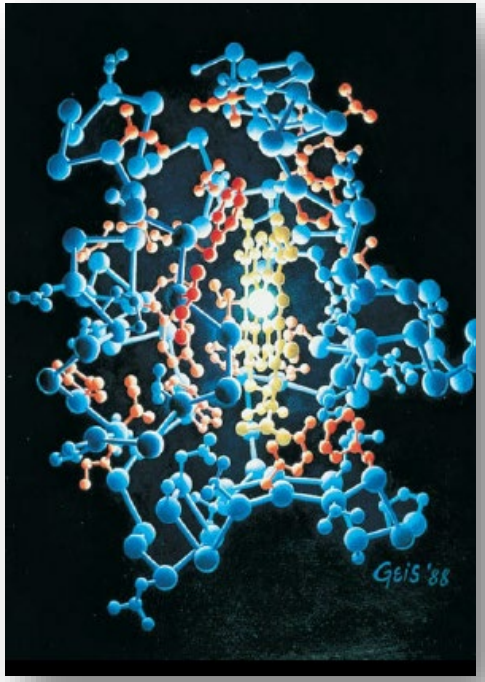
Presentation | 2021-12-14

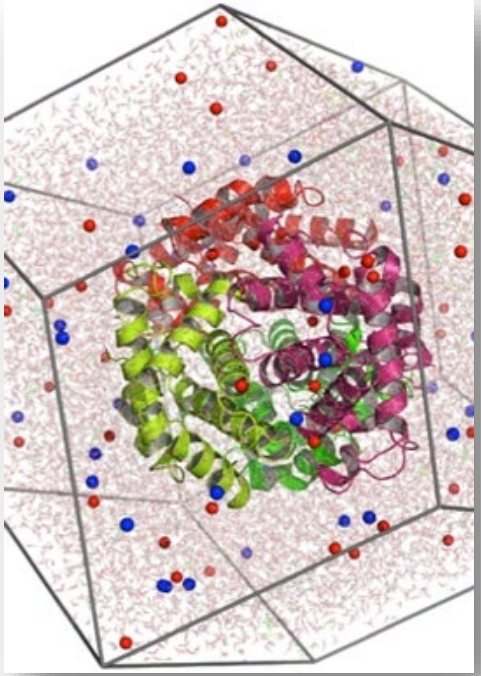
Molecular Dynamics Simulation: Principles, Algorithms and Applications

Yuzhe Wang
CCME, Peking University
wangyuzhe_ccme@pku.edu.cn

Outline



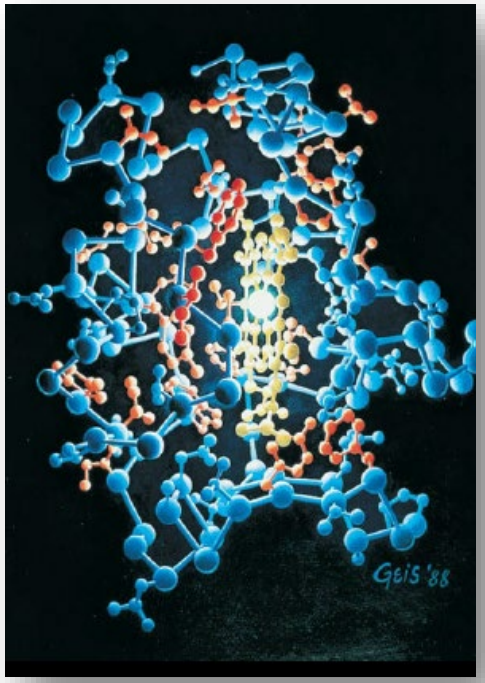
Background
- Proteins: Structure and Functions
- Why Molecular Dynamics Simulations?
- MD Simulation of Complex Systems



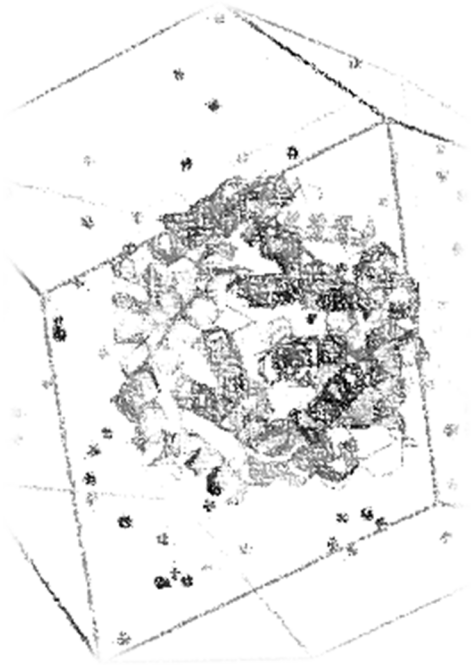
Principles
- Simulation of Dynamics of Particles
- Molecular Mechanics: Force Field
- Preparation to start a Simulation

Algorithms
- The Verlet (“leapfrog”) Algorithm
- Why Leapfrog Algorithm?
- Frontier: Deep Learning for Molecular Dynamics

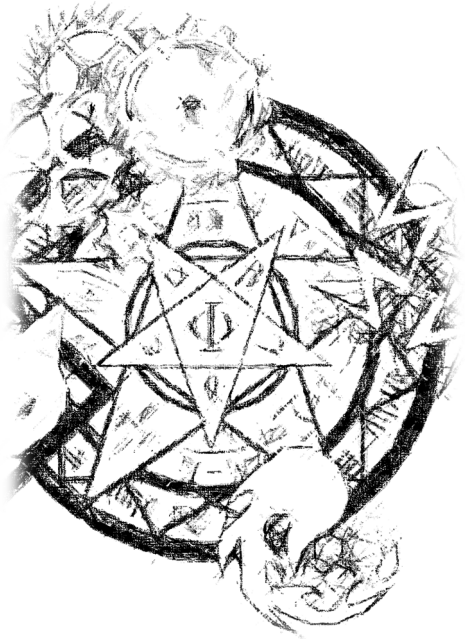
Outline



Background
- Proteins: Structure and Functions
- Why Molecular Dynamics Simulations?
- MD Simulation of Complex Systems



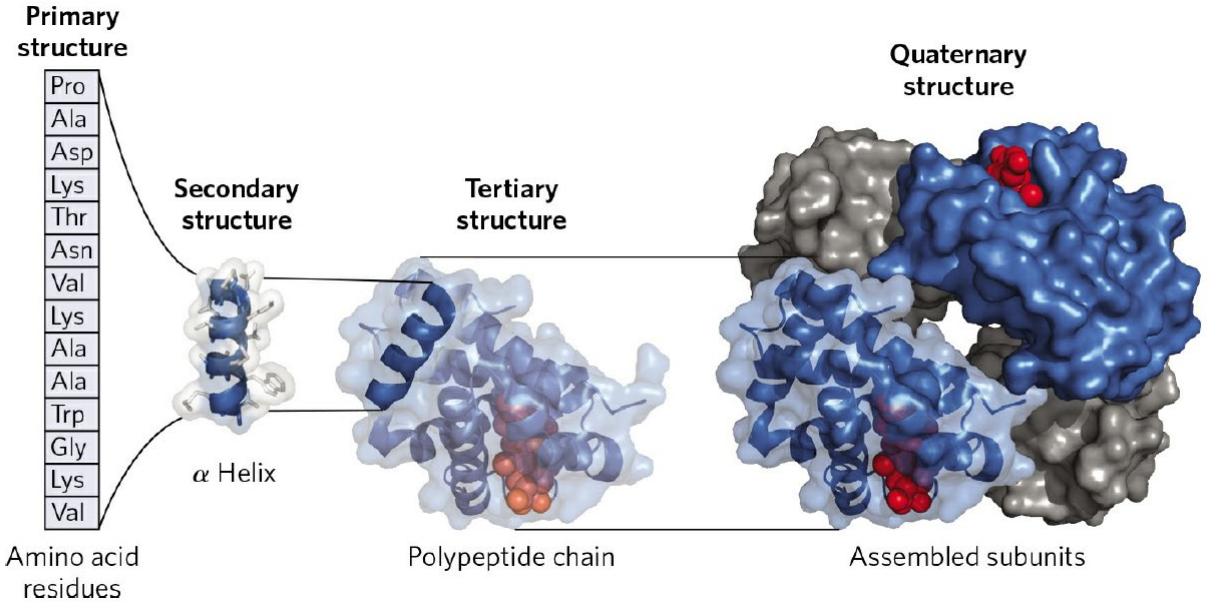
Principles
- Simulation of Dynamics of Particles
- Molecular Mechanics: Force Field
- Preparation to start a Simulation



Algorithms
- The Verlet ("leapfrog") Algorithm
- Why Leapfrog Algorithm?
- Frontier: Deep Learning for Molecular Dynamics

Proteins: Structure and Functions

- Protein Structure
  - **3-Dimensional** arrangement of atoms in a protein
  - **4 Levels** of Structure in Proteins

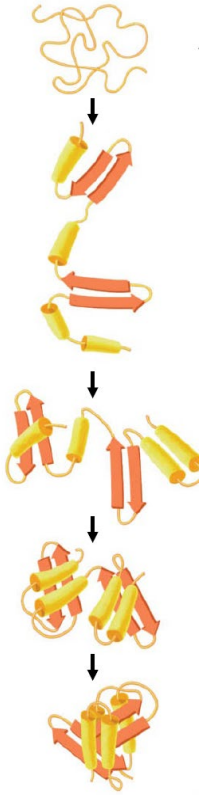


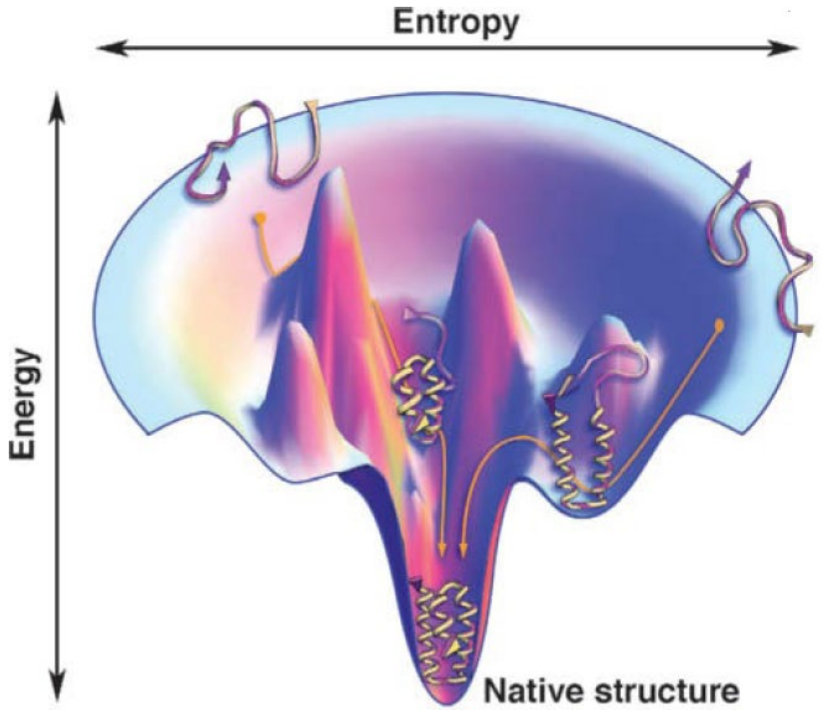
- Protein Functions
  - Antibodies, Enzymes, Messengers …
  - Determined by **Protein Structure**, which is generally difficult to obtain, especially *in vivo*

Nelson, D. L., Lehninger, A. L., & Cox, M. M. (2008). *Lehninger principles of biochemistry*. Macmillan.

Why Molecular Dynamics Simulations?

- Protein Structure Prediction: Dive into Protein Folding
  - Thermodynamics: **free-energy funnel**
  - Low energy conformation is **spontaneously** formed, consistent with the protein structure *in vivo*

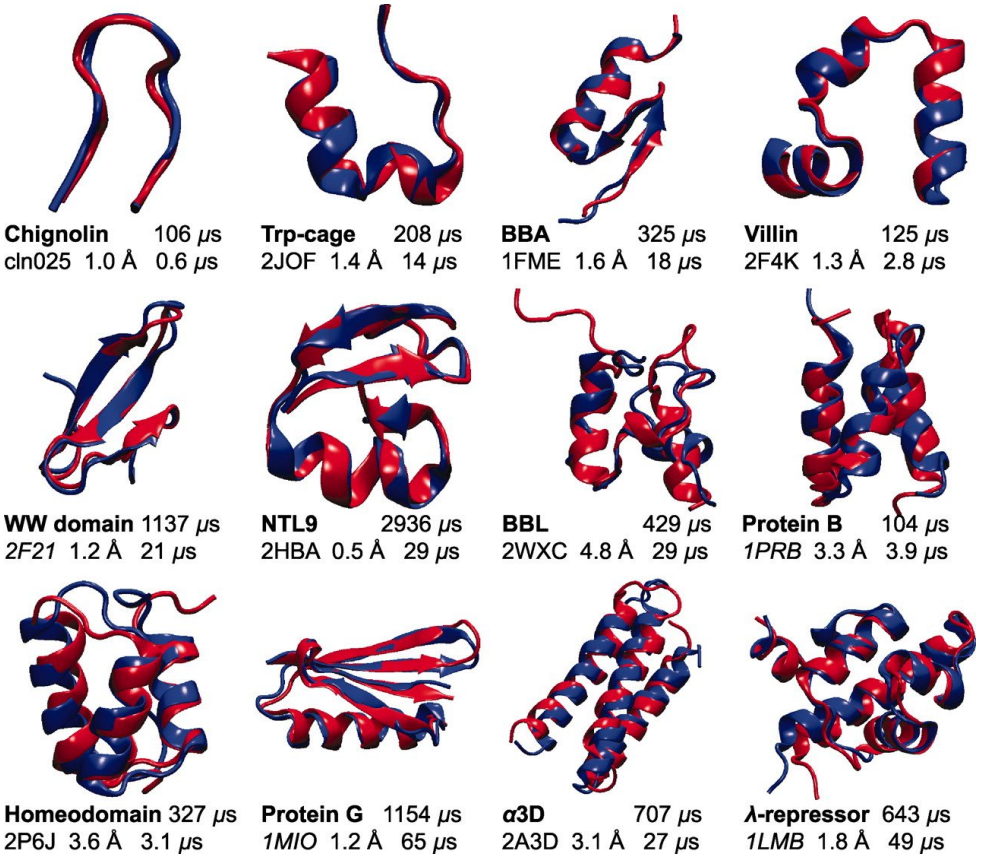




Nelson, D. L., Lehninger, A. L., & Cox, M. M. (2008). *Lehninger principles of biochemistry*. Macmillan.

Why Molecular Dynamics Simulations?

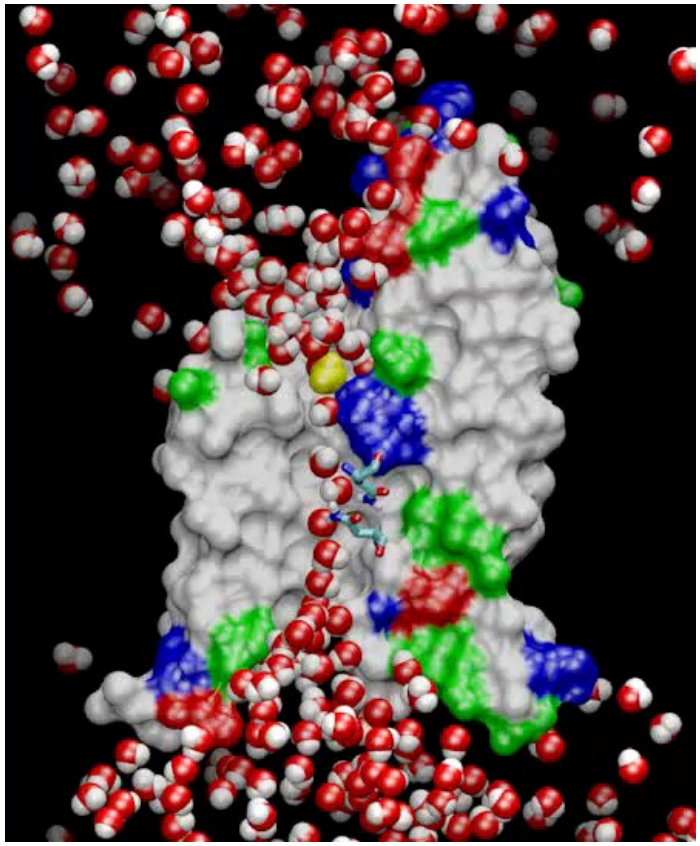
- Protein Structure Formation: Dive into Protein Folding
- Protein Structure Prediction: **Protein Folding Simulations**

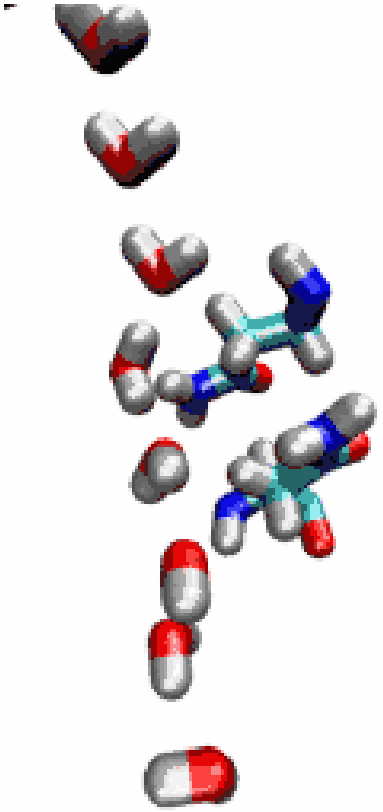


Lindorff-Larsen, K., Piana, S., Dror, R. O., & Shaw, D. E. (2011). How fast-folding proteins fold. *Science*, *334*(6055), 517-520.

MD Simulation of Complex Systems

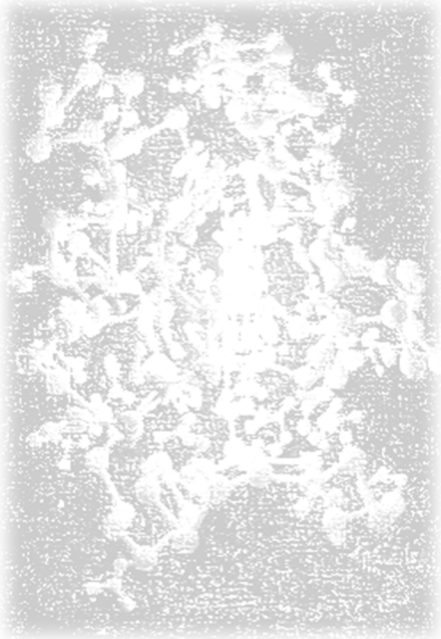
- MD Simulation: A glimpse of life processes inside our cells
  - A Fancy Example: the Selectivity of the **Aquaporin Water Channel** (*Science* 2002)



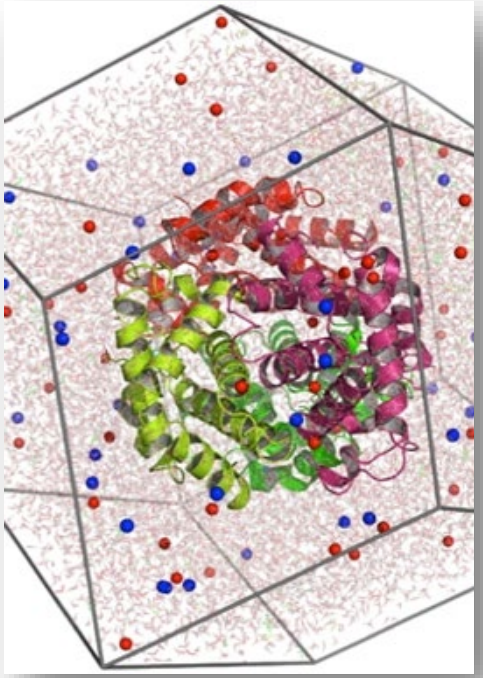


Tajkhorshid, E., Nollert, P., Jensen, M. Ø., Miercke, L. J., O'Connell, J., Stroud, R. M., & Schulten, K. (2002). Control of the selectivity of the aquaporin water channel family by global orientational tuning. *Science, 296* (5567), 525-530.

Outline



Background
- Proteins: Structure and Functions
- Why Molecular Dynamics Simulations?
- MD Simulation of Complex Systems



Principles
- Simulation of Dynamics of Particles
- Molecular Mechanics: Force Field
- Preparation to start a Simulation

Algorithms
- The Verlet ("leapfrog") Algorithm
- Why Leapfrog Algorithm?
- Frontier: Deep Learning for Molecular Dynamics

Simulation of Dynamics of Particles

- Newton's Law of Motion: **Second-order ODE**

$$m\frac{d^2r(t)}{dt^2} = F = -\nabla U(r)$$

- Equivalent **First-order ODEs**

$$\frac{dr(t)}{dt} = v(t)$$

$$m\frac{dv(t)}{dt} = F(t)$$

- Finite Difference Approximation comes to an **(Explicit) Euler Method**

$$v_{n+1} = v_n + \frac{F_n}{m}dt$$

$$r_{n+1} = r_n + v_n dt$$

- **N-body Problem**: for each particle i, we have

$$m_i\frac{d^2r_i(t)}{dt^2} = \sum_j F_{ij}(t) = -\sum_j \nabla_i U(|r_{ij}(t)|)$$

Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.

Molecular Mechanics: Force Field

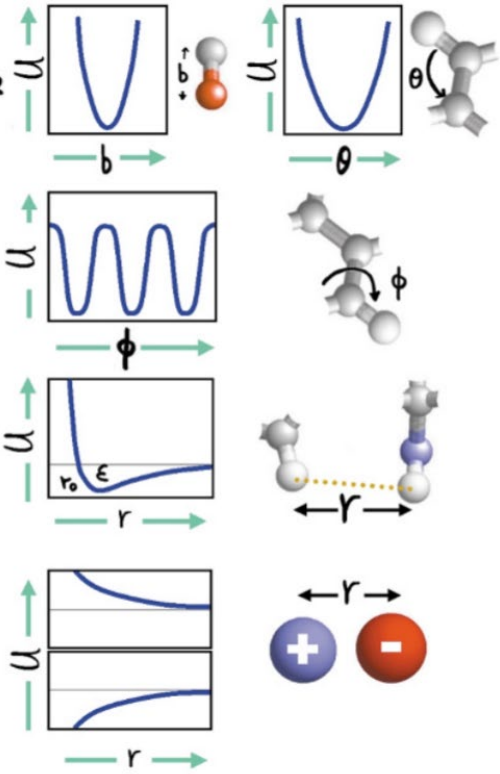
- Potential Energy Function $U = U_{bonded} + U_{non-bonded}$
- **Force Field**: the **function form** of the potential and a **set of parameters** for different types of atoms, bonds and interactions etc.
- A general (empirical) force field

$$U = \sum_{\text{All Bonds}} \tfrac{1}{2} K_b (b - b_0)^2 + \sum_{\text{All Angles}} \tfrac{1}{2} K_\theta (\theta - \theta_0)^2$$

$$+ \sum_{\text{All Torsion Angles}} K_\phi [1 - \cos(n\phi + \delta)]$$

$$+ \sum_{\text{All nonbonded pairs}} \epsilon \left[\left(\frac{r_0}{r}\right)^{12} - 2\left(\frac{r_0}{r}\right)^{6} \right]$$

$$+ \sum_{\text{All partial charges}} 332 q_i q_j / r$$



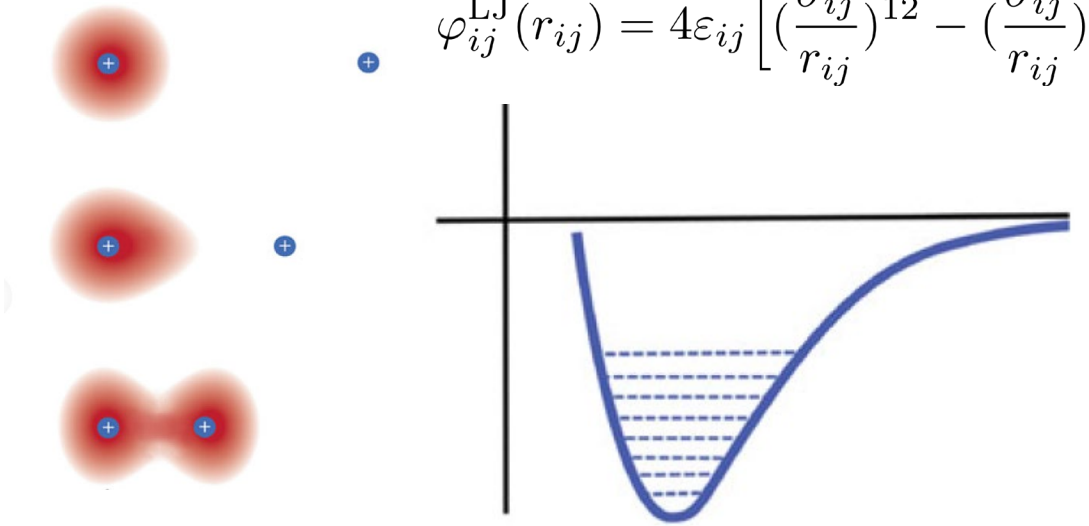
Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.

Levitt, M. (2001). The birth of computational structural biology. *Nature structural biology*, 8(5), 392-393.

Molecular Mechanics: Force Field

- Potential Energy Function $U = U_{bonded} + U_{non-bonded}$
- **Force Field**: the **function form** of the potential and a **set of parameters** for different types of atoms, bonds and interactions etc.
- A general (empirical) force field
- Van der Waals Interaction: **Lennard-Jones Potential**

$$\varphi_{ij}^{\mathrm{LJ}}(r_{ij}) = 4\varepsilon_{ij}\left[(\frac{\sigma_{ij}}{r_{ij}})^{12} - (\frac{\sigma_{ij}}{r_{ij}})^{6}\right]$$



London dispersion

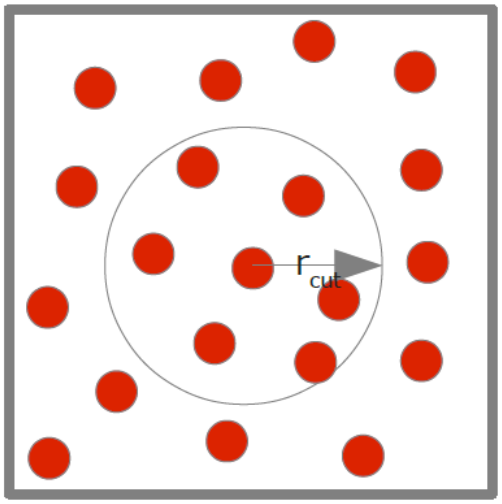
$$+\sum_{\text{All nonbonded pairs}} \varepsilon\left[\left(\frac{r_0}{r}\right)^{12} - 2\left(\frac{r_0}{r}\right)^{6}\right]$$

Pauli repulsion

Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.
Levitt, M. (2001). The birth of computational structural biology. *Nature structural biology*, 8(5), 392-393.

Molecular Mechanics: Force Field

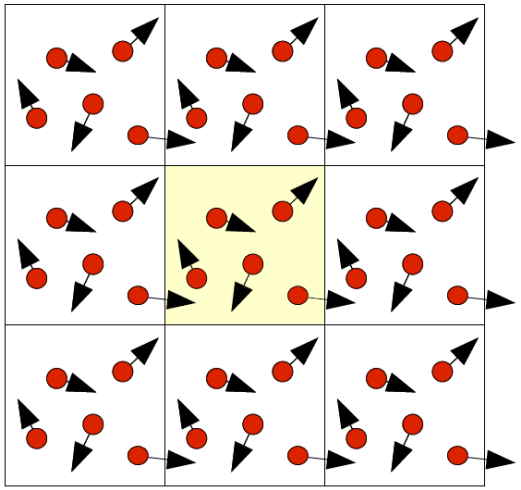
- Potential Energy Function $U = U_{bonded} + U_{non-bonded}$
- **Force Field**: the **function form** of the potential and a **set of parameters** for different types of atoms, bonds and interactions etc.
- A general (empirical) force field
- Van der Waals Interaction: **Lennard-Jones Potential**
- The Potential **Cut-off**
  - In principle pair potentials have **infinite** range
  - For potentials like Lennard-Jones potential the value quickly becomes negligible when atoms gradually separate
  - Apply an (empirical) **cut-off condition** r_{cut}

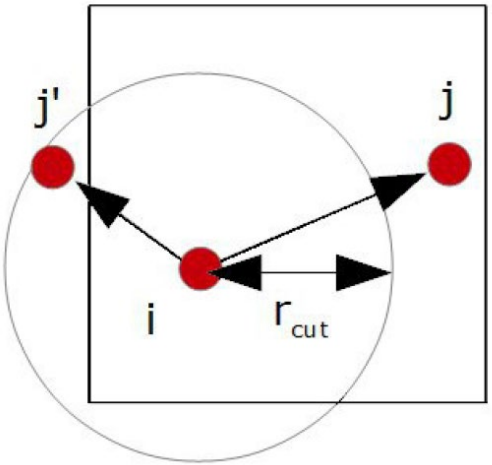


Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.
Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

Preparation to start a Simulation

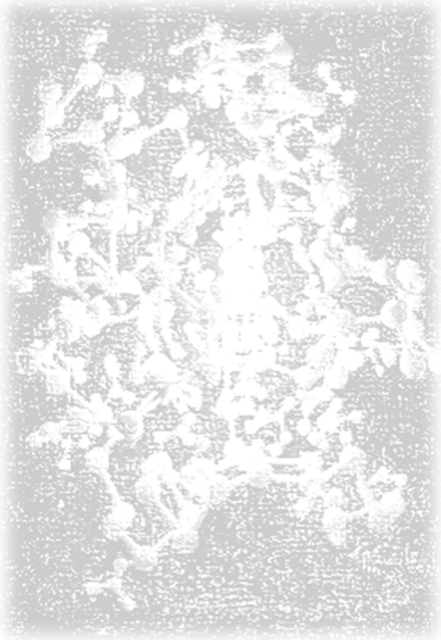
- The **Periodic Boundary Condition (PBC)**
  - The system is in fact **infinite** in extent
  - Idea: the **periodic cell** is assumed to be **replicate** in the same way as a unit cell in crystallography
  - Purpose: create a system **without surfaces**
- The **Minimum Image Convention**
  - The cut-off setting: Net interaction experienced by each atom includes contribution from only **one image** of the other atom in the system
  - Purpose: **only the closest images** of any two atoms contribute to the calculated energy / force
- The **Initial Conditions**
  - Initial **atom positions**: usually obtained by x-ray crystallography or cryo-em (NOT *in vivo*!)
  - Initial **velocities**: usually generated from a series of random seeds, then scaled to match the required system temperature



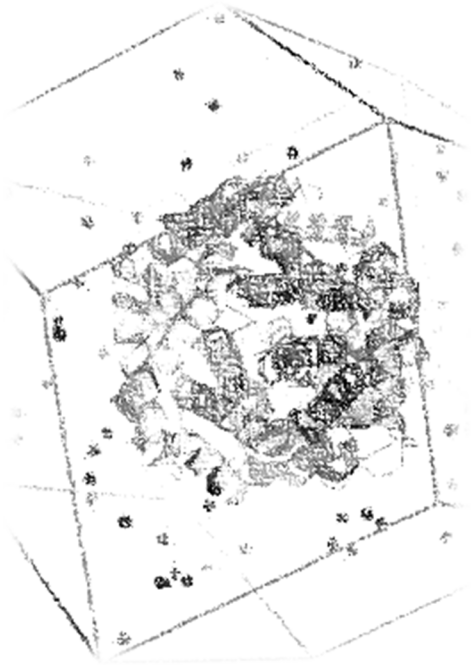


Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

Outline



Background
- Proteins: Structure and Functions
- Why Molecular Dynamics Simulations?
- MD Simulation of Complex Systems



Principles
- Simulation of Dynamics of Particles
- Molecular Mechanics: Force Field
- Preparation to start a Simulation

Algorithms
- The Verlet (“leapfrog”) Algorithm
- Why Leapfrog Algorithm?
- Frontier: Deep Learning for Molecular Dynamics

The Verlet ("leapfrog") Algorithm

- **Algorithm**

$$\boldsymbol{v}_{n+\frac{1}{2}} = \boldsymbol{v}_n + \frac{h}{2}\boldsymbol{M}^{-1}\boldsymbol{F}_n$$

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + h\boldsymbol{v}_{n+\frac{1}{2}}$$

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_{n+\frac{1}{2}} + \frac{h}{2}\boldsymbol{M}^{-1}\boldsymbol{F}_{n+1}$$

where $h := \Delta t$ denotes the (fixed) time interval, and the force

$$\boldsymbol{F}_n = \boldsymbol{F}(\boldsymbol{r}_n) = -\nabla U(\boldsymbol{r}_n)$$

computed at the **end** of a given step is reused at the **start** of the following step, lead to a similar cost to Euler Method

- "Leapfrog"
  - Update positions and velocities at **interleaved** time points, staggered in such a way that they **"leapfrog"** over each other.
- ***Störmer's Rule***
  - Eliminate velocities to obtain

$$\boldsymbol{M}(\boldsymbol{r}_{n+1} - 2\boldsymbol{r}_n + \boldsymbol{r}_{n-1}) = -h^2\nabla U(\boldsymbol{r}_n)$$

Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.

The Verlet ("leapfrog") Algorithm

- **Brief Derivation** of leapfrog algorithm

Consider the Taylor Expansion in terms of the time interval h as

$$\boldsymbol{r}(t+h) = \boldsymbol{r}(t) + \dot{\boldsymbol{r}}(t)h + \ddot{\boldsymbol{r}}(t)\frac{h^2}{2!} + \dddot{\boldsymbol{r}}(t)\frac{h^3}{3!} + \ddddot{\boldsymbol{r}}(t)\frac{h^4}{4!} + O(h^5) \qquad (1)$$

take $t = nh$ where n is some integer, we obtain

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \dot{\boldsymbol{r}}_n h + \ddot{\boldsymbol{r}}_n \frac{h^2}{2!} + \dddot{\boldsymbol{r}}(t)\frac{h^3}{3!} + \ddddot{\boldsymbol{r}}(t)\frac{h^4}{4!} + O(h^5) \qquad (2)$$

Similarly, we have

$$\boldsymbol{r}_{n-1} = \boldsymbol{r}_n - \dot{\boldsymbol{r}}_n h + \ddot{\boldsymbol{r}}_n \frac{h^2}{2!} - \dddot{\boldsymbol{r}}(t)\frac{h^3}{3!} + \ddddot{\boldsymbol{r}}(t)\frac{h^4}{4!} + O(h^5) \qquad (3)$$

$(2) - (3)$ to obtain

$$\boldsymbol{r}_{n+1} - \boldsymbol{r}_{n-1} = 2\dot{\boldsymbol{r}}_n h + \dddot{\boldsymbol{r}}_n \frac{2h^3}{3!} + O(h^5) \qquad (4)$$

Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.
Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

The Verlet ("leapfrog") Algorithm

- **Brief Derivation** of leapfrog algorithm

Devide $2h$ through (4), by definition we have

$$\boldsymbol{v}_n = \frac{1}{2h}(\boldsymbol{r}_{n+1} - \boldsymbol{r}_{n-1}) = \dot{\boldsymbol{r}}_n + \dddot{\boldsymbol{r}}_n \frac{h^2}{3!} + O(h^4) \tag{5}$$

rearrange (2) to give

$$\boldsymbol{r}_{n+1} - \boldsymbol{r}_n = \left(\dot{\boldsymbol{r}}_n + \dddot{\boldsymbol{r}}_n \frac{h^2}{3!}\right) h + \ddot{\boldsymbol{r}}_n \frac{h^2}{2!} + O(h^4) \tag{6}$$

substitute (5) to (6) and divide throughout by h, we have

$$\frac{1}{h}(\boldsymbol{r}_{n+1} - \boldsymbol{r}_n) = \boldsymbol{v}_n + \ddot{\boldsymbol{r}}_n \frac{h}{2!} + O(h^3) \tag{7}$$

hence by Newton's Law of Motion, we obtain

$$\boldsymbol{v}_{n+\frac{1}{2}} = \boldsymbol{v}_n + \frac{h}{2}\boldsymbol{M}^{-1}\boldsymbol{F}_n + O(h^3) \tag{8}$$

Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.
Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

The Verlet ("leapfrog") Algorithm

- **Brief Derivation** of leapfrog algorithm

where we define

$$\boldsymbol{v}_{n+\frac{1}{2}} = \frac{\boldsymbol{r}_{n+1} - \boldsymbol{r}_n}{h} \tag{9}$$

$$\boldsymbol{v}_{n-\frac{1}{2}} = \frac{\boldsymbol{r}_n - \boldsymbol{r}_{n-1}}{h} \tag{10}$$

to represent the half time step velocities.
In conclusion of (8)(9)(10), we have the algorithm

$$\boldsymbol{v}_{n+\frac{1}{2}} = \boldsymbol{v}_n + \frac{h}{2}\boldsymbol{M}^{-1}\boldsymbol{F}_n \tag{11}$$

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + h\boldsymbol{v}_{n+\frac{1}{2}} \tag{12}$$

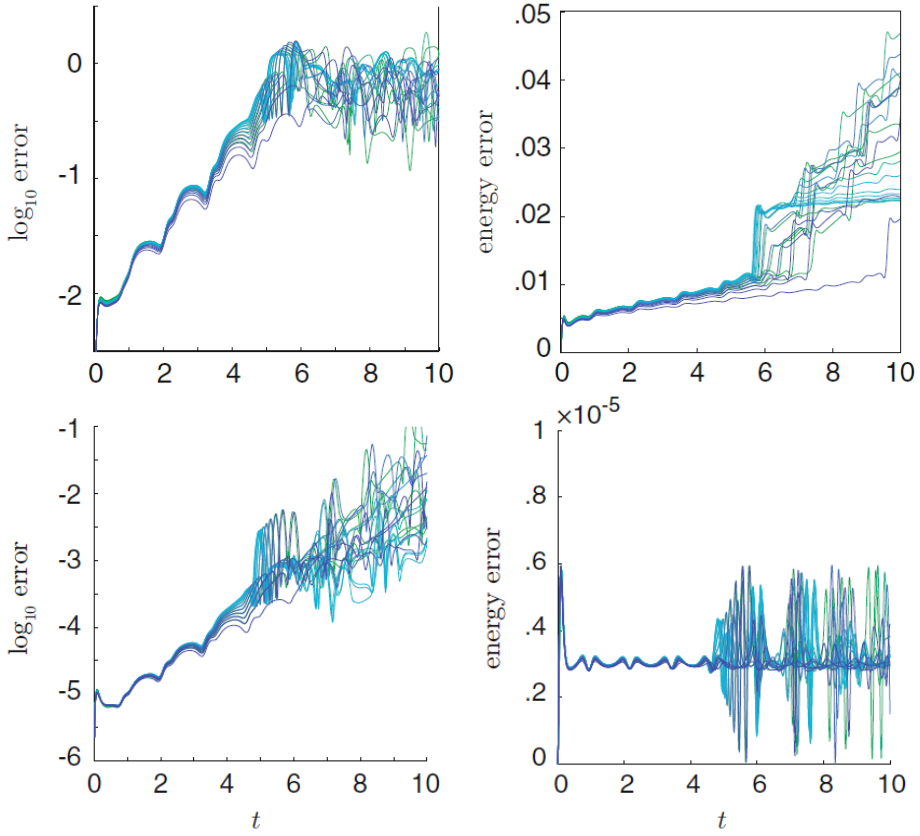
$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_{n+\frac{1}{2}} + \frac{h}{2}\boldsymbol{M}^{-1}\boldsymbol{F}_{n+1} \tag{13}$$

which is consistent with the algorithm stated in the beginning.

Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.
Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

Why Leapfrog Algorithm?

- Leapfrog Algorithm: **Second-Order Method** (Proof is omitted)
  - Why not apply higher-order Runge-Kutta Method (e.g. RK4)?
- **Error Growth** in Numerical Integration of ODEs



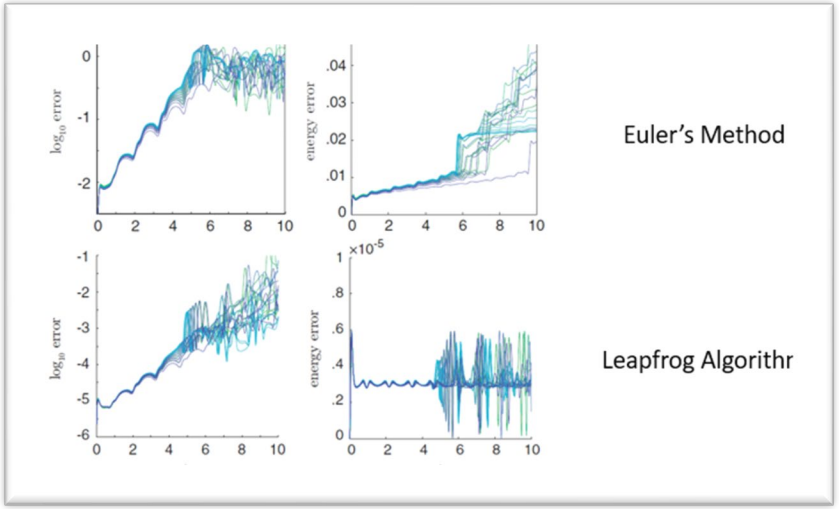
Euler's Method

Leapfrog Algorithm

Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.

Why Leapfrog Algorithm?

- Leapfrog Algorithm: **Second-Order Method** (Proof is omitted)
  - Why not apply higher-order Runge-Kutta Method (e.g. RK4)?
- **Error Growth** in Numerical Integration of ODEs
  - Molecular dynamics trajectories need to be very long compared to the time step, thus **how the error grows in long simulations** is essential
  - **Chaotic nature** of molecular dynamics, sensitivity to perturbations, the **global error** is expected to grow rapidly (exponentially) in time unavoidably
  - However, leapfrog algorithm has significant **stability of energy** that the energy error DO NOT accumulate in time
  - This is actually an extremely **non-trivial** property! Many common numerical algorithms of ODE is not suitable for MD Simulation for this reason.



Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.

Why Leapfrog Algorithm?

- A glimpse of the essence: **Geometric Integrators** for Hamiltonian Systems
  - Methods that conserve a **certain geometric property** *i.e.* **Symplecticness** of the phase flow
  - Mathematically, symplecticness is the property that $d\boldsymbol{p} \times d\boldsymbol{r}$ is a **conserved quantity**
  - Symplectic methods (e.g. leapfrog algorithm) preserve a **perturbed energy-like invariant** *i.e.* these algorithms possess **in-built long term stability** (proof is omitted)

Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.
Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

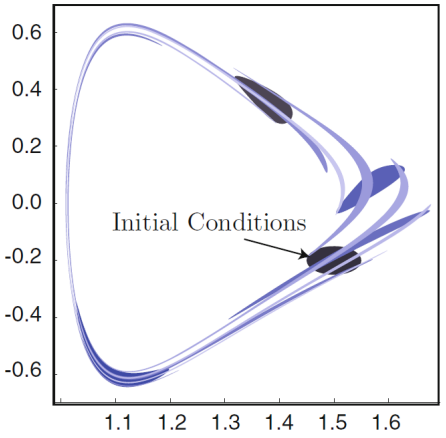
Why Leapfrog Algorithm?

- A glimpse of the essence: **Geometric Integrators** for Hamiltonian Systems
- The Dynamics of a **Hamiltonian system**
  - Hamilton's Equations

$$\frac{d\boldsymbol{q}}{dt} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{p}}$$
$$\frac{d\boldsymbol{p}}{dt} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{q}}$$

  - **Highly restricted** by conservation laws or volume preservation, violation of which leads to a **gradual corruption** of the solution
  - Volume Preserving Flows: **Liouville's Theorem**

$$\frac{\partial}{\partial t}\rho = -\{\rho, \mathcal{H}\}_{\mathrm{PB}} := -\sum_{i=1}^{f}\left(\frac{\partial \rho}{\partial q_i}\frac{dq_i}{dt} - \frac{\partial \rho}{\partial p_i}\frac{dp_i}{dt}\right)$$



Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.
Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

Why Leapfrog Algorithm?

- A glimpse of the essence: **Geometric Integrators** for Hamiltonian Systems
- The Dynamics of a **Hamiltonian system**
- Connection between **MD Simulations** and Hamiltonian System?
  - Quick intro to Statistical Mechanics: **Microcanonical Ensemble (NVE Ensemble)**, energy as a conserved quantity
  - Hamiltonian systems, satisfy Liouville Theorem
  - Hence **symplecticness** of algorithms is required
- The Symplecticness of **Leapfrog Algorithm**
  - The Symplectic nature of the leapfrog algorithm can be shown by deriving the algorithm **directly** from the **Liouville Equation** or **Hamilton's principle of least action**
  - (The latter) finally leads to

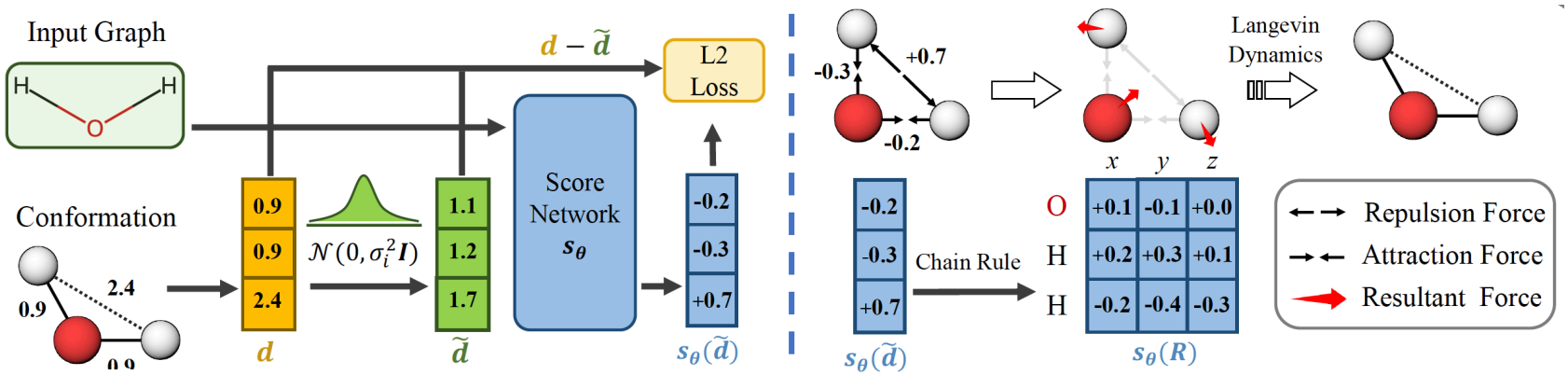
$$\boldsymbol{M}(\boldsymbol{r}_{n+1} - 2\boldsymbol{r}_n + \boldsymbol{r}_{n-1}) = -h^2 \nabla U(\boldsymbol{r}_n)$$

which is ***Störmer's Rule!***

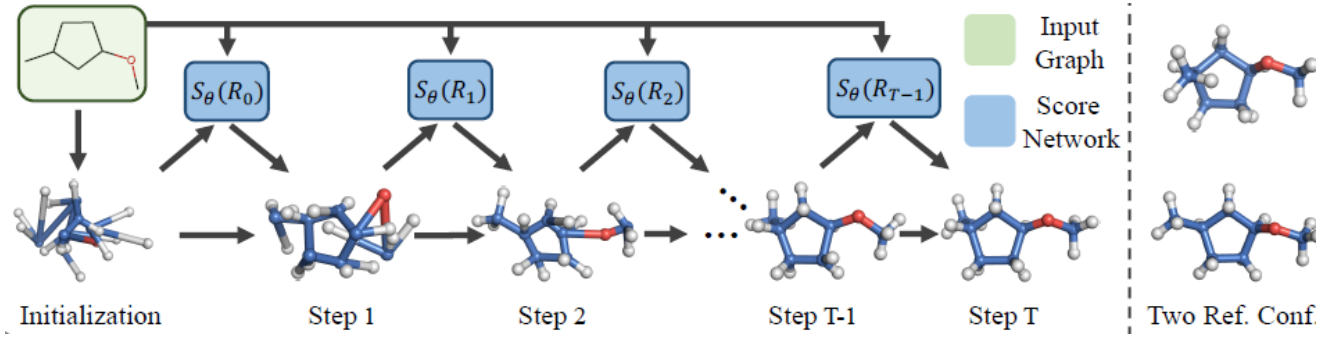
Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.
Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

Frontier: Deep Learning for Molecular Dynamics

- ConfGF (ICML'21)
  - Learning a Force Field via **Graph Isomorphism Network (GINs)**



  - Perform Dynamics Simulation (Conformation Optimization) via **Langevin Dynamics**



Shi, C., Luo, S., Xu, M., & Tang, J. (2021). Learning gradient fields for molecular conformation generation. *arXiv preprint arXiv:2105.03902.*

References

[1] Leimkuhler, B., & Matthews, C. (2016). *Molecular Dynamics*. Springer International PU.

[2] Smith, W. (2014). Elements of molecular dynamics. *GitLab. https://gitlab. com/DL_POLY_Classic/EoMD.*

[3] Leach, A. R., & Leach, A. R. (2001). *Molecular modelling: principles and applications*. Pearson education.

[4] Krauth, W. (2006). *Statistical mechanics: algorithms and computations* (Vol. 13). OUP Oxford.

[5] Rapaport, D. C., & Rapaport, D. C. R. (2004). *The art of molecular dynamics simulation*. Cambridge university press.

[6] Frenkel, D., Smit, B., & Ratner, M. A. (1996). *Understanding molecular simulation: from algorithms to applications* (Vol. 2). San Diego: Academic press.

[7] Nelson, D. L., Lehninger, A. L., & Cox, M. M. (2008). *Lehninger principles of biochemistry*. Macmillan.

Thank you

Yuzhe Wang
CCME, Peking University
wangyuzhe_ccme@pku.edu.cn