries for ECCS "alps-20160816.zip" from
  github (https://github.com/compsci-alliance/many-body-problems)
  or
  ITC-LMS (https://itc-lms.ecc.u-tokyo.ac.jp/portal/login)
  Probably, it will be automatically decompressed.
  (If not, double click the zip file)
- Move the folder to home (or your preferable place)
  cd
  mv Downloads/alps-20160816 .
- Run configure file
  . alps-20160816/bin/alpsvars.sh
  Note! Don’t forget type “.” before “alps-20160816/bin/alpsvars.sh”
  You need this configure every time at the beginning of new terminals.
- Set python environment for python2.7
  pyenv shell anaconda-4.0.0
  This setting also disappears after logout. In order to set python environment for the directory "permanently", please see FAQ page of the lecture.
- Check simplemc and spinmc
  simplemc --help
  spinmc --help

Correct output of spinmc

```
Allowed options:
  --help                          produce help message
  -l [ --license ]                print license conditions
  --mpi                           run in parallel using MPI
  --checkpoint-time arg (=1800) time between checkpoints
  --Tmin arg (=60)                minimum time between checks whether a
                                  simulation is finished
  --Tmax arg (=900)               maximum time between checks whether a
                                  simulation is finished
  -T [ --time-limit ] arg (=0)  time limit for the simulation
  --Nmin arg (=1)                 minimum number of CPUs per simulation
  --Nmax arg (=2147483647)        maximum number of CPUs per simulation
  --write-xml                     write results to XML files
  --input-file arg                input file

Generic classical Monte Carlo program using local or cluster updates
  available from http://alps.comp-phys.org/
  copyright(c) 1999-2007 by Matthias Troyer <troyer@comp-phys.org>
                            Mathias Koerner <mkoerner@comp-phys.org>
 for details see the publication:
 A.F. Albuquerque et al., J. of Magn. and Magn. Materials 310, 1187 (2007).

using the ALPS parallelizing scheduler
  copyright (c) 1994-2006 by Matthias Troyer <troyer@comp-phys.org>.
  see Lecture Notes in Computer Science, Vol. 1505, p. 191 (1998).

based on the ALPS libraries version 2.2.b4
  available from http://alps.comp-phys.org/
  copyright (c) 1994-2013 by the ALPS collaboration.
  Consult the web page for license details.
  For details see the publication:
  B. Bauer et al., J. Stat. Mech. (2011) P05001.
```

Download of example (tutorial) files

- From ITC-LMS: https://itc-lms.ecc.u-tokyo.ac.jp/portal/login
  or
  github: https://github.com/compsci-alliance/many-body-problems

  Download and decompress the example files "ALPS_examples.zip"

- Move the folder ALPS_examples to home (or your preferable place) . In the case of ECCS iMac,
  cd
  mv Downloads/ALPS_examples .

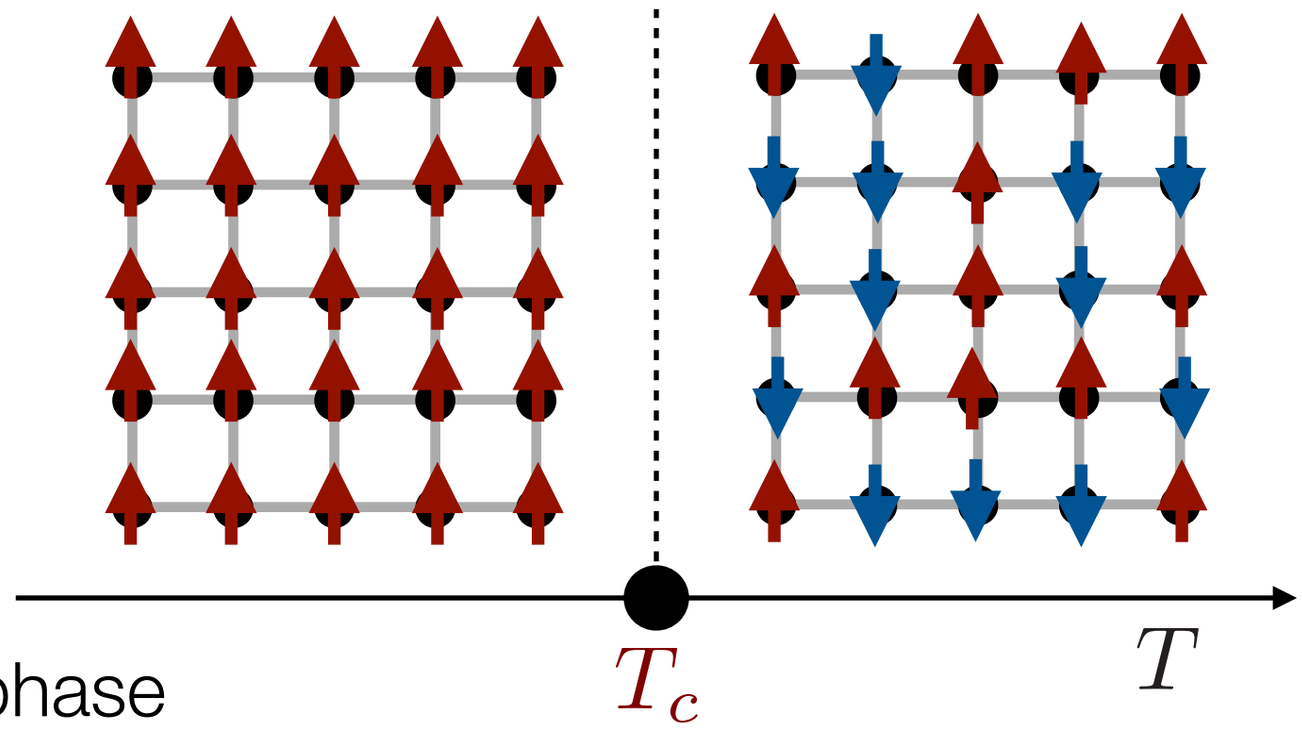
Exercise (not a report)

- Square lattice Ising model

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

  - Continuous transition at $T=T_c$,

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



  - $T>T_c$：Para magnetic phase
  - $T<T_c$：Ferro magnetic phase

- By Monte Carlo simulation (Metropolis algorithm), see the phase transition!

Simulation by **simplemc**

- **simplemc**: classical Monte Carlo simulator using Metropolis Algorithm
- Move to sample_alps
  - *cd*
  - *cd Report1/sample_alps/simplemc*
- *Transform parameter file to XML format*
  - *parameter2xml parm9a*
- Run simulation（It takes approximately five minites）
  - *simplemc parm9a.in.xml*
- Plot the results（Specific heat, Energy, square of the Magnetization）
  - *python plot9a.py*
  - （If you close three windows (graphs), python script will stop)

Explanation of parameter file: parm9a

```
LATTICE="square lattice"
J=1
ALGORITHM="ising"
SWEEPS=65536
L=8
{ T=5.0 }
{ T=4.5 }
{ T=4.0 }
{ T=3.5 }
{ T=3.0 }
{ T=2.9 }
{ T=2.8 }
{ T=2.7 }
...
```

Define lattice: if you write “simple cubic lattice” here, you can simulate 3D

Define interaction: J > 0 ferromagnetic

Define model: ising , xy, heisenberg

Define Monte Carlo steps:

1 MC step = trial of flips for all spin

＊THERMALIZATION

In order to reduce the effect of initial condition
we do THERMALIZATION step MC calculation
without observation.
If you don’t explicitly write it, it is automatically set to 1/8 of SWEEPS

Lattice size：L × L

List of temperatures:

- Parameters in {}: parameter for 1 simulation
  - MC simulation is repeated for # of {}.
- Parameters outside of {}:
  common parameter for all calculation after them.

Explanation of plotting file: plot9a.py

Python script

Extract the result from the data files: **pyalps.loadMeasurements**

Setting for the X and the Y axes: **pyalps.collectXY**

Plotting: **matplotlib**

```
data = pyalps.loadMeasurements(pyalps.getResultFiles(prefix='parm9a'),
    ['Specific Heat', 'Magnetization Density^2', 'Energy Density'])
for item in pyalps.flatten(data):
    item.props['L'] = int(item.props['L'])

magnetization2 = pyalps.collectXY(data, x='T', y='Magnetization Density^2', foreach=['L'])
magnetization2.sort(key=lambda item: item.props['L'])

pyplot.figure()
alpsplot.plot(magnetization2)
pyplot.xlabel('Temperture $T$')
pyplot.ylabel('Magnetization Density Squared $m^2$')
pyplot.legend(loc='best')
```

Simulation by **spinmc**

- **spinmc**: classical Monte Carlo simulator using Metropolis Algorithm or **Cluster Algorithm**
- Move to sample_alps
  - *cd*
  - *cd Report1/sample_alps/spinmc*
- *Transform parameter file to XML format*
  - *parameter2xml parm9a*
- Run simulation
  - *spinmc --Tmin 5 parm9a.in.xml*
    Note. “--Tmin n” set minimum time to check weather a simulation is finished.
    In this example, default value n=60 is too long, Thus, here I recommend to use
    --Tmin 5 or --Tmin 1
- Run evaluation (Different from **simplemc**, we need explicit evaluation for several physical quantities)
  - spinmc_evaluate parm9a.task*.out.xml
- Plot the results（Specific heat, Energy, square of the Magnetization）
  - *python plot9a.py*
  - （If you close three windows (graphs), python script will stop)

Explanation of parameter file: parm9a

```
LATTICE="square lattice"
J=1
Model="Ising"
UPDATE="local"
THERMALIZATION=8192
SWEEPS=65536

L=8
{ T=5.0 }
{ T=4.5 }
{ T=4.0 }
{ T=3.5 }
{ T=3.0 }
{ T=2.9 }
...
```

Define lattice: if you write “simple cubic lattice” here, you can simulate 3D

Define interaction: J > 0 ferromagnetic

Define model: Ising , XY, Heisenberg, Potts (q=3,4,10)

Define algorithm: “local” (Metropolis) or “cluster”

Define Monte Carlo steps:

1 MC step = trial of flips for all spin

THERMALIZATION

In order to reduce the effect of initial condition
we do THERMALIZATION step MC calculation
without observation.
In the case of **spinmc**, we need to set it **explicitly**!

Lattice size：L × L

List of temperatures:

- Parameters in {}: parameter for 1 simulation
  - MC simulation is repeated for # of {}.
- Parameters outside of {}:
  common parameter for all calculation after them.

Explanation of plotting file: plot9a.py

Python script

Extract the result from the data files: **pyalps.loadMeasurements**

Setting for the X and the Y axes: **pyalps.collectXY**

Plotting: **matplotlib**

```
data = pyalps.loadMeasurements(pyalps.getResultFiles(prefix='parm9a'),
    ['Specific Heat', 'Magnetization^2', 'Energy Density'])
for item in pyalps.flatten(data):
    item.props['L'] = int(item.props['L'])

magnetization2 = pyalps.collectXY(data, x='T', y='Magnetization^2', foreach=['L'])
magnetization2.sort(key=lambda item: item.props['L'])

pyplot.figure()
alpsplot.plot(magnetization2)
pyplot.xlabel('Temperture $T$')
pyplot.ylabel('Magnetization Density Squared $m^2$')
pyplot.legend(loc='best')
```

Report problem 1: Auto-correlation functions

- Calculate auto-correlation functions of relevant observables as follows
  1. Select your model.
     - For example, you can choose Ising model on square lattice.
  2. Perform MCMC or MD simulation for the model.
     - In the case of Ising model, you may perform MCMC simulation.
  3. Calculate auto-correlation functions at least two observables in equilibrium.
     - For Ising model, they may be magnetization and energy.
  4. Discuss behaviors of auto-correlation functions (and correlation time) by varying temperatures (or similar relevant quantity) and system sizes.
     - Try at least three temperatures and two system sizes.
       For Ising model, I recommend, $T<T_c$, T_c, and $T>T_c$.
       (Note that, at T<Tc, we need exponentially large time to reverse the magnetization with local update algorithm.)
     - It might be useful to see integrated correlation time $\tau = \int_0^\infty \frac{C(t)}{C(0)} dt \sim \sum_{t=0}^{T} \frac{C(t)}{C(0)}$.
  5. (optional) Change algorithm and compare correlation times among them.
     - For Ising model, you may consider, Metropolis, heat bath or cluster algorithms.

Report problem 1: Tips

- For MCMC or MD simulation, you can use
  - Your own code
  - Open source softwares
  - My sample code (python) (In Report1.zip at ITC-LMS and github)
    Ising.ipynb or Ising.py
    - In order to run the sample codes, you need
      numpy, matplotlib, and numba packages in addition to the python2.7
    - See also the header of the code and FAQ page.
- In order to obtain correct auto-correlation function, we need to care about "thermalization" (initial state dependence) mentioned in the lecture
  - In the case of the sample codes,
    When you increase the system size
    or
    when you change the temperature,
    you may need to increase the "thermalization" parameter which means MC steps discarding before calculating auto-correlation functions.

Report problem 2:

- Try to simulate the following systems (e.g. by editing the parameter file parm9a of ALPS)
  1. Square lattice Ising model of larger system sizes than L=32 (e.g. L=48, 64, …)
     - Plot energy, magnetization^2 and specific heat as functions of temperature.
       Note: Temperature range should contain the critical temperature.
     - Discuss that the relationship among SWEEP, L, and error bar of the above quantities.
     - Plot the “binder ratio of magnetization” and estimate crossing point of them.
       And then, compare it to the true critical point.
       - It can be done by adding “Binder Ratio of Magnetization” (in the case of simplemc) to **pyalps.loadMeasurements** in **plot9a.py** and edit properly **pyalps.collectXY**
     - (optional) Try finite size scaling of the binder ratio, and specific heat.
     - (optional) Try cluster algorithm and compare error bars with those in local updates.
  2. Try to simulate different models
     - Select your model, eg. cubic lattice Ising model, (cubic or square lattice) xy model, Heisenberg model, Potts model, ...
     - Perform the same tasks with the case of the above square lattice Ising model.
       - Note: In the case of 3D, you may need longer time to simulate the model.
         Thus, the largest size becomes smaller than that of 2D.
       - Note: For some models, there is no phase transition in 2D.

Report problem 2: Additional comments1

- You can use **simplemc, spinmc, free applications** or **your own code**
  - If you use **simplemc** or **spinmc**, please attach input files to the report.
  - When you write a report by using **an application not included in ALPS,** please add information of the code (source code or url of the app.) in addition to the information of inputs to reproduce the results.

- Note that several quantities have different names in **simplemc** and **spinmc**.

	simplemc	spinmc
<E>	Energy Density	Energy Density
<M^2>	Magnetization Density^2	Magnetization^2
<C>	Specific Heat	Specific Heat
binder ratio	Binder Ratio of Magnetization	Binder Cumulant

* In the case of *simplemc*, the definition of Binder ratio is reversed from that of *spimc,* which was presented in the lecture No. 4.

Report problem 2: Additional comments2

- The ALPS tutorials are useful to learn how to use the ALPS. **Especially mc-01,02 and 07 are relevant.**
  - http://alps.comp-phys.org/mediawiki/index.php/ALPS_2_Tutorials:Overview

- If you install mac-osx binary, please use **“Prerelease 2.3.0” .**
  - **It exists only in “English” page. (In “Japanese” page, 2.3.0 does not exist!)**
  - Previous versions may not contain **simplemc.**

- If you install ALPS through mac-osx binary or MacPorts, “python” command appeared in previous exercises should be replaced by “**alpspython**”.

Report problem 2: Tips

- If you change the name of parameter file, you also need to edit the corresponding part of the plotting script.
- If you want to see the numbers directly, you can use python script **textout9a.py .**

  If you output the numbers to a file:

  python textout9a.py > filename.txt

  (If you use the script for the binder ratio or other quantities, you need to edit it.)

- Note: if SWEEP is too small, MCMC can not correctly sample the equilibrium ensemble!. If you change the parameter, you should check the SWEEP dependence of the results.
  (ALPS tutorial MC-01 might help you to check convergence of the results)

Report problem 3 (optional)

- Write a comment to the lecture of classical many body systems, e.g.,
  - How do you feel the contents of the lecture?
    - Too easy, Boring, Too difficult, Too biased to the interest of the lecturer, …
  - What topics do you want to learn if you have the next chance?
  - Do you have any idea to improve the quality of the lecture?
    - Tutorial using computers, Use black board, …

Deadline

- Submit your report through the system of ITC-LMS
  - The deadline is 2018 July. 31st.
  - In case you cannot use ITC-LMS, please submit it by e-mail.
    - I will send reply when I receive your reports.
      - If you will not receive any response, please contact me.
- If you have any troubles or questions, please freely ask me
  - at the future lectures,
  - by e-mail: t-okubo@phys.s.u-tokyo.ac.jp
  - or come to my office **Sci. Bldg. #1 940.**
    (It is better to get an appointment by e-mail.)

References (books)

- “A Guild to Simulations in Statistical Physics” D.P. Landau and D. Binder, Cambridge University Press.
- “Computational Physics”, J. Thijssen, Cambridge University Press.
  （「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.
- 「分子シミュレーション」上田顯著、裳華房.