

第8章 分子动力学(Molecular Dynamics, MD)

8.1 引言

8.2 基本思想

8.3 主要技术概要

8.4 原子势函数和分子力场的构造

8.5 系综

8.6 模拟的实施

8.7 计算结果的统计处理

8.1 引言

- 分子动力学(Molecular Dynamics, MD)是一种按研究体系内部的**内禀动力学规律**来计算并确定**位形**的变化,跟踪系统中每个粒子的个体运动;然后根据**统计物理规律**,给出**微观量与宏观可观测量的关系**来研究材料性能的确定性模拟方法。
- ✓ 在给定的外界条件下,建立体系中粒子的运动方程
 - ✓ 直接数值求解得到每个时刻各个粒子的坐标与动量(即相空间运动轨迹)
 - ✓ 利用统计力学方法得到系统的静态和动态特性(宏观性质)

8.1 引言

➤MD的发展历史:

- ✓1957年和1959年, Alder和Wainwright提出经典MD方法, 并应用于理想“硬球”液体模型研究Alder相变(刚性球系统液晶→结晶相)。
- ✓1963年, Rahman采用连续势模型研究了液体。
- ✓1967年, Verlet算法的提出
- ✓20世纪70年代, Less和Edwards发展刚性体系的动力学方法研究了非平衡状态(速度梯度); 随后Ginan推广到具有温度梯度的非平衡系统。
- ✓1980年, Anderson恒压状态的MD研究和Hoover的非平衡MD。
- ✓20世纪80年代, MD方法可对分子内部的一部分自由度施加约束条件并应用到生物大分子的解析与设计。
- ✓1981年, Parrinello和Rahman创立恒压的MD模型, 使得原胞的形状可以改变。
- ✓1984年, Nöse创立恒温MD方法。
- ✓1985年, Car、Parrinello将DFT结合进MD方法发展了第一性原理MD方法(Car-Parrinello方法)。

8.2 基本思想

8.2.1 经典力学定律

- ✓ 分子动力学模拟是用来计算经典多体体系的平衡和传递性质的一种确定性方法。
- ✓ 经典是指体系组成的粒子的运动遵从经典力学定律。

$$F_i(t) = m_i a_i(t)$$

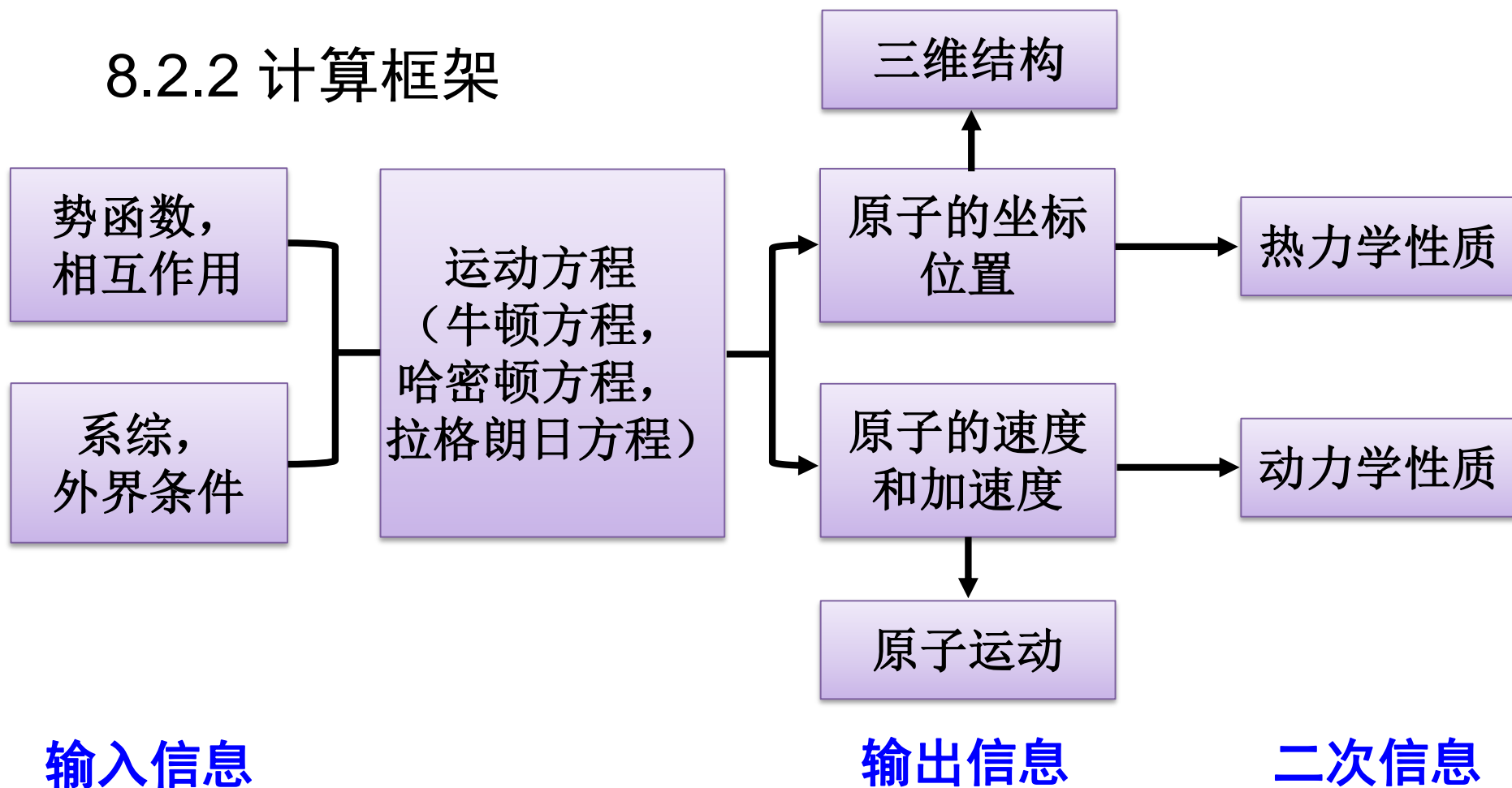
$$m_i \frac{\partial v_i}{\partial t} = F_i(t) = -\frac{\partial U}{\partial r_i} + \dots$$

$$\dot{r}_i(t) = v_i(t)$$

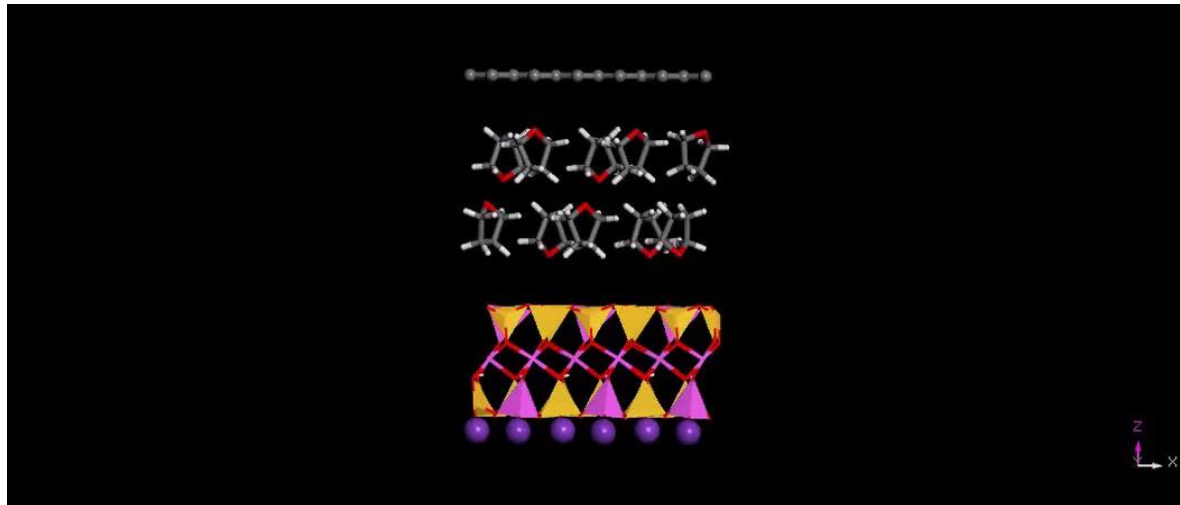
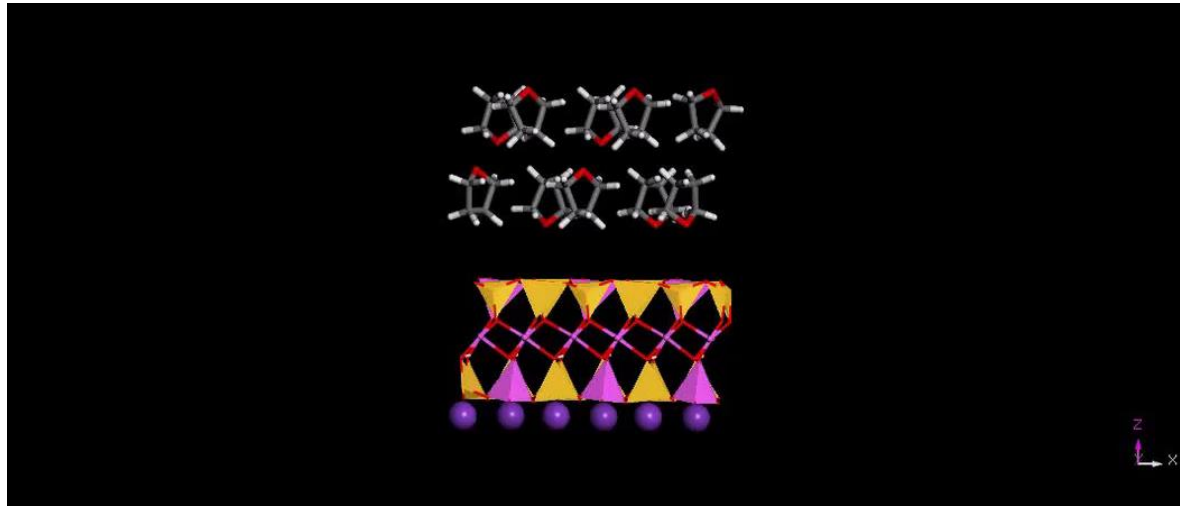
- ✓ 经典运动方程是确定性方程，通过有限差分法进行数值求解。
- ✓ 一旦原子的初始坐标和速度给出，以后任意时刻的坐标和速度都可以确定。

8.2 基本思想

8.2.2 计算框架



Phase Behavior of THF Adlayers without or with Graphene Coating

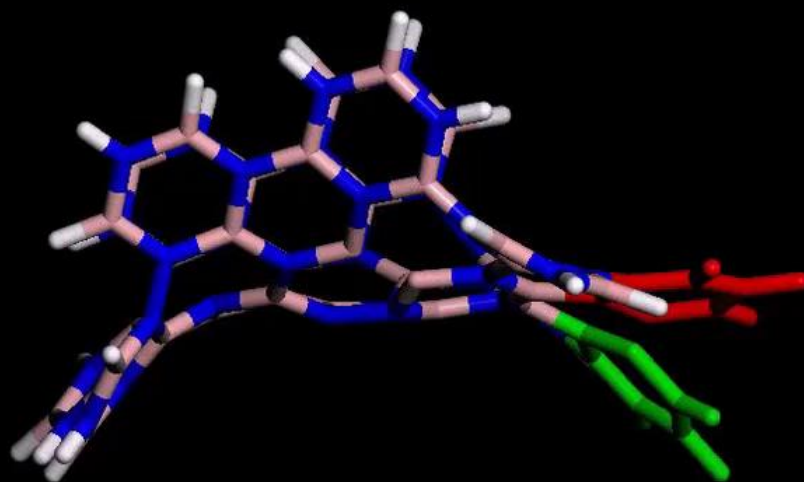


[S. Chen](#), H. Li, P. Cao., and X. C. Zeng, *J. Phys. Chem. C* **2013**, *117*, 21894.

2019/11/11

陈爽 匡亚明学院

AIMD Simulations for Flipping Motion of Boron-Nitride “Pringles”



W. Fa, S. Chen, and X. C. Zeng, *Chem. Commun.*, **2014**, 50, 7444.

2019/11/11

陈爽 匡亚明学院

8.2 基本思想

8.2.3 经典MD中的近似处理

- 经典粒子相互作用，不考虑电子相互作用中的量子效应。
- 力的作用形式，由参数可调的相互作用势函数决定，并实验来验证。
- 模拟体系与实际体系相差较大，一般需要采取周期边界条件来扩展计算体系的大小。
- 时间平均是在有限时间内完成。

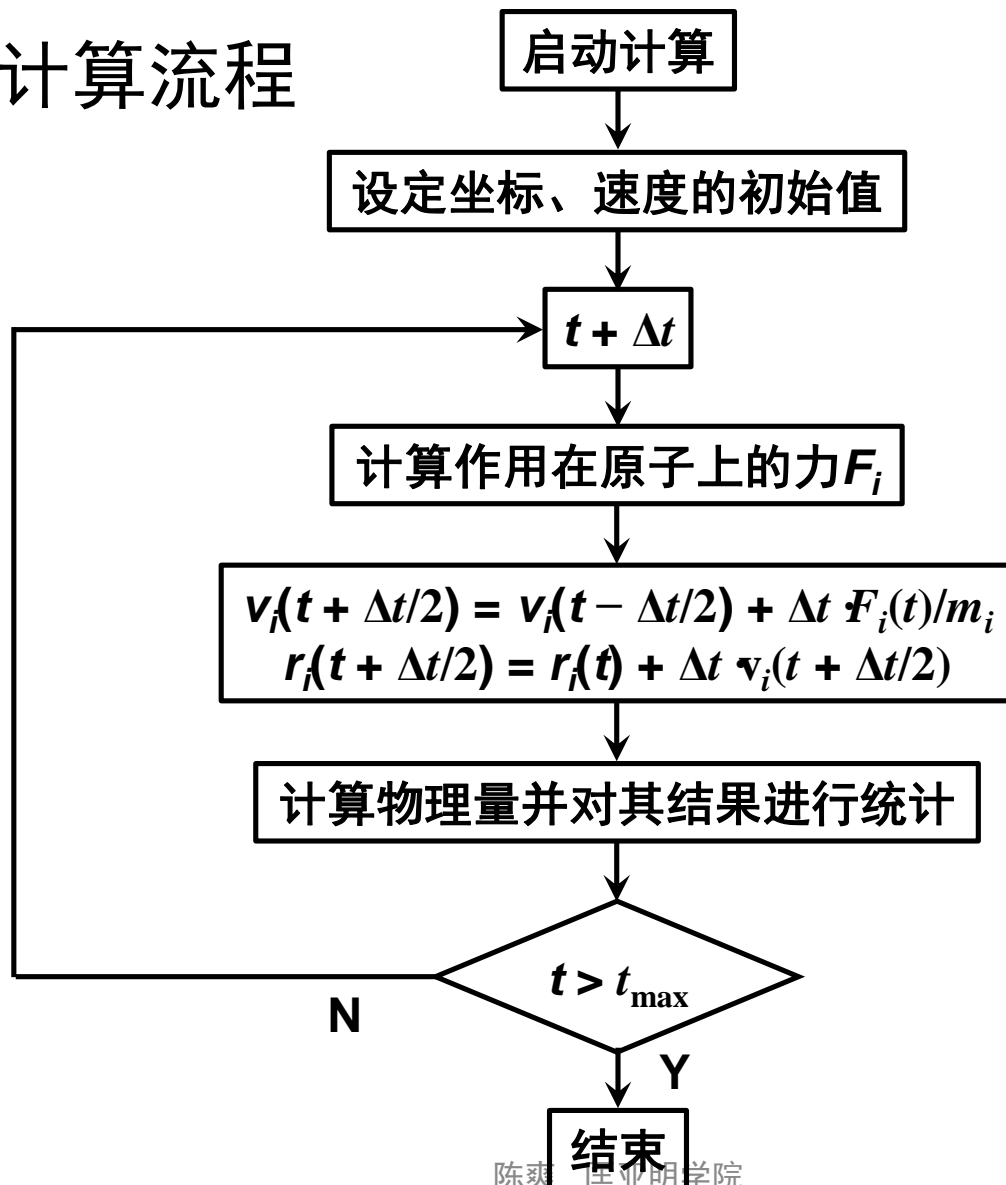
8.2 基本思想

8.2.4 MD的适用范围

- 经典近似对于电荷重新分布的化学反应、键的形成与断裂、解离、极化以及金属离子的化学键都不适用
- 经典MD方法不适用低温
- 经典方法对相对高频率的运动不适用
- 经典方法的适用范围： $\hbar\gamma/k_{\text{B}}T \ll 1$

8.3 主要技术概要

8.3.1 计算流程



8.3 主要技术概要

8.3.1 计算流程

1. 模型的设定，主要是势函数的选取
2. 给定初始条件（初始的位置和速度）

合理的初始条件可以加快系统趋于平衡的时间和步程，获得较好的精度。

3. 趋于平衡计算

适当选择步长，减小误差并让体系达到平衡的时间（弛豫时间）不要过长。

4. 宏观物理量的计算

根基各态历经假说，通过时间平均求解。

8.3 主要技术概要

8.3.2 初始体系的设置

- 给定每个粒子指定的初始坐标和速度
- 最小化弛豫应力
- 初始速度根据伪随机数进行设置，使得体系的总动能与目标温度对应，满足能量均分原理

$$\langle E_k \rangle = \frac{1}{2} \sum_i m_i v_i^2 = N_F k_B T / 2$$

- 单个MD轨迹是无序的，意味着初始条件的一个微小变化就会导致在很短时间内MD模拟偏离正常轨迹。

8.3 主要技术概要

8.3.3 时间步长和势函数

➤时间步长 Δt 的选取：参考原子或分子的特征运动频率

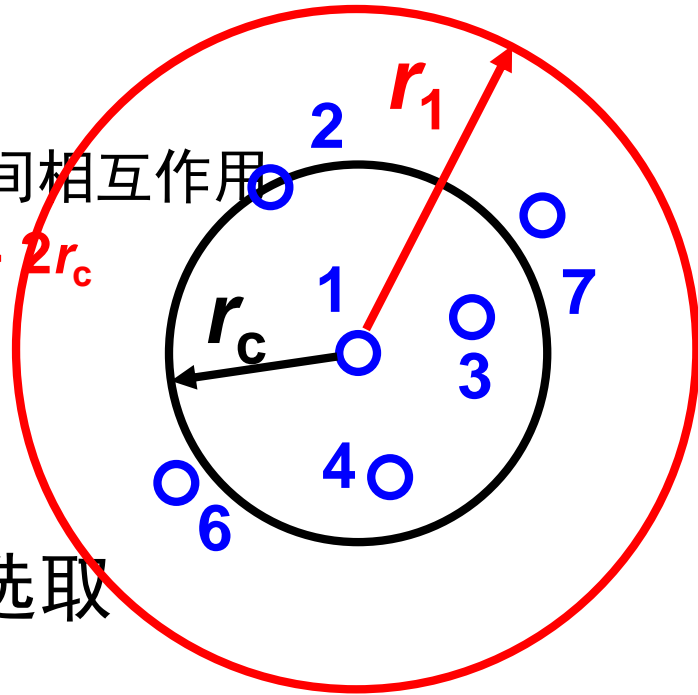
系统	运动型态	积分步长/s
简单原子体系	移动	10^{-14}
刚性分子	移动转动	5×10^{-15}
软性分子，限制键长	移动、转动、扭动	2×10^{-15}
软性分子，无键长限制	移动、转动、扭动、振动	10^{-15} 或 5×10^{-16}

➤势函数是描述原子、分子间相互作用的函数，其选取对模拟结果起决定性作用：文献调研（出处、适用范围）、对材料已知性质验证

8.3 主要技术概要

8.3.4 力的计算方法

- MD模拟所花费的时间90%用来计算原子上的力，正比于原子数目的平方
- 短程力——截断半径法
 - ✓ 只计算截断半径(r_c)以内的离子间相互作用
 - ✓ 保证 L (周期性盒子的长度) $> 2r_c$
 - ✓ 领域 (r_1 领域半径) 列表法
- 长程力——Ewald加和法



8.3.5 运动方程数值求解算法的选取

8.4 原子势函数和分子力场的构造

- 势函数用来描述原子（分子）间相互作用的函数，也称力场。
- 势函数的研究和开发时分子动力学发展的最重要的任务之一。

8.4.1 势函数的简介与分类

- 1903年，Mie指出势函数由两项组成：原子间的排斥和吸引作用。
- 1924年，Lennard-Jones发表负幂指数的L-J势函数。
- 1929年，Morse发表Morse势。
- 1931年，Born和Mayer发表描述离子晶体的Born-Mayer势函数。

8.4 原子势函数和分子力场的构造

8.4.1 势函数的简介与分类

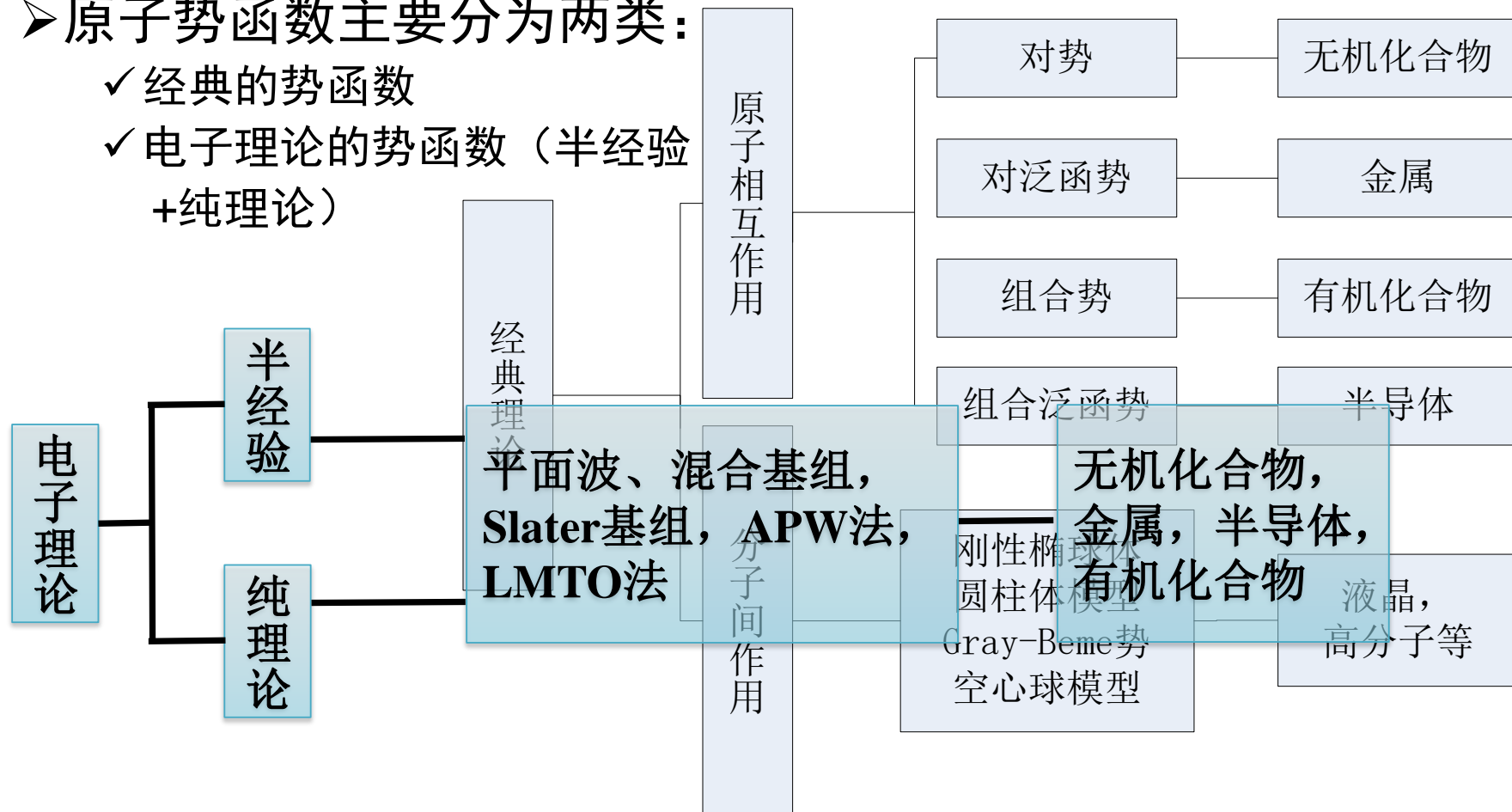
- 20世纪50年代末60年代初，随着计算机的发展MD开始应用于科学研究，其中原子间相互作用的选取是MD模拟的关键。
- 1957年，Alder和Wainwright用硬球势研究凝聚相。
- 1964年，Rahman用连续势研究Ar。
- 1971年，Rahman和Stillinger用连续势研究H₂O。

8.4 原子势函数和分子力场的构造

8.4.1 势函数的简介与分类

➤ 原子势函数主要分为两类：

- ✓ 经典的势函数
- ✓ 电子理论的势函数（半经验+纯理论）



8.4 原子势函数和分子力场的构造

8.4.1 经典理论的原子势函数

势函数	作用因素	典型代表
对势	仅仅决定于原子的坐标	L-J 势、 BMH 势、 GK 势
对泛函势	多体相互作用	EAM 势 有效介质理论
组合势	共价键的相互作用	SW 势
组合泛函势		Abell-Tersoff 势 CheliKowsky-Phillips 势

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

➤ 分子力场(Force Field, FF)根据量子力学的BO近似，将一个分子的能量近似看作构成分子的各个原子的空间坐标的函数，简单地讲就是**分子的能量随分子构型变化而变化**，而描述这种分子能量和分子结构之间关系的**就是分子力场函数**。

- ✓ 分子力场函数来自实验结果的经验公式，对分子能量的模拟比较粗糙
- ✓ 比较精确的量子力学从头算方法，计算量小
- ✓ 在适当范围内，能较为精确的处理大分子复杂体系

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

1. 分子力场的主要讨论对象及其构成

- ✓一套力场函数需有一套联系分子能量和构型的函数
- ✓由**力场参数**加以描述，这些参数大多来自实验或者量子化学计算

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

1. 分子力场的主要讨论对象及其构成

✓力场下势能 U 的表达式

$$U(r^N) = U_{\text{str}} + U_{\text{bend}} + U_{\text{tors}} + U_{\text{cross}} + U_{\text{vdW}} + U_{\text{el}} + U_{\text{pol}}$$

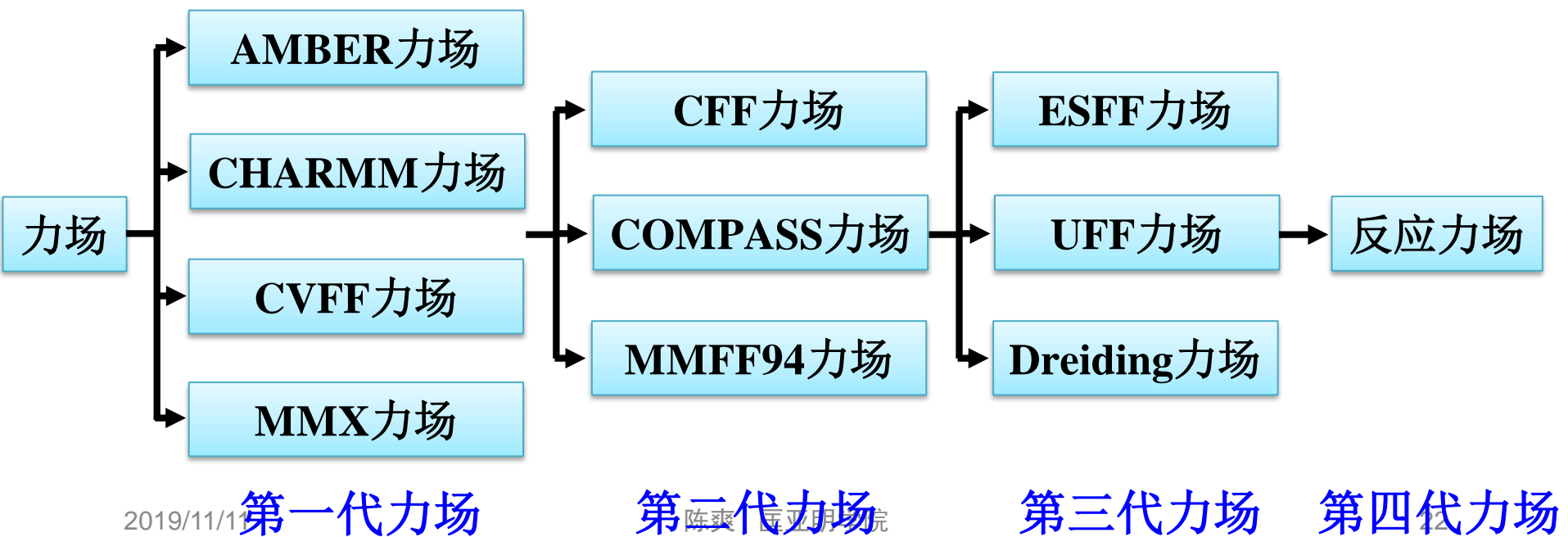
- 键伸缩能：构成分子的各个化学键在键轴方向上的伸缩运动所引起的能量变化；
- 键角弯曲能：键角变化引起的分子能量变化；
- 二面角扭曲能：单键旋转引起分子骨架扭曲所产生的能量变化；
- 交叉能量项：上述作用之间耦合引起的能量变化；
- 非键相互作用：包括静电相互作用、范德华力等与能量有关的非键相互作用。

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

2. 常用力场函数和分类

- ✓不同的分子力场会选取不同的函数形式来描述能量与体系构型之间的关系
- ✓不同的研究小组发展了适用于不同体系的力场函数，在适当范围内能较为精确的处理大分子复杂体系



8.4 原子势函数和分子力场的构造

8.4.2 分子力场

2. 常用力场函数和分类

(1) 传统力场（第一代力场）

- ① AMBER力场：由Kollman课题组开发，适用于较小的蛋白质、核酸、多糖等生物分子，可得到合理的气态分子结构、振动频率。
- ② CHARMM力场：由Karplus课题组开发，适用于小分子体系到溶剂化的大分子体系，可得到结构、振动频率、自由能。
- ③ CVFF力场：Dauber Osguthope发展，适用于有机体系。
- ④ MMX力场：由Allinger等人开发，包括MM2、MM3、MM⁺等，适用于有机化合物、自由基、离子，可得到构型、各种热力学性质、振动光谱等。

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

2. 常用力场函数和分类

(2) 第二代力场：比较第一代，函数形式更复杂，参数更多，计算量更大但更加准确

- ① CFF力场：包括CFF91、PCFF、CFF95等很多力场，可适用于从有机小分子、生物大分子到分子筛等诸多体系。
- ② COMPASS力场：由MSI公司开发，更擅长金属氧化物固体的计算。
- ③ MMFF94力场：由Hagler针对药物设计开发，适用于固态或液态的有机小分子。

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

2. 常用力场函数和分类

(3) 通用力场（第三代力场）：也叫基于规则的力场，它所使用的力场参数是基于分子性质计算所得，用户可通过自主设定一系列分子作为训练集来生成合用的力场参数。

- ① ESFF力场：由MSI公司开发，适用于预测有机、无机分子的结构，但不能计算构型能或准确的振动频率，包含H~Rn元素。
- ② UFF力场：可以计算周期表上所有元素的参数，但计算与分子间作用有关的性质偏差较大。
- ③ Dreiding力场：适用于有机小分子、大分子、主族元素，包含周期表中的所有元素。

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

3. 长程相互作用的计算

- ✓ 分子之间的相互作用可作为长程作用来处理
- ✓ 分为非周期性长程相互作用（中央多极展开法）和周期性长程相互作用（Ewald求和算法等）

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

3.长程相互作用的计算

(1) 中央多极展开法

- 两两原子间考虑静电作用计算量太大
- 将分子视为一个整体，利用分子的电次极计算相互作用（最重要的**考虑分子最低不为零的电极**）
- 各种电极可用适当的电荷排列表示
- 各极作用随 r^{-n} 衰减很快，无需计算长距离的作用
- 中央多极展开法**适用于 r 距离远大于分子尺寸的情况**

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

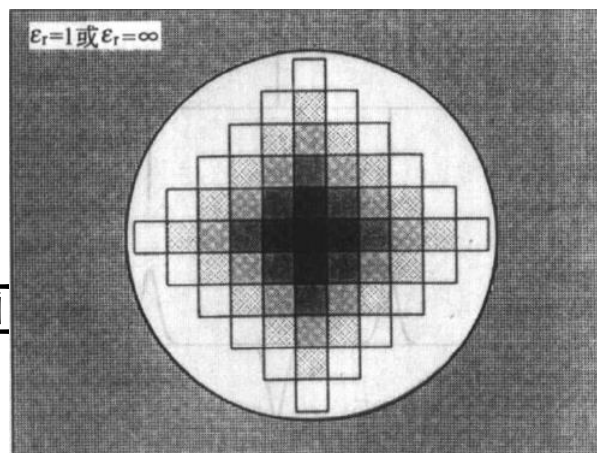
3. 长程相互作用的计算

(2) Ewald求和算法

- 选定计算的盒子，盒子中的粒子除了和盒子内部的粒子相互作用，还和其镜像系统中的粒子发生作用。镜像系统和盒子内部系统完全相同。电荷之间的库仑势为：

$$U = \frac{1}{2} \sum_{|\vec{n}|=0}^{\infty} \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{4\pi\epsilon_0 |\vec{r}_{ij} + \vec{n}|}$$

- 上式收敛太慢且为条件收敛，Ewald发展求和方法将上式转化为两个收敛速度很快的级数。



Ewald求和法的
周期作用区

8.4 原子势函数和分子力场的构造

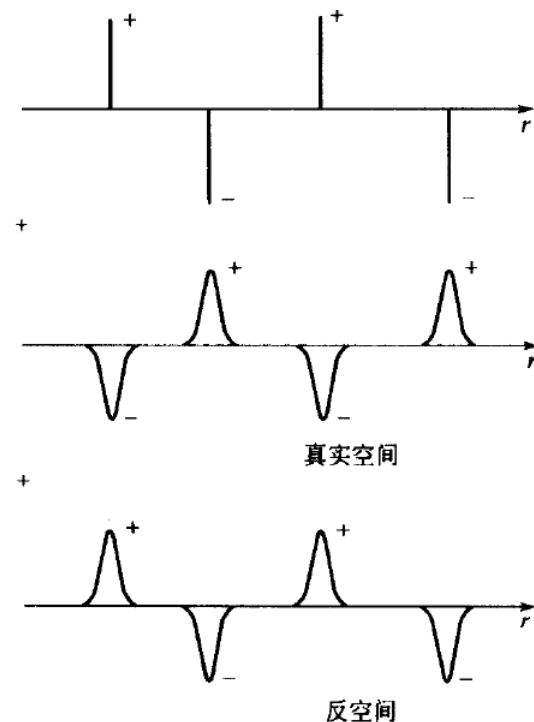
8.4.2 分子力场

3.长程相互作用的计算

(2) Ewald求和算法

- Ewald将每个电荷周围用两个同量反相呈中和的电荷包围
- 电荷分布以Gaussian分布的函数形式表示, 利用此电荷分布式可将点电荷间的加成项转换为电荷以及中性电荷分布间的作用 (实空间+倒空间)
- 这样处理的结果几个镜像就快速收敛
- Ewald方法为各种计算静电作用的方法中准确性最高最为可行的方法, 速度一般为计算无静电作用系统的1/10。

Ewald方法的 电荷分布



8.4 原子势函数和分子力场的构造

8.4.2 分子力场

3.长程相互作用的计算

(3) 格胞（快速）多极子法

- 将所有 $N(N-1)$ 成对作用项简化为 N 项的作用，简化的加成项的形式为：

$$\sum_i \sum_{j < i} \frac{q_i q_j}{|\vec{r}_i - \vec{r}_j|^p}$$

式中 r_i 、 r_j 为表征原子坐标， q_i 、 q_j 为多极距。

- 该体系的静电作用和L-J作用项具有相同的形式。
- 将需要计算的体系分为多个体积相同的格胞，由格胞内部的所有原子计算每一个格胞的多极（电荷、偶极、四极……）。

8.4 原子势函数和分子力场的构造

8.4.2 分子力场

3.长程相互作用的计算

(3) 格胞（快速）多极子法

- 格胞和格胞之间的相互作用是利用中央多极展开的方法进行计算。
- 中央多极展开的方法的条件是格胞之间的距离必须是大于一定距离的，因为大于一个格胞以上的以多极展开的方式计算相互作用的力，而格胞内部的原子对是以成对原子的作用力来计算的。

8.5 系综

- 系综(Ensemble)是指在一定的宏观条件下（约束条件），大量性质和结构完全相同的、处于各种运动状态的、各自独立的系统集合，或称为统计系综。
- 确定模拟体系的大小、规格和相关的模拟条件（**视实际的体系需要而定**）
- MD中常用的系综分类
 1. 微正则系综(Micro-Canonical Ensemble)
 2. 正则系综(Canonical Ensemble)
 3. 等温等压系综 (Constant-Pressure, Constant-Temperature Ensemble)

8.5 系综

8.5.1 微正则系综(NVE)

- 平衡体系为孤立系统：与外界既无能量交换，也无离子交换
- 特性函数熵 $S(N, V, E)$
- 系统的温度(T)和压强(P)会在某一平均值附近起伏变化（涨落）
- 通常用时间平均代替系综平均

8.5 系综

8.5.2 正则系综(NVT)

- 平衡体系为封闭系统，总动量为0，与大热源热接触平衡的恒温系统，二者发生能量交换（处于热平衡状态）
- 特性函数Helmholtz自由能 $A(N, V, T)$
- 系统的总能量(E)和压强(P)会在某一平均值附近起伏变化（涨落）

8.5 系综

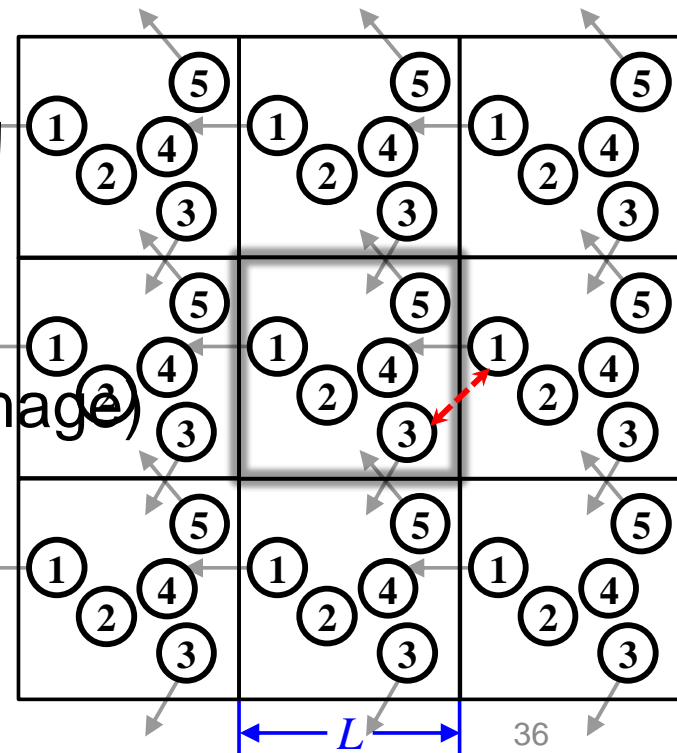
8.5.3 等温等压系综(NPT)

- 体系可采用可移动系统壁的恒温热浴模型表示，通过标度系统的体积来实现压力的调节
- 特性函数Gibbs自由能 $G(N, P, T)$
- 系统的总能量(E)和体积(V)会在某一平均值附近起伏变化（涨落）

8.6 模拟的实施

8.6.1 周期性边界条件

- 实际MD模拟受到有限观测时间和有限系统尺寸的限制
- 实际可操作分子数一般在 $10^2 \sim 10^4$ 数量级
- 对研究体系施加周期性边界条件
- 研究体系的大小取决于实验所测定的体系密度
- 计算系统中分子间的相互作用力时，采取最近镜像方法(Nearest Mirror Image)
- 为了不重复计算粒子的受力，考虑截断半径($r_c < L/2$)，一般 $10 \sim 15 \text{ \AA}$



8.6 模拟的实施

8.6.2 时间积分的处理——积分步长(integration time step, Δt)的选取:

- ✓节省MD模拟计算时间而不失的计算精度
- ✓通常的原则：积分步长应小于系统中最快运动周期的1/10
- ✓各种模拟系统建议采取的积分步长：

系统	运动形态	积分步长(s)
简单分子体系	平动	10^{-14}
刚性分子	平动、转动	5×10^{-15}
软性分子，限制键长	平动、转动、扭动	2×10^{-15}
软性分子，键长不限制	平动、转动、扭动、振动	10^{-15} 或 5×10^{-16}

8.6 模拟的实施

8.6.3 时间积分的处理——数值积分方法的选取

1. Verlet方法（常用于NVE系综的MD微分方程的数值积分求解）

将 $t + \Delta t$ 和 $t - \Delta t$ 的位置坐标分别用时刻 t 的位置坐标进行Taylor展开，则有

$$r(t + \Delta t) = r(t) + \Delta t \cdot v(t) + (\Delta t)^2 \cdot \frac{a(t)}{2} + \dots$$

$$r(t - \Delta t) = r(t) - \Delta t \cdot v(t) + (\Delta t)^2 \cdot \frac{a(t)}{2} - \dots$$

为消去速度项，将两式相加有（忽略高次项）：

$$r(t + \Delta t) \approx 2r(t) - r(t - \Delta t) + (\Delta t)^2 \cdot a(t)$$

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

8.6 模拟的实施

8.6.3 时间积分的处理——数值积分方法的选取

1. Verlet方法

位置坐标两式相减得到速度表达式：

$$v(t) \approx \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t}$$

2. 蛙跃法(Leap-Frog Method)

首先利用速度将二阶常微分方程转换为2个一阶常微分方程：

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

速度的微分由时刻 $t + \Delta t/2$ 和 $t - \Delta t/2$ 的差分给出：

$$\frac{v\left(t + \frac{\Delta t}{2}\right) - v\left(t - \frac{\Delta t}{2}\right)}{\Delta t} = \frac{F(t)}{m}$$

8.6 模拟的实施

8.6.3 时间积分的处理——数值积分方法的选取

2. 蛙跃法(Leap-frog)

则时刻 $t + \Delta t/2$ 的速度为：

$$v\left(t + \frac{\Delta t}{2}\right) = v\left(t - \frac{\Delta t}{2}\right) + \frac{\Delta t}{m} \cdot F(t)$$

进一步利用时刻 $t + \Delta t$ 和 t 的差分给出位置矢量的微分：

$$\frac{r(t + \Delta t) - r(t)}{\Delta t} = v\left(t + \frac{\Delta t}{2}\right)$$

最后得到时刻 $t + \Delta t$ 的位置矢量为：

$$r(t + \Delta t) = r(t) + \Delta t \cdot v\left(t + \frac{\Delta t}{2}\right)$$

8.6 模拟的实施

8.6.3 时间积分的处理——数值积分方法的选取

2. 蛙跃法(Leap-frog)

- ✓如果使用Verlet法或蛙跃法，则可求解关于NVE系综的线性常微分方程。
- ✓Hockey提出的蛙跃法是Verlet算法的变化，这种方法涉及半时间间隔的速度。
- ✓这种算法与Verlet算法相比有两个优点：
 - ① 包含显式速度项
 - ② 收敛速度快，计算量小
- ✓该算法的缺点：位置和速度不同步

8.6 模拟的实施

8.6.3 时间积分的处理——数值积分方法的选取

3. 校正预测法(Predictor-Corrector Method)

因为经典运动粒子的轨迹是连续的，所以可将 $(t + \Delta t)$ 时刻的位置、速度、加速度等在时刻 t 进行Taylor展开：

$$r^P(t + \Delta t) = r(t) + \Delta t v(t) + (\Delta t)^2 \cdot \frac{a(t)}{2} + (\Delta t)^3 \cdot \frac{b(t)}{6} + \dots$$

$$v^P(t + \Delta t) = v(t) + \Delta t \cdot a(t) + \frac{(\Delta t)^2 \cdot b(t)}{2} + \dots$$

$$a^P(t + \Delta t) = a(t) + \Delta t \cdot b(t) + \dots$$

$$b^P(t + \Delta t) = b(t) + \dots$$

若使用上式给出的新的位置矢量 $r^P(t + \Delta t)$ ，则可计算 $(t + \Delta t)$ 时刻的作用力 $F(t + \Delta t)$ 和加速度 $a^C(t + \Delta t)$

8.6 模拟的实施

8.6.3 时间积分的处理——数值积分方法的选取

3. 校正预测法

根据加速度 a^C 可评价预测的加速度 $a^P(t + \Delta t)$ 的误差为：

$$\Delta a(t + \Delta t) = a^C(t + \Delta t) - a^P(t + \Delta t)$$

以此为依据，可得各物理量的校正项为：

$$r^C(t + \Delta t) = r^P(t + \Delta t) + C_0 \Delta a(t + \Delta t)$$

$$v^C(t + \Delta t) = v^P(t + \Delta t) + C_1 \Delta a(t + \Delta t)$$

$$a^C(t + \Delta t) = a^P(t + \Delta t) + C_2 \Delta a(t + \Delta t)$$

$$b^C(t + \Delta t) = b^P(t + \Delta t) + C_3 \Delta a(t + \Delta t)$$

C_0 、 C_1 、 C_2 、 C_3 为常数。

以上所述为Gear的一次预测校正法

8.6 模拟的实施

8.6.3 时间积分的处理——数值积分方法的选取

3. 校正预测法

此算法分三步完成：

- ① 根据Taylor展开，预测新的位置、速度和加速度；
- ② 根据新的位置矢量计算的力和加速度；
- ③ 这个加速度再与Taylor级数展开式中的加速度进行比较，两者之差在校正步里用来校正位置、速度。

该方法的优点：积分步长减小时，Gear的高次预测校正法误差减小。

该方法的缺点：占用计算机的内存大

8.6 模拟的实施

8.6.3 时间积分的处理——数值积分方法的选取

4. Velocity-Verlet算法

此算法可**同时**给出位置、速度和加速度，并且不牺牲精度，给出了显式速度项，计算量适中，目前应用比较广泛。

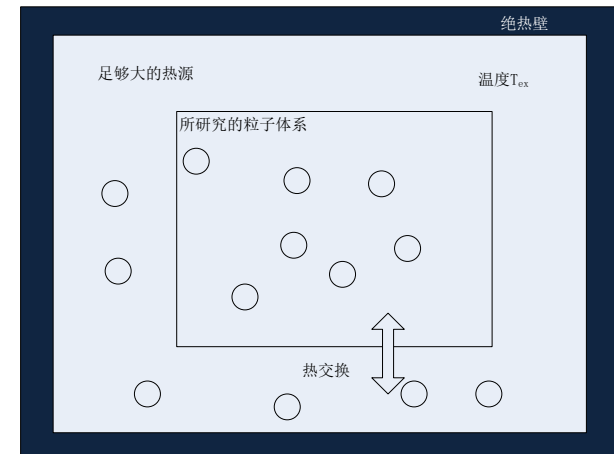
$$r(t + \Delta t) = r(t) + \Delta t \cdot v(t) + (\Delta t)^2 \cdot \frac{a(t)}{2}$$

$$v(t + \Delta t) = v(t) + \frac{1}{2} \Delta t [a(t) + a(t + \Delta t)]$$

$$v(t + \frac{1}{2} \Delta t) = v(t) + \frac{\Delta t}{2} \cdot a(t)$$

$$v(t + \Delta t) = v(t + \frac{1}{2} \Delta t) + (\Delta t) \cdot \frac{a(t + \Delta t)}{2}$$

8.6 模拟的实施



8.6.4 控温方法：对 NVT 系综而言，有4种控温方法，无论何种控温方式，都是将体系与一个巨大的“热浴”相接触，通过热交换恒定体系温度。

1. Velocity Scaling方法（最简单的控温方法）
2. Berendsen热浴
3. Anderson热浴——随机碰撞(Stochastic Collision)
4. Nosé-Hoover热浴——扩展体系(Extended System)

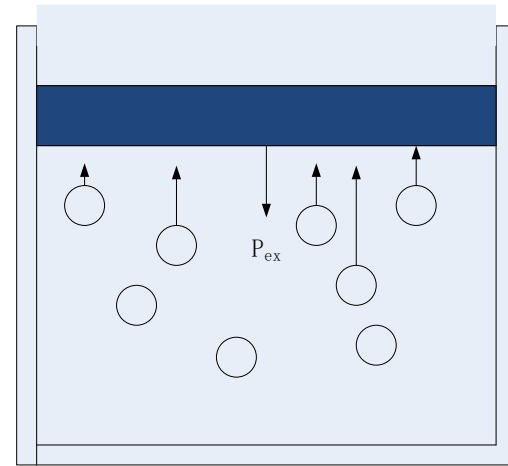
8.6 模拟的实施

8.6.4 控温方法：

四种方法的比较：

- ✓ Velocity Scaling方法和Berendsen热浴2种控温方法相对简单，不能得到严格的正则平均值。
- ✓ Velocity Scaling方法人为拉大了体系中组分的温度差，会导致“热溶剂”、“冷溶质”现象，甚至在体系温度达到目标温度时，溶质的温度也要低于溶剂的温度。
- ✓ Anderson和Nosé-Hoover热浴则可以产生严格的正则系综。
- ✓ Anderson热浴较适用于已达热平衡的体系，Nosé-Hoover热浴较Anderson热浴更稳定，可用于调整未平衡体系的温度

8.6 模拟的实施



8.6.5 控压方法

- 研究体系的体积与维里压力耦合。假想研究体系与一特定重量的活塞作用共同构成系统，活塞的重量视作外压，通过活塞调控研究体系的体积，对体系的压力进行控制，使其达到某一预期值。
- 在特定系综的模拟中，需要采取特殊的方法控制系统压力
 1. Berendsen恒压法
 2. Anderson恒压法
 3. **Parrinello-Rahman恒压法**

8.6 模拟的实施

8.6.5 控压方法

三种方法的比较：

- ✓ Parrinello-Rahman恒压法现被科研工作者广泛应用。它集成了上述Berendsen恒压法以及Anderson恒压法两种方法的长处。
- ✓ Anderson恒压法只有在元胞各方向受到大小相同的压力时才能采用，而Berendsen恒压法虽然能实现同时调节三个方向，可是一旦元胞受到剪切力作用，Berendsen 恒压法将不能同时调节三个方向。
- ✓ Parrinello-Rahman恒压法既能在元胞的体积和形状同时发生改变的条件下完成模拟工作，又能在元胞受到剪切力的情况下工作。

8.7 计算结果的统计处理

➤MD的计算结果分2个层次：

1. 计算模拟直接的输出：包含了平衡态的体系位形信息和体系运动的基本动力学信息；
2. 基于统计热力学得到体系的宏观物理性质（热力学、光学性质等）

8.7 计算结果的统计处理

8.7.1 宏观物理参量的统计平均

MD模拟直接的输出结果是原子的位置坐标和速度。物理参量可以根据位置坐标和速度通过统计处理求出。

$$\langle A \rangle_{\text{ens}} = \frac{1}{N!} \iint dp dr A(p, r) p(p, r)$$

式中 $p(p, r)$ 是分布函数（概率密度）。概率密度因所考察的系综不同而不同。

在MD方法中，我们使用所谓的时间平均等于系综平均的各态历经假设，即

$$\langle A \rangle_{\text{ens}} = \langle A \rangle_{\text{time}}$$

此时有

$$\langle A \rangle_{\text{time}} = \lim_{t_{\text{obs}}} \frac{1}{t_{\text{obs}}} \int A(p, r) dt$$

8.7 计算结果的统计处理

8.7.2 3种形式的统计平均值处理方法

1. 轨迹平均

独立系统的平衡相空间轨迹上取 M_∞ 个不同的取样点，这些取样点是均匀分布在轨迹上的，相隔 Δt 时间的相邻点。对于一个与时间有关的性质物理量 $a(t)$ ，其时间平均值可取为：

$$\langle A \rangle = \frac{1}{M_\infty} \sum_k^{M_\infty} a(k \cdot \Delta t)$$

对于在任意时刻的 $a(t)$ 相对于平均值 $\langle A \rangle$ 的变异数为：

$$V[\langle A \rangle - a(t)] \equiv S^2 = \frac{1}{M_\infty - 1} \cdot \sum_k^{M_\infty} (\langle A \rangle - a(k \cdot \Delta t))^2$$

8.7 计算结果的统计处理

8.7.2 3种形式的统计平均值处理方法

2. 取样平均

当我们不希望对轨迹上所有点(M_{∞})进行取样时, 可在轨迹上取一段, 对此段上 $M(M \ll M_{\infty})$ 个点进行计算。该小段轨迹内 M 个点组成一个集合, 成为样本。此时样本的平均值为:

$$\langle a \rangle = \frac{1}{M} \sum_k^M a(k \cdot \Delta t)$$

在任意时刻的 $a(t)$, 针对此平均值的变异数为:

$$V[\langle a \rangle - a(t)] \equiv \sigma^2 = \frac{1}{M-1} \cdot \sum_k^M (\langle a \rangle - a(k \cdot \Delta t))^2$$

8.7 计算结果的统计处理

8.7.2 3种形式的统计平均值处理方法

3. 取样平均的期望值

自整个轨迹上取 M_S 个片段轨迹，每段轨迹都包含 M 个点，任意其中一个取样段（例如第 k 个），都有它自己的平均值 $\langle a \rangle_k$ ，所以共有 M_S 个平均值。

取这 M_S 个平均值再进行平均，则成为平均值的期望值

$$E[\langle a \rangle] = \frac{1}{M_S} \sum_k^{M_S} \langle a \rangle_k$$

则这些取样段平均值对 $E[\langle a \rangle]$ 进行性质函数分布，称为取样分布(Sampling Distribution)。

对任意一个取样段的平均值相对平均值的期望值的变异数为：

$$V[E[\langle a \rangle] - \langle a \rangle] \equiv \frac{1}{M_S - 1} \cdot \sum_k^{M_S} (E[\langle a \rangle] - \langle a \rangle_k)^2$$

References

1. 单斌，陈征征，陈蓉编，材料学的纳米尺度计算模拟：从基本原理到算法实现，华中科技大学出版社，**2015**。
2. 江建军，缪灵，梁培，马新国编，计算材料学：设计实践方法，高等教育出版社，**2010**。
3. 坚增运，刘翠霞，吕志刚编，计算材料学，化学工业出版社，**2012**。
4. 陈正隆，徐为人，汤立达编，分子模拟的理论与实践，化学工业出版社，**2007**。
5. A. R. Leach, *Molecular Modelling: Principles and Applications (2nd Edition)*, **2001**.
6. D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications (2nd Edition)*, **2002**.