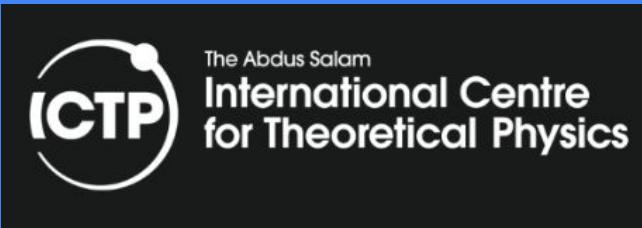


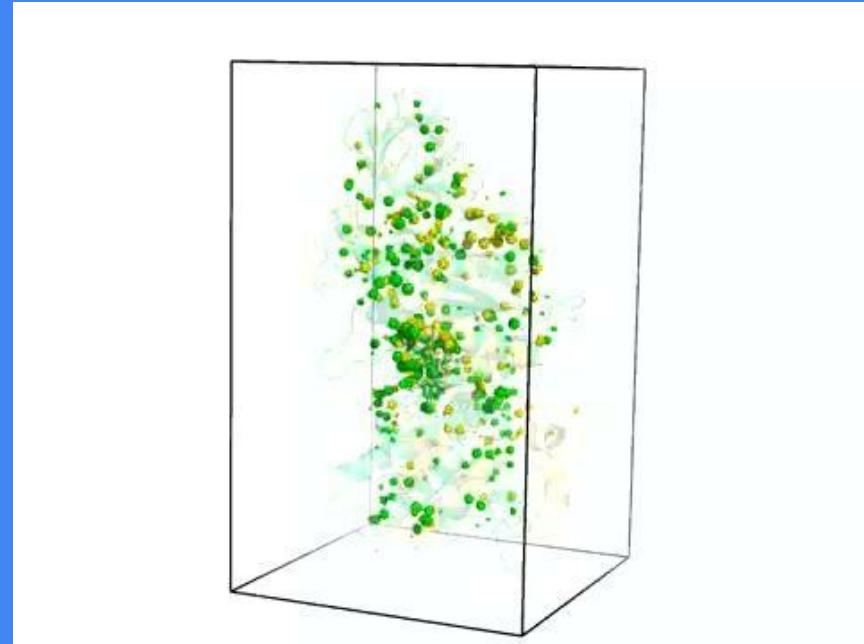
# Molecular Dynamics Study for the Iodide Symporter

Alfonso Leyva  
Pontificia Universidad Javeriana  
13.09.23



# Outline

- People to be grateful with
- Basics of X-Ray
- Basics of NIS
- Basics of Computer Simulations
- Preliminary Results & Future Perspectives



# Pontificia Universidad Javeriana - Structural Biophysics & Biochemistry Group



@JHU & @ Vanderbilt Mario Bianchet



@JHU & @ Vanderbilt Nancy Carrasco



[nature](#) > [articles](#) > [article](#)

Article | [Published: 14 December 2022](#)

## Structural insights into the mechanism of the sodium/iodide symporter

[Silvia Ravera](#), [Juan Pablo Nicola](#), [Glicella Salazar-De Simone](#), [Fred J. Sigworth](#), [Erkan Karakas](#), [L. Mario Amzel](#), [Mario A. Bianchet](#) & [Nancy Carrasco](#) 

[Nature](#) 612, 795–801 (2022) | [Cite this article](#)

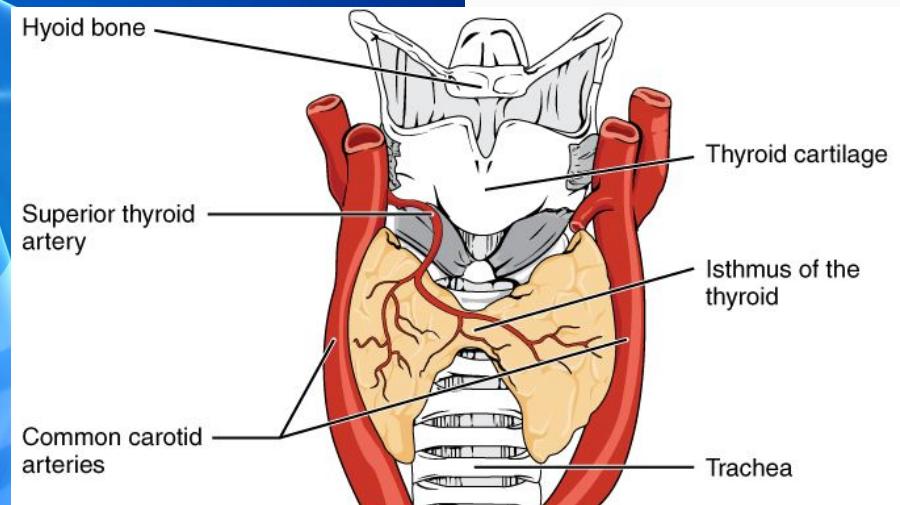
# Structural insights into the mechanism of the sodium/iodide symporter

## Abstract

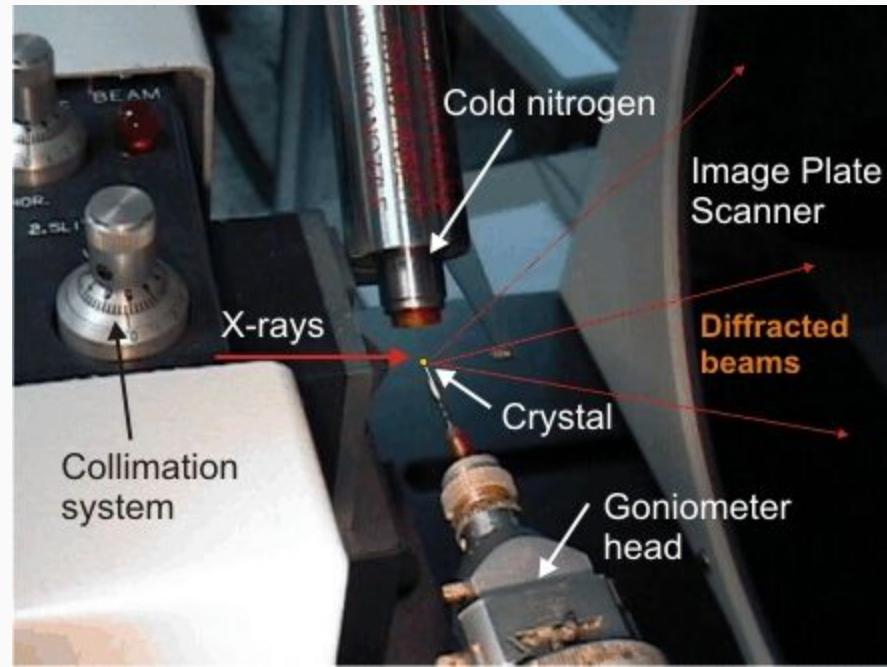
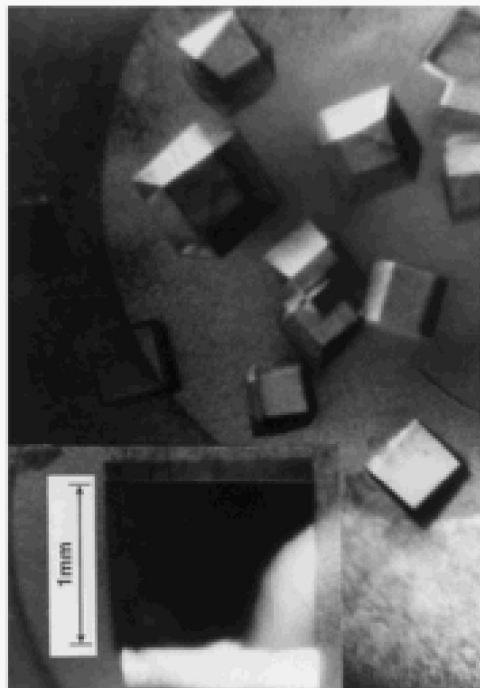
---

The sodium/iodide symporter (NIS) is the essential plasma membrane protein that mediates active iodide ( $I^-$ ) transport into the thyroid gland, the first step in the biosynthesis of the thyroid hormones—the master regulators of intermediary metabolism. NIS couples the inward translocation of  $I^-$  against its electrochemical gradient to the inward transport of  $Na^+$  down its electrochemical gradient<sup>1,2</sup>. For nearly 50 years before its molecular identification<sup>3</sup>, NIS was the molecule at the centre of the single most effective internal radiation cancer therapy: radioiodide ( $^{131}I^-$ ) treatment for thyroid cancer<sup>2</sup>. Mutations in NIS cause congenital hypothyroidism, which must be treated immediately after birth to prevent stunted growth and cognitive deficiency<sup>2</sup>. Here we report three structures of rat NIS, determined by single-particle cryo-electron microscopy: one with no substrates bound; one with two  $Na^+$  and one  $I^-$  bound; and one with one  $Na^+$  and the oxyanion perrhenate bound. Structural analyses,

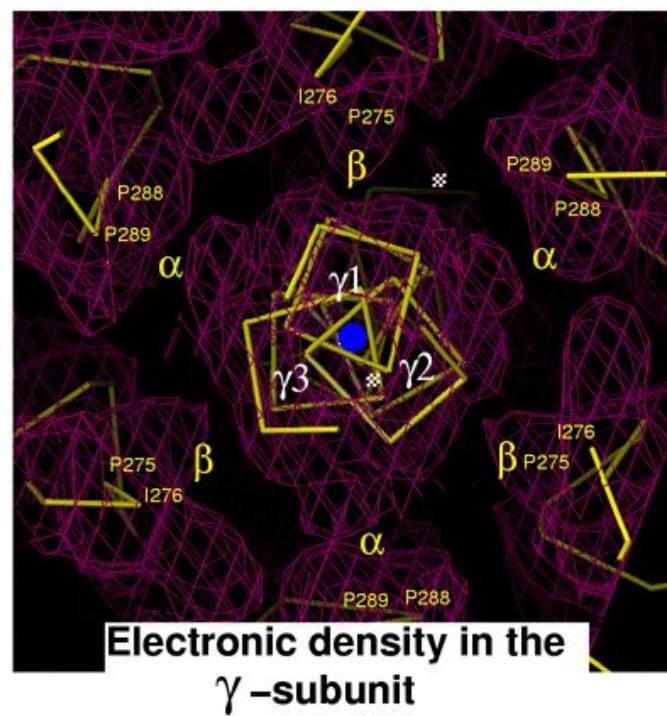
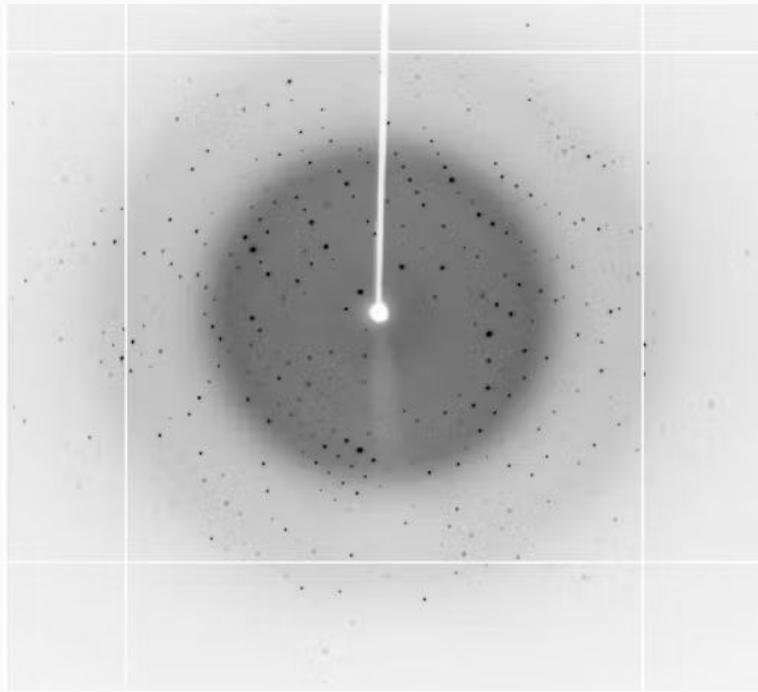
# Thyroid - NIS



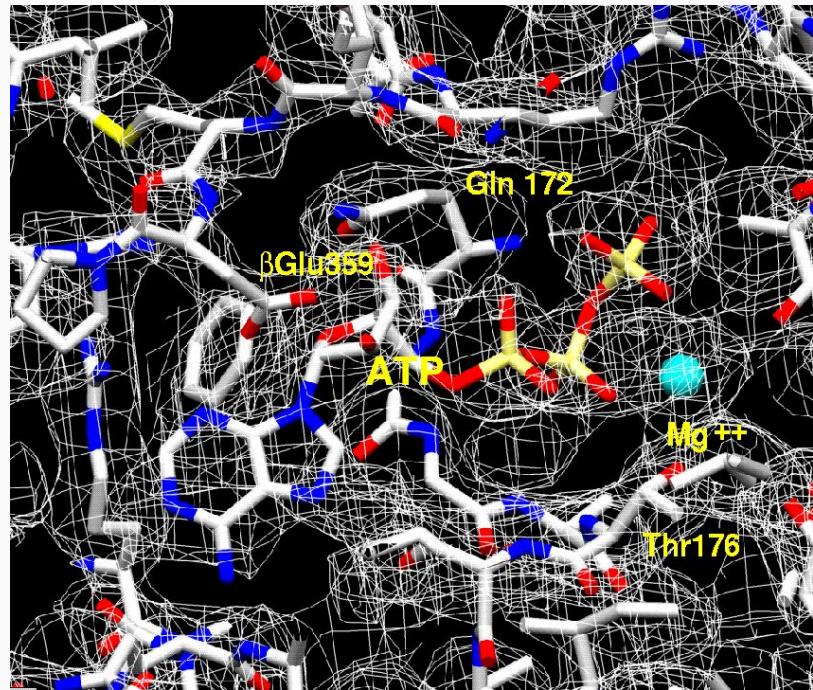
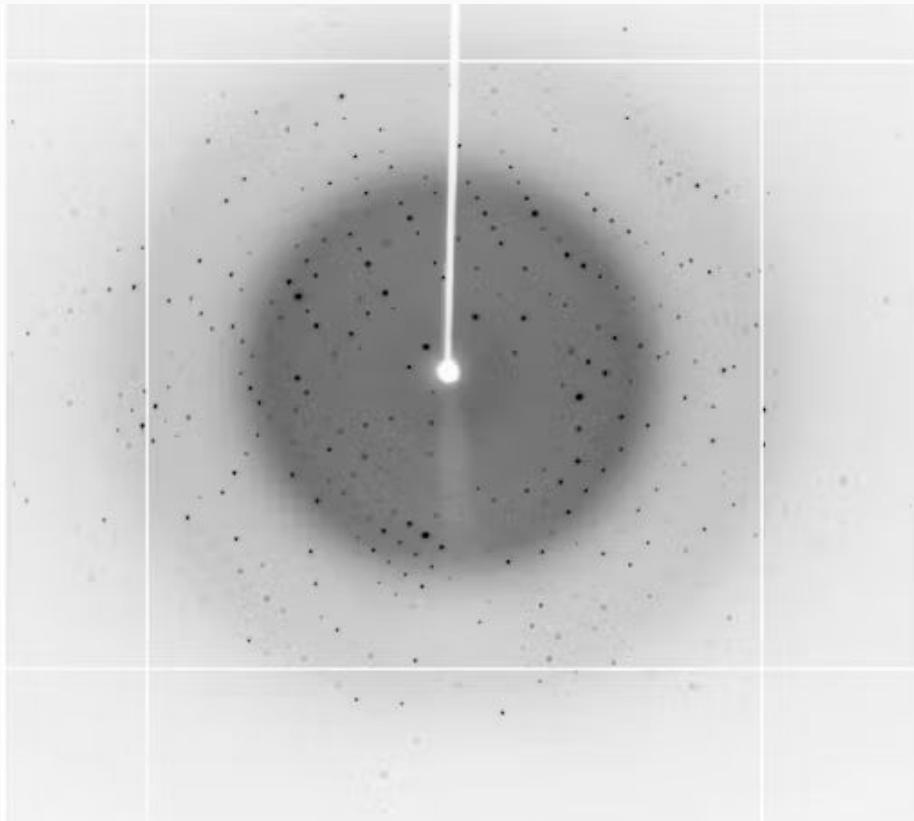
# • Very Basics of X-Ray Crystallography



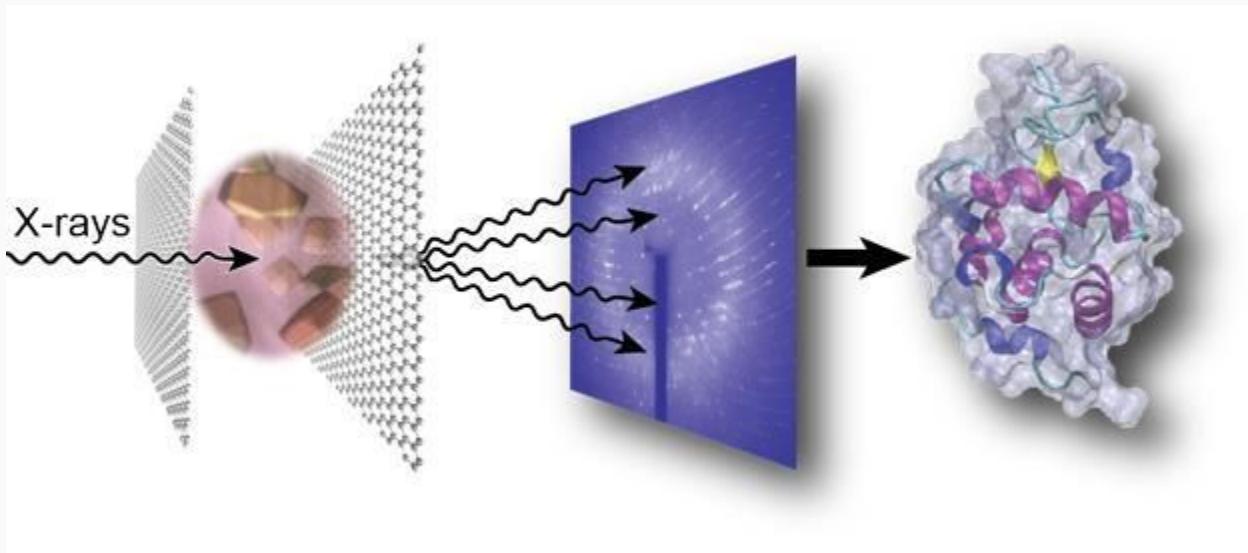
# • Very Basics of X-Ray Crystallography



# • Very Basics of X-Ray Crystallography

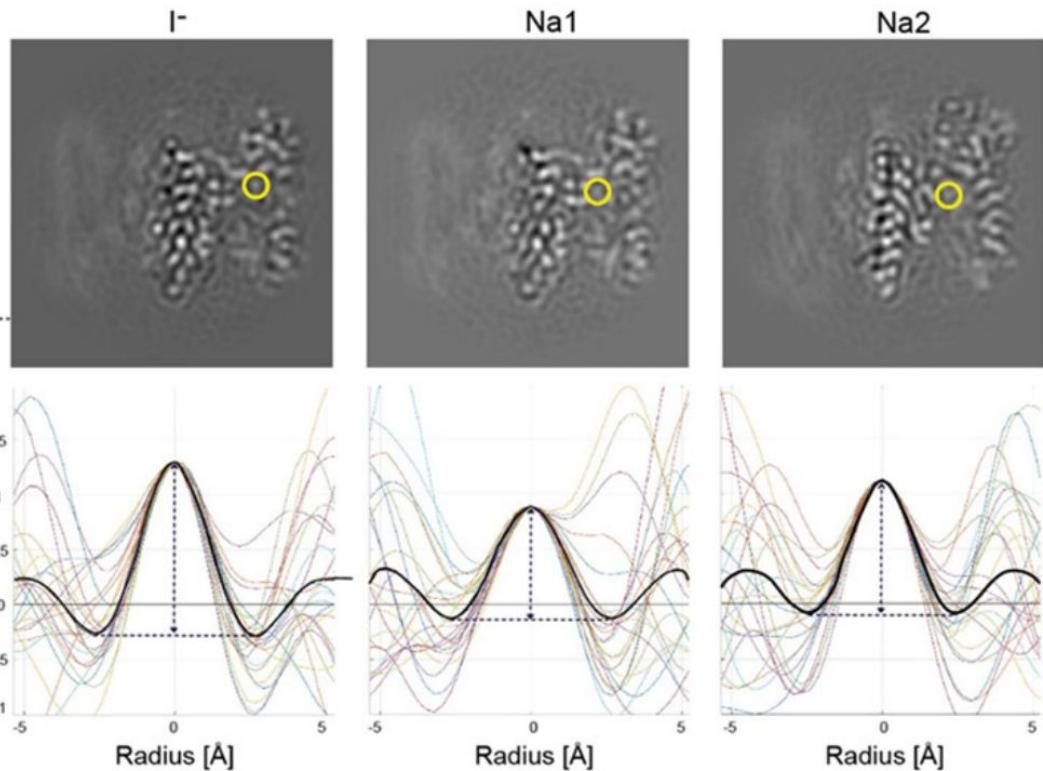
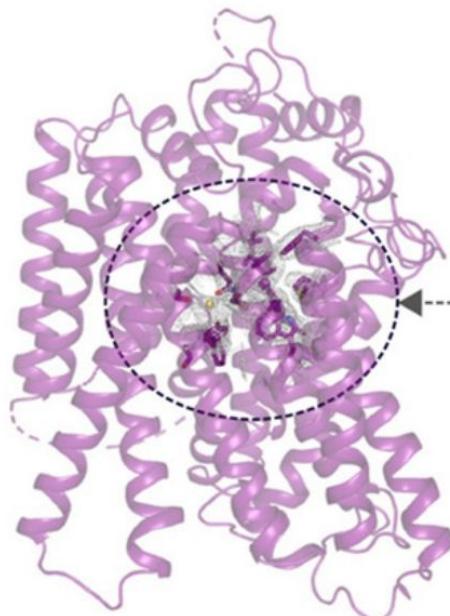


# • Very Basics of X-Ray Crystallography



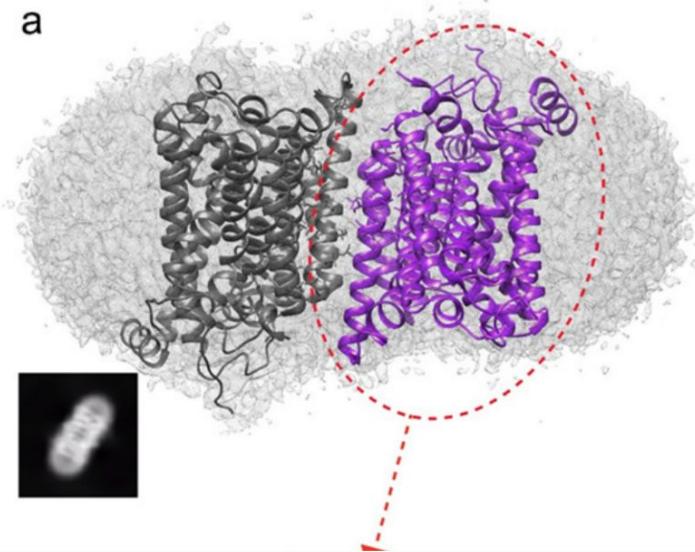
# Basics of NIS

a

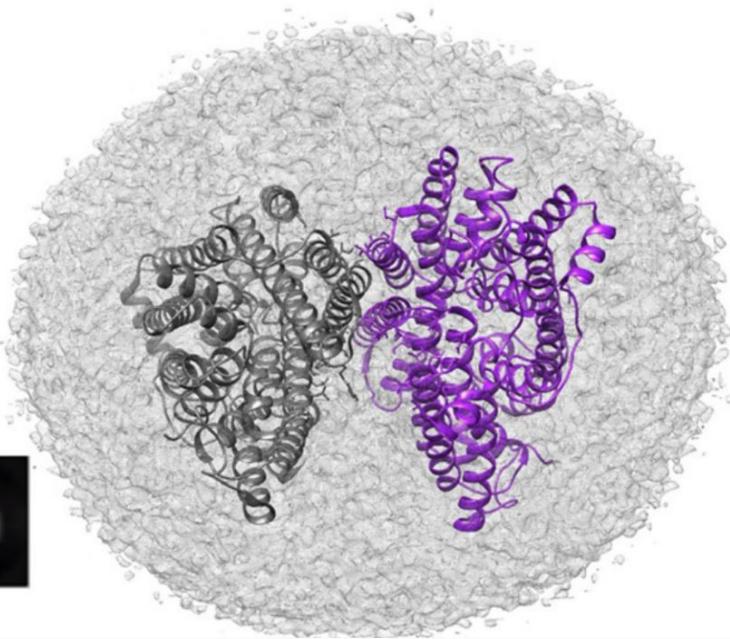
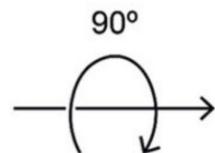


# Basics of NIS

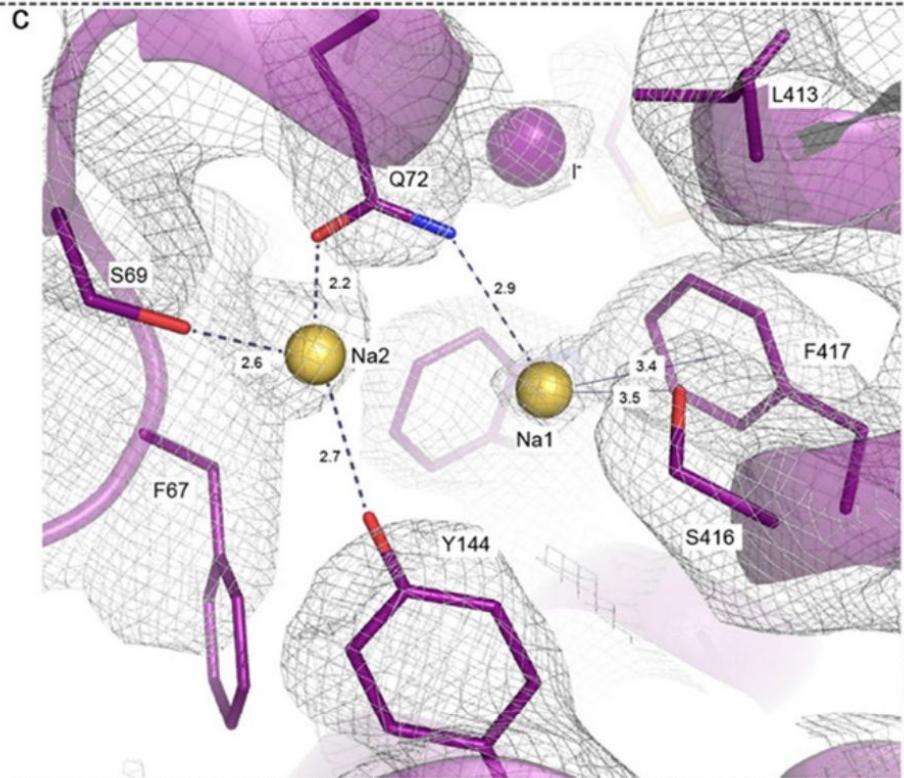
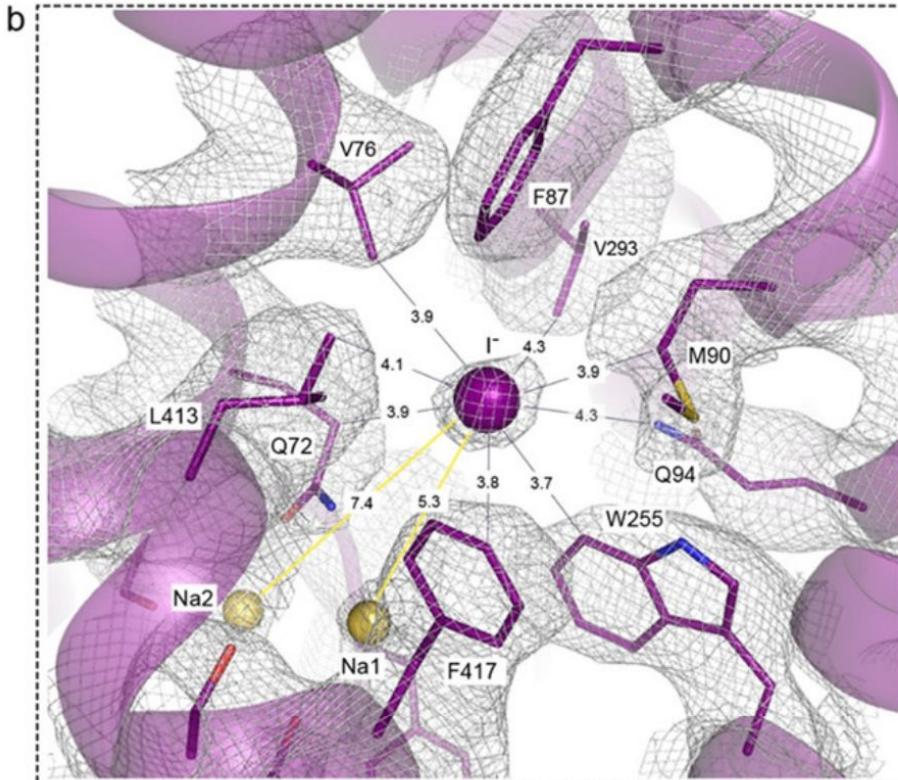
a



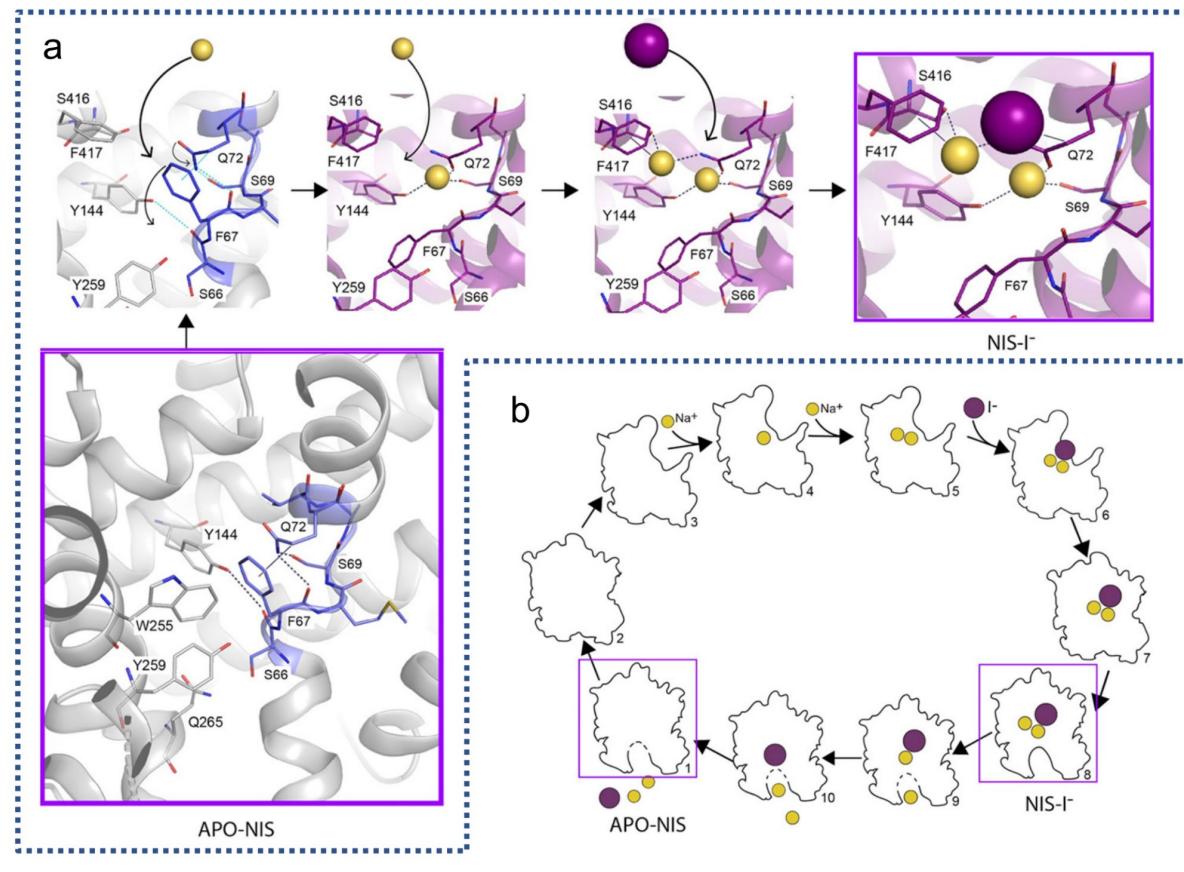
b



# Basics of NIS

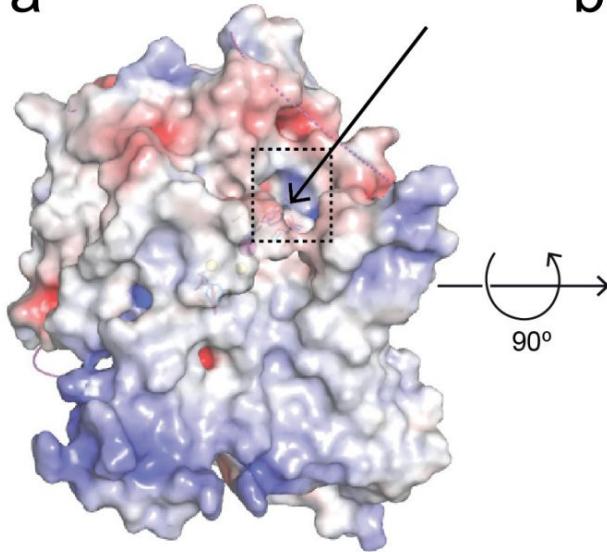


# Basics of NIS

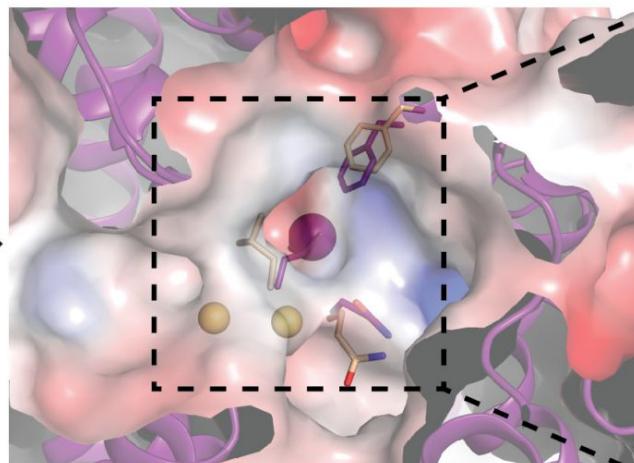


# Basics of NIS

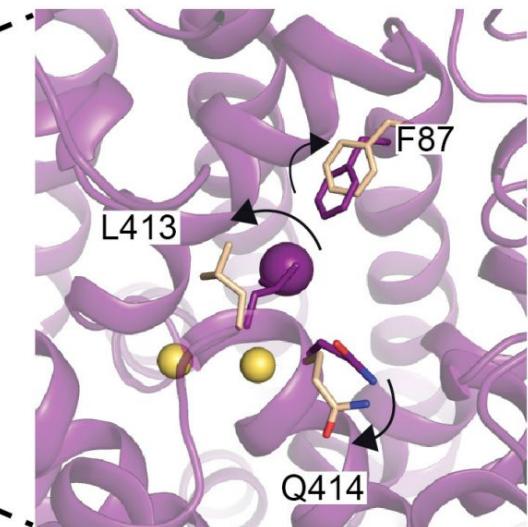
a



b



c



# Molecular Dynamics (MD) Simulation

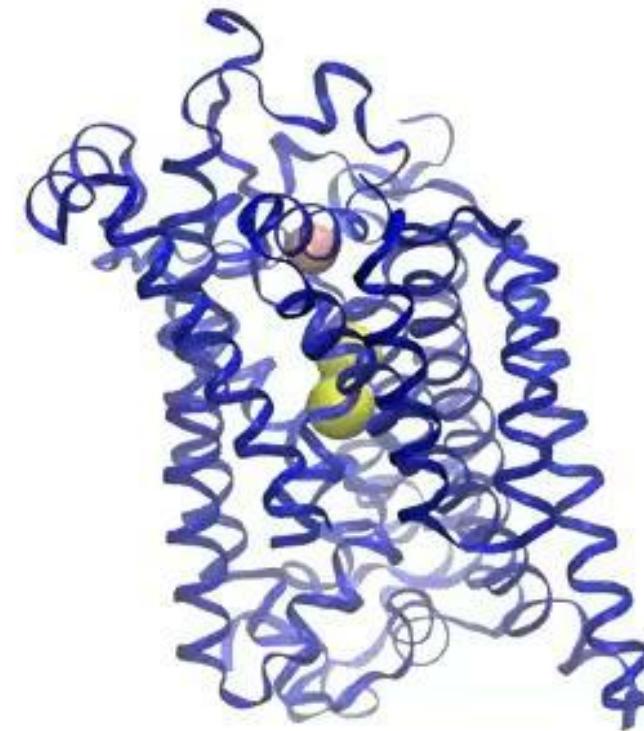
- Computer simulation method

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.

# Basics of NIS



## 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.

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

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

.....

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

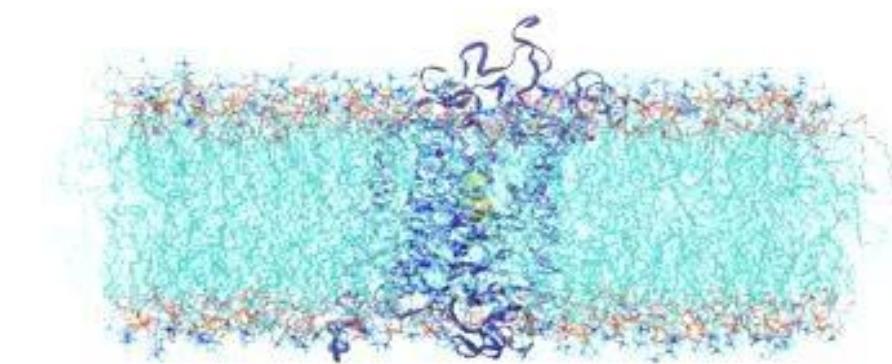
.....

for  $i = 1, \dots, N$

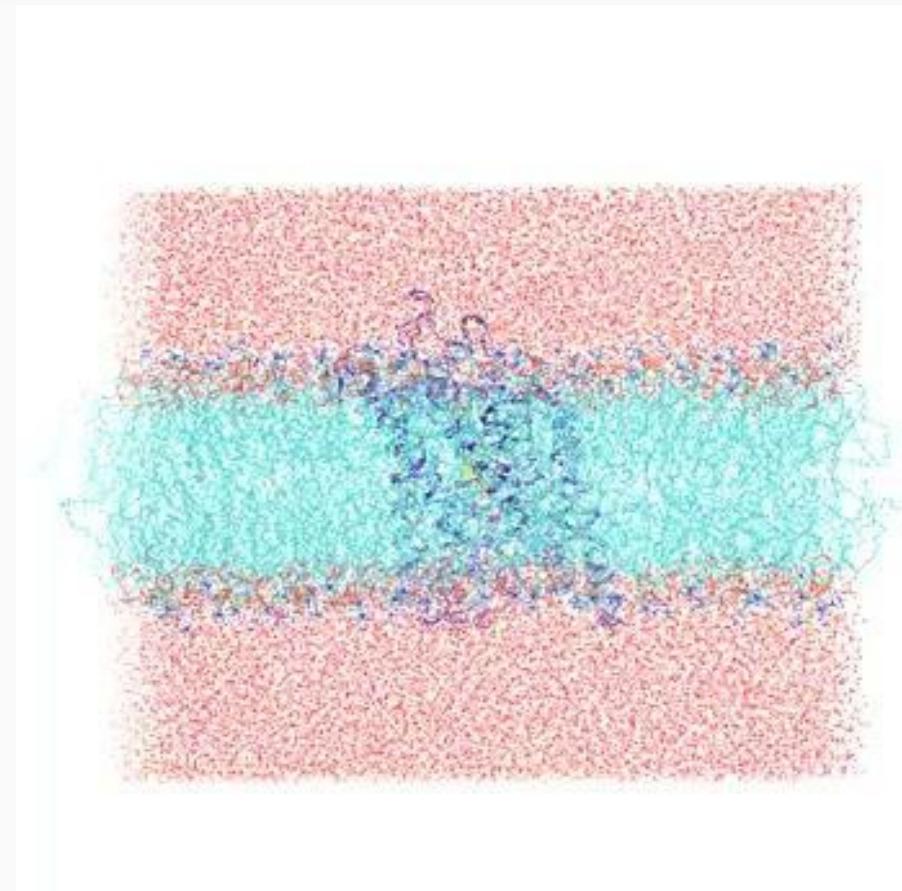
An initial value problem:

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

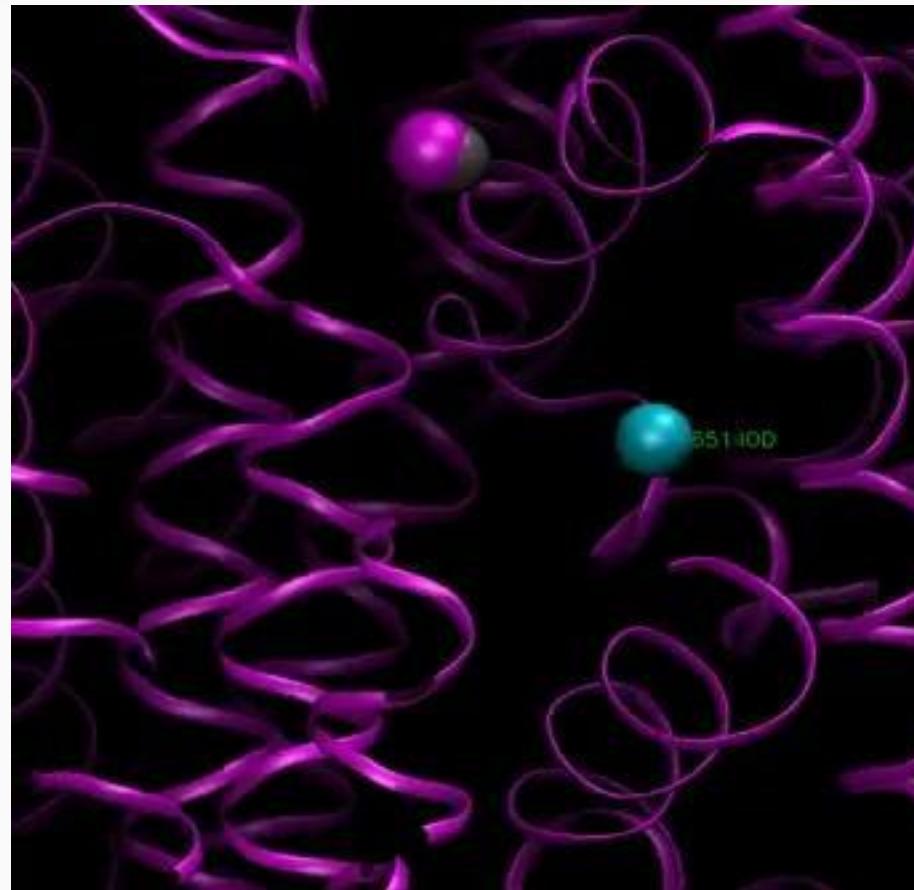
# Basics of NIS



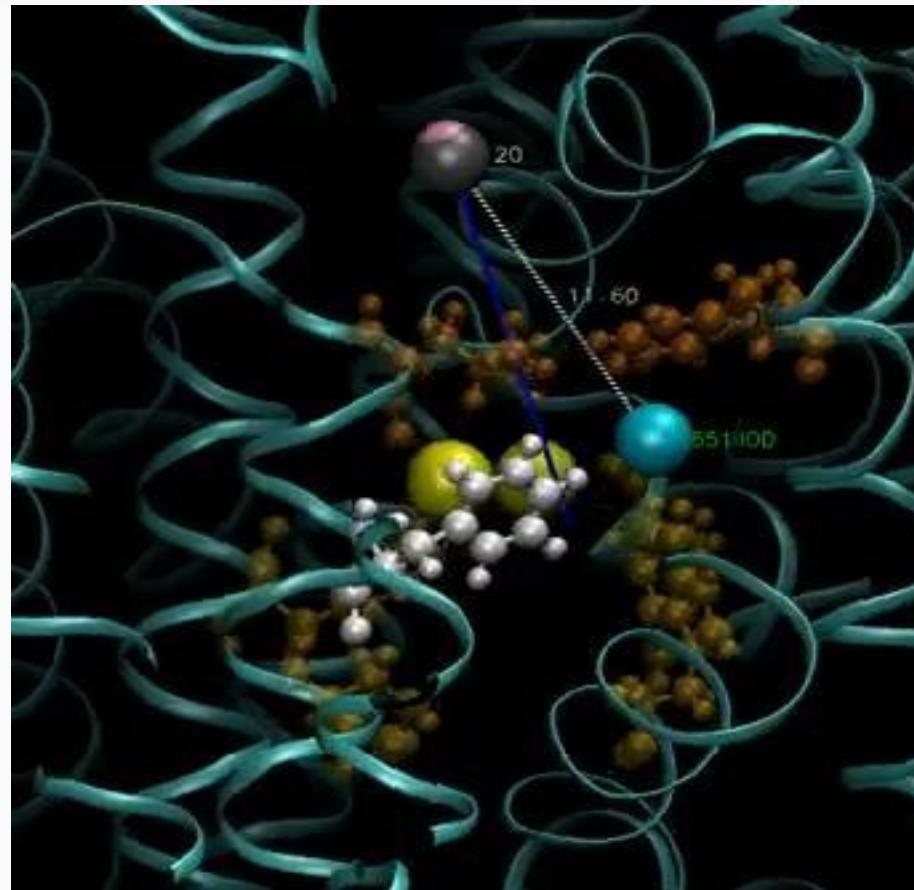
# Basics of NIS



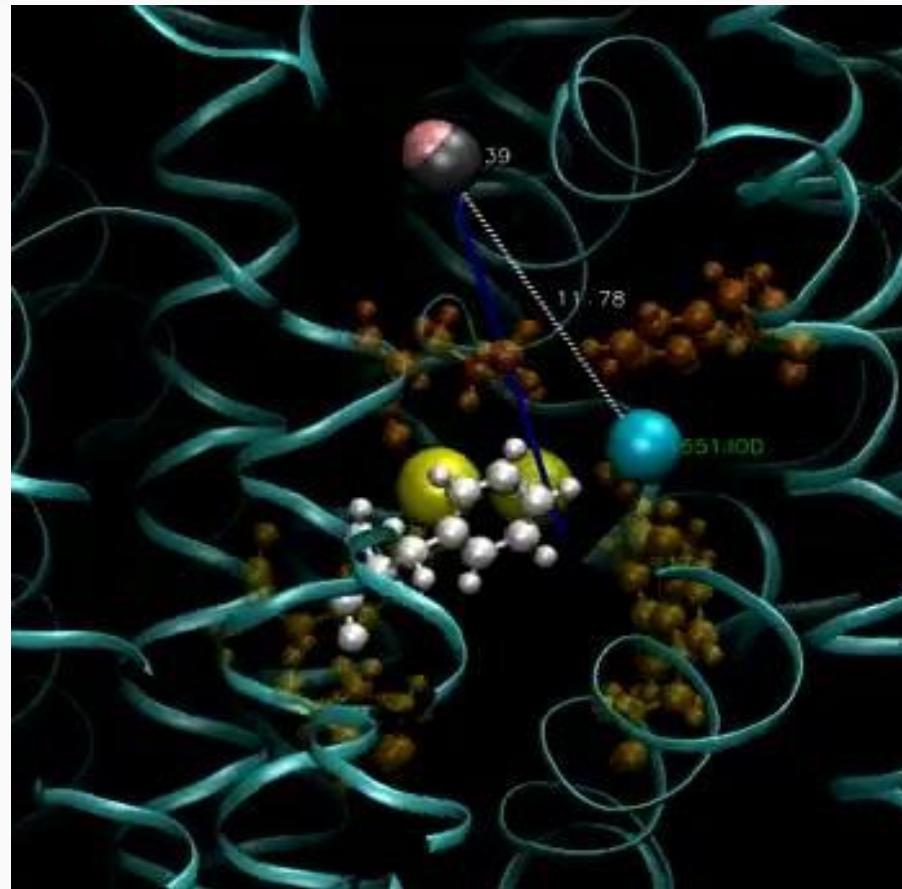
# Basics of NIS



# Basics of NIS

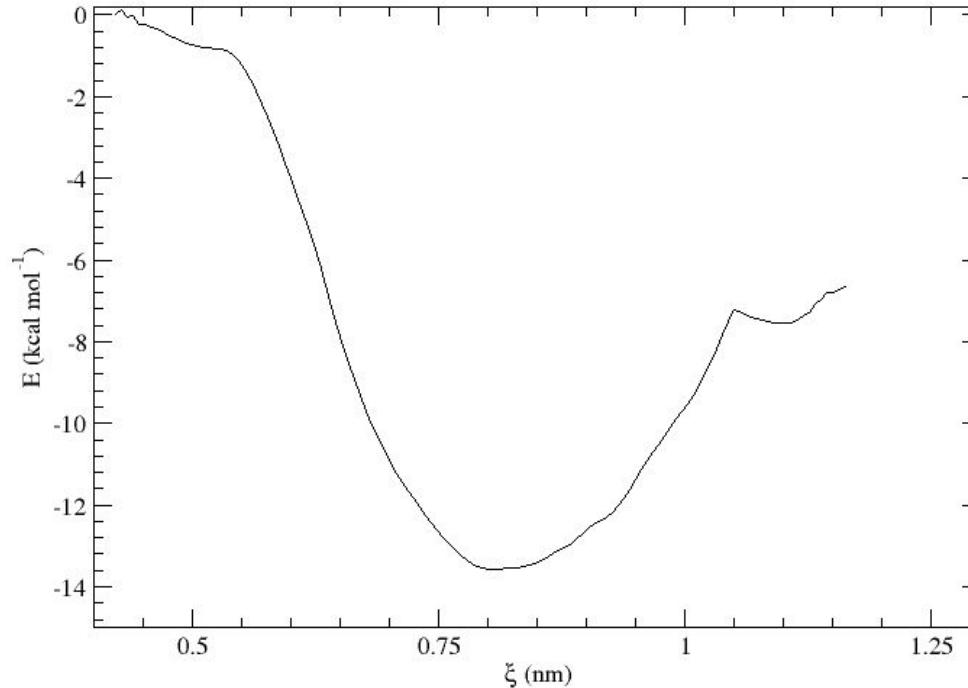


# Basics of NIS

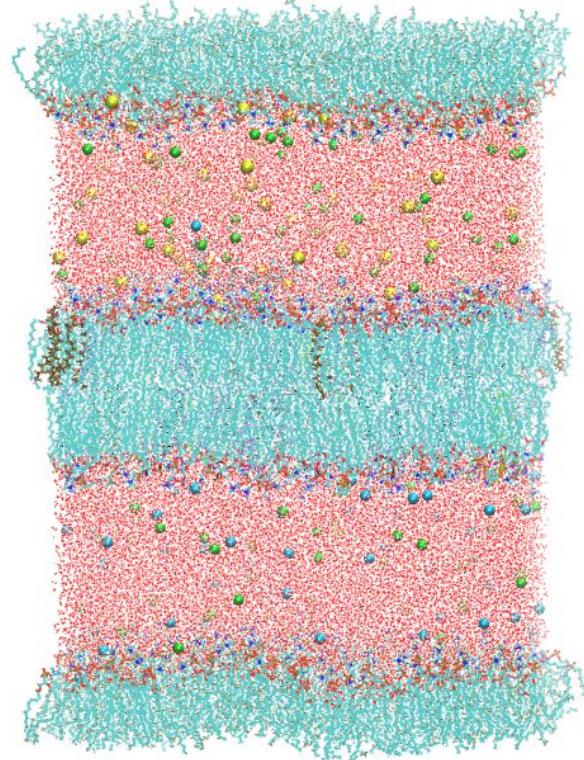


# Basics of NIS

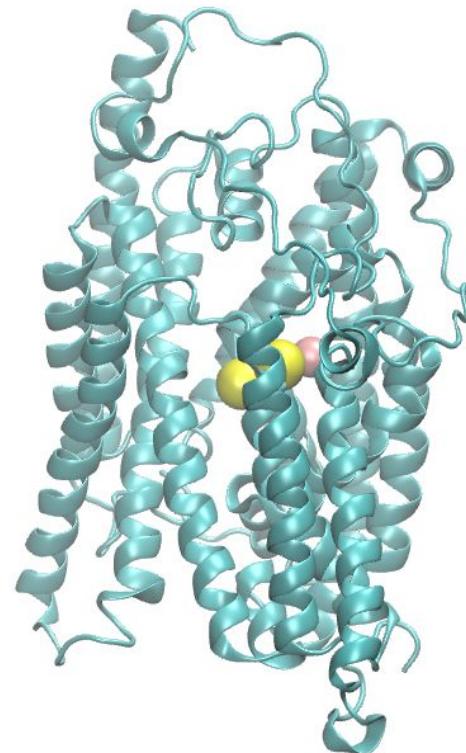
Umbrella potential



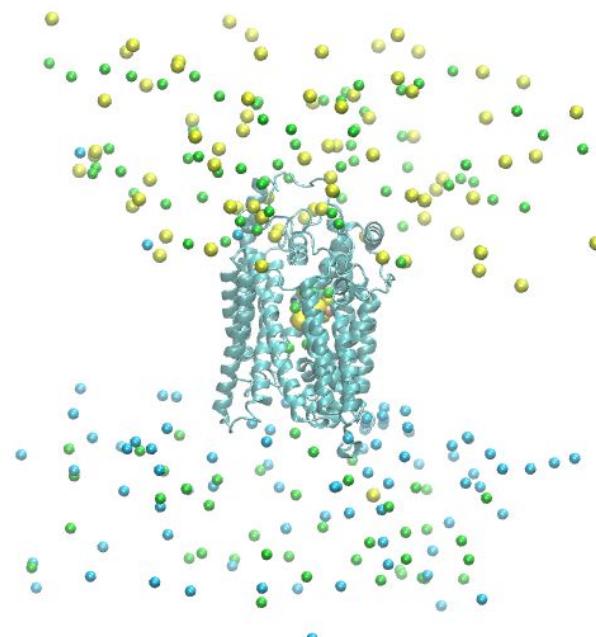
# Basics of NIS



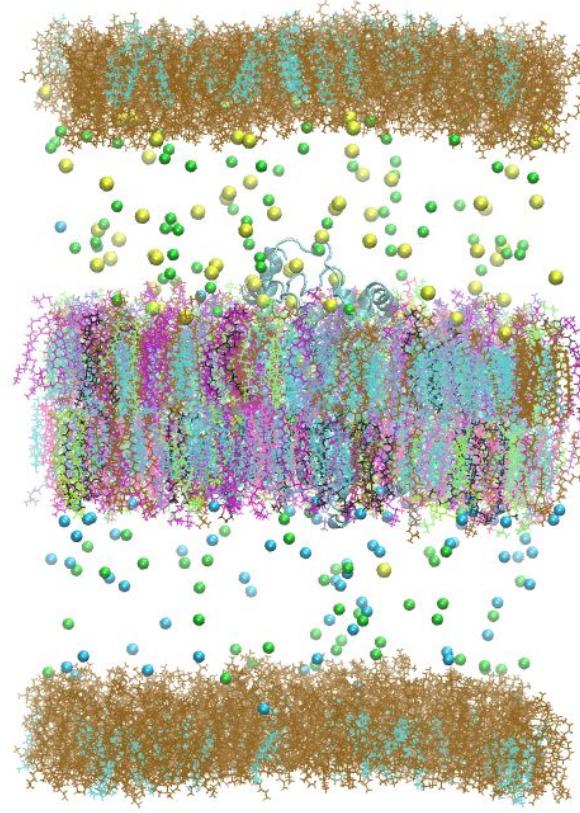
# Basics of NIS



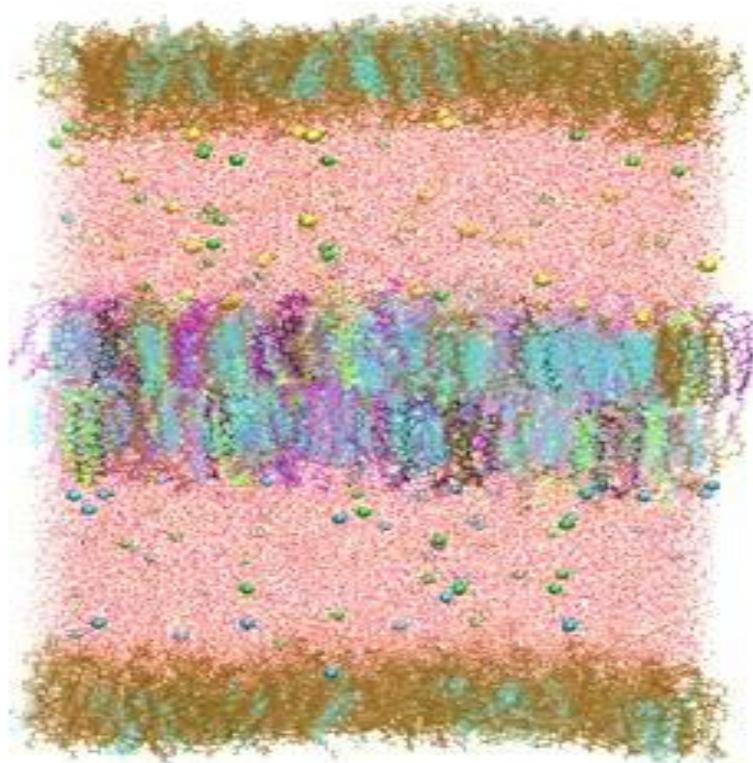
# Basics of NIS



# Basics of NIS



# Basics of NIS



# MD Data

## Gromacs

Ensembles NVE then NPT

Thermostat Nose-Hoover (temperature coupling)  
Barostat Parrinello-Rahman (pressure coupling)  
Electrostatics Ewald  
Cut-off 1.4 nm (electrostatics and VdW)  
Integrator leap-frog algorithm

200000 Atoms

Running time 2.5 micro-seconds

500000 Atoms

Running time 1.5 micro-seconds

## MD Data

Coordinated by the GROMACS project leaders:  
Paul Bauer, Berk Hess, and Erik Lindahl

GROMACS version: 2022.4

Precision: mixed

Memory model: 64 bit

MPI library: MPI

OpenMP support: enabled (GMX\_OPENMP\_MAX\_THREADS = 128)

GPU support: CUDA

SIMD instructions: AVX\_512

CPU FFT library: fftw-3.3.8-sse2-avx-avx2-avx2\_128-avx512

GPU FFT library: cuFFT

Running on 1 node with total 64 cores, 64 processing units, 4 compatible GPUs

Hardware detected on host icgpu04 (the node of MPI rank 0):

CPU info:

Vendor: Intel

Brand: Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz

Family: 6 Model: 106 Stepping: 6



Search...

[Home](#)[◆ Welcome to RIT!](#)[Services ▾](#)[Resources](#)[Partners](#)[Support](#)[About Us](#)

## ARCH & JHU Rockfish

The Advanced Research Computing at Hopkins (ARCH) –formerly known as MARCC– is a shared computing facility at Johns Hopkins University that enables research, discovery, and learning, relying on the use and development of advanced computing. To learn more, please [click here](#).

Last Updated: December 9, 2022



New Faculty Welcome Pack

[Resources](#)  
[Support](#)  
[FAQs](#)

### Location

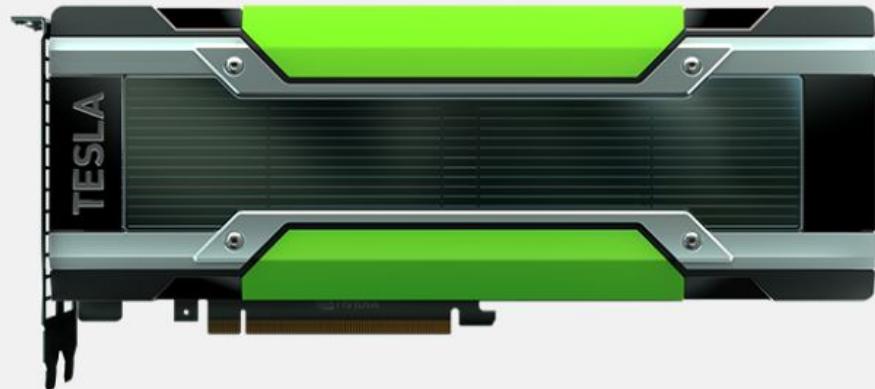
5801 Smith Avenue  
Davis Building  
Baltimore, MD 21209

### Contact

[RITServices@jhu.edu](mailto:RITServices@jhu.edu)

## TESLA K80 ACCELERATOR FEATURES AND BENEFITS

- › 4992 NVIDIA CUDA cores with a dual-GPU design
- › Up to 2.91 teraflops double-precision performance with NVIDIA GPU Boost
- › Up to 8.73 teraflops single-precision performance with NVIDIA GPU Boost
- › 24 GB of GDDR5 memory
- › 480 GB/s aggregate memory bandwidth
- › ECC protection for increased reliability
- › Server-optimised to deliver the best throughput in the data center



# 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.

## 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:  $\{r_i(t), v_i(t)\}_{i=1}^N$

# Molecular Dynamics Fundamentals

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

$$\text{Hamiltonian } H(\{r_i, p_i\}) = \left( \sum_{i=1}^N \frac{p_i^2}{2m_i} \right) + U(\{r_i, p_i\}) \\ \Rightarrow \text{physical quantity } A(\{r_i, p_i\})$$

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 correlation,....

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

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

# Molecular Dynamics Fundamentals

## 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:  $P(t) = \frac{Nk_B T}{V} - \frac{1}{3V} \left\langle \sum_{i=1}^N \sum_{j=i+1}^N r_{ij} \cdot F_{ij} \right\rangle$   
where  $r_{ij} = r_i - r_j$ ,  $F_{ij}$  = force exerted on  $i$  by  $j$
- Shear viscosity  $\eta$ :  
$$6\eta k_B TVt = \sum_{\alpha\beta=xy,yz,zx} \left\langle \left( \sum_{j=1}^N m_j r_{j\alpha}(t) v_{j\beta}(t) - \sum_{j=1}^N m_j r_{j\alpha}(0) v_{j\beta}(0) \right)^2 \right\rangle$$
- Diffusion coefficient:  $\text{MSD } \left\langle \frac{1}{N} \sum_{i=1}^N (r_i(t_0 + t) - r_i(t_0))^2 \right\rangle = 6Dt$

## Time scale and system size can be handled by MD simulations

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

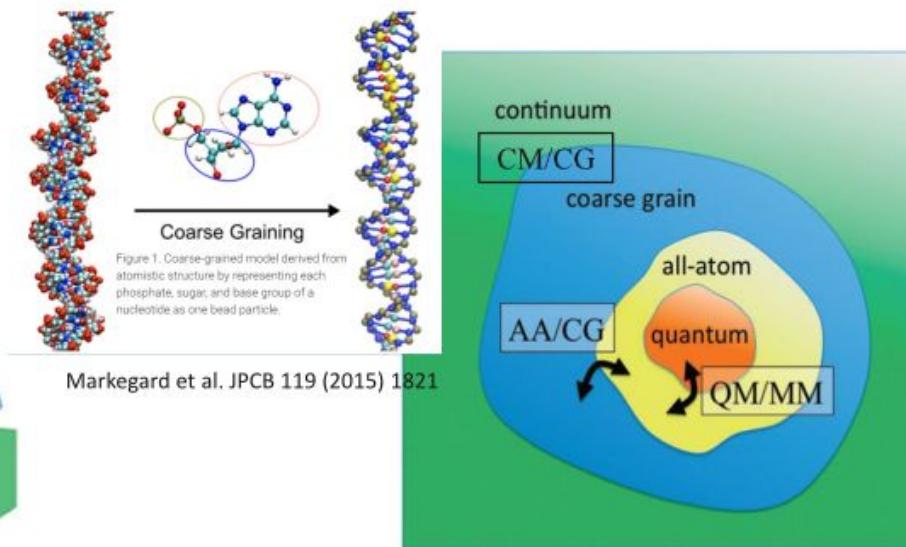
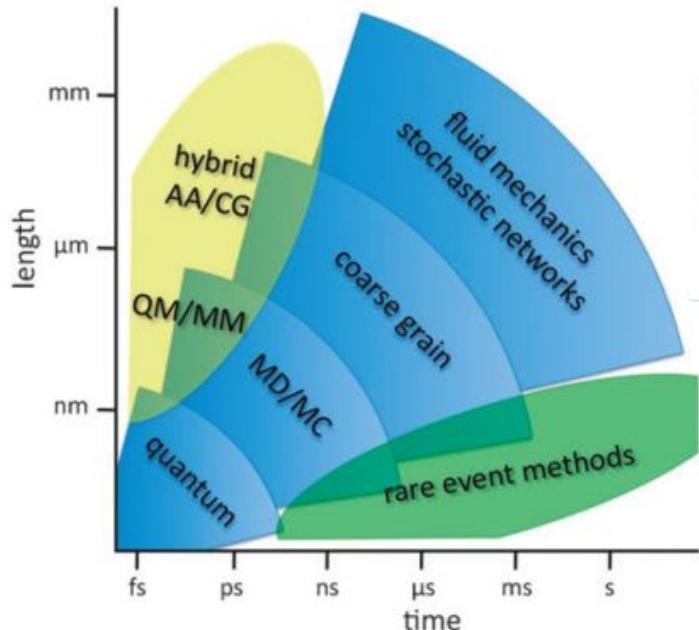
⇒ Characteristic time scale:  $t_u = \sigma \sqrt{M/\varepsilon} \sim 6.3 \text{ ps}$  : [energy] = [mass]  $\left[ \frac{\text{length}}{\text{time}} \right]^2$

⇒ time step to integrate Eq. of motion:  $\Delta t \sim 0.001 - 0.01$   $t_u \sim 6.3 - 63 \text{ fs}$

- No. of molecules:  $< \sim 10^6$  [limited by the memory of computer]  
⇒ system size:  $L \sim 100 \sigma \sim 100 \text{ nm}$  ← 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]  
⇒ total simulation time up to  $\sim 63 \mu\text{s}$  ← of order of microseconds

# Molecular Dynamics Fundamentals

## Length & time scales accessible by different simulation methods



Nielsen et al. PCCP 12 (2010) 12401

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

$$m \frac{dv}{dt} = F \quad \text{approx. by} \quad m \frac{\Delta v}{\Delta t} \cong F, \quad \text{which is} \quad \frac{\Delta v}{\Delta t} = \frac{v(t+\Delta t) - v(t)}{\Delta t} \cong \frac{F(t)}{m}$$
$$\frac{dr}{dt} = v \quad \text{approx. by} \quad \frac{\Delta r}{\Delta t} \cong v, \quad \text{which is} \quad \frac{\Delta r}{\Delta t} = \frac{r(t+\Delta t) - r(t)}{\Delta t} \cong v(t)$$

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:

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

---- an initial value problem (IVP)

## Other methods learned in Numerical Analysis (1)

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

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

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

$$k_1 = hG(t_n, x_n, v_n)$$

$$\ell_1 = hF(t_n, x_n, v_n)$$

$$k_2 = hG(t_n + 0.5h, x_n + 0.5k_1, v_n + 0.5\ell_1)$$

$$\ell_2 = hF(t_n + 0.5h, x_n + 0.5k_1, v_n + 0.5\ell_1)$$

$$k_3 = hG(t_n + 0.5h, x_n + 0.5k_2, v_n + 0.5\ell_2)$$

$$\ell_3 = hF(t_n + 0.5h, x_n + 0.5k_2, v_n + 0.5\ell_2)$$

$$k_4 = hG(t_n + h, x_n + k_3, v_n + \ell_3)$$

$$\ell_4 = hF(t_n + h, x_n + k_3, v_n + \ell_3)$$

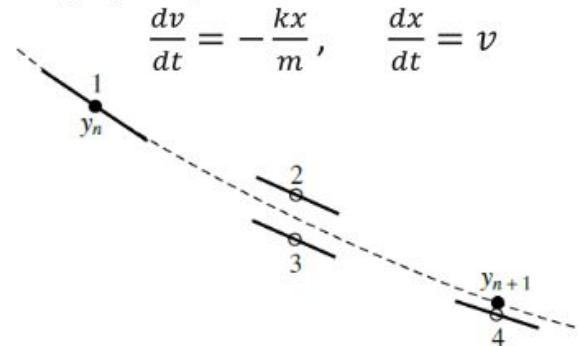
$$\Rightarrow x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$v_{n+1} = v_n + \frac{1}{6}(\ell_1 + 2\ell_2 + 2\ell_3 + \ell_4)$$

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

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

(ex) Simple Harmonic Oscillator



## Other methods learned in Numerical Analysis (2)

- Predictor-corrector method of order q: (PEC) → (PEC) → (PEC) → ...

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

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

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  $r^{(p)}(t + \Delta t)$ .

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

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

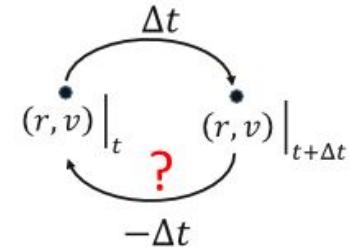
TABLE 4.1 Values of  $\alpha_i$  Parameters in Gear's Predictor-Corrector Algorithm\* for Second-Order Differential Equations Using Predictors of Order q

$\alpha_i$	$q = 3$	$q = 4$	$q = 5$
$\alpha_0$	$\frac{1}{6}$	$\frac{19}{120}$	$\frac{3}{16}$
$\alpha_1$	$\frac{5}{6}$	$\frac{3}{4}$	$\frac{251}{360}$
$\alpha_2$	1	1	1
$\alpha_3$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{11}{18}$
$\alpha_4$	—	$\frac{1}{12}$	$\frac{1}{6}$
$\alpha_5$	—	—	$\frac{1}{80}$

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

# Molecular Dynamics Fundamentals

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



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

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

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

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

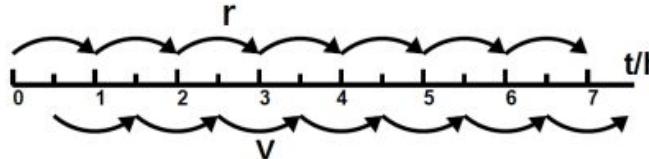
# Molecular Dynamics Fundamentals

**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:

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

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

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

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 $\Delta t$ ?

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

where  $\omega_j$  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  $\Delta t$ , the smaller the truncation error, and the shorter duration time and the trajectory distance can be explored.

Note: Extremely small  $\Delta t$  also lead to large error in the calculation because of computer's round-off error.

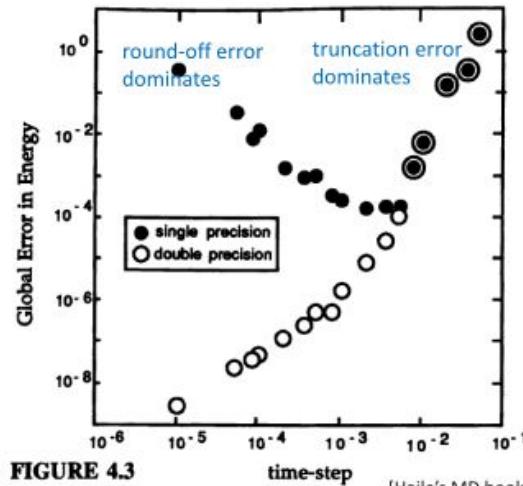
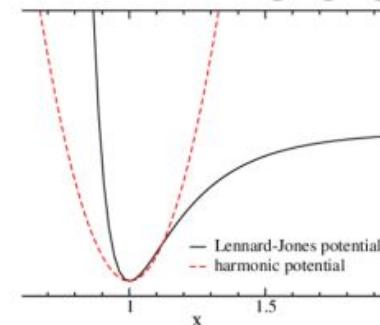


FIGURE 4.3

[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  $i$ th 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:

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

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

The equation of motion for particle  $i$ ,

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

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

$$m_i \frac{dv_{iz}}{dt} = F_{iz}, \quad \frac{dr_{iz}}{dt} = v_{iz}, \quad i = 1, \dots, N, \quad [\text{totally, } 6N \text{ ODEs}]$$

# Molecular Dynamics Fundamentals

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

### (I) non-bonded interactions:

a. Lennard-Jones interaction:

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

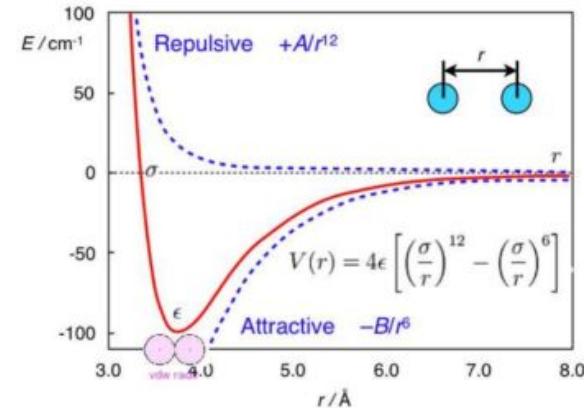
b. Buckingham potential:

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

c. Coulomb interaction:

$$u_{col}(r) = \frac{q_1 q_2}{4\pi\epsilon\epsilon_0 r}$$

Can be further distinguished into “[short-range interaction](#)”: a, b  
and “[long-range interaction](#)” : c



# Molecular Dynamics Fundamentals

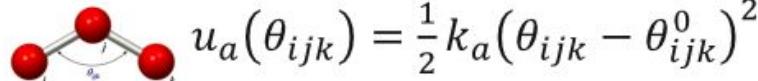
## (II) bonded interaction

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

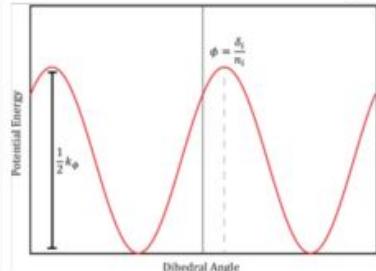
b. Morse potential:

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

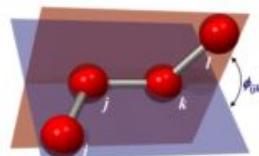
c. Harmonic angle potential:



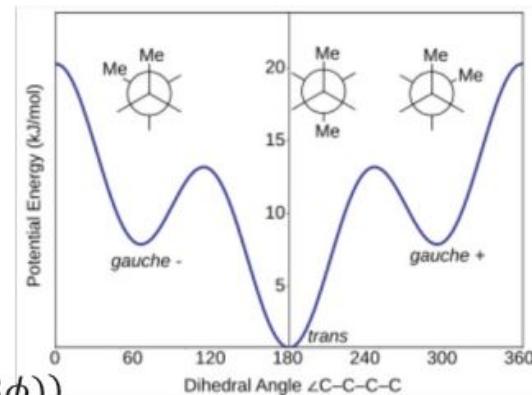
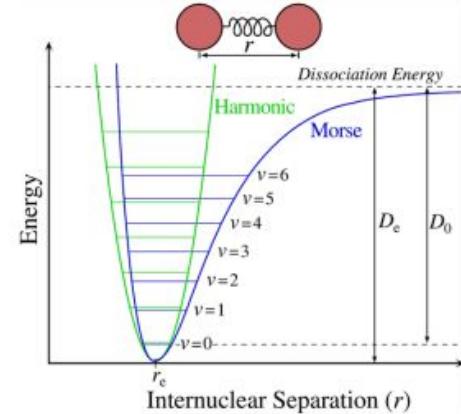
d. Dihedral angle potential:



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



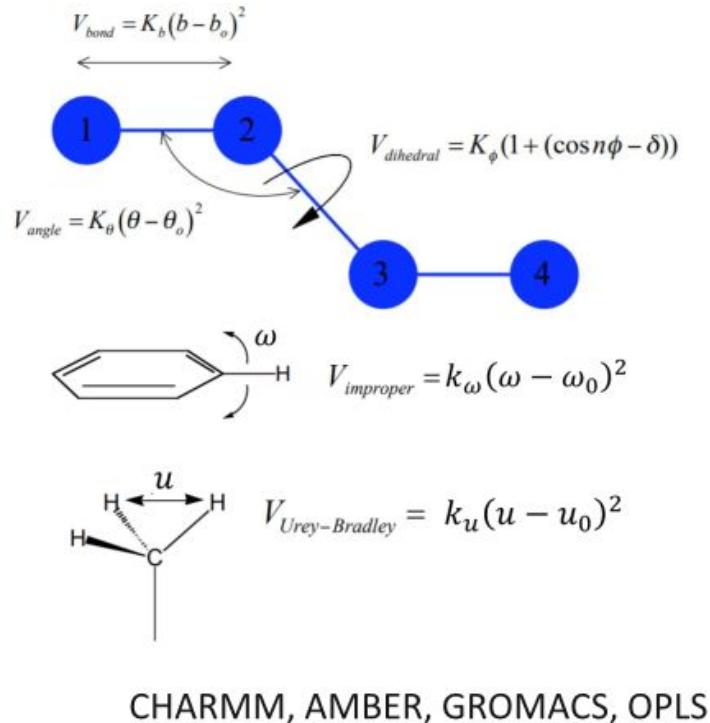
$$u_h(\phi) = k_1(1 + \cos(\phi)) + k_2(1 + \cos(2\phi)) + k_3(1 + \cos(3\phi))$$



# Molecular Dynamics Fundamentals

## Class I Force field:

$$E = \sum_{bonds} k_b(b - b_0)^2 + \sum_{angles} k_\theta(\theta - \theta_0)^2$$
$$+ \sum_{dihedrals} k_\phi(1 + \cos(n\phi - \delta))$$
$$+ \sum_{impropers} k_\omega(\omega - \omega_0)^2$$
$$+ \sum_{Urey-Bradley} k_u(u - u_0)^2$$
$$+ \sum_{vdW} 4\varepsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^6 \right] + \sum_{coulomb} \frac{q_i q_j}{4\pi \varepsilon \varepsilon_0 r_{ij}}$$



# Molecular Dynamics Fundamentals

## Class II Force field: involving further anharmonic terms and cross terms

$$\begin{aligned} E = & E_{vdw} + E_{coul} + \\ & \sum_{bonds} [K_{b,2}(b - b_o)^2 + K_{b,3}(b - b_o)^3 + K_{b,4}(b - b_o)^4] \\ & + \sum_{angles} [K_{\theta,2}(\theta - \theta_o)^2 + K_{\theta,3}(\theta - \theta_o)^3 + K_{\theta,4}(\theta - \theta_o)^4] \\ & + \sum_{dihedrals} [K_{\phi,1}(1 - \cos\phi) + K_{\phi,2}(1 - \cos 2\phi) + K_{\phi,3}(1 - \cos 3\phi)] \\ & + \sum_{impropers} K_{\chi}\chi^2 \\ & + \sum_{bonds} \sum_{bonds'} K_{bb'}(b - b_o)(b' - b_o') + \sum_{angles} \sum_{angles'} K_{\theta\theta'}(\theta - \theta_o)(\theta' - \theta_o') \\ & + \sum_{bonds} \sum_{angles} K_{b\theta}(b - b_o)(\theta - \theta_o) \\ & + \sum_{bonds} \sum_{dihedrals} (b - b_o)[K_{\phi,b1} \cos\phi + K_{\phi,b2} \cos 2\phi + K_{\phi,b3} \cos 3\phi] \\ & + \sum_{angles} \sum_{dihedrals} (\theta - \theta_o)[K_{\phi,\theta1} \cos\phi + K_{\phi,\theta2} \cos 2\phi + K_{\phi,\theta3} \cos 3\phi] \\ & + \sum_{angles} \sum_{angles'} \sum_{dihedrals} K_{\theta\theta'\phi}(\theta - \theta_o)(\theta' - \theta_o') \cos\phi \end{aligned}$$

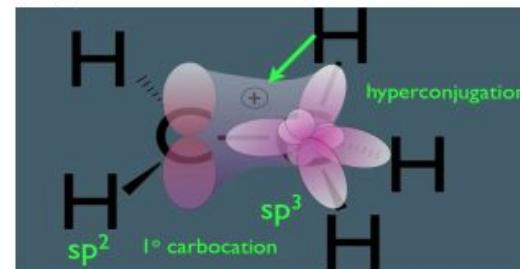
MM3, UFF, MMFF

# Molecular Dynamics Fundamentals

- **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**

$$\begin{aligned} E = & E_{coulomb} + E_{dispersion} + E_{polarization} \\ & + E_{charge-transfer} + E_{exchange-repulsion} \end{aligned}$$

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

# Molecular Dynamics Fundamentals

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:  $\langle T \rangle, \langle P \rangle, \langle \mu \rangle, \dots$
  - $\Leftrightarrow$  NVT ensemble:  $\langle E \rangle, \langle P \rangle, \langle \mu \rangle, \dots$
  - $\Leftrightarrow$  NPT ensemble:  $\langle E \rangle, \langle V \rangle, \langle \mu \rangle, \dots$
  - $\Leftrightarrow$   $\mu$ VT ensemble:  $\langle E \rangle, \langle P \rangle, \langle N \rangle, \dots$
- in the limit of thermodynamics

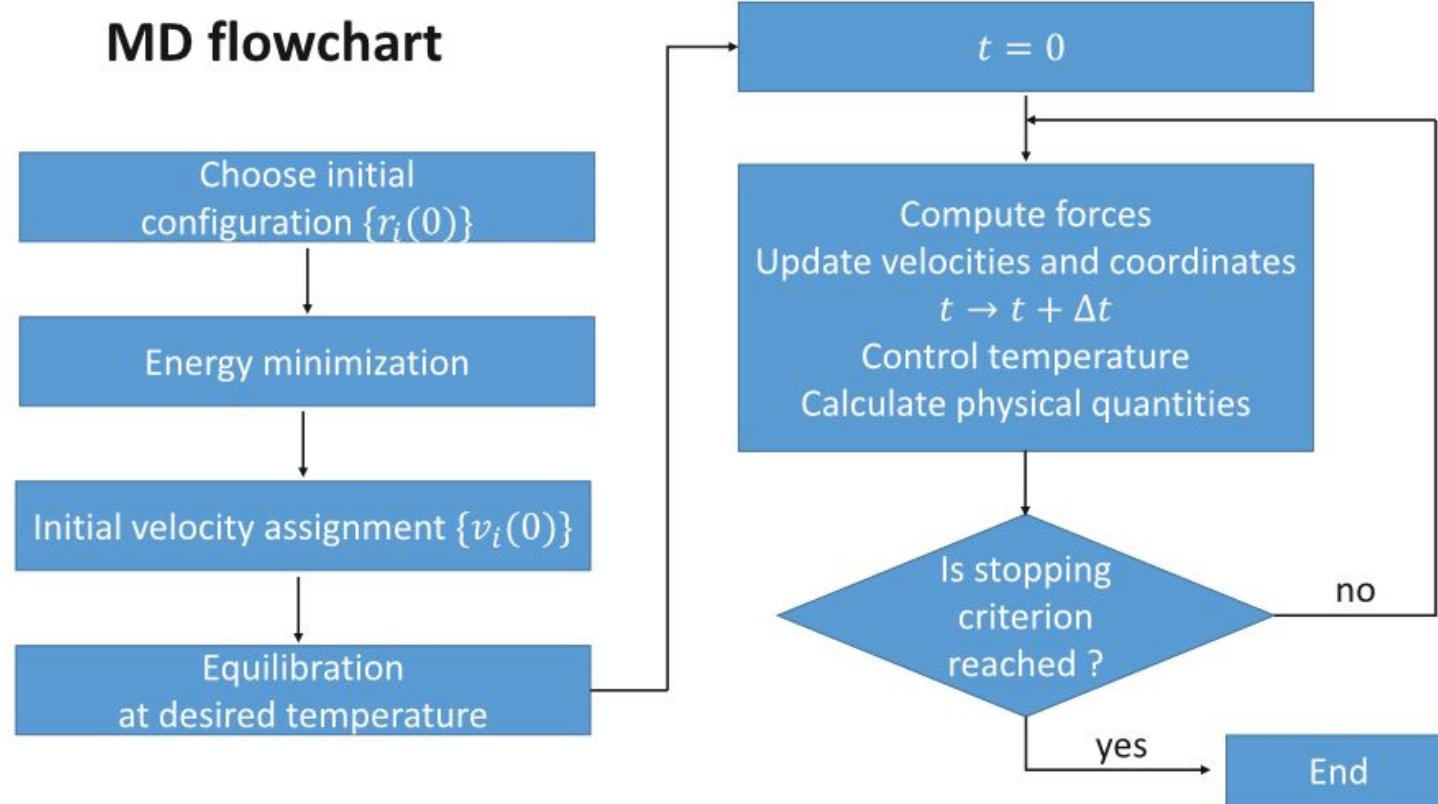
Forcing the system to relax at the desired temperature  $T_D$

(ex) velocity scaling method:

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

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

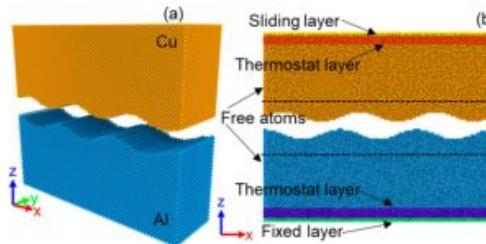
# Molecular Dynamics Fundamentals



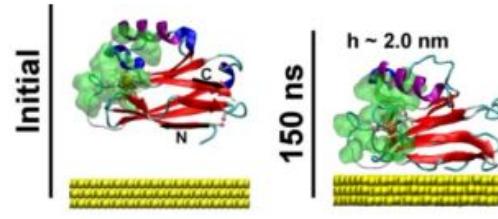
# Molecular Dynamics Fundamentals

- Initial configuration:

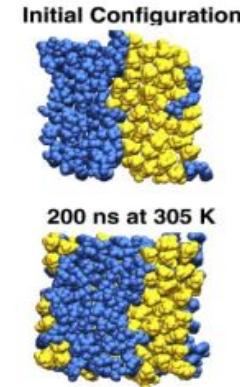
X-ray, NMR, lattices, random distribution, ...



[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):

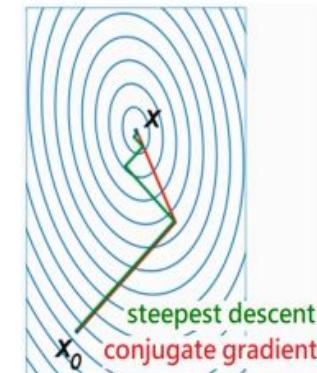
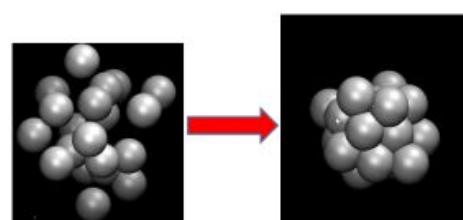
steepest descent, conjugate gradient, Newton-Raphson,...

- + not care about velocity

- + Force is computed and

- atoms move in the direction of force

- + molecular dynamics at  $T=0$



# Molecular Dynamics Fundamentals

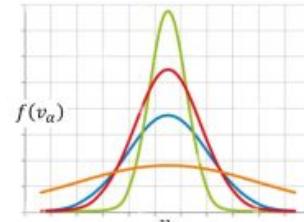
- Assignment of initial velocities

⇒ 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:  $v_{i\alpha} \leftarrow v_{i\alpha} \sqrt{T_D / \left(\frac{2}{3(N-1)k_B} KE\right)}$



## 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.  
⇒ Not possible to have 100% sure for reaching of equilibrium

# Molecular Dynamics Fundamentals

## 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  
}
```

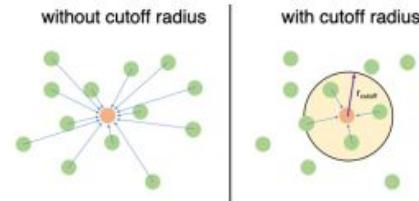
```
Subroutine force(F) {  
    for i = 1 to N; do  
        F[i] = 0  
    done  
    for i = 1 to N-1; do  
        for j = i+1 to N; do  
            compute: fij  
            F[i] = F[i] + fij  
            F[j] = F[j] - fij  
        done  
    done  
}
```

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

# Molecular Dynamics Fundamentals

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

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



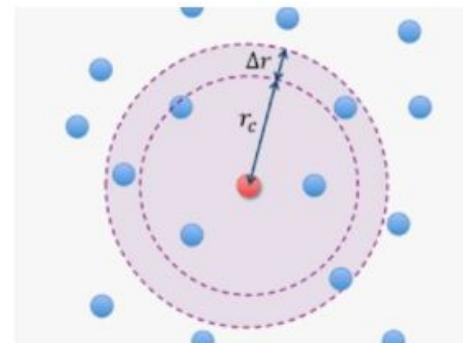
- BUT constructing a **neighbor list** is an  $O(N^2)$  algorithm !

```
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 Neighbor(j)  
    done  
  done
```

No. of neighbors  $\approx \rho_0 \left( \frac{4}{3} \pi r_c^3 \right)$   
(about a constant, indep. of N)

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

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



## Ewald sum method:

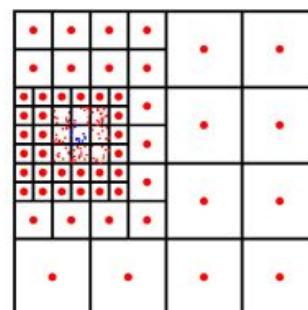
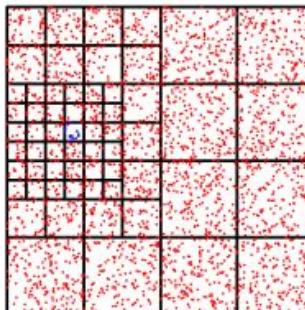
choosing good parameters  $\alpha, r_c, k_c \Rightarrow O(N^{1.5})$

Incorporate **Fast Fourier transform** to solve discrete Poisson equation

**Particle Mesh Ewald**  $\Rightarrow O(N \log N)$

algorithm using multipole expansion to calculate electric field by grouping charges

**Fast Multipole Method**  $\Rightarrow O(N)$



### Top 10 algorithms from the 20<sup>th</sup> 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:**  $v_i \leftarrow v_i \sqrt{\frac{T_D}{T}}$ ,  $T = \frac{2K}{3(N-1)k_B}$  (too crude, very violent!)
- **Berendsen thermostat:**  $v_i \leftarrow v_i \sqrt{1 + \frac{\Delta t}{\tau} \left( \frac{T_D}{T} - 1 \right)}$ ,  $\tau$ = 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.

$$v_{ix,y,z} \leftarrow \sqrt{k_B T / m_i} Z \quad \text{if the particle } i \text{ collides with the heat bath}$$

- **Gaussian thermostat:** constraint method, isokinetic

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

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

## 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

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

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

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

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

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

Eq. of motion:  $\frac{d\vec{p}_i}{dt} = -\frac{\partial U}{\partial \vec{r}_i} - \frac{p_\eta}{Q} \vec{p}_i$

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

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

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

$(p_\eta, \eta)$  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  $\{p_{\eta_j}, \eta_j\}_{j=1}^M$

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

## Isothermal-isobaric method (MTK 1994) : NPT ensemble

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

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

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

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

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

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

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

Conserved quantity: non-Hamiltonian

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

## Calculation of static quantities

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

RMSD is calculated between two sets of atomic coordinates.

It is a measure of how much the protein conformation has changed.

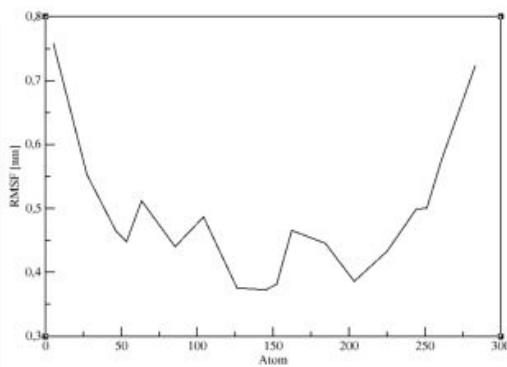
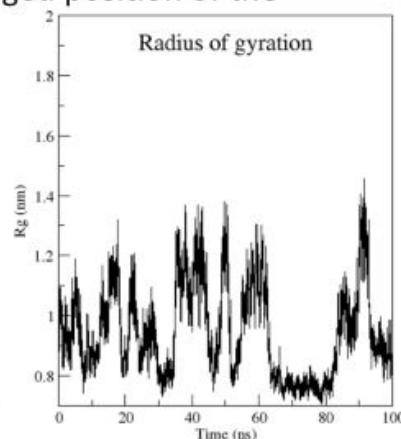
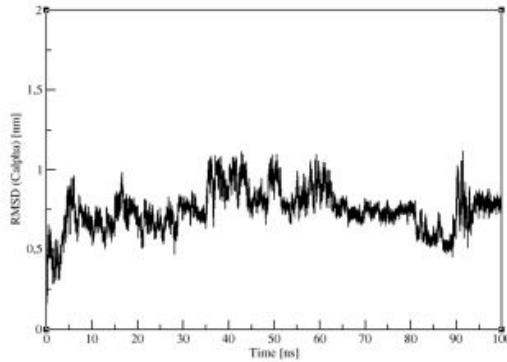
- **Root Mean Square Fluctuation:**  $\text{RMSF}_i = \sqrt{\frac{1}{T} \sum_t (\vec{r}_i(t) - \langle \vec{r}_i(t) \rangle)^2}$

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:**

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

a measure for the compactness  
of a structure



## 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  $t \rightarrow \infty$

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

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

$D$  is a time integral of the velocity correlation function

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

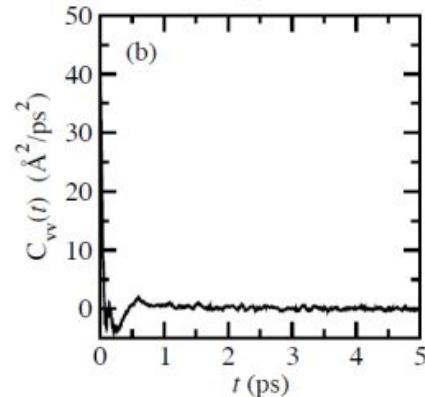
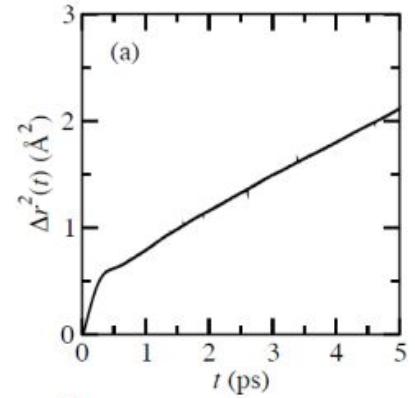
$$\text{where } G_{\alpha\beta}(t) = \int_0^t P_{\alpha\beta}(t') dt'$$

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

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

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

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



## 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 factorization
  - + calculations of static and dynamic properties, correlation functions

**Thanks for your attention!  
Any Questions?**