

$$\langle \Delta r^2(t) \rangle = \frac{1}{N} \sum_{i=1}^N [\mathbf{r}_i(t) - \mathbf{r}_i(0)]^2 \quad (4.1.3.6-3)$$

对于固体，它等于常数（图4.1.3.6-2）。而在液态时，它是与时间成正比的，与（1.4.1.1-7）式相同，这是因为液态中的原子可以流动扩散。

第三个因子是对关联分布函数 $g(r)$ （2.3.4.2-2）式，晶态时它有特定的结构。

4.1.4 运动方程积分算法

4.1.4.1 Verlet 型算法

计算了粒子间的相互作用力后，就可以对Newton运动方程积分。最简单通常也是最好的方法是用Verlet算法，将（3.1.2.8-3）式写为

$$x(t+h) = 2x(t) - x(t-h) + h^2 F(t)/m + O(h^4), \quad (4.1.4.1-1)$$

在对位置的大量积分步之后，其坐标的累计误差将达到 $O(h^2)$ 。该算法在时间反演（ $h \rightarrow -h$ ）下是不变的，意味着将运动方程积分一段时间后再用当前的两个点起始做时间反演的的话，相空间中的轨迹将完全沿原路返回。

注意Verlet算法中并不使用速度来计算新的位置。但我们需要知道速度以计算动能 K ，以此来监测总能 $E = K + V$ 是否守恒，它是分子动力学中验证计算结果是否正确的最重要判据。根据空间轨迹的知识可以求得速度，它为中心差分

$$v(t) = \frac{x(t+h) - x(t-h)}{2h} + O(h^2), \quad (4.1.4.1-2)$$

即用该算法得到的速度是一阶的。Verlet算法中的速度和位置是不对称的，最早的方法中速度远没位置重要，这是与相空间中轨迹的速度坐标和位置坐标等价的概念不符的，更为现代的方法已经改正了这点。

利用半步长的速度值，

$$v(t+h/2) = \frac{x(t+h) - x(t)}{h} + O(h^2), \quad (4.1.4.1-3)$$

可以将Verlet算法改写成对精度误差敏感性较小的蛙跳法，

$$\begin{aligned} v(t+h/2) &= v(t-h/2) + hF(t)/m + O(h^2) \\ x(t+h) &= x(t) + hv(t+h/2) + O(h^4) \end{aligned} \quad (4.1.4.1-4)$$

该方法是从Verlet算法导出的，产生相同的空间轨迹。需注意的是，速度和位置并未在同一时间定义，结果是动能和势能也并未同时定义。因此蛙跳法中不能直接计算总能量。

还可以将Verlet算法重写成一种同时计算位置和速度的形式 - 速度Verlet法。坐标用Taylor展开，

$$x(t+h) = x(t) + v(t)h + h^2 F(t)/2m, \quad (4.1.4.1-5)$$

速度则是用

$$v(t+h) = v(t) + h[F(t+h) + F(t)]/2m. \quad (4.1.4.1-6)$$

只有在计算完新的位置和新的力之后，才可以计算新的速度。该算法实际上等价于原来的 Verlet 算法，为了证明这点，将式 (4.1.4.1-5) 改写成

$$x(t+2h) = x(t+h) + v(t+h)h + h^2 F(t+h)/2m; \quad (4.1.4.1-7)$$

$$x(t) = x(t+h) - v(t)h - h^2 F(t)/2m, \quad (4.1.4.1-8)$$

两式相加得

$$x(t+2h) + x(t) = 2x(t+h) + [v(t+h) - v(t)]h + h^2 [F(t+h) - F(t)]/2m, \quad (4.1.4.1-9)$$

代入 (4.1.4.1-6) 式后，得

$$x(t+2h) + x(t) = 2x(t+h) + h^2 F(t+h)/m, \quad (4.1.4.1-10)$$

它正是在 $t+h$ 时刻的 (4.1.4.1-1) 式。

另外有两种算法产生与 Verlet 算法相同的轨迹，但速度的估计更好。第一个是 Beeman 算法，它看上去与 Verlet 算法相当不同：

$$x(t+h) = x(t) + v(t)h + h^2 [4F(t) - F(t-h)]/6m; \quad (4.1.4.1-11)$$

$$v(t+h) = v(t) + h[2F(t+h) + 5F(t) - F(t-h)]/6m, \quad (4.1.4.1-12)$$

但将式 (4.1.4.1-12) 代入到式 (4.1.4.1-11) 后就可以验证，位置其实满足 Verlet 算法：

$$\begin{aligned} x(t+2h) &= x(t+h) + v(t+h)h + h^2 [4F(t+h) - F(t)]/6m \\ &= x(t+h) + v(t)h + h^2 F(t+h)/m + h^2 [4F(t) - F(t-h)]/6m. \\ &= 2x(t+h) - x(t) + h^2 F(t+h)/m \end{aligned} \quad (4.1.4.1-13)$$

但该算法下，速度比原来的 Verlet 算法更准确，结果使总能量守恒更好些。

Beeman 算法的缺点是速度的表达式并没有时间可逆对称性。这一问题一个很简单的解决方法是采用速度校正的 Verlet 算法，其中位置和速度的误差级为 $O(h^4)$ 。对 $x(t+2h)$ 、 $x(t+h)$ 、 $x(t-h)$ 和 $x(t-2h)$ 分别作 Taylor 展开，有

$$\begin{aligned} x(t+2h) &= x(t) + v(t)2h + \dot{v}(t)(2h)^2/2! + \ddot{v}(t)(2h)^3/3! + \dots \\ x(t+h) &= x(t) + v(t)h + \dot{v}(t)h^2/2! + \ddot{v}(t)h^3/3! + \dots \\ x(t-h) &= x(t) - v(t)h + \dot{v}(t)h^2/2! - \ddot{v}(t)h^3/3! + \dots \\ x(t-2h) &= x(t) - v(t)2h + \dot{v}(t)(2h)^2/2! - \ddot{v}(t)(2h)^3/3! + \dots \end{aligned} \quad (4.1.4.1-14)$$

联立这些方程后，得

$$12v(t)h = 8[x(t+h) - x(t-h)] - [x(t+2h) - x(t-2h)] + O(h^4)。 \quad (4.1.4.1-15)$$

或者是, 将 $v(t+h/2)$ 和 $v(t-h/2)$ 分别作 Taylor 展开, 得

$$\begin{aligned} v(t) &= [v(t+h/2) + v(t-h/2)]/2 + [\dot{v}(t-h) - \dot{v}(t+h)]/16 + O(h^4) \\ &= [v(t+h/2) + v(t-h/2)]/2 + [F(t-h) - F(t+h)]/16m \end{aligned} \quad (4.1.4.1-16)$$

注意, 式 (4.1.4.1-15) 中的速度 $v(t)$ 需要首先知道其后时间下的位置, $x(t+h)$ 和 $x(t+2h)$ 。而式 (4.1.4.1-16) 中的速度 $v(t+h/2)$ 需要其后时间的力 $F(t+h)$ 。

4.1.4.2 Gear 预报 - 校正法

分子动力学中最常用的预报 - 校正法来自于 Gear 设计的一组方法, 它由预报 - 力计算 - 修正等三步构成。其基本思想是用 t 时刻位置和初始的 n 阶导数的信息来预测在 $t+h$ 时刻位置和 n 阶导数; 计算在预测位置的力或加速度, 但可以发现这些加速度并不等于预测值; 因此, 用该误差作为校正信号调整对加速度的预测, 再进一步改进对位置和其它 $n-1$ 阶导数的估计。例如, $n=5$ 的情形下:

1) 预报: 将 $x(t+h)$ 用五阶 Taylor 级数展开, 于是, 每一步需要知道 1-5 阶导数, 它们也由 Taylor 展开给出:

$$x(t+h) = x(t) + \dot{x}(t)h + \ddot{x}(t)h^2/2! + x^{(3)}(t)h^3/3! + x^{(4)}(t)h^4/4! + x^{(5)}(t)h^5/5!; \quad (4.1.4.2-1)$$

$$\dot{x}(t+h) = \dot{x}(t) + \ddot{x}(t)h + x^{(3)}(t)h^2/2! + x^{(4)}(t)h^3/3! + x^{(5)}(t)h^4/4!; \quad (4.1.4.2-2)$$

$$\ddot{x}(t+h) = \ddot{x}(t) + x^{(3)}(t)h + x^{(4)}(t)h^2/2! + x^{(5)}(t)h^3/3!; \quad (4.1.4.2-3)$$

$$x^{(3)}(t+h) = x^{(3)}(t) + x^{(4)}(t)h + x^{(5)}(t)h^2/2!; \quad (4.1.4.2-4)$$

$$x^{(4)}(t+h) = x^{(4)}(t) + x^{(5)}(t)h; \quad (4.1.4.2-5)$$

$$x^{(5)}(t+h) = x^{(5)}(t)。 \quad (4.1.4.2-6)$$

2) 力计算: 计算作用力 $F(t+h)$, 即预测二阶导数 $\ddot{x}_p(t+h) = F(t+h)/m$, 其误差为

$$\Delta\ddot{x} = \ddot{x}(t+h) - \ddot{x}_p(t+h)。 \quad (4.1.4.2-7)$$

3) 校正: Gear 的二阶微分方程中的 n 阶校正为

$$x_c^{(m)} = x_p^{(m)} + C_m \Delta\ddot{x} \quad (m=1, \dots, n-1)。 \quad (4.1.4.2-8)$$

其中 C_m 为 m 阶导数的校正系数, 它们的选取原则是为了使算法最为稳定。 $n=4$ 时, $C_0 = \frac{1}{6}, C_1 = \frac{5}{6}, C_2 = 1, C_3 = \frac{1}{3}$; $n=5$ 时, $C_0 = \frac{19}{120}, C_1 = \frac{3}{4}, C_2 = 1, C_3 = \frac{1}{2}, C_4 = \frac{1}{12}$;

$n=6$ 时, $C_0 = \frac{3}{16}$, $C_1 = \frac{251}{360}$, $C_2 = 1$, $C_3 = \frac{11}{18}$, $C_4 = \frac{1}{6}$, $C_5 = \frac{1}{60}$ 。

4.1.4.3 算法的优劣判据

一个好的分子动力学模拟程序必须基于一个好的对 Newton 运动方程积分的算法。因此, 算法的选择极为重要。好的算法应该满足几个方面的要求。

1) 虽然计算的速度要求显得很重要, 但实际上真正花在积分运动方程上的时间很少, 至少对原子或简单分子体系是这样。

2) 准确度对较大的时间步长来说更重要, 因为所能使用的时间步长越长, 单位模拟时间内力的计算量越少。这意味着采用允许用长时间步长的成熟算法是有利的, 其代价是, 需要存储粒子坐标的高阶导数信息, 占据更多的内存。

3) 能量守恒是一个重要的判据, 但实际上需要区分两种能量守恒, 即短时间的和长时间的。复杂的高阶算法通常在短时间内 (如在几个时间步内) 有很好的能量守恒性, 然而时间较长时通常会有总能量漂移。相比较而言, Verlet 型的算法仅有中等的短时间能量守恒性, 但并没有长时间漂移。

因此, 初看起来, 最重要的要有一个能同时在短时间和长时间准确地预测所有粒子的轨迹算法。但事实上, 这样的算法并不存在。其根本原因是, 对于用分子动力学模拟所研究的体系, 其相空间轨迹敏感地依赖于初始条件。这意味着两个初始靠得很近的轨迹随时间的演绎将会显著分开。我们可以将算法所导致的积分误差作为在体系的真实轨迹和模拟所产生的轨迹间的初始细微差别的来源。可以认为, 任何积分误差, 不管它多么小, 总会导致模拟轨迹与同初始条件一致的真实轨迹显著分开。

4.1.4.4 算法的时间可逆性

牛顿运动方程是时间可逆的, 我们的算法也应如此。事实上, 包括预报-校正法在内的许多算法并不是时间可逆的。这就是说, 在这样的算法中, 未来和过去的相空间坐标并不对称, 如在某一瞬间将所有粒子的动量反向, 体系并不在相空间上倒溯它的轨迹, 即便是模拟可以无穷的数值精度进行。只有在无穷短的时间步长的极限情况下, 这样的算法才是可逆的。

更为重要的是, 许多似乎合理的算法与 Hamilton 运动方程在另一方面不同: 真实的 Hamilton 动力学保持相空间体积元的大小不变, 但许多数值方法, 尤其是那些时间不可逆的方法不重现这一体积保持性。考虑对应于一个特定能量的所有的轨迹, 它们被包含在相空间的一个体积 Ω 中。如果让 Hamilton 运动方程作用于这一体积中的所有点上 (即让体积随时间演绎变化), 那么一段时间后仍是同样的体积。然而, 一个非体积保持的算法将会映射体积 Ω 到另一较大的体积 Ω' 上。足够长时间之后, 将会大大地扩展在相空间中体系的体积。这与能量守恒是不相容的。因此, 有理由认为不可逆算法将会有严重的长时间能量漂移问题。可逆的算法不会改变相空间中体积元的大小, 这一性质还不足以保证不会有长时间能量漂移, 但至少它是与其一致的。

通过计算老的到新的相空间坐标变换的 Jacobi 行列式是否为 1, 可以检查一个算法是否是体积保持的。最后, 应注意的是, 即便我们积分一个时间可逆的算法, 在数值实现时, 也几乎不可能是真正时间可逆的。这是因为浮点运算的有限精度导致截断误差。

至于 Verlet 算法在上述几点上的优劣表现为: 1) 它是最快的算法, 但这一点相对来说并不重要; 2) 它对于长时间步不是特别准确, 因此应当更频繁地计算作用力; 3) 它需要的内存最少, 这在模拟较大体系时是有用的, 但一般来说这并不是一个关键的优点; 4) 它的短时间能量守恒性尚可, 尤其是用的较准确的速度型表达式时; 5) 最重要的是它很少有长时间能量漂移现象, 这是由其时间可逆性和体积保持性造成的。事实上, 尽管 Verlet 算法并不精确地使体系总能量不变, 但在无限短的时间步长的极限时, 它确实使一个拟 Hamilton 函数守恒, 来逼近真实的 Hamilton 函数。

4.1.4.5 Verlet 算法的 Liouville 公式

前面我们从计算方法的角度讨论了积分牛顿运动方程的不同算法。至于什么使得一个算法成为一个好算法, Liouville 定理提供了许多深入见解。

考虑经典多体体系中, 任一依赖于 N 个粒子的坐标和动量的函数 f , 它通过坐标和动量的时间关系而依赖于时间, 其时间导数为

$$\dot{f} = \frac{df}{dt} = \sum_i \left(\dot{q}_i \frac{\partial f}{\partial q_i} + \dot{p}_i \frac{\partial f}{\partial p_i} \right) \equiv \dot{\mathbf{q}} \cdot \frac{\partial f}{\partial \mathbf{q}} + \dot{\mathbf{p}} \cdot \frac{\partial f}{\partial \mathbf{p}} \equiv i\hat{L}f \quad (4.1.4.5-1)$$

这里定义了 Liouville 算子 \hat{L} , 其中矢量表示 N 个粒子的坐标。可对上式形式积分得到

$$f[\mathbf{p}(t), \mathbf{q}(t)] = \exp(i\hat{L}t) f[\mathbf{p}(0), \mathbf{q}(0)]。 \quad (4.1.4.5-2)$$

将 Liouville 算子分为两项, 分别为

$$i\hat{L}_q \equiv \dot{\mathbf{q}} \cdot \frac{\partial}{\partial \mathbf{q}}; \quad i\hat{L}_p \equiv \dot{\mathbf{p}} \cdot \frac{\partial}{\partial \mathbf{p}}, \quad (4.1.4.5-3)$$

如果式 (4.1.4.5-2) 仅有第一个位置算子, 则 Taylor 展开后可得

$$\begin{aligned} f(t) &= \exp(i\hat{L}_q t) f(0) = f(0) + i\hat{L}_q t f(0) + \frac{(i\hat{L}_q t)^2}{2!} f(0) + \dots \\ &= \exp \left[\dot{\mathbf{q}}(0) t \cdot \frac{\partial}{\partial \mathbf{q}} \right] f(0) = \sum_{n=0}^{\infty} \frac{[\dot{\mathbf{q}}(0) t]^n}{n!} \cdot \frac{\partial^n}{\partial \mathbf{q}^n} f(0) \quad , \quad (4.1.4.5-4) \\ &= f \left\{ \mathbf{p}(0), [\mathbf{q}(0) + \dot{\mathbf{q}}(0) t] \right\} \end{aligned}$$

因此, 位置算子 $\exp(i\hat{L}_q t)$ 的作用仅是对位置进行移动。相似地, 动量算子 $\exp(i\hat{L}_p t)$ 的影响也仅是动量的移动。虽然总的 Liouville 算子等于 $i\hat{L} = i\hat{L}_q + i\hat{L}_p$, 但是 $\exp(i\hat{L}t)$ 不等于 $\exp(i\hat{L}_q t) \exp(i\hat{L}_p t)$, 这是因为它们是非对易算子。因此, 我

们利用非对易算子 A 和 B 的 Trotter 恒等式,

$$\begin{aligned}\exp(\hat{A} + \hat{B}) &= \lim_{P \rightarrow \infty} \left[\exp(\hat{A}/2P) \exp(\hat{B}/P) \exp(\hat{A}/2P) \right]^P \\ &= \left[\exp(\hat{A}/2P) \exp(\hat{B}/P) \exp(\hat{A}/2P) \right]^P \exp[O(P^{-2})]\end{aligned}\quad (4.1.4.5-5)$$

定义

$$\frac{\hat{A}}{P} \equiv \frac{i\hat{L}_p t}{P} \equiv h\dot{\mathbf{p}}(0) \cdot \frac{\partial}{\partial \mathbf{p}}; \quad \frac{\hat{B}}{P} \equiv \frac{i\hat{L}_q t}{P} \equiv h\dot{\mathbf{q}}(0) \cdot \frac{\partial}{\partial \mathbf{q}}, \quad (4.1.4.5-6)$$

式中 $h = t/P$ 为离散的时间步长。则 (4.1.4.5-2) 式中, 一个时间步长对应着运用算子 $\exp(i\hat{L}_p h/2) \exp(i\hat{L}_q h) \exp(i\hat{L}_p h/2)$ 一次。我们来看看该算子对粒子位置及动量的作用结果。首先是算子 $\exp(i\hat{L}_p h/2)$ 的作用,

$$\exp(i\hat{L}_p h/2) f[\mathbf{p}(0), \mathbf{q}(0)] = f\left[\mathbf{p}(0) + \dot{\mathbf{p}}(0) \frac{h}{2}, \mathbf{q}(0)\right]. \quad (4.1.4.5-7)$$

接下来, 将 $\exp(i\hat{L}_q h)$ 作用于上式, 注意 $\dot{\mathbf{q}}$ 中时间已经过了 $h/2$,

$$\exp(i\hat{L}_q h) f\left[\mathbf{p}(0) + \dot{\mathbf{p}}(0) \frac{h}{2}, \mathbf{q}(0)\right] = f\left[\mathbf{p}(0) + \dot{\mathbf{p}}(0) \frac{h}{2}, \mathbf{q}(0) + \dot{\mathbf{q}}\left(\frac{h}{2}\right) h\right]. \quad (4.1.4.5-8)$$

最后, 再运用算子 $\exp(i\hat{L}_p h/2)$ 一次, 注意多了一个 $\dot{\mathbf{p}}$, 其中时间又过了 h ,

$$\begin{aligned}\exp(i\hat{L}_p h/2) f\left[\mathbf{p}(0) + \dot{\mathbf{p}}(0) \frac{h}{2}, \mathbf{q}(0) + \dot{\mathbf{q}}\left(\frac{h}{2}\right) h\right] \\ = f\left[\mathbf{p}(0) + \dot{\mathbf{p}}(0) \frac{h}{2} + \dot{\mathbf{p}}(h) \frac{h}{2}, \mathbf{q}(0) + \dot{\mathbf{q}}\left(\frac{h}{2}\right) h\right]\end{aligned}\quad (4.1.4.5-9)$$

算子作用的每一步对应着一次简单的位移运算。特别注意到, q 的位移仅是 p 的函数 (因为 $\dot{q} = p/m$), 而 p 的位移仅是 q 的函数 (因为 $\dot{p} = F/m$)。从 $\{q(0), p(0)\}$ 到 $\{q(h), p(h)\}$ 的变换的 Jacobi 行列式仅是三个基本变换的 Jacobi 行列式的积。但由于每一个 Jacobi 行列式等于 1, 总的 Jacobi 行列式也等于 1。也就是说, 这一算法是体积保持的。

现在考虑这一串运算对位置和动量的总的影晌, (4.1.4.5-9) 表明:

$$p(h) = p(0) + \dot{p}(0) \frac{h}{2} + \dot{p}(h) \frac{h}{2} = p(0) + \frac{h}{2} [F(0) + F(h)]; \quad (4.1.4.5-10)$$

$$q(h) = q(0) + \dot{q}\left(\frac{h}{2}\right) h = q(0) + h\dot{q}(0) + \frac{h^2}{2m} F(0). \quad (4.1.4.5-11)$$

这些就是速度 Verlet 算法的方程 (4.1.4.1-5) - (4.1.4.1-6) 式。

4.1.5 碰撞轨迹的失稳性

4.1.5.1 Lyapunov 失稳性

相空间轨迹对初始条件的细微差别是极为敏感的。仍用 \mathbf{q} 表示 N 个粒子体系相空间中轨迹在时刻 t 时的位置坐标，它是 $t=0$ 时刻初始位置和动量的函数

$$\mathbf{q}(t) = f[\mathbf{p}(0), \mathbf{q}(0); t]. \quad (4.1.5.1-1)$$

如果扰动初始条件，如对动量增加一个小量 δ ，在 t 时刻可以得到不同的位置坐标值，

$$\mathbf{q}'(t) = f[\mathbf{p}(0) + \delta, \mathbf{q}(0); t], \quad (4.1.5.1-2)$$

记 \mathbf{q} 与 \mathbf{q}' 之间的差为 $\Delta\mathbf{q}$ 。足够短的时间之内， $\Delta\mathbf{q}$ 与 δ 和 t 成正比。但长时间下，其相关系数呈时间的指数发散，即

$$|\Delta\mathbf{q}| \sim \delta \exp(\mu t), \quad (4.1.5.1-3)$$

轨迹的这一行为即是所谓的 Lyapunov 失稳性，它是我们不能准确地预测轨迹的根源。

指数 μ 被称为 Lyapunov 指数（更准确的说法是最大 Lyapunov 指数，实际上有 $6N$ 个 Lyapunov 指数，但其中最大的指数决定了初始靠近的那些轨迹在长时间后会指数分离并发散开来）。假设希望在时间间隔 $0 < t < t_{\max}$ 内对 $\Delta\mathbf{q}$ 保持一个一定的限度 Δ_{\max} ，那么所能承受的初始误差则为， $\delta \sim \Delta_{\max} \exp(-\mu t_{\max})$ 。因此，初始条件的可接受的误差随运行时间长度 t_{\max} 呈指数下降。

图 (4.1.5.1-1) 显示两个几乎相同条件下的模拟结果，第二个模拟与第一个的差别仅在于 1000 个粒子中的两个粒子在 x 方向的速度改变了 $+10^{-10}$ 和 -10^{-10} （相对速度单位）。统计所有粒子位置差的平方和后，可以发现，均方偏差值 $\sum_{i=1}^N |\mathbf{r}_i(t) - \mathbf{r}_i'(t)|^2$ 的确随时间指数增长。1000 时间步以后，初始很接近的两个体系变得几乎完全不相关。模拟中使用的完全是通常的参数（密度、温度、时间步长），时间只有 1000 步，而大多数分子动力学模拟需要 10^5 步。

但该 Lyapunov 失稳性对整个分子动力学模拟结果来说，问题不是很严重。这是因为，分子动力学模拟的目标并不是精确地预测一个已知初始条件的体系将会发生什么，而是对其平衡态的统计平均感兴趣。我们希望预测一个已知其部分信息（如总能量）而不是全部信息的初始态体系的平均行为，而不是要知道相空间中的准确真实轨迹。因此，它与预测卫星在空间轨道的数值方法有本质不同。在后一种情况下，我们必须预测真实的轨道，而不能作统计预测。然而，在分子动力学模拟中，只要保证各态历经性，长时间的计算时间步下，轨迹基本上是随机性的对相空间进行了抽样，这时的统计预测是足够好。

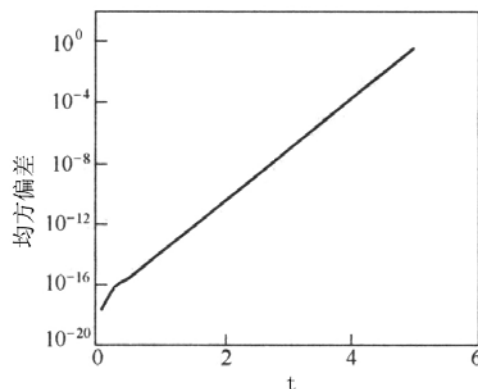


图 4.1.5.1-1 Lennard-Jones 体系模拟中的 Lyapunov 失稳性。两个初始靠得很近的轨迹之间均方距离和与计算时间步的依赖性。 $t=5$ 为 1000 个时间步。

4.1.5.2 碰撞失稳性简例

为了理解碰撞失稳性的起因,以及决定性运动方程是怎样产生随机性的运动轨迹的,我们现在考察两个简单的经典力学中的碰撞体系。设一个半径为 σ 的硬圆盘限制在半径为 R 的二维圆周内运动,模型A中该圆周是连续光滑的全反射硬壁,而模型B中圆周是由固定的32个相同硬圆盘组成。因为只有一个圆盘在运动,故相空间是4维的,圆盘与壁发生弹性碰撞,故总能量即粒子动能守恒,相空间中粒子轨迹在一个3维的恒定能量曲面上运动。

设粒子运动速度 \mathbf{v} 的径向分量为 \mathbf{v}_r ,圆切线方向的分量为 \mathbf{v}_t 。模型A中,粒子与壁碰撞并全反射后, $\mathbf{v}_r \rightarrow -\mathbf{v}_r$, $\mathbf{v}_t \rightarrow \mathbf{v}_t$ 。图(4.1.5.2-1)中,取 $R=5\sigma$,圆盘初始位置为 $x(0)=y(0)=-R/\sqrt{2}$ 。首先取初始速度为 $v_x(0)=2$, $v_y(0)=1.4$,总能量为2.98(圆盘质量取为1),经过100次碰撞后的轨迹基本上是周期性的,呈9星型;然后对初始条件进行微扰,将初始速度取为 $v'_x(0)=2.0010$, $v'_y(0)=1.3986$,总能量仍然保持为2.98,因此该扰动轨迹也在同一能量曲面上,只不过初始速度方向发生了0.12%的变化。扰动轨迹是准周期性的,两种轨迹间的差为

$$d(t) = \sqrt{[x(t)-x'(t)]^2 + [y(t)-y'(t)]^2}, \quad (4.1.5.2-1)$$

计算结果说明,该偏差是与时间成正比的,

$$d(t) \propto t, \quad (4.1.5.2-2)$$

经过100次碰撞后,其偏差值也不到一个 σ 。相空间轨迹在由能量曲面规定的允

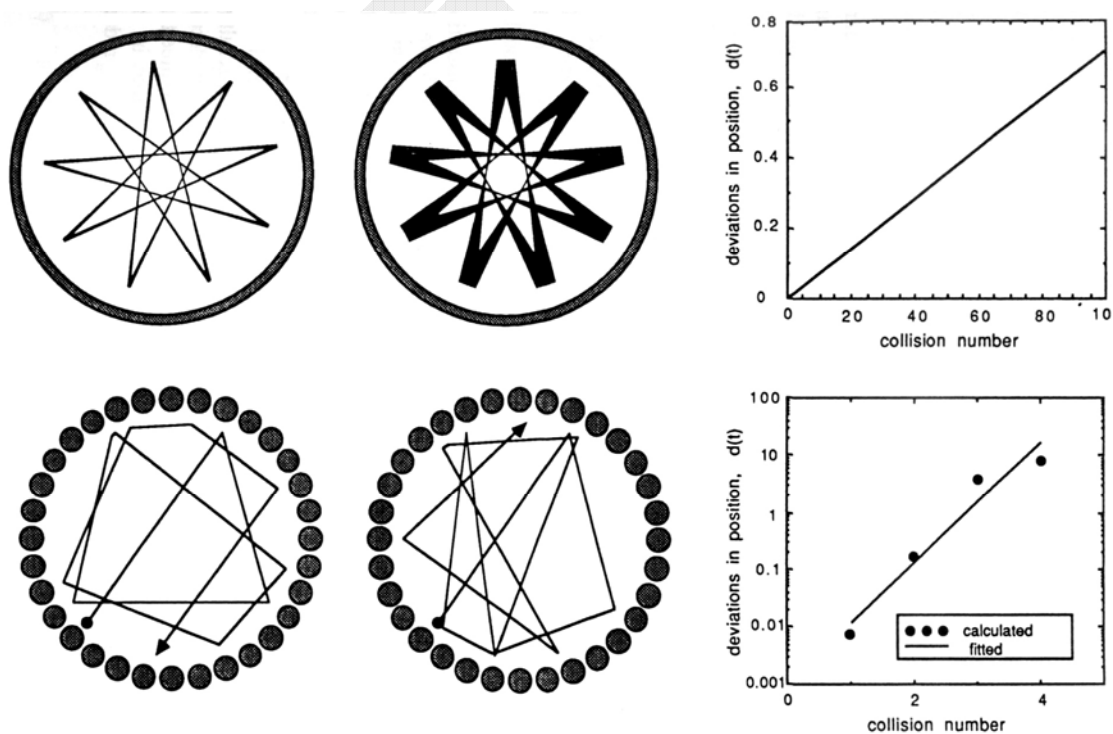
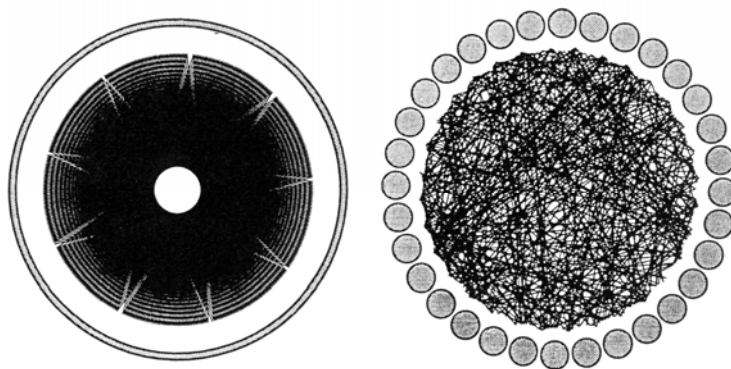


图4.1.5.2-1(上) 圆盘与圆环边界的碰撞轨迹稳定性。(左) 准周期性轨道;(中) 初始速度微扰后的轨道;(右) 两个轨迹间的偏离与时间成线性关系。

图4.1.5.2-2(下) 圆盘与圆盘边界的碰撞轨迹稳定性。(左) 无规轨道;(中) 初始速度微扰后的轨道;(右) 两个轨迹间的偏离与时间成指数增长关系。

图4.1.5.2-3 经过400次碰撞后，准周期性轨迹（左）与混沌轨迹（右）的比较。



许的区域缓慢有规则的移动，只要时间足够长，表面上的每一点都将经过（各态历经性）。

而在模型B中，碰撞后速度的变化规则同上，只不过需要确定不同碰撞点处的局域表面法向。图（4.1.5.2-2）中，计算的参数和条件与图（4.1.5.2-1）中是相同的，因此相空间中轨迹是在和模型A中限定的相同区域中运动。图中显示12次碰撞后的轨迹，可见，在这个模型中，轨迹不是有规则的，对初始条件的一点小微扰就足以完全改变轨迹的整个型态，其偏差是随时间而成指数上升趋势，

$$d(t) \propto \exp(\lambda t), \quad (4.1.5.2-3)$$

经过4次碰撞后，其偏差值已经达到圆周直径大小（ 10σ ）。

因此，模型B中的轨迹是非常不稳定的，在构型空间中即是以随机的方式覆盖整个可能的运动区域，轨迹呈混沌状；而模型A中轨迹是规则准周期性的（图4.1.5.2-3）。两个模型的差别仅在于圆盘是如何与边界相碰撞的。模型A中，径向（表面法向）速度和切向速度是独立的，相应的动能分量也各是守恒的，由于体系有两个自由度和两个守恒量，因此运动是周期或准周期的；模型B中，由于边界的作用，径向速度和切向速度是耦合在一起的，所以只有总能一个量是守恒量，故运动是混沌型的。这个例子形象地说明了碰撞失稳性的起因。在分子动力学的多体相互作用中，实际情形更为复杂。不仅一个粒子要与多个粒子碰撞，而且弹性碰撞时有能量转移，粒子的速度会发生变化。在高密度下，一个粒子将容易陷入周围的粒子包围中。

由此，我们可知，经典力学的决定性方程可以在构型空间中造成随机性轨迹，在相空间中也是如此。算法的误差（截断误差）、浮点数精度的舍入误差、初始值的任意小的不确定性都可能使轨迹结果发生极大变化。但重要的是，只要保证总能量守恒，这些轨迹都可以对相空间中的允许运动区域进行随机性抽样，保证长时间平均结果不变。因此，采用高阶算法没有实际益处，而且初始值可以随意给定，它们在计算中的实际效果只是改变计算时间而已，对统计结果的准确性没有什么影响。

4.1.6 原子间相互作用势模型

原子间的结合力决定了材料的结构以及其它物性，现在我们要问，对于真实材料，势能函数 $U(\mathbf{r}_1, \dots, \mathbf{r}_N)$ 如何确定？真实材料中，通常有四种原子间的结合键，即金属键、离子键、共价键、van der Waals 键。van der Waals 力一般是非常弱的，其它三种键合的角色主要是由价电子扮演的。实际作用势函数模型必须基于量子力学。

4.1.6.1 Born-Oppenheimer 近似

相互作用的原子体系实际上是由核和电子构成的，它们之间有着相互作用，体系的 Hamilton 量为，

$$\hat{H} = \sum_i \frac{\hat{\mathbf{p}}_i^2}{2M_i} + \sum_n \frac{\hat{\mathbf{p}}_n^2}{2m} + \frac{1}{2} \sum_{ij} \frac{Z_i Z_j e^2}{|\mathbf{R}_i - \mathbf{R}_j|} + \frac{1}{2} \sum_{mn'} \frac{e^2}{|\mathbf{r}_n - \mathbf{r}_{n'}|} - \sum_{in} \frac{Z_i e^2}{|\mathbf{R}_i - \mathbf{r}_n|}, \quad (4.1.6.1-1)$$

其中，下标 i 和 j 表示原子核， n 和 n' 表示电子。原则上，应该求解 Schrodinger 方程得到波函数 $\Psi(\mathbf{R}_i, \mathbf{r}_n)$ ，再以此推导体系的其它物理量。但实际上，对约含有大量粒子组成的多体体系，该解几乎是不可能准确得到的。

由于核的质量远大于电子，其运动时间尺度要比电子的长两个数量级， $\omega_{\text{电子}}/\omega_{\text{核}} \sim \sqrt{M/m} \sim 100$ ，因此可将核认为是不动的，只需考虑电子的运动。在 Born-Oppenheimer 近似下，体系的总波函数可写为核的部分与电子部分之积， $\Psi(\mathbf{R}_i, \mathbf{r}_n) = \Xi(\mathbf{R}_i) \Phi(\mathbf{r}_n; \mathbf{R}_i)$ ，电子波函数 $\Phi(\mathbf{r}_n; \mathbf{R}_i)$ 中核的坐标作为参数。于是，Schrodinger 方程分解为两个方程：

$$\hat{H}_{el} \Phi(\mathbf{r}_n; \mathbf{R}_i) = U(\mathbf{R}_i) \Phi(\mathbf{r}_n; \mathbf{R}_i); \quad (4.1.6.1-2)$$

$$\hat{H}_{el} = \sum_n \frac{\hat{\mathbf{p}}_n^2}{2m} + \frac{1}{2} \sum_{ij} \frac{Z_i Z_j e^2}{|\mathbf{R}_i - \mathbf{R}_j|} + \frac{1}{2} \sum_{mn'} \frac{e^2}{|\mathbf{r}_n - \mathbf{r}_{n'}|} - \sum_{in} \frac{Z_i e^2}{|\mathbf{R}_i - \mathbf{r}_n|}; \quad (4.1.6.1-3)$$

$$\sum_i \left[\frac{\hat{\mathbf{p}}_i^2}{2M_i} + U(\mathbf{R}_i) \right] \Xi(\mathbf{R}_i) = E \Xi(\mathbf{R}_i), \quad (4.1.6.1-4)$$

(4.1.6.1-3) 式是电子部分的方程，其中能量本征值 $U(\mathbf{R}_i)$ 依赖于核坐标，它就是原子间势能函数的定义。注意，所有电子的自由度已经含在该电子的能量本征值中。然而，它在式 (4.1.6.1-4) 中却决定了原子核的运动。一旦我们知道了该势能函数，在经典极限下可以用 Newton 运动方程代替核的 Schrodinger 方程。

经典近似成立的条件是，粒子的热运动 de Broglie 波长 $\Lambda = \hbar \sqrt{2\pi/Mk_B T}$ 远比粒子间距 a 要小。对于锂、氩等元素，液态时 $\Lambda/a \sim 0.1$ ，重元素更小。因此，对这些元素，分子动力学计算中采用 Newton 方程是很好的近似。但是，对于氢、氘、氚等轻元素近似较差。不管是什么体系，当温度 T 非常低时，量子效应都是非常重要的。

4.1.6.2 两体势与多体势模型

但是,上述方程并没有真正解决势能函数问题。可能的解决途径有两条。第一种方法就是经验地选择某些能够体现真实材料中势效应的函数形式,即经验势模型。另外一种方法是在某种近似下同时求解描述电子运动的方程(4.1.6.1-2)式和核运动的 Newton 方程,这就是量子的第一原理(或称从头计算法)分子动力学方法,它要求大量的计算资源,而无需经验势。这里我们只考虑经验势模型。

早期的分子动力学模拟常采用经验性径向对称势描述原子间的相互作用,函数形式为指数形或高次多项式,势能中的参数可通过与块材的物理性质拟合获得。势能函数中,原子间的相互作用势能是按一对一对的贡献给出的。LJ 势能是最常用的两体势,它特别适用于惰性气体元素。对这些元素原子,没有外层电子可用于成键,因此,原子间的相互作用仅仅是通过 van der Waals 力。其它两体势如 Morse 势(2.3.4.1-5)式等也适合于富勒烯(C_{60})和 van der Waals 键占优势的分子晶体。实际研究的往往是金属和半导体等材料,它们不能用两体势模型处理。

将金属的一些物理性质和 LJ 势函数模型产生的结果相比较,就可以发现这些材料中的多体效应。如内聚能和熔点的比值 $E_c/k_B T_m$ 约为 30,而 LJ 的结果约为 10,说明金属的内聚能力要比两体势能体系远要高,两体势对晶态的贡献不大。实际立方晶体的两个弹性常数比 C_{12}/C_{44} 约为 1.5~3.7,而两体势模型下,可解析地推得为 1(Cauchy 关系),金的高比值(3.7)正表示了它的很好的延展性。对半导体来说,两体势给出的结果更差。如压力下,硅经历四面体 $\rightarrow\beta$ 锡 \rightarrow 简立方 \rightarrow 面心立方的相变,说明结构之间的能量差异很大,内聚能与配位数没有太大的关系,但是两体势模型都比较偏好于密堆积结构。

现今,已提出许多多体函数形式用来更多地表征化学键的物理和化学性质。然后,还需要具体求出其中的参数以用于分子动力学计算。这类模型或多或少都有着唯象模型的痕迹。第一原理分子动力学方法也可以用来产生经典分子动力学方法中所需的多体经验势。

4.1.6.3 金属的多体势

构造金属的多体势能基于的物理思想是:原子所在处的局域电子密度是关键变量,而局域电子密度依赖于原子附近的局域密度或配位数,当原子局域环境非常密集时,键合作用将减弱。按此思想构造出的势模型都具有如下的泛函形式:

$$U = \sum_{i=1}^N u(n_i) + \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^N V(r_{ij}), \quad (4.1.6.3-1)$$

其中 V 是描述排斥作用部分的两体势,一般通过拟合实验数据得到。 $u(n_i)$ 表示吸引相互作用,它是第 i 个原子近邻的广义配位数 n_i 的函数, n_i 则为近邻原子密度的迭加,

$$n_i = \sum_{j=1 \neq i}^N \rho(r_{ij}). \quad (4.1.6.3-2)$$

不同的势模型有不同的参数形式，它们甚至是用分子动力学方法计算出的结果进行调整而得到的。

该势函数形式下的力为，

$$\mathbf{F}_i = - \sum_{j \neq i} \left\{ V'(r_{ij}) + [u'(n_i) + u'(n_j)] \rho'(r_{ij}) \right\} \frac{\mathbf{r}_{ij}}{r_{ij}}, \quad (4.1.6.3-1)$$

其计算仅比两体势稍微复杂一些，势函数和力都只依赖于两体间距，并无3体或多体项。由于势能仍是径向距离的函数，没有角度或方向依存性，因此难于用来模拟过渡金属等共价键效应较强的金属。

4.1.6.4 半导体的多体势

参考文献

- [1] J.M. Haile, Molecular Dynamics Simulation (Wiely, 1992)。
- [2] F. Ercolessi, A Molecular Dynamics Primer, <http://www.fisica.uniud.it/~ercolessi/>