

# Monte Carlo

Goal: Utilizing Random Numbers.

Motivation: Obviously.

# Monte Carlo

- Direct
- Stratified
- Symmetric
- Importance
- Inverse Function
- Metropolis-Hastings
- Why Variance

# Monte Carlo

- Direct
- Stratified
- Symmetric
- Importance
- Inverse Function
- Metropolis-Hastings
- Why Variance


## 简单随机抽样 初中数学

All Images Videos Shopping News Web Maps : More

 hgxxw.net  
http://www.hgxxw.net › ... · [Translate this page](#) :

### 简单随机抽样

1、**简单随机抽样**：设一个总体的个体数为 $N$ ．如果通过逐个抽取的方法从中抽取一个样本，且每次抽取时各个个体被抽到的机会相等，就称这样的抽样为**简单随机抽样**．

 博客园  
https://www.cnblogs.com › zhgma... · [Translate this page](#) :

### 9.1.1 简单随机抽样- 贵哥讲数学

6 May 2023 — **抽样**调查的核心是样本的代表性，每个个体被抽到的概率相等，样本数据能够反应总体. (2)  
调查对象的全体称为总体，组成总体的每一个调查对象称为个体。从总体 ...

 百度文库  
https://wk.baidu.com › view · [Translate this page](#) :

### 用简单随机抽样方法估计全班同学的平均身高》优秀教学设计

初中数学七年级《数学活动：用**简单随机抽样** ... 活动的目的是在调查总体已知的情况下，让学生切身感受**简单随机抽样**中样本与总体之间的关系，从而体会**简单随机抽样**中的合理性和 ...

 233网校  
https://www.233.com › zxshuxue · [Translate this page](#) :

### 初中数学教师资格证面试教案模板：《简单随机抽样》

29 Mar 2020 — **简单随机抽样**的含义：一般地，设一个总体有 $N$ 个个体，从中逐个不放回地抽取 $n$ 个个体作为样本( $n \leq N$ )，如果每次抽取时总体内的各个个体被抽到的机会都相等，则这种 ...

# Monte Carlo

- Direct
- Stratified
- Symmetric
- Importance
- Inverse Function
- Metropolis-Hastings
- Why Variance

## 2.1.3

### 分层抽样

当总体由有明显差别的几部分组成时,为了使抽取的样本更好地反映总体的情况,常采用分层抽样.将总体中各个个体按某种特征分成若干个互不重叠的几部分,每一部分叫做层,在各层中按层在总体中所占比例进行简单随机抽样或系统抽样,这种抽样方法叫做**分层抽样**.

例如,某中学高中学生有 900 名,为了考察他们的体重状况,打算抽取容量为 45 的一个样本.已知高一有 400 名学生,高二有 300 名学生,高三有 200 名学生.采用分层抽样,样本容量与总体容量的比为  $45:900=1:20$ ,所以在高一、高二、高三 3 个层面上取的学生数分别为  $\frac{400}{20}$ ,  $\frac{300}{20}$ ,  $\frac{200}{20}$ , 即分别抽取 20, 15, 10 名学生.当有些层面上抽取的学生数用除法算出的结果不是整数时,可作适当的细微调整,使抽取的学生数为整数.假如上例中高一、高二、高三的学生数分别为 402, 296, 202, 则这三个层面上取的学生数用除法计算分别为  $\frac{402}{20}=20.1$ ;  $\frac{296}{20}=14.8$ ;  $\frac{202}{20}=10.1$ , 每层还是分别按 20, 15, 10 名学生抽取.在 3 个层面上抽样时,可以采用简单随机抽样方法.

分层抽样的优点是,使样本具有较强的代表性,而且在各层抽样时,又可灵活地选用不同的抽样法.因此,分层抽样应用比较广泛.



#### 练习 A

1. 某校高一学生共 500 名,经调查,喜欢数学的学生占全体学生的 30%,不喜欢数学的人数占 40%,介于两者之间的学生占 30%.为了考查学生的期中考试的数学成绩,如何用分层抽样抽取一个容量为 50 的样本.
2. 某公司有员工 500 人,其中不到 35 岁的有 125 人,35~49 岁的有 280 人,50 岁以上的有 95 人.为了调查员工的身体健康状况,从中抽取 100 名员工,用分层抽样应当怎样抽取?

# Monte Carlo

- Direct
- Stratified
- Symmetric
- Importance
- Inverse Function
- Metropolis-Hastings
- Why Variance

撒点算圆周率：  
普通 vs 分层 vs 对称

# Monte Carlo

- Direct
- Stratified
- Symmetric
- **Importance**
- Inverse Function
- Metropolis-Hastings
- Why Variance

重要性抽样 (Importance Sampling)

# Monte Carlo

- Direct
- Stratified
- Symmetric
- Importance
- Inverse Function
- Metropolis-Hastings
- Why Variance

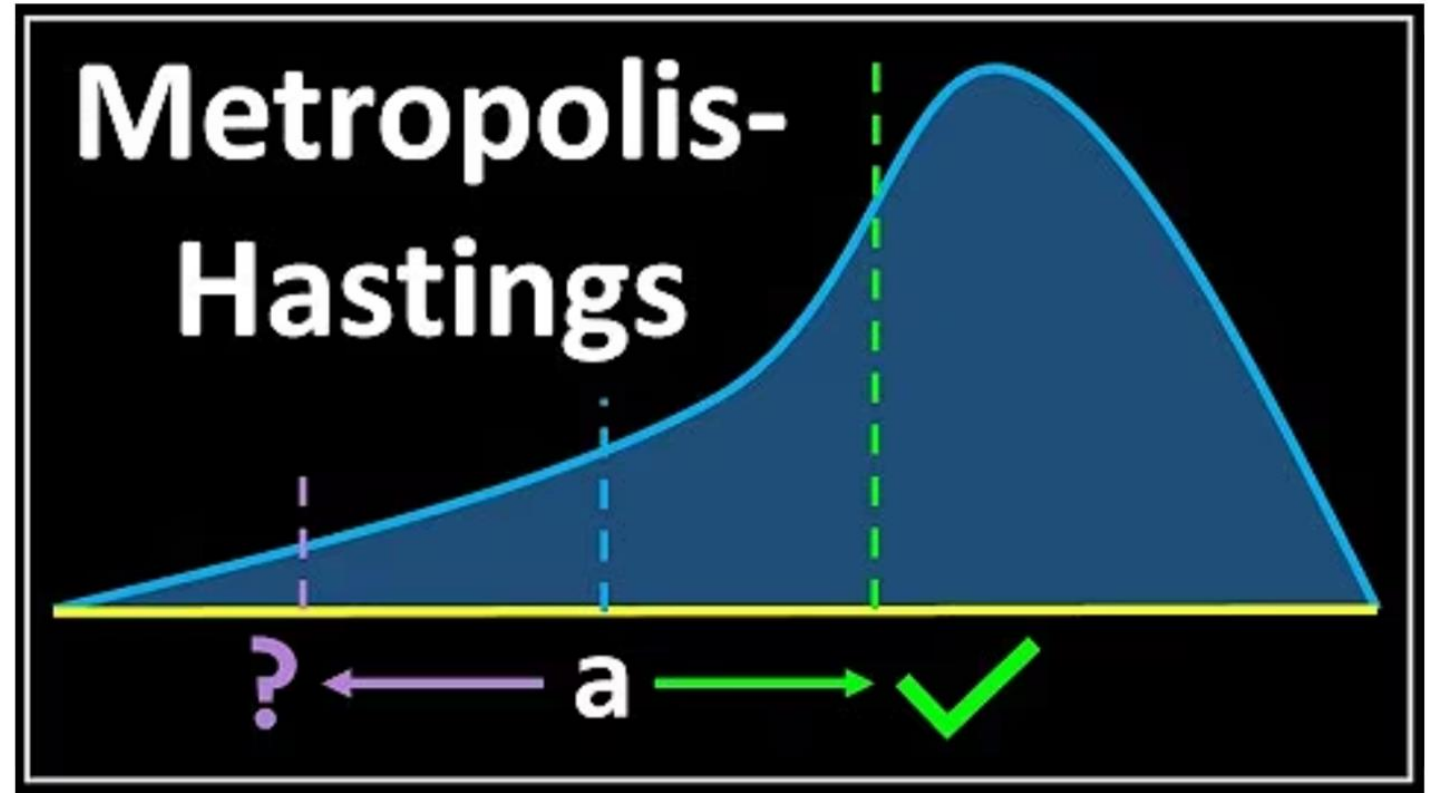
$$P(\xi \leq x) = P(F(\xi) \leq F(x)) = P(\eta \leq F(x)) = F(x)$$

$$\eta = F(\xi) \text{ and } \eta \sim U(0, 1)$$

**Can Get Any Distribution! (In Theory)**

# Monte Carlo

- Direct
- Stratified
- Symmetric
- Importance
- Inverse Function
- Metropolis-Hastings
- Why Variance



**Can Get Any Distribution!**



# Monte Carlo

- Direct
- Stratified
- Symmetric
- Importance
- Inverse Function
- Metropolis-Hastings
- Why Variance

Metropolis-Hastings的数学推导

Metropolis-Hastings的代码实现

为什么Metropolis-Hastings应  
和重要性采样联用?

为什么低维也要用Metropolis-  
Hastings?

**Can Get Any Distribution!**

# Monte Carlo

- Direct
- Stratified
- Symmetric
- Importance
- Inverse Function
- Metropolis-Hastings
- **Why Variance**

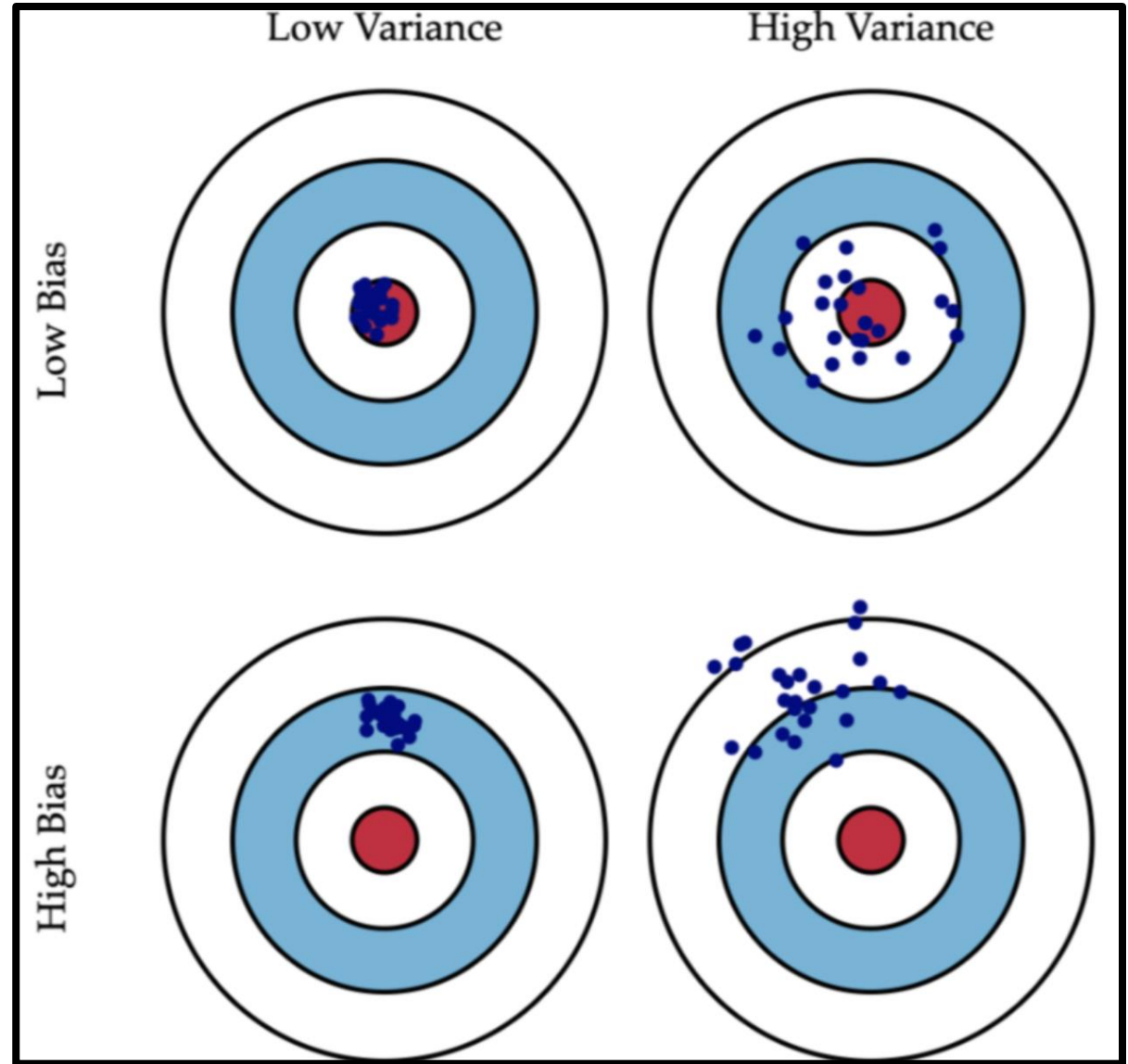


Figure From Internet

# Monte Carlo | Trivial

- Generating Random Number in Uniform(0, 1): 平方取中法 ([Middle Square Method](#))、线性同余法 ([Linear Congruential Generator](#))
- From Uniform(0, 1) to Any Arbitrary Distribution: 叠加原理、反函数法 ([Inverse-Sampling](#))、Metropolis-Hastings、变换、舍选 ([Accept-Reject](#))、复合。

现在的 MATLAB, Numpy, Julia, Mathematica 产生随机数的时候真的会用平方取中法和线性同余法吗？



Thought for a couple of seconds

不，目前的 MATLAB、NumPy、Julia 和 Mathematica 在生成随机数时通常**不**使用平方取中法（Middle Square Method）或简单的线性同余法（Linear Congruential Generator, LCG）。这些平台大多采用更先进和高效的伪随机数生成算法，以确保生成的随机数具有更好的统计特性和更高的计算效率。以下是各个平台常用的随机数生成器：

#### 1. MATLAB:

- **默认生成器:** 梅森旋转算法（Mersenne Twister）。
- **其他选项:** 包括组合递归法（Comb Recursive）、多重递归法（Multiple Recursive）、ISAAC、Threefry 等。

#### 2. NumPy:

- **旧版本:** 使用梅森旋转算法。
- **新版本:** 采用 PCG64（Permuted Congruential Generator）等更现代的生成器，通过 `numpy.random.Generator` 接口进行管理。

#### 3. Julia:

- **默认生成器:** 梅森旋转算法（Mersenne Twister）。
- **其他选项:** 包括 PCG、XORSHIFT 等，可以通过 `Random` 模块进行选择和管理。

#### 4. Mathematica:

- **默认生成器:** 使用高度优化和统计性能优良的随机数生成算法，具体实现细节通常由 Wolfram Research 内部维护，不公开详细信息，但绝不使用平方取中法或简单的线性同余法。

这些现代生成器相比于平方取中法和简单的线性同余法，具有更长的周期、更好的分布特性以及更高的安全性（在需要时），因此被广泛应用于科学计算、统计分析、模拟仿真等领域。

**总结:** 平方取中法和简单的线性同余法由于其统计性能较差和周期较短，已不再被主流的数学和科学计算软件作为默认的随机数生成方法使用。

不，当前这些软件通常使用更现代、性能更好的算法（如梅森旋转算法等）来生成随机数，而不采用平方取中法或简单的线性同余法。

我在教材上看到，蒙特卡洛方法，想从任意分布中抽样时，可以用反函数法、变换法、舍选法、复合函数法，我很好奇现在的matlab和python的从指数分布、正态分布等抽样的库函数是否真的用这些方法。



Thought for a few seconds

确实，蒙特卡洛方法中提到的反函数法、变换法、舍选法（拒绝采样）、复合函数法等都是从任意分布中抽样的基本方法。然而，现代编程语言中的标准库函数（如 MATLAB 和 Python 的 NumPy 库）在实现从特定分布（如指数分布、正态分布等）的抽样时，通常采用更高效和优化的算法，而不仅仅依赖于这些基本方法。以下是一些具体的实现细节：

## 1. 正态分布 (Normal Distribution)

- **Box-Muller 变换**：这是一个经典的方法，通过将两个均匀分布的随机数转换为两个独立的标准正态分布随机数。虽然 Box-Muller 方法简单直观，但它涉及到三角函数的计算，计算效率相对较低。
- **Marsaglia 极坐标法 (Marsaglia Polar Method)**：这是对 Box-Muller 方法的优化，避免了直接计算三角函数，通过极坐标变换提高了计算效率。
- **Ziggurat 算法**：这是现代库中常用的高效算法，能够快速生成大量的正态分布随机数。Ziggurat 算法通过分割概率密度函数的区域，并利用查找表和预计算的数据，大幅提升了生成速度。

库实现：

- **Python (NumPy)** 和 **MATLAB** 通常使用类似 Ziggurat 的方法来生成正态分布随机数，因为这种方法在速度和效率上优于基本的 Box-Muller 或 Marsaglia 方法。

[Link](#)

# Finite Diff

Goal: Solve Differential Equations.

Motivation: Obviously.

# Finite Element

Goal: Solve Differential Equations.

Motivation: Calculus of Variations.

# Finite Diff vs Finite Elem

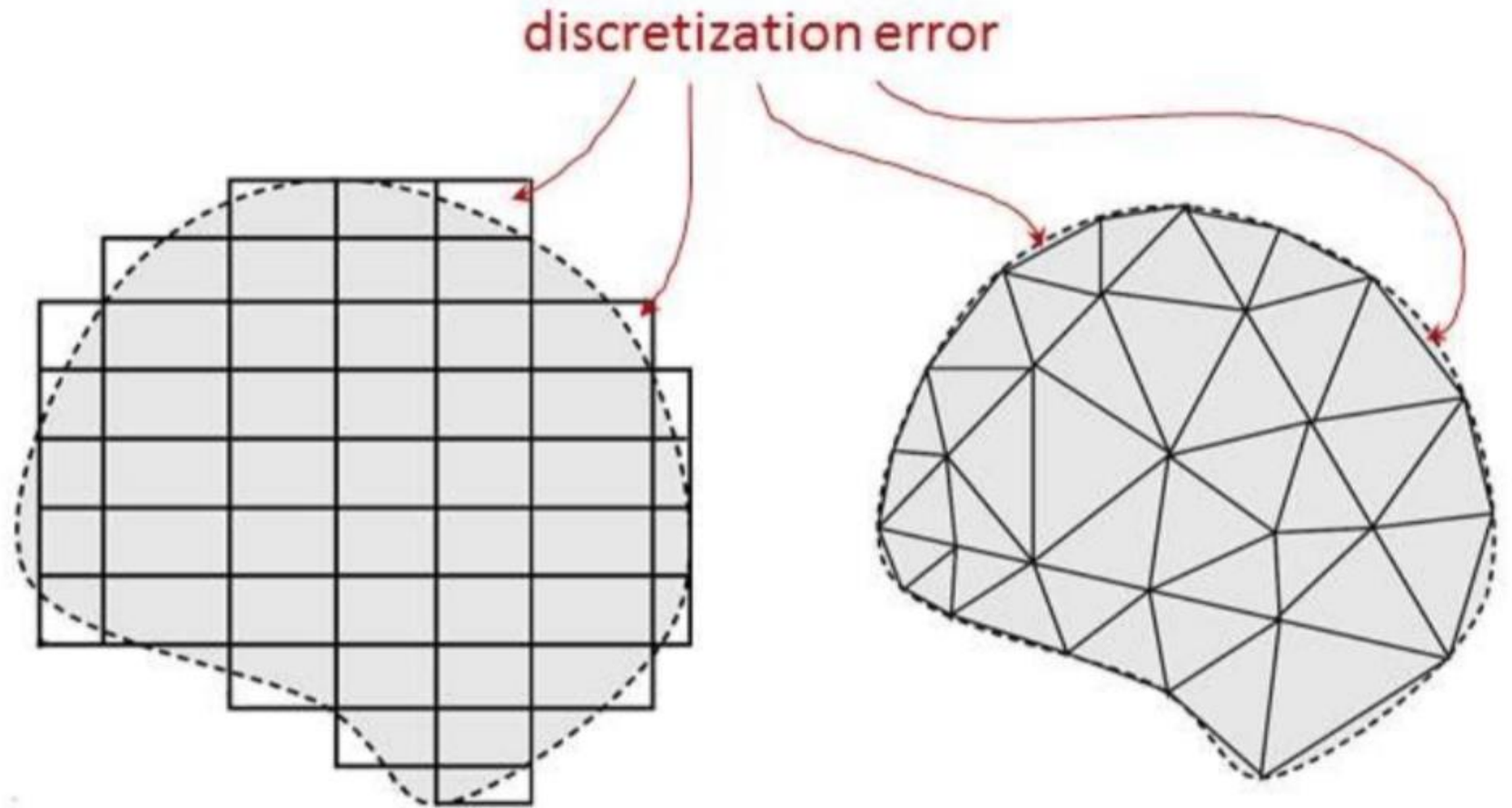


Figure From Internet

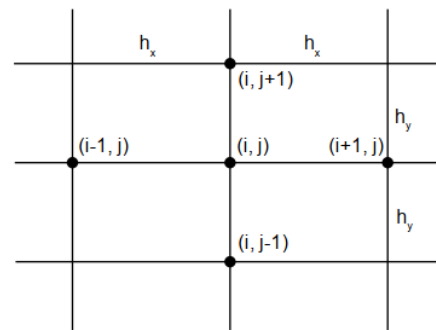


# Finite Diff

## 数值求解偏微分方程

五点差分法（或菱形格式，两个意思是一样的）求解下面的偏微分方程：

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + f\phi = q$$



$h_x$  是  $x$  方向的步长,  $h_y$  是  $y$  方向的步长, 下标  $(i, j)$  代表在格点  $(i, j)$  上函数的取值。

$$\left. \frac{\partial^2 \phi}{\partial x^2} \right|_{(i,j)} \rightarrow \frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{h_x^2}$$

$$\left. \frac{\partial^2 \phi}{\partial y^2} \right|_{(i,j)} \rightarrow \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{h_y^2}$$

所以

$$\frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{h_x^2} + \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{h_y^2} + f_{i,j}\phi_{i,j} = q_{i,j}$$

特殊情况  $h_x = h_y = h, f = 0$ , 那就转化成泊松方程了

$$\phi_{i,j} = \frac{1}{4}(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1} - h^2 q_{i,j})$$

如果题目问你的是拉普拉斯方程, 那么  $q_{i,j}$  也没了（但愿你们没有忘记数理方程）

# Finite Elem

## 五、有限元素方法

以满足第一类边界条件的二维平面场泊松方程为例：

$$\begin{cases} \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = -\frac{\rho}{\epsilon} \\ \phi|_L = \phi_0 \end{cases}$$

首先将场域 $D$ 划分为三角形单元，并对各个元素节点编号，如元素 $e$ 的三个顶点编号逆时针顺序取为 $i, j, m$ 。场域划分后，总的泛函等于所有三角形单元泛函之和：

$$I(\phi) = \sum_{e=1}^{e_0} I_e(\phi^{(e)}) = I_1(\phi) - I_2(\phi) = \frac{1}{2} \sum_{e=1}^{e_0} (\Phi)_e^T (K)_e (\Phi)_e - \sum_{e=1}^{e_0} (\Phi)_e^T (P)_e$$

泛函取极值：

$$\frac{d}{d\phi_i} (I(\phi)) = 0$$

得：

$$(K)(\Phi) = (P)$$

**Ritz法 和 FEM 的关系？**

**理论上怎么保证 Ritz法 是对的呢？**

# 叠加来逼近

## 2. Rayleigh–Ritz 法与有限元法都在对基函数做线性叠加吗？

**简短回答：**是的，二者从本质上都是在选定一个有限维函数空间（由一组基函数张成），然后用该有限维子空间的线性组合来逼近未知解。

### 2.1 Rayleigh–Ritz 法 (Ritz 法)

- Rayleigh–Ritz 法在求解变分问题时，先选取一组 **全局** 的（往往在整个区域都起作用的）基函数  $\{\phi_1, \phi_2, \dots, \phi_N\}$ ，令近似解

$$u_N(x) = \sum_{j=1}^N \alpha_j \phi_j(x).$$

- 再通过最小势能原理（或更一般的变分原理），求出系数  $\alpha_j$ 。这就是\*\*\*基函数的线性叠加\*\*\*。

### 2.2 有限元法 (FEM)

- 有限元法同样选取了一个有限维子空间  $V_h \subset V$ ，只是这个子空间通常是由**局部支撑**的分片多项式基函数  $\{\phi_1^h, \dots, \phi_{N_h}^h\}$  构成。同样令

$$u_h(x) = \sum_{j=1}^{N_h} \alpha_j \phi_j^h(x).$$

- 通过投影（Galerkin）或最小化能量泛函等方式得到方程组，解出  $\alpha_j$ 。依旧是\*\*\*基函数的线性叠加\*\*\*。

二者的**主要区别**在于：

- 基函数形式：**Rayleigh–Ritz 法中常用的基函数通常是全局性的，如三角函数、多项式、正交多项式等；有限元法则采用“分片多项式 + 局部支撑”。
- 适用的几何和工程需求：**有限元法通过网格剖分，可适应复杂几何和大规模计算；Rayleigh–Ritz 更常用于简化几何、理论分析或特定问题场景。

但它们都遵从同一个“基于基函数线性组合进行逼近”的思路。

# 历史的掠影

我们的问题是：即使各级近似极值的极限确为真正的极值，即

$$\lim_{n \rightarrow \infty} \Pi_n = \Pi \quad (4.4)$$

极小化序列(或极大化序列)的极限是不是真正的极限函数呢？即下式

$$\lim_{n \rightarrow \infty} f_n(x) = y(x) \quad (4.5)$$

是不是一定成立呢？

我们的答案是有条件的。也即是说：极小化序列(或极大化序列)的极限并不无条件地是真正的极值函数，这样的结论是维尔斯特拉斯(K. Weierstrass, 1870)首先用一个特例指出的<sup>[1]</sup>。

在维尔斯特拉斯的短文发表以前，人们(包括黎曼(Riemann)在内)没有对这个问题提出过怀疑，都认为极小化序列的极限必然给出真正的极限函数。维尔斯特拉斯用一个特例否定了当时为大家所公认的所谓“狄利克雷原理”。震动了当时数学界和理论物理学界，使黎曼、汤姆逊、狄利克雷等有关工作都失去了理论根据。

## 3. 弗立德里斯条件

人们经过了长期的努力，研究这些条件，得到了部分的解决。其中最有实用价值而又最有成效的工作是弗立德里斯(K. Friedrichs)在1934—1947年间完成的<sup>[2]</sup>。我们在这里不准备详细叙述弗立德里斯的理论证明，但将简单地引用他的结论，弗立德里斯所研究的问题都是有关以椭圆型线性微分方程为欧拉方程的泛函，例如

## § 4.3 立兹变分近似法

立兹变分近似法通称立兹(Ritz)法<sup>[3]</sup>。立兹法(1908)开创了弹性力学诸问题的近似解。这种近似解使用方便，也有一定准确性，大大促进了弹性理论在工程上的推广使用。当立兹法在1908年提出来的时候，并没有什么理论基础，只是在实践中证明它是行之有效的，它的理论基础是在人们多年努力之后，特别是在弗立德里斯(1934, 1946, 1947)<sup>[2a, 2b, 2c]</sup>的努力之后，才开始有了头绪。



# 构造的泛函

(A) 斯脱姆-刘维耳型的二阶常微分方程

斯脱姆-刘维耳型二阶常微分方程

$$Au = -\frac{d}{dx} \left[ p(x) \frac{du}{dx} \right] + q(x)y = f(x) \quad (4.20)$$

的泛函

$$\Pi(u) = (Au, u) - 2(f, u) \quad (4.21)$$

其中  $A$  为算子

$$A = -\frac{d}{dx} \left[ p(x) \frac{d}{dx} \right] + q(x) \quad (4.22a)$$

$(Au, u)$ ,  $(f, u)$  分别为  $Au, u$ , 以及  $f, u$  的内积.

泊桑方程可以写成

$$Au = -\left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f(x, y), \quad (4.25)$$

当  $f(x, y) = 0$  时, 上式称为拉普拉斯方程. 泊桑方程 (4.25) 式的泛函也可以写成 (4.21) 式的形式, 其中

$$A = -\left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \quad (4.26a)$$

$$(Au, u) = -\iint_S \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) u dx dy \quad (4.26b)$$

$$(f, u) = \iint_S f(x, y) u dx dy \quad (4.26c)$$

# 正定的证明

## § 4.2 泛函 $(Au, u)$ 的正定性, 泛函的极值和极值函数

现在让我们逐一检查前一节所列举的泛函  $(Au, u)$  都是正定的, 并证明泛函  $\Pi(u)$  的极值都给出满足欧拉方程的极值函数.

### 1. 斯脱姆-刘维耳型的二阶常微分方程

首先证明斯脱姆-刘维耳型的二阶常微分方程 (4.20) 式的泛函,  $(Au, u)$  是正定的. 我们研究的边界条件为固定边界条件,

$$u(x_1) = u(x_2) = 0 \quad (4.45)$$

通过分部积分, 并利用边界条件 (4.45) 式, 从 (4.22b) 式得

### 3. 泊桑方程(狄立克雷问题, 牛曼问题, 混合边界问题)

关于泊桑方程的正定条件比较复杂, 下面主要是弗立德里斯的工作<sup>[2b]</sup>

泊桑方程的泛函 (4.21) 式在下面三类的边界条件下, 都能证明

$$(Au, u) = -\iint_S \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) u dx dy \quad (4.81)$$

是正定的. 这三类边界条件是

## 更多的理论

**简言之：**“Friedrichs 条件”及其相关不等式是**确保**双线性型  $a(\cdot, \cdot)$  在合适的 Sobolev 空间中满足“强制性”的常用工具；而 Lax–Milgram 定理要求的正是“连续性 + 强制性”。因此二者的关系是：

- **Lax–Milgram 定理** = 核心要求： $a$  连续 + 强制；
- **Friedrichs 条件/不等式** = 用于在典型 PDE 场景中“证明或确保强制性”的重要手段。

# Molecular Dynamics

Goal: Use Newton Laws to Simulate.

Motivation: Obviously.

# Newton

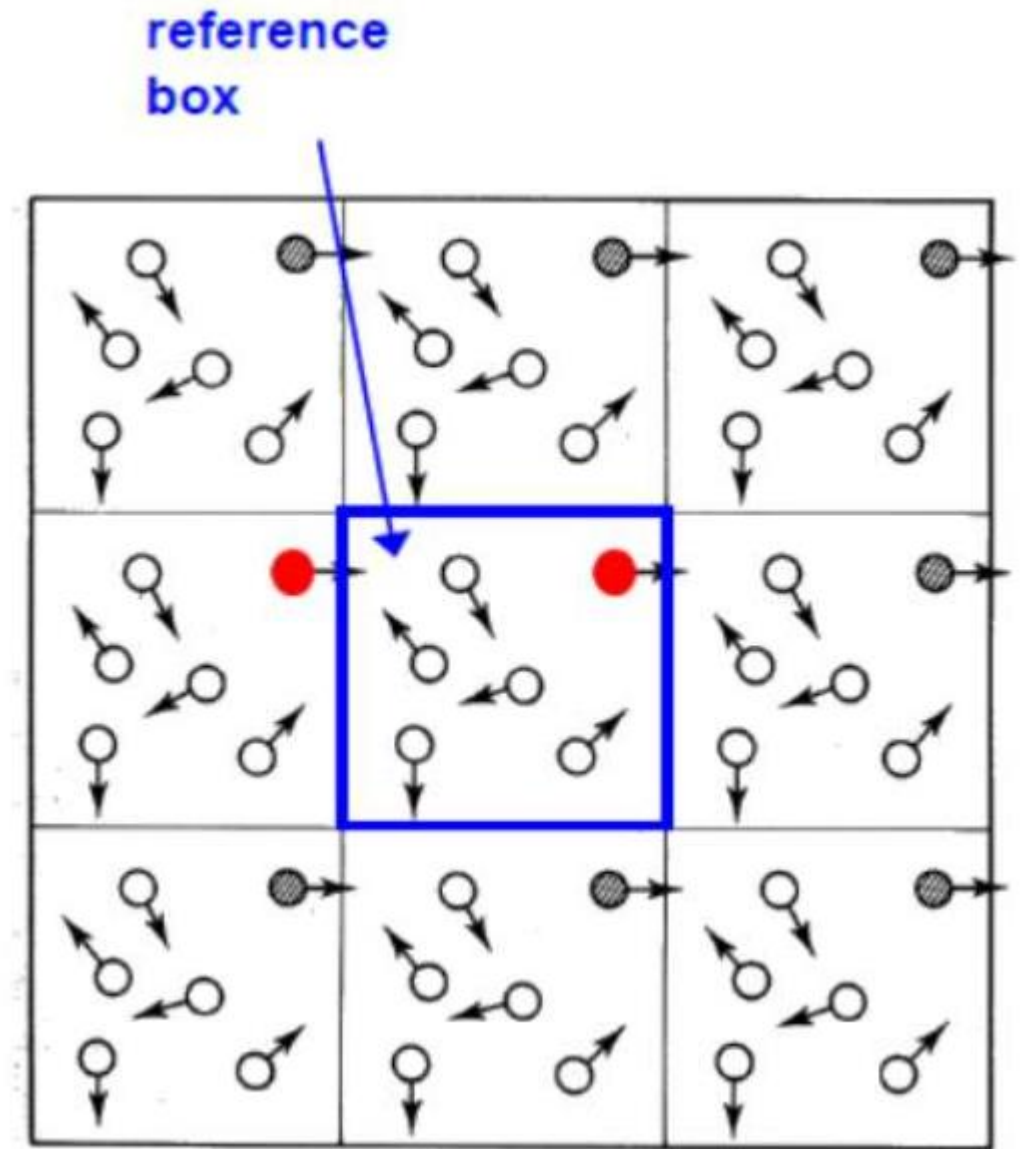
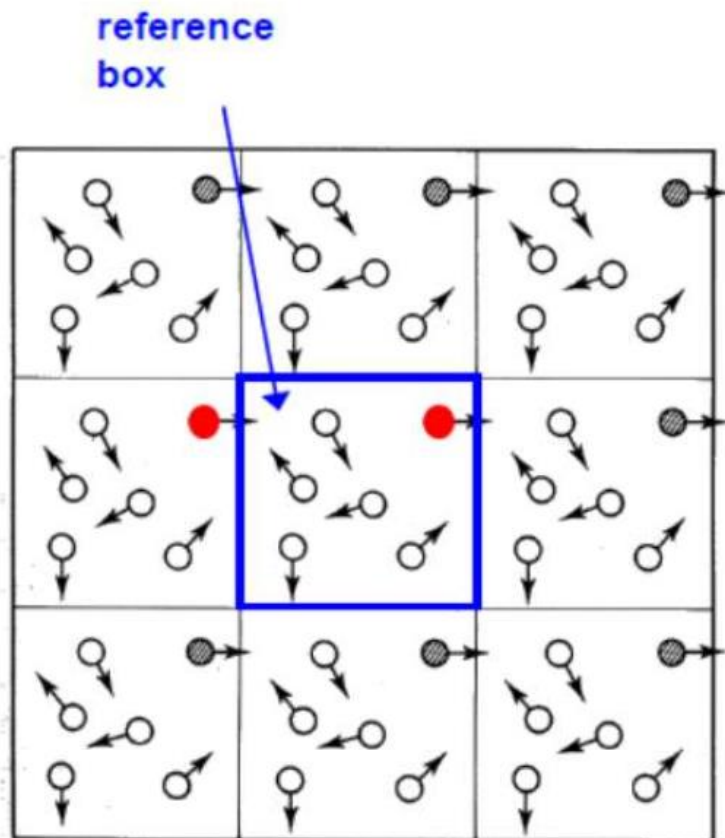


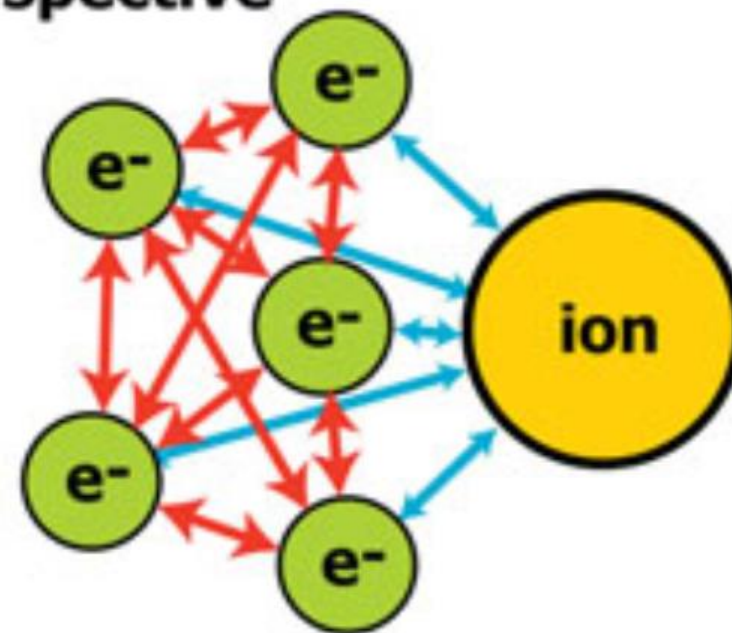
Figure From Zeying





$$\underline{F} = ma$$

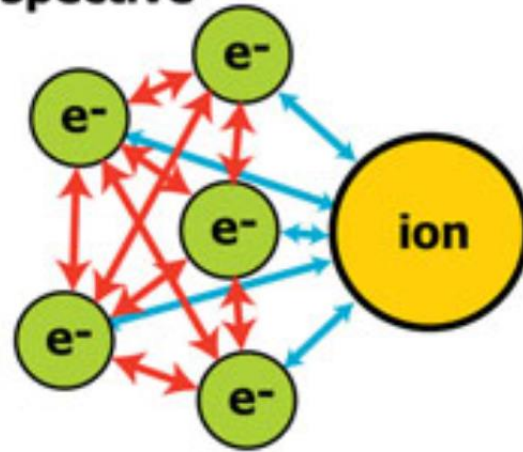
**Many-Body  
Perspective**



$$i\hbar \frac{\partial \Phi}{\partial t} = \underline{\hat{H}} \Phi$$

# QM- Hartree Fock

**Many-Body  
Perspective**



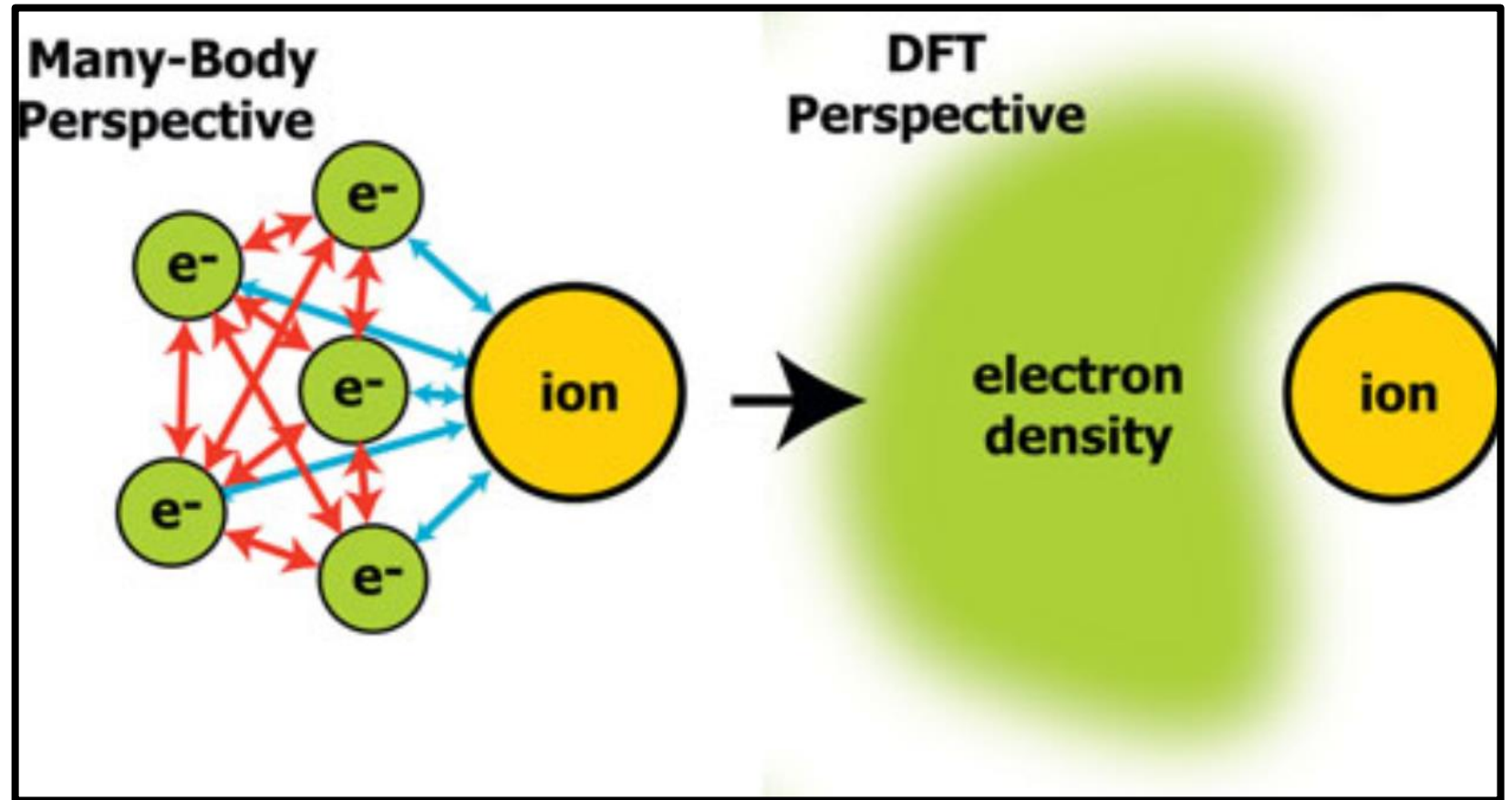
$$\begin{aligned}
 H = & -\sum_{i=1} \frac{\hbar^2}{2m_e} \nabla_i^2 + \frac{1}{2} \sum_{i,j} \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_{I=1} \frac{\hbar^2}{2M_I} \nabla_I^2 \\
 & + \frac{1}{2} \sum_{I,J} \frac{e^2}{4\pi\epsilon_0} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} - \sum_{i,I} \frac{e^2}{4\pi\epsilon_0} \frac{Z_I}{|\mathbf{R}_I - \mathbf{r}_i|} \\
 = & T_e + V_{ee} + T_I + V_{II} + V_{e-I}
 \end{aligned}$$

$$H_e = -\sum_{i=1} \frac{\hbar^2}{2m_e} \nabla_i^2 + \frac{1}{2} \sum_{i,j} \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_{i,I} \frac{e^2}{4\pi\epsilon_0} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I^0|}$$

$$\psi(\{r_i\}) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \varphi_1(r_1), \varphi_2(r_1), \dots, \varphi_N(r_1) \\ \varphi_1(r_2), \varphi_2(r_2), \dots, \varphi_N(r_2) \\ \dots \\ \varphi_1(r_N), \varphi_2(r_N), \dots, \varphi_N(r_N) \end{vmatrix}$$

Figure From [Lixin He](#)

# QM-Density



## The Aufbau Principle is a rule of thumb

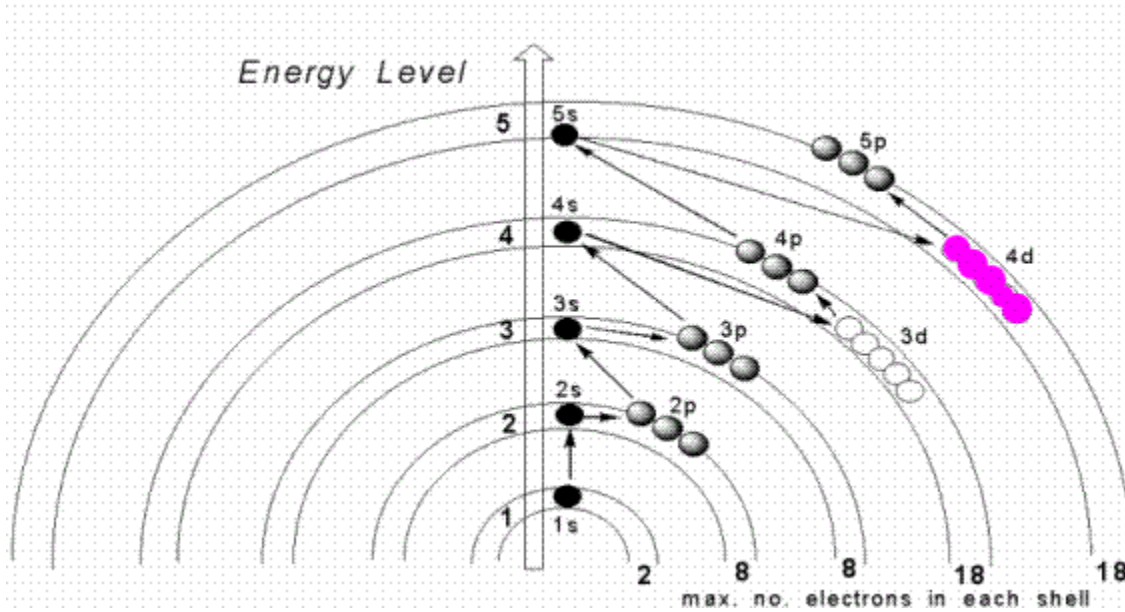


Figure From Internet

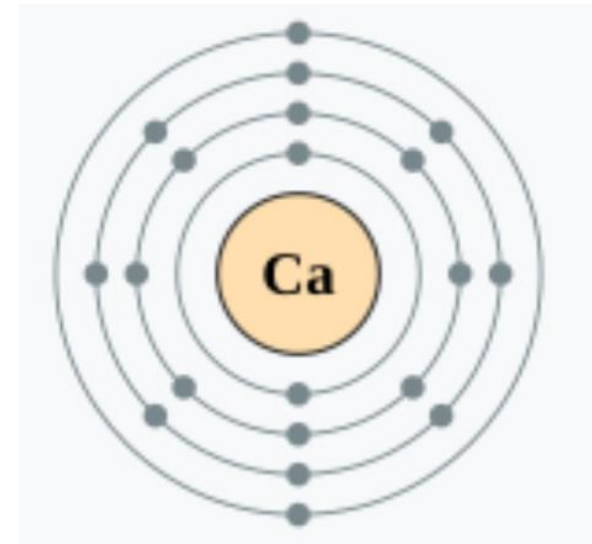


Figure From Internet

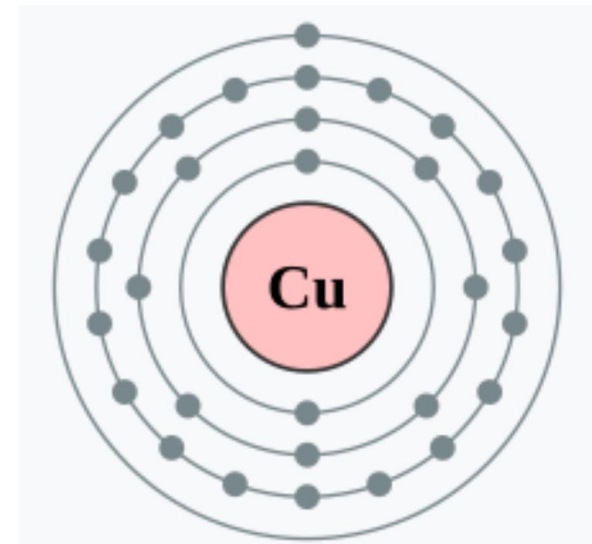
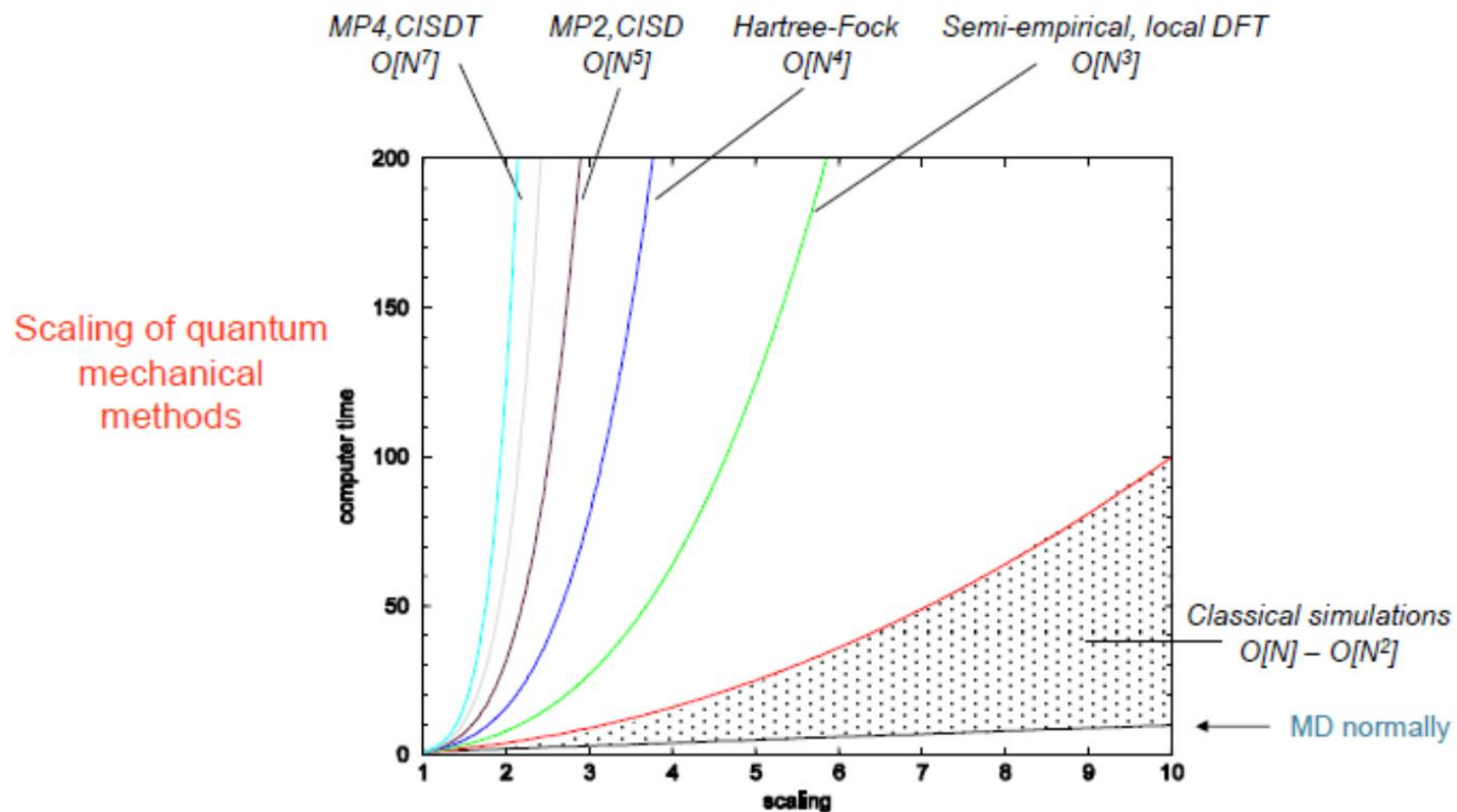


Figure From Internet

## Quantum or Classical?



1. A full quantum description only possible for small systems (100's atoms).
2. For the **thermodynamic** properties of large systems quantum effects generally negligible.

# Molecular Dynamics

- Time-Scale
- Save Compu Time
  - Fixing High Frequency Motions
  - Periodic Boundary Condition
  - Only One Image
  - Cutoff Distance
  - Neighbor Lists
  - Chemical Compound/Coarse-Grained
  - Continuum Model for Water
  - Parallel

Year	Length ( $\mu$ s)	Protein	Platform
2006	2	Rhodopsin	Blue Gene/L <sup>a</sup>
2007	2	Villin HP-35	GROMACS <sup>b</sup>
2008	10	WW domain	NAMD <sup>b</sup>
2009	1,031	BPTI	Anton



# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- Conti Water Model
- Parallel

## Application of constraints using Lagrange multipliers

Distance constraints:

**SHAKE算法**

$$\sigma_k(\vec{r}) \equiv r_{k_1 k_2}^2 - d_{k_1 k_2}^2 = 0 \quad k=1, 2, \dots, N_c$$

**LINCS算法**

非迭代算法

速度更快，易于并行

适用大的模拟体系



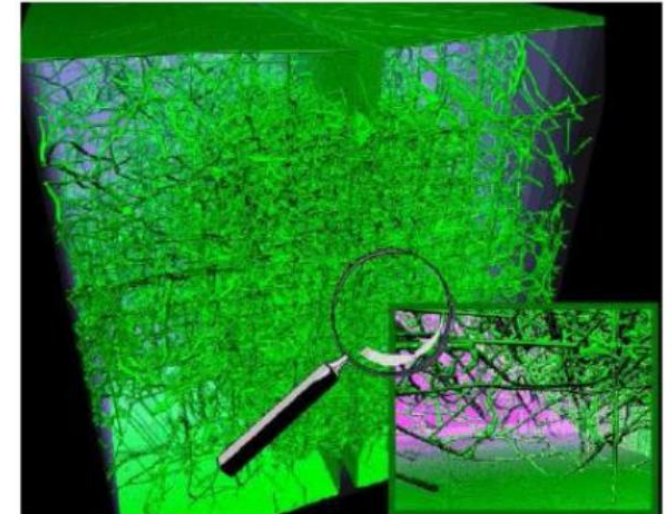
# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- Conti Water Model
- Parallel

## 边界条件

分子动力学模拟方法往往用于研究大块物质在给定密度下的性质，而实际计算模拟不可能在几乎无穷大的系统中进行。

- Generally want to compare to experiments performed on bulk or bulk-like systems
- Simulating a mole of material is impractical
  - Largest simulation to date was  $10^9$  particles
- We know that interfaces perturb structure, dynamics, thermodynamics
- How do we remove (or control) surface effects in simulation?

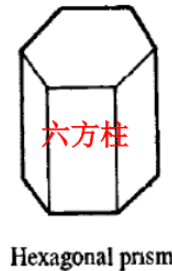
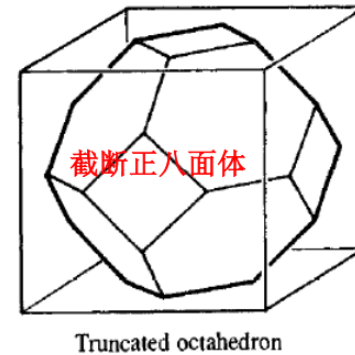
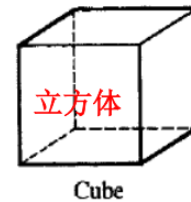


边界条件的重要性：对少量粒子的模拟可以预测体系的宏观性质

# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- Conti Water Model
- Parallel

## 元胞的形状



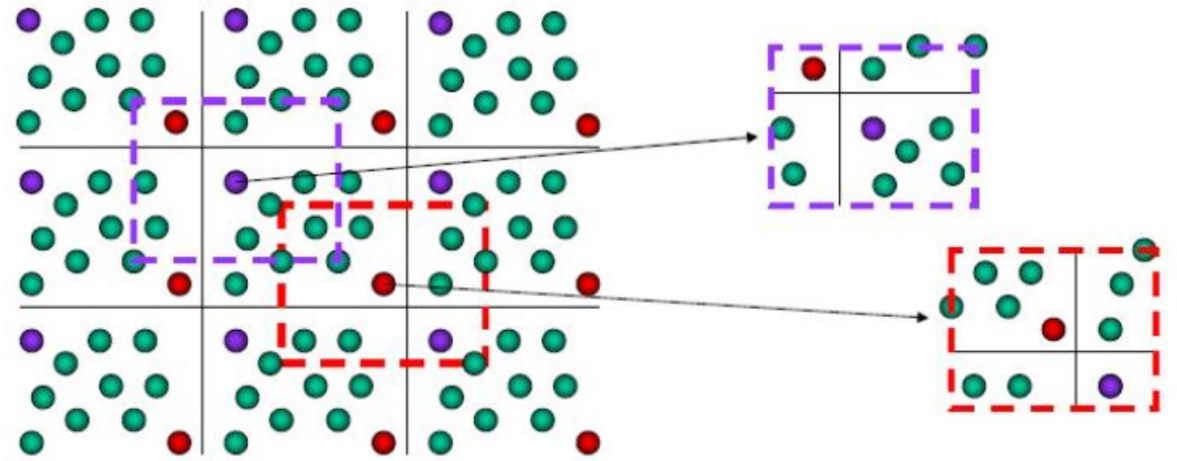
1. 立方体盒子最简单：可视化，程序实现容易。
2. 盒子应该反映分子的形状：如模拟一个球形分子不应用立方体盒子；六方柱适合模拟DNA分子。
3. 截断正八面体和十二面体盒子适用于球形分子，且体积小。

# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- Conti Water Model
- Parallel

## 最小像力约定

- Which particles interact under PBC?
- Minimum image convention (MI) – each particle in the central cell interacts with the nearest periodic copy of other particles



# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- Conti Water Model
- Parallel

## 非键能量和力

MD计算量最大的部分: **the non-bonded energies/forces.**

原则上, 需要计算两两原子之间的相互作用, 算法复杂度为 $N^2$ .

解决方案: 引入截断距离 ( **non-bonded cutoff** ) :  
**the interactions between all pairs of atoms that are further apart than the cutoff value are set to zero.**

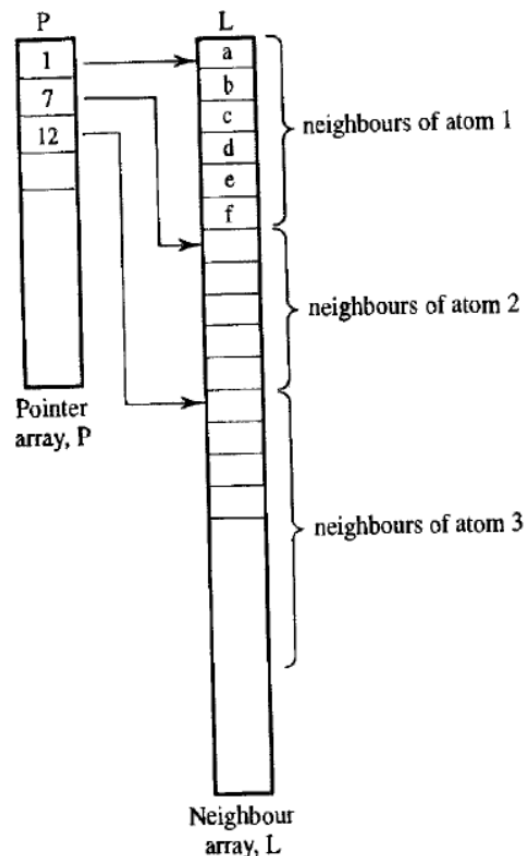
如果相互作用是短程力, 可以在长度 $r_c$ 处截断

$V(r_c)$ 必须足够小, 使截断不会显著影响模拟结果

# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- Conti Water Model
- Parallel

## Verlet邻居列表



邻居列表数组 $L$ 和指针数组 $P$

指针数组：原子 $i$ 的第一个邻居的位置，该原子的最后一个邻居存储在 $L$ 数组的 $P[i+1]-1$

原子 $i$ 的邻居从 $L[P[i]]$ 到 $L[P[i+1]-1]$

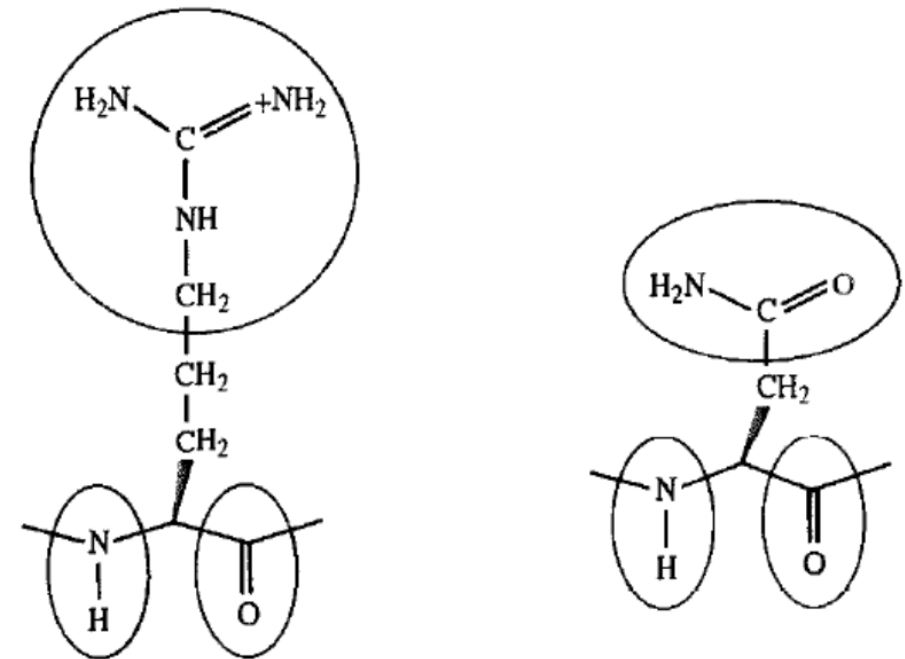
邻居列表定期更新，期间保持不变

# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- Conti Water Model
- Parallel

## Charge Groups

根据“化学直觉”定义‘charge groups’，包含数目较少的原子。

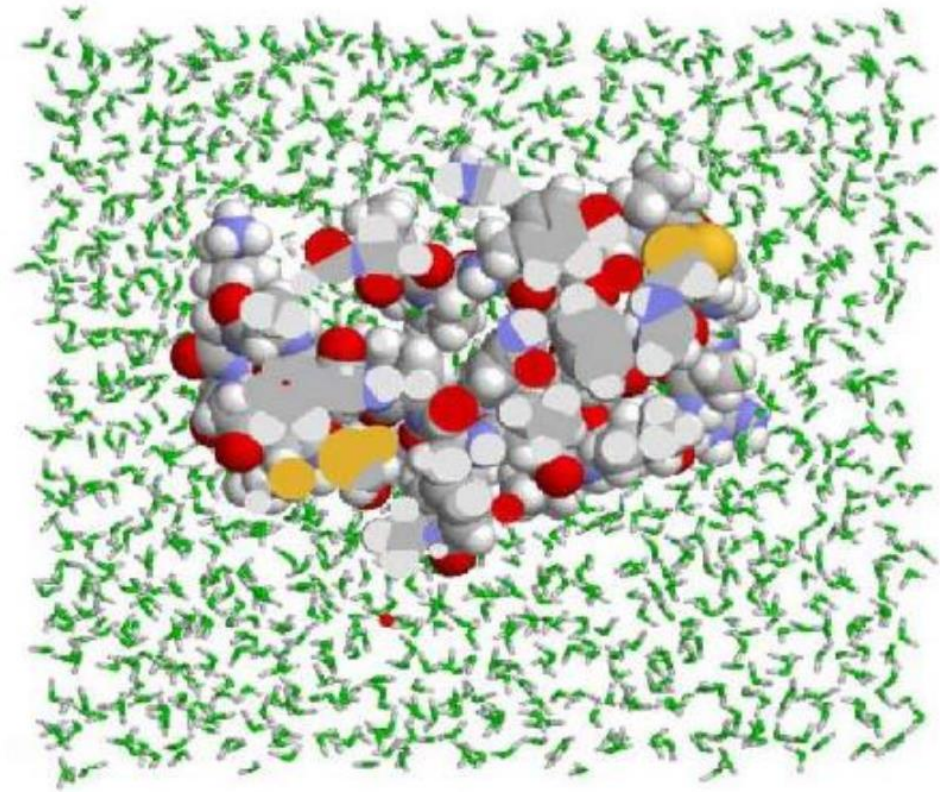




# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- **Conti Water Model**
- Parallel

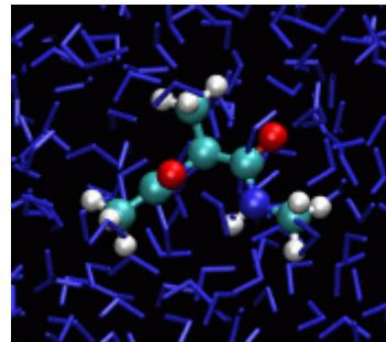
*Protein in water*



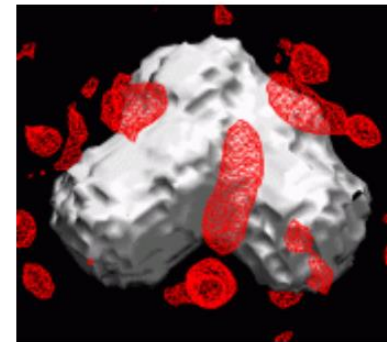
# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- **Conti Water Model**
- Parallel

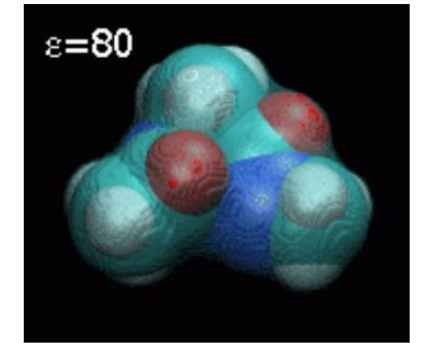
## 溶剂（水）模型



显式水模型  
Explicit Solvent



Average Density



隐式水模型  
Implicit Solvent

<http://feig.bch.msu.edu/main-research-methodology.html>

将水视为连续介质：a continuum with electrostatic, entropic, and viscous properties that match water.



# Save Time | MD

- Fixing High Freq
- PBC
  - Only One Image
- Cutoff Distance
  - Neighbor Lists
- Coarse-Grained
- Conti Water Model
- Parallel

## MD软件

- GROMACS
- Amber
- NAMD
- Charmm
- etc

### MD并行算法

粒子分解 (Particle Decomposition)

PD也称为力分解 (force decomposition)

### MD并行算法

域分解 (Domain Decomposition)

DD算法的原理：模拟盒子被分解为一些相同大小的区域小盒子，一般取边长等于或者大于截断半径；

