

Computational Science for many-body problems

- This class is from 15:10 to 16:40 (90 min.).
- During the lecture, please mute your microphone and turn off your camera.
- When you have questions or comments, please use the raise hand button or write a message on the chat box.
 - After the lecturer allow you to unmute the microphone, please ask your questions or comments.
 - Questions in Japanese are also welcome.
- If you have any troubles, please let us know on the chat box.

分子動力学法とその応用

Molecular Dynamics Simulation and Its Application

理学系研究科 物理学専攻 大久保 豪

Contents of No. 4

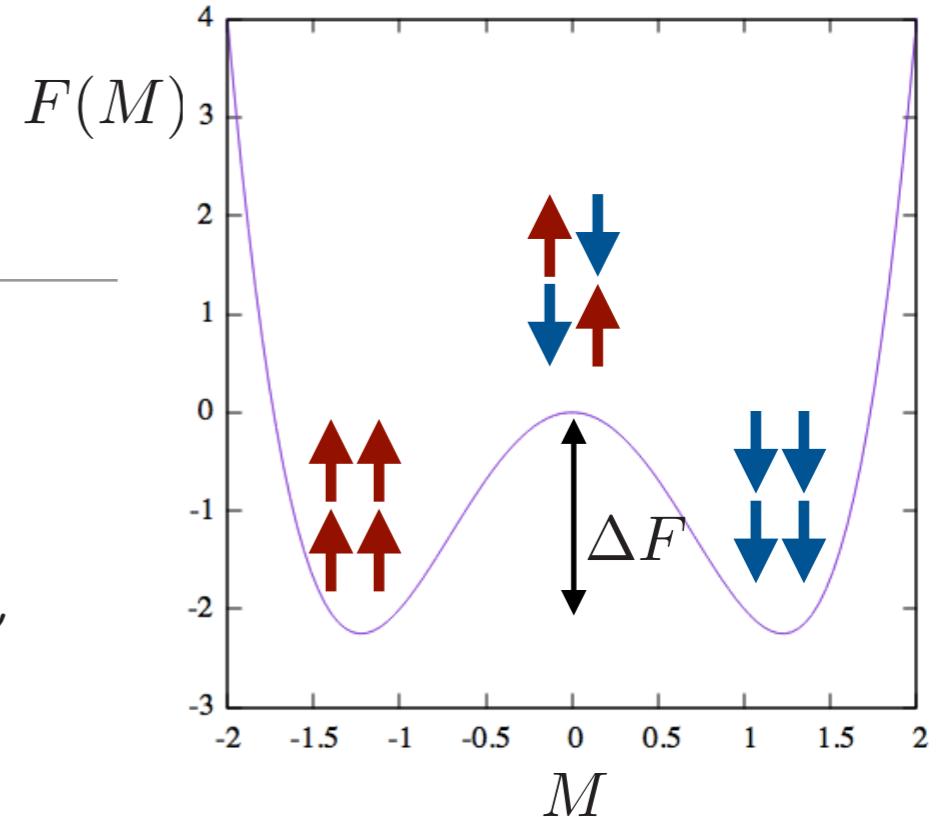
- Standard Monte Carlo method
 - Monte Carlo integration
 - Importance sampling and Markov Chain Monte Carlo (MCMC)
 - Examples of MCMC
 - Metropolis-Hastings sampling
 - Heat-bath sampling (Gibbs sampling)
- Application to classical spin systems
 - Local update by using Metropolis sampling
 - Global update as the cluster update
- Computational Science using Monte Carlo method
 - Important tips to obtain reliable results
 - Application and analysis in the case of critical phenomena

Contents

- Basics of MD simulation
 - Newtonian equation, purpose of MD simulation
 - Examples of numerical integrations
- NVE ensemble: standard MD simulation
 - Symplectic integral
- Control temperature and pressures (**quick review**)
 - Velocity scaling and Nosé-Hoover method
 - Andersen method for pressure

Application to Classical spin system

Free energy landscape



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”.

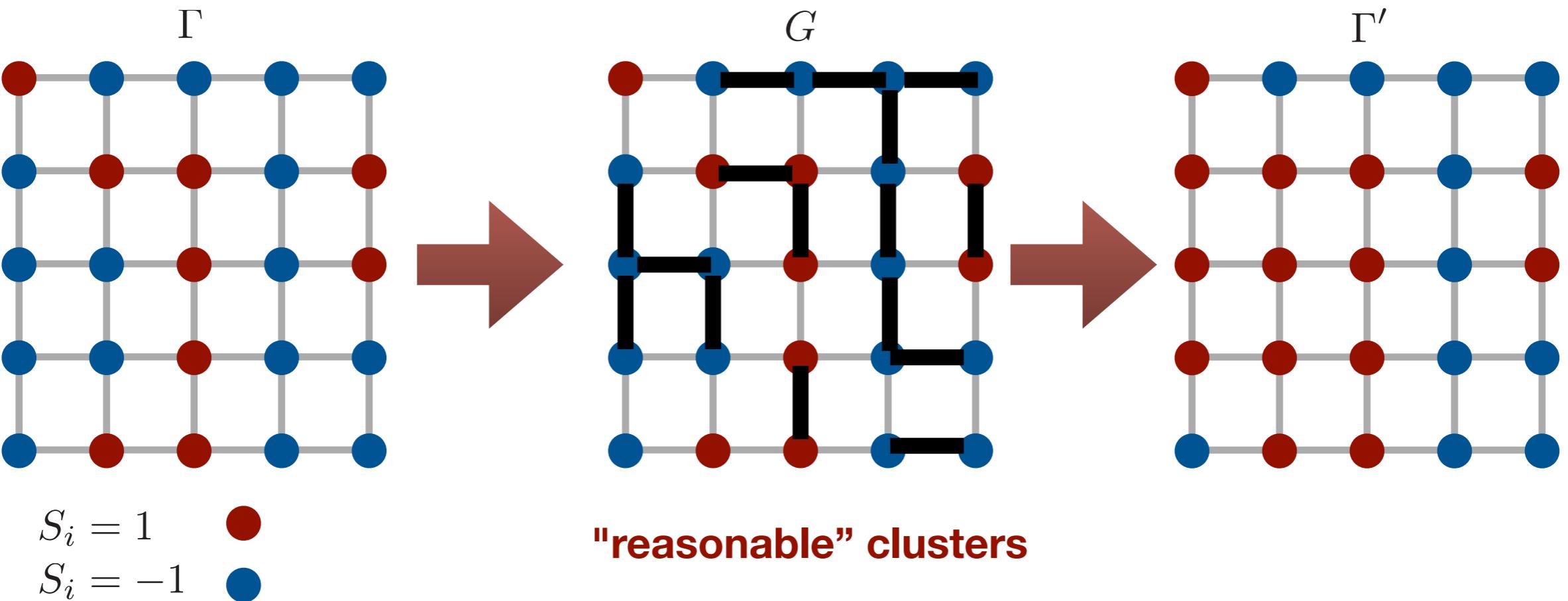
Simultaneous change of spins in “large cluster”

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).

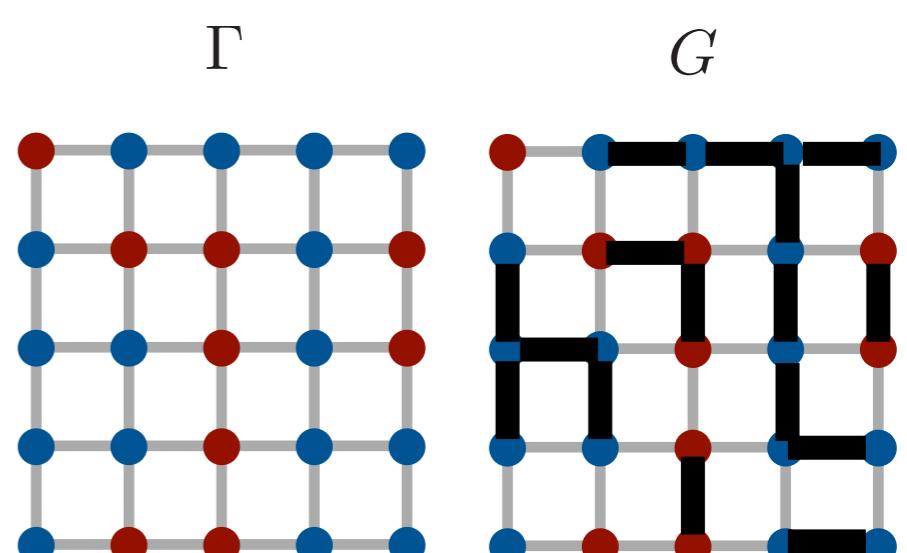
$$e^{\beta J S_i S_j} = \sum_{g=0} e^{-\beta J} + \sum_{g=1} \delta_{S_i, S_j} (e^{\beta J} - e^{-\beta J}) = \sum_{g=0,1} w(g, S_i, S_j)$$

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

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

$$G = \{g_{i,j}\}$$

$$\Gamma = \{S_i\}$$



$S_i = 1$ ●
 $S_i = -1$ ●

$g_{i,j} = 1$ —
 $g_{i,j} = 0$ —

Markov chain in extended (G, Γ) space

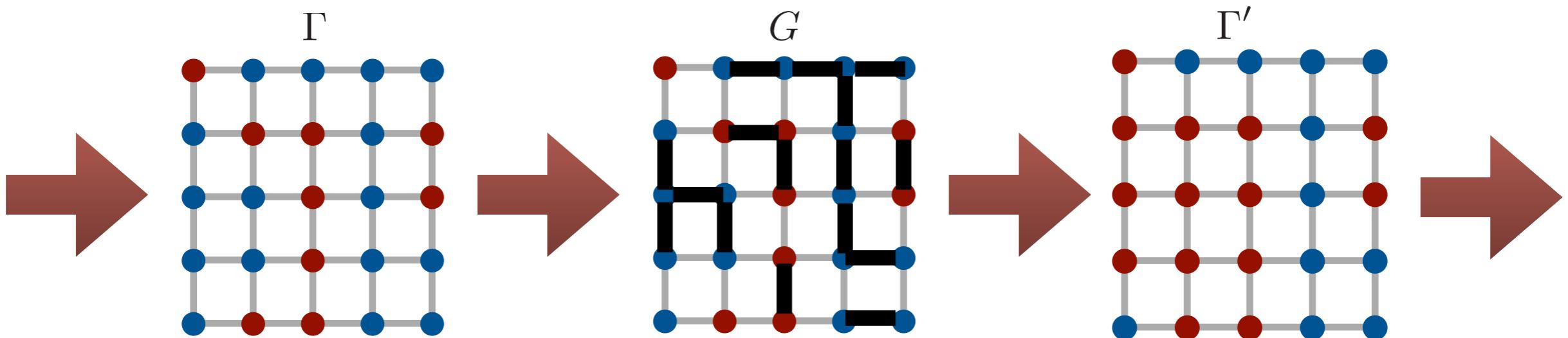
$$Z = \sum_G \sum_{\Gamma} \prod_{\langle i,j \rangle} w(g_{i,j}, S_i, S_j) = \sum_G \sum_{\Gamma} W(G, \Gamma)$$

We consider to update Γ and G alternatively:

$$\cdots \rightarrow (G_{t-1}, \Gamma_t) \rightarrow (\textcolor{red}{G}_t, \Gamma_t) \rightarrow (G_t, \Gamma_{t+1}) \rightarrow (\textcolor{red}{G}_{t+1}, \Gamma_{t+1}) \rightarrow \cdots$$

This update can be symbolically written as

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



Markov chain in extended (G, Γ) space

$$w_{(S_i, S_j) \rightarrow 0} = \begin{cases} 1 & (S_i \neq S_j) \\ e^{-2\beta J} & (S_i = S_j) \end{cases}$$

We assign the transition probabilities as follows:

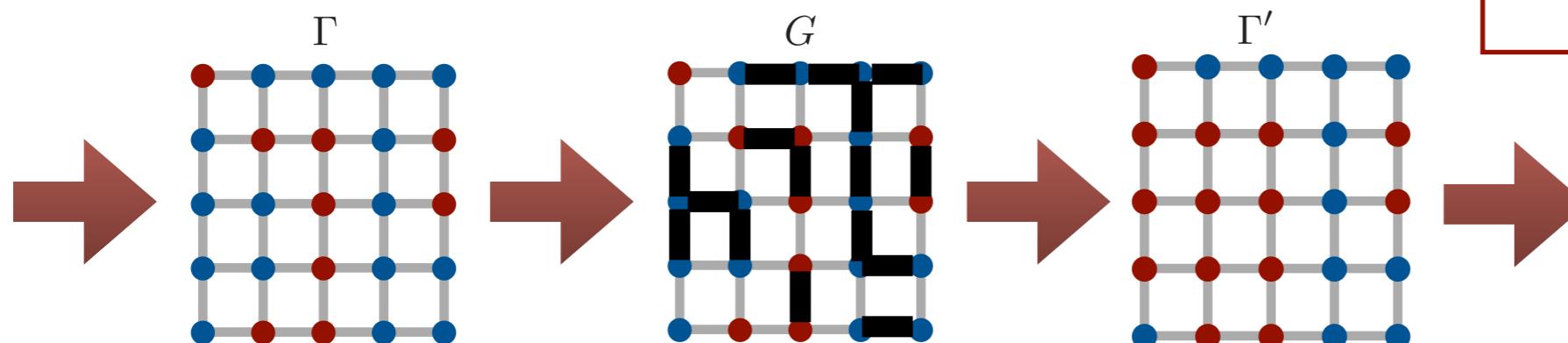
Transition probabilities (as like the heat bath)

$$\Gamma \rightarrow G' : W_{(G, \Gamma) \rightarrow (G', \Gamma)} = \frac{W(G', \Gamma)}{\sum_{G''} W(G'', \Gamma)} = \prod_{\langle i, j \rangle} w_{(S_i, S_j) \rightarrow g_{ij}}$$

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

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 $\underline{\Gamma_0 = (S_1, S_2, \dots, S_N)}$

loop t

 loop $\langle i, j \rangle$

- 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 $\langle i, j \rangle$

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

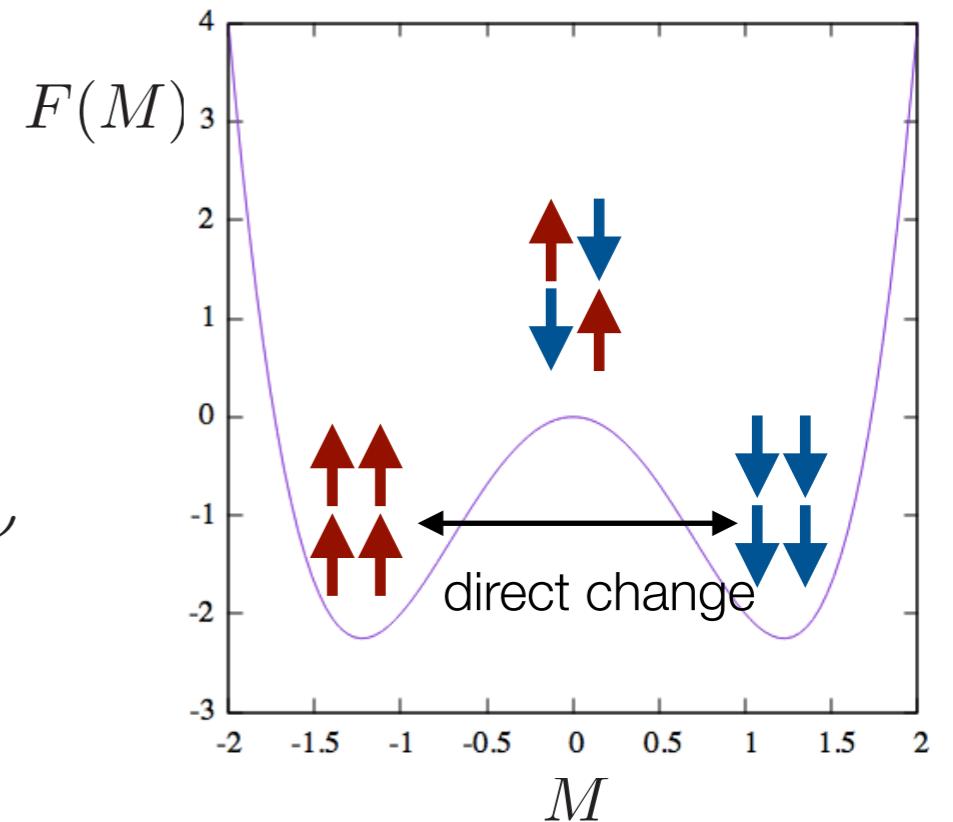
(Under a finite magnetic field, we need to modify the probability.)

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”

Free energy landscape

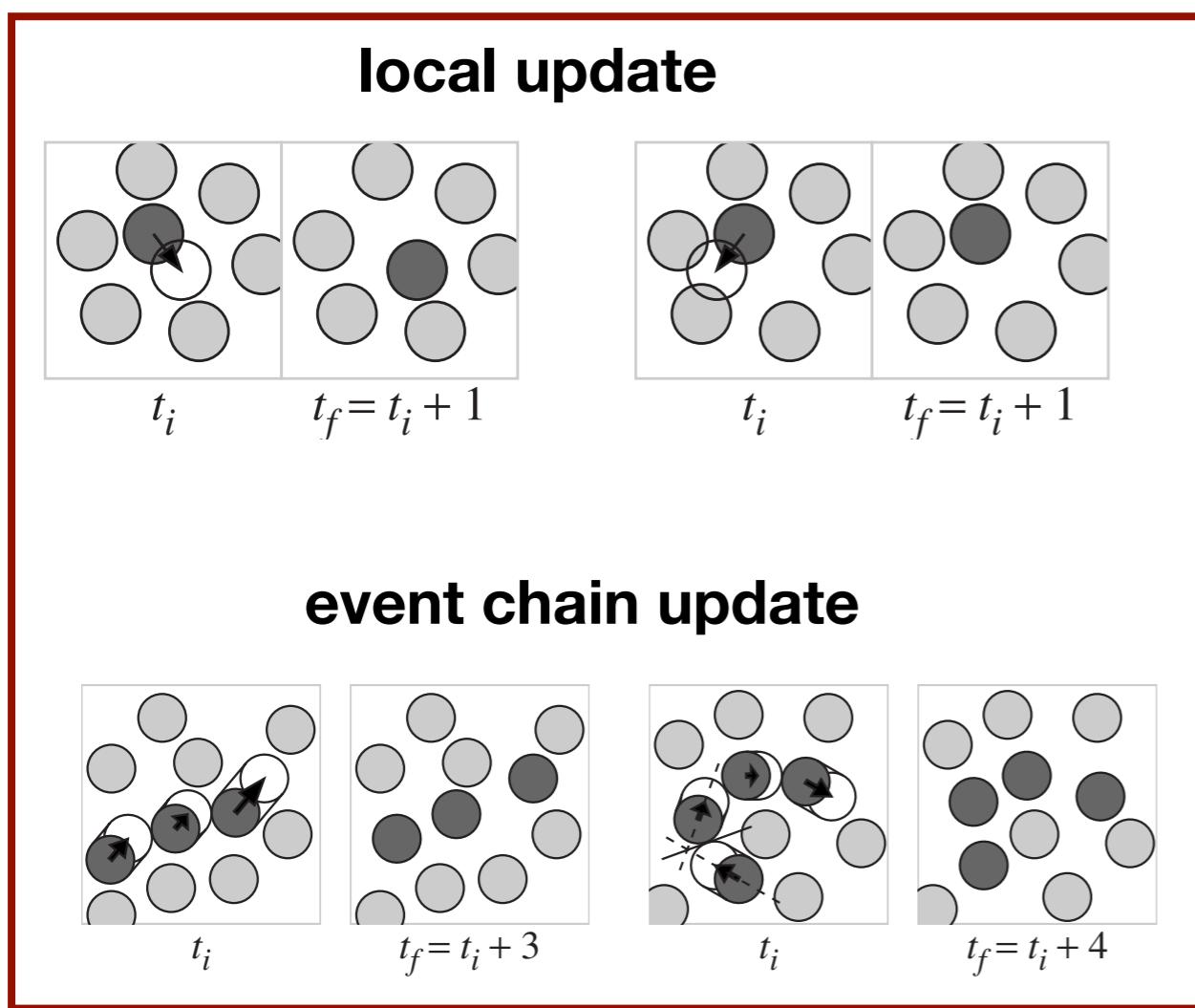


*Linear size of cluster $\sim \xi$

Event-chain Monte Carlo

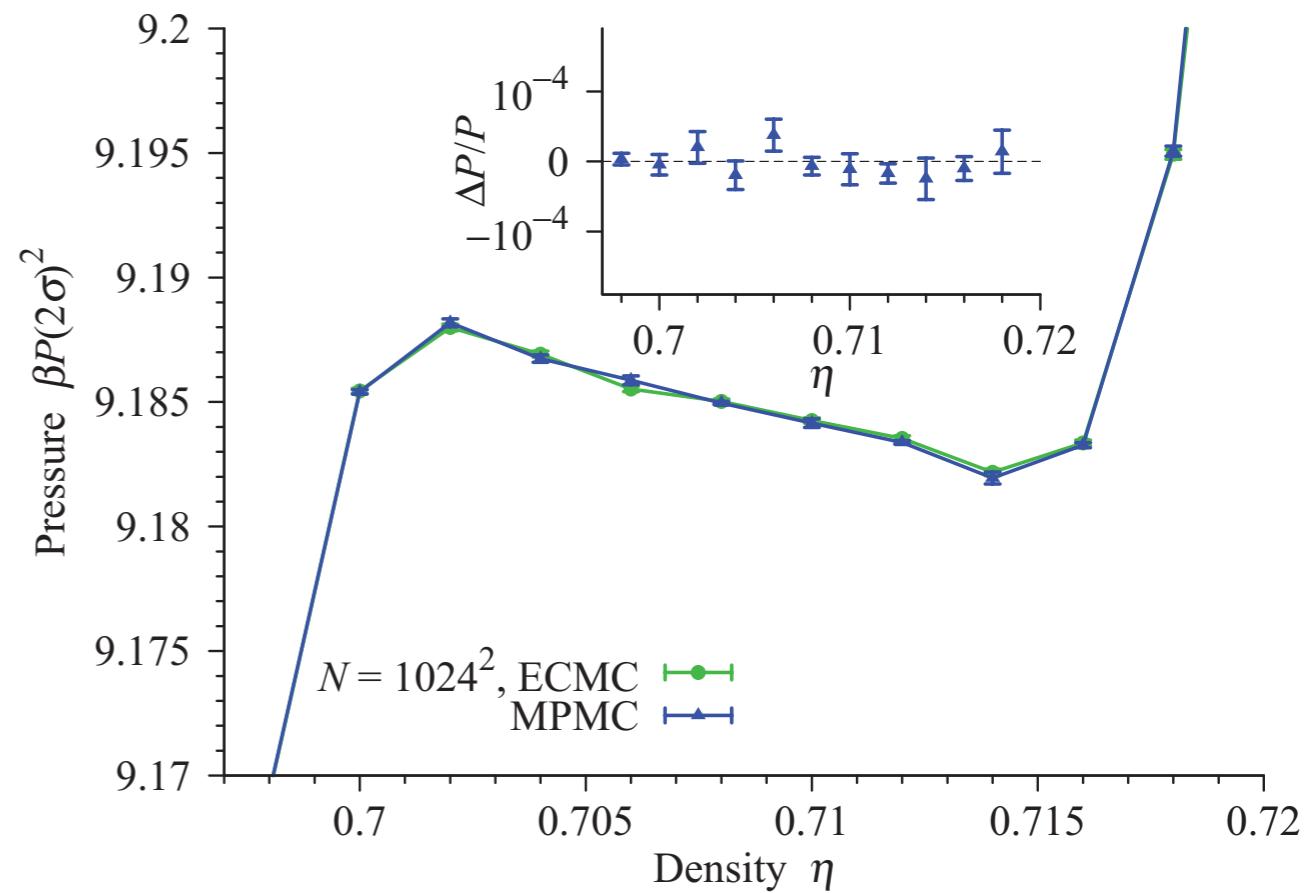
A "global" update for particle system (hard spheres)

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



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

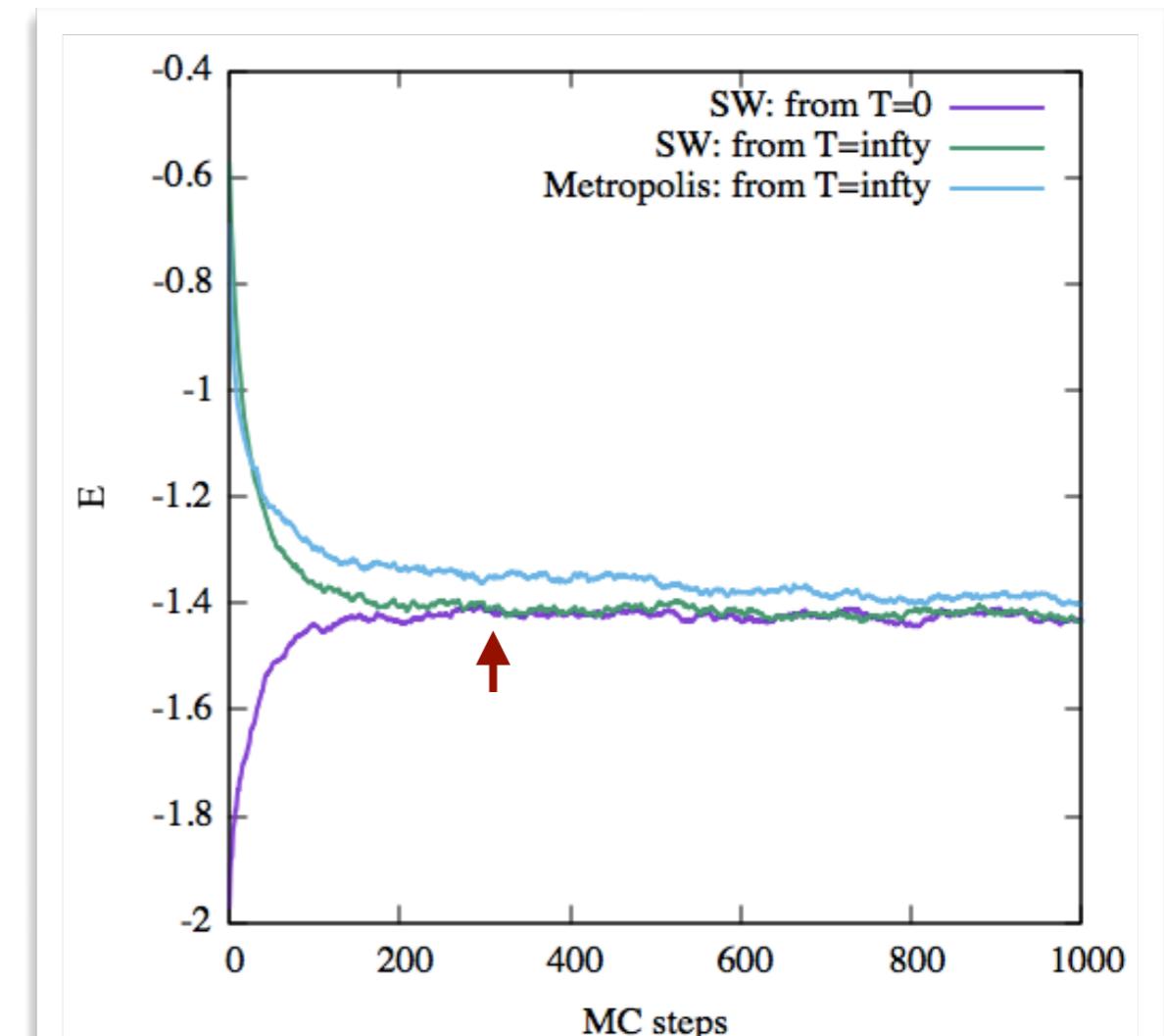
Important tips for real calculations 1

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 initial several MC steps
- Change MC steps and compare results
- Change initial state
-



Important tips for real calculations 2

We need to estimate the **statistical errors**.

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

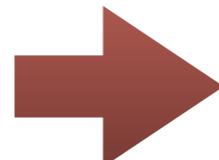
→ Standard error: $\epsilon^2 = \langle \bar{A}^2 \rangle - \langle \bar{A} \rangle^2$ $\epsilon \propto \sqrt{\frac{1 + 2\tau}{T}}$

Maximum likelihood estimation for standard error

Prepare “independent” M samples for \bar{A} : $\{\bar{A}_1, \bar{A}_2, \dots, \bar{A}_M\}$

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

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



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

Example: Application to critical phenomena

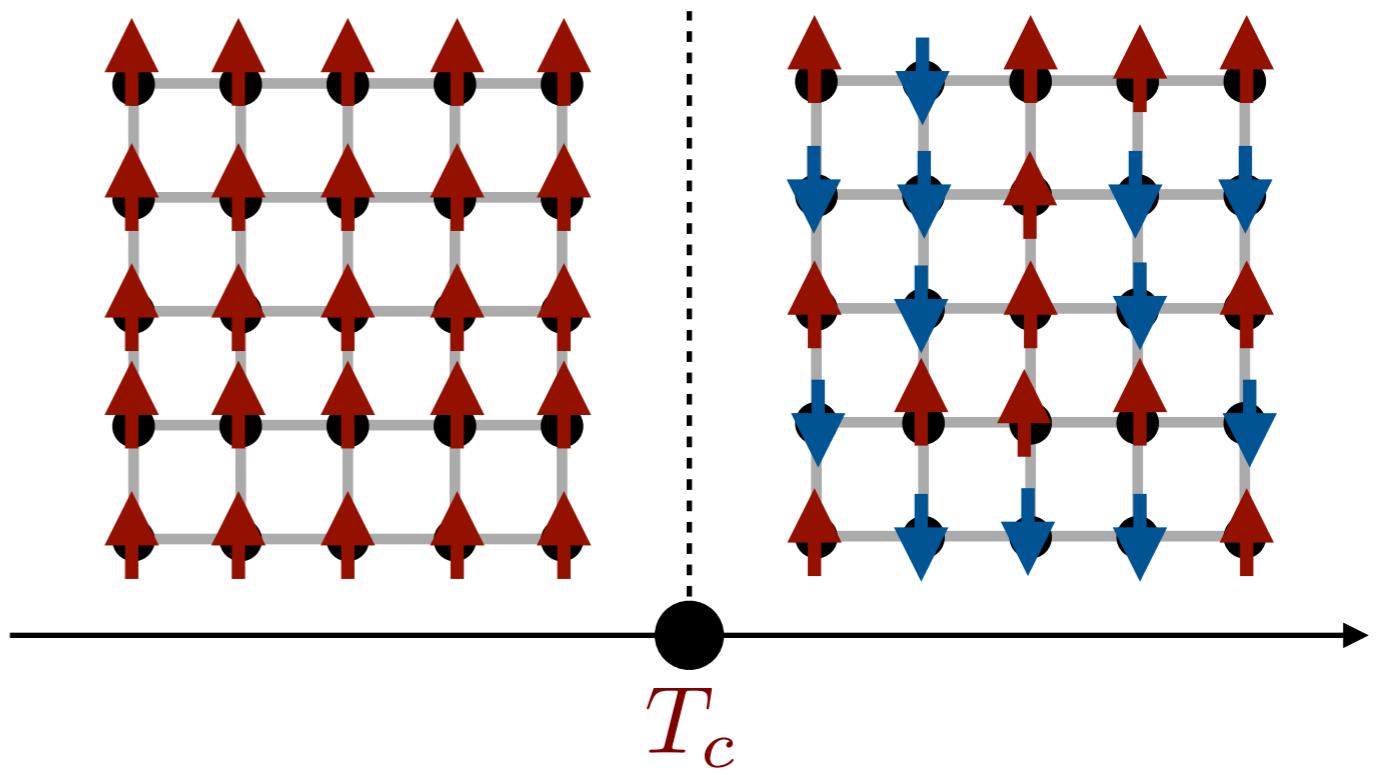
- Square lattice Ising model

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

- Continuous phase transition at $T=T_c$

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

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



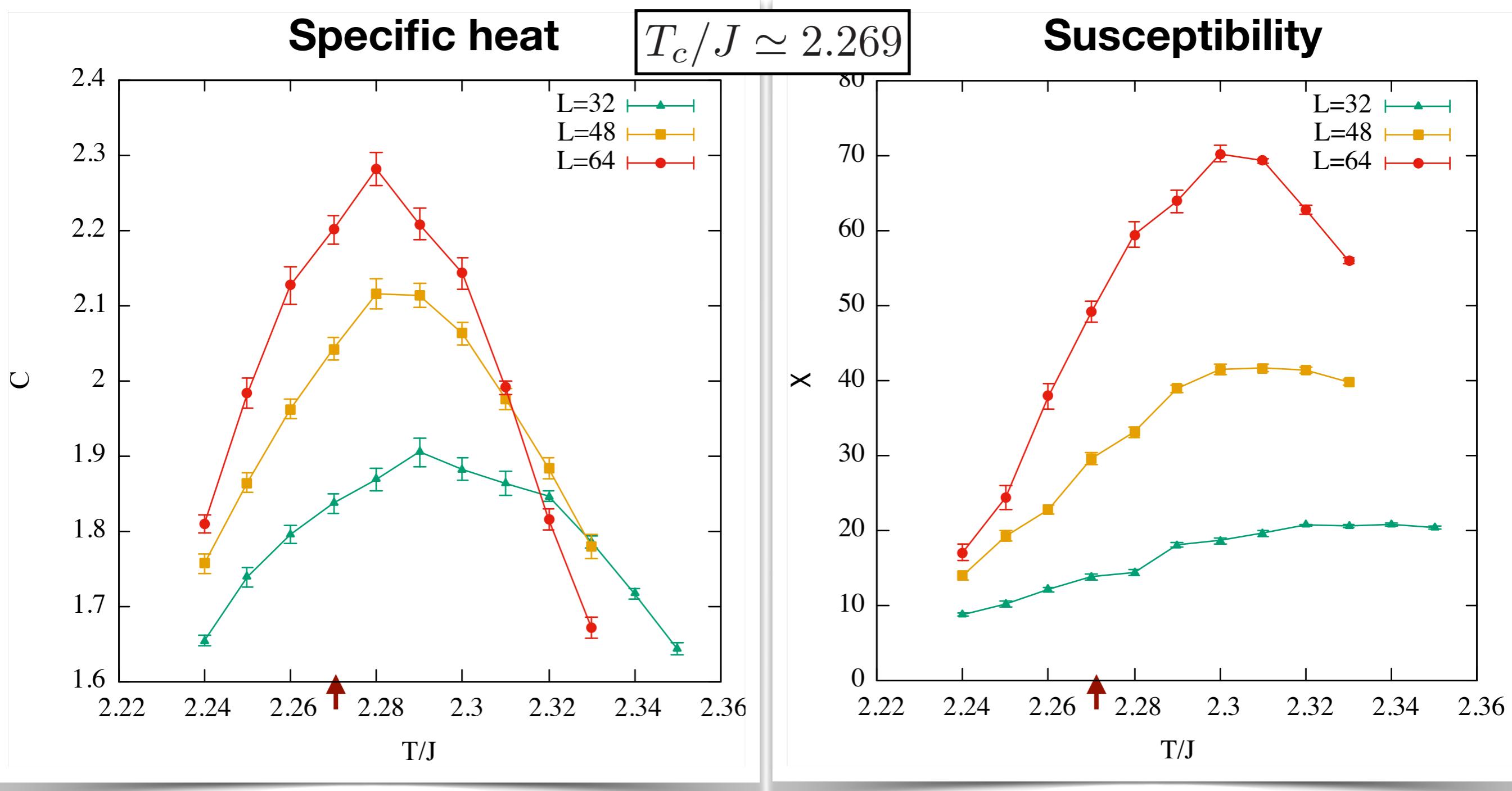
- Monte Carlo Simulations

- Using spinmc in **ALPS**: Simulator for classical spin system by MCMC
- My sample code

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**.

Calculated data (ALPS tutorial 7b)



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

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

Data analysis: Finite size scaling (outline)

Near the critical point (transition temperature):

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

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

$$t = T - T_c$$

$$y_t, y_h : \text{scaling exponent} \quad \longleftrightarrow \quad y_t = 1/\nu, \quad y_h = (d + \gamma/\nu)/2$$

By taking derivatives, we see

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

Physical quantity obeys common scaling function independent of L .

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

x : scaling dimension

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

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

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

Data analysis: Finite size scaling (outline)

Similarly, the energy and the specific heat obey:

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

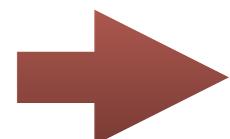
$$C = \frac{\partial^2 f}{\partial T^2} = L^{2y_t-d} g_C(tL^{y_t}) = L^{2/\nu-d} g_C(tL^{1/\nu}) = (L^{\alpha/\nu} g_C(tL^{1/\nu}))$$

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

Scaling form of general quantities are

$$O = L^{-x_o} g_o(tL^{1/\nu})$$

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: $\langle M^2 \rangle$

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

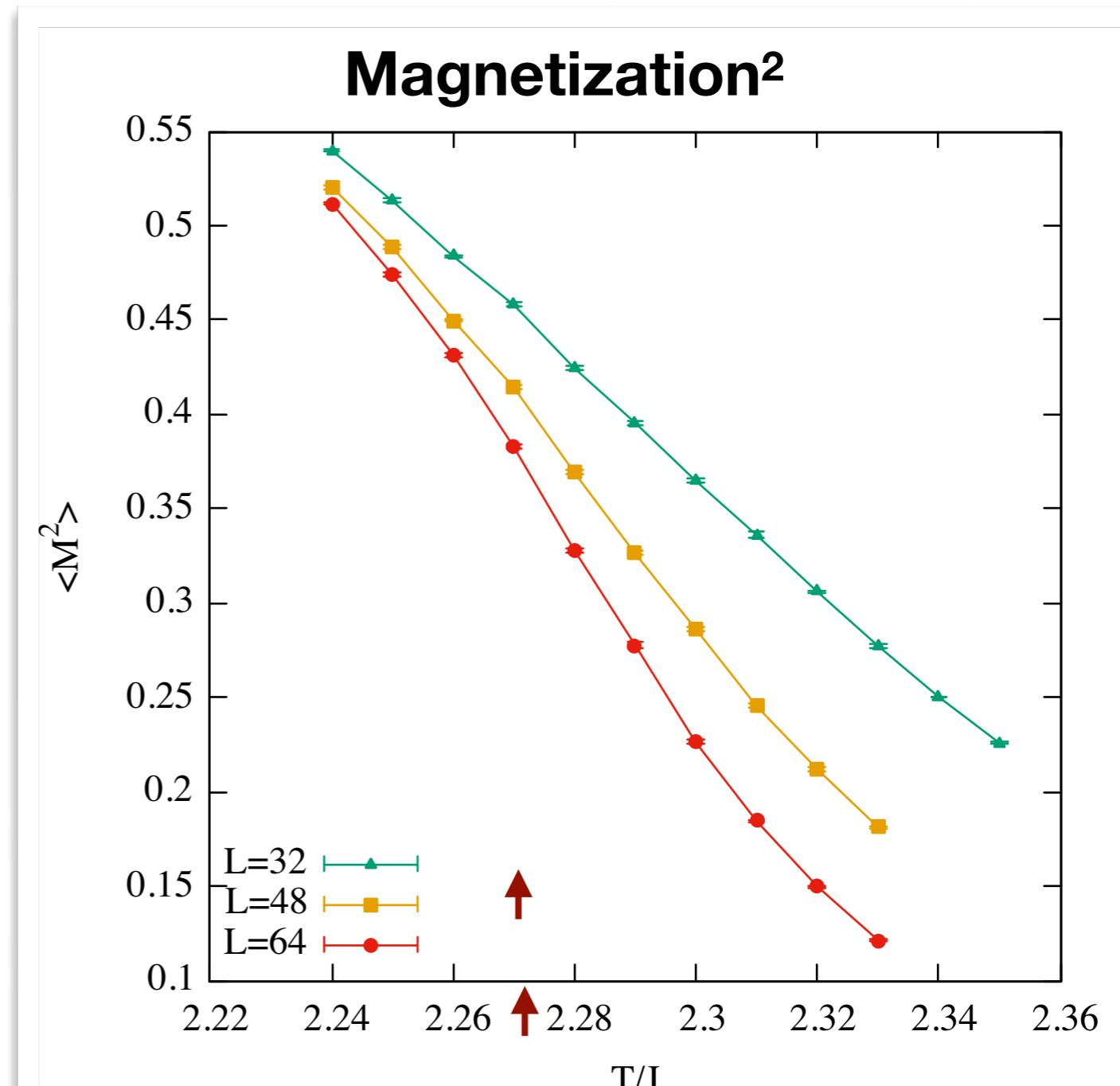
(Q. We cannot use $\langle M \rangle$. Why?)

→ 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

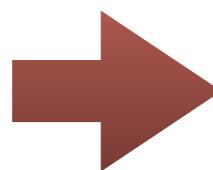
Binder ratio

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

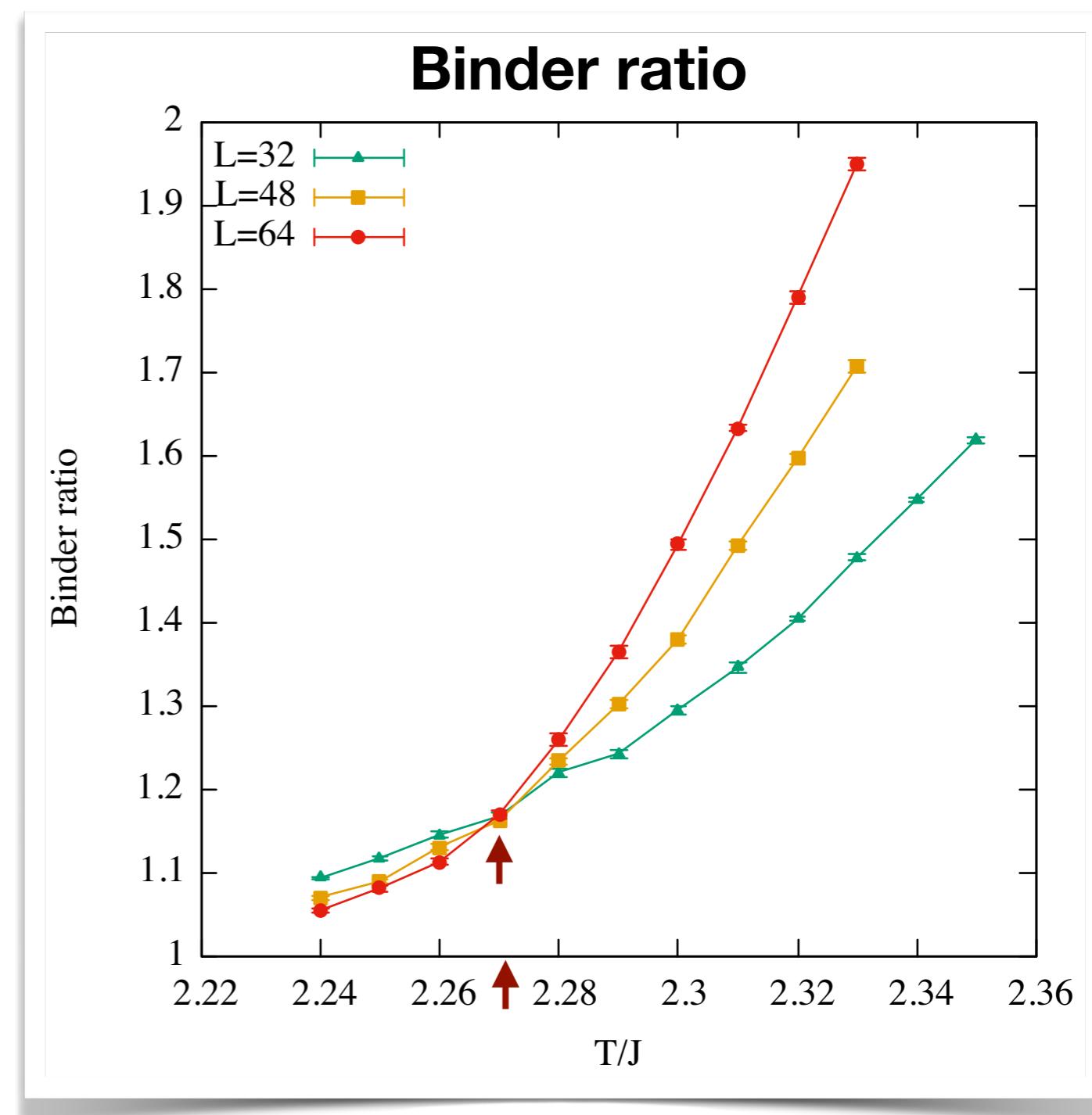
$$b = 3 \quad (T \rightarrow \infty)$$

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

The scaling dimension of b is exactly zero.



At T_c , the size dependence disappears in leading order!



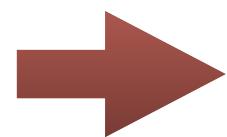
Finite size scaling: Binder ratio

Binder ratio

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

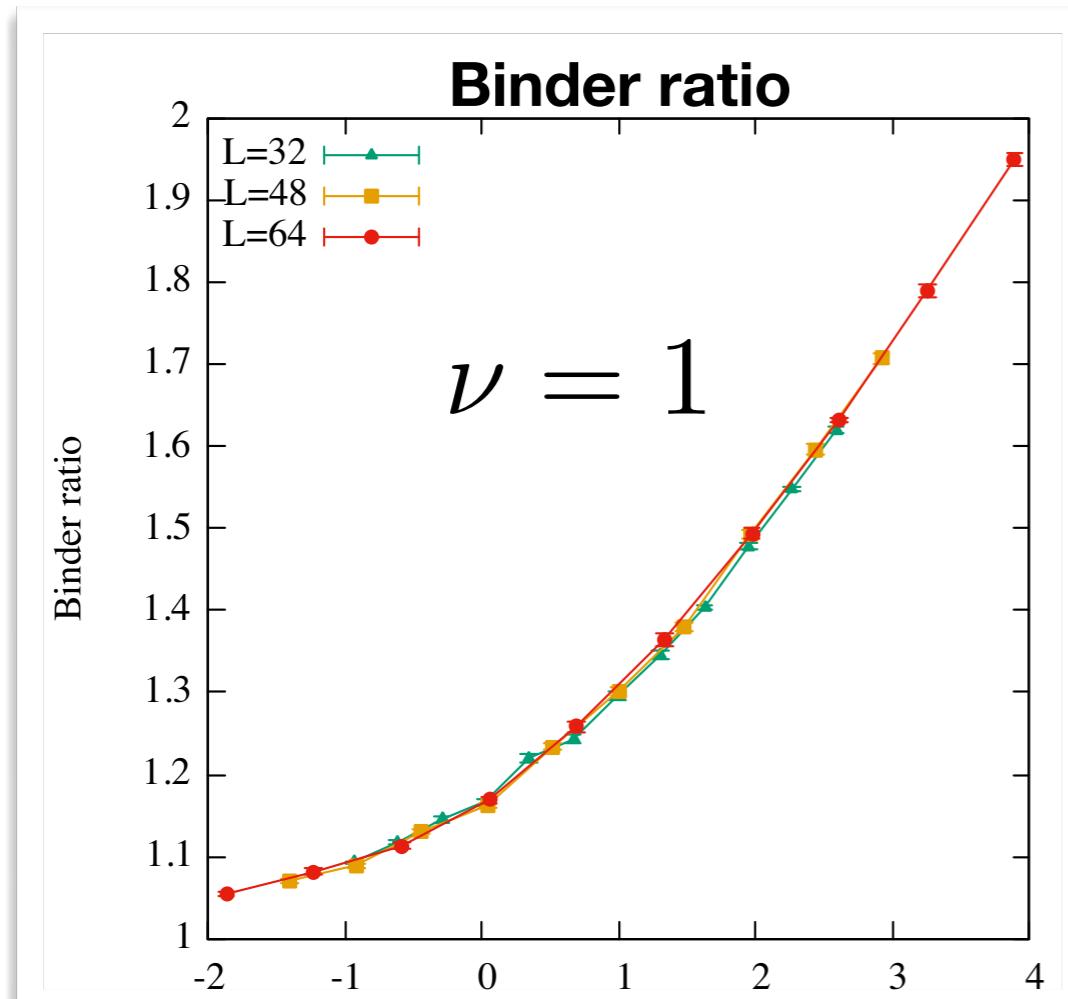
Finite size scaling around T_c

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

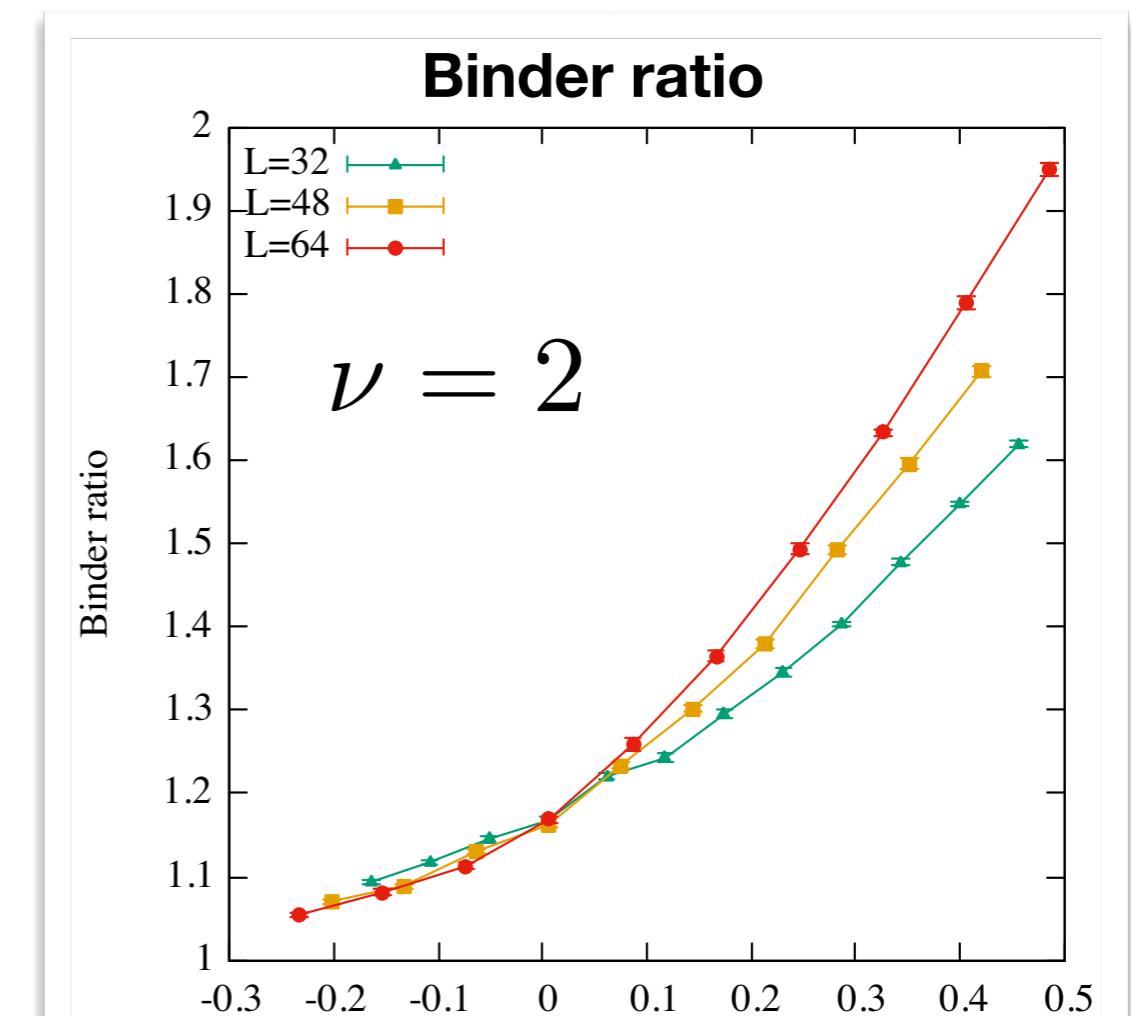


We can determine critical exponent!

$$\nu = 1$$



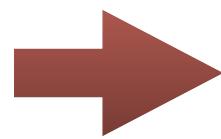
$$(T - T_c)L^{1/\nu}$$



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

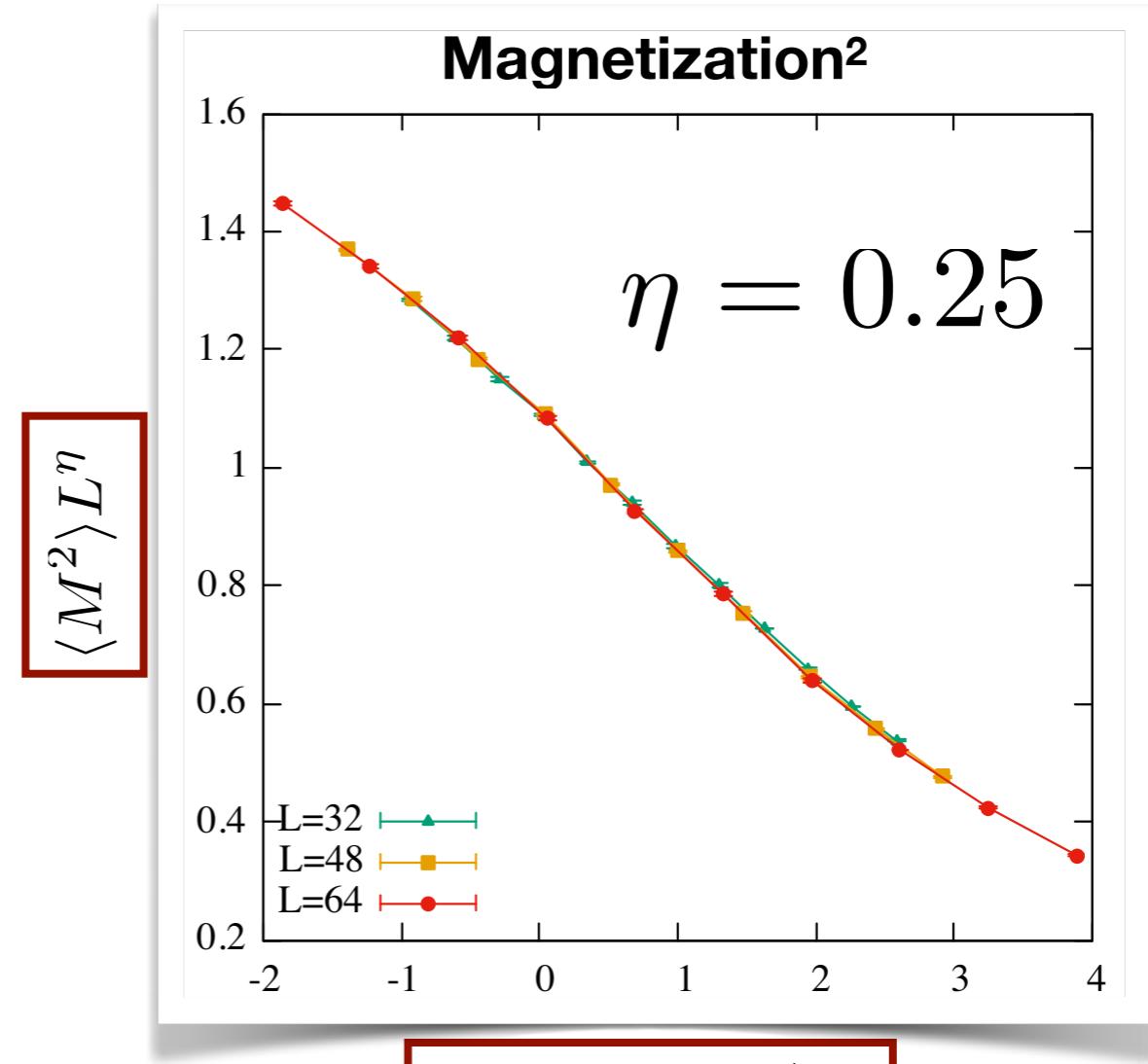
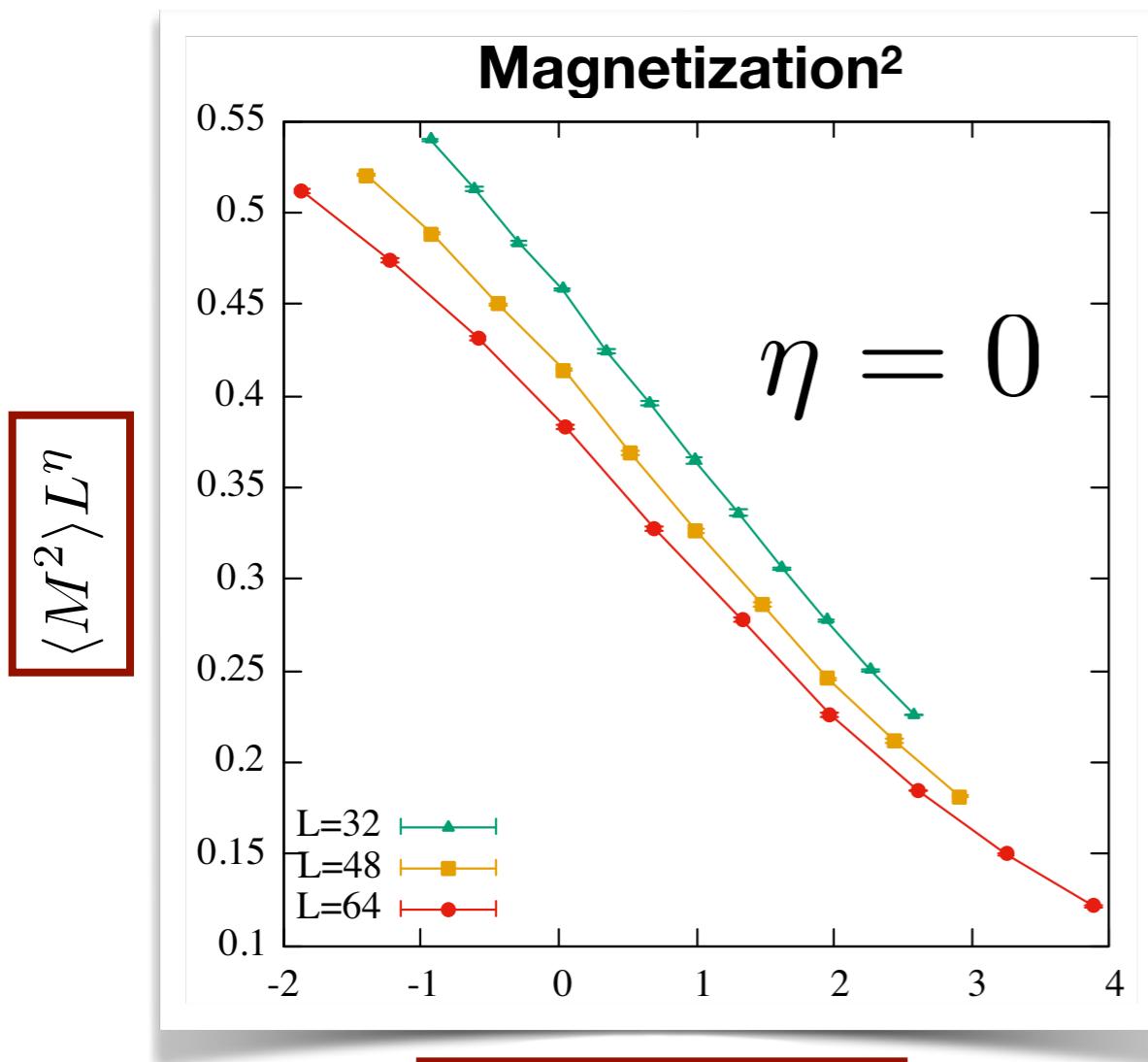
Finite size scaling: Magnetization

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



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

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



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

Exercise2: finite size scaling

Try the finite size scaling of, eg. 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*, and *numba* modules (*numba* is used for speed up).
 - In case you do not want to install numba,
please use "wo_jit" versions ([It is very slow...](#)).

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.

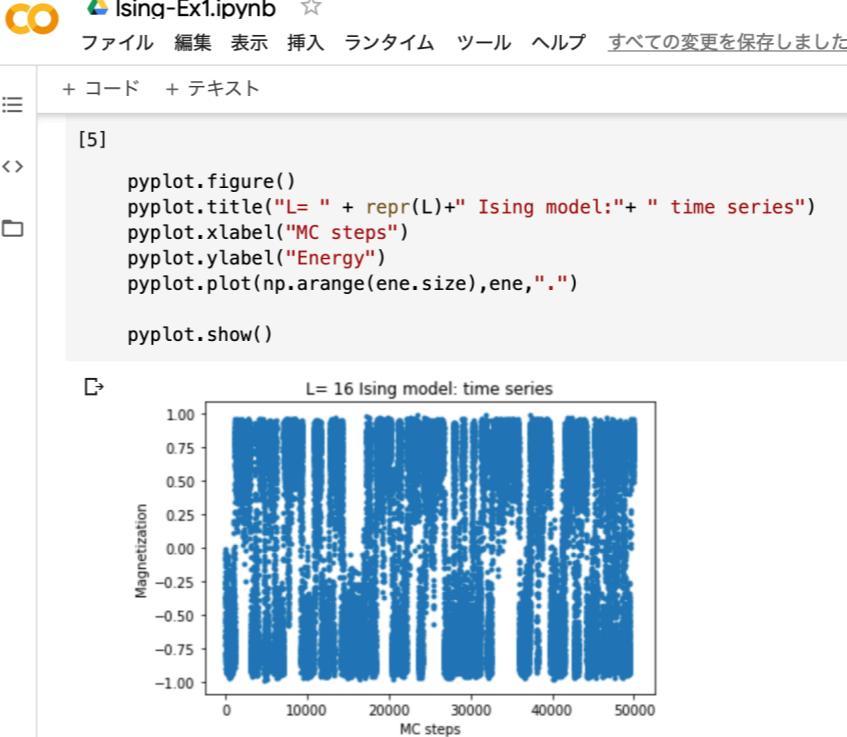
python3 Ising-Ex2.py --L_list 4 8 16 32

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.

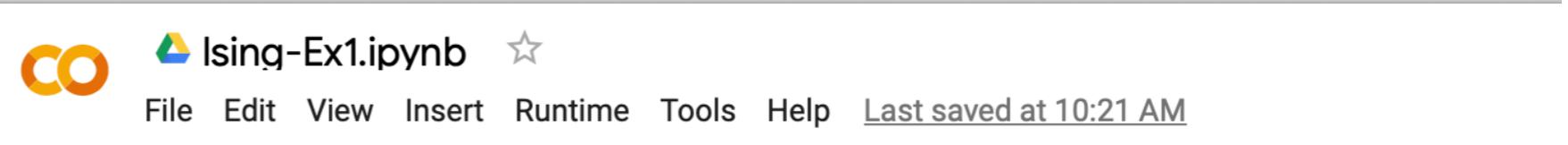


The screenshot shows the Google Colab interface. At the top, there's a toolbar with icons for file operations, a search bar, and a save button. Below the toolbar, the title "Ising-Ex1.ipynb" is visible. The main area has two code cells. The first cell, labeled [5], contains Python code for plotting a time series of magnetization. The second cell shows a plot titled "L= 16 Ising model: time series". The x-axis is labeled "MC steps" and ranges from 0 to 50,000. The y-axis is labeled "Magnetization" and ranges from -1.00 to 1.00. The plot displays a highly fluctuating blue line between -1.00 and 1.00, indicating the time series of magnetization for a 16x16 Ising model.

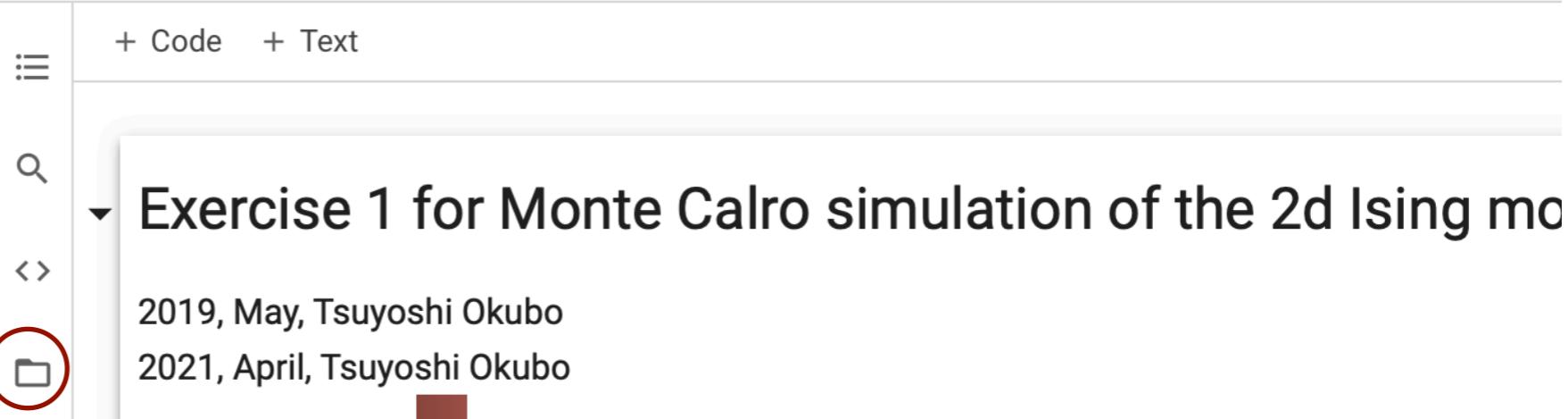
```
pyplot.figure()
pyplot.title("L= " + repr(L)+" Ising model:"+ " time series")
pyplot.xlabel("MC steps")
pyplot.ylabel("Energy")
pyplot.plot(np.arange(ene.size),ene,".")
pyplot.show()
```

How to use Google Colab

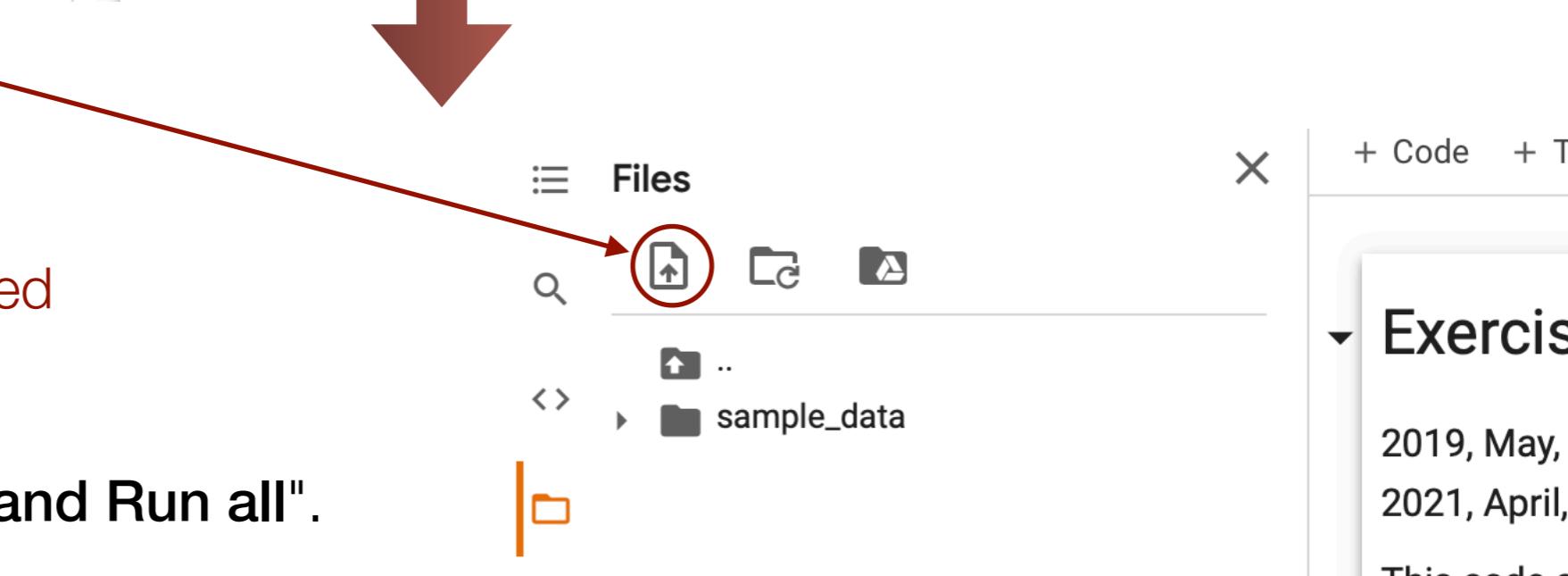
1. Open, e.g., *Ising-Ex1.ipynb*, in Google colab
 - Select "File/upload notebook" ("ファイル/ノートブックをアップロード") and upload *Ising-Ex1.ipynb*



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.
(「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.)
- "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.)
-

Contents

- Basics of MD simulation
 - Newtonian equation, purpose of MD simulation
 - Examples of numerical integrations
- NVE ensemble: standard MD simulation
 - Symplectic integral
- Control temperature and pressures (**quick review**)
 - Velocity scaling and Nosé-Hoover method
 - Andersen method for pressure

Basics of MD simulation

Newtonian equation, purpose of MD simulation

Target: Newtonian mechanics

N-particle system:

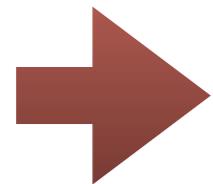
$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i(\{\mathbf{r}_i\}) \quad i = 1, 2, \dots, N$$

e.g.

$$\mathbf{F}_i(\{\mathbf{r}_i\}) \equiv \sum_{j \neq i} F(|\mathbf{r}_i - \mathbf{r}_j|) \hat{\mathbf{r}}_{ij}$$

Unit vector

$$\hat{\mathbf{r}}_{ij} = \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|}$$



Molecular Dynamics (MD) simulation:

Solve the newtonian equation numerically.

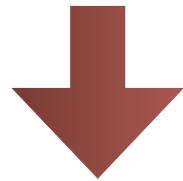
Standard flow of MD simulation

1. Determine the model.
 - Potential energies, constraints (e.g. polymers)
 - Periodic boundary, open boundary, ...
2. Prepare initial conditions. $\{\mathbf{r}_i(t = 0), \mathbf{v}_i(t = 0)\}$
3. Calculate forces acting to all particles. $\{\mathbf{F}_i(\{\mathbf{r}_i(t)\})\}$
4. Change positions and velocities by a discrete method $\{\mathbf{r}_i(t + \Delta t), \mathbf{v}_i(t + \Delta t)\}$
5. Calculate physical quantities and control them if we need
 - Constant temperature, Constant pressure, ... $T(\{\mathbf{r}_i(t), \mathbf{v}_i(t)\}),$
 $P(\{\mathbf{r}_i(t), \mathbf{v}_i(t)\}),$
...
6. Analyze trajectories



Periodic boundary condition (will be skipped)

A particle interacts with all other particles in “image cells”.



Short-range interaction

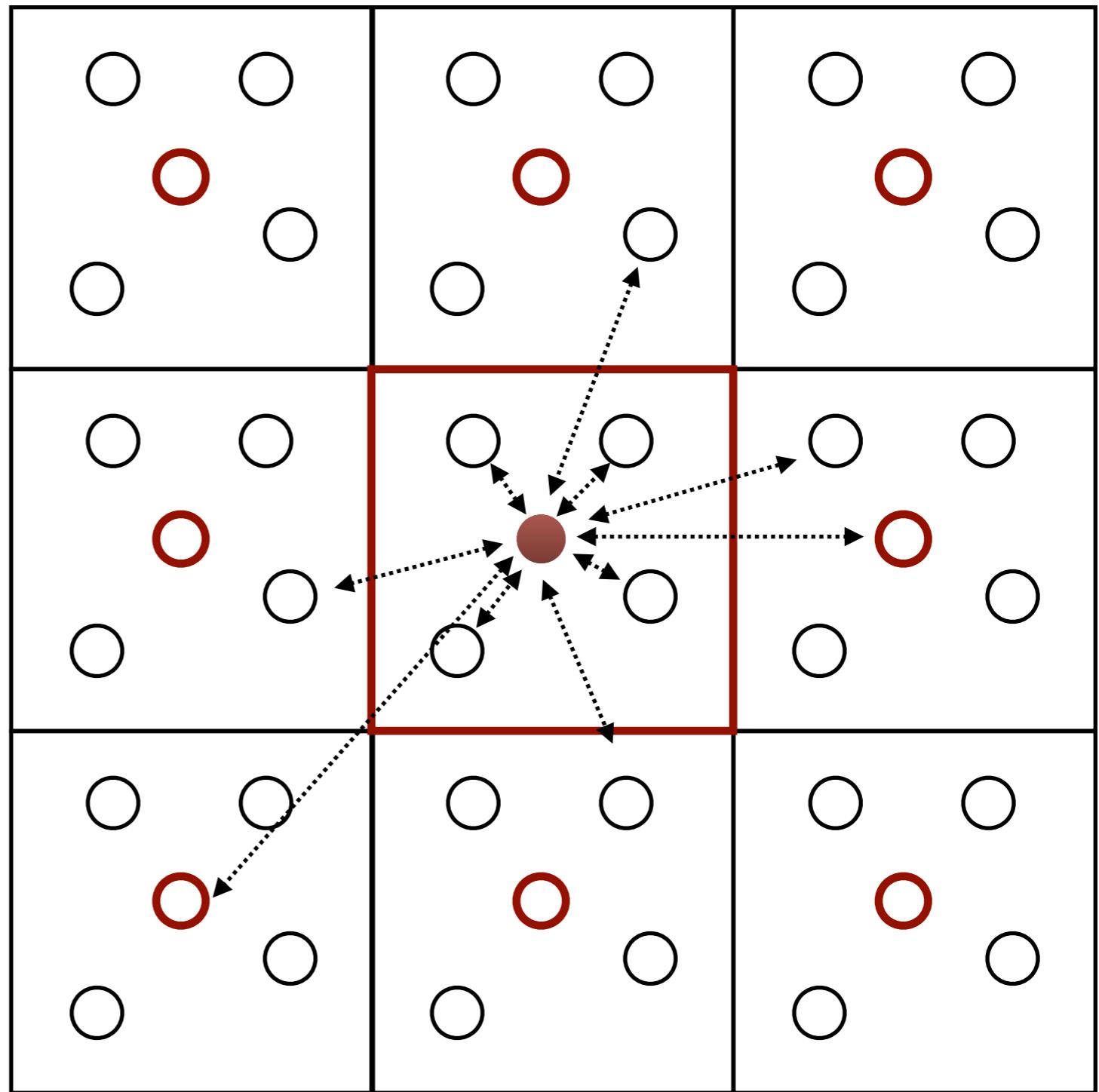
e.g. LJ potential

→ Introduce cut-off

Long-range interaction

e.g. Coulomb potential

-
- Ewald sum
 - Multipole expansion



Purpose of MD simulation: Equilibrium properties

By MD simulation, we can calculate equilibrium properties.

Usual Newtonian dynamics give us the NVE ensemble.

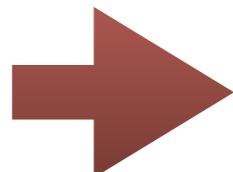
$$\langle \hat{O} \rangle_{NVE} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt \hat{O}(\Gamma(t))$$

By using temperature or pressure controls, we can also obtain other ensemble averages.

$$\langle \hat{O} \rangle_{NVT} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt \hat{O}(\underline{\Gamma_{NVT}(t)})$$
$$\langle \hat{O} \rangle_{NPT} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt \hat{O}(\underline{\Gamma_{NPT}(t)})$$

Modified dynamics!

Note: For large N limit, difference among ensembles is negligible.



We can use any ensembles for simulation.

Purpose of MD simulation: Equilibrium dynamics

By MD simulation, we can also calculate equilibrium dynamics

$$\langle \hat{A} \hat{B}(\Delta t) \rangle_{NVE} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt \hat{A}(\Gamma(t)) \hat{B}(\Gamma(t + \Delta t))$$

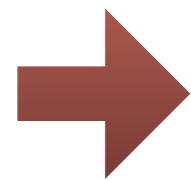
$$\langle \hat{A} \hat{B}(\Delta t) \rangle_{NPT} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt \hat{A}(\Gamma_{NPT}(t)) \hat{B}(\underline{\Gamma_{NPT}(t + \Delta t)})$$

$$\langle \hat{A} \hat{B}(\Delta t) \rangle_{NVT} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt \hat{A}(\Gamma_{NVT}(t)) \hat{B}(\underline{\Gamma_{NVT}(t + \Delta t)})$$

Modified dynamics!

Note:

In this case, as far as I know, **there is no proof** that the modified dynamics for different ensembles give us same results in large N limit.



Probably, it is better to use NVE ensemble, after proper initialization using NPT or NVT dynamics.

Purpose of MD simulation: Non-Equilibrium

We can also calculate non-equilibrium properties using MD.

- Applying external fields
- Observing relaxation from initial conditions

e.g. linear response coefficients

- We can calculation the coefficient from equilibrium simulation by using **Green-Kubo formula**:

$$\gamma = V\beta \int_0^\infty \langle J(0)J(t) \rangle dt$$

- It can be obtained by MD simulation **applying the external field**.
 - Usually the non-equilibrium calculation gives us smaller error.

Examples of numerical integrations

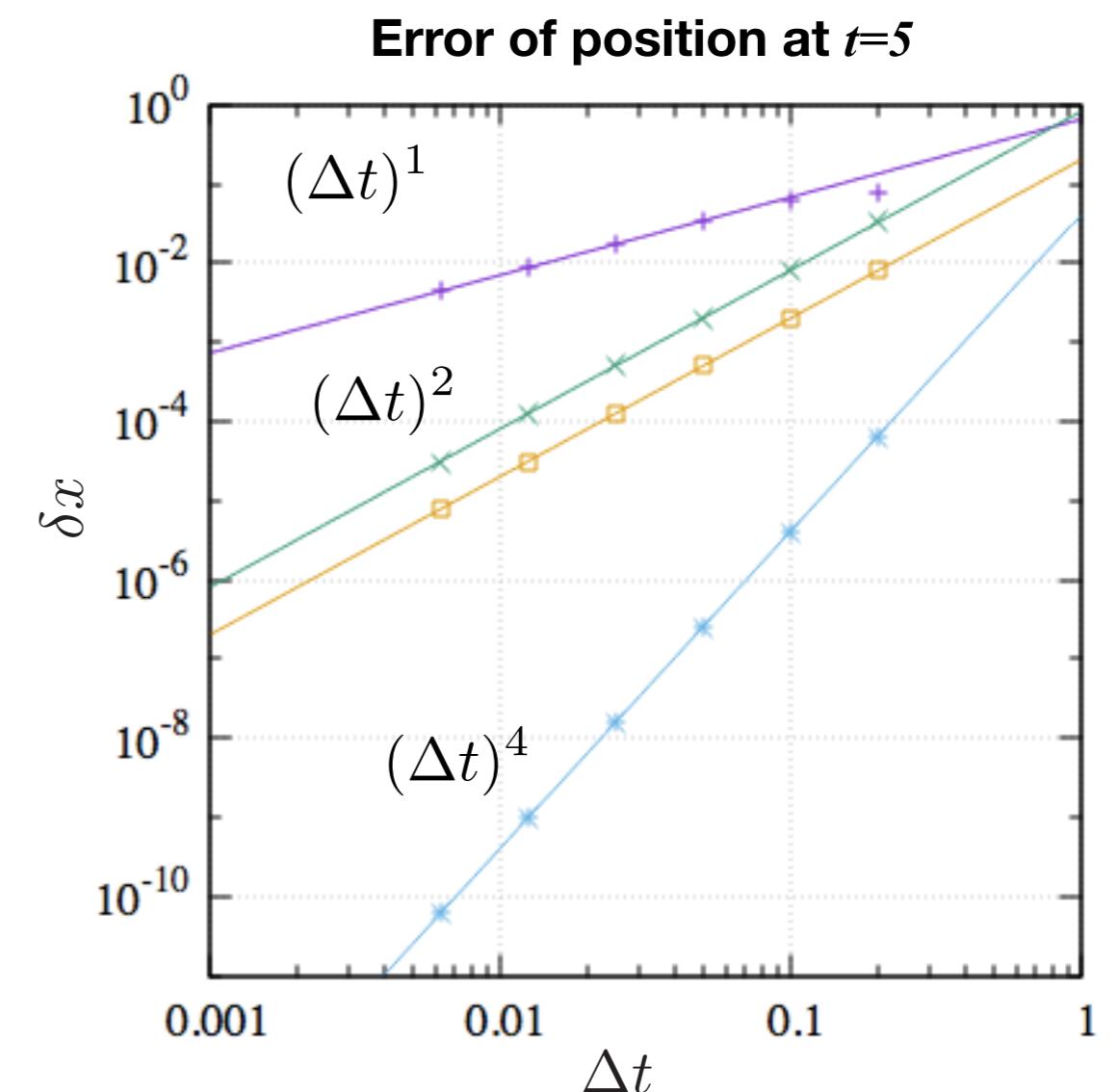
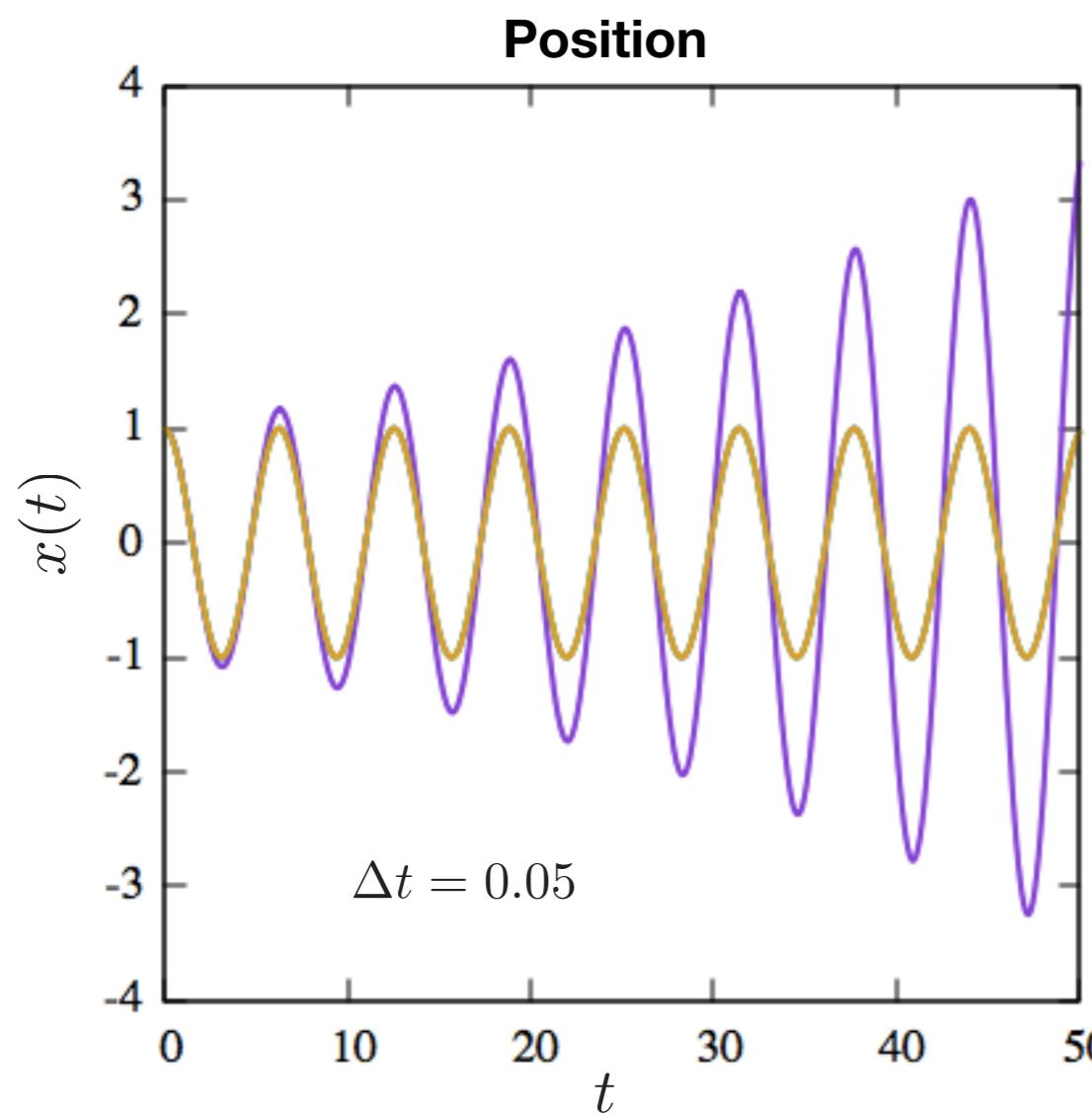
Numerical integration: Basics

Example: 1d harmonic oscillator

$$\mathcal{H}(x) = \frac{1}{2}x^2 + \frac{1}{2}v^2 \quad \rightarrow \quad \begin{aligned}\frac{dv}{dt} &= -x \\ \frac{dx}{dt} &= v\end{aligned}$$

Several explicit methods

Euler —————
Improved Euler —————
4th Runge-Kutta —————
Verlet —————



Numerical integration: accuracy and cost

Important points for molecular dynamics simulation

- Error
- Stability
- Number of force calculations

Main part of the cpu cost

e.g.

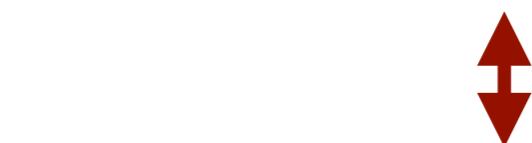
$$\mathbf{F}_i(\{\mathbf{r}_i\}) \equiv \sum_{j \neq i} F(|\mathbf{r}_i - \mathbf{r}_j|) \hat{\mathbf{r}}_{ij}$$

	order of error	#of force calculation	initial condition
Euler	Δt	1	$r(0), v(0)$
Improved Euler	$(\Delta t)^2$	2	$r(0), v(0)$
4th Runge-Kutta	$(\Delta t)^4$	4	$r(0), v(0)$
Verlet	$(\Delta t)^2$	1	$r(0), r(\Delta t)$ (velocity Verlet: $r(0), v(0)$)
Predictor-Corrector	$(\Delta t)^5$	2 (or 1)	$r(0), r'(0), r''(0), r'''(0), r''''(0)$

Numerical integration: instability (energy drift)

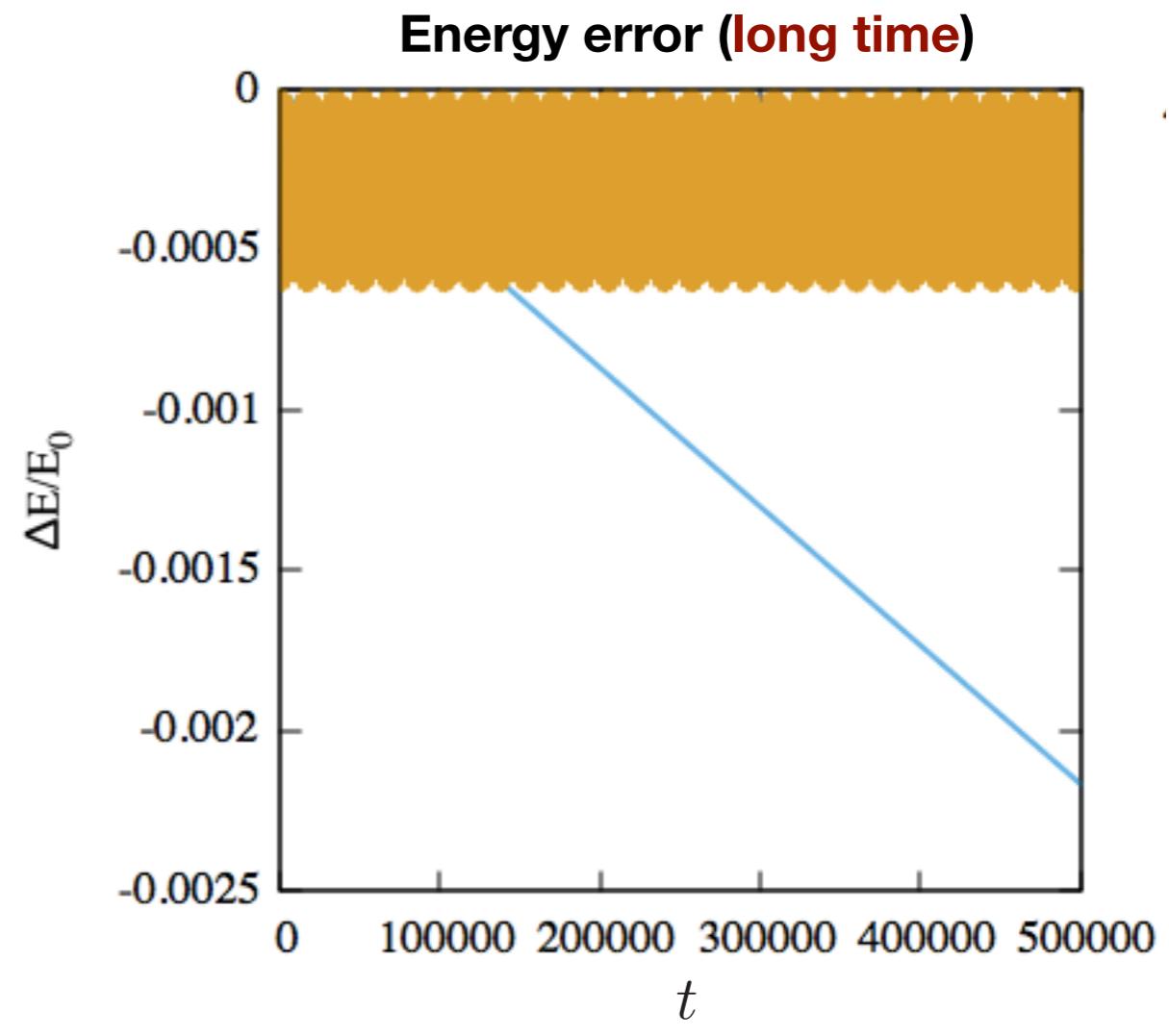
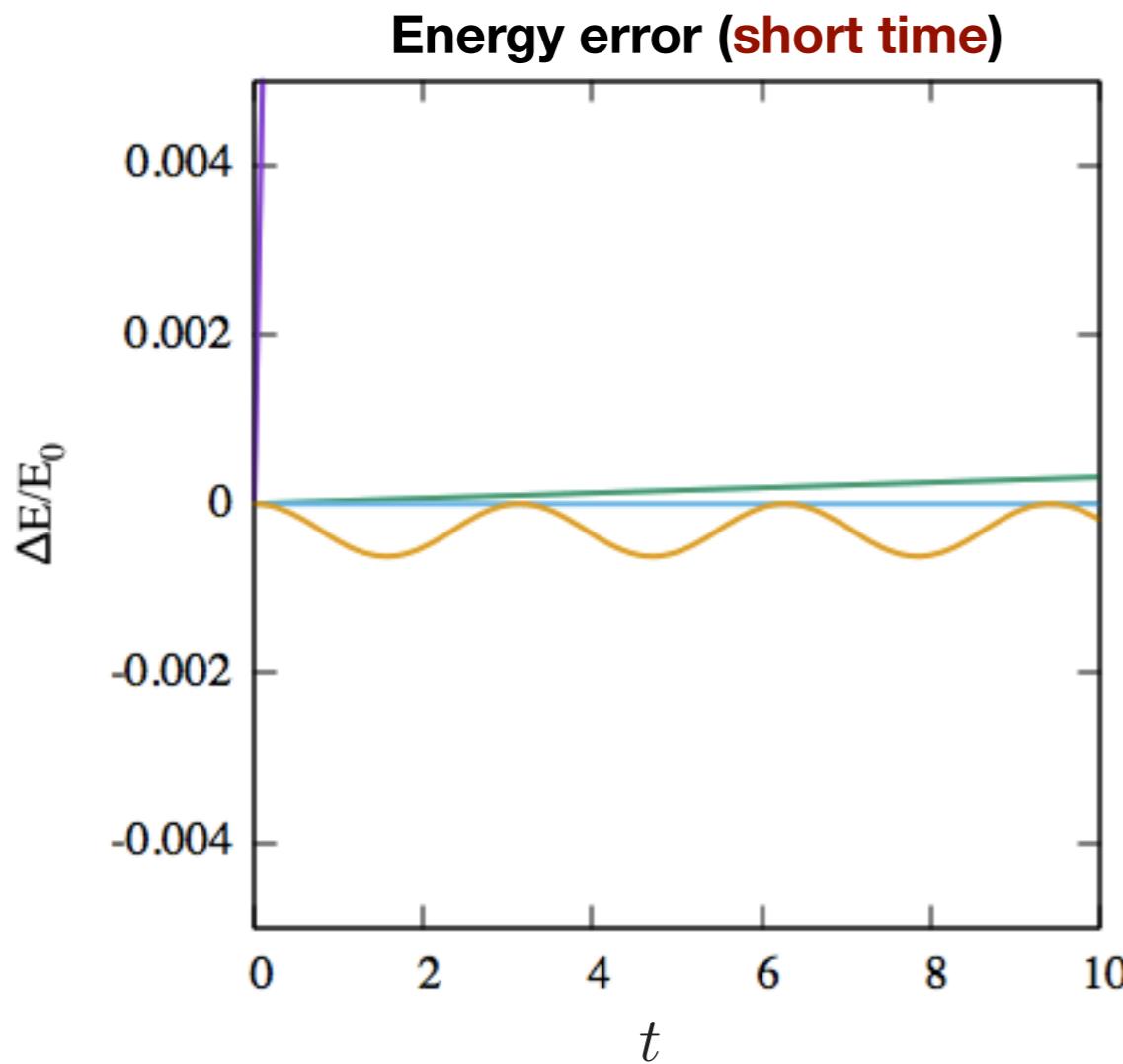
Example: 1d harmonic oscillator

Usual methods shows a drift of energy!
(Predictor-Corrector also shows large energy drift)



Verlet shows a very small energy drift.

Euler
Improved Euler
4th Runge-Kutta
Verlet



Better methods for molecular dynamics simulation

Verlet method:

- Error
 - $(\Delta t)^2 \rightarrow$ not bad
- Stability
 - It seems to be so stable!
- Number of force calculations
 - Only 1 force calculation for 1 step

Verlet method:

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &= 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \frac{(\Delta t)^2}{m_i} \mathbf{F}_i(\{\mathbf{r}_i(t)\}) \\ \mathbf{v}_i(t) &= \frac{\mathbf{r}_i(t + \Delta t) - \mathbf{r}_i(t - \Delta t)}{2\Delta t} \end{aligned}$$

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i(\{\mathbf{r}_i\})$$

Velocity Verlet method:

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{(\Delta t)^2}{2m_i} \mathbf{F}_i(\{\mathbf{r}_i(t)\}) \\ \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \Delta t \frac{\mathbf{F}_i(\{\mathbf{r}_i(t)\}) + \mathbf{F}_i(\{\mathbf{r}_i(t + \Delta t)\})}{2m_i} \end{aligned}$$

Leap-frog method:

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i \left(t + \frac{\Delta t}{2} \right) \Delta t \\ \mathbf{v}_i \left(t + \frac{\Delta t}{2} \right) &= \mathbf{v}_i \left(t - \frac{\Delta t}{2} \right) + \Delta t \frac{\mathbf{F}_i(\{\mathbf{r}_i(t)\})}{m_i} \end{aligned}$$

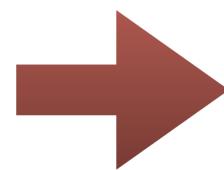
These methods are basically equivalent.
They are based on the second-order
symplectic integration scheme.

NVE ensemble: symplectic integrator

Hamilton mechanics

Hamilton mechanics

$$\mathcal{H}(\{q_i\}, \{p_i\})$$



$$\frac{dq_i}{dt} = \frac{\partial \mathcal{H}}{\partial p_i}$$
$$\frac{dp_i}{dt} = -\frac{\partial \mathcal{H}}{\partial q_i}$$

Any quantities: $A(t) = A[\{q_i(t)\}, \{p_i(t)\}]$

$$\frac{dA}{dt} = \{A, \mathcal{H}\}$$

Poisson bracket:

$$\{u, v\} = \sum_i \left(\frac{\partial u}{\partial q_i} \frac{\partial v}{\partial p_i} - \frac{\partial u}{\partial p_i} \frac{\partial v}{\partial q_i} \right)$$

Liouville operator: $i\mathcal{L} = \{ \quad , \mathcal{H} \}$

$$\frac{dA}{dt} = i\mathcal{L}A$$



$$A(t) = e^{it\mathcal{L}}A(0)$$

Unitary operator

Liouville's theorem

Distribution function: $\rho(\{q_i\}, \{p_i\}; t)$

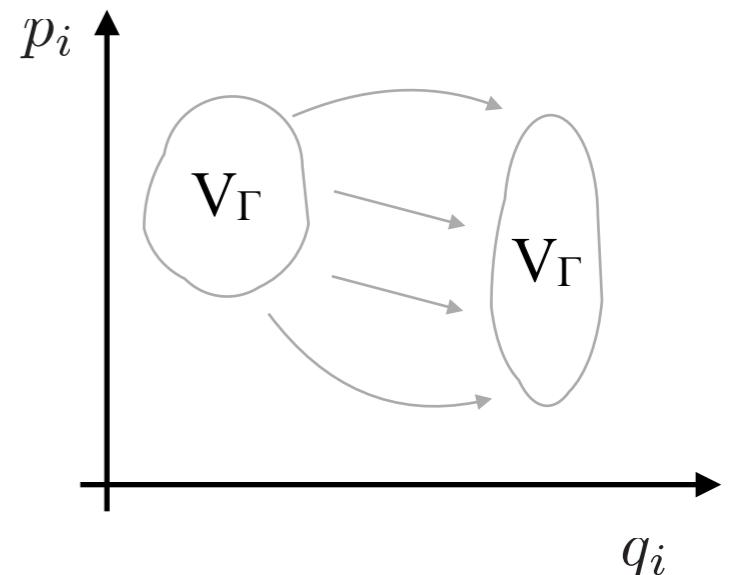
Liouville equation

$$\frac{\partial \rho}{\partial t} = \{\mathcal{H}, \rho\} = -i\mathcal{L}\rho$$

→
$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial t} + \sum_i \left(\frac{\partial \rho}{\partial q_i} \frac{\partial \mathcal{H}}{\partial p_i} - \frac{\partial \rho}{\partial p_i} \frac{\partial \mathcal{H}}{\partial q_i} \right) = \frac{\partial \rho}{\partial t} + i\mathcal{L}\rho = 0$$

Liouville's theorem:

Along Hamiltonian mechanics,
the volume in phase space is conserved.



Canonical transformation (正準変換)

Hamilton dynamics: $\frac{dq_i}{dt} = \frac{\partial \mathcal{H}}{\partial p_i}, \quad \frac{dp_i}{dt} = -\frac{\partial \mathcal{H}}{\partial q_i}$

For $2n$ -dim. vector representation:

$$\Gamma = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}$$

→ Time evolution of Γ :

$$\frac{d\Gamma}{dt} = J \frac{\partial \mathcal{H}}{\partial \Gamma}$$

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad : 2n \times 2n \text{ matrix} \\ (n=3N \text{ in 3d system})$$

$$\mathbf{1} \quad : n \times n \text{ identity matrix}$$

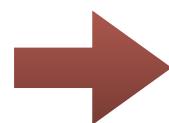
Symplectic condition

Canonical transformation: $\Gamma \rightarrow \Gamma' = (\mathbf{q}'(\mathbf{q}, \mathbf{p}), \mathbf{p}'(\mathbf{q}, \mathbf{p}))$

Jacobian matrix S

$$S_{ij} = \frac{\partial \Gamma'_i}{\partial \Gamma_j}, \quad S = \begin{pmatrix} \frac{\partial \mathbf{q}'}{\partial \mathbf{q}} & \frac{\partial \mathbf{q}'}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{p}'}{\partial \mathbf{q}} & \frac{\partial \mathbf{p}'}{\partial \mathbf{p}} \end{pmatrix}$$

Time evolution of Γ' :



$$\begin{aligned} \frac{d\Gamma'}{dt} &= S \frac{d\Gamma}{dt} = SJ \frac{\partial \mathcal{H}}{\partial \Gamma} \\ &= SJS^T \frac{\partial \mathcal{H}}{\partial \Gamma'} = J \frac{\partial \mathcal{H}}{\partial \Gamma'} \end{aligned}$$

From the relation $\frac{d\Gamma'_i}{dt} = \sum_j \frac{\partial \Gamma'_i}{\partial \Gamma_j} \frac{d\Gamma_j}{dt}$

$$\frac{\partial}{\partial \Gamma_i} = \sum_j \frac{\partial \Gamma'_j}{\partial \Gamma_i} \frac{\partial}{\partial \Gamma'_j} = \sum_j (S^T)_{ij} \frac{\partial}{\partial \Gamma'_j}$$

Definition of the canonical transformation

Canonical transformation satisfies the symplectic condition:

$$SJS^T = J$$

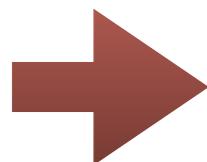
(This condition is actually the necessary and sufficient condition for canonical transformation.)

Infinitesimal time evolution

$$\Gamma(\mathbf{q}(t), \mathbf{p}(t)) \rightarrow \Gamma'(\mathbf{q}(t + \Delta t), \mathbf{p}(t + \Delta t))$$

$$\begin{aligned} q' &= \mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \Delta t \frac{\partial \mathcal{H}}{\partial \mathbf{p}(t)} \\ p' &= \mathbf{p}(t + \Delta t) = \mathbf{p}(t) - \Delta t \frac{\partial \mathcal{H}}{\partial \mathbf{q}(t)} \end{aligned}$$

This is a canonical transformation,
when Δt is infinitesimal.



Exact Hamiltonian dynamics satisfies the symplectic condition.
(The symplectic condition contains Liouville's theorem.)



For finite Δt (Euler method), it breaks the symplectic condition,
and the volume of phase space is **not** conserved.
(Main reason for the energy drift.)

If we can construct discrete approximations satisfying
the symplectic condition, we obtain more stable methods!

Symplectic integrator

Symplectic integrator:

Discrete approximation of Hamilton dynamics
satisfying the symplectic condition.

$$e^{it\mathcal{L}} \simeq \dots$$

→ Because the volume of phase space is conserved,
the energy does not drift along this dynamics.
(if there is no numerical error.)

If the Hamiltonian can be decomposed, for example

$$\mathcal{H} = K(\{p_i\}) + V(\{q_i\})$$

$$i\mathcal{L} = i\mathcal{L}_K + i\mathcal{L}_V$$

→ There is a systematic derivation of
symplectic integrators.

Decomposition of exponential operator

Symplectic integrator:

$$e^{it\mathcal{L}} = \prod_{k=1}^n [e^{ia_k t \mathcal{L}_K} e^{ib_k t \mathcal{L}_V}] + O(t^{n+1})$$

$$\sum_{k=1}^n a_k = \sum_{k=1}^n b_k = 1$$

$$\mathcal{H} = K(\{p_i\}) + V(\{q_i\})$$

$$i\mathcal{L} = i\mathcal{L}_K + i\mathcal{L}_V$$

Note: $e^{ia_k t \mathcal{L}_K}, e^{ia_k t \mathcal{L}_V}$ satisfy the symplectic condition

$n=1 \quad e^{it\mathcal{L}} \simeq e^{it\mathcal{L}_K} e^{it\mathcal{L}_V}$

$$q(t + \Delta t) = q(t) + p(t)\delta t$$

Euler like equation
(but this is more stable!)

$n=2 \quad e^{it\mathcal{L}} \simeq e^{i\frac{t}{2}\mathcal{L}_K} e^{it\mathcal{L}_V} e^{i\frac{t}{2}\mathcal{L}_K}$

$$p(t + \frac{\Delta t}{2}) = p(t) + F(q(t)) \frac{\Delta t}{2}$$

Exactly equal to
(Velocity) Verlet method

$$q(t + \Delta t) = q(t) + p(t + \frac{\Delta t}{2})\Delta t$$

$$p(t + \Delta t) = p(t + \frac{\Delta t}{2}) + F(q(t + \Delta)) \frac{\Delta t}{2}$$

Control temperature (quick review)

Temperature control: velocity scaling

The most simplest method for temperature setting: **Velocity Scaling**

(L. V. Woodcock, Chem. Phys. Lett. **10**, 257 (1971).)

Total kinetic energy: $K = \sum_i \frac{p_i^2}{2m_i}$

Under the canonical (NVT) ensemble

$$\langle K \rangle = \frac{3}{2} N k_B T \quad (\text{Equipartition of energy in 3d})$$



Define **effective temperature** of a snapshot:

$$T_{\text{eff}} \equiv \frac{2K}{3Nk_B}$$

Rescale velocities every time step as

$$p'_i = p_i \sqrt{\frac{T}{T_{\text{eff}}}} \quad \rightarrow \quad K' = \frac{3}{2} N k_B T$$

Results of the velocity scaling

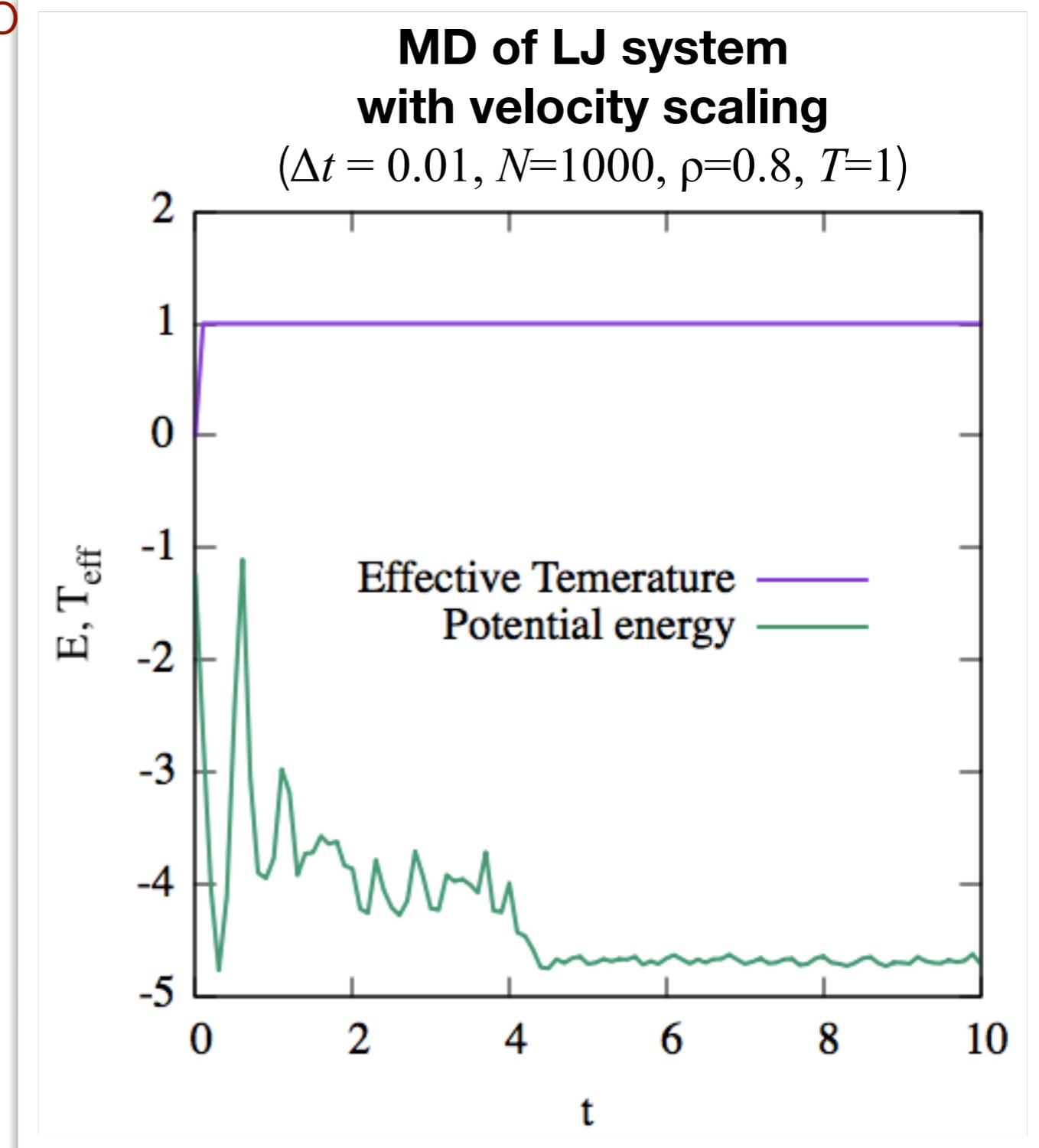
Total kinetic energy is artificially fixed to

$$K = \frac{3}{2} N k_B T$$

Under velocity scaling dynamics,
the trajectories do not
necessarily obey
the canonical ensemble.

However,

- We can use it for an initialization for NVE ensemble.
- Position fluctuation could be effectively similar to that of NVT ensemble.



Temperature control: Langevin dynamics

Langevin dynamics

$$\frac{d\mathbf{p}_i}{dt} = \underline{F_i(\{q_i\}) - \gamma \mathbf{p}_i} + \underline{\mathbf{R}_i}$$

Dissipation

Random force

(Gaussian white noise)

$$\langle \mathbf{R}_i(t) \rangle = \mathbf{0}$$

$$\langle \mathbf{R}_i(0) \mathbf{R}_j(t) \rangle = 2D_i \delta_{ij} \delta(t)$$



Long-time average of Langevin dynamics becomes the canonical ensemble with temperature T , if random forces satisfy the relation

$$D_i = \frac{k_B T}{m_i \gamma}$$

Einstein relation

Fluctuation-dissipation theorem

Temperature control: Nosé thermostat

Nose thermostat

S. Nosé, Mol. Phys., **52**, 255 (1984). S. Nosé, J. Chem. Phys., **81**, 511 (1984).

Extended Hamiltonian

System with a “heat bath”

$$\mathcal{H}_N = \sum_i \frac{(\mathbf{p}'_i)^2}{2m_i s^2} + V(\{\mathbf{q}_i\}) + \frac{P_s^2}{2Q} + gk_B T \ln s$$

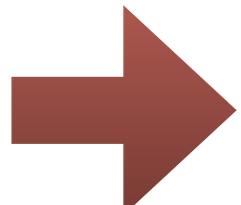
Original Hamiltonian
with scaled momentum

$$\mathcal{H}\left(\left\{\frac{\mathbf{p}'_i}{s}\right\}, \{\mathbf{q}_i\}\right)$$

Heat-bath
s: scale factor for time

$$t' = st$$
$$\mathbf{p}'_i = s\mathbf{p}_i$$

Canonical equation
(along \mathbf{t}')



$$\frac{d\mathbf{p}'_i}{dt'} = -\frac{\partial V}{\partial \mathbf{q}_i} = \mathbf{F}_i(\{\mathbf{q}_i\})$$

$$\frac{d\mathbf{q}_i}{dt'} = \frac{\mathbf{p}'_i}{m_i s^2}$$

$$\frac{dP_s}{dt'} = \frac{1}{s} \left(\sum_i \frac{(\mathbf{p}'_i)^2}{m_i s^2} - gk_B T \right)$$

$$\frac{ds}{dt'} = \frac{P_s}{Q}$$

Temperature control: Nosé-Hoover method

Nosé-Hoover dynamics

Real-time dynamics with $\zeta = \frac{ds}{dt'}$ (W. G. Hoover, Phys. Rev. A, **31**, 1695 (1985).)

$$\frac{d\mathbf{q}_i}{dt} = \frac{\mathbf{p}_i}{m_i}$$

$$\mathbf{p}_i = \frac{\mathbf{p}'_i}{s} \quad t = \frac{t'}{s}$$

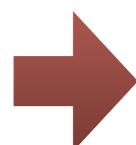
$$\frac{d\mathbf{p}_i}{dt} = \mathbf{F}_i(\{\mathbf{q}_i\}) - \zeta \underline{\mathbf{p}_i}$$

$$\frac{d\zeta}{dt} = \frac{gk_B}{Q} \left[\frac{1}{gk_B} \sum_i \frac{\mathbf{p}_i^2}{2m_i} - T \right] = \frac{1}{\tau^2} [T_{\text{eff}} - T]$$

$$g = 3N \text{ (# of DOF)}$$

$$\tau^2 = \frac{Q}{gk_B} \text{ (characteristic time scale)}$$

New degree of freedom represents viscosity: ζ



It changes the sign depending on the difference between the effective temperature and the aimed temperature.

(It also accelerates the velocity if $T_{\text{eff}} < T$)

* This dynamics is not symplectic.

Symplectic version: Nosé-Poincare method

S. D. Bond, et.al. J. Comp. Phys. **151**, 114 (1999)

Nosé-Hoover dynamics becomes NVT ensemble

Short proof:

(Based on Hisashi Okumura's review paper,
“分子動力学シミュレーションにおける温度・圧力制御”)

$$\mathcal{H}_N = \mathcal{H} \left(\left\{ \frac{\mathbf{p}'_i}{s} \right\}, \{\mathbf{q}_i\} \right) + \frac{P_s^2}{2Q} + gk_B T \ln s$$

MD on (q, p', \mathbf{t}') dynamics yields NVE ensemble of H_N

$$\begin{aligned} \rightarrow \lim_{\tau' \rightarrow \infty} \frac{1}{\tau'} \int_0^{\tau'} dt' O\left(\left\{ \frac{\mathbf{p}'_i}{s} \right\}, \{\mathbf{q}_i\}\right) &= \frac{\int d\mathbf{p}'_i d\mathbf{q}_i dP_s ds O\left(\left\{ \frac{\mathbf{p}'_i}{s} \right\}, \{\mathbf{q}_i\}\right) \delta(E - \mathcal{H}_N)}{\int d\mathbf{p}'_i d\mathbf{q}_i dP_s ds \delta(E - \mathcal{H}_N)} \\ s^{3N} \text{ comes from } \mathbf{p}'_i = s\mathbf{p}_i &\quad = \frac{\int d\mathbf{p}_i d\mathbf{q}_i dP_s ds s^{3N} O(\{\mathbf{p}_i\}, \{\mathbf{q}_i\}) \delta(E - \mathcal{H} - \frac{P^2}{2Q} - gk_B T \ln s)}{\int d\mathbf{p}_i d\mathbf{q}_i dP_s ds s^{3N} \delta(E - \mathcal{H} - \frac{P^2}{2Q} - gk_B T \ln s)} \end{aligned}$$

from

$$\int ds s^{3N} \delta(E - \mathcal{H} - \frac{P^2}{2Q} - gk_B T \ln s) = \frac{1}{gk_B T} e^{-\frac{3N+1}{gk_B T} (\mathcal{H} + \frac{P^2}{2Q} - E)}$$

$$= \frac{\int d\mathbf{p}_i d\mathbf{q}_i O(\{\mathbf{p}_i\}, \{\mathbf{q}_i\}) e^{-\frac{3N+1}{gk_B T} \mathcal{H}}}{\int d\mathbf{p}_i d\mathbf{q}_i e^{-\frac{3N+1}{gk_B T} \mathcal{H}}}$$

$$\delta(f(x)) = \frac{\delta(x - x_0)}{|f'(x_0)|}$$



Canonical ensemble if $g = 3N + 1$

$(f(x_0) = 0)$

Nosé-Hoover dynamics becomes NVT ensemble 2

Time average on $\textcolor{brown}{t}$:

$$\rightarrow \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt O(\{\mathbf{p}_i\}, \{\mathbf{q}_i\}) = \lim_{\tau' \rightarrow \infty} \frac{\tau'}{\tau} \frac{1}{\tau'} \int_0^{\tau'} dt' \frac{O(\{\mathbf{p}_i\}, \{\mathbf{q}_i\})}{s}$$

from
$$\boxed{\tau = \int_0^{\tau'} \frac{1}{s} dt'}$$

$$\rightarrow = \frac{\lim_{\tau' \rightarrow \infty} \frac{1}{\tau'} \int_0^{\tau'} dt' \frac{O(\{\mathbf{p}_i\}, \{\mathbf{q}_i\})}{s}}{\lim_{\tau' \rightarrow \infty} \frac{1}{\tau'} \int_0^{\tau'} dt' \frac{1}{s}}$$

$$= \frac{\int d\mathbf{p}_i d\mathbf{q}_i O(\{\mathbf{p}_i\}, \{\mathbf{q}_i\}) e^{-\frac{3N}{gk_B T} \mathcal{H}}}{\int d\mathbf{p}_i d\mathbf{q}_i e^{-\frac{3N}{gk_B T} \mathcal{H}}}$$

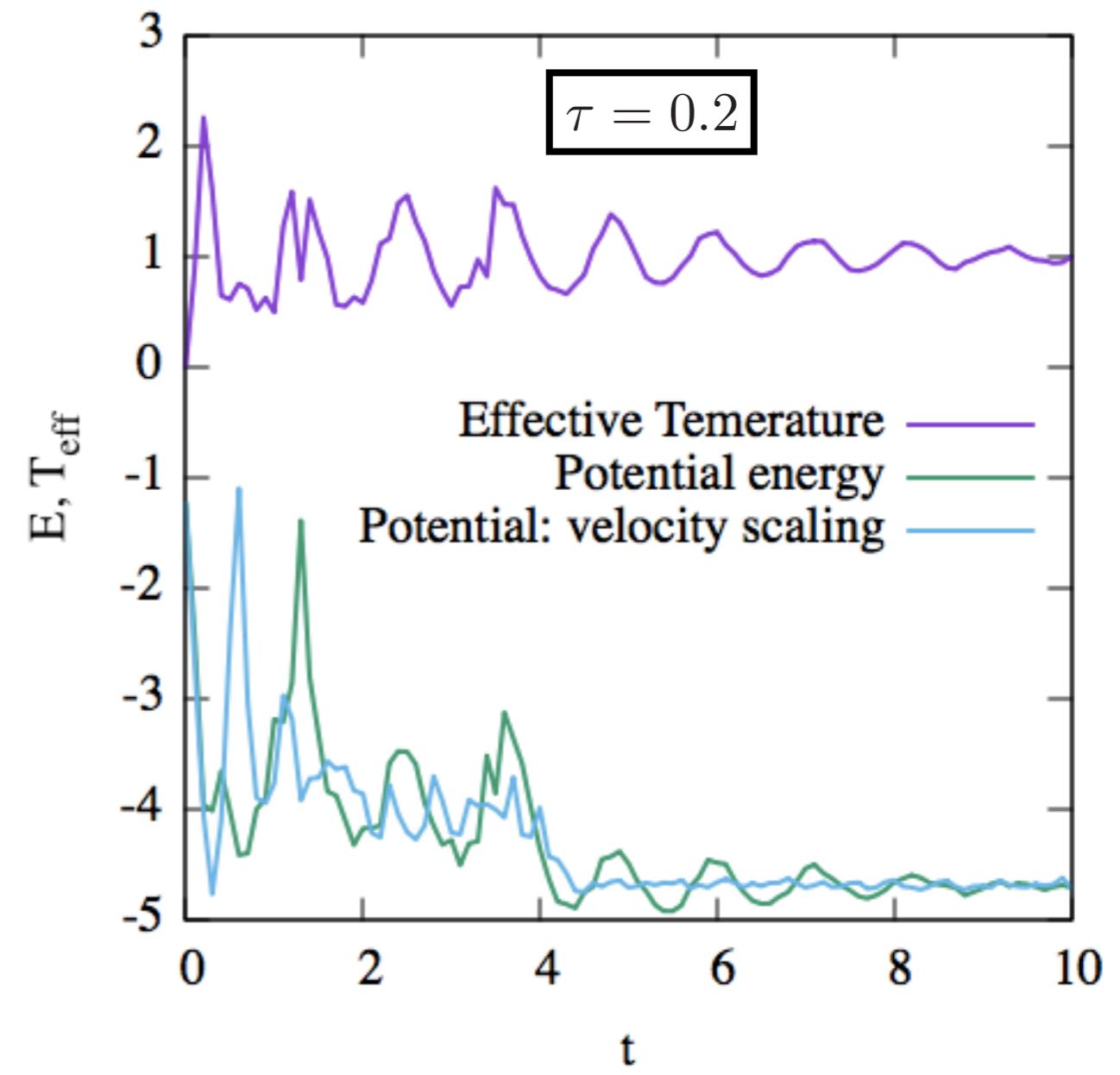
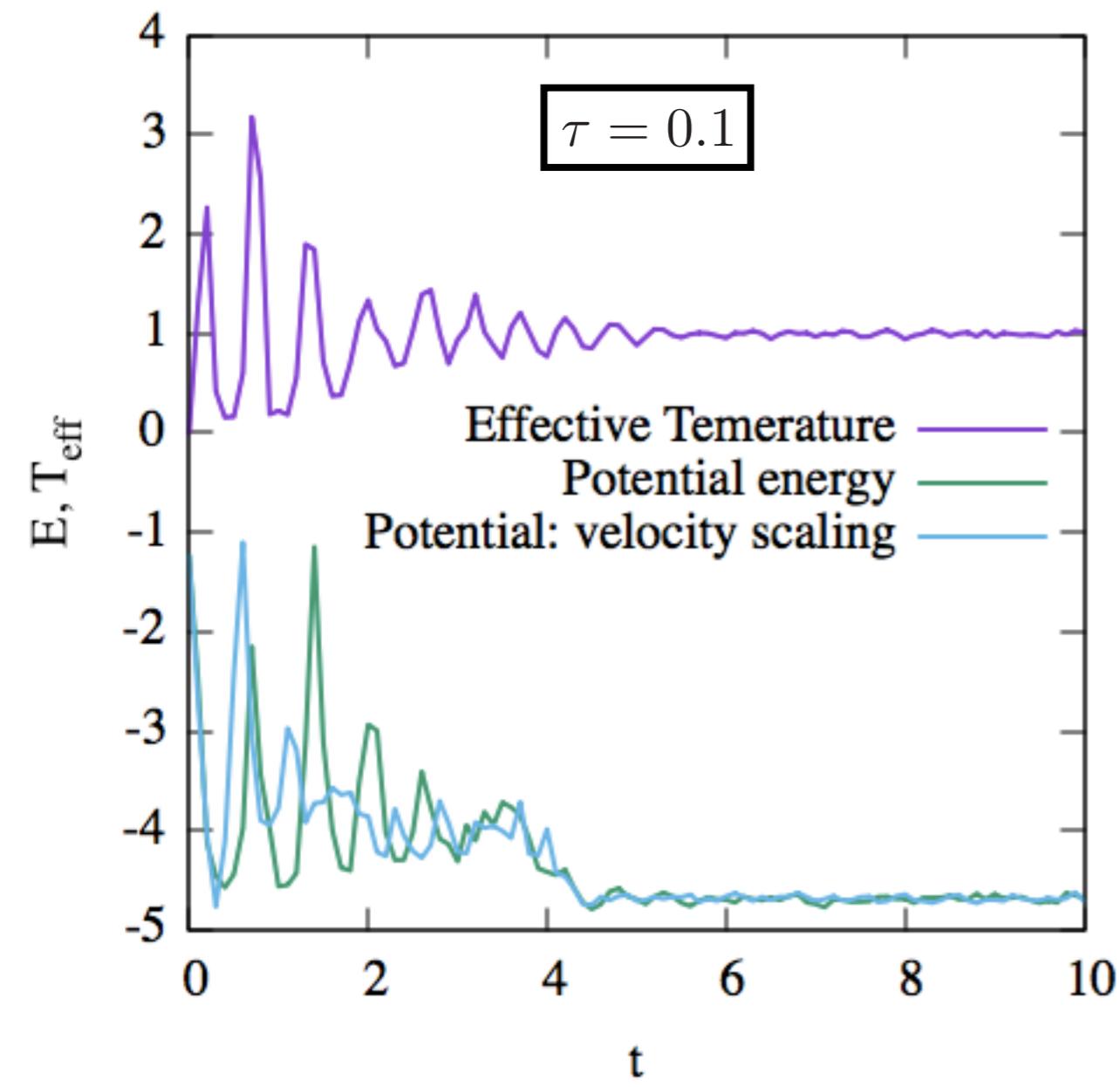
\rightarrow $\boxed{\text{Canonical ensemble if } g = 3N}$

Results of the Nose-Hoover dynamics

- Temperature behaves like damped oscillation.
 - Period is related to τ (or Q)
- Potential energy converges almost same value with that of velocity scaling.

MD of LJ system

($\Delta t = 0.01$, $N=1000$, $\rho=0.8$, $T=1$)



Control pressure (will be skipped)

Pressure control: Andersen method

H. C. Andersen, J. Chem. Phys. **72** (1980) 2384.

Extended Hamiltonian

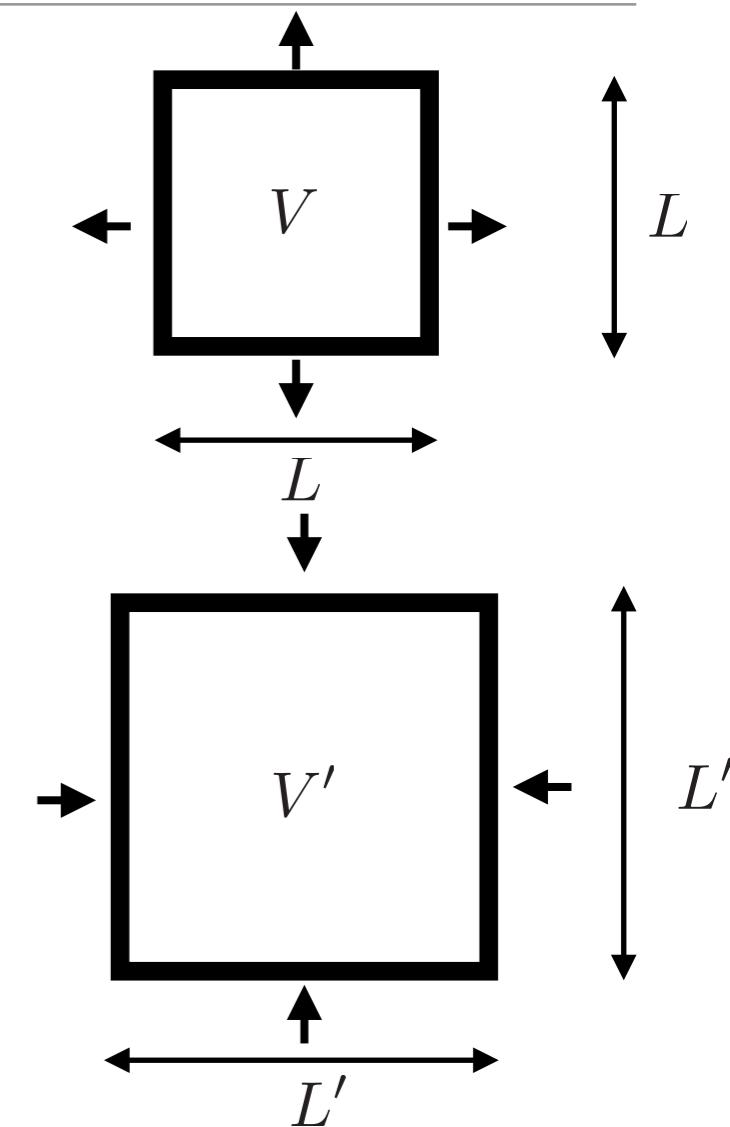
System with a “piston”

$$\mathcal{H}_A = \sum_i \frac{\tilde{p}_i^2}{2m_i V^{\frac{2}{3}}} + V_p(\{V^{\frac{1}{3}}\tilde{q}_i\}) + \frac{P_V^2}{2M} + PV$$

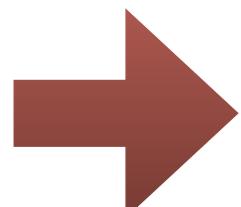
Original Hamiltonian with scaled coordinate and momentum

Piston

$$\begin{aligned}\tilde{q}_i &= V^{-\frac{1}{3}} q_i \\ \tilde{p}_i &= V^{\frac{1}{3}} p_i\end{aligned}$$



Canonical equation



$$\begin{aligned}\frac{d\tilde{q}_i}{dt} &= \frac{\tilde{p}_i}{m_i V^{\frac{2}{3}}} & \frac{dV}{dt} &= \frac{P_V}{M} \\ \frac{d\tilde{p}_i}{dt} &= V^{\frac{1}{3}} \mathbf{F}_i(\{V^{\frac{1}{3}}\tilde{q}_i\}) & \frac{dP_V}{dt} &= \frac{1}{3V} \sum_i \left[\frac{\tilde{p}_i^2}{m_i V^{\frac{2}{3}}} + \mathbf{F}_i \cdot (V^{\frac{1}{3}}\tilde{q}_i) \right] - P\end{aligned}$$

Pressure control: Andersen method

H. C. Andersen, J. Chem. Phys. **72** (1980) 2384.

In original coordinates

$$\begin{aligned}\frac{d\mathbf{q}_i}{dt} &= \frac{\mathbf{p}_i}{m_i} + \frac{\dot{V}}{3V}\mathbf{q}_i & \frac{dV}{dt} &= \frac{P_V}{M} \\ \frac{d\mathbf{p}_i}{dt} &= \mathbf{F}_i - \frac{\dot{V}}{3V}\mathbf{p}_i & \frac{dP_V}{dt} &= \frac{1}{3V} \sum_i \left[\frac{\mathbf{p}_i^2}{m_i} + \mathbf{F}_i \cdot \mathbf{q}_i \right] - P \\ &&&\text{---} \\ &&= P_{\text{eff}} - P & P_{\text{eff}} : \text{virial theorem}\end{aligned}$$

New degree of freedom controls the pressure like a piston.

→ P_v changes the sign depending on the difference between the effective pressure and the aimed pressure.

Andersen method gives us “approximate” NPH ensemble.

$H = \text{Enthalpy}$

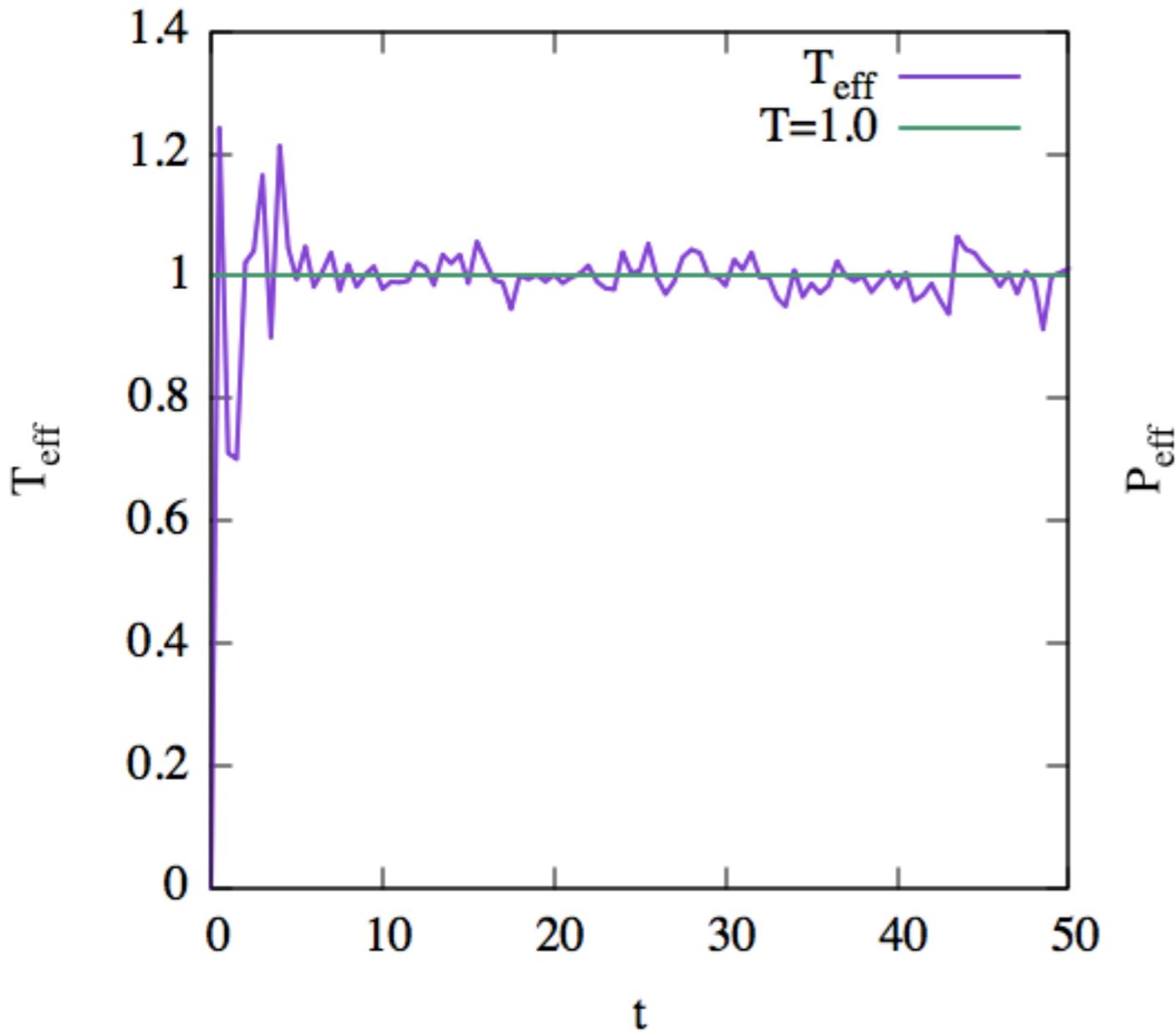
NPT ensemble

MD of LJ system

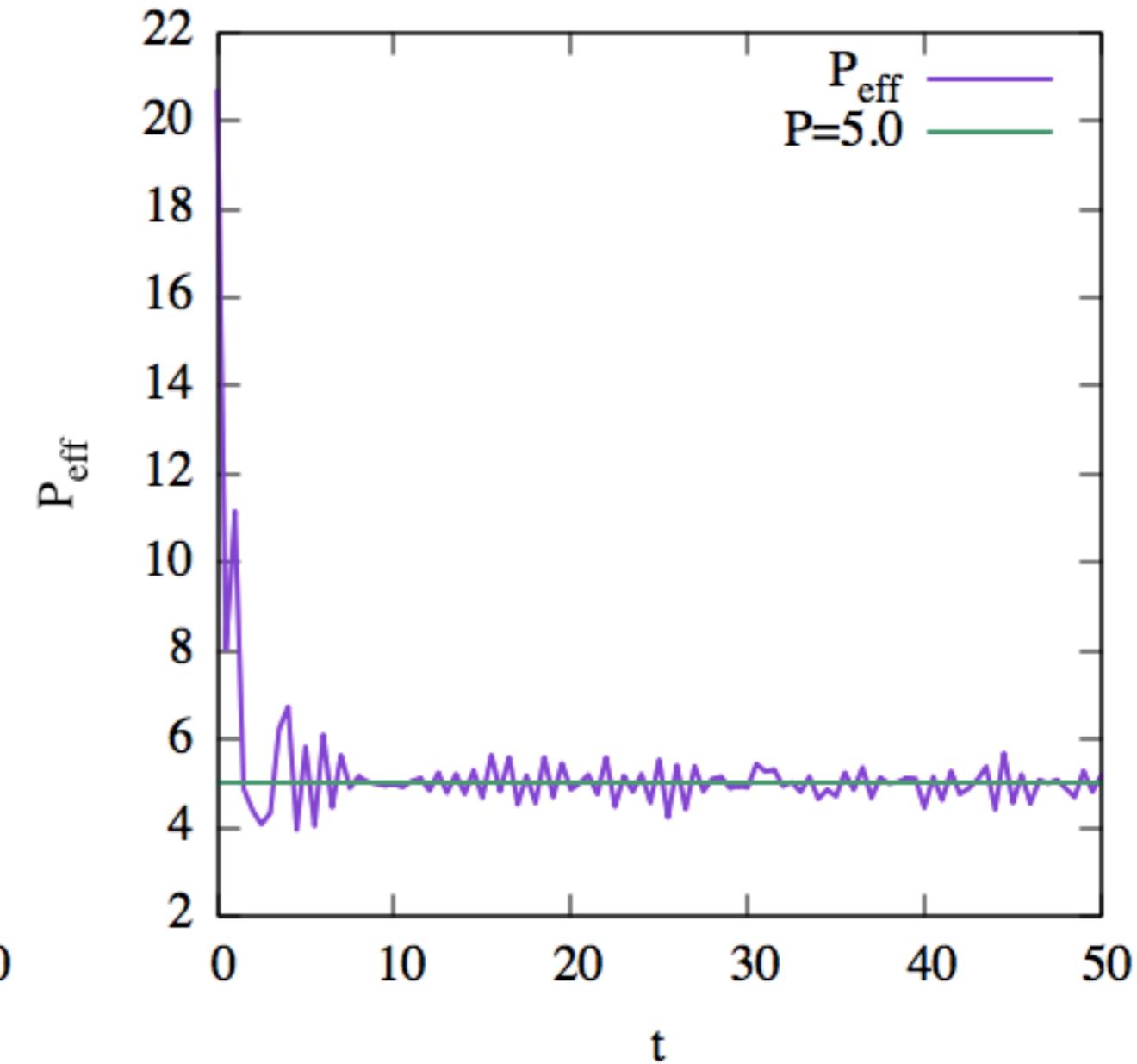
($\Delta t = 0.005, N=1000, T=1, P=5$)

By combining temperature and pressure controls,
we can obtain NPT ensemble. e.g. Nosé-Andersen method

Temperature



Pressure



Exercise: MD simulation of LJ particles(not a report)

Let's try MD simulation of LJ particles with NVE, NVT, and NPT ensembles.

- In NVE simulation (e.g. by Verlet method), see the conservation of the total energy.
- By using, velocity scaling or Nose-Hoover thermostat, try to control temperature.
- By combining temperature control and pressure control try to simulate NPT ensemble.

To perform these exercise, you may use,

- Your own code
- LAMMPS
 - <http://lammps.sandia.gov>
- MDACP (for NVE simulation.)
 - <http://mdacp.sourceforge.net/index.html>
- My sample codes for **jupyter notebook** (or **python3**).
 - To run the sample code you need
 - numpy, and numba (numba is used for speed up)
 - Jupyter notebook version contains auto-correlation analysis

Next (5/18)

Classical

Quantum

- 1st: Many-body problems in physics
- 2nd: Why many-body problem is hard to solve
- 3rd: Classical statistical models and numerical simulation**
- 4th: Classical Monte Carlo method and its applications**
- 5th: Molecular dynamics simulation and its applications**
- 6th: Extended ensemble method for Monte Carlo methods**
- 7th: Quantum lattice models and numerical simulation
- 8th: Quantum Monte Carlo methods
- 9th: Applications of quantum Monte Carlo methods
- 10th: Linear algebra of large and sparse matrices for quantum many-body problems
- 11th: Krylov subspace methods and their applications to quantum many-body problems
- 12th: Large sparse matrices, and quantum statistical mechanics
- 13th: Parallelization for many-body problems