

# Simulating classical many body systems

Picture a system consisting of many “classical” objects, for instance in real space.

# Simulating classical many body systems

Picture a system consisting of many “classical” objects, for instance in real space.

- ▶ Assume particles are spherically symmetric

# Simulating classical many body systems

Picture a system consisting of many “classical” objects, for instance in real space.

- ▶ Assume particles are spherically symmetric
- ▶ Following our “normal” statistical physics description, at any point in time we can describe the system completely by  $\{\mathbf{r}^N, \mathbf{p}^N\}$

# Simulating classical many body systems

Picture a system consisting of many “classical” objects, for instance in real space.

- ▶ Assume particles are spherically symmetric
- ▶ Following our “normal” statistical physics description, at any point in time we can describe the system completely by  $\{\mathbf{r}^N, \mathbf{p}^N\}$
- ▶ The equations of motion that govern the time evolution of the system are simply Newton’s equations of motion:

$$\sum_{j=1}^N \mathbf{F}_{ij} = m_i \mathbf{a}_i \quad (1)$$

- ▶ e.g.  $3N$  equations in 3d

# Simulating classical many body systems

Picture a system consisting of many “classical” objects, for instance in real space.

- ▶ Assume particles are spherically symmetric
- ▶ Following our “normal” statistical physics description, at any point in time we can describe the system completely by  $\{\mathbf{r}^N, \mathbf{p}^N\}$
- ▶ The equations of motion that govern the time evolution of the system are simply Newton’s equations of motion:

$$\sum_{j=1}^N \mathbf{F}_{ij} = m_i \mathbf{a}_i \quad (1)$$

- ▶ e.g.  $3N$  equations in 3d

**Want:** To be able to predict and study (numerically) the phase behaviour of this system.

# Simulating classical many body systems

Two main routes:

- ▶ Simulations that “mimic” the dynamics of the system, e.g.
  - ▶ Molecular dynamics, Langevin dynamics, Brownian dynamics

# Simulating classical many body systems

Two main routes:

- ▶ Simulations that “mimic” the dynamics of the system, e.g.
  - ▶ Molecular dynamics, Langevin dynamics, Brownian dynamics
- ▶ Simulations that only care about the statistics, i.e. generate configurations according to a known distribution (e.g. Boltzmann), e.g.
  - ▶ Monte Carlo simulations

# Simulating classical many body systems

Two main routes:

- ▶ Simulations that “mimic” the dynamics of the system, e.g.
  - ▶ Molecular dynamics, Langevin dynamics, Brownian dynamics
- ▶ Simulations that only care about the statistics, i.e. generate configurations according to a known distribution (e.g. Boltzmann), e.g.
  - ▶ Monte Carlo simulations

## Dynamics:

- ▶ works roughly like a “real” experiment
- ▶ measures the “time” average:

$$\bar{A} = \frac{1}{\tau} \int_0^{\tau} A(t) dt$$



# Simulating classical many body systems

Two main routes:

- ▶ Simulations that “mimic” the dynamics of the system, e.g.
  - ▶ Molecular dynamics, Langevin dynamics, Brownian dynamics
- ▶ Simulations that only care about the statistics, i.e. generate configurations according to a known distribution (e.g. Boltzmann), e.g.
  - ▶ Monte Carlo simulations

## Dynamics:

- ▶ works roughly like a “real” experiment
- ▶ measures the “time” average:

$$\bar{A} = \frac{1}{\tau} \int_0^{\tau} A(t) dt$$

## Statistic simulations:

- ▶ based on equilibrium statistical physics
- ▶ measures the “ensemble” average:

$$\langle A \rangle = \frac{1}{M} \sum_M A_M$$

where  $A_M$  is measured from a selection of configurations according to the correct distribution.

# Molecular Dynamics: An intro

**Goal:** solve numerically Newton's equations of motion to determine  $\mathbf{r}_i(t)$

# Molecular Dynamics: An intro

**Goal:** solve numerically Newton's equations of motion to determine  $\mathbf{r}_i(t)$

Many algorithms have been developed to solve Newton's equations of motion, but they are not all equally good!

Things to think about:

# Molecular Dynamics: An intro

**Goal:** solve numerically Newton's equations of motion to determine  $\mathbf{r}_i(t)$

Many algorithms have been developed to solve Newton's equations of motion, but they are not all equally good!

Things to think about:

- ▶ **Algorithm speed:** Speed is not really important as calculating the interactions takes up most of the computational burden

# Molecular Dynamics: An intro

**Goal:** solve numerically Newton's equations of motion to determine  $\mathbf{r}_i(t)$

Many algorithms have been developed to solve Newton's equations of motion, but they are not all equally good!

Things to think about:

- ▶ **Algorithm speed:** Speed is not really important as calculating the interactions takes up most of the computational burden
- ▶ **Time steps:** The larger the time step, the fewer the number of force evaluations, the faster the code: ideally want an algorithm that is accurate for large time steps.

# Molecular Dynamics: An intro

**Goal:** solve numerically Newton's equations of motion to determine  $\mathbf{r}_i(t)$

Many algorithms have been developed to solve Newton's equations of motion, but they are not all equally good!

Things to think about:

- ▶ **Algorithm speed:** Speed is not really important as calculating the interactions takes up most of the computational burden
- ▶ **Time steps:** The larger the time step, the fewer the number of force evaluations, the faster the code: ideally want an algorithm that is accurate for large time steps.
- ▶ **Time reversibility:** Original equations are time reversible, hence ideally so should the integration algorithm be

# Molecular Dynamics: An intro

**Goal:** solve numerically Newton's equations of motion to determine  $\mathbf{r}_i(t)$

Many algorithms have been developed to solve Newton's equations of motion, but they are not all equally good!

Things to think about:

- ▶ **Algorithm speed:** Speed is not really important as calculating the interactions takes up most of the computational burden
- ▶ **Time steps:** The larger the time step, the fewer the number of force evaluations, the faster the code: ideally want an algorithm that is accurate for large time steps.
- ▶ **Time reversibility:** Original equations are time reversible, hence ideally so should the integration algorithm be
- ▶ **Accuracy:** Energy should be conserved.

# Molecular Dynamics: An intro

**Goal:** solve numerically Newton's equations of motion to determine  $\mathbf{r}_i(t)$

Many algorithms have been developed to solve Newton's equations of motion, but they are not all equally good!

Things to think about:

- ▶ **Algorithm speed:** Speed is not really important as calculating the interactions takes up most of the computational burden
- ▶ **Time steps:** The larger the time step, the fewer the number of force evaluations, the faster the code: ideally want an algorithm that is accurate for large time steps.
- ▶ **Time reversibility:** Original equations are time reversible, hence ideally so should the integration algorithm be
- ▶ **Accuracy:** Energy should be conserved.

Perhaps most common compromise: **Verlet algorithm**



# Molecular Dynamics: Verlet Algorithm

Taylor expand  $\mathbf{r}(t)$  around  $t$ :

$$r(t + \Delta t) = r(t) + \dot{r}(t)\Delta t + \frac{\ddot{r}(t)}{2}\Delta t^2 + \frac{\dddot{r}(t)}{3!}\Delta t^3 + \dots$$

# Molecular Dynamics: Verlet Algorithm

Taylor expand  $\mathbf{r}(t)$  around  $t$ :

$$r(t + \Delta t) = r(t) + \dot{r}(t)\Delta t + \frac{\ddot{r}(t)}{2}\Delta t^2 + \frac{\dddot{r}(t)}{3!}\Delta t^3 + \dots$$

- ▶  $\dot{r}(t) = v(t)$  is the velocity

# Molecular Dynamics: Verlet Algorithm

Taylor expand  $\mathbf{r}(t)$  around  $t$ :

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \dot{\mathbf{r}}(t)\Delta t + \frac{\ddot{\mathbf{r}}(t)}{2}\Delta t^2 + \frac{\dddot{\mathbf{r}}(t)}{3!}\Delta t^3 + \dots$$

- ▶  $\dot{\mathbf{r}}(t) = \mathbf{v}(t)$  is the velocity
- ▶  $\ddot{\mathbf{r}}(t) = \frac{\mathbf{f}(t)}{m}$  is the acceleration and hence proportional to the force

# Molecular Dynamics: Verlet Algorithm

Taylor expand  $\mathbf{r}(t)$  around  $t$ :

$$r(t + \Delta t) = r(t) + \dot{r}(t)\Delta t + \frac{\ddot{r}(t)}{2}\Delta t^2 + \frac{\dddot{r}(t)}{3!}\Delta t^3 + \dots$$

- ▶  $\dot{r}(t) = v(t)$  is the velocity
- ▶  $\ddot{r}(t) = \frac{f(t)}{m}$  is the acceleration and hence proportional to the force

Similarly

$$r(t - \Delta t) = r(t) - \dot{r}(t)\Delta t + \frac{\ddot{r}(t)}{2}\Delta t^2 - \frac{\dddot{r}(t)}{3!}\Delta t^3 + \dots$$

# Molecular Dynamics: Verlet Algorithm

Taylor expand  $\mathbf{r}(t)$  around  $t$ :

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \dot{\mathbf{r}}(t)\Delta t + \frac{\ddot{\mathbf{r}}(t)}{2}\Delta t^2 + \frac{\dddot{\mathbf{r}}(t)}{3!}\Delta t^3 + \dots$$

- ▶  $\dot{\mathbf{r}}(t) = \mathbf{v}(t)$  is the velocity
- ▶  $\ddot{\mathbf{r}}(t) = \frac{\mathbf{f}(t)}{m}$  is the acceleration and hence proportional to the force

Similarly

$$\mathbf{r}(t - \Delta t) = \mathbf{r}(t) - \dot{\mathbf{r}}(t)\Delta t + \frac{\ddot{\mathbf{r}}(t)}{2}\Delta t^2 - \frac{\dddot{\mathbf{r}}(t)}{3!}\Delta t^3 + \dots$$

Hence

$$\mathbf{r}(t + \Delta t) + \mathbf{r}(t - \Delta t) = 2\mathbf{r}(t) + \ddot{\mathbf{r}}(t)\Delta t^2 + \mathcal{O}(\Delta t^4)$$

# Molecular Dynamics: Verlet Algorithm

Taylor expand  $\mathbf{r}(t)$  around  $t$ :

$$r(t + \Delta t) = r(t) + \dot{r}(t)\Delta t + \frac{\ddot{r}(t)}{2}\Delta t^2 + \frac{\dddot{r}(t)}{3!}\Delta t^3 + \dots$$

- ▶  $\dot{r}(t) = v(t)$  is the velocity
- ▶  $\ddot{r}(t) = \frac{f(t)}{m}$  is the acceleration and hence proportional to the force

Similarly

$$r(t - \Delta t) = r(t) - \dot{r}(t)\Delta t + \frac{\ddot{r}(t)}{2}\Delta t^2 - \frac{\dddot{r}(t)}{3!}\Delta t^3 + \dots$$

Hence

$$r(t + \Delta t) + r(t - \Delta t) = 2r(t) + \ddot{r}(t)\Delta t^2 + \mathcal{O}(\Delta t^4)$$

$$r(t + \Delta t) \approx 2r(t) - r(t - \Delta t) + \frac{f(t)}{m}\Delta t^2 \quad (2)$$

# Molecular Dynamics: Verlet Algorithm

How does this algorithm stack up?

# Molecular Dynamics: Verlet Algorithm

How does this algorithm stack up?

- ▶ **Fast:** only  $r(t)$ ,  $f(t)$ , and a stored  $r(t - \Delta t)$  necessary (i.e. no velocities come into play!)



# Molecular Dynamics: Verlet Algorithm

How does this algorithm stack up?

- ▶ **Fast:** only  $r(t)$ ,  $f(t)$ , and a stored  $r(t - \Delta t)$  necessary (i.e. no velocities come into play!)
- ▶ **Time steps:** not particularly great for large time steps (only order  $\Delta t^4$ , and generally higher order should be able to have larger time steps - however, higher order usually have problems with time reversibility and long-time energy drifts)

# Molecular Dynamics: Verlet Algorithm

How does this algorithm stack up?

- ▶ **Fast:** only  $r(t)$ ,  $f(t)$ , and a stored  $r(t - \Delta t)$  necessary (i.e. no velocities come into play!)
- ▶ **Time steps:** not particularly great for large time steps (only order  $\Delta t^4$ , and generally higher order should be able to have larger time steps - however, higher order usually have problems with time reversibility and long-time energy drifts)
- ▶ **Time reversibility:** yes

# Molecular Dynamics: Verlet Algorithm

How does this algorithm stack up?

- ▶ **Fast:** only  $r(t)$ ,  $f(t)$ , and a stored  $r(t - \Delta t)$  necessary (i.e. no velocities come into play!)
- ▶ **Time steps:** not particularly great for large time steps (only order  $\Delta t^4$ , and generally higher order should be able to have larger time steps - however, higher order usually have problems with time reversibility and long-time energy drifts)
- ▶ **Time reversibility:** yes
- ▶ **Accuracy:** short term energy conservation fair and importantly, little long term energy drift

# Molecular Dynamics at constant temperature

MD as we just derived it is in a microcanonical ensemble (NVE).

# Molecular Dynamics at constant temperature

MD as we just derived it is in a microcanonical ensemble (NVE).

- ▶ NVE not necessarily the most useful - working at constant temperature is often much more useful than energy

# Molecular Dynamics at constant temperature

MD as we just derived it is in a microcanonical ensemble (NVE).

- ▶ NVE not necessarily the most useful - working at constant temperature is often much more useful than energy
- ▶ To change to an NVT ensemble we need to introduce a thermostat

# Molecular Dynamics at constant temperature

MD as we just derived it is in a microcanonical ensemble (NVE).

- ▶ NVE not necessarily the most useful - working at constant temperature is often much more useful than energy
- ▶ To change to an NVT ensemble we need to introduce a thermostat

- ▶ perhaps the most simple thermostat



# Andersen Thermostat

- ▶ perhaps the most simple thermostat
- ▶ **Idea:** couple system to a heat bath that imposes the desired temperature

- ▶ perhaps the most simple thermostat
- ▶ **Idea:** couple system to a heat bath that imposes the desired temperature
- ▶ **Method:** introduce stochastic impulse forces that act occasionally on randomly selected particles. Between stochastic events, system evolves at constant energy.

- ▶ perhaps the most simple thermostat
- ▶ **Idea:** couple system to a heat bath that imposes the desired temperature
- ▶ **Method:** introduce stochastic impulse forces that act occasionally on randomly selected particles. Between stochastic events, system evolves at constant energy.
- ▶ **In practice:**
  - ▶ each timestep, select a number of particles to undergo a collision with the heat bath
  - ▶ the selected particles receive a new velocity drawn from a Maxwell-Boltzmann distribution at the desired temperature
  - ▶ if using the Verlet algorithm, then the position of the particle at time  $t - \Delta t$  must be recalculated

- ▶ perhaps the most simple thermostat
- ▶ **Idea:** couple system to a heat bath that imposes the desired temperature
- ▶ **Method:** introduce stochastic impulse forces that act occasionally on randomly selected particles. Between stochastic events, system evolves at constant energy.
- ▶ **In practice:**
  - ▶ each timestep, select a number of particles to undergo a collision with the heat bath
  - ▶ the selected particles receive a new velocity drawn from a Maxwell-Boltzmann distribution at the desired temperature
  - ▶ if using the Verlet algorithm, then the position of the particle at time  $t - \Delta t$  must be recalculated
- ▶ the coupling strength is determined by the frequency of stochastic collisions per particle

- ▶ perhaps the most simple thermostat
- ▶ **Idea:** couple system to a heat bath that imposes the desired temperature
- ▶ **Method:** introduce stochastic impulse forces that act occasionally on randomly selected particles. Between stochastic events, system evolves at constant energy.
- ▶ **In practice:**
  - ▶ each timestep, select a number of particles to undergo a collision with the heat bath
  - ▶ the selected particles receive a new velocity drawn from a Maxwell-Boltzmann distribution at the desired temperature
  - ▶ if using the Verlet algorithm, then the position of the particle at time  $t - \Delta t$  must be recalculated
- ▶ the coupling strength is determined by the frequency of stochastic collisions per particle
- ▶ **Comment:** This makes the dynamics slightly unphysical.

**Question:** Can we design a thermostat that is not stochastic in nature?

**Question:** Can we design a thermostat that is not stochastic in nature?

→ **Nosé Hoover Thermostat**

**Question:** Can we design a thermostat that is not stochastic in nature?

→ **Nosé Hoover Thermostat**

**Idea:** Add an extra variable to the Lagrangian ( $s$ ) in order to build molecular dynamics simulations subject to a constant-temperature constraint.



**Question:** Can we design a thermostat that is not stochastic in nature?

→ **Nosé Hoover Thermostat**

**Idea:** Add an extra variable to the Lagrangian ( $s$ ) in order to build molecular dynamics simulations subject to a constant-temperature constraint.

**Nosé Extended Lagrangian:**

$$\mathcal{L}_{NOSE} = \sum_{i=1}^N \frac{m_i}{2} \dot{\mathbf{r}}_i^2 - U(\mathbf{r}^N) + \frac{Q}{2} \dot{s}^2 - \frac{L}{\beta} \ln s$$

where

- ▶  $Q$ : an “effective” mass associated with  $s$
- ▶  $L$ : a parameter which will be fixed during the derivation

## Molecular Dynamics in an NVT ensemble

$$\mathcal{L}_{NOSE} = \sum_{i=1}^N \frac{m_i}{2} s^2 \dot{\mathbf{r}}_i^2 - U(\mathbf{r}^N) + \frac{Q}{2} \dot{s}^2 - \frac{L}{\beta} \ln s$$

**Associated momenta:**

$$\mathbf{p}_i \equiv \frac{\partial \mathcal{L}_{NOSE}}{\partial \dot{\mathbf{r}}_i} = m_i s^2 \dot{\mathbf{r}}_i$$

$$p_s \equiv \frac{\partial \mathcal{L}_{NOSE}}{\partial \dot{s}} = Q \dot{s}$$

## Molecular Dynamics in an NVT ensemble

$$\mathcal{L}_{NOSE} = \sum_{i=1}^N \frac{m_i}{2} s^2 \dot{\mathbf{r}}_i^2 - U(\mathbf{r}^N) + \frac{Q}{2} \dot{s}^2 - \frac{L}{\beta} \ln s$$

**Associated momenta:**

$$\mathbf{p}_i \equiv \frac{\partial \mathcal{L}_{NOSE}}{\partial \dot{\mathbf{r}}_i} = m_i s^2 \dot{\mathbf{r}}_i$$

$$p_s \equiv \frac{\partial \mathcal{L}_{NOSE}}{\partial \dot{s}} = Q \dot{s}$$

**Associated Hamiltonian** (Recall  $\mathcal{H} = \sum_i p_i q_i - \mathcal{L}$ ):

$$\mathcal{H}_{NOSE} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i s^2} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + \frac{L}{\beta} \ln s$$

## Molecular Dynamics in an NVT ensemble

$$\mathcal{L}_{NOSE} = \sum_{i=1}^N \frac{m_i}{2} s^2 \dot{\mathbf{r}}_i^2 - U(\mathbf{r}^N) + \frac{Q}{2} \dot{s}^2 - \frac{L}{\beta} \ln s$$

**Associated momenta:**

$$\mathbf{p}_i \equiv \frac{\partial \mathcal{L}_{NOSE}}{\partial \dot{\mathbf{r}}_i} = m_i s^2 \dot{\mathbf{r}}_i$$

$$p_s \equiv \frac{\partial \mathcal{L}_{NOSE}}{\partial \dot{s}} = Q \dot{s}$$

**Associated Hamiltonian** (Recall  $\mathcal{H} = \sum_i p_i q_i - \mathcal{L}$ ):

$$\mathcal{H}_{NOSE} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i s^2} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + \frac{L}{\beta} \ln s$$

**Corresponding partition function**

$$Q_{NOSE}(E, V, N) = \frac{c}{N!} \int d p_s d s d \mathbf{p}^N d \mathbf{r}^N \delta(E - \mathcal{H}_{NOSE})$$

with  $c$  a constant.

## Molecular Dynamics in an NVT ensemble

Recall that if  $f(x)$  is a function with roots at  $x_i$ , then

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

# Molecular Dynamics in an NVT ensemble

Recall that if  $f(x)$  is a function with roots at  $x_i$ , then

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

Let

$$\mathbf{p}' = \frac{\mathbf{p}}{s} \qquad \mathcal{H}(\mathbf{p}', \mathbf{r}) = \sum_{i=1}^N \frac{\mathbf{p}_i'^2}{2m_i} + U(\mathbf{r}^N)$$

## Molecular Dynamics in an NVT ensemble

Recall that if  $f(x)$  is a function with roots at  $x_i$ , then

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

Let

$$\mathbf{p}' = \frac{\mathbf{p}}{s} \quad \mathcal{H}(\mathbf{p}', \mathbf{r}) = \sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{2m_i} + U(\mathbf{r}^N)$$

Plugging this in, we obtain

$$\begin{aligned} Q_{NOSE} &= \frac{c}{N!} \int d\mathbf{p}_s ds d\mathbf{p}^N d\mathbf{r}^N \delta(E - \mathcal{H}_{NOSE}) \\ &= \frac{c}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N d\mathbf{p}_s ds \, s^{3N} \delta\left(\sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{2m_i} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + \frac{L}{\beta} \ln s - E\right) \end{aligned}$$

## Molecular Dynamics in an NVT ensemble

Recall that if  $f(x)$  is a function with roots at  $x_i$ , then

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

Let

$$\mathbf{p}' = \frac{\mathbf{p}}{s} \quad \mathcal{H}(\mathbf{p}', \mathbf{r}) = \sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{2m_i} + U(\mathbf{r}^N)$$

Plugging this in, we obtain

$$\begin{aligned} Q_{NOSE} &= \frac{c}{N!} \int d\mathbf{p}_s ds d\mathbf{p}^N d\mathbf{r}^N \delta(E - \mathcal{H}_{NOSE}) \\ &= \frac{c}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N d\mathbf{p}_s ds \, s^{3N} \delta\left(\sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{2m_i} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + \frac{L}{\beta} \ln s - E\right) \\ &= \frac{c}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N d\mathbf{p}_s ds \, \frac{\beta s^{3N+1}}{L} \delta\left(s - \exp\left(-\beta \frac{\mathcal{H}(\mathbf{p}', \mathbf{r}) + p_s^2/(2Q) - E}{L}\right)\right) \end{aligned}$$



## Molecular Dynamics in an NVT ensemble

Recall that if  $f(x)$  is a function with roots at  $x_i$ , then

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

Let

$$\mathbf{p}' = \frac{\mathbf{p}}{s} \quad \mathcal{H}(\mathbf{p}', \mathbf{r}) = \sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{2m_i} + U(\mathbf{r}^N)$$

Plugging this in, we obtain

$$\begin{aligned} Q_{NOSE} &= \frac{c}{N!} \int d\mathbf{p}_s ds d\mathbf{p}^N d\mathbf{r}^N \delta(E - \mathcal{H}_{NOSE}) \\ &= \frac{c}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N d\mathbf{p}_s ds \, s^{3N} \delta\left(\sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{2m_i} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + \frac{L}{\beta} \ln s - E\right) \\ &= \frac{c}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N d\mathbf{p}_s ds \, \frac{\beta s^{3N+1}}{L} \delta\left(s - \exp\left(-\beta \frac{\mathcal{H}(\mathbf{p}', \mathbf{r}) + p_s^2/(2Q) - E}{L}\right)\right) \\ &= \frac{C}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N e^{-\beta \frac{3N+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r})} \end{aligned}$$

## Molecular Dynamics in an NVT ensemble

$$Q_{NOSE} = \frac{C}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N e^{-\beta \frac{3N+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r})}$$

## Molecular Dynamics in an NVT ensemble

$$Q_{NOSE} = \frac{C}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N e^{-\beta \frac{3N+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r})}$$

Associated average value of a measurable  $A$

$$\langle A(\mathbf{p}', \mathbf{r}) \rangle = \frac{\int d\mathbf{p}'^N d\mathbf{r}^N A(\mathbf{p}', \mathbf{r}) e^{-\beta \frac{3N+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r})}}{\int d\mathbf{p}'^N d\mathbf{r}^N e^{-\beta \frac{3N+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r})}}$$

# Molecular Dynamics in an NVT ensemble

$$Q_{NOSE} = \frac{C}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N e^{-\beta \frac{3N+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r})}$$

Associated average value of a measurable  $A$

$$\langle A(\mathbf{p}', \mathbf{r}) \rangle = \frac{\int d\mathbf{p}'^N d\mathbf{r}^N A(\mathbf{p}', \mathbf{r}) e^{-\beta \frac{3N+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r})}}{\int d\mathbf{p}'^N d\mathbf{r}^N e^{-\beta \frac{3N+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r})}}$$

We now choose  $L = 3N + 1$ :

$$\langle A(\mathbf{p}', \mathbf{r}) \rangle = \langle A(\mathbf{p}', \mathbf{r}) \rangle_{\text{canonical}} \quad \text{with Hamiltonian } \mathcal{H}(\mathbf{p}', \mathbf{r})$$

# Molecular Dynamics in an NVT ensemble

We now have “real” and “virtual” variables

real	virtual
$\mathbf{r}'$	$\mathbf{r}$
$\mathbf{p}'$	$\mathbf{p}/s$
$s'$	$s$
$\Delta t'$	$\Delta t/s$

# Molecular Dynamics in an NVT ensemble

We now have “real” and “virtual” variables

real	virtual
$\mathbf{r}'$	$\mathbf{r}$
$\mathbf{p}'$	$\mathbf{p}/s$
$s'$	$s$
$\Delta t'$	$\Delta t/s$

If we measure at equal time intervals in  $\Delta t$ , then the intervals are not equally spaced in “real” time .

→ if we were to simulate the Nosé Hamiltonian, “real” time would fluctuate during simulation!

# Molecular Dynamics in an NVT ensemble

We now have “real” and “virtual” variables

real	virtual
$\mathbf{r}'$	$\mathbf{r}$
$\mathbf{p}'$	$\mathbf{p}/s$
$s'$	$s$
$\Delta t'$	$\Delta t/s$

If we measure at equal time intervals in  $\Delta t$ , then the intervals are not equally spaced in “real” time .

→ if we were to simulate the Nosé Hamiltonian, “real” time would fluctuate during simulation!

**Question:** How do we sample at equal intervals in “real” time?

# Molecular Dynamics in an NVT ensemble

Simple math shows that

$$\bar{A} = \lim_{\tau' \rightarrow \infty} \frac{1}{\tau'} \int_0^{\tau'} dt' A(\mathbf{p}(t')/s(t'), \mathbf{r}(t')) = \left\langle \frac{A(\mathbf{p}', \mathbf{r})}{s} \right\rangle / \left\langle \frac{1}{s} \right\rangle$$



# Molecular Dynamics in an NVT ensemble

Simple math shows that

$$\bar{A} = \lim_{\tau' \rightarrow \infty} \frac{1}{\tau'} \int_0^{\tau'} dt' A(\mathbf{p}(t')/s(t'), \mathbf{r}(t')) = \left\langle \frac{A(\mathbf{p}', \mathbf{r})}{s} \right\rangle / \left\langle \frac{1}{s} \right\rangle$$

Going back to the partition function, we can show

$$\begin{aligned} \left\langle \frac{A(\mathbf{p}', \mathbf{r})}{s} \right\rangle / \left\langle \frac{1}{s} \right\rangle &= \frac{\int d\mathbf{p}'^N d\mathbf{r}^N A(\mathbf{p}', \mathbf{r}) \exp(-\beta \mathcal{H}(\mathbf{p}', \mathbf{r}) 3N/L)}{\int d\mathbf{p}'^N d\mathbf{r}^N \exp(-\beta \mathcal{H}(\mathbf{p}', \mathbf{r}) (3N+1)/L)} \\ &= \frac{\int d\mathbf{p}'^N d\mathbf{r}^N \exp(-\beta \mathcal{H}(\mathbf{p}', \mathbf{r}) 3N/L)}{\int d\mathbf{p}'^N d\mathbf{r}^N \exp(-\beta \mathcal{H}(\mathbf{p}', \mathbf{r}) (3N+1)/L)} \\ &= \langle A(\mathbf{p}', \mathbf{r}) \rangle_{\text{canonical}} \end{aligned}$$

where we have assumed that  $L = 3N$

# Molecular Dynamics in an NVT ensemble

Simple math shows that

$$\bar{A} = \lim_{\tau' \rightarrow \infty} \frac{1}{\tau'} \int_0^{\tau'} dt' A(\mathbf{p}(t')/s(t'), \mathbf{r}(t')) = \left\langle \frac{A(\mathbf{p}', \mathbf{r})}{s} \right\rangle / \left\langle \frac{1}{s} \right\rangle$$

Going back to the partition function, we can show

$$\begin{aligned} \left\langle \frac{A(\mathbf{p}', \mathbf{r})}{s} \right\rangle / \left\langle \frac{1}{s} \right\rangle &= \frac{\frac{\int d\mathbf{p}'^N d\mathbf{r}^N A(\mathbf{p}', \mathbf{r}) \exp(-\beta \mathcal{H}(\mathbf{p}', \mathbf{r}) 3N/L)}{\int d\mathbf{p}'^N d\mathbf{r}^N \exp(-\beta \mathcal{H}(\mathbf{p}', \mathbf{r}) (3N+1)/L)}}{\frac{\int d\mathbf{p}'^N d\mathbf{r}^N \exp(-\beta \mathcal{H}(\mathbf{p}', \mathbf{r}) 3N/L)}{\int d\mathbf{p}'^N d\mathbf{r}^N \exp(-\beta \mathcal{H}(\mathbf{p}', \mathbf{r}) (3N+1)/L)}} \\ &= \langle A(\mathbf{p}', \mathbf{r}) \rangle_{\text{canonical}} \end{aligned}$$

where we have assumed that  $L = 3N$

→ The canonical partition function for the “real” variables.

# Molecular Dynamics in an NVT ensemble

- ▶ Nosé equations of motion can be written in terms of either virtual or real variables, but not convenient to simulate “virtual” variables.

# Molecular Dynamics in an NVT ensemble

- ▶ Nosé equations of motion can be written in terms of either virtual or real variables, but not convenient to simulate “virtual” variables.
- ▶ Virtual variables are determined from Nosé Hamiltonian.

# Molecular Dynamics in an NVT ensemble

- ▶ Nosé equations of motion can be written in terms of either virtual or real variables, but not convenient to simulate “virtual” variables.
- ▶ Virtual variables are determined from Nosé Hamiltonian.

“Real” equations of motion:

$$\frac{d\mathbf{r}'_i}{dt'} = s \frac{d\mathbf{r}_i}{dt} = s \frac{\partial \mathcal{H}_{NOSE}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}'_i}{m_i}$$

# Molecular Dynamics in an NVT ensemble

- ▶ Nosé equations of motion can be written in terms of either virtual or real variables, but not convenient to simulate “virtual” variables.
- ▶ Virtual variables are determined from Nosé Hamiltonian.

“Real” equations of motion:

$$\begin{aligned}\frac{d\mathbf{r}'_i}{dt'} &= s \frac{d\mathbf{r}_i}{dt} = s \frac{\partial \mathcal{H}_{NOSE}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}'_i}{m_i} \\ \frac{p'_i}{dt'} &= s \frac{d(\mathbf{p}_i/s)}{dt} = \frac{d\mathbf{p}_i}{dt} - \frac{\mathbf{p}_i}{s} \frac{ds}{dt} = -\frac{\partial \mathcal{H}_{NOSE}}{\partial \mathbf{r}_i} - \frac{\mathbf{p}'_i}{s} \frac{\partial \mathcal{H}_{NOSE}}{\partial p_s} \\ &= -\frac{\partial U(\mathbf{r}'^N)}{\partial \mathbf{r}'_i} - \frac{s' p'_s}{Q} \mathbf{p}'_i\end{aligned}$$

# Molecular Dynamics in an NVT ensemble

- ▶ Nosé equations of motion can be written in terms of either virtual or real variables, but not convenient to simulate “virtual” variables.
- ▶ Virtual variables are determined from Nosé Hamiltonian.

“Real” equations of motion:

$$\begin{aligned}\frac{d\mathbf{r}'_i}{dt'} &= s \frac{d\mathbf{r}_i}{dt} = s \frac{\partial \mathcal{H}_{NOSE}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}'_i}{m_i} \\ \frac{p'_i}{dt'} &= s \frac{d(\mathbf{p}_i/s)}{dt} = \frac{d\mathbf{p}_i}{dt} - \frac{\mathbf{p}_i}{s} \frac{ds}{dt} = -\frac{\partial \mathcal{H}_{NOSE}}{\partial \mathbf{r}_i} - \frac{\mathbf{p}'_i}{s} \frac{\partial \mathcal{H}_{NOSE}}{\partial p_s} \\ &= -\frac{\partial U(\mathbf{r}'^N)}{\partial \mathbf{r}'_i} - \frac{s' p'_s}{Q} \mathbf{p}'_i \\ \frac{ds'}{dt'} &= s \frac{ds}{dt} = s \frac{\partial \mathcal{H}_{NOSE}}{\partial p_s} = \frac{(s')^2 p'_s}{Q}\end{aligned}$$

# Molecular Dynamics in an NVT ensemble

- ▶ Nosé equations of motion can be written in terms of either virtual or real variables, but not convenient to simulate “virtual” variables.
- ▶ Virtual variables are determined from Nosé Hamiltonian.

“Real” equations of motion:

$$\begin{aligned}\frac{d\mathbf{r}'_i}{dt'} &= s \frac{d\mathbf{r}_i}{dt} = s \frac{\partial \mathcal{H}_{NOSE}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}'_i}{m_i} \\ \frac{p'_i}{dt'} &= s \frac{d(\mathbf{p}_i/s)}{dt} = \frac{d\mathbf{p}_i}{dt} - \frac{\mathbf{p}_i}{s} \frac{ds}{dt} = -\frac{\partial \mathcal{H}_{NOSE}}{\partial \mathbf{r}_i} - \frac{\mathbf{p}'_i}{s} \frac{\partial \mathcal{H}_{NOSE}}{\partial p_s} \\ &= -\frac{\partial U(\mathbf{r}'^N)}{\partial \mathbf{r}'_i} - \frac{s' p'_s}{Q} \mathbf{p}'_i \\ \frac{ds'}{dt'} &= s \frac{ds}{dt} = s \frac{\partial \mathcal{H}_{NOSE}}{\partial p_s} = \frac{(s')^2 p'_s}{Q} \\ \frac{d(s' p'_s/Q)}{dt'} &= \frac{s}{Q} \frac{dp_s}{dt} = -\frac{s}{Q} \frac{\partial \mathcal{H}_{NOSE}}{\partial s} = \left( \sum_i \frac{\mathbf{p}_i'^2}{m_i} - \frac{L}{\beta} \right) / Q\end{aligned}\quad (3)$$



## Molecular Dynamics in an NVT ensemble

Can show that these equations of motion conserve the “energy”

$$E_{NOSE} = \sum_{i=1}^N \frac{\mathbf{p}_i'^2}{2m_i} + U(\mathbf{r}^N) + \frac{s'^2 p_s'^2}{2Q} + L \frac{\ln s'}{\beta}$$

Note that this is *not* a Hamiltonian - equations of motion cannot be derived from it.

# Molecular Dynamics in an NVT ensemble

Hoover rewrote in terms of so-called “thermodynamic friction coefficient” (note that in previous equations,  $s$ ,  $p_s$  and  $Q$  only occur as

$$\xi = s' p'_s / Q$$

New equations of motion

$$\dot{\mathbf{r}}' = \frac{\mathbf{p}'_i}{m_i}$$

$$\dot{\mathbf{p}}' = -\frac{\partial U(\mathbf{r}'^N)}{\partial \mathbf{r}'_i} - \xi \mathbf{p}'_i$$

$$\dot{\xi} = \left( \sum_i \frac{p_i'^2}{m_i} - \frac{L}{\beta} \right) / Q$$

$$\frac{\dot{s}'}{s'} = \xi$$

Note that the first three equations make a complete set. The last is not necessary for simulations, but can be used as a check.

# Molecular Dynamics in an NVT ensemble

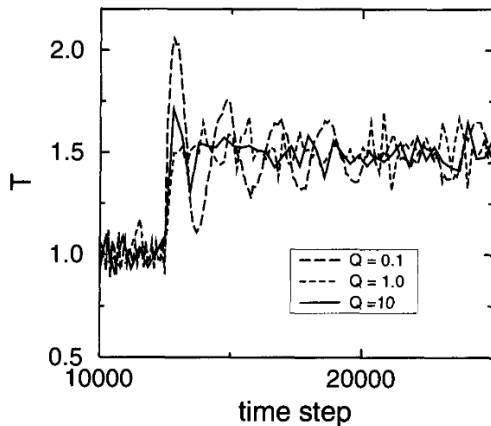
Nosé Hoover Equations of motion:

$$\dot{\mathbf{r}}' = \frac{\mathbf{p}'_i}{m_i} \quad ; \quad \dot{\mathbf{p}}' = -\frac{\partial U(\mathbf{r}'^N)}{\partial \mathbf{r}'_i} - \xi \mathbf{p}'_i \quad ; \quad \dot{\xi} = \left( \sum_i \frac{p_i'^2}{m_i} - \frac{L}{\beta} \right) / Q$$

How can we interpret them?

- ▶  $\xi > 0$  slows down all particles
- ▶  $\xi < 0$  speeds up all particles
- ▶ time evolution of  $\xi$  is proportional to the difference between the kinetic energy of the system and the kinetic energy of a system with temperature  $T$ .
- ▶  $Q$  sets how quickly  $\xi$  can change in magnitude

## Molecular Dynamics in an NVT ensemble



**Figure:** Lennard-Jones fluid. Taken from “Understanding Molecular Simulations” by Frenkel and Smit

Extra points:

- ▶ Can show that Nosé-Hoover algorithm only works when there is a single constant of motion - normally  $E_{NOSE}$

Extra points:

- ▶ Can show that Nosé-Hoover algorithm only works when there is a single constant of motion - normally  $E_{NOSE}$
- ▶ Leads to a problem when no external force is present as the momentum should be conserved, solution: initialize with zero momentum

Extra points:

- ▶ Can show that Nosé-Hoover algorithm only works when there is a single constant of motion - normally  $E_{NOSE}$
- ▶ Leads to a problem when no external force is present as the momentum should be conserved, solution: initialize with zero momentum
- ▶ For more general simulations, other modifications necessary - see e.g. Nosé Hoover chains (which couple to a chain of thermostats)

Extra points:

- ▶ Can show that Nosé-Hoover algorithm only works when there is a single constant of motion - normally  $E_{NOSE}$
- ▶ Leads to a problem when no external force is present as the momentum should be conserved, solution: initialize with zero momentum
- ▶ For more general simulations, other modifications necessary - see e.g. Nosé Hoover chains (which couple to a chain of thermostats)
- ▶ How to handle NPT? Follow a similar procedure and add a barostat.



## Extra points:

- ▶ Can show that Nosé-Hoover algorithm only works when there is a single constant of motion - normally  $E_{NOSE}$
- ▶ Leads to a problem when no external force is present as the momentum should be conserved, solution: initialize with zero momentum
- ▶ For more general simulations, other modifications necessary - see e.g. Nosé Hoover chains (which couple to a chain of thermostats)
- ▶ How to handle NPT? Follow a similar procedure and add a barostat.
- ▶ There are many MD simulators available online (LAMMPS, Gromacs, Espresso) - but one should use them carefully

## Extra points:

- ▶ Can show that Nosé-Hoover algorithm only works when there is a single constant of motion - normally  $E_{NOSE}$
- ▶ Leads to a problem when no external force is present as the momentum should be conserved, solution: initialize with zero momentum
- ▶ For more general simulations, other modifications necessary - see e.g. Nosé Hoover chains (which couple to a chain of thermostats)
- ▶ How to handle NPT? Follow a similar procedure and add a barostat.
- ▶ There are many MD simulators available online (LAMMPS, Gromacs, Espresso) - but one should use them carefully
- ▶ Two classic references on Molecular Dynamics include:
  - ▶ “Understanding Molecular Simulations” by Frenkel and Smit. Note that the Nose-Hoover derivation follows this book closely.
  - ▶ “Computer Simulations of Liquids” by Allen and Tildesley

Extra points:

- ▶ Can show that Nosé-Hoover algorithm only works when there is a single constant of motion - normally  $E_{NOSE}$
- ▶ Leads to a problem when no external force is present as the momentum should be conserved, solution: initialize with zero momentum
- ▶ For more general simulations, other modifications necessary - see e.g. Nosé Hoover chains (which couple to a chain of thermostats)
- ▶ How to handle NPT? Follow a similar procedure and add a barostat.
- ▶ There are many MD simulators available online (LAMMPS, Gromacs, Espresso) - but one should use them carefully
- ▶ Two classic references on Molecular Dynamics include:
  - ▶ “Understanding Molecular Simulations” by Frenkel and Smit. Note that the Nose-Hoover derivation follows this book closely.
  - ▶ “Computer Simulations of Liquids” by Allen and Tildesley

**Question:** What happens when interactions are discontinuous? (Hard particles, square wells, etc)? Clearly we cannot integrate the way we have described thus far.

Extra points:

- ▶ Can show that Nosé-Hoover algorithm only works when there is a single constant of motion - normally  $E_{NOSE}$
- ▶ Leads to a problem when no external force is present as the momentum should be conserved, solution: initialize with zero momentum
- ▶ For more general simulations, other modifications necessary - see e.g. Nosé Hoover chains (which couple to a chain of thermostats)
- ▶ How to handle NPT? Follow a similar procedure and add a barostat.
- ▶ There are many MD simulators available online (LAMMPS, Gromacs, Espresso) - but one should use them carefully
- ▶ Two classic references on Molecular Dynamics include:
  - ▶ “Understanding Molecular Simulations” by Frenkel and Smit. Note that the Nose-Hoover derivation follows this book closely.
  - ▶ “Computer Simulations of Liquids” by Allen and Tildesley

**Question:** What happens when interactions are discontinuous? (Hard particles, square wells, etc)? Clearly we cannot integrate the way we have described thus far.

**Solution:** Event driven molecular dynamics

## Event driven molecular dynamics

- ▶ Designed to study systems with discontinuous interactions, such as hard spheres, square wells, etc.

## Event driven molecular dynamics

- ▶ Designed to study systems with discontinuous interactions, such as hard spheres, square wells, etc.
- ▶ In these systems changes in the velocities of the particles only occur at specific times, called **events**.

## Event driven molecular dynamics

- ▶ Designed to study systems with discontinuous interactions, such as hard spheres, square wells, etc.
- ▶ In these systems changes in the velocities of the particles only occur at specific times, called **events**.
- ▶ **Idea:** Move system discontinuously through time hopping from event to event.

## Event driven molecular dynamics

- ▶ Designed to study systems with discontinuous interactions, such as hard spheres, square wells, etc.
- ▶ In these systems changes in the velocities of the particles only occur at specific times, called **events**.
- ▶ **Idea:** Move system discontinuously through time hopping from event to event.
- ▶ **Events:** Events are generally collisions or particles entering or exiting wells. New velocities are determined by requiring conservation of energy and momentum.



# Event driven molecular dynamics

- ▶ Designed to study systems with discontinuous interactions, such as hard spheres, square wells, etc.
- ▶ In these systems changes in the velocities of the particles only occur at specific times, called **events**.
- ▶ **Idea**: Move system discontinuously through time hopping from event to event.
- ▶ **Events**: Events are generally collisions or particles entering or exiting wells. New velocities are determined by requiring conservation of energy and momentum.
- ▶ **Prediction of Events**: Simulation method is mainly focused on predicting and intelligently scheduling events. The scheduling is usually done with a so-called “binary event tree”.

## Event driven molecular dynamics

- ▶ Designed to study systems with discontinuous interactions, such as hard spheres, square wells, etc.
- ▶ In these systems changes in the velocities of the particles only occur at specific times, called **events**.
- ▶ **Idea**: Move system discontinuously through time hopping from event to event.
- ▶ **Events**: Events are generally collisions or particles entering or exiting wells. New velocities are determined by requiring conservation of energy and momentum.
- ▶ **Prediction of Events**: Simulation method is mainly focused on predicting and intelligently scheduling events. The scheduling is usually done with a so-called “binary event tree”.
- ▶ Extensions to this method also allow for constant temperature simulations
- ▶ References:
  - ▶ “The event-driven approach to N-body simulation”, Progress of Theoretical Physics Supplement **178**, 5 (2009), D.C. Rapaport (This is a reasonable introduction. Note that same author has written a book on the topic).
  - ▶ “Discontinuous molecular dynamics for semiflexible and rigid bodies”, Journal of Chemical Physics **126**, 074105 (2007), L.H. de la Peña, R. van Zon, J. Schofield