MHPC
Molecular dynamics

Sabine Reißer

Thermostat

```
// Read initial positions
read_positions
// Set random velocities
randomize_velocities

// Compute initial neighbor list
 compute_list
// compute initial forces
compute_forces

for(istep=0;istep<nstep;istep++){
    thermostat(dt/2)
    velocity += (force*dt/2)/mass
    position += velocity*dt

    // Check whether the neighbour list has to be recomputed
    check_list

    if(recompute_list){
      compute_list
    }

    compute_forces

    velocity += (force*dt/2)/mass
    thermostat(dt/2)
}
```

Thermostat

Energy constant
Number of particles constant
Volume constant

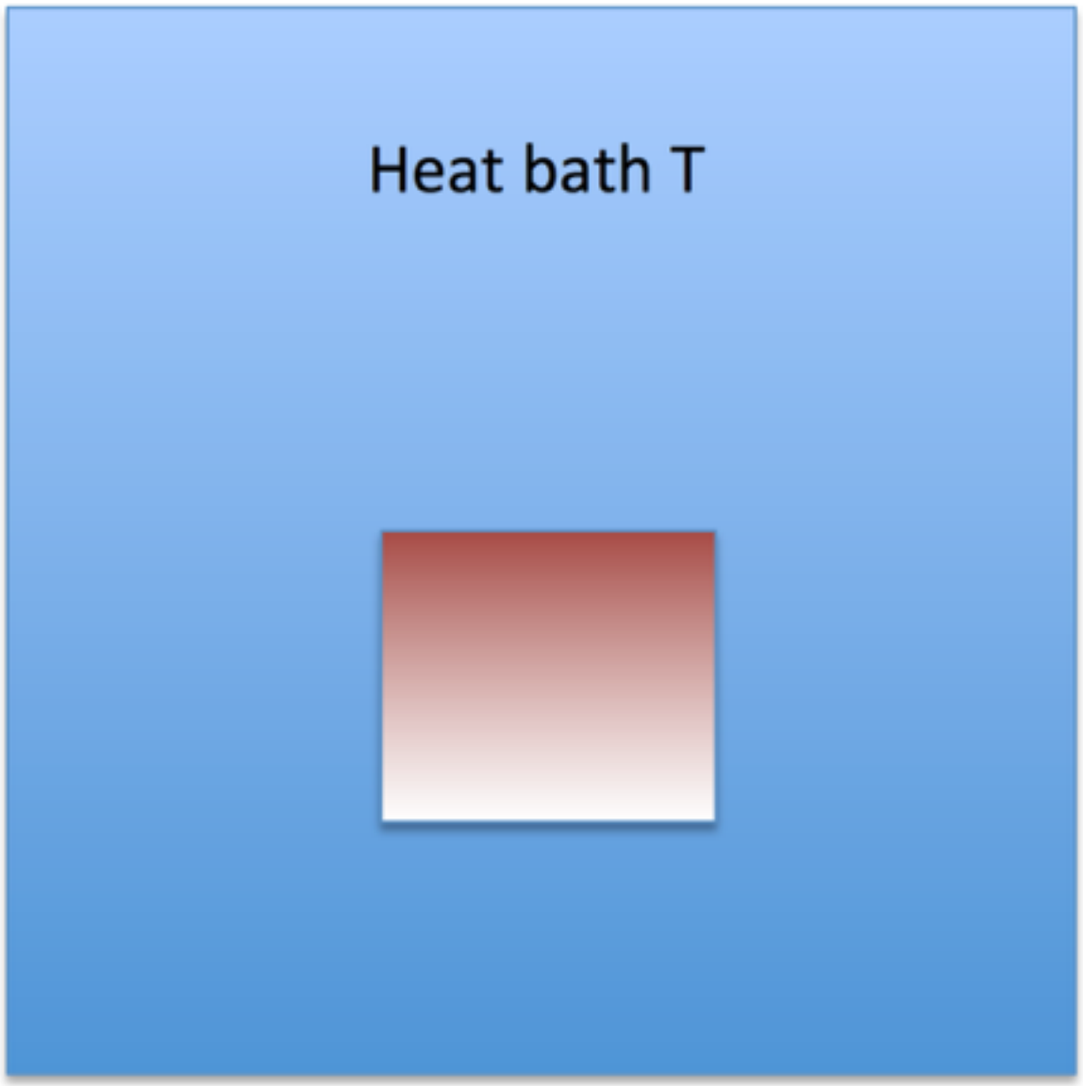
NVE
(microcanonical ensemble)

Temperature constant
Number of particles constant
Volume constant

NVT
(Canonical ensemble)

Thermostat: algorithm to control temperature in MD simulation

Canonical ensemble



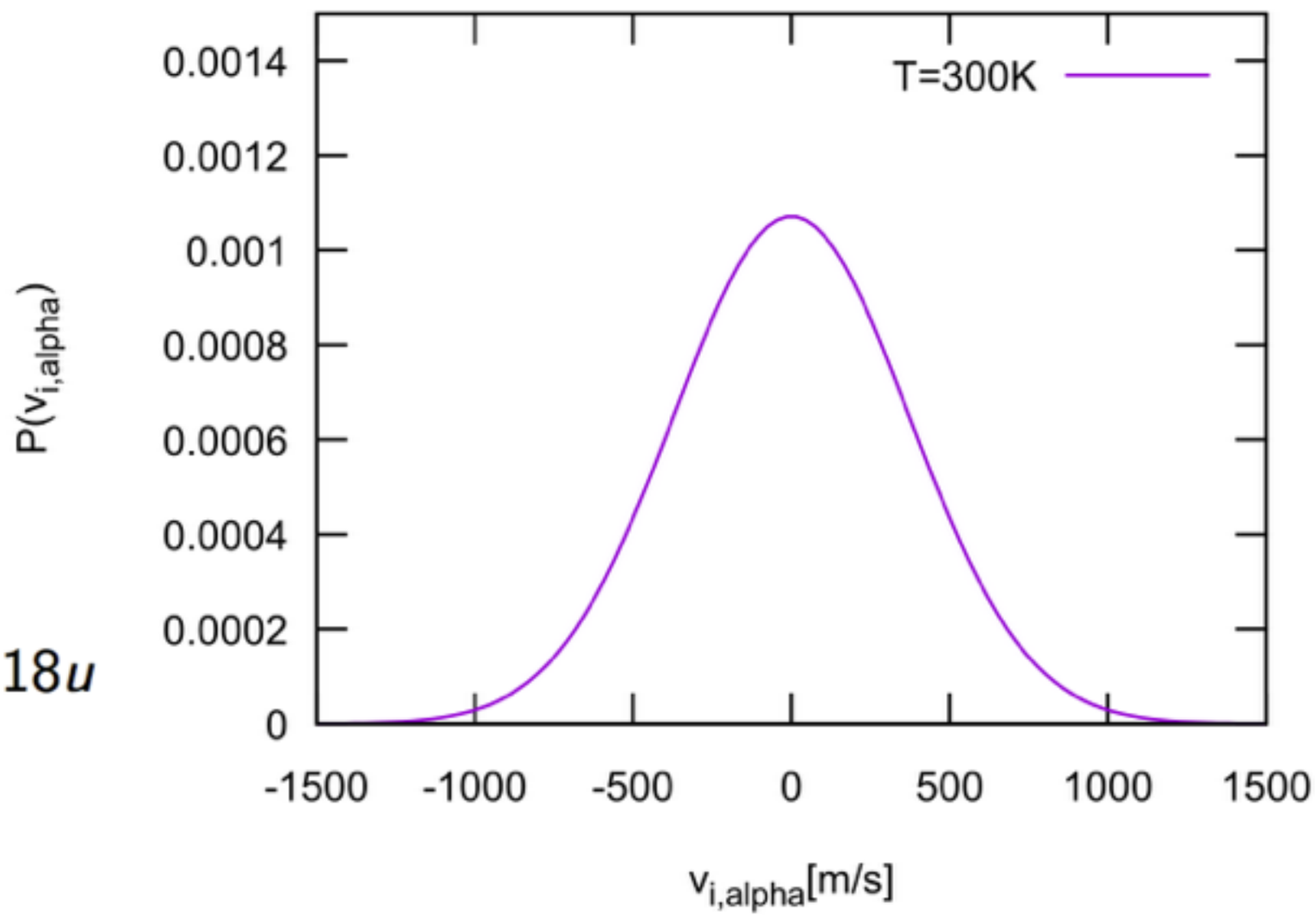
System at thermal equilibrium with heat bath at temperature T

MD in canonical ensemble

Velocities according to Maxwell-Boltzmann distribution

$$P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_{i,\alpha}}{2 k_B T}\right)$$

$$m = m_{H_2O} = 18u$$

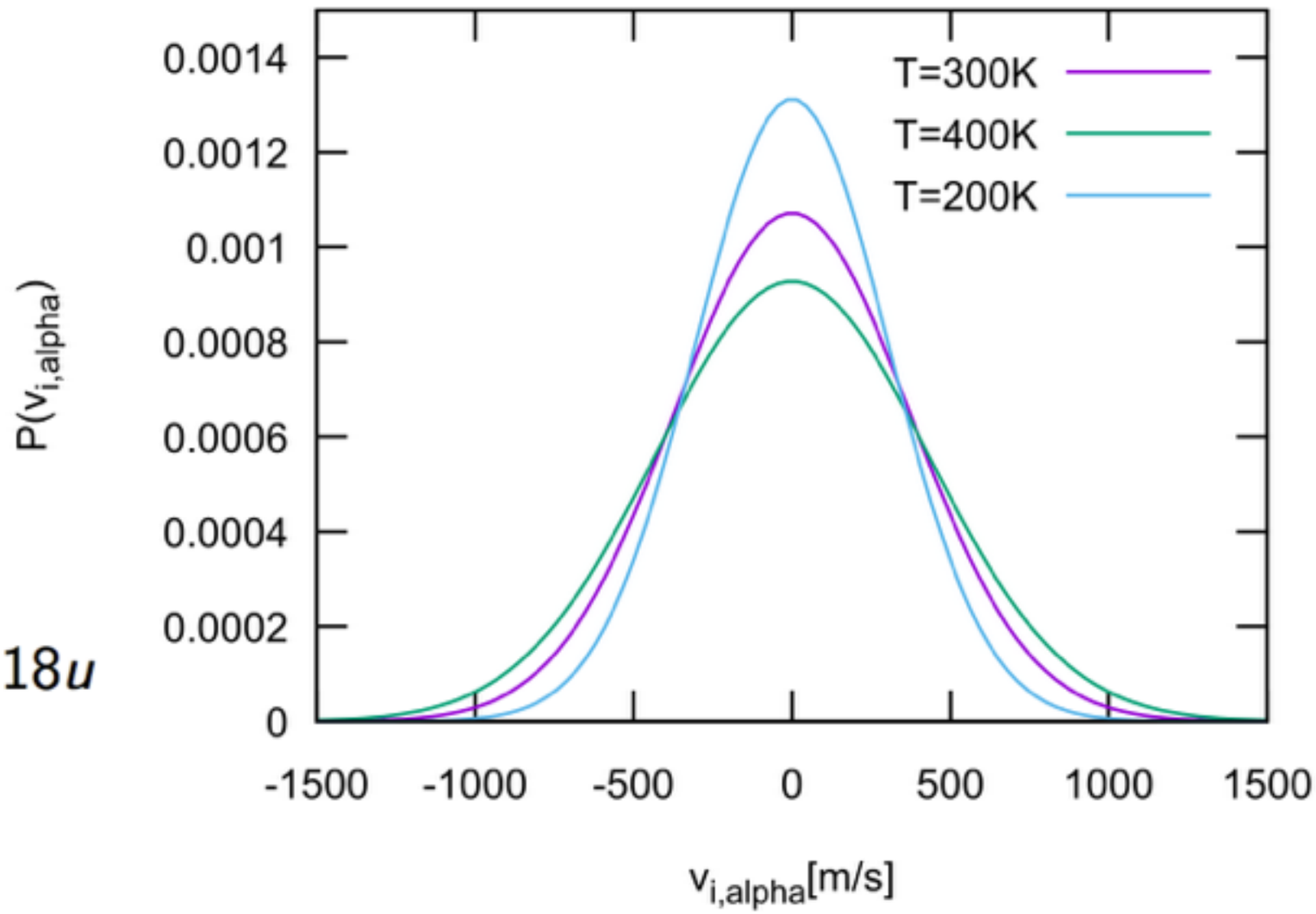


MD in canonical ensemble

Velocities according to Maxwell-Boltzmann distribution

$$P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_{i,\alpha}}{2 k_B T}\right)$$

$$m = m_{H_2O} = 18u$$



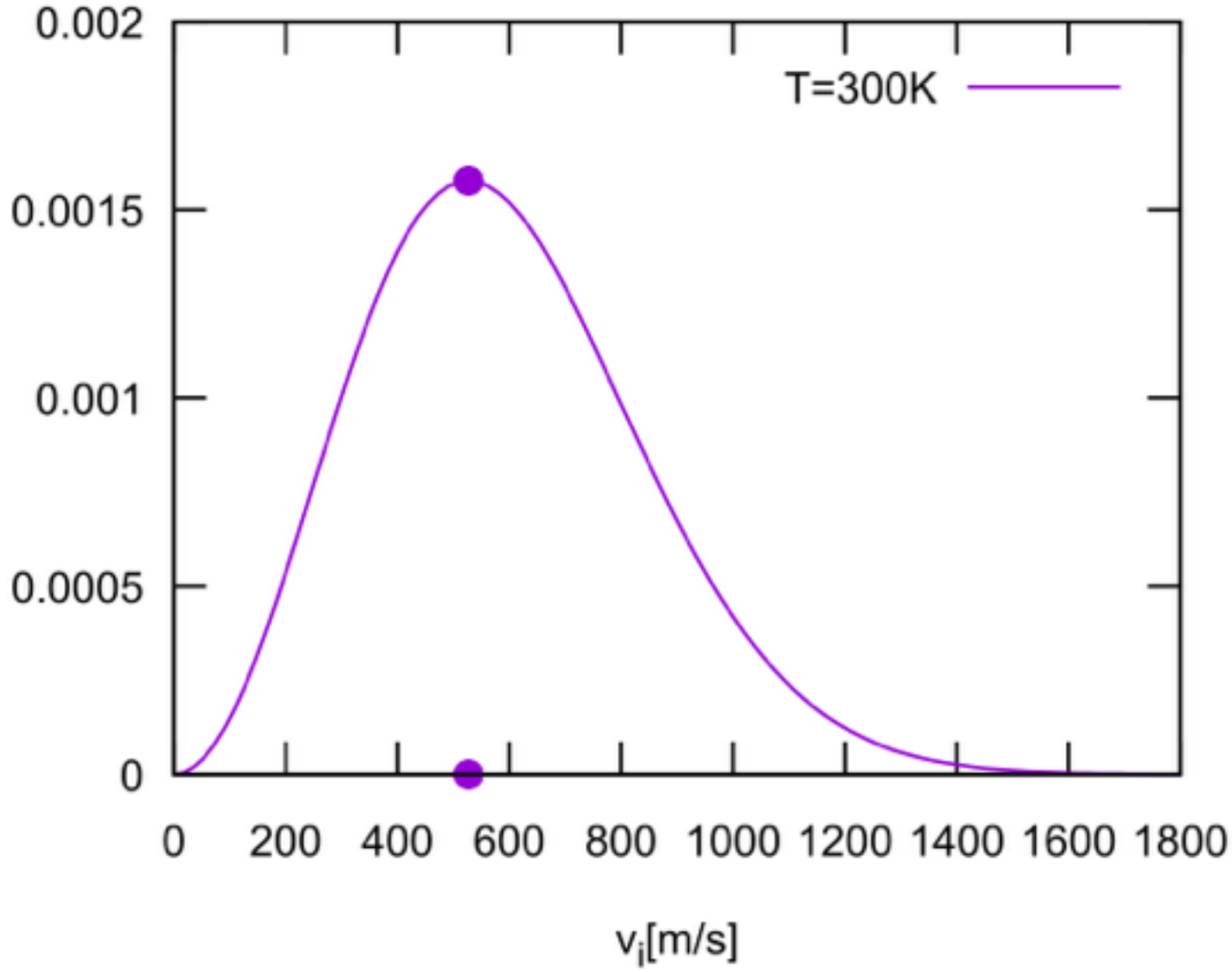
MD in canonical ensemble

Velocities according to Maxwell-Boltzmann distribution

$$P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_{i,\alpha}}{2 k_B T}\right)$$

$$P(v_i) = 4\pi \left(\frac{m}{2\pi k_B T}\right)^{3/2} v_i^2 \exp\left(-\frac{m v_i^2}{2 k_B T}\right)$$

$$m = m_{H_2O} = 18u$$



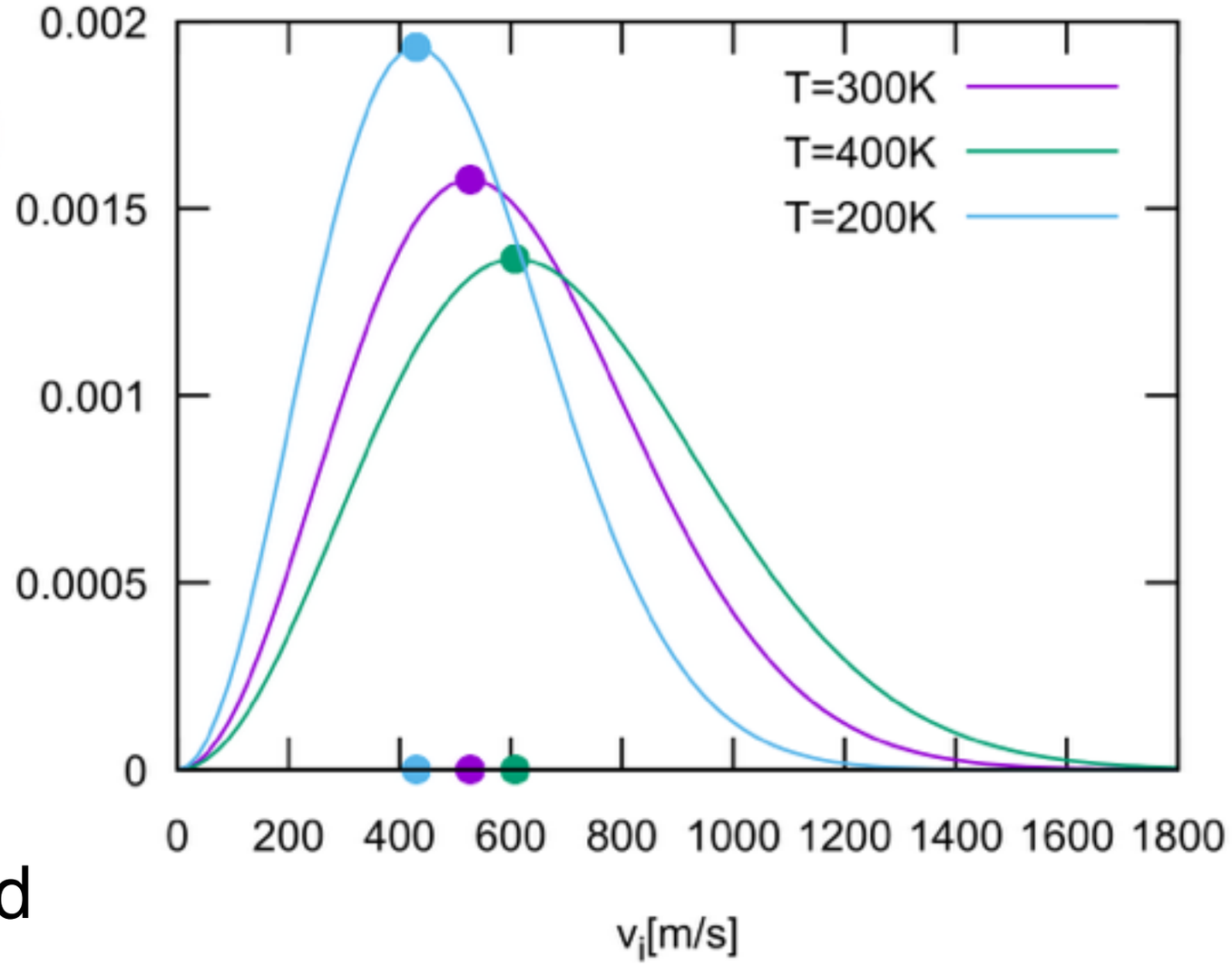
MD in canonical ensemble

Velocities according to Maxwell-Boltzmann distribution

$$P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_{i,\alpha}}{2 k_B T}\right)$$

$$P(v_i) = 4\pi \left(\frac{m}{2\pi k_B T}\right)^{3/2} v_i^2 \exp\left(-\frac{m v_i^2}{2 k_B T}\right)$$

$$m = m_{H_2O} = 18u$$



Typical speed ≈ speed of sound

MD in canonical ensemble

Velocities according to Maxwell-Boltzmann distribution

$$P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_{i,\alpha}}{2 k_B T}\right)$$

Equipartition theorem:

average kinetic energy per degree of freedom related to T

$$\frac{1}{2} m_i \left\langle v_{i,\alpha}^2 \right\rangle = \frac{1}{2} k_B T$$

How to control the temperature?

Calculate instantaneous temperature:

$$k_B T(t) = \frac{1}{N_f} \sum_{i,\alpha} m_i v_{i,\alpha}^2(t)$$

Rescale velocities:

$$v_{i,\alpha}^{new} = \sqrt{\frac{T_{ref}}{T}} v_{i,\alpha}^{old}$$

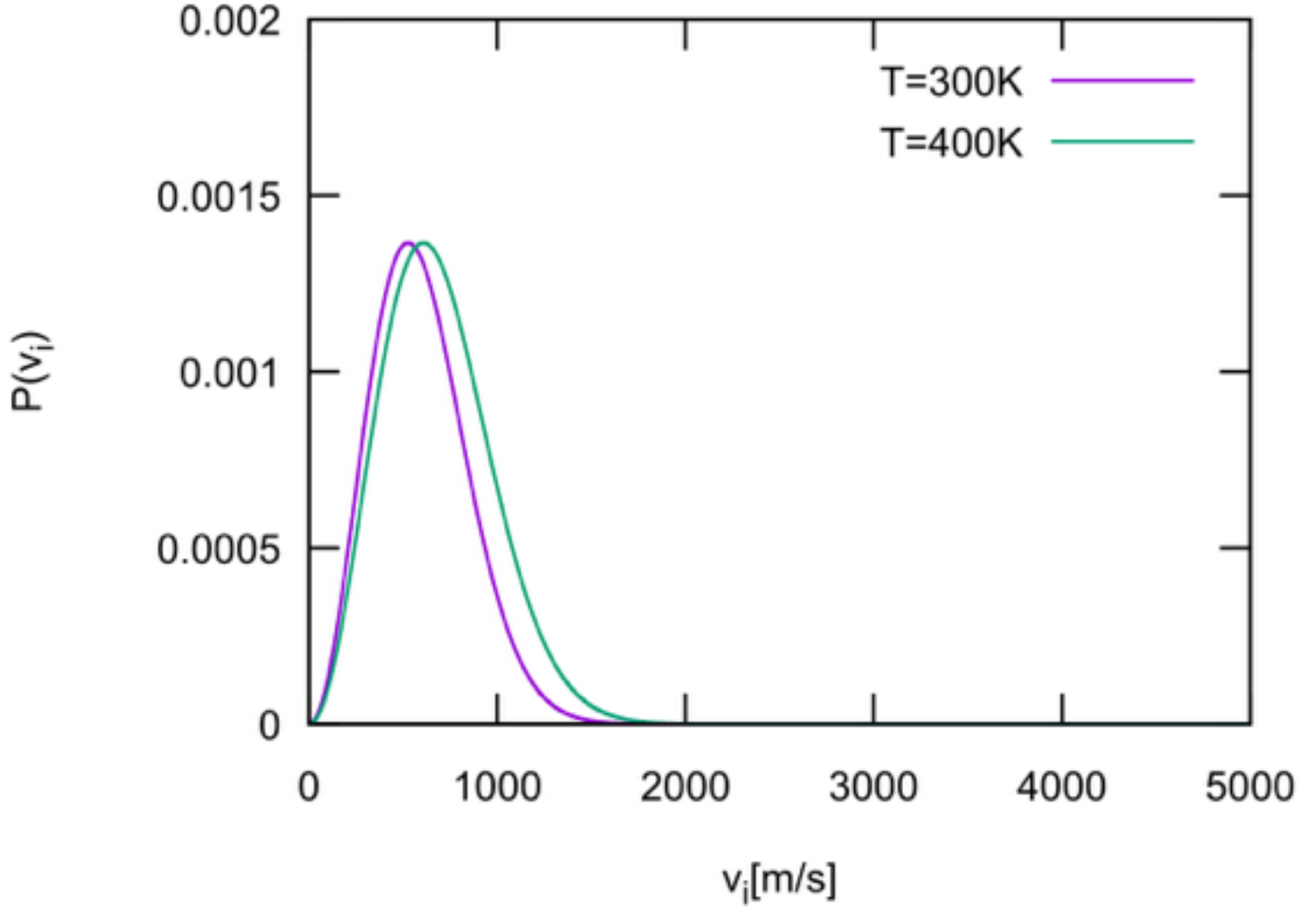
How to control the temperature?

Calculate instantaneous temperature:

$$k_B T(t) = \frac{1}{N_f} \sum_{i,\alpha} m_i v_{i,\alpha}^2(t)$$

Rescale velocities:

$$v_{i,\alpha}^{new} = \sqrt{\frac{T_{ref}}{T}} v_{i,\alpha}^{old}$$



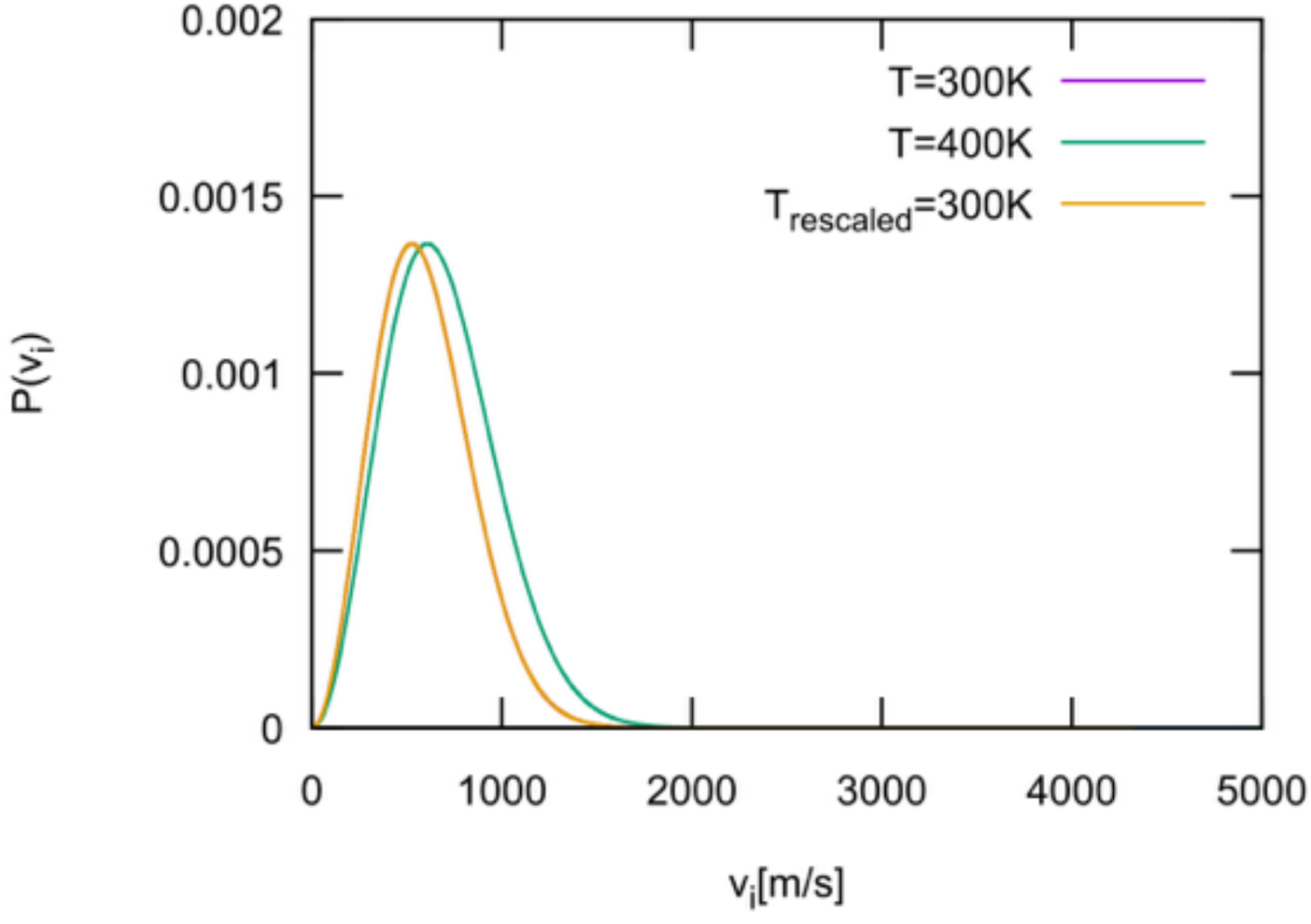
How to control the temperature?

Calculate instantaneous temperature:

$$k_B T(t) = \frac{1}{N_f} \sum_{i,\alpha} m_i v_{i,\alpha}^2(t)$$

Rescale velocities:

$$v_{i,\alpha}^{new} = \sqrt{\frac{T_{ref}}{T}} v_{i,\alpha}^{old}$$



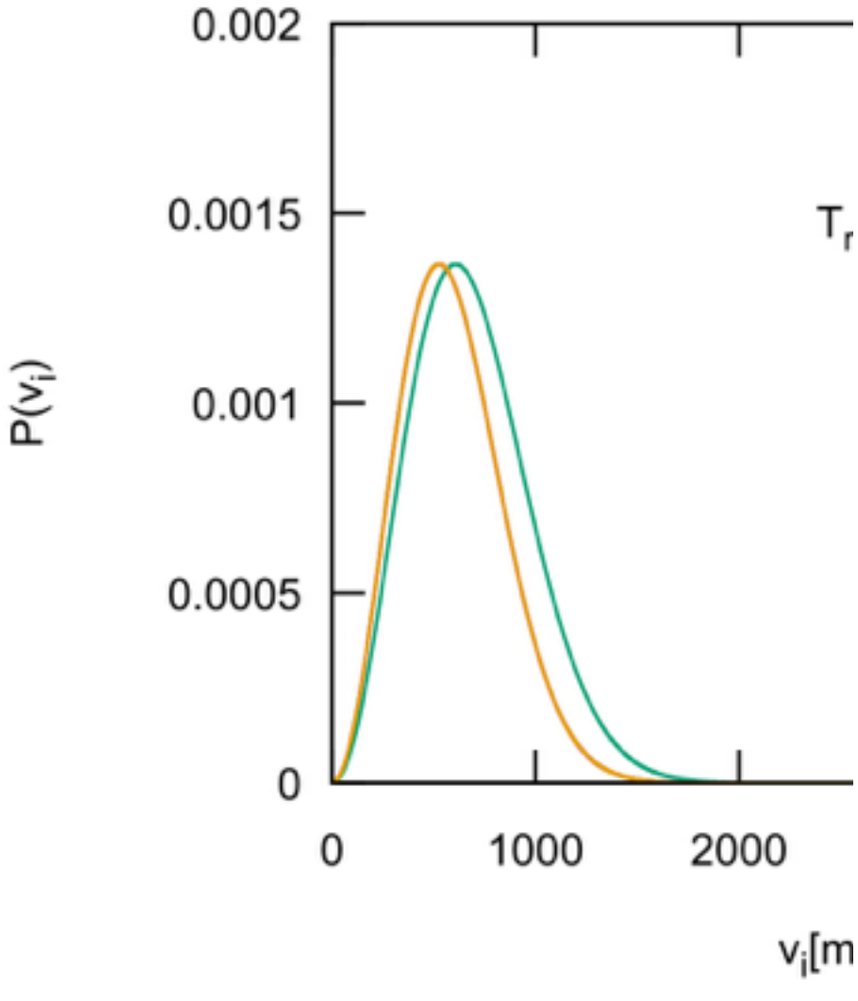
How to control the temperature?

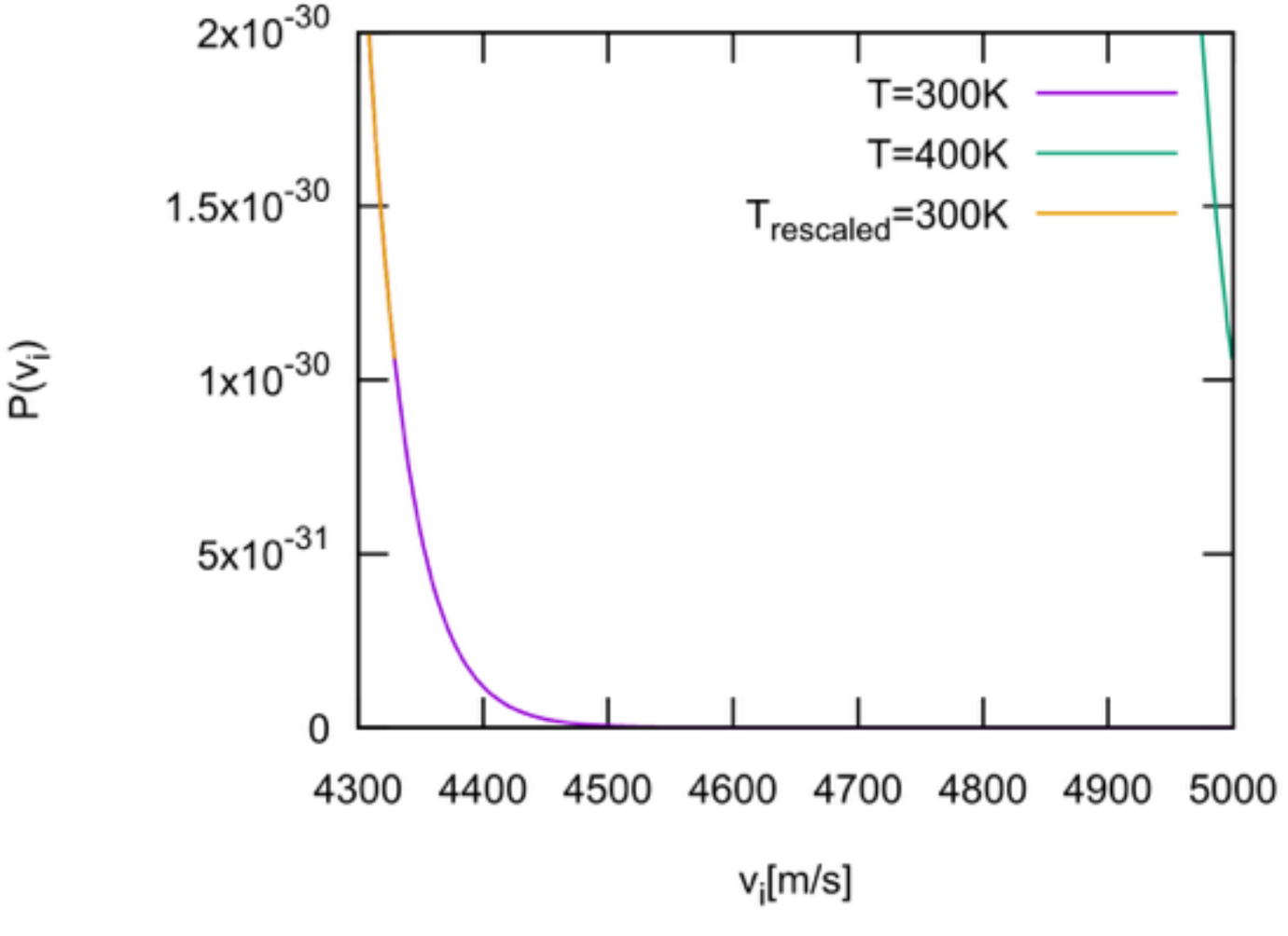
Calculate instantaneous temperature:

$$k_B T(t) = \frac{1}{N_f} \sum_{i,\alpha} m_i v_{i,\alpha}^2(t)$$

Rescale velocities:

$$v_{i,\alpha}^{new} = \sqrt{\frac{T_{ref}}{T}} v_{i,\alpha}^{old}$$





How to control the temperature?

Calculate instantaneous temperature:

$$k_B T(t) = \frac{1}{N_f} \sum_{i,\alpha} m_i v_{i,\alpha}^2(t)$$

Rescale velocities:

$$v_{i,\alpha}^{new} = \sqrt{\frac{T_{ref}}{T}} v_{i,\alpha}^{old}$$

Tail of velocity distribution is cut off!

The Flying Ice Cube: Velocity Rescaling in Molecular Dynamics Leads to Violation of Energy Equipartition

STEPHEN C. HARVEY,[1] ROBERT K.-Z. TAN,[1] THOMAS E. CHEATHAM III[2,*]

[1]*Department of Biochemistry and Molecular Genetics, University of Alabama at Birmingham, Birmingham, Alabama 35294*

[2]*Department of Pharmaceutical Chemistry, University of California, San Francisco, California 94143*

Received 31 July 1997; accepted 4 December 1997

How to control the temperature?

Calculate instantaneous temperature:

$$k_B T(t) = \frac{1}{N_f} \sum_{i,\alpha} m_i v_{i,\alpha}^2(t)$$

Rescale velocities:

$$v_{i,\alpha}^{new} = \sqrt{\frac{T_{ref}}{T}} v_{i,\alpha}^{old}$$

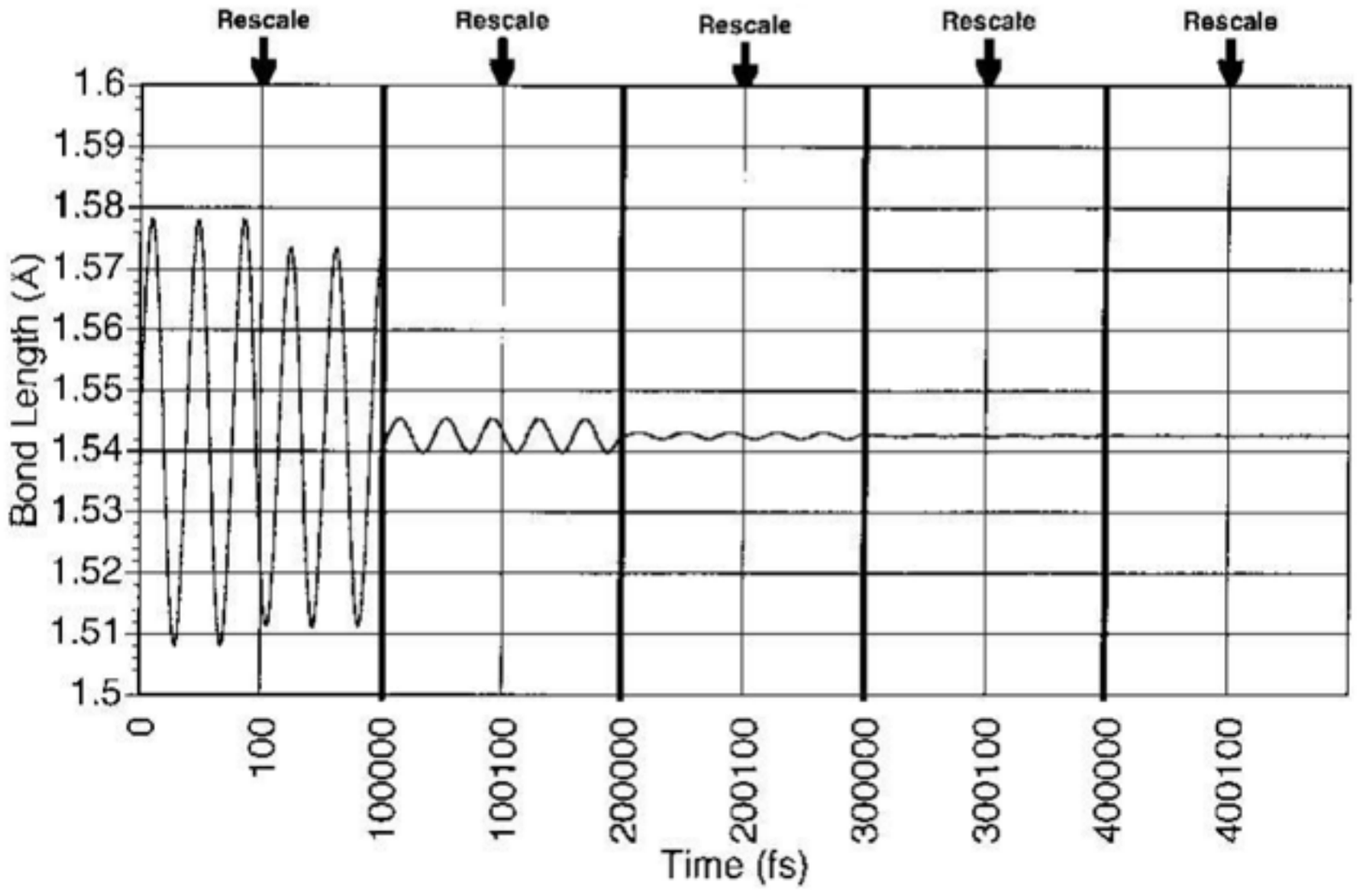
Tail of velocity dis

The Flying Ice Cube
Rescaling in Molecu
Leads to Violation o
Energy Equipartitio

STEPHEN C. HARVEY,[1] ROBERT
THOMAS E. CHEATHAM III[2,*]

[1]Department of Biochemistry and Molecular Genet
Birmingham, Alabama 35294
[2]Department of Pharmaceutical Chemistry, Unive

Received 31 July 1997; accepted 4 December 1997



Berendsen thermostat

Scale velocities by

$$c_t = \sqrt{1 - \frac{\Delta t}{\tau}\left(1 - \frac{T_{ref}}{T}\right)}$$

$$\tau \rightarrow \infty,\ c_t \rightarrow 1$$

$$\tau \rightarrow \Delta t,\ c_t \rightarrow \sqrt{\frac{T_{ref}}{T}}$$

Does not yield canonical ensemble
Controls temperature very reliably → is still used for equilibration (short)

Andersen thermostat

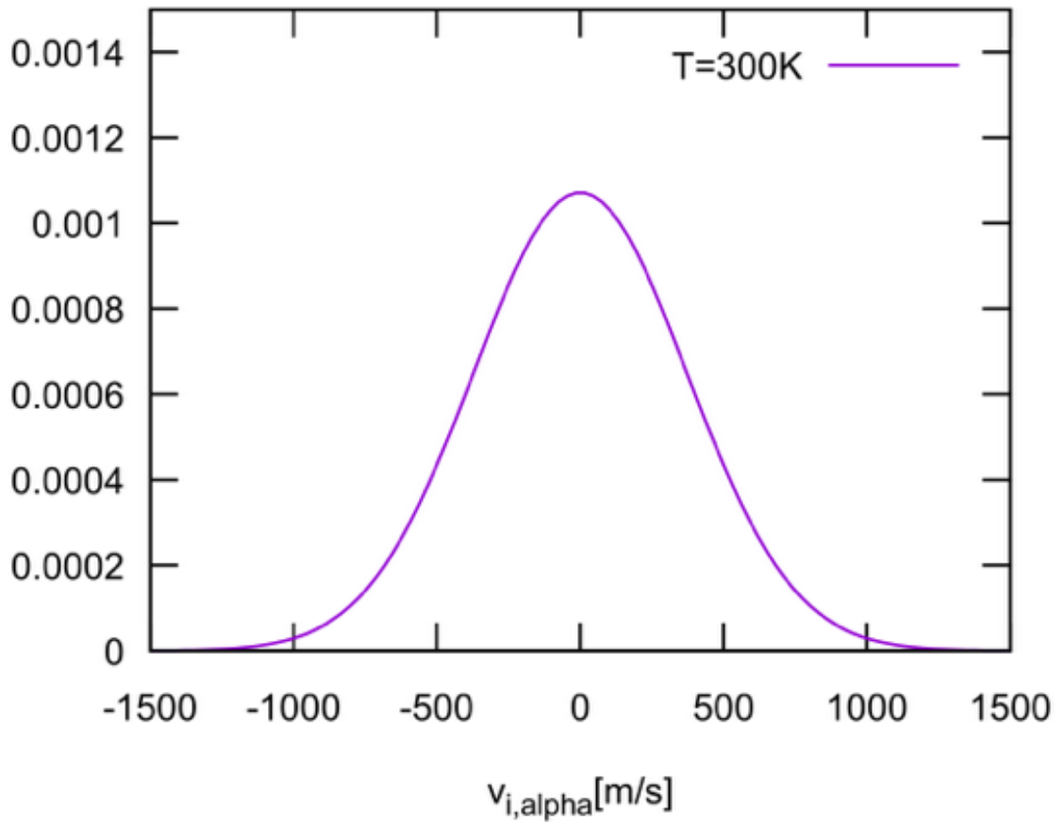
Perturbation with a stochastic term

1. Initialize system (positions and velocities)
2. Select n particles for collision with heat bath. Probability for a particle to be selected in Δt is $\gamma \Delta t$, γ = strength of coupling
3. For selected particle draw new velocity from Maxwell-Boltzmann distribution

$$P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_{i,\alpha}}{2 k_B T}\right)$$

Large γ → strong coupl. → dynamics are altered

Small γ → weak coupl. → poor T control



Langevin thermostat

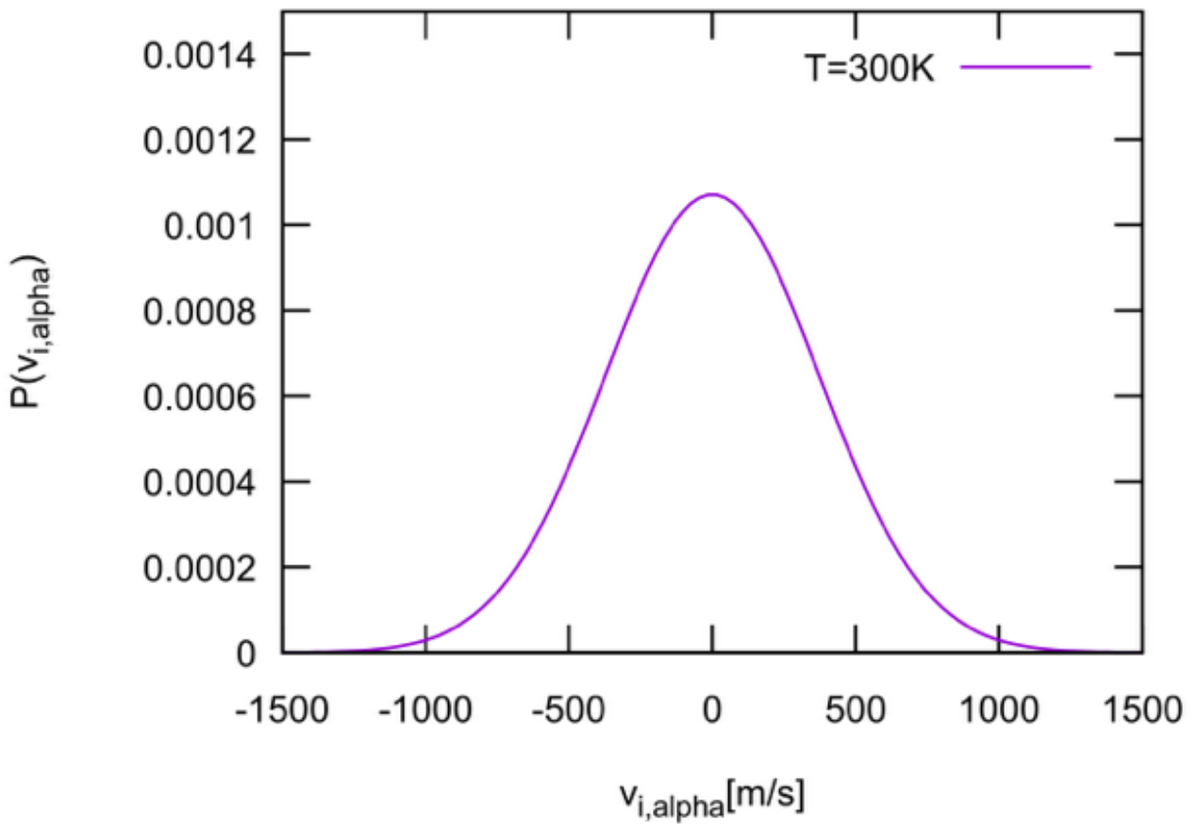
Perturb velocities with a Gaussian term

$$
\begin{aligned}
v_{new} &= c_1 v_{old} + c_2 N(0, \sqrt{k_B T/m}) \\
c_1^2 + c_2^2 &= 1 \\
c_1 &= exp(-\gamma \Delta t) \\
c_2 &= \sqrt{1 - c_1^2}
\end{aligned}
$$

$$
P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_{i,\alpha}}{2 k_B T}\right)
$$

$\gamma \rightarrow 0,\ c_1 \rightarrow 1$ no thermostat

$\gamma \rightarrow \infty,\ c_1 \rightarrow 0$ Andersen thermostat with strong coupling



Yields correct ensemble!

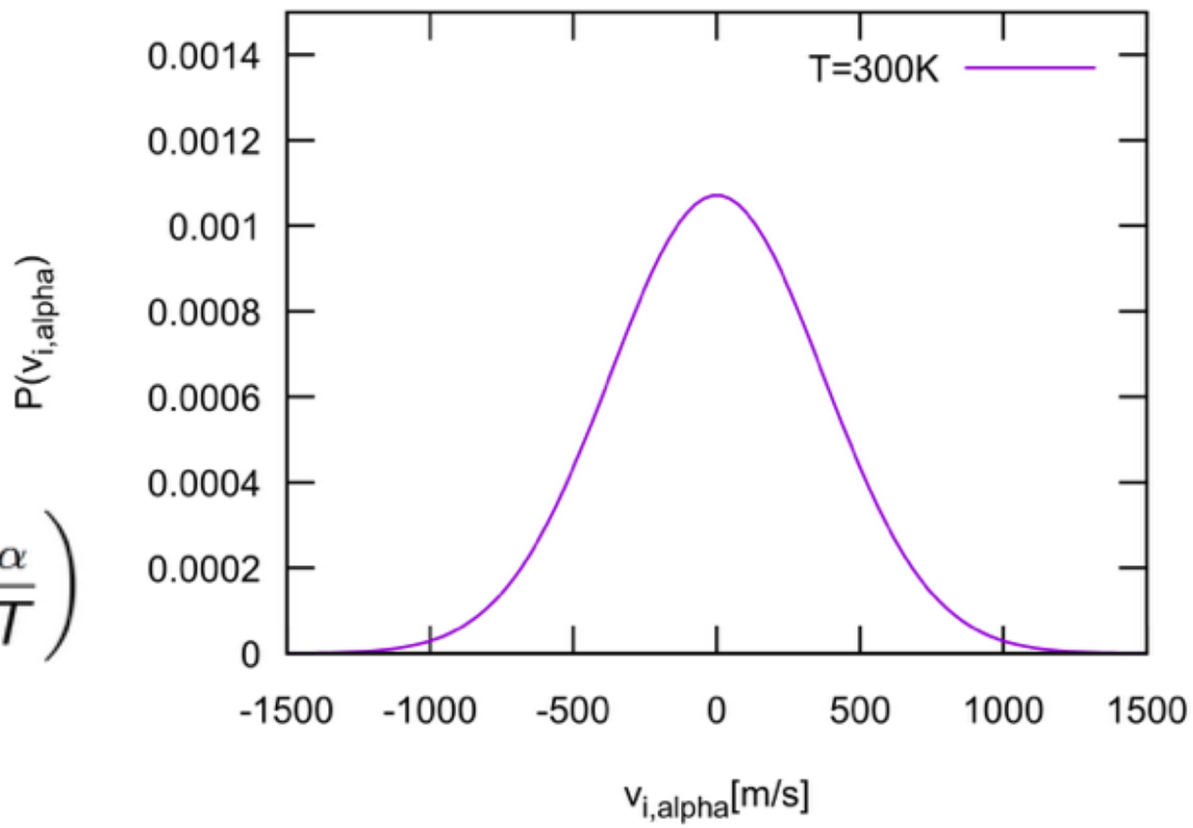
Langevin thermostat in simplemd

```
// Read initial positions
read_positions
// Set random velocities
randomize_velocities

// Compute initial neighbor list
 compute_list
// compute initial forces
compute_forces

for(istep=0;istep<nstep;istep++){
    thermostat(dt/2)
    velocity += (force*dt/2)/mass
    position += velocity*dt

    // Check whether the neighbour list has to be recomputed
    check_list

    if(recompute_list){
      compute_list
    }

    compute_forces

    velocity += (force*dt/2)/mass
    thermostat(dt/2)
}
```

Initial randomization of velocities

```
void randomize_velocities()
 {
// randomize the velocities according to the temperature
  for(int iatom=0;iatom<natoms;iatom++) for(int i=0;i<3;i++)
      velocities[iatom][i]=sqrt(temperature/masses[iatom])*random.Gaussian();
}
```

$$P(v_{i,\alpha}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_{i,\alpha}}{2 k_B T}\right)$$



Langevin thermostat

```
Thermostat() {

// Langevin thermostat, implemented as decribed in Bussi and Parrinello, Phys. Rev. E (2007)
// it is a linear combination of old velocities and new, randomly chosen, velocity,
// with proper coefficients

  double c1,c2;
  c1=exp(-friction*dt);

  for(int iatom=0;iatom<natoms;iatom++){
    c2=sqrt((1.0-c1*c1)*temperature/masses[iatom]);
    for(int i=0;i<3;i++){
      engint+=0.5*masses[iatom]*velocities[iatom][i]*velocities[iatom][i];
      velocities[iatom][i]=c1*velocities[iatom][i]+c2*random.Gaussian();
      engint-=0.5*masses[iatom]*velocities[iatom][i]*velocities[iatom][i];
    }
  }
}
```

K+V+engint is conserved (useful for checking the timestep)

Time average and ensemble average

$$A = \langle A(r(t), p(t)) \rangle_{time} = \frac{1}{N_{steps}} \sum_{t=0}^{N_{steps}} A(r(t), p(t))$$

$$A = \langle A(r, p) \rangle_{ensemble} = \sum_{t=0}^{N_{steps}} A(r, p) \rho_{ensemble}(r, p)$$

ρ = probability to find system in *(r, p)*

Ergodicity: $\langle A(r(t), p(t)) \rangle_{time} = \langle A(r, p) \rangle_{ensemble}$

In canonical ensemble

$$\rho(r,p) = \frac{1}{Z} e^{-E(r,p)/k_B T}$$

$$E(r,p) = K(p) + V(r)$$

$(r,p) \rightarrow$ microstate i, $Z = \sum_i e^{-E_i/k_B T}$ canonical partition function

$$\text{if } A = A(r) \rightarrow \langle A \rangle = \sum_i \rho(r_i) A(r_i)$$

$$\rho(r_i) = \frac{1}{Z_V} e^{-V(r_i)/k_B T}$$

$$Z_V = \sum_i e^{-V(r_i)/k_B T}$$

How to generate correct ensemble if we don't know Z?

Metropolis

Use detailed balance $$\rho(r_i)\pi(r_i \to r_j) = \rho(r_j)\pi(r_j \to r_i)$$

$$\pi(r_i \to r_j) = \alpha(r_i \to r_j)\mathrm{acc}(r_i \to r_j)$$

↑ trial probability (e.g. random)

$$\mathrm{acc}(r_i \to r_j) = \begin{cases} \rho(r_j)/\rho(r_i) & \text{if } \rho(r_j) < \rho(r_i) \\ 1 & \text{if } \rho(r_j) \geq \rho(r_i) \end{cases}$$

acceptance

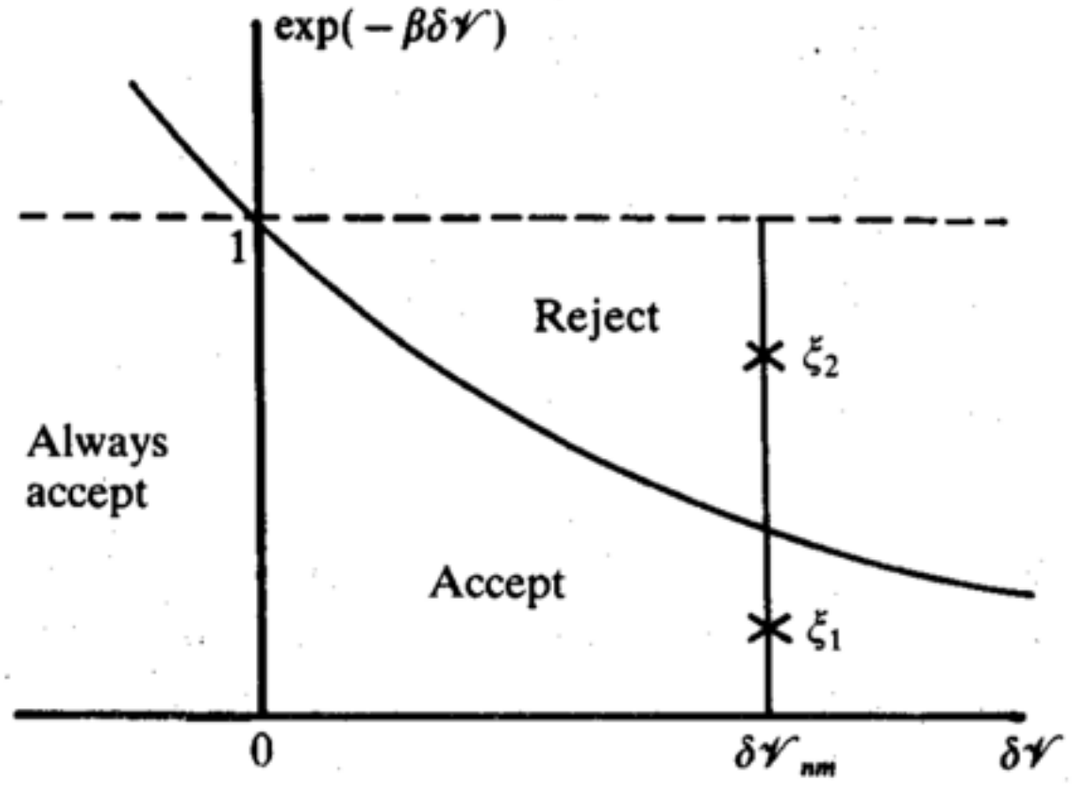
$$\frac{\rho(r_j)}{\rho(r_i)} = \frac{\pi(r_i \to r_j)}{\pi(r_j \to r_i)} = e^{-(V(r_j)-V(r_i))/k_B T}$$

no partition function

Metropolis, Rosenbluth, Rosenbluth, Teller and Teller. Equation of State Calculations by Fast Computing Machines. 1953

Monte Carlo - Metropolis algorithm

1. Start from initial condition $\mathbf{r}_1$
2. Calculate potential energy $V(\mathbf{r}_1)$
3. Propose a move $\mathbf{r}_1 \rightarrow \mathbf{r}_2$ (e.g. random)
4. Calculate $V(\mathbf{r}_2)$
5. Calculate $acc(\mathbf{r}_1 \rightarrow \mathbf{r}_2) = \exp(-(V(\mathbf{r}_2) - V(\mathbf{r}_1)/kBT)$
6. if $V(\mathbf{r}_2) \leq V(\mathbf{r}_1)$
   accept the move
   else
   extract a random number n in $[0,1]$
   if $n < acc(\mathbf{r}_1 \rightarrow \mathbf{r}_2)$
   accept the move
   else
   reject



Monte Carlo - Metropolis algorithm

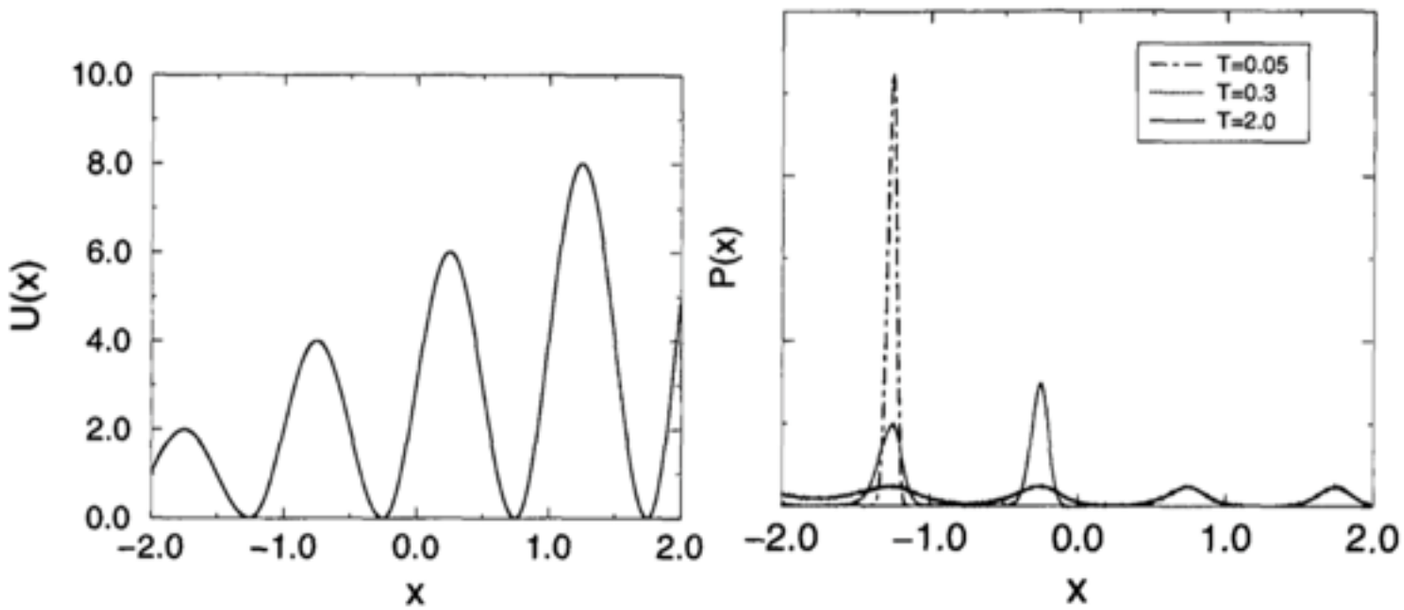
Possible to generate samples with desired distribution

Allows evaluation of average

$$\langle A \rangle = \sum_i \rho_{NVT}(r_i) A(r_i)$$

Parallel tempering

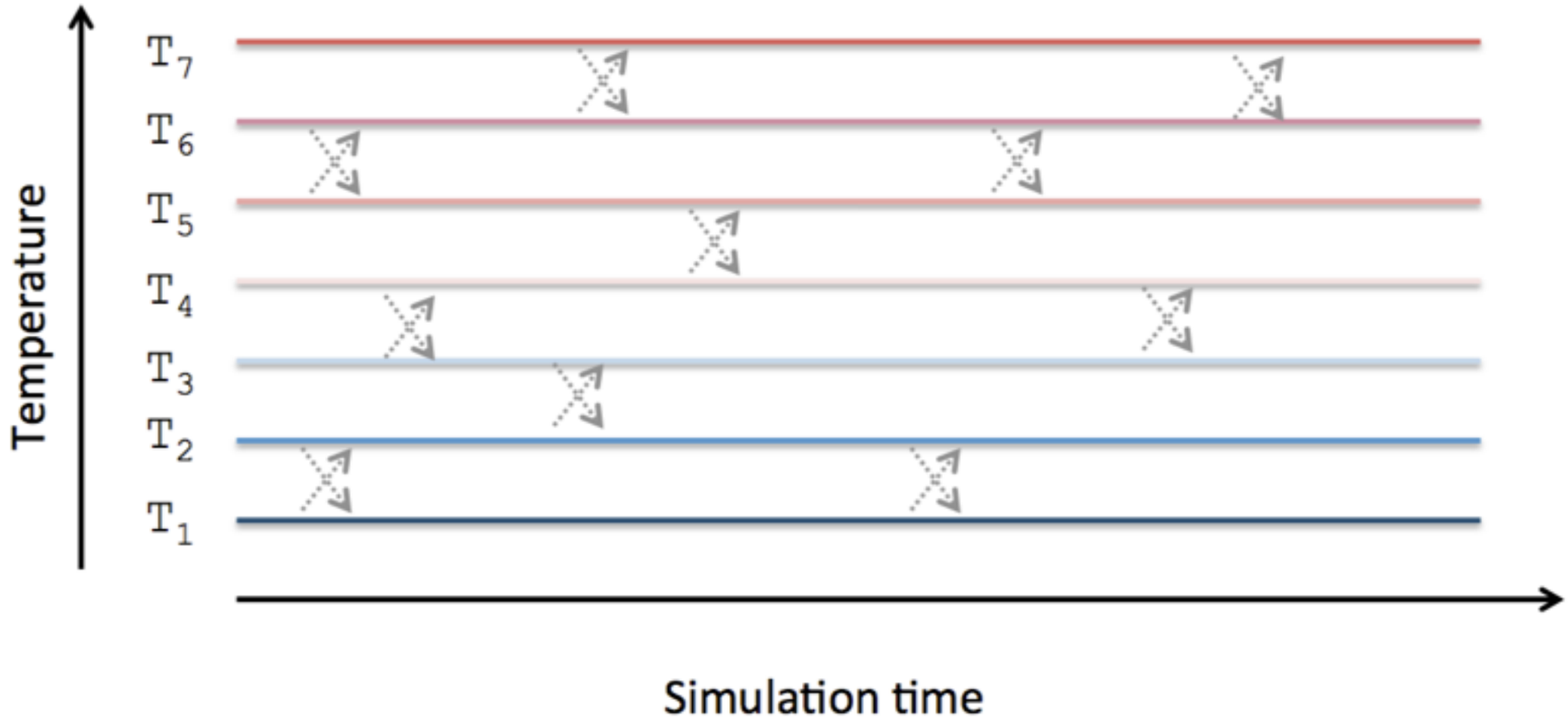
Sampling problem - stuck in a minimum



One-dimensional example potential

- If initially in one minimum, you will stay there
- at high temperatures you can overcome energy barriers

Parallel tempering

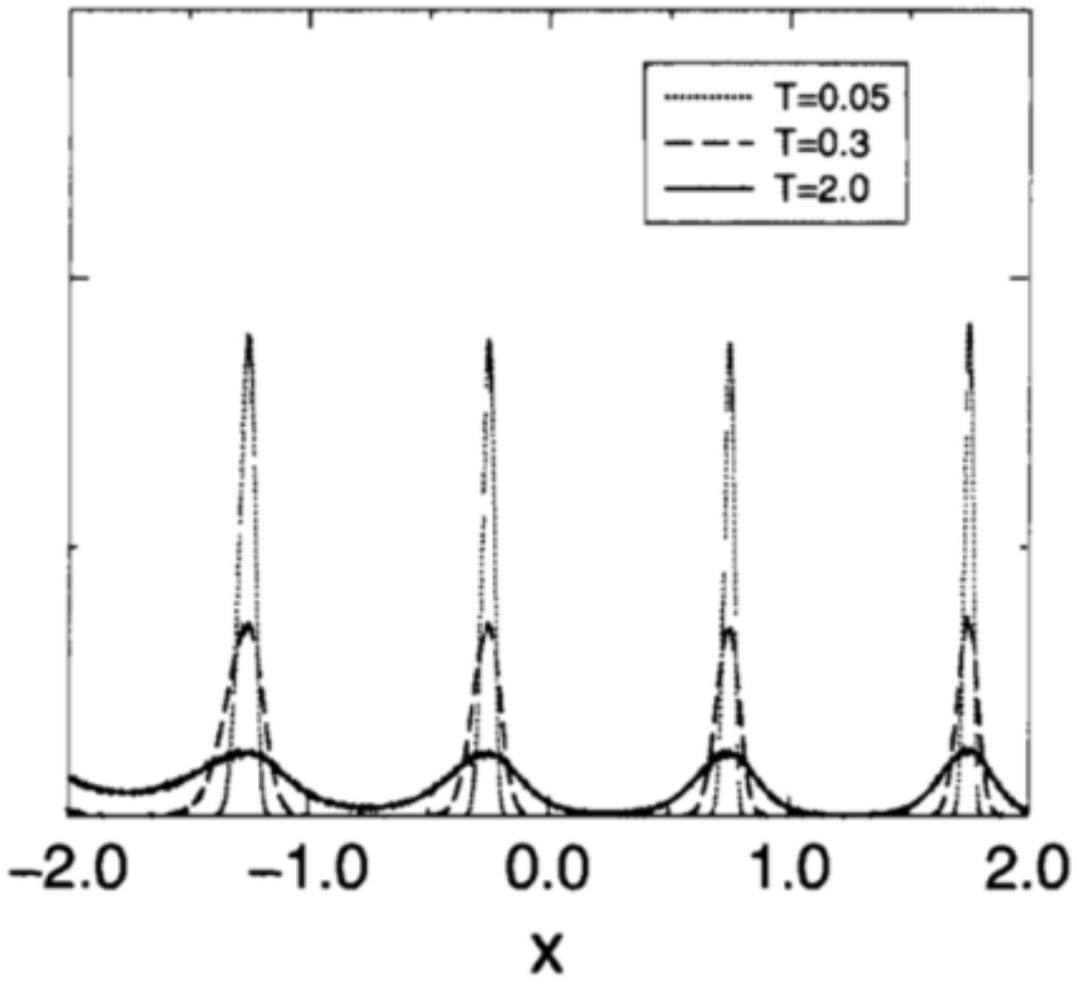


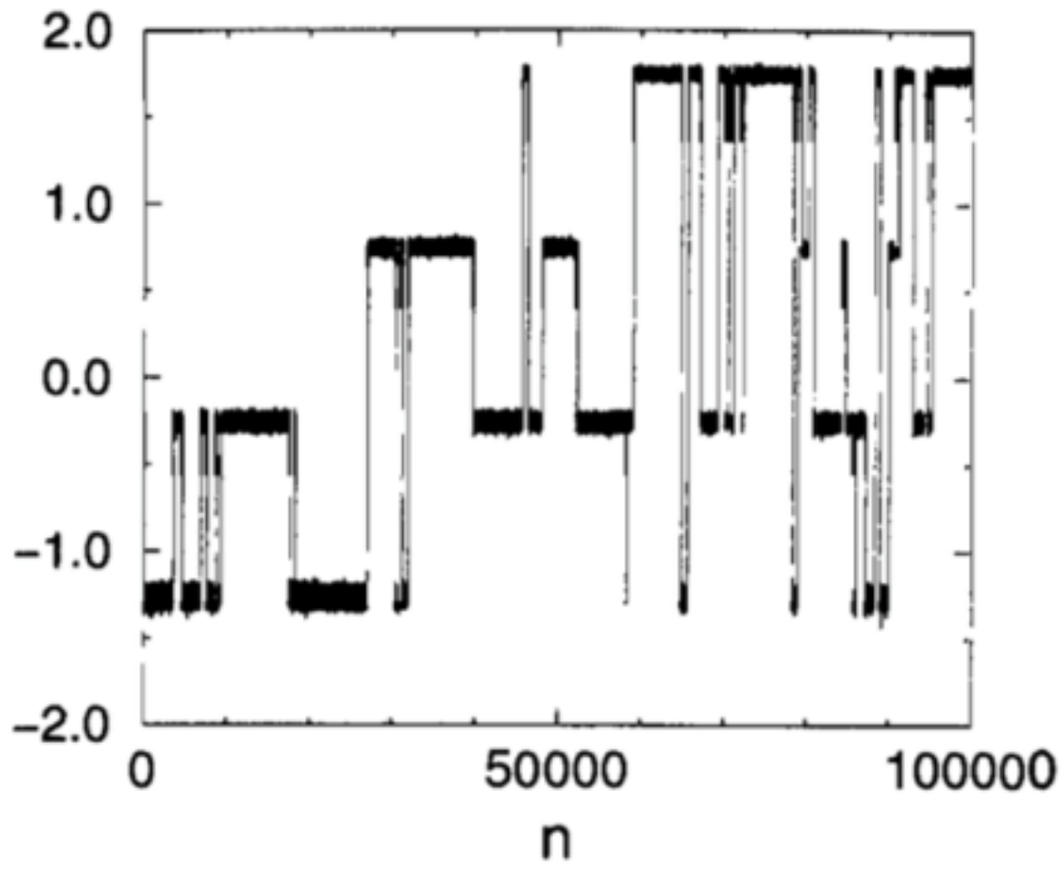
Idea: run parallel simulations at different temperatures.

Use a “swap” move to exchange between replicas

Accept/reject using the standard Metropolis criterion

Parallel tempering





One-dimensional example potential

Now the whole space is sampled, even by the replica at low T

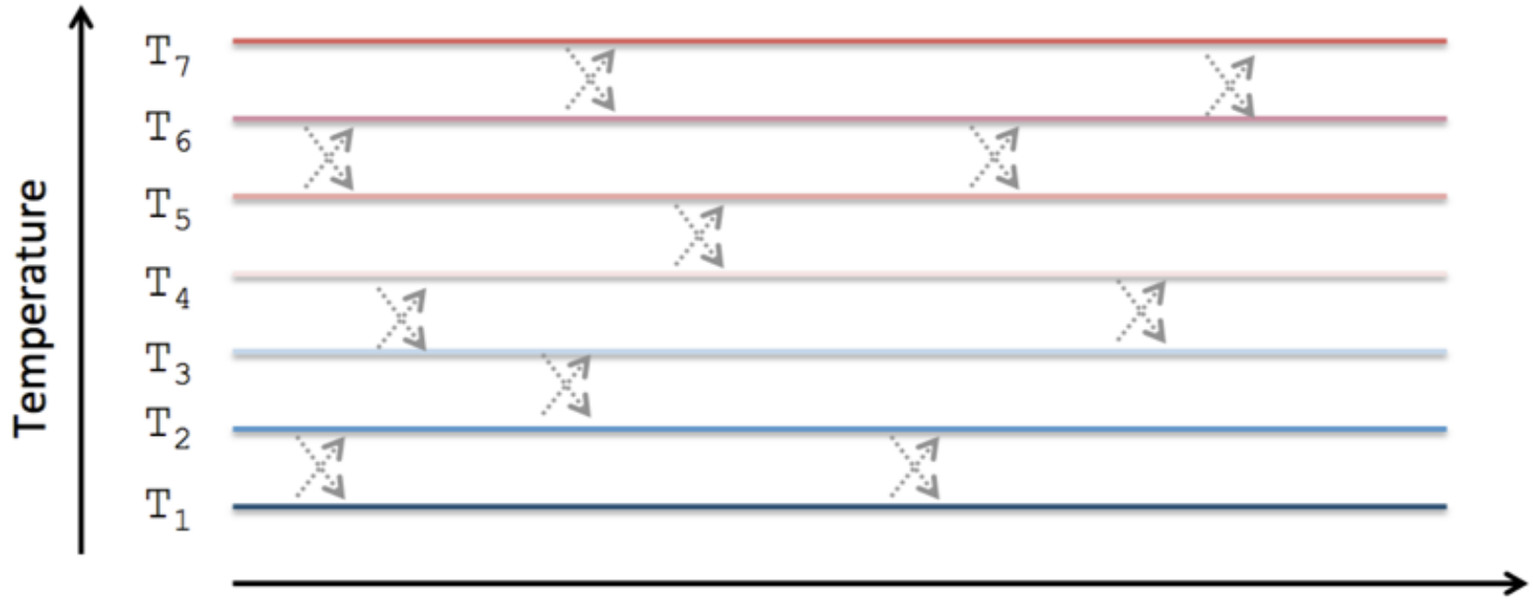
Parallel tempering

$$\mathrm{acc}(r_i \to r_j) = \begin{cases} \rho(r_i)/\rho(r_j) & \text{if } \rho(r_j) < \rho(r_i) \\ 1 & \text{if } \rho(r_j) \geq \rho(r_i) \end{cases}$$

acceptance

$$\mathrm{acc} = e^{-(1/k_B T_j - 1/k_B T_i)(V(r_i) - V(r_j))}$$

Parallel tempering



1. Create different replicas running at different temperatures
2. Every t steps attempt swap move between replica i and j
   calculate acc = $\exp((1/kBT_i - 1/kBT_j)(V(\mathbf{r}_i) - V(\mathbf{r}_j)))$
   if acc > 1
     swap the coordinates
   else
        if random.U01()<acc
            swap coordinates

ATT! If swap is accepted, rescale velocities $v_i(new) = sqrt(T_i/T_j)$; $v_j(new) = sqrt(T_j/T_i)$

Books on MD

Tamar Schlick: Molecular Modeling an Simulation: an Interdisciplinary guide (2010)

Allen & Tildesley: Computer Simulations of liquids (1991)