

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 毅

Outline in the last week

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

References: cluster update for long-range interactions

- K. Fukui and S. Todo,
"Order-N cluster Monte Carlo method for spin systems with long-range interactions"
Journal of Computational Physics, **228**, 2629 (2009).
- M. Sasaki and F. Matsubara
"Stochastic Cutoff Method for Long-Range Interacting Systems"
J. Phys. Soc. Jpn. **77** 024004 (2008).
- Both methods treat interactions **stochastically**.

Computational science using MCMC methods

Example: Application to critical phenomena

- Square lattice Ising model

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

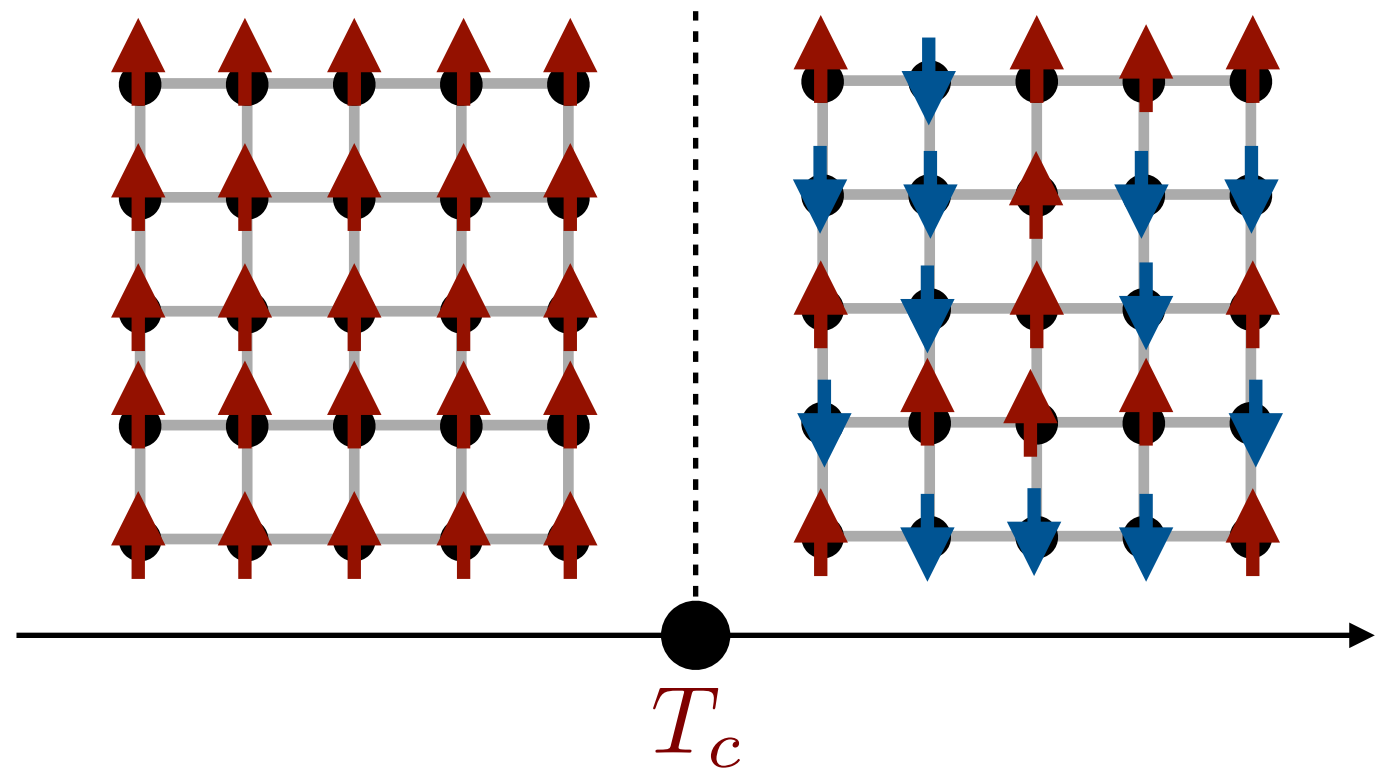
- Continuous phase transition at $T=T_c$

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

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

- Monte Carlo Simulations

- Using spinmc in **ALPS**: Simulator for classical spin system by MCMC
- (or my exercise codes)



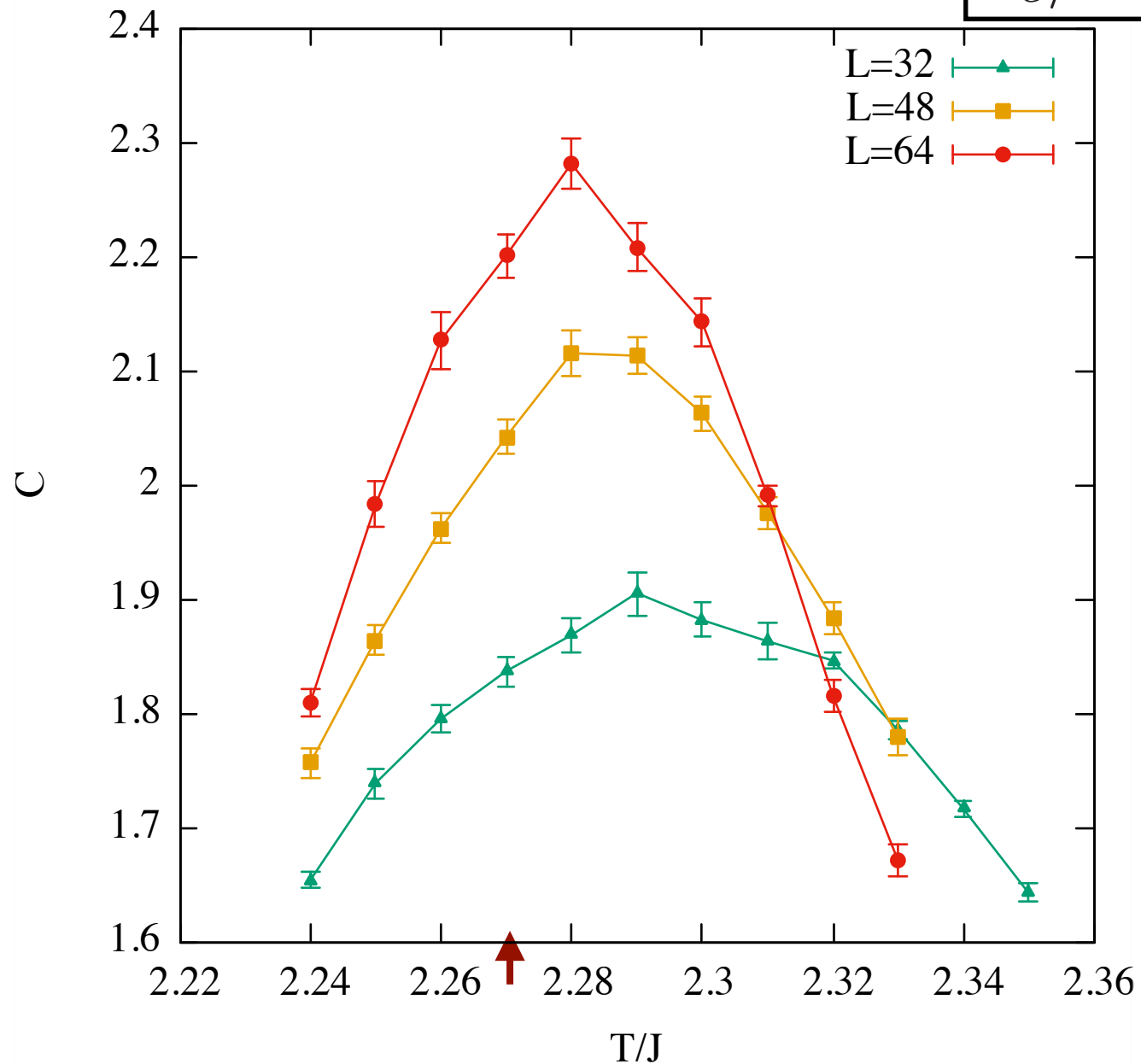
ALPS (Applications and Libraries for Physical Simulation)

- 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)

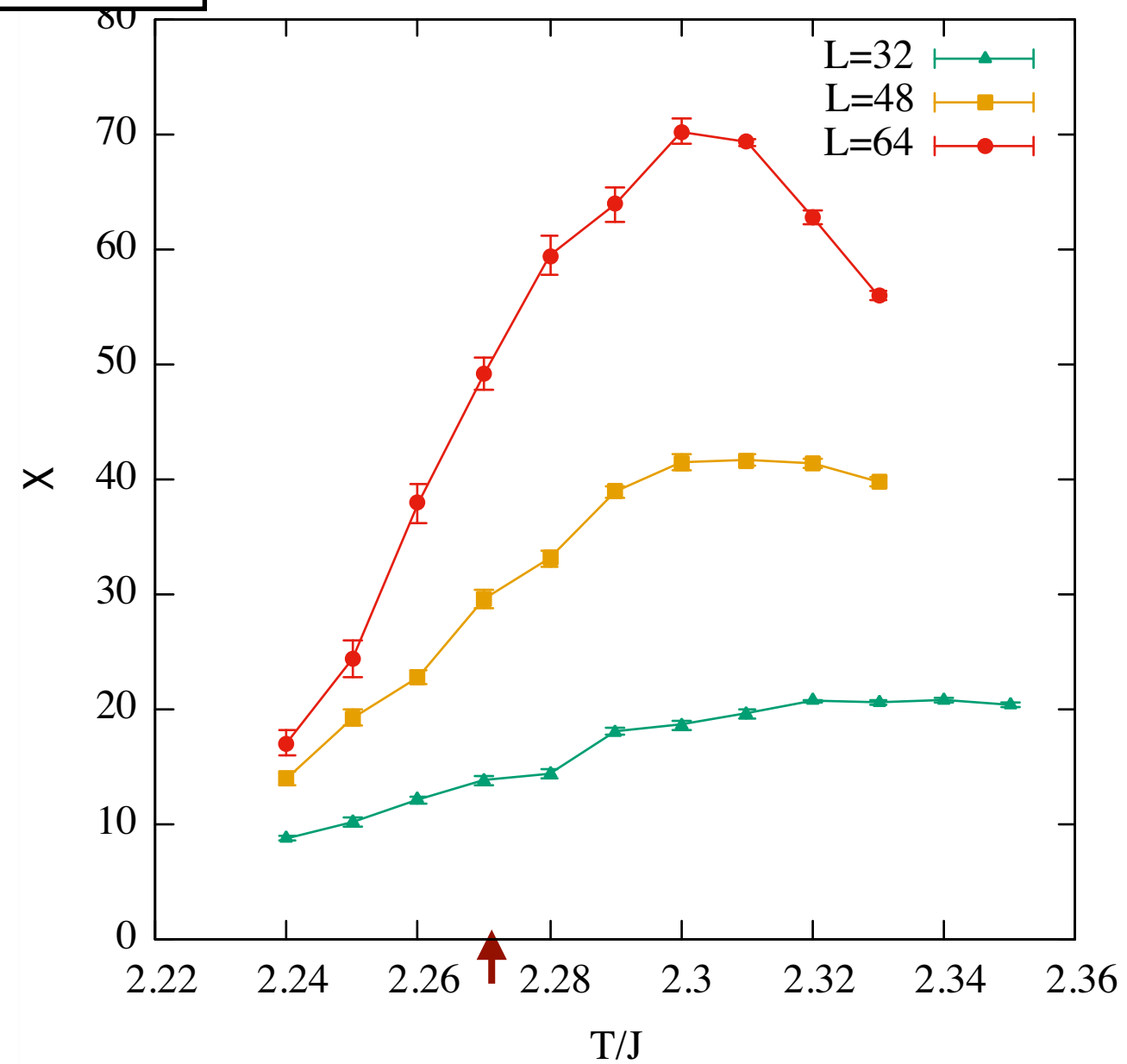
Specific heat

$$T_c/J \simeq 2.269$$



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

Susceptibility



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

Data analysis: Finite size scaling (outline)

Near the critical point (transition temperature):

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

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

$$t = T - T_c$$

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

By taking derivatives, we see

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

Physical quantity obeys **common scaling function independent of L** .

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

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 $\nu d = 2 - \alpha, 2 - \eta = \frac{\gamma}{\nu}, \dots$

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

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

 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.

Example of finite size scaling: Binder ratio

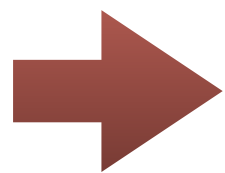
Binder ratio

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

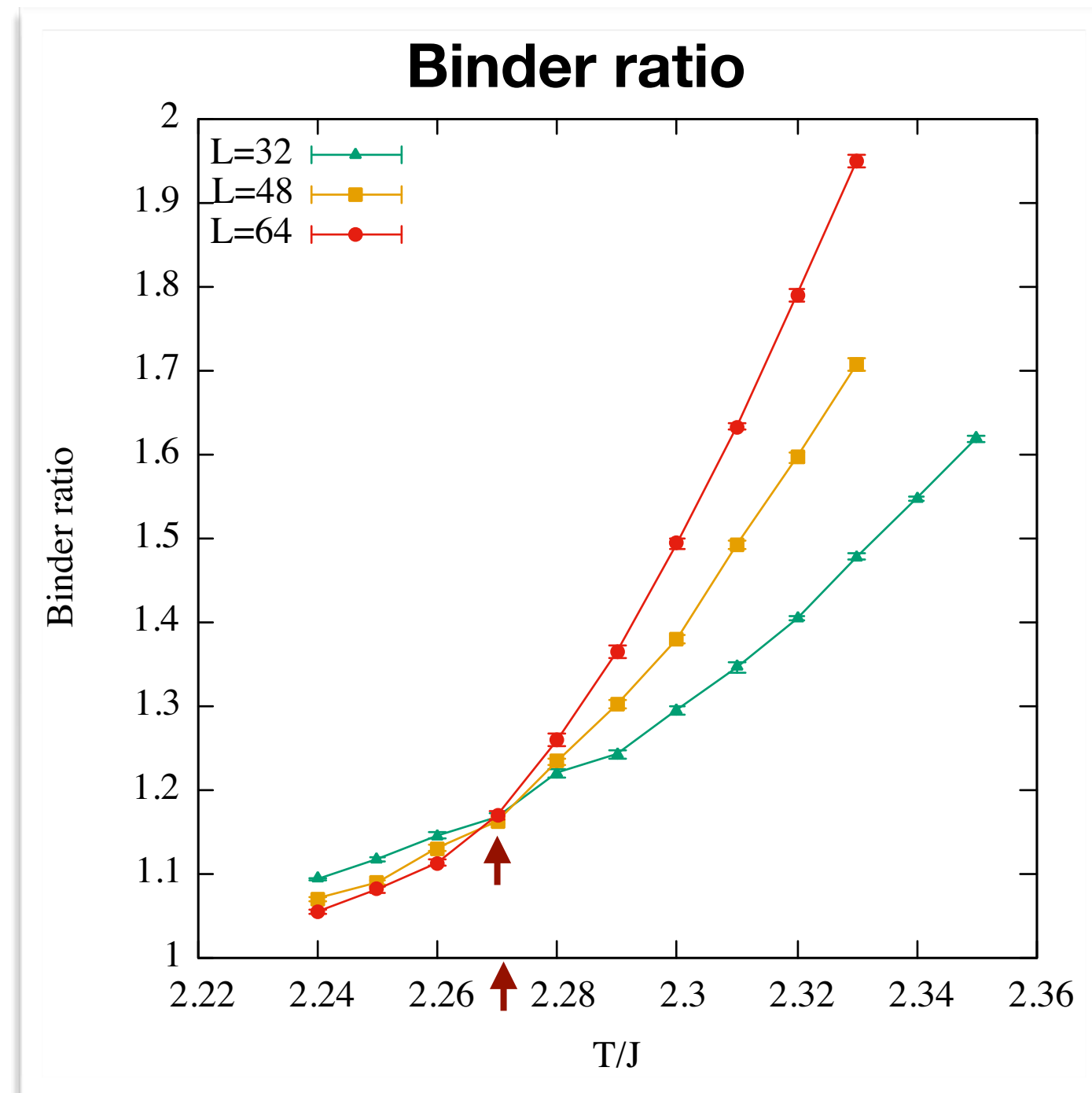
$$b = 3 \quad (T \rightarrow \infty)$$

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

The **scaling dimension** of b
is exactly zero.



At T_c , the size dependence
disappears in leading order!



$$T_c/J \simeq 2.269$$

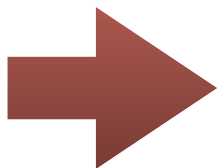
Finite size scaling

Binder ratio

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

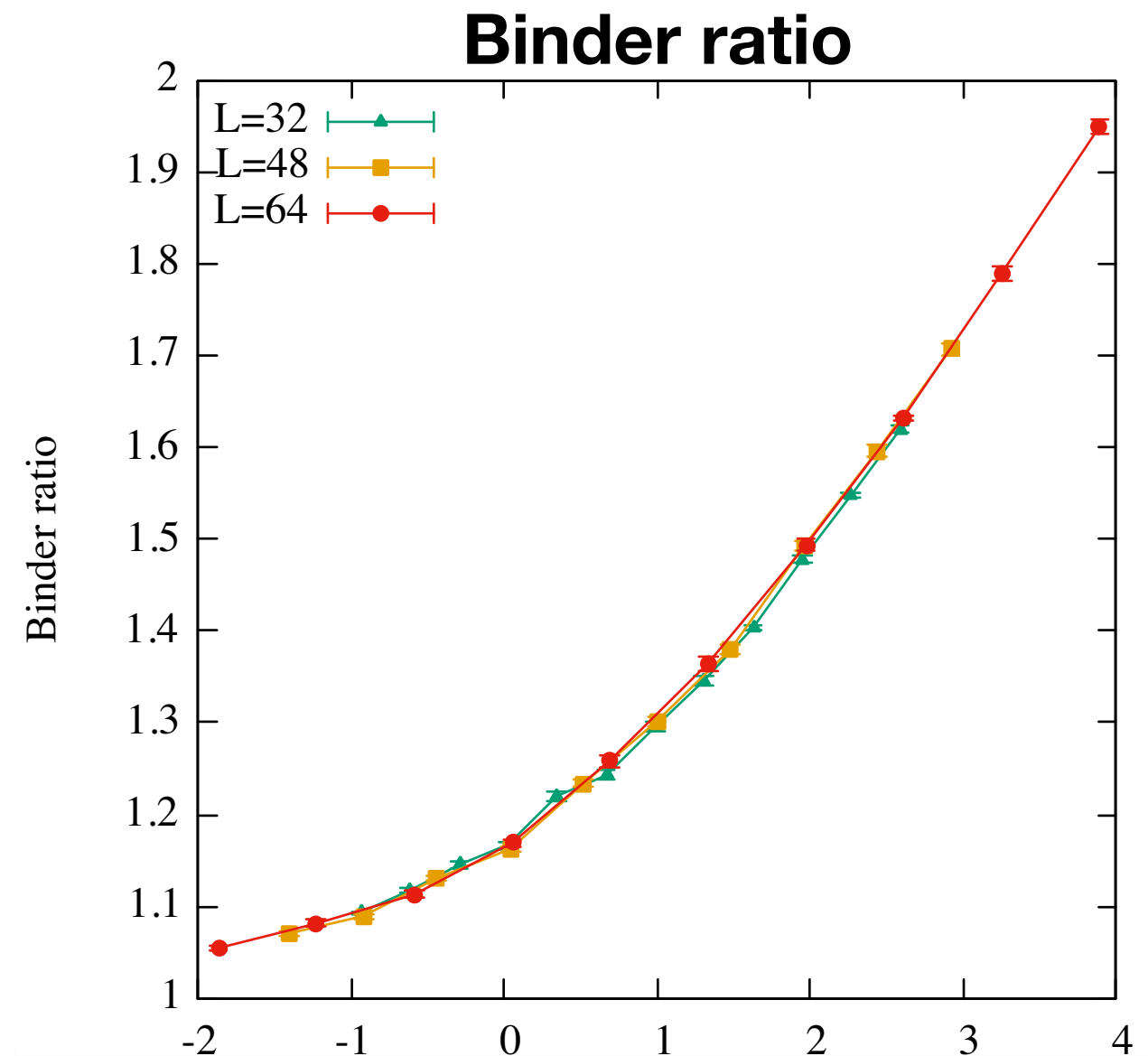
Finite size scaling around T_c

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



We can determine critical exponent!

$$\nu = 1$$



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

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 (**recommended**):

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

For python:

python Ising-Ex1.py or *python 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.

python Ising-Ex1.py -h

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

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

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

@ECCS

Unfortunately, default python is **python2.7**.

Before running my code, please change it to **python3**, e.g.,

pyenv shell anaconda3-4.4.0

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

Newtonian equation, purpose of MD simulation

Target: Newtonian mechanics

N-particle system:

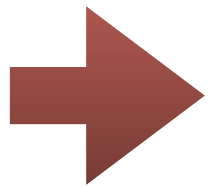
$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i(\{\mathbf{r}_i\}) \quad i = 1, 2, \dots, N$$

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, ...

$$\begin{aligned} &T(\{\mathbf{r}_i(t), \mathbf{v}_i(t)\}), \\ &P(\{\mathbf{r}_i(t), \mathbf{v}_i(t)\}), \\ &\dots \end{aligned}$$

6. Analyze trajectories

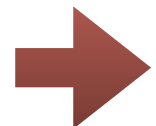
Periodic boundary condition

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



Short-range interaction

e.g. LJ potential



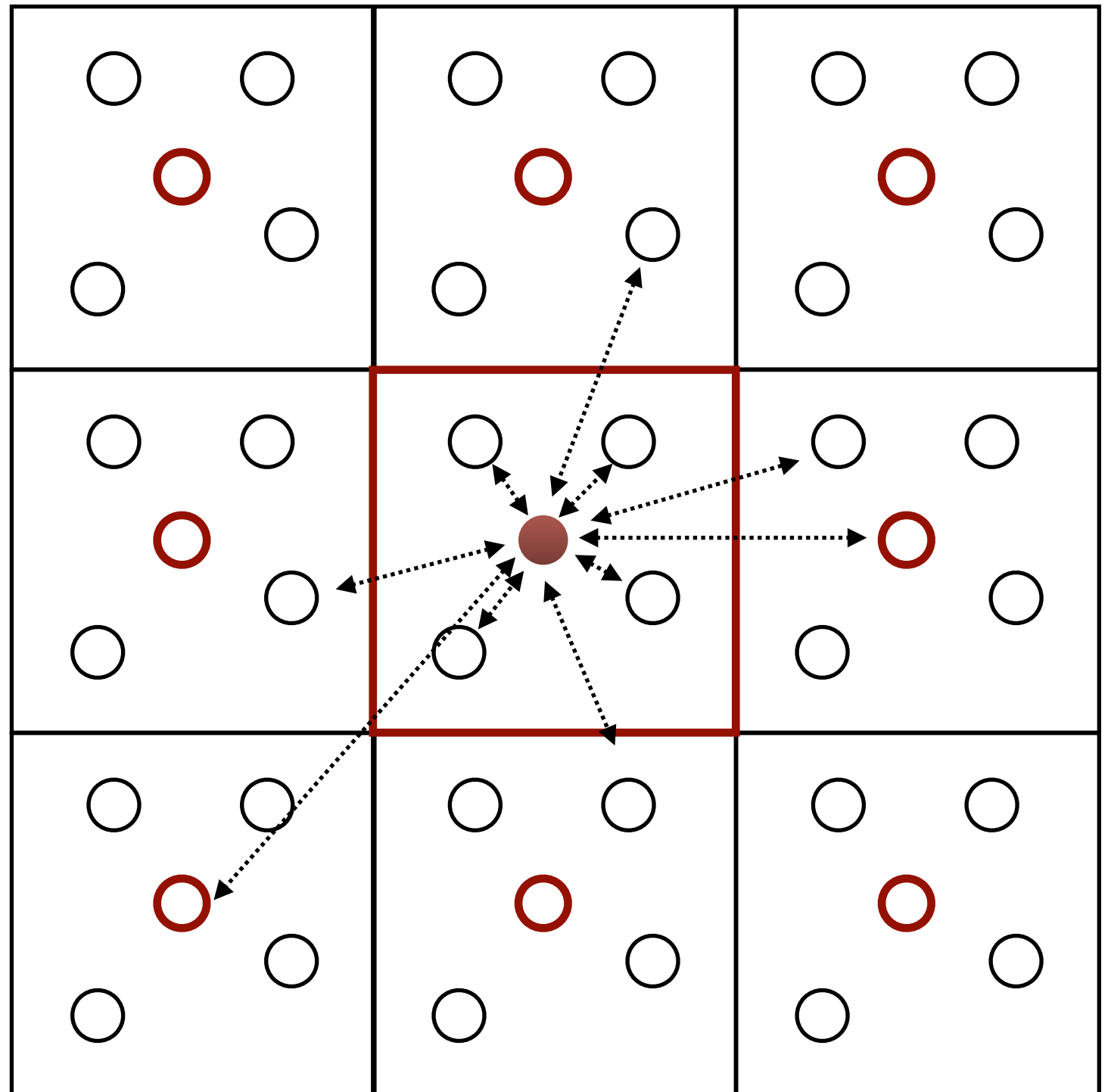
Introduce cut-off

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.

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

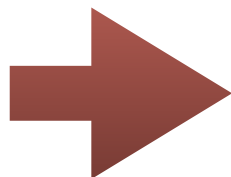
By using temperature or pressure controls, we can also obtain other ensemble averages.

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

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

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

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

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

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


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 calculate the coefficient from equilibrium simulation by using **Green-Kubo formula**:

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

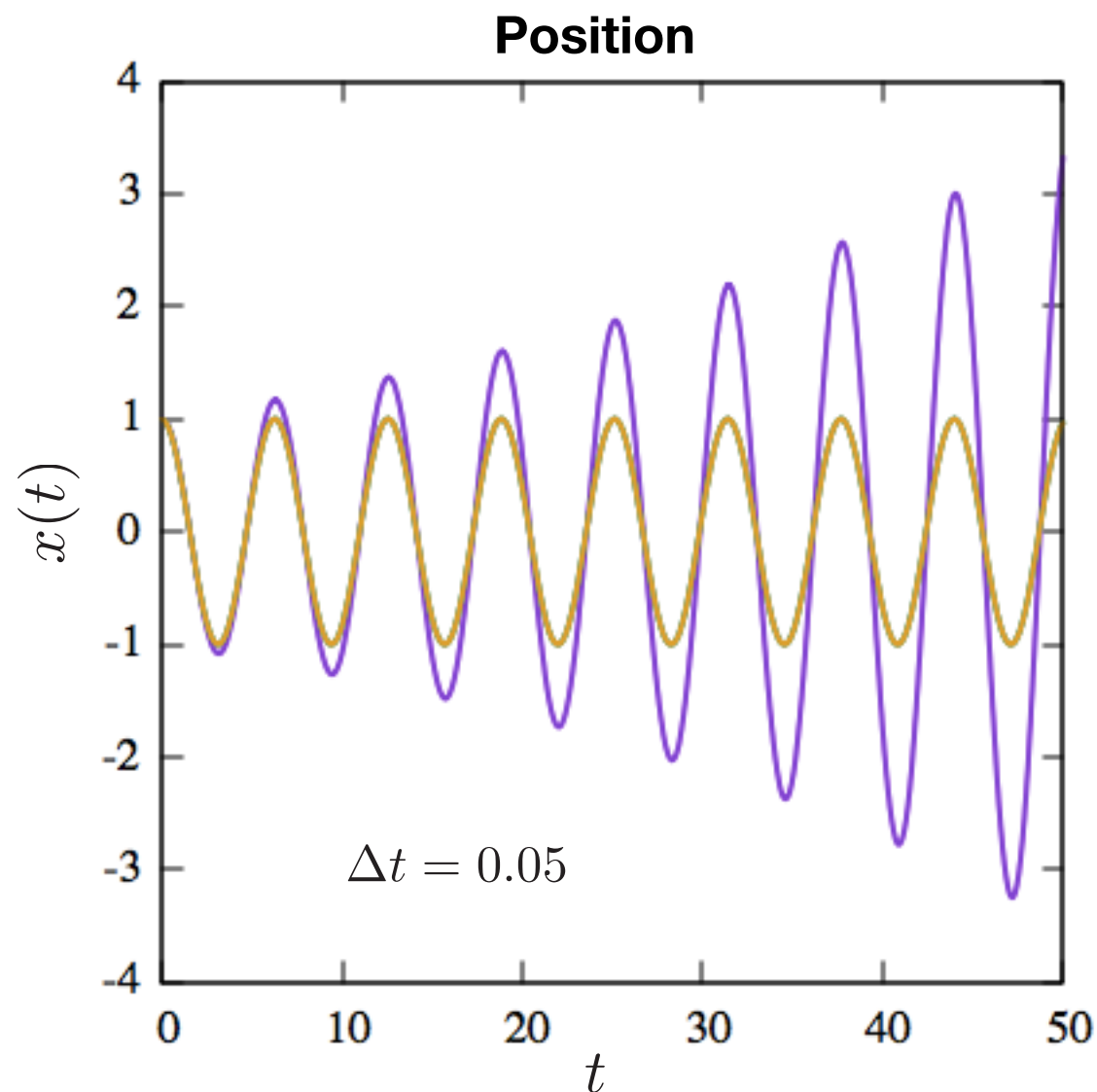
- It can be obtained by MD simulation **applying the external field**.
 - Usually the non-equilibrium calculation gives us smaller error.

Examples of numerical integrations

Numerical integration: Basics

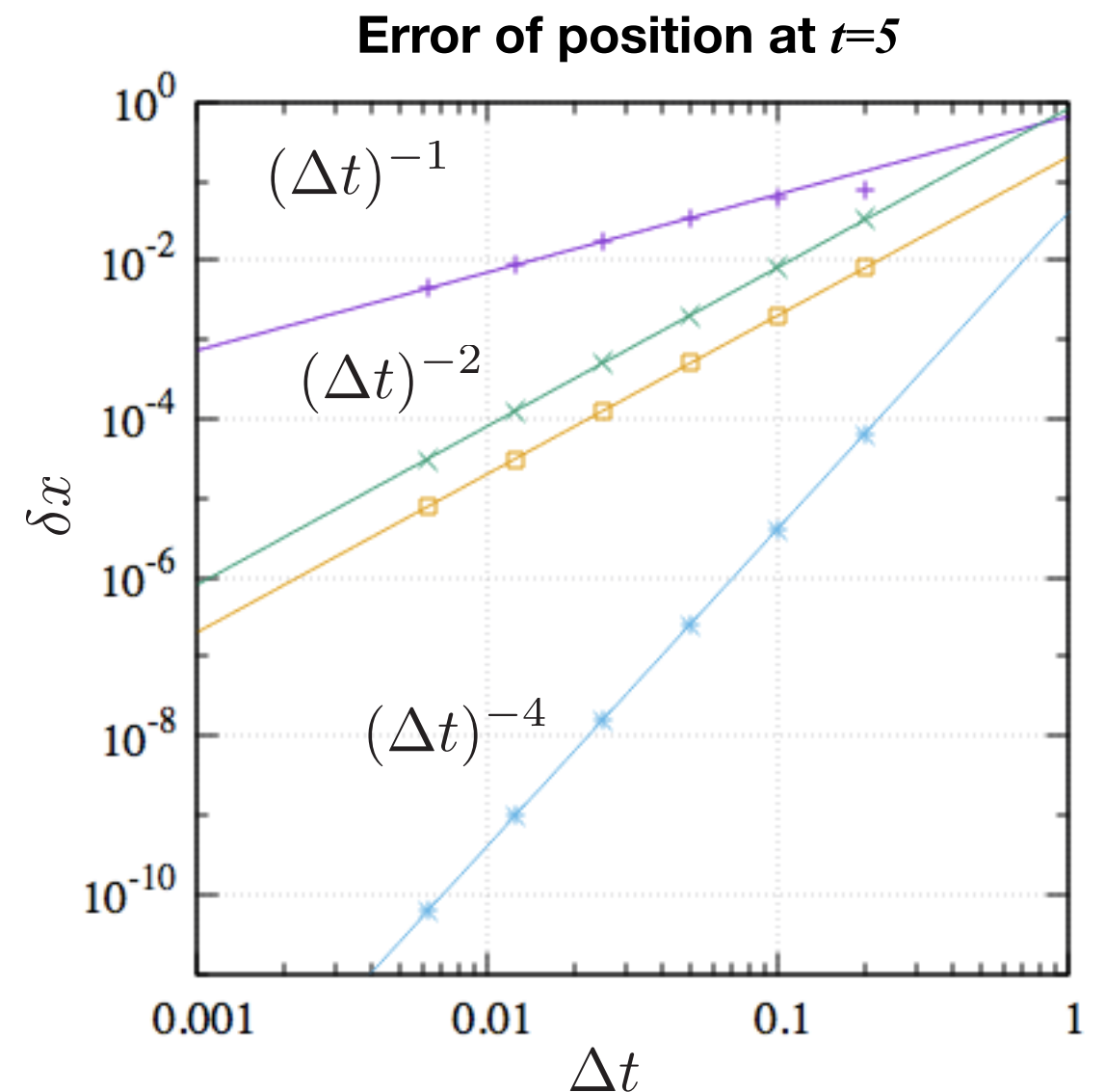
Example: 1d harmonic oscillator

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



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)	$r(0), r'(0), r''(0),$ $r'''(0), r''''(0), r'''''(0)$

Numerical integration: instability (energy drift)

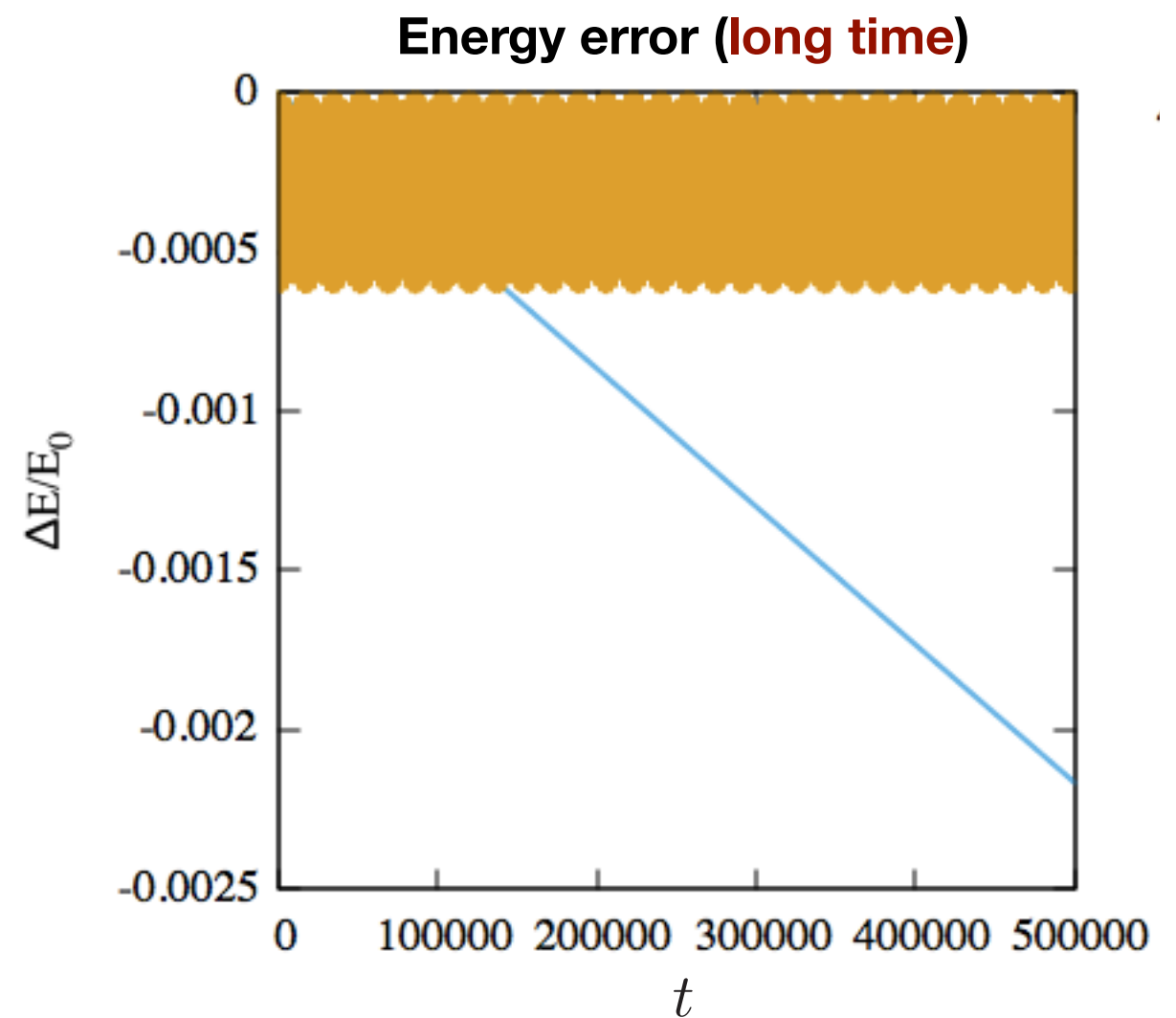
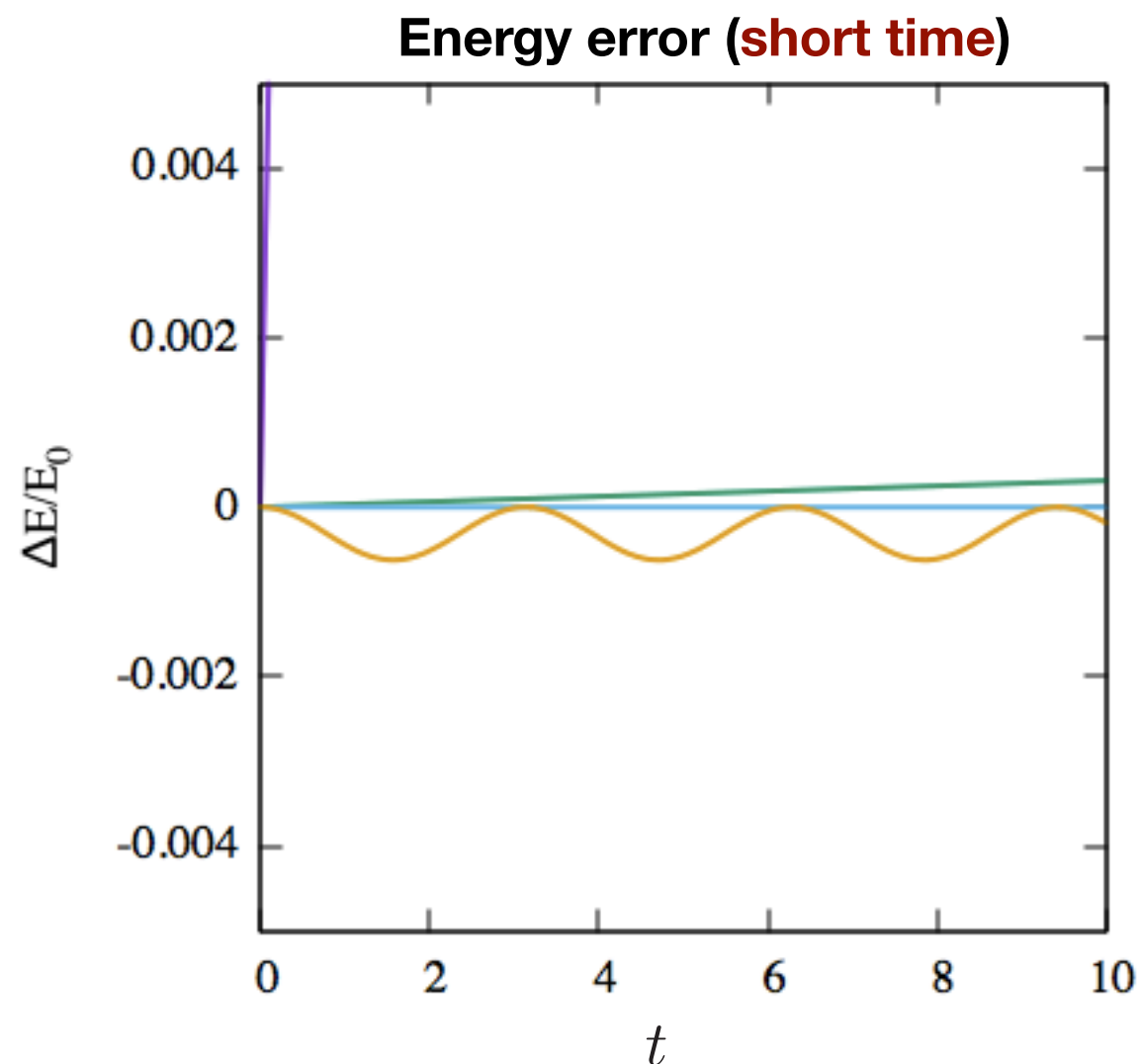
Example: 1d harmonic oscillator

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 so stable!
- Number of force calculations
 - Only 1 force calculation for 1 step

Verlet method:

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

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

Velocity Verlet method:

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

Leap-frog method:

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

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

NVE ensemble: symplectic integrator

Hamilton mechanics

Hamilton mechanics

$$\mathcal{H}(\{q_i\}, \{p_i\}) \rightarrow \begin{aligned} \frac{dq_i}{dt} &= \frac{\partial \mathcal{H}}{\partial p_i} \\ \frac{dp_i}{dt} &= -\frac{\partial \mathcal{H}}{\partial q_i} \end{aligned}$$

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

$$\frac{dA}{dt} = i\mathcal{L}A \rightarrow A(t) = \underline{e^{it\mathcal{L}}} A(0)$$

Unitary operator

Liouville's theorem

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

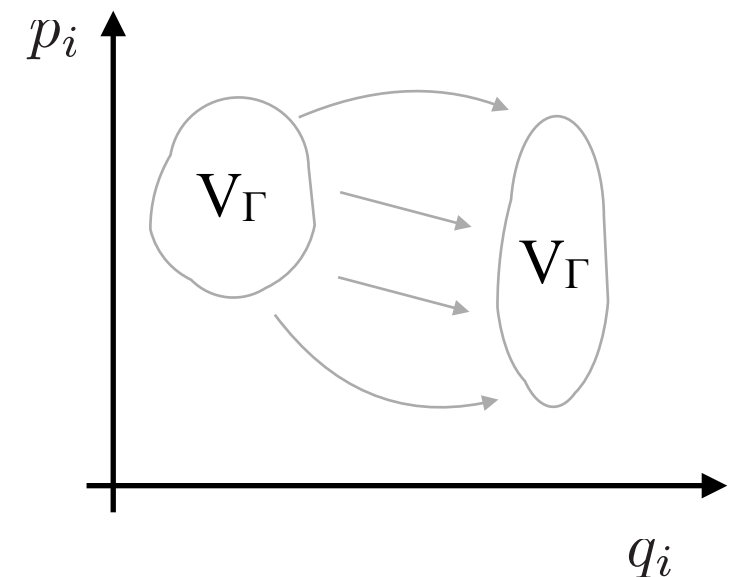
Liouville equation

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

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

Liouville's theorem:

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



Canonical transformation (正準変換)

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

For $2n$ -dim. vector representation:

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

➡ Time evolution of $\mathbf{\Gamma}$:

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

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

Symplectic condition

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

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

Time evolution of Γ' :

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

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 satisfy the symplectic condition: $S J S^T = J$

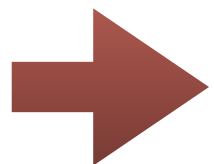
(This condition is actually the necessary and sufficient condition for canonical transformation.)

Infinitesimal time evolution

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

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

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:

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

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

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

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

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

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

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

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

Euler like equation
(but this is more stable!)

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

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

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

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

Exactly equal to
(Velocity) Verlet method

Control temperature

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

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

Results of the velocity scaling

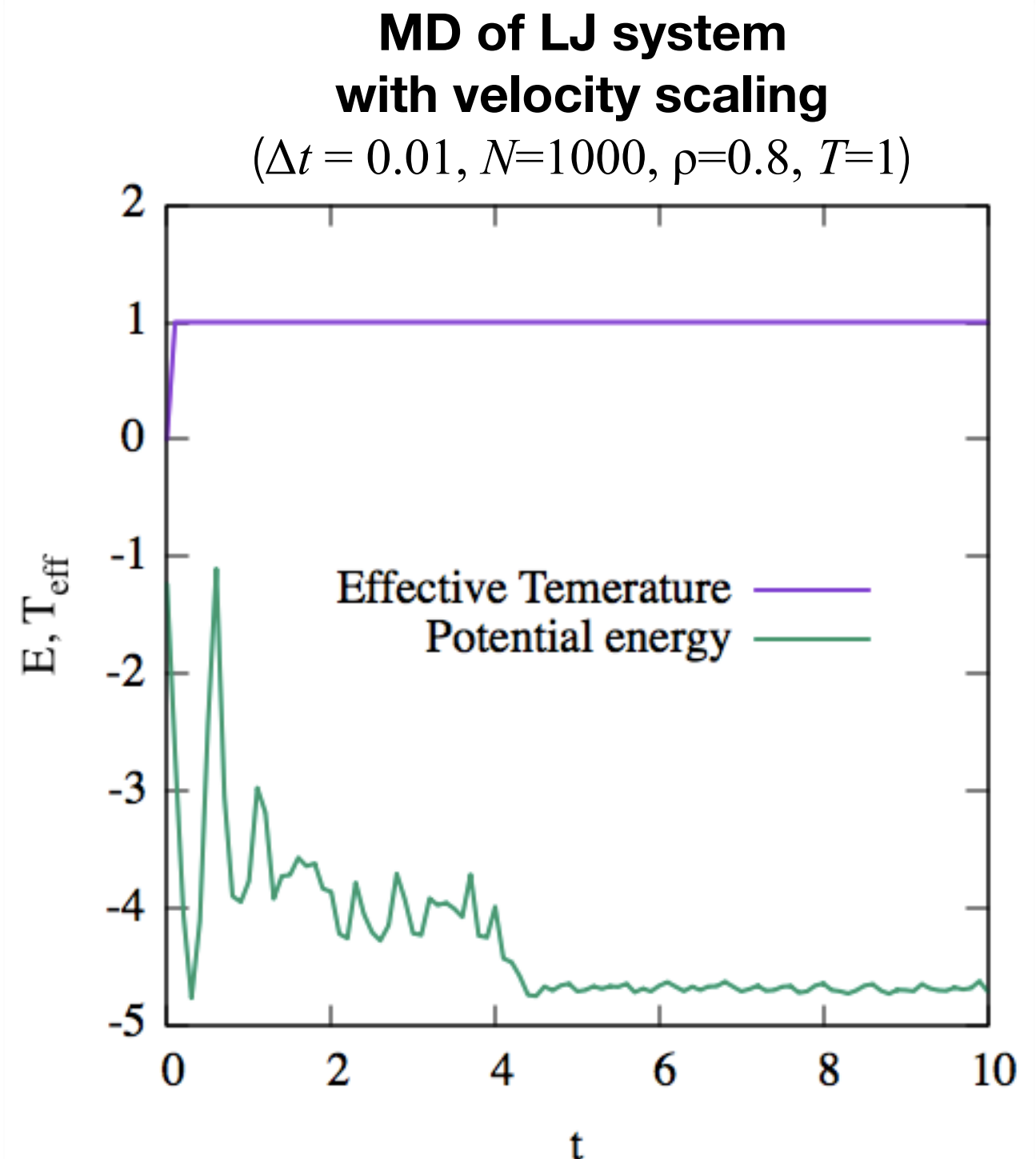
Total kinetic energy is **artificially fixed to**

$$K = \frac{3}{2}Nk_B T$$

➔ 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

$$\frac{d\mathbf{p}_i}{dt} = \mathbf{F}_i(\{\mathbf{q}_i\}) - \underbrace{\gamma\mathbf{p}_i}_{\text{Dissipation}} + \underbrace{\mathbf{R}_i}_{\text{Random force}}$$

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”

$$\mathcal{H}_N = \underbrace{\sum_i \frac{(\mathbf{p}'_i)^2}{2m_i s^2} + V(\{\mathbf{q}_i\})}_{\text{Original Hamiltonian with scaled momentum}} + \underbrace{\frac{P_s^2}{2Q} + gk_B T \ln s}_{\text{Heat-bath}}$$

Original Hamiltonian
with scaled momentum

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

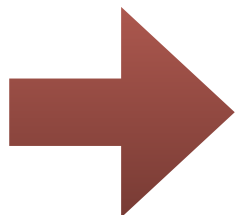
Heat-bath

s : scale factor for time

$$t' = st$$

$$\mathbf{p}'_i = s\mathbf{p}_i$$

Canonical equation
(along 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).)

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

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

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

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

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

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

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', t') dynamics yields NVE ensemble of H_N

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

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

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

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

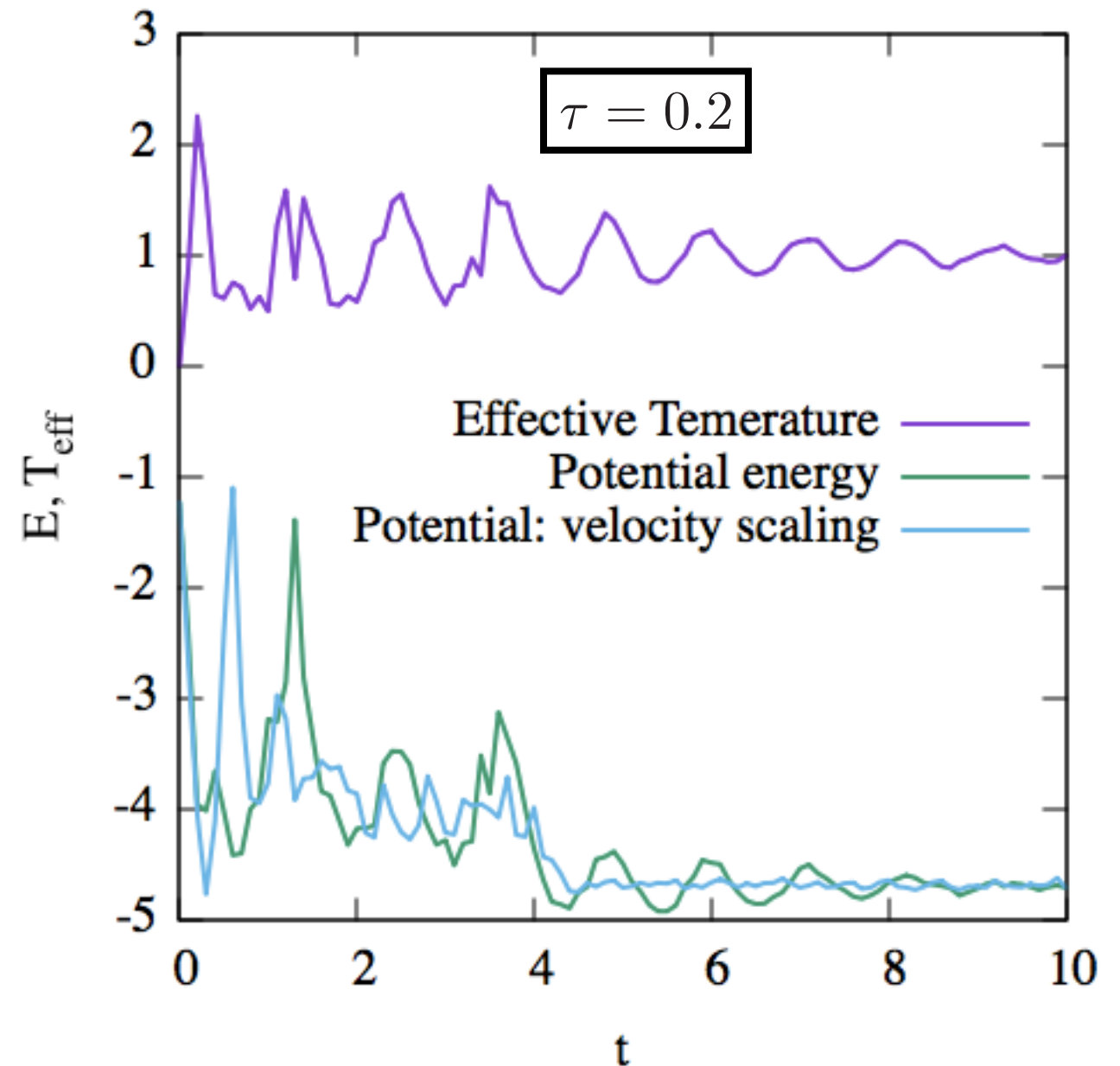
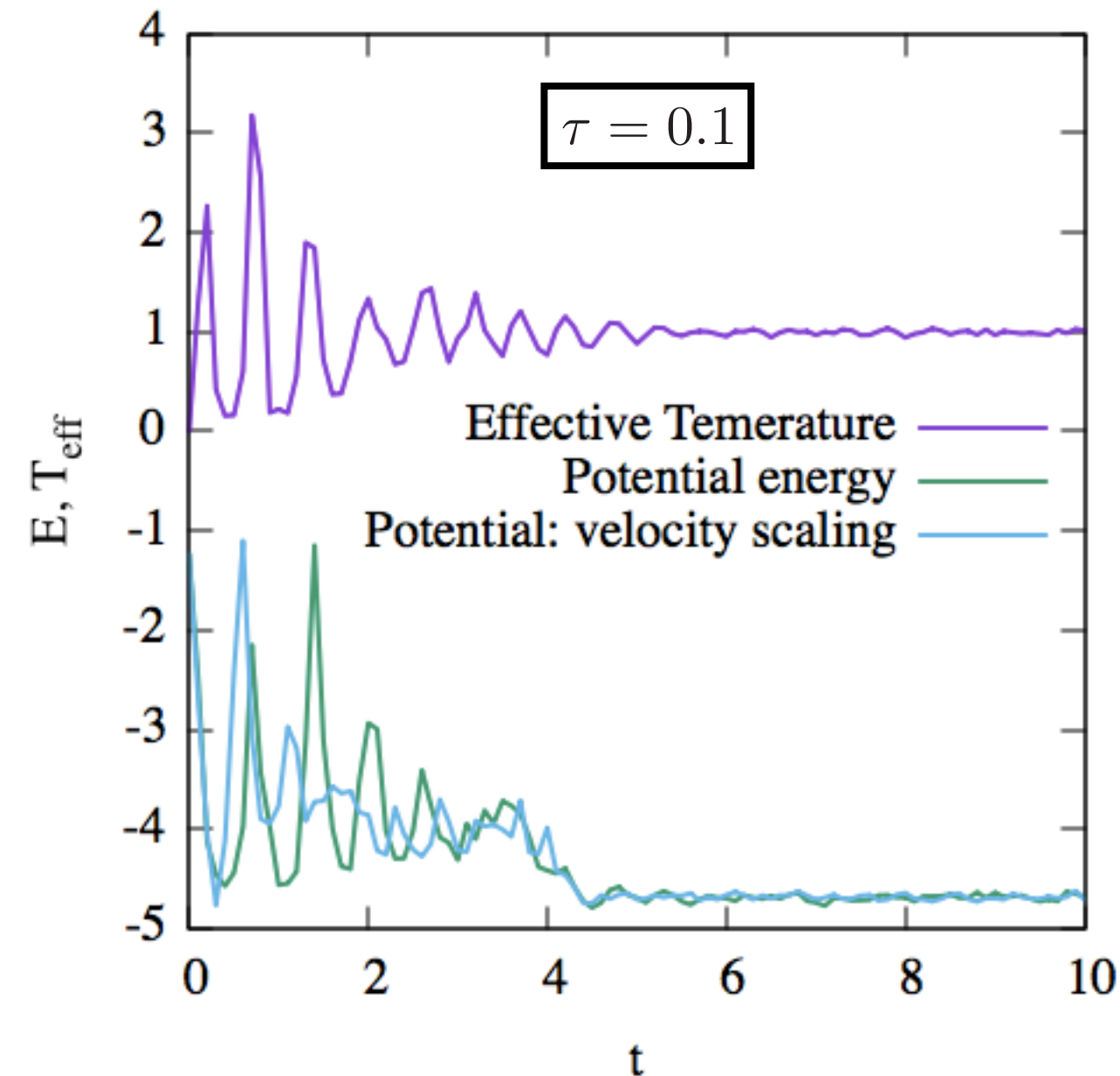
\rightarrow Canonical ensemble if $g = 3N$

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 (might be skipped)

Pressure control: Andersen method

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

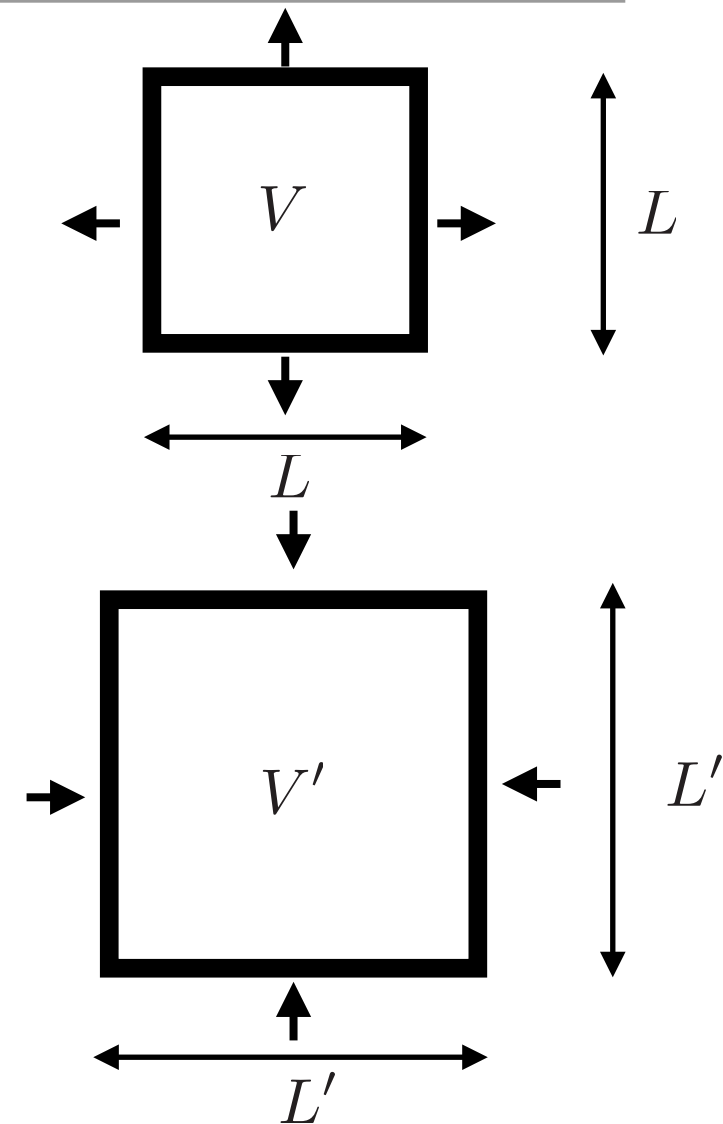
Extended Hamiltonian System with a “piston”

$$\mathcal{H}_A = \underbrace{\sum_i \frac{\tilde{\mathbf{p}}_i^2}{2m_i V^{\frac{2}{3}}}}_{\text{Original Hamiltonian with scaled coordinate and momentum}} + \underbrace{V_p(\{V^{\frac{1}{3}} \tilde{\mathbf{q}}_i\}) + \frac{P_V^2}{2M} + PV}_{\text{Piston}}$$

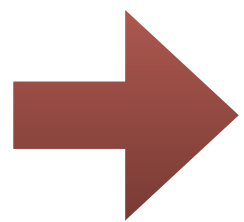
Original Hamiltonian with scaled coordinate and momentum

Piston

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



Canonical equation



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

Pressure control: Andersen method

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

In original coordinates

$$\begin{aligned}\frac{d\mathbf{q}_i}{dt} &= \frac{\mathbf{p}_i}{m_i} + \frac{\dot{V}}{3V}\mathbf{q}_i & \frac{dV}{dt} &= \frac{P_V}{M} \\ \frac{d\mathbf{p}_i}{dt} &= \mathbf{F}_i - \frac{\dot{V}}{3V}\mathbf{p}_i & \frac{dP_V}{dt} &= \frac{1}{3V} \sum_i \left[\frac{\mathbf{p}_i^2}{m_i} + \mathbf{F}_i \cdot \mathbf{q}_i \right] - P \\ & & & \underbrace{\hspace{10em}}_{P_{\text{eff}} : \text{virial theorem}} \\ & & & = P_{\text{eff}} - P\end{aligned}$$

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 = Enthalpy

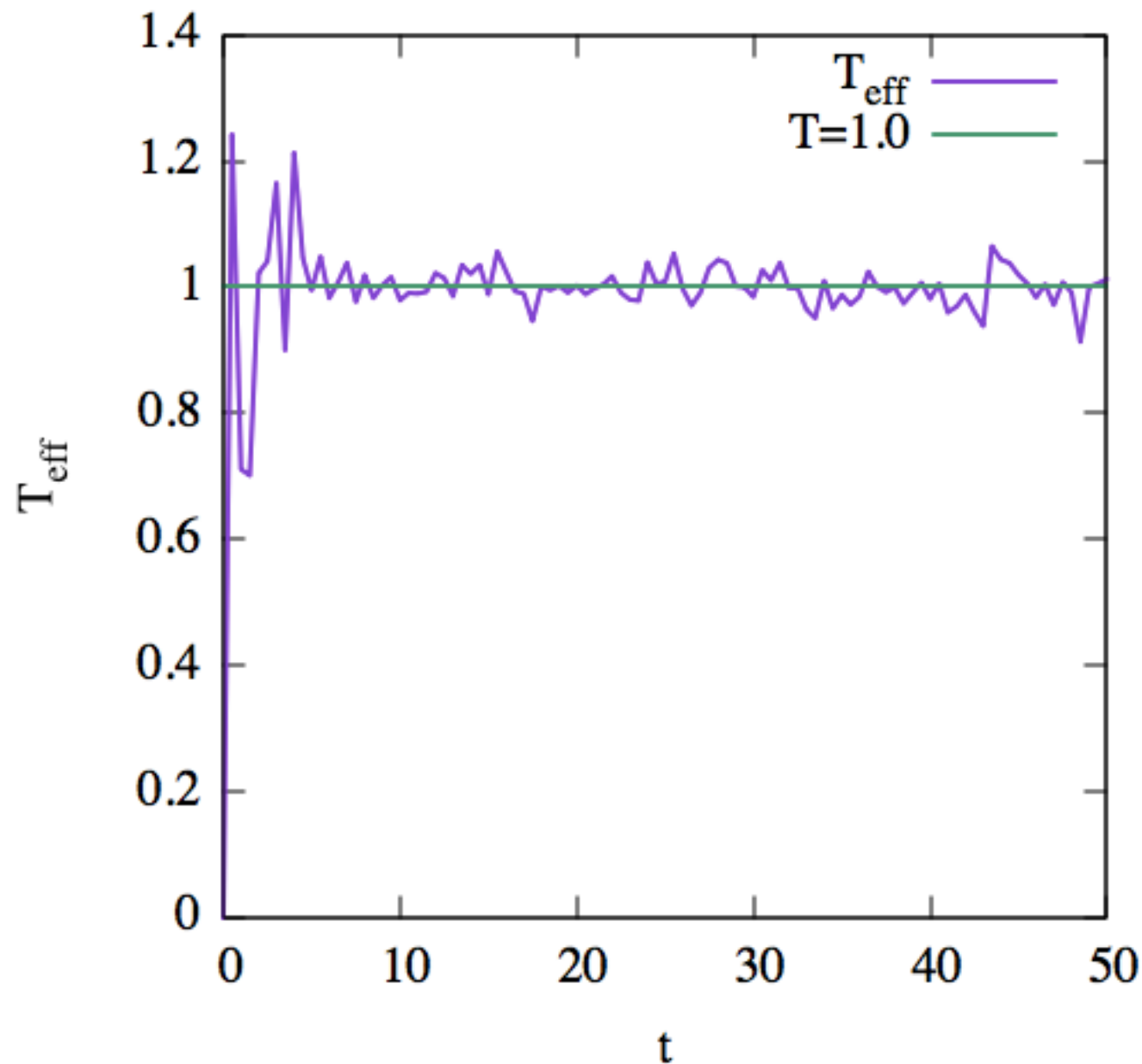
NPT ensemble

MD of LJ system

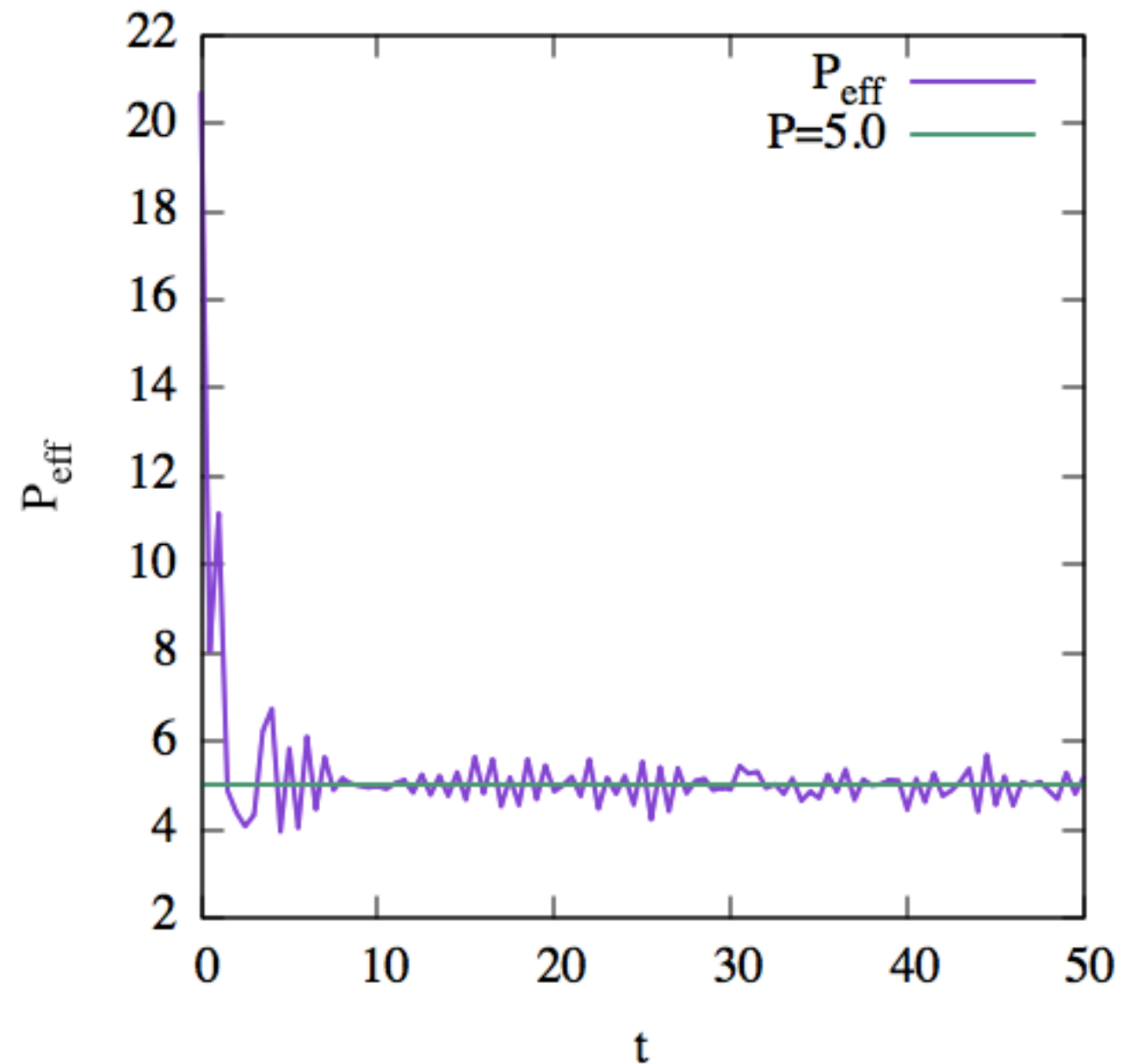
($\Delta t = 0.005$, $N=1000$, $T=1$, $P=5$)

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 can 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**).
 - In order to run the sample code you need
 - numpy, and numba (numba is used for speed up)

Next week (5/21)

Classical

1st: Introduction

2nd: Difficulties in many-body problems

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 statistical models and numerical simulation

8th: Quantum Monte Carlo methods

Quantum

9th: Applications of quantum Monte Carlo methods

10th: Quantum many-body problems and huge sparse matrices

11th: Krylov subspace methods and its applications

12th: Sparse matrices and quantum statistical mechanics

13th: Parallelized algorithm in many-body problems