National Center for Theoretical Sciences
Physics Division 國家理論科學研究中心 物理組

NCTS 2022 Frontiers of Complex Systems Winter School

2022/02/10
09:05~10:35

Fundamentals of Molecular Dynamics Simulations

Pai-Yi Hsiao (蕭百沂)
Department of Engineering and System Science
National Tsing Hua University

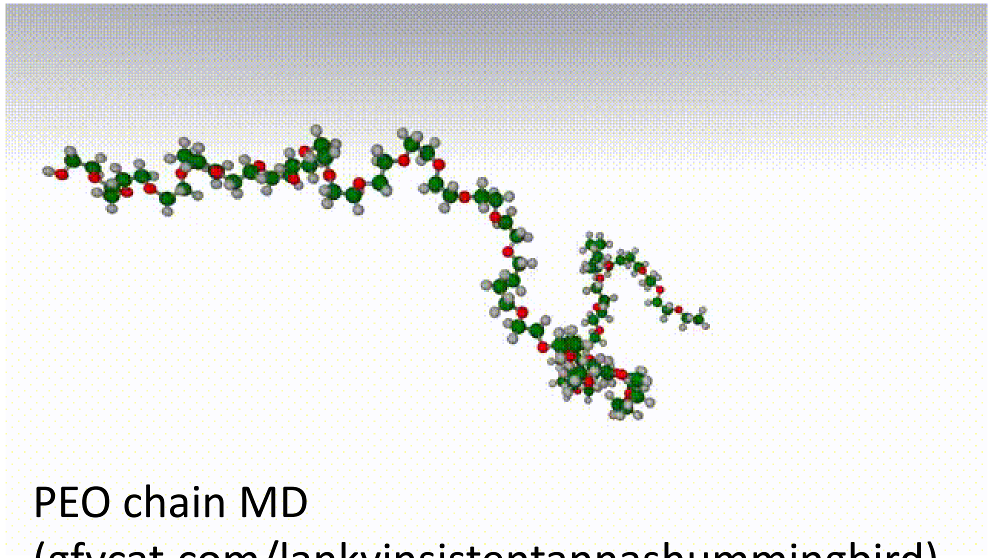
Molecular Dynamics (MD) Simulation

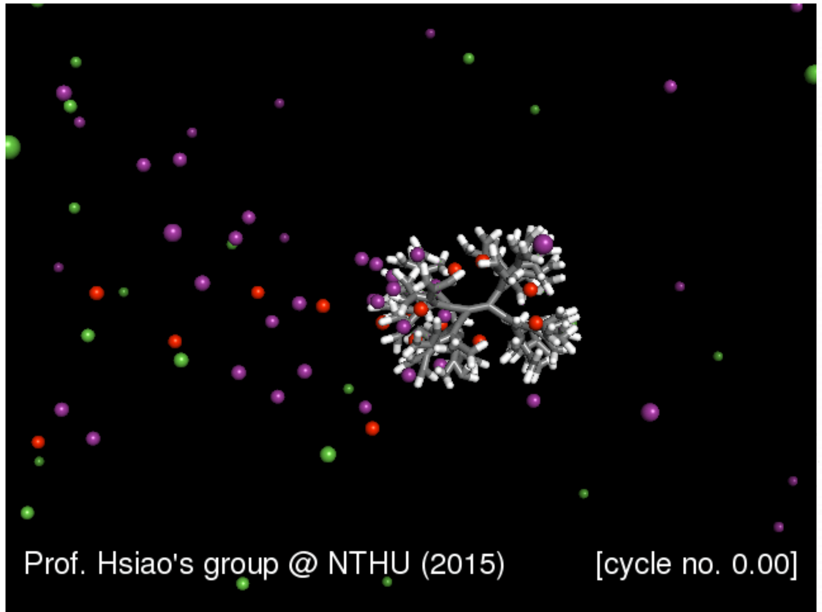
- Computer simulation method

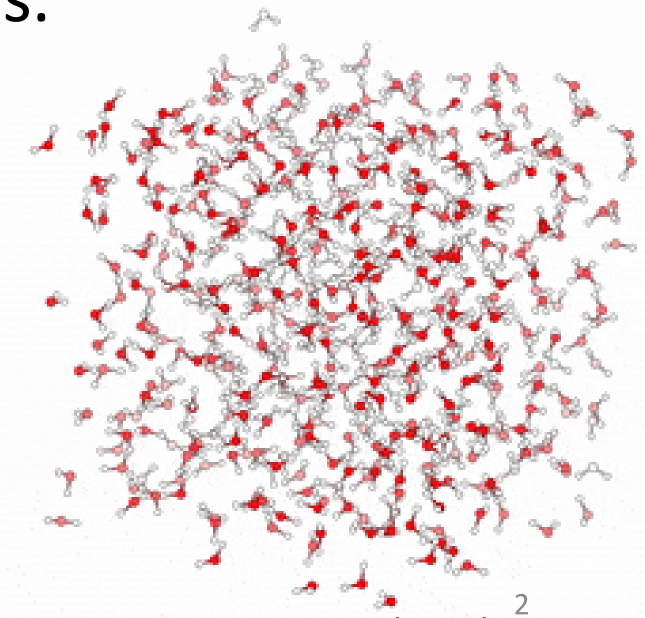
which simulates dynamic evolution of a microscopic system (ie. physical movements of atoms and molecules),

The behavior and properties of the system are then analyzed.

Typical size of system can be investigated: $< 10^6$ atoms.


PEO chain MD (gfycat.com/lankyinsistentannashummingbird)




water MD (wiki)

Primary goal of MD simulation: computing the trajectories of atoms/particles in a system

- Numerical method is applied to solve Newton’s equations of motion.

$$m_1 \frac{dv_1}{dt} = F_1(r_1, r_2, \dots), \quad \frac{dr_1}{dt} = v_1$$

$$m_2 \frac{dv_2}{dt} = F_2(r_1, r_2, \dots), \quad \frac{dr_2}{dt} = v_2$$

……..

$$m_i \frac{dv_i}{dt} = F_i(r_1, r_2, \dots), \quad \frac{dr_i}{dt} = v_i$$

……..

for $i = 1, \dots, N$

An initial value problem:

Given $\{r_i(0), v_i(0)\}_{i=1}^{N}$, find atomic trajectories at any time moment: $\{r_i(t), v_i(t)\}_{i=1}^{N}$

Physics of a system are known if we know the trajectory, more precisely, the phase-space trajectory $\{r_i(t), p_i(t)\}_{i=1}^{N}$

Hamitonian $H(\{r_i, p_i\}) = \left(\sum_{i=1}^{N} \frac{p_i^2}{2m_i}\right) + U(\{r_i, p_i\})$

$\Rightarrow$ physical quantity $A(\{r_i, p_i\})$

Topics of study: any kind of energy, pressure, temperature, force, interaction, ….

static properties: molecular shape, size, crystalline structure, dislocation, g(r), …

dynamic properties: diffusion, heat transfer, time correlaction,….

responses: shearing, stress, strain, electric field, temperature gradient …..

Domains of application: physics, chemistry, engineering, materials sciences, molecular biology,…

Examples:

- Instantaneous Temperature: $T(t) = \frac{2}{3(N-1)k_B}\left(\sum_{i=1}^{N}\frac{p_i^2}{2m_i}\right)$

- Pressure: $P(t) = \frac{Nk_BT}{V} - \frac{1}{3V}\left\langle\sum_{i=1}^{N}\sum_{j=i+1}^{N} r_{ij}\cdot F_{ij}\right\rangle$

  where $r_{ij} = r_i - r_j,$ F_{ij} = force exerted on i by j

- Shear viscosity η:

$$6\eta k_BTVt = \sum_{\alpha\beta=xy,yz,zx}\left\langle\left(\sum_{j=1}^{N} m_j r_{j\alpha}(t)v_{j\beta}(t) - \sum_{j=1}^{N} m_j r_{j\alpha}(0)v_{j\beta}(0)\right)^2\right\rangle$$

- Diffusion coefficient: MSD $\left\langle\frac{1}{N}\sum_{i=1}^{N}\left(r_i(t_0+t) - r_i(t_0)\right)^2\right\rangle = 6Dt$

Time scale and system size can be handled by MD simulations

- Size of a molecule (monomer): $\sigma \sim 10$ Å
- Mass of a molecule (monomer): M ~ 100 g/mol
- Energy: $\varepsilon \sim k_B T = (1.38\times10^{-23}) \times 300$ J

$\Rightarrow$Characteristic time scale: $t_u = \sigma\sqrt{M/\varepsilon}$ ~ 6.3 ps : $[\text{energy}] = [\text{mass}]\left[\frac{\text{length}}{\text{time}}\right]^2$

$\Rightarrow$ time step to integrate Eq. of motion: $\Delta t \sim 0.001 - 0.01\ t_u$ ~ 6.3- 63 fs

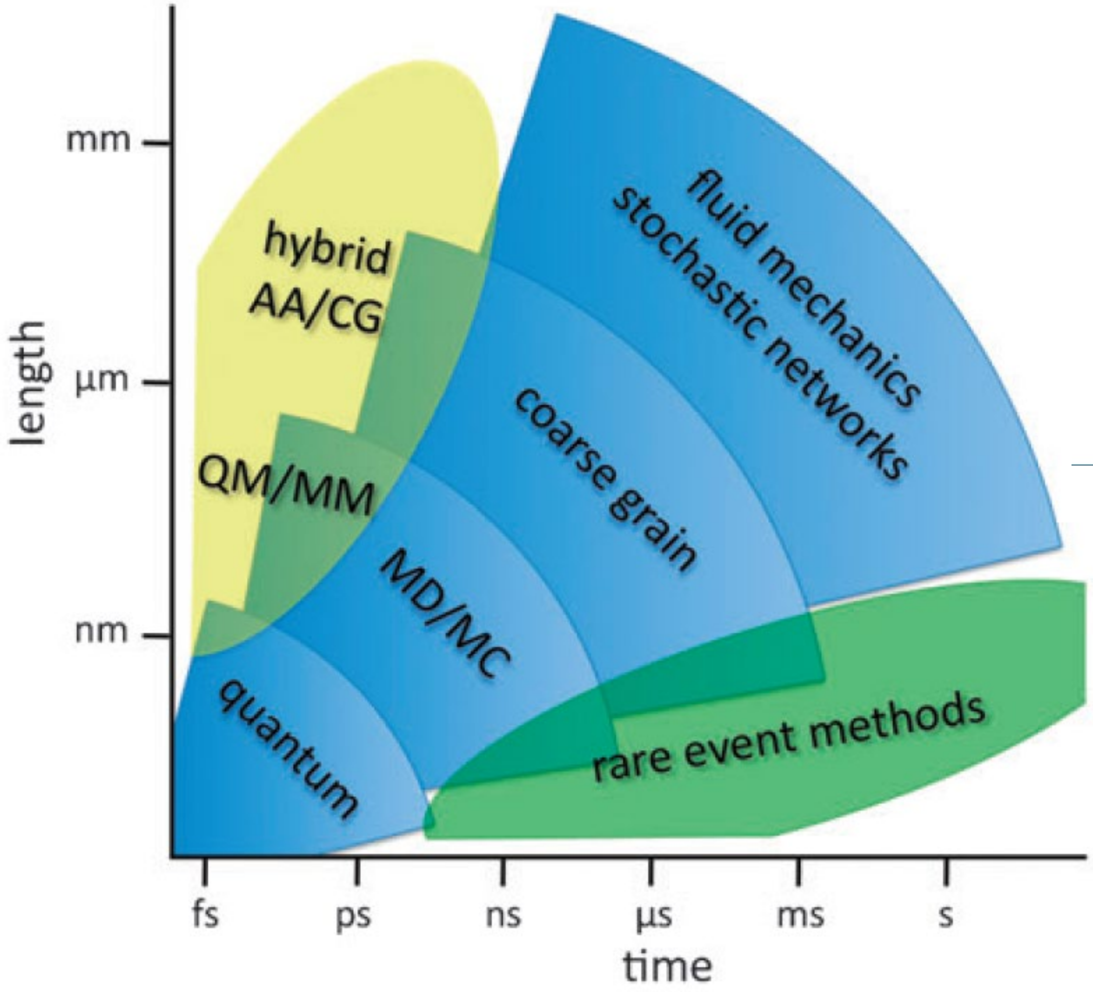
- No. of molecules: < ~ 10^6 [limited by the memory of computer]

$\Rightarrow$ system size: L ~ 100 σ ~ 100 nm $\leftarrow$ of order of nanometers

- Maximum no. of steps for integration: 10^9
  (4 byte integer: 2^31=2.14 $\times 10^9$) [limited by the speed of computer]

$\Rightarrow$ total simulation time up to ~ 63 μs $\leftarrow$ of order of microseconds

Length & time scales accessible by different simulation methods



Nielsen et al. PCCP 12 (2010) 12401

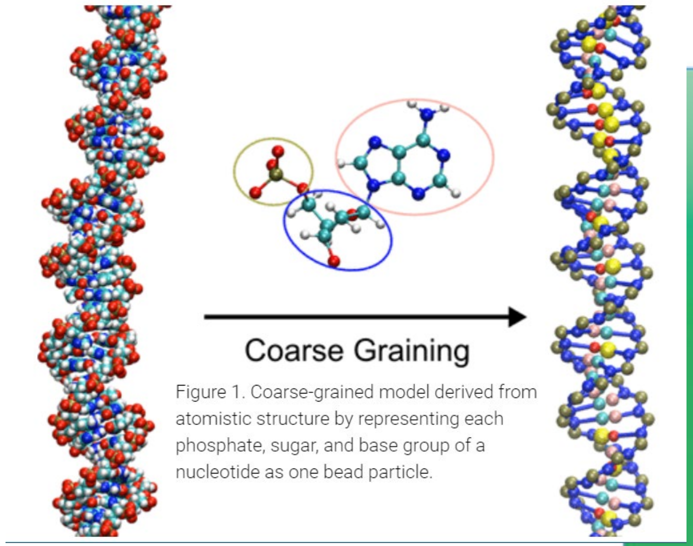
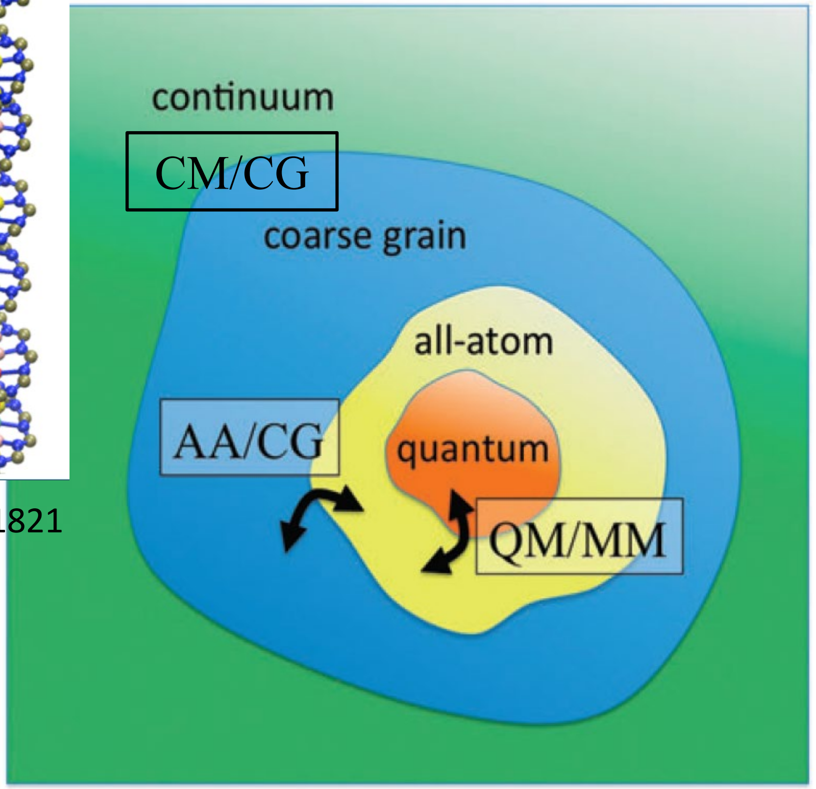


Figure 1. Coarse-grained model derived from atomistic structure by representing each phosphate, sugar, and base group of a nucleotide as one bead particle.

Markegard et al. JPCB 119 (2015) 1821



CM: continuum mechanics
CG: coarse grained
AA: all-atom
MM: molecular mechanics
QM: quantum mechanics

How to solve the equations of motion?

Finite difference method: approximating the derivatives by finite differences

$m\frac{dv}{dt} = F$ approxi. by $m\frac{\Delta v}{\Delta t} \cong F$, which is $\frac{\Delta v}{\Delta t} = \frac{v(t+\Delta t)-v(t)}{\Delta t} \cong \frac{F(t)}{m}$

$\frac{dr}{dt} = v$ approxi. by $\frac{\Delta r}{\Delta t} \cong v$, which is $\frac{\Delta r}{\Delta t} = \frac{r(t+\Delta t)-r(t)}{\Delta t} \cong v(t)$

we have $v(t+\Delta t) \cong v(t) + \frac{F(t)}{m}\Delta t$ [Euler's method: truncation error $O(\Delta t^2)$]

$r(t+\Delta t) \cong r(t) + v(t)\Delta t$

Physical meaning: the velocity and position at the next time step can be predicted by using the information of the velocity, position, force at the current step.

The trajectory can be calculated, step by step, if the initial value is given:

$\{r_i(0), v_i(0)\}_{i=1}^{N} \rightarrow \{r_i(\Delta t), v_i(\Delta t)\} \rightarrow \{r_i(2\Delta t), v_i(2\Delta t)\} \rightarrow \{r_i(3\Delta t), v_i(3\Delta t)\} \rightarrow$ ……

---- **an initial value problem (IVP)**

Other methods learned in Numerical Analysis (1)

- **Runge-Kutta method (4th order)** for 2nd order ODE; truncation error $O(\Delta t^4)$

$$\frac{dv}{dt} = F(t, x, v), \qquad \frac{dx}{dt} = G(t, x, v), \qquad x(0) = x_0, \qquad v(0) = v_0$$

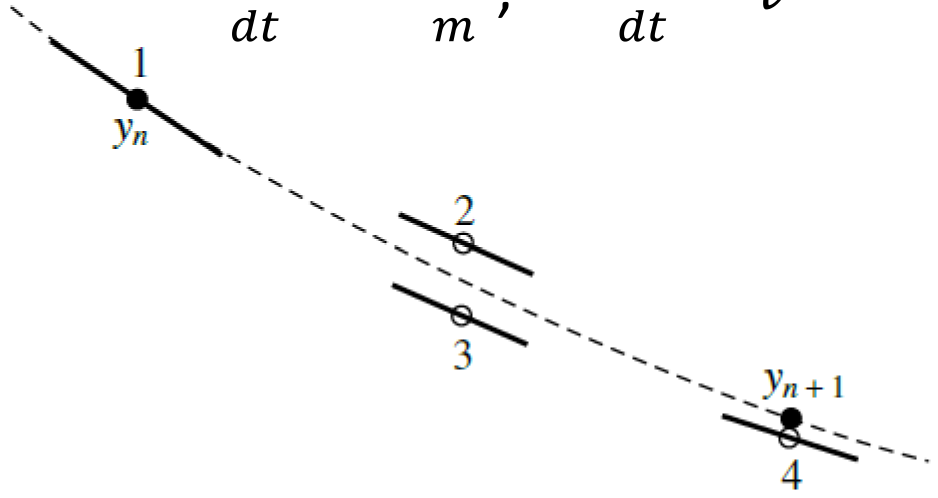
Time step: $\Delta t = h, \quad t_n = n\Delta t$

$$k_1 = hG(t_n,\ x_n,\ v_n)$$
$$\ell_1 = hF(t_n,\ x_n,\ v_n)$$
$$k_2 = hG(t_n + 0.5h,\ x_n + 0.5k_1,\ v_n + 0.5\ell_1)$$
$$\ell_2 = hF(t_n + 0.5h,\ x_n + 0.5k_1,\ v_n + 0.5\ell_1)$$
$$k_3 = hG(t_n + 0.5h,\ x_n + 0.5k_2,\ v_n + 0.5\ell_2)$$
$$\ell_3 = hF(t_n + 0.5h,\ x_n + 0.5k_2,\ v_n + 0.5\ell_2)$$
$$k_4 = hG(t_n + h,\ x_n + k_3,\ v_n + \ell_3)$$
$$\ell_4 = hF(t_n + h,\ x_n + k_3,\ v_n + \ell_3)$$
$$\Rightarrow\ x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$v_{n+1} = v_n + \frac{1}{6}(\ell_1 + 2\ell_2 + 2\ell_3 + \ell_4)$$

For time steps $t_1, t_2, \dots, t_M$

(ex) Simple Harmonic Oscillator

$$\frac{dv}{dt} = -\frac{kx}{m}, \qquad \frac{dx}{dt} = v$$



$$x(0), v(0) \rightarrow x(\Delta t), v(\Delta t) \rightarrow x(2\Delta t), v(2\Delta t) \rightarrow x(3\Delta t), v(3\Delta t) \rightarrow \dots\dots$$

Other methods learned in Numerical Analysis (2)

- Predictor-corrector method of order q: (PEC)$\rightarrow$ (PEC) $\rightarrow$ (PEC) $\rightarrow$ …

$$r(t+\Delta t) = r(t) + \frac{r'(t)}{1!}\Delta t + \frac{r''(t)}{2!}\Delta t^2 + \frac{r'''(t)}{3!}\Delta t^3 + \ldots$$

$$r'(t+\Delta t) = r'(t) + \frac{r''(t)}{1!}\Delta t + \frac{r'''(t)}{2!}\Delta t^2 + \frac{r''''(t)}{3!}\Delta t^3 + \ldots$$

Let $\xi = \begin{pmatrix} r \\ r'\Delta t/1! \\ r''\Delta t^2/2! \\ r'''\Delta t^3/3! \\ r''''\Delta t^4/4! \end{pmatrix}$ (1) Prediction: $\xi^{(p)}|_{t+\Delta t} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 3 & 6 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \xi|_t$

(2) Calculate the acceleration $r''(t+\Delta t)$ from the predicted position $r^{(p)}(t+\Delta t)$.

Discrepancy $\Delta R \equiv \frac{\Delta t^2}{2!}\left(r''(t+\Delta t) - r''^{(p)}(t+\Delta t)\right)$

(3) Correction: $\xi|_{t+\Delta t} = \xi^{(p)}|_{t+\Delta t} + \Delta R \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}$

Truncation error: $O(\Delta t^{q+1})$

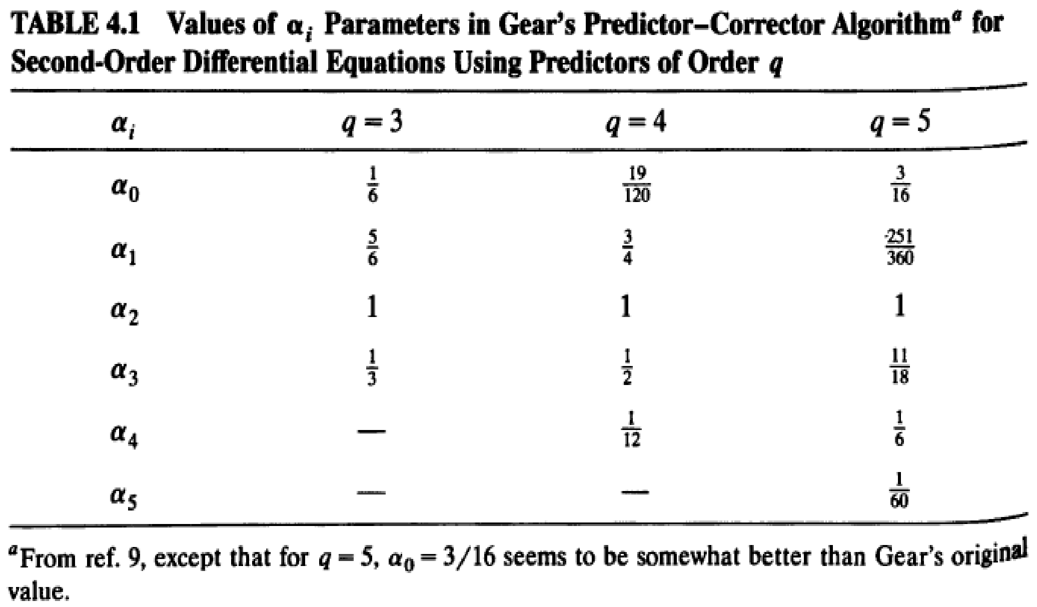


TABLE 4.1 Values of α_i Parameters in Gear's Predictor–Corrector Algorithm[a] for Second-Order Differential Equations Using Predictors of Order q

α_i	$q=3$	$q=4$	$q=5$
α_0	$\frac{1}{6}$	$\frac{19}{120}$	$\frac{3}{16}$
α_1	$\frac{5}{6}$	$\frac{3}{4}$	$\frac{251}{360}$
α_2	1	1	1
α_3	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{11}{18}$
α_4	—	$\frac{1}{12}$	$\frac{1}{6}$
α_5	—	—	$\frac{1}{60}$

[a]From ref. 9, except that for $q=5$, $\alpha_0 = 3/16$ seems to be somewhat better than Gear's original value.

Chasing only for the accuracy of calculation: Essential requirement: time-reversal symmetry, is not held in the above methods

$$\begin{tikzcd} (r,v)\big|_{t} \arrow[r, bend left, "\Delta t"] & (r,v)\big|_{t+\Delta t} \arrow[l, bend left, "?", "-\Delta t"'] \end{tikzcd}$$

- Verlet's method (1967): a time-reversible algorithm,

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + r''(t)\Delta \mathrm{t}^2 + \mathrm{O}(\Delta t^4)$$

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + O(\Delta t^3)$$

derived by adding and subtracting the two Taylor series expansions:

$$r(t+\Delta t) = r(t) + r'(t)\Delta t + \frac{1}{2} r''(t)\Delta t^2 + \frac{1}{3!} r'''(t)\Delta t^3 + O(\Delta t^4)$$

$$r(t-\Delta t) = r(t) - r'(t)\Delta t + \frac{1}{2} r''(t)\Delta t^2 - \frac{1}{3!} r'''(t)\Delta t^3 + O(\Delta t^4)$$

Advantages: simplicity, good stability, time symmetry,

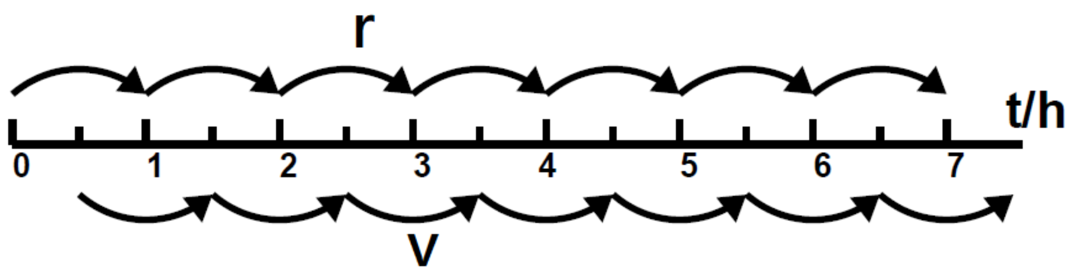
suitable for studying molecular dynamics

Leapfrog algorithm, a modern version of Verlet's method

* position is updated by using midpoint rule for velocity
* velocity is also updated by using midpoint for position

$$r(t_1) = r(t_0) + v(t_{0.5})\Delta t$$

$$v(t_{1.5}) = v(t_{0.5}) + a(r(t_1))\Delta t$$



We'd like the position and velocity being calculated at the same time:

(1) velocity Verlet:

$$v(t_{n+0.5}) = v(t_n) + a(r(t_n))\frac{\Delta t}{2}$$
$$r(t_{n+1}) = r(t_n) + v(t_{n+0.5})\Delta t$$
$$v(t_{n+1}) = v(t_{n+0.5}) + a(r(t_{n+1}))\frac{\Delta t}{2}$$

(2) position Verlet:

$$r(t_{n+0.5}) = r(t_n) + v(t_n)\frac{\Delta t}{2}$$
$$v(t_{n+1}) = v(t_n) + a(r(t_{n+0.5}))$$
$$r(t_{n+1}) = r(t_{n+0.5}) + v(t_{n+1})\frac{\Delta t}{2}$$

By this way, we have the value (r, v) at each time step $t = n\Delta t$.

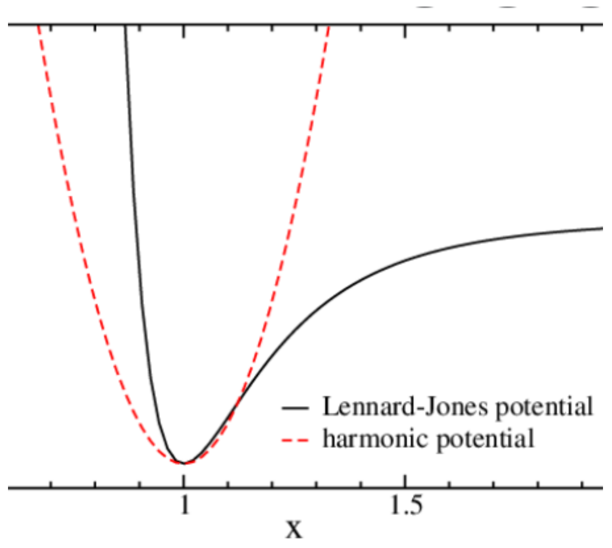
So physical quantities can be calculated at each time step.

How to choose integration time step Δt ?

- Generally, $\omega_j \Delta t < 1$

where ω_j is the frequency of interaction j.

So that, the oscillation behavior due to the interaction can be described correctly in the numerical method.



- The smaller Δt, the smaller the truncation error, and the shorter duration time and the trajectory distance can be explored.

Note: Extremely small Δt also lead to large error in the calculation because of computer’s round-off error.

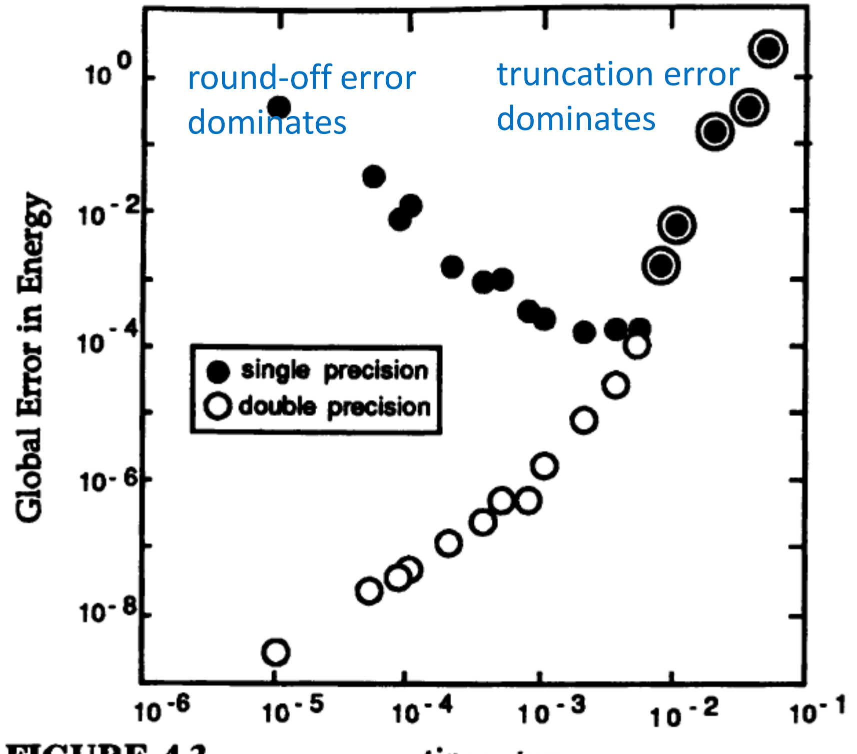


FIGURE 4.3

[Haile’s MD book]

Calculation of force

- Total potential energy $= U(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots)$

  The force exerted on the ith particle: $\vec{F}_i = -\nabla_i \, U(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots) = -\frac{\partial}{\partial \vec{r}_i} U(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots)$

(ex) particles interacting with each other via van der Waals interaction:

$$U(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots) = \frac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{-A}{\left|\vec{r}_i - \vec{r}_j\right|^6}$$

$$\vec{F}_i = -\frac{\partial}{\partial \vec{r}_i} U = \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{-6A}{r_{ij}^8} \vec{r}_{ij} \equiv \sum_{\substack{j=1 \\ j \neq i}}^{N} \vec{f}_{ij}$$

The equation of motion for particle i,

$$m_i \frac{dv_{ix}}{dt} = F_{ix}, \quad \frac{dr_{ix}}{dt} = v_{ix}$$

$$m_i \frac{dv_{iy}}{dt} = F_{iy}, \quad \frac{dr_{iy}}{dt} = v_{iy}$$

$$m_i \frac{dv_{iz}}{dt} = F_{iz}, \quad \frac{dr_{iz}}{dt} = v_{iz}, \quad i = 1, \dots, N,$$

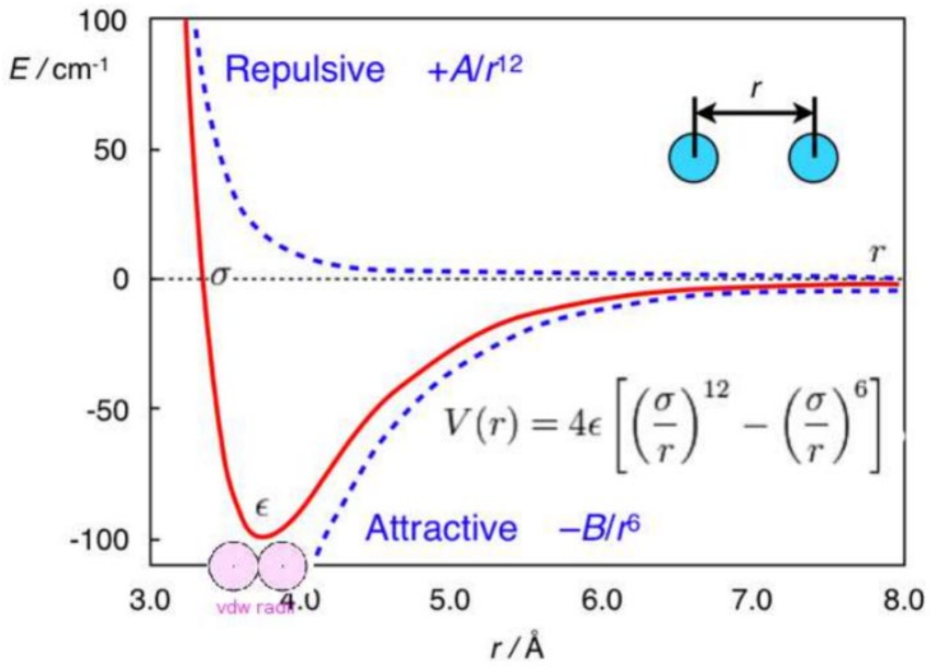
[totally, $6N$ ODEs]

Interaction: (I) non-bonded, (II) bonded interaction

(I) non-bonded interactions:

a. Lennard-Jones interaction:

$$u_{LJ}(r) = \frac{A}{r^{12}} - \frac{B}{r^6} = 4\varepsilon_{LJ}\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$



b. Buckingham potential:

$$u_{Bh}(r) = Ae^{-Cr} - \frac{B}{r^6}$$

c. Coulomb interaction:

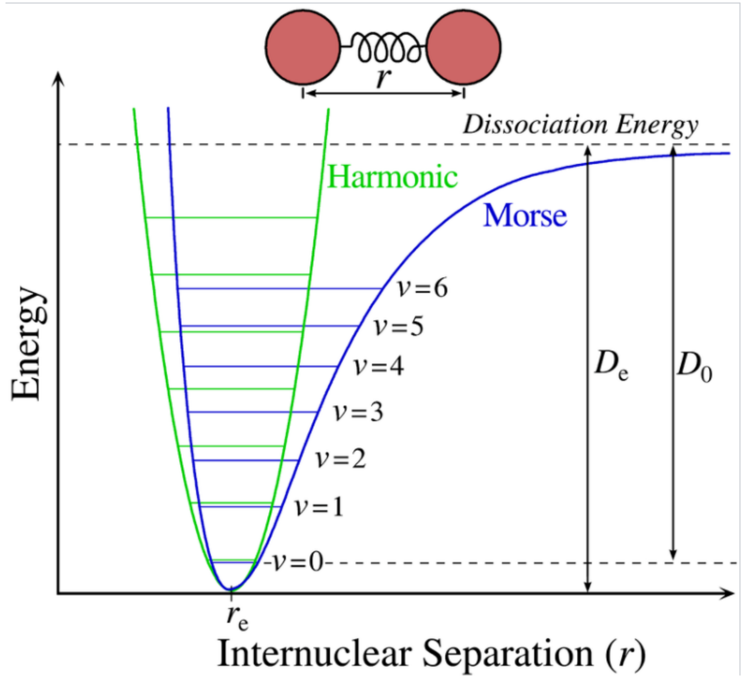
$$u_{col}(r) = \frac{q_1 q_2}{4\pi\varepsilon\varepsilon_0 r}$$

Can be further distinguished into "short-range interaction": a, b

and "long-range interaction" : c

(II) bonded interaction

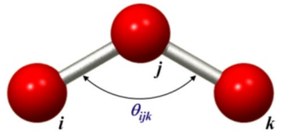
a. Harmonic bond: $u_{bd}(r_{ij}) = \frac{1}{2}k_b(r_{ij} - b_{ij})^2$



b. Morse potential:

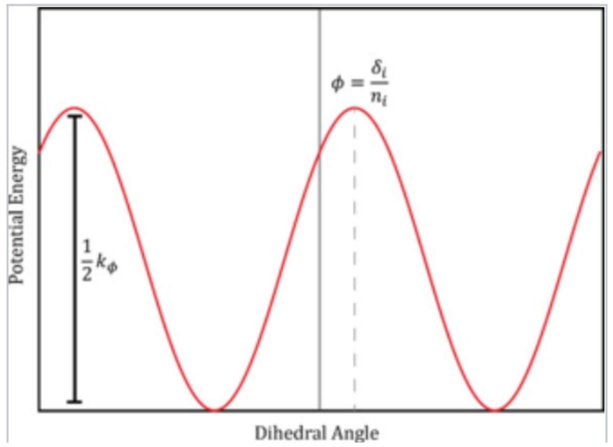
$$u_{mr}(r_{ij}) = D_{ij}[1 - \exp(-B_{ij}(r_{ij} - b_{ij})]^2$$

c. Harmonic angle potential:

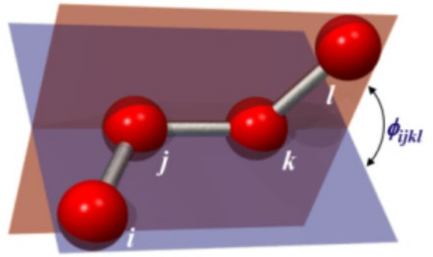


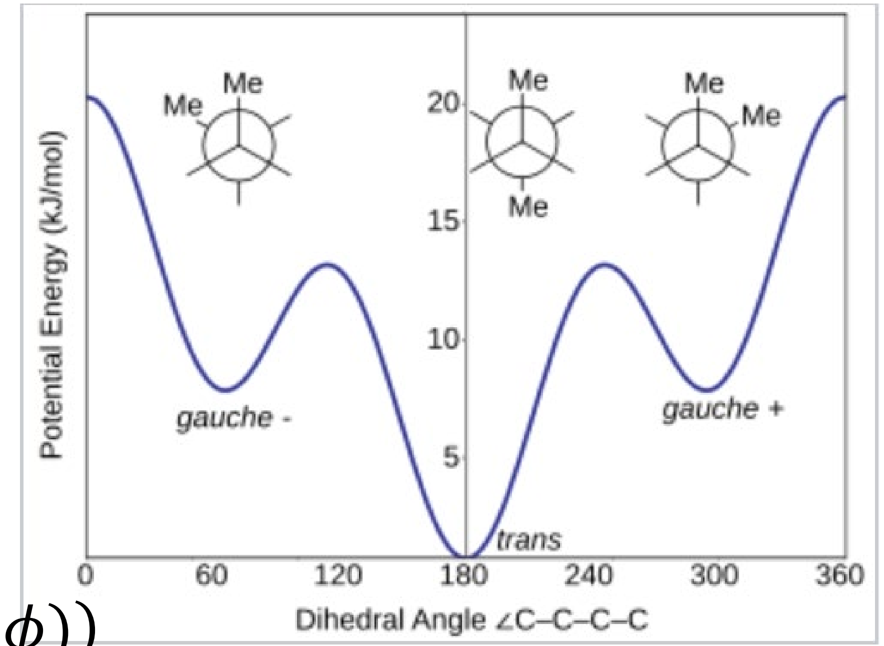
$$u_a(\theta_{ijk}) = \frac{1}{2}k_a(\theta_{ijk} - \theta^0_{ijk})^2$$

d. Dihedral angle potential:



$$u_h(\phi_{ijkl}) = k_h(1 + \cos(n\phi_{ijkl} - \phi_0))$$





$$u_h(\phi) = k_1(1 + \cos(\phi)) + k_2(1 + \cos(2\phi)) + k_3(1 + \cos(3\phi))$$

Class I Force field:

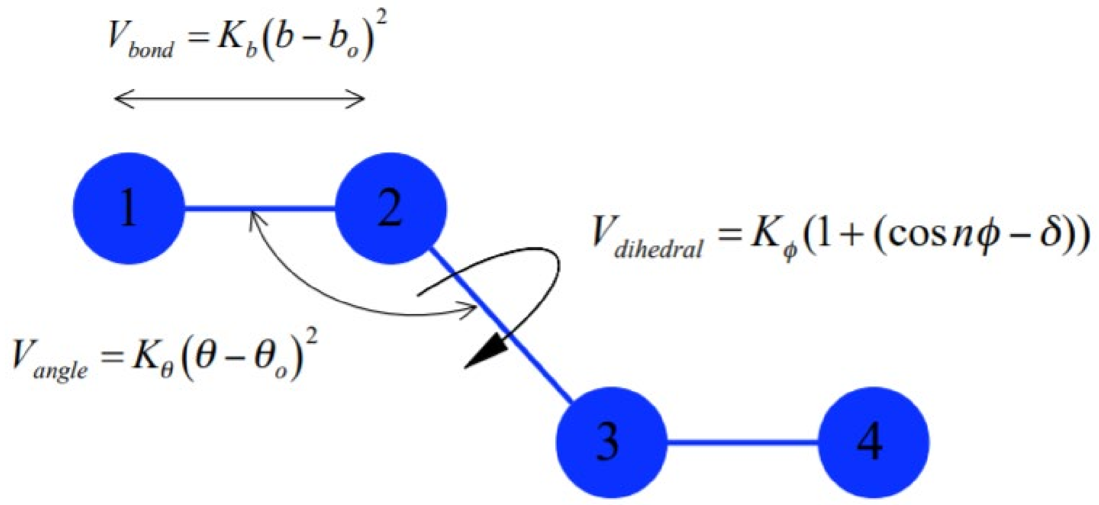
$$E = \sum_{bonds} k_b(b-b_0)^2 + \sum_{angles} k_\theta(\theta-\theta_0)^2$$

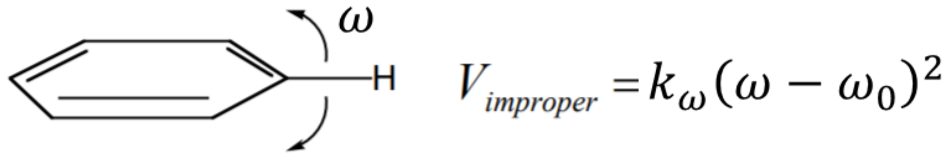
$$+ \sum_{dihedrals} k_\phi(1+\cos(n\phi-\delta))$$

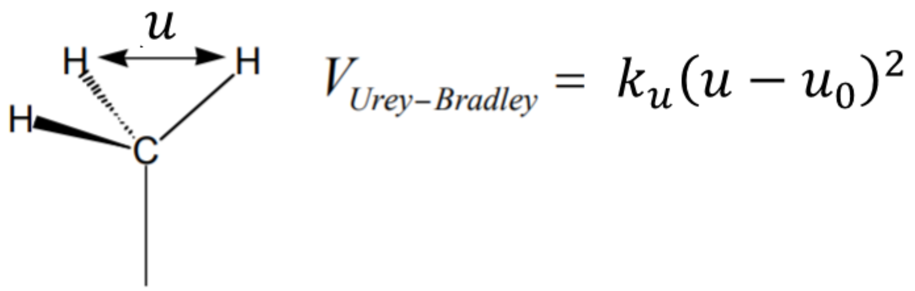
$$+ \sum_{impropers} k_\omega(\omega-\omega_0)^2$$

$$+ \sum_{Urey-Bradley} k_u(u-u_0)^2$$

$$+ \sum_{vdW} 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] + \sum_{coulomb} \frac{q_i q_j}{4\pi\varepsilon\varepsilon_0 r_{ij}}$$


$V_{bond} = K_b(b-b_o)^2$
1
2
$V_{dihedral} = K_\phi(1+(\cos n\phi-\delta))$
$V_{angle} = K_\theta(\theta-\theta_o)^2$
3
4





CHARMM, AMBER, GROMACS, OPLS

Class II Force field:

involving further anharmonic terms and cross terms

$$E = E_{vdw} + E_{coul} +$$

$$\sum_{bonds}\left[K_{b,2}(b-b_o)^2 + K_{b,3}(b-b_o)^3 + K_{b,4}(b-b_o)^4\right]$$

$$+\sum_{angles}\left[K_{\theta,2}(\theta-\theta_o)^2 + K_{\theta,3}(\theta-\theta_o)^3 + K_{\theta,4}(\theta-\theta_o)^4\right]$$

$$+\sum_{dihedrals}\left[K_{\phi,1}(1-\cos\phi) + K_{\phi,2}(1-\cos 2\phi) + K_{\phi,3}(1-\cos 3\phi)\right]$$

$$+\sum_{impropers} K_\chi \chi^2$$

$$+\sum_{bonds}\sum_{bonds'} K_{bb'}(b-b_o)(b'-b_o') + \sum_{angles}\sum_{angles'} K_{\theta\theta'}(\theta-\theta_o)(\theta'-\theta_o')$$

$$+\sum_{bonds}\sum_{angles} K_{b\theta}(b-b_o)(\theta-\theta_o)$$

$$+\sum_{bonds}\sum_{dihedrals}(b-b_o)\left[K_{\phi,b1}\cos\phi + K_{\phi,b2}\cos 2\phi + K_{\phi,b3}\cos 3\phi\right]$$

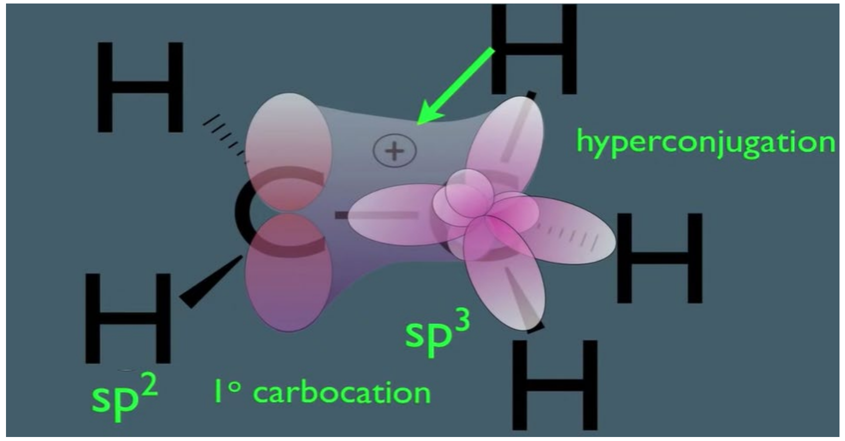
$$+\sum_{angles}\sum_{dihedrals}(\theta-\theta_o)\left[K_{\phi,\theta 1}\cos\phi + K_{\phi,\theta 2}\cos 2\phi + K_{\phi,\theta 3}\cos 3\phi\right]$$

$$+\sum_{angles}\sum_{angles'}\sum_{dihedrals} K_{\theta\theta'\phi}(\theta-\theta_o)(\theta'-\theta_o')\cos\phi$$

MM3, UFF, MMFF

- **Class III Force Field**
  considering more chemical effects such as electronegativity and hyperconjugation, and polarization (treating electrostatic interactions with higher-order moments up to quadrupoles)

  CHARMM, ABMER, AMOEBA



- **Ab initio Force Field**

$$E = E_{coulomb} + E_{dispersion} + E_{polarization} + E_{charge-transfer} + E_{exchange-repulsion}$$

[Xu et al. JCP 148 (2018) 090901]

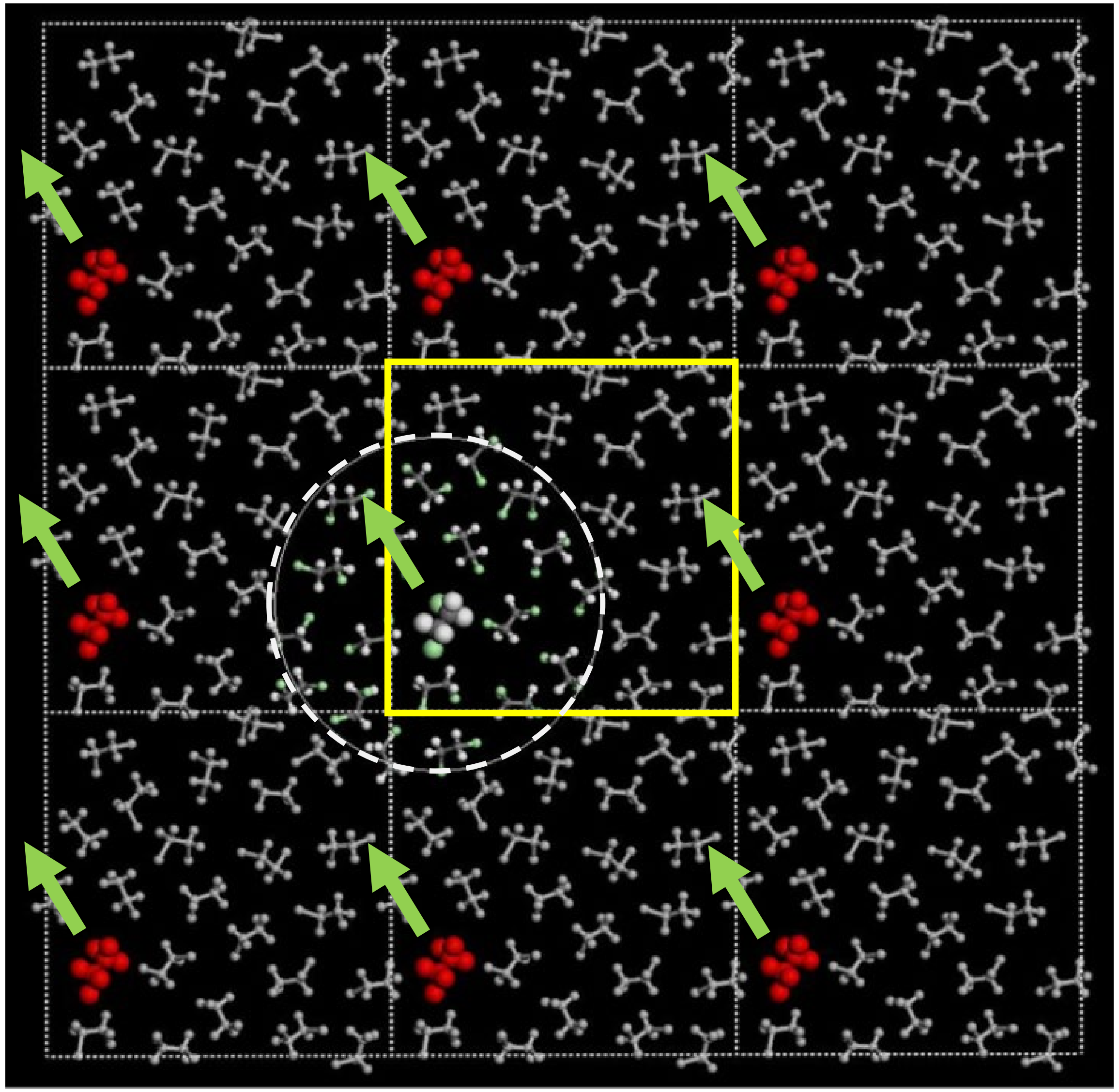
Only small system size (~100 nm) can be simulated by MD. Unexpected boundary or interface effect will come in?

N particle system:

no. particles near bd. $\approx 6\left(\sqrt[3]{N}\right)^2$

fraction. particles near bd. $\approx \dfrac{\sim 6N^{2/3}}{N} = 6N^{-1/3}$

- In a real system, $N = 10^{23}$
  fraction near bd $\approx 1.3 \times 10^{-7}$
- In a simulation, $N = 10^5$
  fraction near bd ≈ 0.13



Method to reduce the unexpected boundary effect

Periodic Boundary Condition (pbc)

NOTE: Calculation of interaction under pbc:

Minimum-image convention

particle interacts with the closest image of the remaining particles in the system

MD simulation is naturally run under fixed NVE condition :

- number of particles N is fixed
- Hamiltonian does not depend on time $\Rightarrow$ total energy E is conserved
- box size of simulation is not changed

How to perform simulations at a desired temperature or other ensembles ?

Fact: “equivalence of ensembles”:

NVE ensemble: <T>, <P>, <μ>, …

$\Leftrightarrow$ NVT ensemble: <E>, <P>, <μ>, …

$\Leftrightarrow$ NPT ensemble: <E>, <V>, <μ>, …

$\Leftrightarrow$ μVT ensemble: <E>, <P>, <N>, …

in the limit of thermodynamics

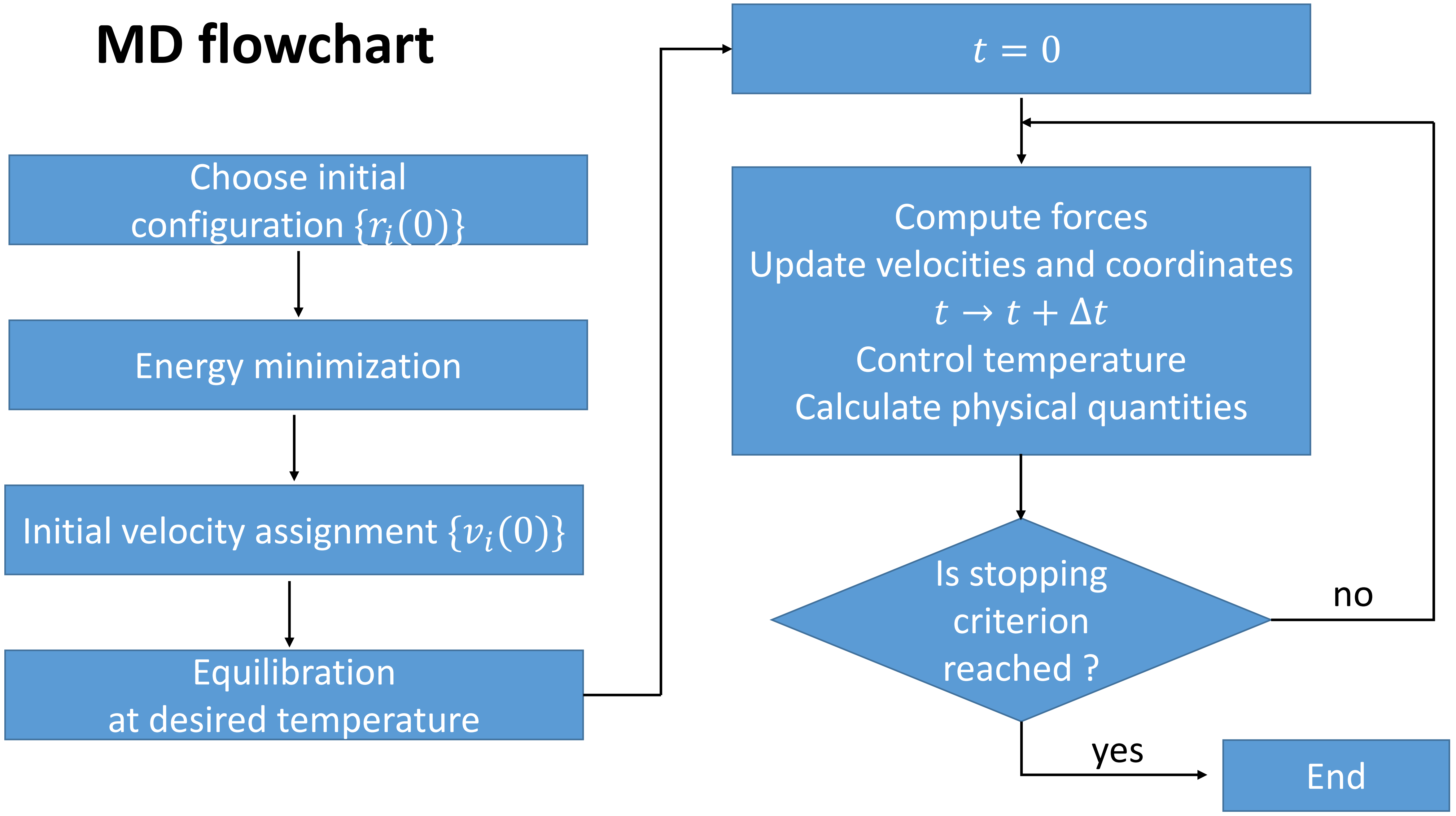
Forcing the system to relax at the desired temperature T_D

(ex) velocity scaling method:

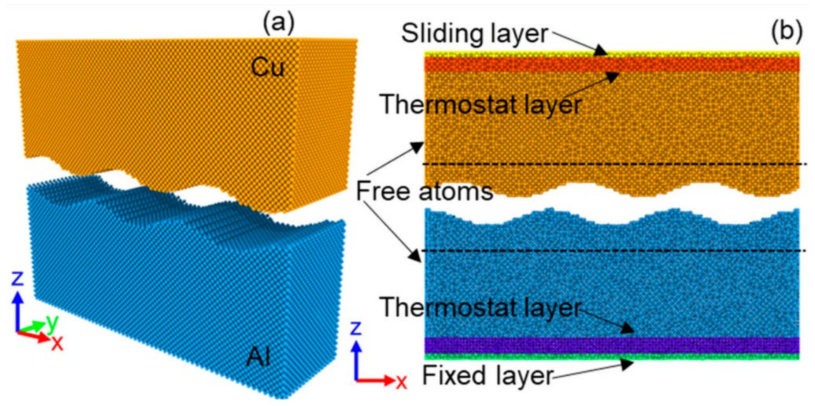
$$v_i \leftarrow a v_i \quad \text{with } a = \sqrt{T_D/T(t)}$$

$$T(t) = \frac{2}{3(N-1)k_B}\left(\sum_{i=1}^{N}\frac{1}{2}m_i v_i^2\right)$$

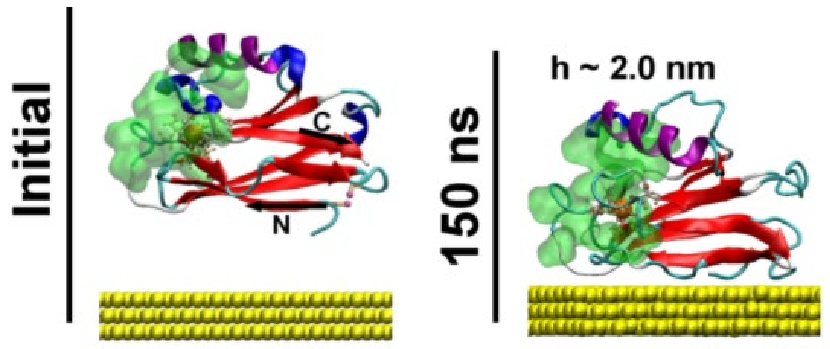
MD flowchart


Choose initial configuration $\{r_i(0)\}$
Energy minimization
Initial velocity assignment $\{v_i(0)\}$
Equilibration at desired temperature
$t = 0$
Compute forces
Update velocities and coordinates
$t \to t + \Delta t$
Control temperature
Calculate physical quantities
Is stopping criterion reached ?
no
yes
End

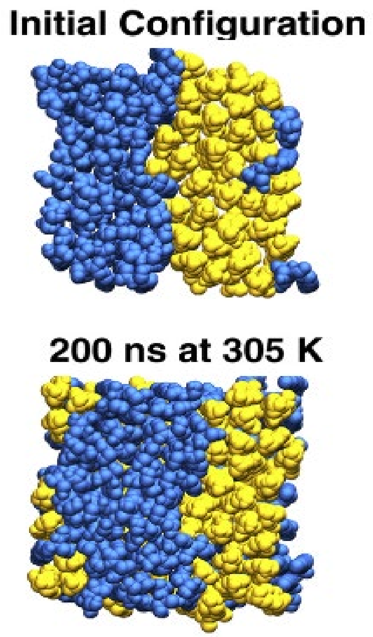
- **Initial configuration**:
  X-ray, NMR, lattices, random distribution, …



[Yang et al Materials 12 (2019) 1240]



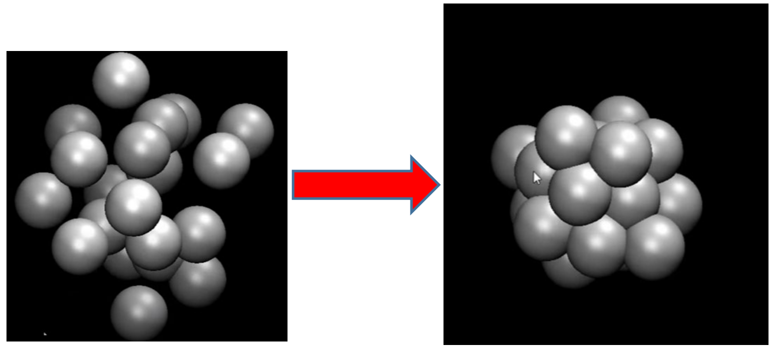
[Ortega et al Biomol. 9 (2019) 611]

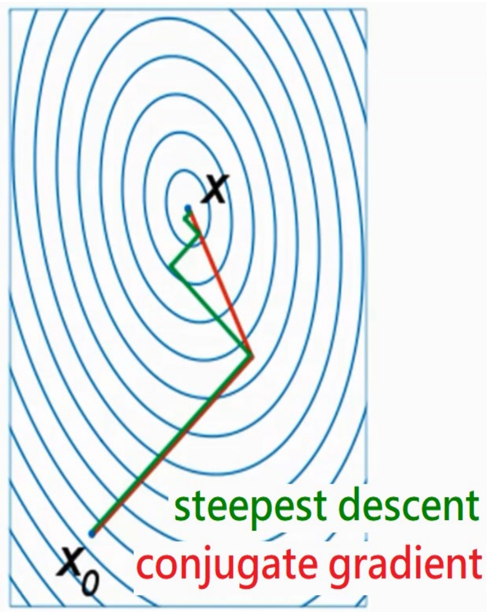


[Moore et al Biophys J. 114 (2018) 113]

- **Energy minimization** (stress relaxation):
  steepest descent, conjugate gradient, Newton-Raphson,...

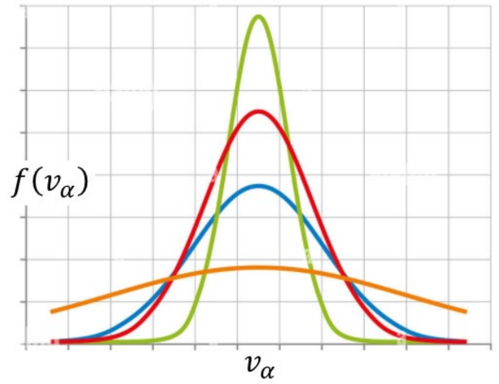
+ not care about velocity
+ Force is computed and
  atoms move in the direction of force
+ molecular dynamics at T=0





- Assignment of initial velocities

$\Rightarrow$ set the velocity according to Boltzmann distribution



$$f(v_{i\alpha}) = \sqrt{\frac{m_i}{2\pi k_B T}} \exp\left(-\frac{m_i v_{i\alpha}^2}{2k_B T}\right), \quad \alpha = x, y, z$$

+ Shift the velocities to produce zero total linear momentum: $v_{i\alpha} \leftarrow v_{i\alpha} - \left(\frac{1}{N}\sum_{j=1}^{N} v_{j\alpha}\right)$

+ Scale to the desired temperature: $v_{i\alpha} \leftarrow v_{i\alpha}\sqrt{T_D/(\frac{2}{3(N-1)k_B}\mathrm{KE})}$

How to generate velocities with a Gaussian distribution?

Box-Muller method:

let U_1 and U_2 be indep. random variables uniformly distributed on the interval $(0,1)$

$\Rightarrow$ $Z_1 = \sqrt{-\ln U_1}\cos(2\pi U_2)$

$Z_2 = \sqrt{-\ln U_1}\sin(2\pi U_2)$ indep. random variables with standard normal distribution

- Set $v_{i\alpha} = \sqrt{k_B T/m_i}\; Z_{1,2}$

Equilibration

- Measurements: taken after the system has been equilibrated
- Justification for equilibration:

  + any kind of energy: potential & kinetic energy, total energy,…

  + molecular structures, conformations, order parameters, …

  + any kind of distributions

  + properties of thermodynamics

  + fluctuations, etc.

- Different physical quantities have different equilibration time.

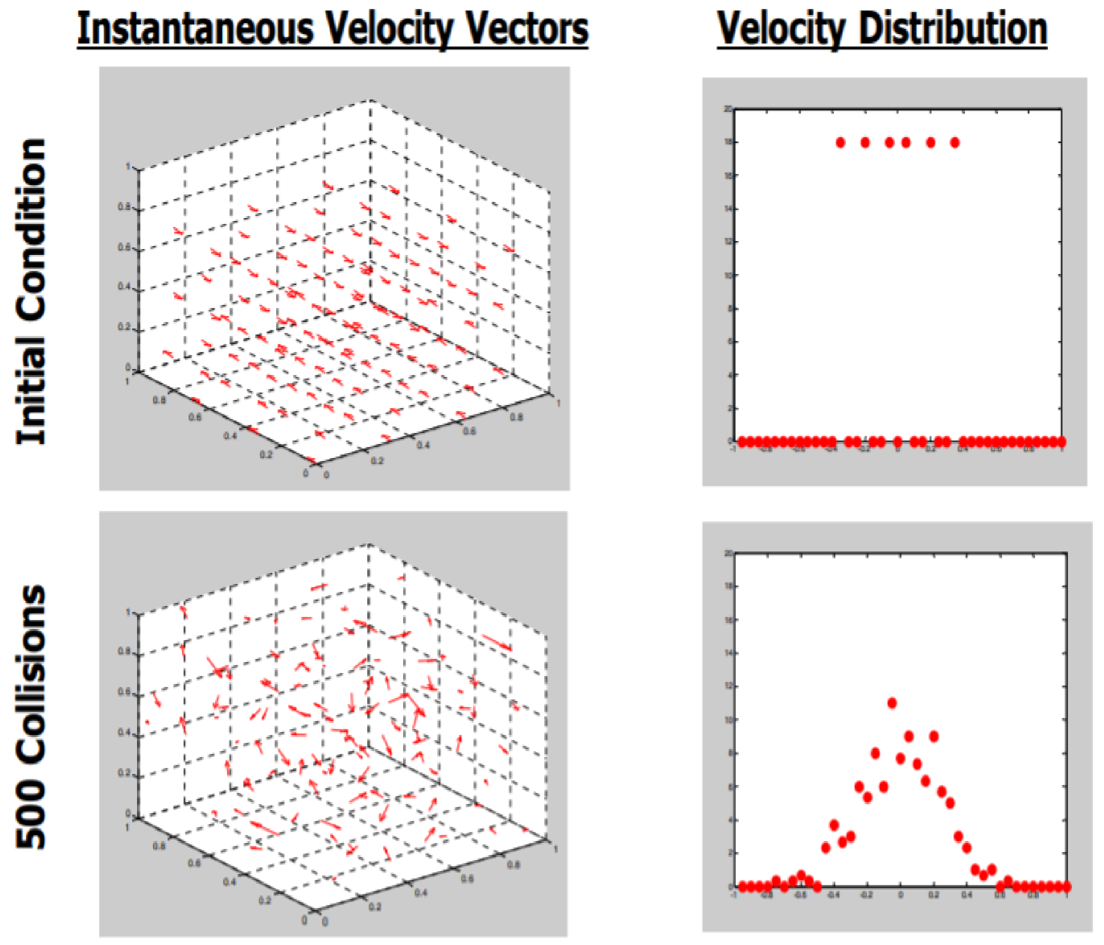
$\Rightarrow$ Not possible to have 100% sure for reaching of equilibrium

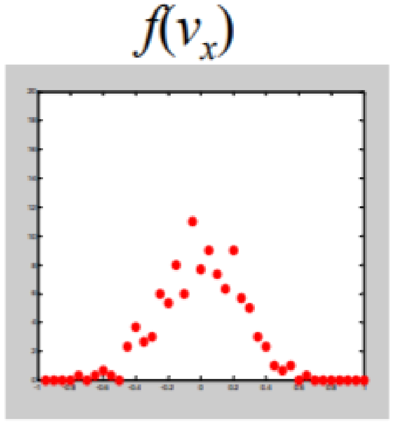
Boltzmann H- function: $H(f) = \int_{\Omega} f(x,v) \ln f(x,v)\, dv dx$

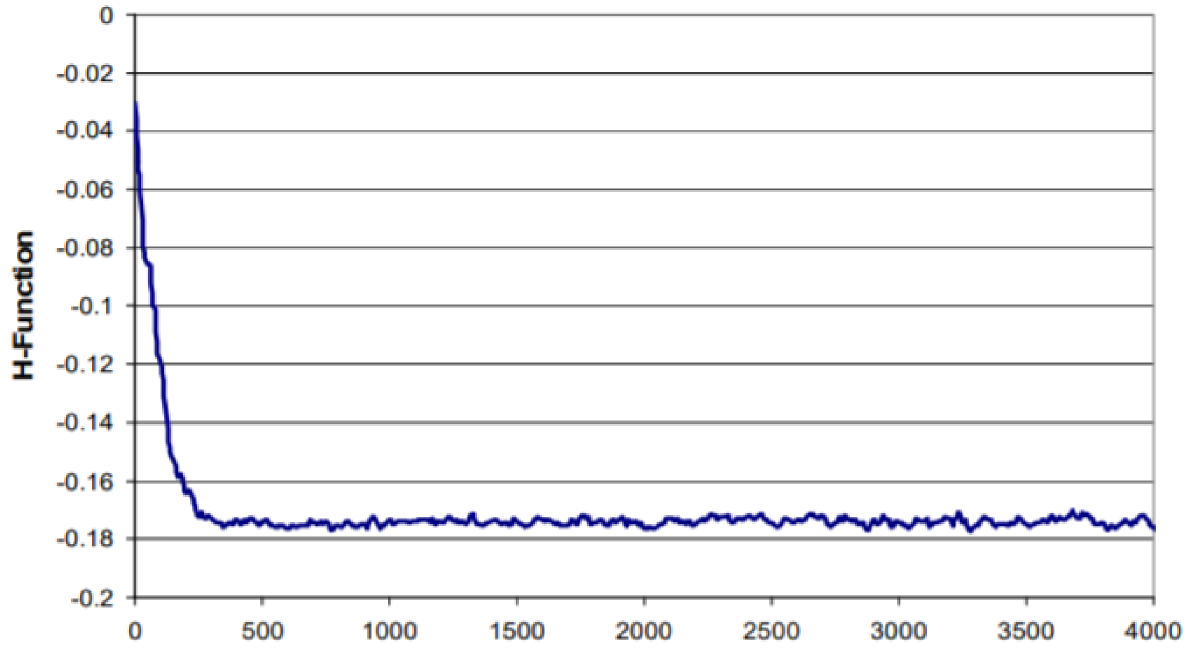
- A decreasing function with time until reaching a minimum
- Shannon-Boltzmann Entropy: $S(f) = -H(f)$

$$H_x = \sum_{\Delta v_x} f(v_x) \ln f(v_x) \Delta v_x$$

$$H = \tfrac{1}{3}\left(H_x + H_y + H_z\right)$$







[Martini’s lecture, Purdue University]

Algorithms

```
Program MD {
    initialization: r, v, minimization,…

    t = 0
    call  force(F)
    while  t <= tmax;    do
      call vel_Verlet(r, v, F)
      t = t + delt
      call quantities(r,v) if it's the moment
      call controlT(v)      if it's necessary
    done
}
```

O(N) algorithm

```
Subroutine vel_Verlet(r, v, F) {
    for i = 1 to N; do
       v[i] = v[i] + (F[i]/m[i])*delt/2
       r[i] = r[i] + v[i]*delt
    done
    call force(F);
    for i = 1 to N;    do
       v[i] = v[i] + (F[i]/m[i])*delt/2
    done
}
```

```
Subroutine controlT(v) {
    KE=0
    for i = 1 to N; do
       KE = KE + m[i]*v[i]*v[i] /2
    done
    T = 2/(3*(N-1)) * KE
    rho = sqrt(Td/T)
    for i = 1 to N; do
       v[i] = rho* v[i]
    done
}
```

```
Subroutine  force(F) {
    for i = 1 to N;    do
       F[i] = 0
    done
    for i = 1 to N-1;    do
       for j = i+1 to N;   do
          compute:  fij
          F[i] = F[i] + fij
          F[j] = F[j] – fij
       done
    done
}
```

$O(N^2)$ complexity : not efficient

Non-bonded interaction: short-range vs. long-range

- **Short-range interaction**: $u(r) \sim 1/r^n$ with $n > 3$:

interaction energy can be calculated by introducing a cutoff distance r_c,
and then corrected later

$$U = \int_0^\infty u(r)\rho(r)4\pi r^2 dr = \int_0^{r_c} u(r)\rho(r)4\pi r^2 dr + U_{corr}$$

$$U_{corr} \cong \int_{r_c}^\infty u(r)\rho_0 4\pi r^2 dr = \frac{4\pi\rho_0}{3-n} r^{3-n}\Big|_{r_c}^\infty = \frac{4\pi\rho_0}{(n-3)r_c^{n-3}}$$ converge if $n > 3$

(ex) van der Waals interaction: $u(r) \sim -1/r^6$

- **Long-range interaction**: $u(r) \sim 1/r^n$ with $n \le 3$:

cannot calculate the interaction energy correctly by introducing a cutoff,
even if $r_c = \infty$

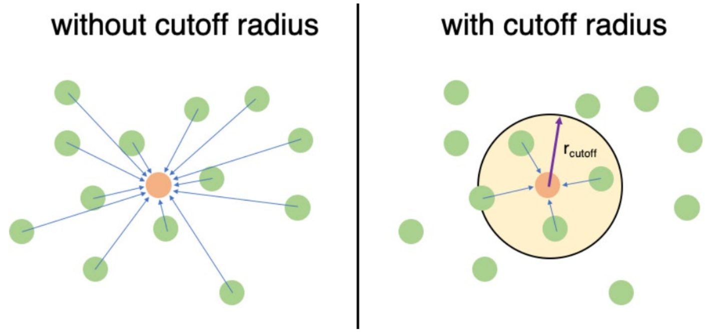
(ex) charge-charge interaction: $u(r) \sim 1/r$

charge-dipole interaction: $u(r) \sim 1/r^2$

diploe-dipole interaction: $u(r) \sim 1/r^3$

Neighbor list method for short-range interaction: O(N^2) complexity $\rightarrow$ O(N) complexity !?

$$u(r) = \begin{cases} u(r), r \leq r_c \\ 0, \quad r > r_c \end{cases}$$



```
Subroutine  force(F) {
   for i = 1 to N;    do
      F[i] = 0
   done
   for i = 1 to N;    do
      for j∈Neighbor(i) ;   do
         compute:  fij
         F[i] = F[i] + fij
      done
   done
}
```

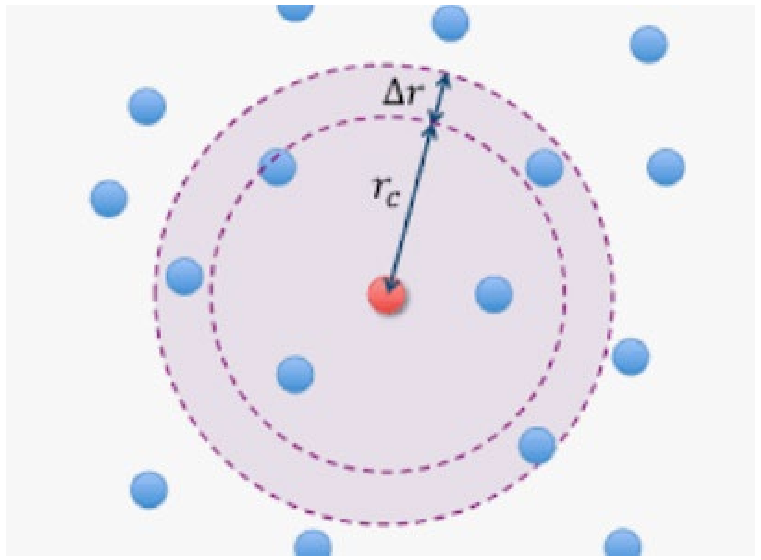
No. of neighbors $\approx \rho_0 \left(\frac{4}{3}\pi r_c^3\right)$

(about a constant, indep. of N)

- BUT constructing a **neighbor list** is an O(N^2) algorithm !

```
for i = 1 to N-1;    do
    for j = i+1 to N ;   do
        if rij<rc  then
            put j in Neighbor(i)
            put i in Nieghbor(j)
    done
done
```

- **Verlet's solution**: constructing a "larger" neighbor list at the place so that the list is "valid" for several time steps.

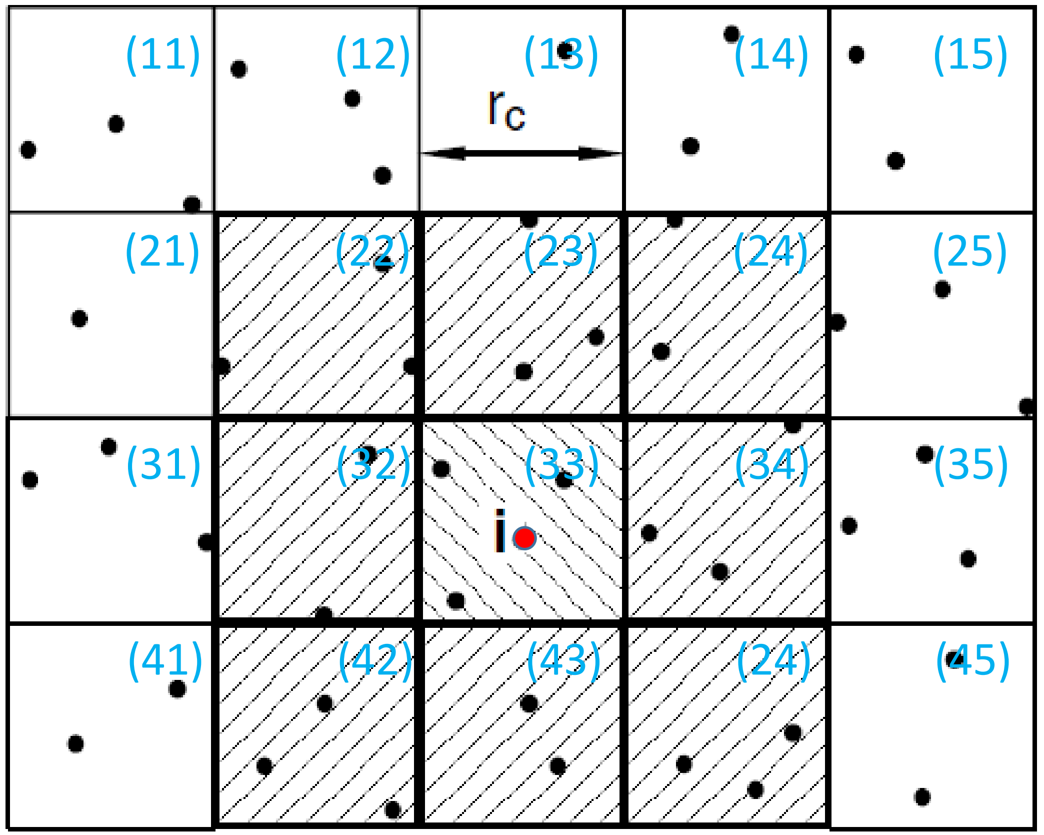


Cell list method: O(N) algorithm

- Simulation box is divided into cells of size $r_c \times r_c \times r_c$;

A particle interacts with those particles in the same cell and in the neighbor cells

```
Subroutine  force(F) {
   for i = 1 to N;    do
      F[i] = 0
   done
   for i = 1 to N;    do
       determine the cell no (a,b,c) which contains i
       for j∈ Cell(a,b,c)  and neighborCells ;   do
          compute:  fij
          F[i] = F[i] + fij
        done
   done
}
```



No. of partilces $\approx \rho_0(27r_c^3)$
(about a constant, indep. of N)

- Constructing **"cell list"**: also an O(N)-algorithm

```
for i = 1 to N;    do
       link i to a cell (a,b,c)
done
```

NOTE: number of particles j involving in the calculation of force:
(cell list method) >> (neighbor list method)

$$\rho_0(27r_c^3) \gg \rho_0\left(\frac{4}{3}\pi(r_c + r_s)^3\right)$$

Hybrid method: cell-list ⊕ neighbor-list

- Idea: using cell list to construct neighbor list and then using neighbor list to calculate forces : a pure O(N) algorithm

```
Subroutine ConstructVerletNeighborList
(I) make cell list:
    for i = 1 to N;   do
          link i to a cell (a,b,c)
    done

(II) build up neighbor list:
for i = 1 to N;   do
     determine the cell no (a,b,c) which contains i
     for j ∈ Cell(a,b,c) and neighborCells;   do
        if rij<(rc+rs)  then
            put j in Neighbor(i)
            put i in Nieghbor(j)
      done
  done
```

```
Subroutine  force(F) {

  call ConstructVerletNeighborList if it is not valid

  for i = 1 to N;   do
     F[i] = 0
  done

  for i = 1 to N;   do
     for j∈Neighbor(i) ;   do
        compute:  fij
        F[i] = F[i] + fij
      done
  done
}
```

Complicated in coding but more efficient as N is large!

How to calculate long-range interaction?

- Coulomb interaction under pbc:

$$U_{coul} = \frac{1}{2}\sum_n' \sum_i \sum_j \frac{q_i q_j}{4\pi\varepsilon|r_{ij}+nL|}$$

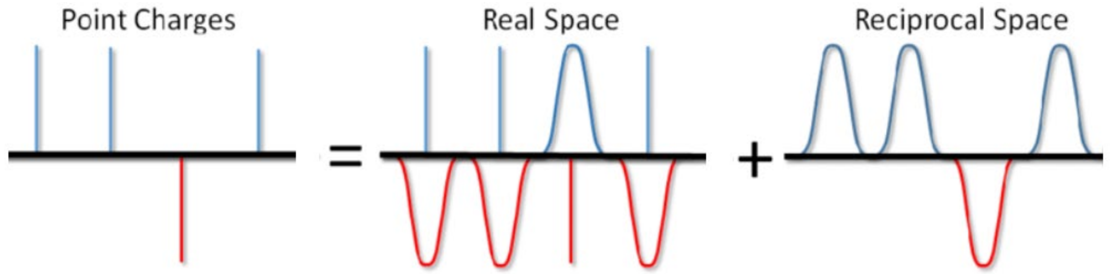
Gaussian law: $-\nabla^2\phi(r) = \frac{\rho(r)}{\varepsilon}$

$\phi_p(r) = \frac{q}{4\pi\varepsilon|r|} \quad \leftrightarrow \quad \rho_p(r) = q\delta(r)$

$\phi_G(r) = \frac{q\,\mathrm{erf}(\sqrt{\alpha}|r|)}{4\pi\varepsilon|r|} \quad \leftrightarrow \quad \rho_G(r) = q\left(\frac{\alpha}{\pi}\right)^{3/2}\exp(-\alpha r^2)$

$$\Rightarrow U_{coul} = \frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N}\frac{q_i q_j\,\mathrm{erfc}(\sqrt{\alpha}r)}{4\pi\varepsilon} + \frac{1}{2V}\sum_{k\neq 0}\frac{\exp(-k^2/4\pi)}{\varepsilon k^2}|\rho(k)|^2 - \sqrt{\frac{\alpha}{\pi}}\sum_{i=1}^{N}\frac{q_i^2}{4\pi\varepsilon}$$

where $\rho(k) = \sum_{i=1}^{N} q_i \exp(ik\cdot r_i)$



Ewald (1921): $\frac{1}{r} = \frac{1-\mathrm{erf}(\sqrt{\alpha}r)}{r} + \frac{\mathrm{erf}(\sqrt{\alpha}r)}{r}$

where

$\frac{\mathrm{erfc}(\sqrt{\alpha}r)}{r}$ is short-ranged,

can be computed with a cutoff r_c

$\frac{\mathrm{erf}(\sqrt{\alpha}r)}{r}$ is long-ranged,

computed with Fourier transform

Ewald sum:

cutoff in real space: r_c

cutoff in k-space : k_c

complexity: O(N)+O(N^2)

Ewald sum method:

choosing good parameters $\alpha, r_c, k_c \Rightarrow$ O($N^{1.5}$)

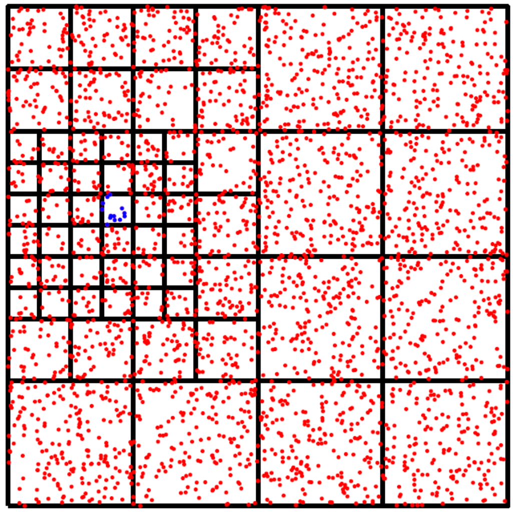
Incorporate Fast Fourier transform to solve discrete Poisson equation

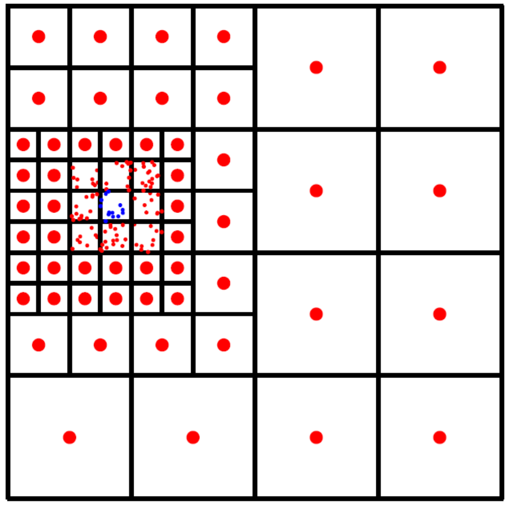
Particle Mesh Ewald $\Rightarrow$ O(N logN)

---

algorithm using multipole expansion to calculate electric field by grouping charges

Fast Multipole Method $\Rightarrow$ O(N)





Top 10 algorithms from the 20th century (SIAM News by B. Cipra)

- 1946: The Metropolis Algorithm
- 1947: Simplex Method
- 1950: Krylov Subspace Method
- 1951: The Decompositional Approach to Matrix Computations
- 1957: The Fortran Optimizing Compiler
- 1959: QR Algorithm
- 1962: Quicksort
- 1965: Fast Fourier Transform
- 1977: Integer Relation Detection
- 1987: Fast Multipole Method

Temperature controls

- **Velocity scaling method**: $v_i \leftarrow v_i\sqrt{\frac{T_D}{T}}, \quad T = \frac{2\mathrm{K}}{3(N-1)k_B}$ (too crude, very violent!)
- **Berendsen thermostat**: $v_i \leftarrow v_i\sqrt{1 + \frac{\Delta t}{\tau}\left(\frac{T_D}{T} - 1\right)}, \quad \tau$= relaxation time
- **Andersen thermostat**: every particle has certain probability at each time step to undergo a collision with a heat bath. If undergoing a collision, a new velocity drawn from the Boltzmann distribution corresponding to the desired temperature is assigned to the particle.

  $v_{ix,y,z} \leftarrow \sqrt{k_B T/m_i}Z$ if the particle i collides with the heat bath
- **Gaussian thermostat**: constraint method, isokinetic

$$\dot{r}_i = \frac{p_i}{m}, \quad \dot{p}_i = F_i - \alpha\frac{p_i}{m}$$

$$\frac{dT}{dt} = 0 = \frac{\sum_i \ v_i \cdot \dot{p}_i}{3(N-1)k_B} \quad \Rightarrow \quad \alpha = \frac{\sum_i \ F_i \cdot v_i}{\sum_i \ v_i \cdot v_i}$$

Nose-Hoover thermostat

- Nose's (1984) Extended Lagrangian: $L_{ext} = \sum_{i=1}^{N} \frac{1}{2} m_i \dot{\vec{r}}_i^{\,2} s^2 - U(\{\vec{r}_i\}) + \frac{1}{2} Q \dot{s}^2 - G(s)$

  momentum $\vec{\pi}_i = \frac{\partial L_{ext}}{\partial \dot{\vec{r}}_i} = m_i \dot{\vec{r}}_i s^2, \quad \pi_s = \frac{\partial L_{ext}}{\partial \dot{s}} = Q\dot{s}$

  extended Hamiltonian: $H_{ext} = \left[\sum_{i=1}^{N} \frac{\vec{\pi}_i^2}{2m_i s^2} + U(\{\vec{r}_i\})\right] + \frac{\pi_s^2}{2Q} + G(s) = H + \frac{\pi_s^2}{2Q} + G(s)$

  NVE ensemble for dynamic variables $(\{\vec{\pi}_i, \vec{r}_i\}, \pi_s, s)$: Partition function

$$Z_{ext} = \frac{1}{N!} \iint \iint d\pi_s ds d\vec{\pi}_i^N d\vec{r}_i^N \ \delta(H_{ext} - E)$$
$$= \frac{1}{N!} \iint \iint d\pi_s ds d\vec{p}_i^N d\vec{r}_i^N \ \delta\left(\left[\sum_{i=1}^{N} \frac{\dot{\vec{p}}_i^{\,2}}{2m_i} + U(\{\vec{r}_i\})\right] + \frac{\pi_s^2}{2Q} + G(s) - E\right), \quad \text{Here } \vec{p}_i = \vec{\pi}_i / s$$
$$= \frac{1}{N!} \iint d\vec{p}_i^N d\vec{r}_i^N \exp\left(-\frac{H}{k_B T}\right) \qquad \text{if we choose } G(s) = (3N+1) k_B T \ln s$$

  $\Rightarrow$ NVT ensemble for dynamic variables $(\{\vec{p}_i, \vec{r}_i\})$

- Hoover's modification (1985): non-Hamiltonian

$$H_{NH}(\{\vec{p}_i, \vec{r}_i\}, p_s, s) = \sum_{i=1}^{N} \frac{\vec{p}_i^2}{2m_i} + U(\{\vec{r}_i\}) + \frac{p_s^2 s^2}{2Q} + 3N k_B T \ln s$$

MKT form: Nose-Hoover Hamiltonian (1992 Martyna, Klein, Tuckerman)

- $H\left(\{\vec{p}_i,\vec{r}_i\},p_\eta,\eta\right)=\sum_{i=1}^{N}\frac{\vec{p}_i^2}{2m_i}+U(\{\vec{r}_i\})+\frac{p_\eta^2}{2Q}+Lk_BT\eta,\quad L=3N$, non-Hamiltonian

  Eq. of motion:
$$\frac{d\vec{p}_i}{dt}=-\frac{\partial U}{\partial\vec{r}_i}-\frac{p_\eta}{Q}\vec{p}_i$$
$$\frac{d\vec{r}_i}{dt}=\frac{\vec{p}_i}{m_i}$$
$$\frac{dp_\eta}{dt}=\left(\sum_{i=1}^{N}\frac{\vec{p}_i^2}{m_i}\right)-Lk_BT$$
$$\frac{d\eta}{dt}=\frac{p_\eta}{Q}$$

(p_η,η) plays the role of thermostat; trajectory $\{\vec{p}_i,\vec{r}_i\}$ fulfills NVT ensemble

- Nose-Hoover chain: thermostat is controlled by M dynamic variables $\left\{p_{\eta_j},\eta_j\right\}_{j=1}^{M}$

$$H\left(\{\vec{p}_i,\vec{r}_i\},\{p_{\eta_j},\eta_j\}\right)=\sum_{i=1}^{N}\frac{\vec{p}_i^2}{2m_i}+U(\{\vec{r}_i\})+\sum_{j=1}^{M}\frac{p_{\eta_j}^2}{2Q_j}+Lk_BT\eta_1+\sum_{j=2}^{M}k_BT\eta_j$$

Isothermal-isobaric method (MTK 1994) : NPT ensemble

- Equation of motion: $\frac{d\vec{r}_i}{dt} = \frac{\vec{p}_i}{m_i} + \frac{p_\epsilon}{W}\vec{r}_i$

$$\frac{d\vec{p}_i}{dt} = \vec{F}_i - \frac{p_\eta}{Q}\vec{p}_i - \frac{p_\epsilon}{W}\vec{p}_i(1+\frac{3}{L})$$

$$\frac{dV}{dt} = \frac{3V}{W}p_\epsilon$$

$$\frac{dp_\epsilon}{dt} = 3V(P_{int} - P) + \left(\frac{3}{L}\sum_{i=1}^{N}\frac{\vec{p}_i^2}{m_i}\right) - \frac{p_\eta}{Q}p_\epsilon$$

$$\frac{d\eta}{dt} = \frac{p_\eta}{Q}$$

$$\frac{dp_\eta}{dt} = \left(\sum_{i=1}^{N}\frac{\vec{p}_i^2}{m_i}\right) + \frac{p_\epsilon^2}{W} - (L+1)k_BT$$

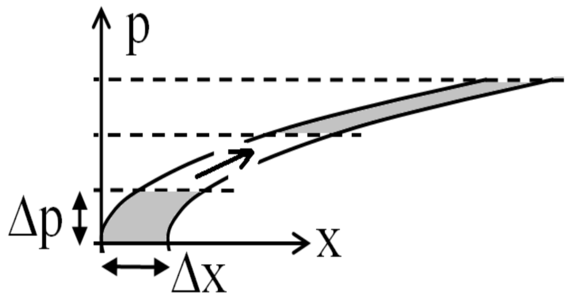
where $P_{int} = \frac{1}{3V}(\sum_{i=1}^{N}\frac{\vec{p}_i^2}{m_i} + \sum_{i=1}^{N}\vec{r}_i \cdot \vec{F}_i - 3V\frac{\partial U(\{\vec{r}_i\},V)}{\partial V}), \quad L = 3N$

Conserved quantity: non-Hamiltonian

$$H\left(\{\vec{p}_i,\vec{r}_i\},p_\epsilon,V,p_\eta,\eta\right) = \sum_{i=1}^{N}\frac{\vec{p}_i^2}{2m_i} + U(\{\vec{r}_i\},V) + \frac{p_\epsilon^2}{2W} + PV + \frac{p_\eta^2}{2Q} + (L+1)k_BT\eta$$

Generic way to develop a time-reversible & symplectic integrator

- How to find a numerical scheme which solves the Hamiltonian equation $\dot{q} = \frac{\partial H}{\partial p}, \dot{p} = -\frac{\partial H}{\partial q}$ and also conserves the symplectic 2-form $dp \wedge dq$? [Liouville's phase volume invariance theorem]



Let $A(X)$ be any physical function defined in the phase space, $X = (q_1, q_2, \dots, p_1, p_2, \dots)$

Consider the dynamic equation $\frac{dA}{dt} = iLA$

where $iL = \sum_{\alpha=1}^{3N} \left[\frac{\partial H}{\partial p_\alpha} \frac{\partial}{\partial q_\alpha} - \frac{\partial H}{\partial q_\alpha} \frac{\partial}{\partial p_\alpha} \right] \equiv iL_q + iL_p$ is a Liouville operator

$\Rightarrow$ Solution $A(X_t) = e^{iLt} A(X_0)$ $\neq e^{iL_q t} e^{iL_p t} A(X_0)$ because not commutative $[iL_q, iL_p] \neq 0$

Applying Trotter theorem(1959): $[B, C] \neq 0,\ e^{B+C} = \lim_{P\to\infty} \left[e^{C/2P} e^{B/P} e^{C/2P} \right]^P$

We have $e^{iLt} = \left[e^{iL_p \Delta t/2} e^{iL_q \Delta t} e^{iL_p \Delta t/2} \right]^P$

or $e^{iLt} = \left[e^{iL_q \Delta t/2} e^{iL_p \Delta t} e^{iL_q \Delta t/2} \right]^P$ where $t = P\Delta t$

- $e^{iLt} = \left[e^{iL_p\Delta t/2}e^{iL_q\Delta t}e^{iL_p\Delta t/2}\right]^n = (e^{iL_p\Delta t/2}e^{iL_q\Delta t}e^{iL_p\Delta t/2}) \ldots (e^{iL_p\Delta t/2}e^{iL_q\Delta t}e^{iL_p\Delta t/2})$

$$e^{0.5iL_p\Delta t}A(q,p) = \exp\left(\frac{\Delta t}{2}\sum_\alpha \frac{-\partial H}{\partial q_\alpha}\frac{\partial}{\partial p_\alpha}\right)A(q,p) = \exp\left(\frac{\Delta t}{2}\sum_\alpha \dot{p}_\alpha \frac{\partial}{\partial p_\alpha}\right)A(q,p) = A(q,\ p+\frac{\Delta t}{2}\dot{p})$$

$$e^{iL_q\Delta t}A(q,p) = \exp\left(\Delta t\sum_\alpha \frac{\partial H}{\partial p_\alpha}\frac{\partial}{\partial q_\alpha}\right)A(q,p) = \exp\left(\Delta t\sum_\alpha \dot{q}_\alpha \frac{\partial}{\partial q_\alpha}\right)A(q,p) = A(q+\dot{q}\Delta t,\ p)$$

$$e^{0.5iL_p\Delta t}A(q,p) = \exp\left(\frac{\Delta t}{2}\sum_\alpha \frac{-\partial H}{\partial q_\alpha}\frac{\partial}{\partial p_\alpha}\right)A(q,p) = \exp\left(\frac{\Delta t}{2}\sum_\alpha \dot{p}_\alpha \frac{\partial}{\partial p_\alpha}\right)A(q,p) = A(q,\ p+\frac{\Delta t}{2}\dot{p})$$

that is "**velocity Verlet algorithm**":

$$v \leftarrow v + \frac{F}{m}\frac{\Delta t}{2}$$
$$r \leftarrow r + v\Delta t$$
$$v \leftarrow v + \frac{F}{m}\frac{\Delta t}{2}$$

Similarly, $e^{iLt} = \left[e^{iL_q\Delta t/2}e^{iL_p\Delta t}e^{iL_q\Delta t/2}\right]^n = (e^{iL_q\Delta t/2}e^{iL_p\Delta t}e^{iL_q\Delta t/2}) \ldots (e^{iL_q\Delta t/2}e^{iL_p\Delta t}e^{iL_q\Delta t/2})$

position Verlet algorithm:

$$r \leftarrow r + v\frac{\Delta t}{2}$$
$$v \leftarrow v + \frac{F}{m}\Delta t$$
$$r \leftarrow r + v\frac{\Delta t}{2}$$

Application (1): multiple time-scale integration

- Eq of motion: $\dot{r} = \frac{p}{m}, \qquad \dot{p} = F_{\text{fast}} + F_{\text{slow}}$ $\qquad [\ \omega_{\text{fast}} > \omega_{\text{slow}}\ ]$

  fast force F_{fast}, such as bonded interaction: small δt is needed. $\qquad [\ \omega_{\text{fast}}\delta t < 1\ ]$

  slow force F_{slow}, like non-bonded interaction: $\Delta t > \delta t$, say $\Delta t = n\delta t$ $\qquad [\omega_{\text{slow}}\Delta t < 1]$

- Liouville operator $iL = iL_r + iL_{p_f} + iL_{p_s} = \frac{p}{m}\frac{\partial}{\partial r} + F_{\text{fast}}\frac{\partial}{\partial p} + F_{\text{slow}}\frac{\partial}{\partial p}$

- Trotter scheme: $\exp(iL\Delta t) = \exp\left(iL_{p_s}\Delta t/2\right)\exp\left((iL_{\mathrm{r}} + iL_{p_f})\Delta t\right)\exp\left(iL_{p_s}\Delta t/2\right)$

$$\Rightarrow \exp(iL\Delta t) = \exp\left(\frac{\Delta t}{2}F_{\text{slow}}\frac{\partial}{\partial p}\right)\left[\exp\left(\frac{\delta t}{2}F_{\text{fast}}\frac{\partial}{\partial p}\right)\exp\left(\delta t\frac{p}{m}\frac{\partial}{\partial r}\right)\exp\left(\frac{\delta t}{2}F_{\text{fast}}\frac{\partial}{\partial p}\right)\right]^n \exp\left(\frac{\Delta t}{2}F_{\text{slow}}\frac{\partial}{\partial p}\right)$$

```
p = p + 0.5*Δt*F_slow
for i = 1 to n
    p = p + 0.5*δt*F_fast
    r = r + δt*p/m
    recalculate fast force
    p = p + 0.5*δt*F_fast
endfor
recalculate slow force
p = p + 0.5*Δt*F_slow
```

$\longleftarrow$

RESPA algorithm

$\Delta t = n\delta t$

Two or more time scales

Application (2): Symplectic integrator for Nose-Hoover thermostat

Eq. of motion:

$$\frac{d\vec{p}_i}{dt} = \vec{F}_i - \frac{p_\eta}{Q}\vec{p}_i$$

$$\frac{d\vec{r}_i}{dt} = \frac{\vec{p}_i}{m_i}$$

$$\frac{dp_\eta}{dt} = \left(\sum_{i=1}^{N}\frac{\vec{p}_i^2}{m_i}\right) - Lk_BT$$

$$\frac{d\eta}{dt} = \frac{p_\eta}{Q}$$

- $iL = \frac{d}{dt} = iL_r + iL_v + iL_{NHT}$

where $iL_r = \sum_i \ \vec{v}_i \frac{\partial}{\partial \vec{r}_i}$

$$iL_v = \sum_i \ \frac{\vec{F}_i}{m_i}\frac{\partial}{\partial \vec{v}_i}$$

$$iL_{NHT} = iL_\eta + iL_{vv} + iL_{v_\eta}$$

with

$$iL_\eta = v_\eta \frac{\partial}{\partial \eta}$$

$$iL_{vv} = -\sum_i \ v_\eta \vec{v}_i \frac{\partial}{\partial \vec{v}_i}$$

$$iL_{v_\eta} = G\frac{\partial}{\partial v_\eta}$$

$$G = \frac{1}{Q}\left[\sum_{i=1}^{N}\frac{\vec{p}_i^2}{m_i} - Lk_BT\right]$$

Trotter factorization:

$$\exp(iL\Delta t) = \exp\left(iL_{NHT}\frac{\Delta t}{2}\right)\exp((iL_\mathrm{r} + iL_v)\Delta t)\exp\left(iL_{NHT}\frac{\Delta t}{2}\right)$$

where

$$\exp\left(iL_{NHT}\frac{\Delta t}{2}\right) = \exp\left(iL_{v_\eta}\frac{\Delta t}{4}\right)\exp\left(iL_{vv}\frac{\Delta t}{2}\right)\exp\left(iL_\eta\frac{\Delta t}{2}\right)\exp\left(iL_{v_\eta}\frac{\Delta t}{4}\right)$$

$$\exp((iL_\mathrm{r} + iL_v)\Delta t) = \exp\left(iL_r\frac{\Delta t}{2}\right)\exp(iL_v\Delta t)\exp\left(iL_r\frac{\Delta t}{2}\right)$$

$$e^{iL_{v_\eta}\Delta t/4}: v_\eta \leftarrow v_\eta + G\frac{\Delta t}{4}$$

$$e^{iL_{v_\eta}\Delta t/2}: \eta \leftarrow \eta + v_\eta\frac{\Delta t}{2}$$

$$e^{iL_{vv}\Delta t/2}: \vec{v}_i \leftarrow \vec{v}_i e^{-v_\eta\Delta t/2}$$

$$e^{iL_r\Delta t/2}: \vec{r}_i \leftarrow \vec{r}_i + \vec{v}_i\frac{\Delta t}{2}$$

$$e^{iL_v\Delta t}: \vec{v}_i \leftarrow \vec{v}_i + \frac{\vec{F}_i}{m_i}\Delta t$$

Algorithm

$$\exp\left(iL_{NHT}\frac{\Delta t}{2}\right) = \exp\left(iL_{v_\eta}\frac{\Delta t}{4}\right)\exp\left(iL_{vv}\frac{\Delta t}{2}\right)\exp\left(iL_{\eta}\frac{\Delta t}{2}\right)\exp\left(iL_{v_\eta}\frac{\Delta t}{4}\right)$$

$$\exp((iL_r + iL_v)\Delta t) = \exp\left(iL_r\frac{\Delta t}{2}\right)\exp(iL_v\Delta t)\exp\left(iL_r\frac{\Delta t}{2}\right)$$

```
Subroutine integrate {
    call NHT(KE)
    call posi_Verlet(KE)
    call NHT(KE)
}
```

```
Subroutine NHT(KE) {
    G=(2*KE - L*T)/Q
    v_eta = v_eta  + G*dt/4
    eta = eta + v_eta*dt/2
    s = exp( - v_eta*dt/2 )
    for i = 1 to N; do
        v[i] = v[i]*s
    done
    KE = KE *s*s
    G=(2*KE - L*T)/Q
    v_eta = v_eta  + G*dt/4
}
```

```
Subroutine posi_Verlet (KE) {
    KE = 0
    for i = 1 to N;  do
        r[i] = r[i] + v[i]*dt/2
    done
    call Force(r)
    for i = 1 to N;  do
        v[i] = v[i] + (f[i]/m[i])*dt
        KE  =  KE + 0.5*m[i]*v[i]*v[i]
    done
    for i = 1 to N;  do
        r[i] = r[i] + v[i]*dt/2
    done
}
```

$$e^{iL_{v_\eta}\Delta t/4}: v_\eta \leftarrow v_\eta + G\frac{\Delta t}{4}$$

$$e^{iL_{v_\eta}\Delta t/2}: \eta \leftarrow \eta + v_\eta\frac{\Delta t}{2}$$

$$e^{iL_{vv}\Delta t/2}: \vec{v}_i \leftarrow \vec{v}_i e^{-v_\eta\Delta t/2}$$

$$e^{iL_r\Delta t/2}: \vec{r}_i \leftarrow \vec{r}_i + \vec{v}_i\frac{\Delta t}{2}$$

$$e^{iL_v\Delta t}: \vec{v}_i \leftarrow \vec{v}_i + \frac{\vec{F}_i}{m_i}\Delta t$$

Calculation of static quantities

- **Root Mean Square Deviation** : $\mathrm{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\vec{r}_{iA} - \vec{r}_{iB})^2}$

  RMSD is calculated between two sets of atomic coordinates.
  It is a measure of how much the protein conformation has changed.

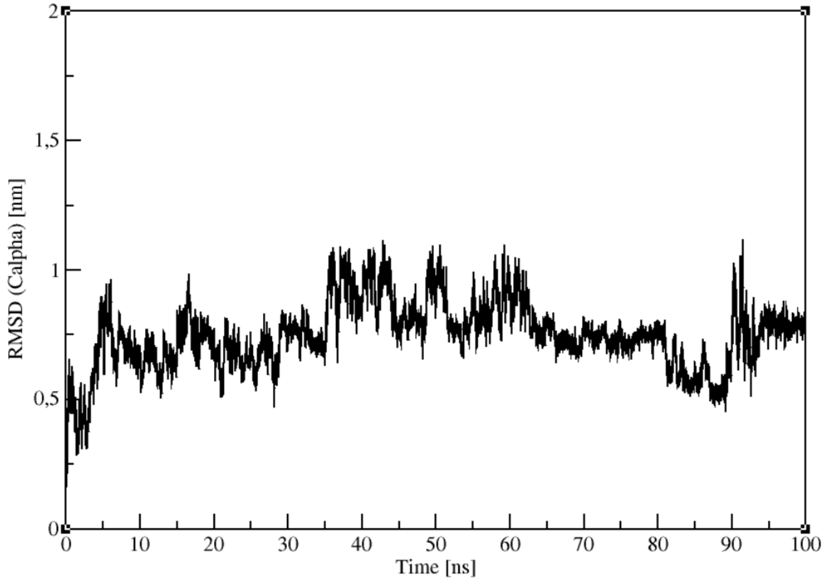
- **Root Mean Square Fluctuation**: $\mathrm{RMSF}_i = \sqrt{\frac{1}{T}\sum_{t}(\vec{r}_i(t) - \langle\vec{r}_i(t)\rangle)^2}$

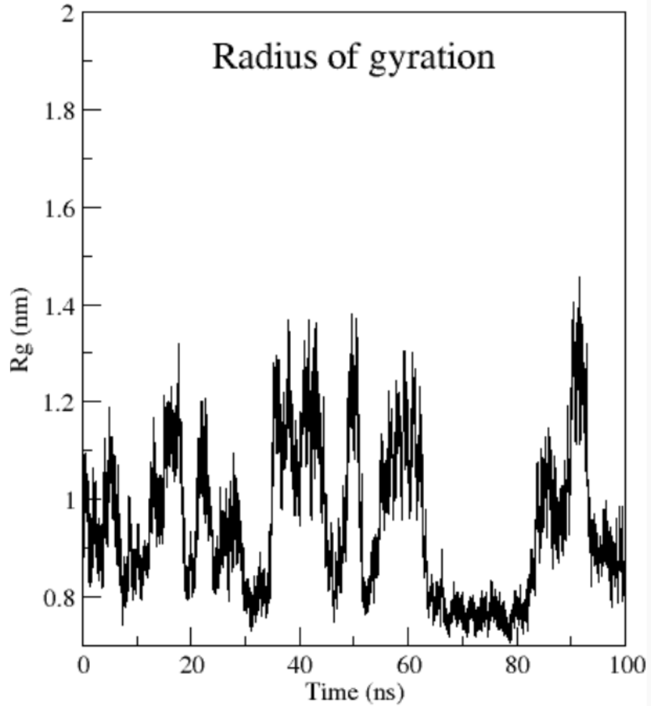
  It measures the average deviation of a particle over time from a reference position (typically the time-averaged position of the particle).

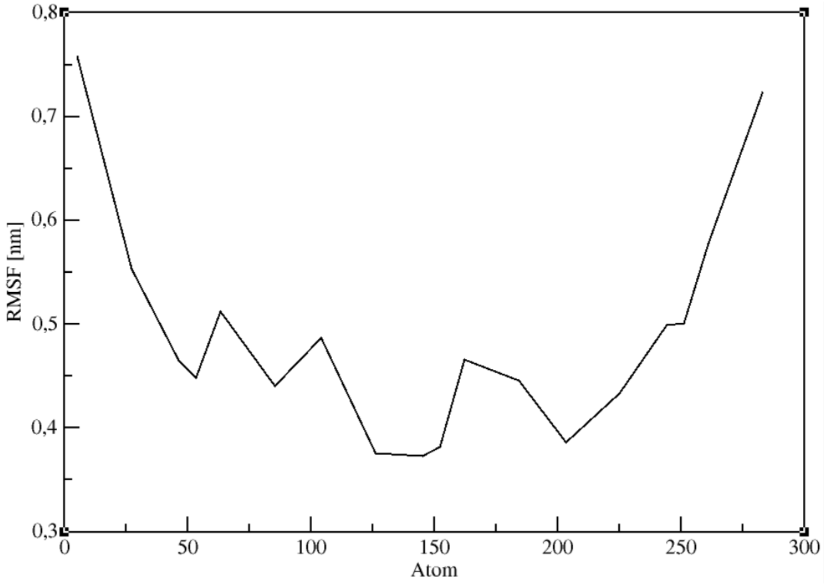
- **Radius of Gyration**:

  $\mathrm{R_g} = \sqrt{\sum_{i=1}^{N} m_i(\vec{r}_i - \vec{r}_{cm})^2 \,/\, \sum_{i=1}^{N} m_i}$

  a measure for the compactness of a structure







http://www.strodel.info/index_files/lecture/html/tutorial.html

Calculation of dynamic quantities

- **Green-Kubo formula**: $\frac{d}{dt}\langle \Delta A(t)^2\rangle = 2\int_0^t \langle \dot{A}(\tau)\dot{A}(0)\rangle d\tau$

  Diffusion coefficient D: $\langle \Delta r(t)^2\rangle = 6Dt$ as $t \to \infty$

$$D = \frac{1}{6}\frac{d}{dt}\sum_\alpha \left\langle \left(r_\alpha(t) - r_\alpha(0)\right)^2 \right\rangle, \quad t \to \infty \text{, Einstein relation}$$

$$\Rightarrow D = \frac{1}{3}\int_0^\infty \langle \vec{v}(\tau)\cdot\vec{v}(0)\rangle d\tau = \frac{1}{3}\int_0^\infty \frac{1}{N}\sum_{i=1}^N \langle \vec{v}_i(\tau)\cdot\vec{v}_i(0)\rangle\, d\tau$$

D is a time integral of the velocity correlation function

Shear viscosity: $\eta = \frac{V}{20k_BT}\frac{d}{dt}\sum_{\alpha\beta}\left\langle \left(G_{\alpha\beta}(t) - G_{\alpha\beta}(0)\right)^2 \right\rangle, t \to \infty$

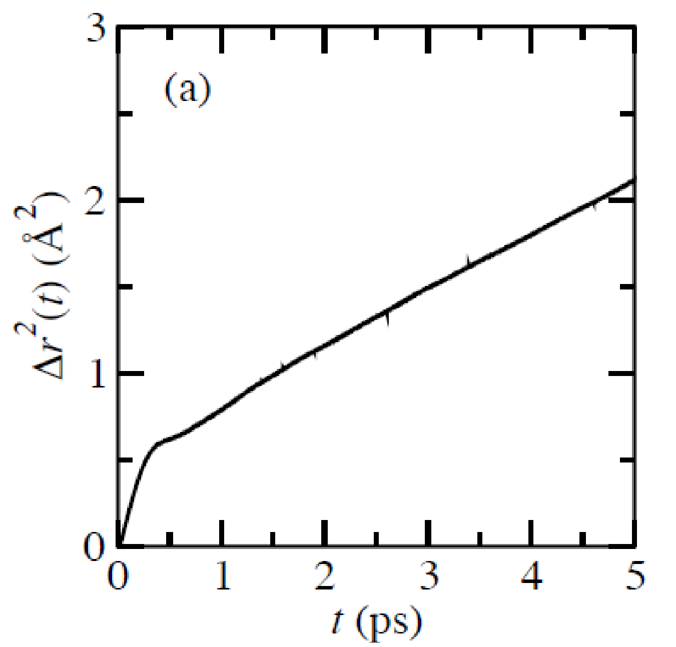
where $G_{\alpha\beta}(t) = \int_0^t P_{\alpha\beta}(t')dt'$

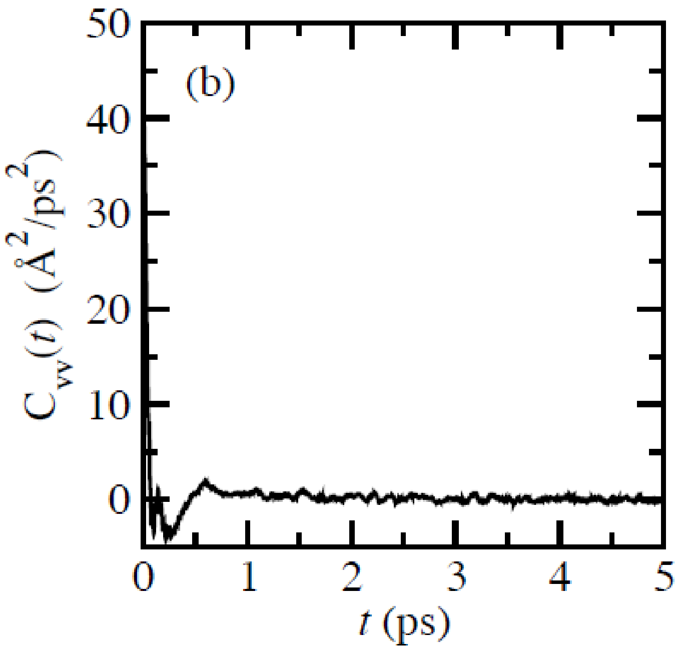
Pressure tensor: $P_{\alpha\beta} = \omega_{\alpha\beta}\left(\frac{\sigma_{\alpha\beta}+\sigma_{\beta\alpha}}{2} - \frac{\delta_{\alpha\beta}}{3}\mathrm{tr}(\sigma)\right),$

$$\omega_{\alpha\beta} = 1 \text{ if } \alpha \neq \beta, \omega_{\alpha\beta} = \frac{4}{3} \text{ if } \alpha = \beta$$

Stress tensor: $\sigma_{\alpha\beta} = \frac{1}{V}\left[\sum_{i=1}^N m v_{i\alpha}v_{j\beta} + \sum_{i=1}^N\sum_{j=i+1}^N r_{ij\alpha}F_{ij\beta}\right]$

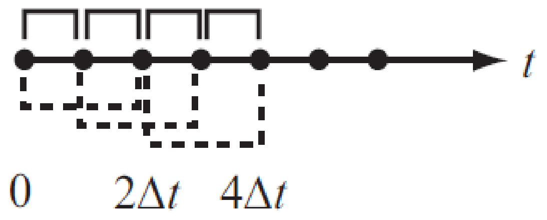
Green-Kubo: $\eta = \frac{V}{10k_BT}\int_0^\infty \sum_{\alpha\beta}\langle P_{\alpha\beta}(\tau)P_{\alpha\beta}(0)\rangle d\tau$





Tuckerman's book

Calculating time correlation function from a trajectory



- Direct method: O(M^2) complexity

$$C_{AB}(t) = \frac{1}{T}\int_0^T a(X_\tau)b(X_{\tau+t})d\tau$$

$$\Rightarrow C_{AB}(n\Delta t) = \frac{1}{M-n}\sum_{m=1}^{M-n} a(X_{m\Delta t})b(X_{(m+\mathrm{n})\Delta t}), n = 1, \ldots, M, \Delta t = T/M$$

- Fast Fourier Transform method: O(M logM) (Wiener-Khinchin theorem)

$$C_{AB}(t) = \frac{1}{T}\int_{-T/2}^{T/2} a(X_\tau)b(X_{\tau+t})d\tau = \frac{1}{T}\int_{-\infty}^{\infty} \tilde{a}(\omega)\tilde{b}^*(\omega)e^{i\omega t}d\omega$$

using FFT to calculate $\tilde{a}_k$ and $\tilde{b}_k$, and then FFT^{-1} to compute $C_{AB}(n\Delta t)$

$$\tilde{a}_k = \sum_{n=0}^{M-1} a(X_{n\Delta t})e^{-i2\pi nk/M}$$

$$\tilde{b}_k = \sum_{n=0}^{M-1} b(X_{n\Delta t})e^{-i2\pi nk/M}$$

$$C_{AB}(n\Delta t) = \frac{1}{M}\sum_{k=0}^{M-1} \tilde{a}_k\tilde{b}_k^* e^{i2\pi nk/M} , n = 1, \ldots, M$$

Summary

- Fundamentals of molecular dynamics simulation have been reviewed.

  + integrating scheme to solve Eq. of motion: Verlet algorithm

  + force fields: class I, II, III

  + short range interaction: neighbor list $\oplus$ cell list

  + long range interaction: Ewald sum and other methods

  + temperature control & ensembles

  + generic way to develop symplectic integrator : Trotter facterization

  + calculations of static and dynamic properties, correlation functions

Acknowledgements to Ministry of Science and Technology.

Thank you for your attention!

MD simulation applied to macroscopic system: Using MD technique to solve Eq. of motion

(ex) Granular dynamics : Hertz-Mindlin model

[normal force firstly studied by Hertz 1881; tangential force firstly studied by Mindlin 1949]

$$\vec{F}_{ij} = \vec{F}_n + \vec{F}_t{}' = \sqrt{\delta R_*}\left[(k_n \delta \hat{n} - m_* \gamma_n \vec{v}_n) + (-k_t \Delta \vec{S}_t - m_* \gamma_t \vec{v}_t)\right]$$

where

k_n, k_t: elastic constant for normal, tangential force

γ_n, γ_t: damping constant for normal, tangential contact

$m_* = \frac{m_i m_j}{m_i + m_j}$, $R_* = \frac{R_i R_j}{R_i + R_j}$ reduced mass and radius

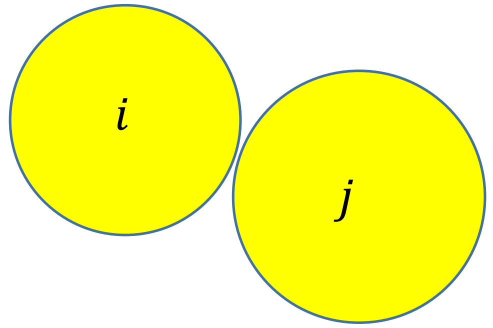
$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$, $r_{ij} = |\vec{r}_{ij}|$, $\hat{n} = \vec{r}_{ij}/r_{ij}$

$\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$, $\vec{v}_n = (\vec{v}_{ij} \cdot \hat{n})\hat{n}$,

$\vec{v}_t = \vec{v}_{ij} - \vec{v}_n - (\vec{\omega}_i R_i + \vec{\omega}_j R_j) \times \hat{n}$, $\hat{t} = \vec{v}_t / |\vec{v}_t|$

$\Delta \vec{S}_t$ = tangential displacement

$\delta = R_i + R_j - r_{ij}$ overlap distance



Coulomb criterion $\vec{F}_t$ is limited by $\mu_t|\vec{F}_n|$:

$$\vec{F}_t = -\min\{\ \mu_t |\vec{F}_n| \hat{t},\ \vec{F}_t'\ \}$$

$$\Rightarrow \vec{F}_{ij} = \vec{F}_{ij}^n + \vec{F}_{ij}^t$$

Algorithm: velocity Verlet

```
Subroutine vel_Verlet(r, v, F,  θ, ω, τ) {
   for i = 1 to N; do
      v[i] = v[i] + (F[i]/m[i])*delt/2
      ω[i] = ω[i] + (τ[i]/I[i])*delta/2

      r[i] = r[i] + v[i]*delt
   done

   call force_torque (F, τ);

   for i = 1 to N;    do
      v[i] = v[i] + (F[i]/m[i])*delt/2
     ω[i] = ω[i] + (τ[i]/I[i])*delta/2
   done
}
```

$$\frac{d(m\vec{v})}{dt} = \vec{F} \qquad \frac{d(I\vec{\omega})}{dt} = \vec{\tau}$$

$$\frac{d\vec{r}}{dt} = \vec{v}$$

$$\vec{F}_i = \sum_{j=1}^{N} \left(\vec{F}_{ij}^{n} + \vec{F}_{ij}^{t}\right)$$

$$\vec{\tau}_i = \sum_{j=1}^{N} \vec{\tau}_{ij} = \sum_{j=1}^{N} \left(R_i \hat{r}_{ij}\right) \times \vec{F}_{ij} = \sum_{j=1}^{N} \left(R_i \hat{r}_{ij}\right) \times \vec{F}_{ij}^{t}$$