科学通报 2009 年 第 54 卷 第 6 期: 734 ~ 739
论 文 www.scichina.com csb.scichina.com
《中国科学》杂志社
SCIENCE IN CHINA PRESS

分子动力学中平行六面体的邻位算法

崔志伟①, 孙毅①*, 曲建民②

① 哈尔滨工业大学航天科学与力学系, 哈尔滨 150001;
② G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta 30332-0405, USA
* 联系人, E-mail: sunyi@hit.edu.cn

2008-10-10 收稿, 2009-01-12 接受
国家自然科学基金(批准号: 10472028)、高等学校博士学科点专项科研基金和黑龙江省杰出青年基金资助项目

摘要 在分子动力学模拟中, Verlet 列表法、元胞链接列表法和其他一些改进的算法已经被用来提高计算效率. 然而, 这些方法大部分应用在立方体系. 本文扩展了原始算法的应用领域, 使其适用于 NP 系综, 并可以在外载荷作用下进行系统仿真. 此外, 对于 Metric-tensor 的压强算法提出相应的标度矩阵, 使得邻位算法可以应用在新的压强算法上. 同时, 还针对平行六面体盒子提出了一种联合算法. 该方法综合了改进的 Verlet 列表法和元胞列表法的优点, 可以快速确定平行六面体盒子中每个原子的邻位原子, 从而提高仿真速度. 最后, 利用 Lennard-Jones 势函数的分子动力学模拟实验验证了本算法的有效性和正确性.

关键词
分子动力学模拟
邻位算法
平行六面体盒子

在分子动力学中, Verlet列表法[1]、元胞链接法[2]以及一些改进算法[3~10]的邻位算法能减少短程相互作用势的计算消耗. 为简单起见, 大部分的邻位算法都以立方体作为模拟系统来阐述其原理. 因此, 这些算法看起来似乎可以直接应用到平行六面体系统中, 但事实上, 只有对原始的邻位算法进行改进才能正确地应用在非立方体系统上.

在很多的分子动力学模拟中, 需要使用平行六面体盒子[11~14]. 例如, 在摩擦力、应力等载荷作用下模拟材料的力学响应以及结构相变等, 都需要使用可变形的盒子.

材料的弹性常数是一个很重要的力学参数, 因为它能反映材料对载荷的响应以及势函数的准确性[15]. 计算弹性常数通常使用NPT系综下的应变涨落方法[16~18]. 而现在的邻位算法由于其局限性不能进行此类计算. 此外, 在模拟相变时, 形状和体积的变化也存在相同的问题[19]. 上述问题说明, 需要一种可变形盒子的新型加速算法.

本文把原始的邻位算法扩展到平行六面体的情况. 对于立方体系统的情况, 新方法虽然在效率上略逊于现有方法. 然而, 新方法可以普遍应用于 NV, NP 系综中. 为了证明算法的有效性和正确性, 本文进行了多次仿真, 并进行了比较.

1 方法

(ⅰ) Metric-tensor算法中的标度矩阵. 本文中, 我们采用Metric-tensor压强算法[20], 由于它能完美地修正 Parrinello 和 Rahman 压强算法[21]的缺点. 在Metric-tensor算法中, 引入一个对称矩阵$G=h^{\mathrm{T}}h$, 作为系统的动力学变量, 只有 6 个独立的变量: 3 个描述盒子边长的主对角线元素, 3 个描述盒子夹角的非主对角线元素, 可以完全描述仿真盒子的大小和形状. 此外, G最大的优势在于, 它的演化与h无关, 也就是说与空间笛卡尔坐标系无关. 然而, 正是这个优点使其无法导出原子的位置和速度等量. 此外, Verlet列表法需要通过h来计算所有原子的位移. 因此在本节中, 通过一个固定的空间笛卡尔坐标系来解决这个问题. 当然, 在得到包含当前晶格矢量$\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$的

引用格式: 崔志伟, 孙毅, 曲建民. 分子动力学中平行六面体的邻位算法. 科学通报, 2009, 54(6): 734~739
Cui Z W, Sun Y, Qu J M. The neighbor list algorithm for a parallelepiped box in molecular dynamics simulations. Chinese Sci Bull, 2009, 54, doi: 10.1007/s11434-009-0197-0

标度矩阵H的同时，保留了Metric-tensor算法的优点.

我们知道, Metric-tensor 算法与系统的方向无关，因此系统对于刚体转动是不变的. 不失一般性，笛卡尔坐标系可以通过以下方式和系统固定在一起: x 轴和晶格矢量 $\boldsymbol{a}$ 重合, y 轴位于晶格矢量 $\boldsymbol{a}$, $\boldsymbol{b}$ 组成的平面内. 这里需要强调的是，空间笛卡尔坐标系的引入仅仅起到一个桥梁作用. 通过它可以把压强算法和其他需要笛卡尔坐标系的算法联系起来，同时对于原始的 Metric-tensor 算法没有任何影响. 通过上述参考坐标系的建立，可以得到标度矩阵 H. 注意，坐标系的选择不是唯一的. 前面给出的坐标系(图 1)只是为了方便. 下面的 H 为上三角阵:

$$H=\begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} =\begin{bmatrix} \sqrt{G_{11}} & G_{12}/H_{11} & G_{13}/H_{11} \\ 0 & \sqrt{G_{22}-H_{12}^2} & (G_{23}-H_{12}H_{13})/H_{22} \\ 0 & 0 & \sqrt{G_{33}-H_{13}^2-H_{23}^2} \end{bmatrix}.$$

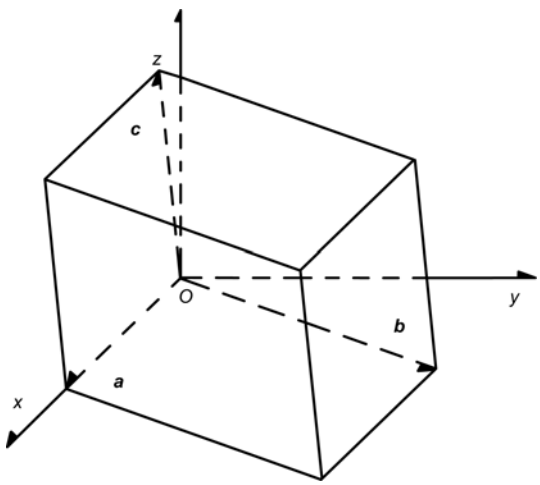

图 1　Metric-tensor 算法中笛卡尔坐标系和标度矩阵 H

可以注意到这里H只有 6 个非零项. 这是因为通过x轴和xy平面固定于盒子上减少了 3 个刚体转动自由度. 这种参考坐标系使H的独立元素从一般的 9 个减少到了 6 个，与G相同. 通过H, 所有系统信息都可以得到在空间笛卡尔参考坐标系下的结果. 注意，这种算法既不同于上三角阵[22]，也不同于包含晶格矢量的对称矩阵[23]. 后者是基于标度矩阵h的，h在演化的同时还要满足一些约束以保持其特点. 我们提出的方法通过矩阵G计算得到，G在随着时间变化的过程中并没有引入任何约束.

当然，这里不可避免地需要一些额外的时间花销来计算矩阵 H. 然而，新算法允许使用 Verlet 列表法，这样可以在很大程度上节省这些额外的时间开销.

(ii) 平行六面体的Verlet列表法. 对于相对较小或者接近平衡态的系统，传统的Verlet列表法是非常有效的[10,24]. 使用此算法，每个原子都有一个包含周围半径r_s以内原子的邻位列表, r_s比截断半径r_c略大，例如 r_s=1.1r_c. 原则是: 当 $\max_i|\Delta\boldsymbol{r}_i|=\max_i|r_i(t)-r_i(t_0)|>(r_s-r_c)/2$ 成立，邻位列表就要更新. 对于平行六面体盒子，原子位移 $|\Delta\boldsymbol{r}_i|$ 则改成 $|h(t)s_i(t)-h(t_0)s_i(t_0)|$，其中 $0\le s_i<1$. 由于h和h_0是不同的，因此，引入一个标志数组(FlagAtom)记录仿真系统穿越边界的次数(图 2). FlagAtom的初始值为零. 如果当前时刻的s大于 1，为了保证系统总的原子数N的不可变性，要用s–1 替换s. 然而，该原子在笛卡尔坐标系下的真正坐标是$h(s+1)$，而不是hs. 这种情况在s<0 时也会出现. 相应的，当$s\ge1$ 时标志数组应该加 1，当s<0 时标志数组应该减 1. 因此，每个原子的真实位移应该通过h(s+FlagAtom)–h_0s_0 来计算. 更新Verlet列表后，s就成为s_0，并且将FlagAtom清零. 也就是说，更新标志通过标志数组和周期性边界条件共同完成. 有了标志数组，新的Verlet算法不仅可以应用于NV系综，还可以应用到NP系综.

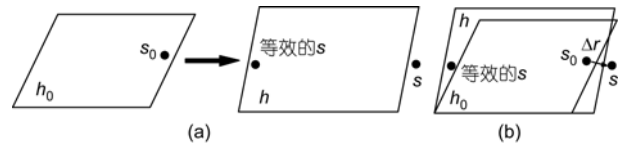

图 2　平行六面体的 Verlet 列表法

(a) h 和 s 的演化; (b) 真实的位移 Δr

(iii) 保守的最大截断半径. 对于立方体盒子，最大截断半径r_{cmax}应该满足$r_{cmax}\le L/2$ 以避免重复计算，其中L是盒子的边长. 由于周期性边界条件，这个关系同样也适用于平行六面体系统(图 3(a)). 偏离出本系统的部分可以由周围系统渗入的部分等价补偿. 然而，如文献[25]中所述，在某些特殊的场合需要修正此准则. 这里，我们提出一个更保守的准则来描述最大截断半径. 对于平行六面体盒子，边长的一

半$L/2$ 可以被一个更通用的距离——内切球半径代替, 即$r_{c\max} \leq r_{\text{ins}}$ (图 3(b)). 通过简单的几何计算, 可以得到

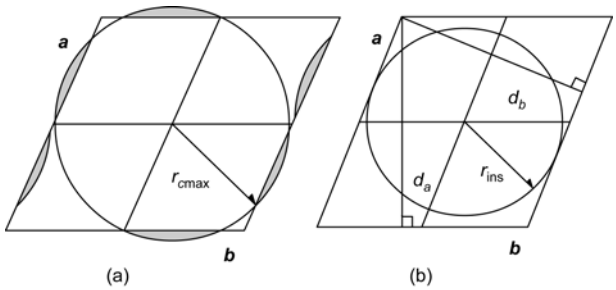

图 3 截断半径

(a) 原始定义; (b) 保守定义

$$r_{\text{ins}} = \frac{1}{2}\min\{d_a, d_b, d_c\} \leqslant \frac{1}{2}\min\{|\boldsymbol{a}|, |\boldsymbol{b}|, |\boldsymbol{c}|\}, \tag{1}$$

$$d_a = \frac{V}{|\boldsymbol{b}\times\boldsymbol{c}|}, \quad d_b = \frac{V}{|\boldsymbol{a}\times\boldsymbol{c}|}, \quad d_c = \frac{V}{|\boldsymbol{a}\times\boldsymbol{b}|}, \tag{2}$$

$$V = \det h = \boldsymbol{a}\cdot(\boldsymbol{b}\times\boldsymbol{c}), \tag{3}$$

其中 d_a, d_b, d_c 是三对平行平面间的垂直距离(图 3(b)). 对于立方体, d_a, d_b, d_c 就是 L. (1)式定义的最大截断半径非常保守. 在大规模系统中, 通常使用元胞列表法来加速分子动力学模拟, 元胞列表法依靠截断半径的定义把盒子分割成子格.

(iv) 平行六面体的元胞列表法. 元胞链接列表法在大规模的系统或者处于非平衡态的情况下能有效地提高计算效率. 其基本思想是: 将盒子按照一定的长度划分成若干子格. 所有原子根据其位置分配到不同的子格中, 同时产生一个原子的目录列表. 当计算作用力和能量时, 在一个子格中的原子 i 和周围相邻的所有子格中的原子相互作用. 对于立方体盒子, 它被分割为 $M\times M\times M$ 个子格. 每个子格的长度 L/M 不小于 r_c/k ($k \in N$). k=1 就是经典的元胞列表法, k=2, 3,…是改进算法. k 越大, 需要的存储空间越大. 然而, 此算法并不适用于平行六面体(图 4). 事实上, 盒子的边长是用来判断截断半径的, 而垂线是划分网格的, 对于立方体, $L=d_a$, 这其中隐藏了真正的变量: 垂线. 在平行六面体中, 由于垂线小于任何一边, 因此相邻子格才能包含中心子格每个原子的所有邻位原子. 因此, 刚刚提到的保守截断半径也就改变了网格的划分, 即可变网格. 类似的, 平行六面体分成

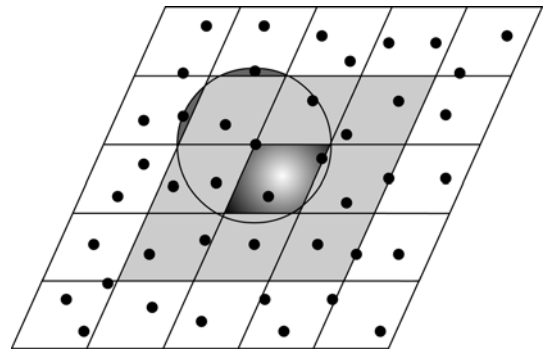

图 4 原始元胞列表法在平行六面体中的缺陷

$M_a\times M_b\times M_c$ 个相同的、非立方体的子格, 然而, M_a, M_b, M_c 和 M 有如下区别:

$$M = \left[\frac{L}{\Delta}\right], \quad M_\alpha = \left[\frac{d_\alpha}{\Delta}\right] (\alpha = a,b,c),$$

其中$\Delta=r_c/k$, 方括号表示取不大于的整数. 因此在仿真过程中, 新算法允许平行六面体盒子大小和方向的任意改变. 由于外载荷的作用, 系统会产生振动, 因此 $d_\alpha(\alpha = a,b,c)$ 每一步都需要重新计算. 这就意味着有可能需要重新划分网格, 即不能使用固定网格.

举例说明, 如果系统被压缩, 固定网格的内切球半径就有可能小于 r_c/k, 使得一些在截断半径范围里的相邻原子不能参与作用力和能量的计算.

(v) 元胞列表法的快速排序. 在立方体系统中, 依靠k的值, 计算耗时出现两种不同的模式. 当k较小时, 总是依次在子格中查找原子. 然而, 当$k\geq 5$ 时, 会产生一些变化. 首先, 依照Movable hash算法, 每个子格仅仅包含 1 个或 0 个原子[7]. 由于 0 原子的子格占的比例很大, 原子循环的速度已经快于子格循环. 其次, 增加了存储空间和构建邻位列表的时间. 事实上, 真正的瓶颈在于随着k的增大, 网格变得很敏感, 有可能每步MD都需要重新划分网格, 这将耗费很多的时间. 由于元胞列表法在k较大时存在一些缺点, 因此, 一种高效的元胞算法(k=1)则变得非常可贵. 最近, 一种快速排序算法[26]显示出了诸多优点: k=1, 即需要最小的存储空间, 并且可以高效查找邻位原子. 其主要原理如下: 首先, 按照最原始的元胞列表法, 将系统分割成子格, 此时的边长等于或者稍大于r_c. 所有的原子根据它的位置被分配到各个子格中. 当计算作用力和能量时, 首先通过原子位置矢量与归一化矢量$\boldsymbol{p}$的点乘将位于中心子格的原子和相邻的

原子排序，然后通过截断半径确定相邻原子. 因为快速排序和$N\log N$成正比，可以减少很多计算时间. 这种方法的核心概念是点乘. 为了完成点乘，需要使用归一化的矢量$\boldsymbol{p}$来描述某子格及其邻位子格的中心，即

$$\boldsymbol{r}_i \cdot \frac{\boldsymbol{p}}{|\boldsymbol{p}|} - \boldsymbol{r}_c < \boldsymbol{r}_j \cdot \frac{\boldsymbol{p}}{|\boldsymbol{p}|}, \tag{4}$$

其中$\boldsymbol{r}_i$和$\boldsymbol{r}_j$分别是邻位子格和中心子格中原子的位置. 事实上，只能通过$\boldsymbol{p}=h\boldsymbol{p}_1$得到矢量$\boldsymbol{p}$，$\boldsymbol{p}_1$是矢量$\boldsymbol{p}$在晶格坐标系下的坐标. 例如，$\boldsymbol{p}_1=(1,0,0)$(图 5). 因此，(4)式可写成如下形式:

$$\boldsymbol{s}_i^{\mathrm{T}} G\boldsymbol{p}_1 - r_c\sqrt{\boldsymbol{p}_1^{\mathrm{T}} G\boldsymbol{p}_1} \leqslant \boldsymbol{s}_j^{\mathrm{T}} G\boldsymbol{p}_1. \tag{5}$$

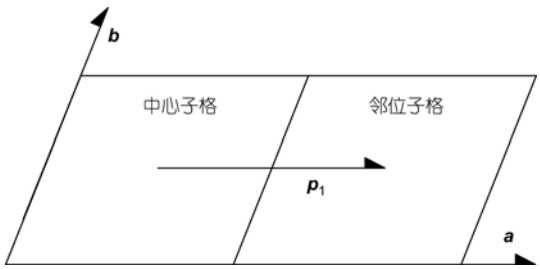

图 5　中心子格和邻位子格之间的矢量 $\boldsymbol{p}_1$

此外，在快速排序算法中，由于点乘，位于边界的原子不能使用边界条件. 如果仿真系统分为$M_a\times M_b\times M_c$个子格，边界子格的数量是 $2[M_aM_b+M_a(M_c-2)+(M_b-2)(M_c-2)]$. 如果假设$M_a=M_b=M_c=m$，边界子格数大约为 $6/m$，例如，m=50，其百分比则大于 10%. 因此，计算周期边界条件下点乘的效率将对 MD 模拟的速度产生很大的影响. 这里，我们用两个数组来说明这个问题. 第一个数组用来记录边界子格，第二个数组(FlagCell)是标志数组. 如果中心子格是左边界子格，它的一个邻位是右边界子格，标志数组应标志为−1，对于相反的情况标志为 1. 这些都是在初始化阶段进行的工作. 当对(5)式的点乘进行排序时，可以首先检查当前的子格是不是边界子格. 如果是，(5)式中的排序算法应该加入一项针对周期性边界条件的$\boldsymbol{p}_1^{\mathrm{T}}G$FlagCell. 当重新划分网格时，都需要重新分配标志数组 FlagCell.

(vi) 联合算法. 根据上述算法，现在给出一种联合算法，通过 Verlet、元胞列表法、标度矩阵 H 和快速排序算法来加速 MD 模拟.

当系统接近平衡态时，Verlet列表算法具有一个有效的邻位列表和较低的列表更新频率. 然而，构建列表时需要一个与N^2 成正比的计算花销[10,24]. 这种花销使其无法应用于快速移动的系统，此时原子的邻位更新频率较高. 相反，元胞列表法是一个与N成正比的[10,24]邻位列表更新算法，因此适用于大型和非平衡态的系统. 可以结合两种算法的优点，形成一种联合算法[27]. 这种算法具有较快的计算速度，同时能应用到NP系综.

通过改进的邻位算法，可以构建邻位列表，具体步骤如下:

(1) 使用元胞列表法，按 r_s 划分模拟系统.

(2) 通过晶格坐标系下的位置把每个原子分配到相应的子格中.

(3) 使用快速排序法构建 Verlet 列表.

事实上，不一定要严格按照上述步骤建立列表. 在每一步分子动力学中，可以首先检查 Verlet 列表是否需要更新. 如果需要，接着考虑是否需要更新网格的划分. 如果需要重新划分网格，重复步骤(1)到(3). 否则只需要进行步骤(2)和(3). 可以看出，新的邻位列表算法是与 N 成正比的算法. 同时，新算法也可以应用到固定的 h 系统，只需要多花费一点计算时间.

2　仿真结果

在本节中，我们用Lenneard-Jones势函数进行分子动力学模拟来说明上述算法的正确性. 为了更好地描述问题，用NPT和NVT系综，Nose-Poincare控温[28]，Metric-tensor控压[20]，Generalized leapfrog scheme积分算法[29]. 使用真实物理单位. 参数如下所示: σ= 3.405 Å，ε= 119.8 kJ/mol，质量m=39.948，截断半径$r_c=2.5\sigma$，Verlet半径$r_s=1.1r_c$，时间步长Δt=1 fs. 一系列不同原子数和密度的仿真用来证明新算法的各种特性.

第一个仿真对比了在相同的截断半径下新算法和直接计算方法的结果. 采用 NPT 系统，剪切应力$\tau_{xy}=5.0\times10^7$ Pa，温度 30 K. 系统包含 500 个原子，密度为 $1.0\sigma^3$. 表 1 给出了计算结果. 注意，当仿真结束时，仿真盒子从一个 L = 26.75 Å 的立方体变成一个$H=(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})$的平行六面体:

$$H = \begin{pmatrix} 26.85 & -0.824 & -0.019 \\ 0 & 26.85 & -0.015 \\ 0 & 0 & 26.63 \end{pmatrix} \text{Å}.$$

表 1 NPT 系综中联合算法和直接计算法比较

平均能量 a)	直接计算法	联合算法
总能量	−0.000000006553354	−0.000000006553148
势能	−0.072508910167620	−0.072508910169594
动能	0.003869885755219	0.003869885757195

a) 平均能量, 总能量除以总原子数目和总分子动力学仿真步数

忽略由两种算法的不同原子序列带来的截断误差, 可以证明新算法的正确性.

第二个仿真比较了邻位列表构建的时间. 这里的主要参数是 k, 从 1 到 4 变化. 同样, 仿真中使用不同数目的原子. 图 6 给出了仿真结果, 表明联合法构建邻位列表的时间与总原子数 N 成比例. 注意, 元胞邻位法和快速排序法结合的算法比纯粹的元胞法效率高.

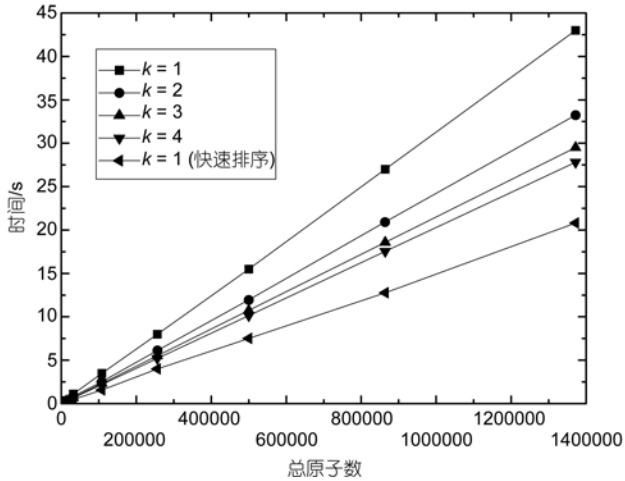

图 6 不同 k 下联合法建立邻位列表所消耗的时间

最后一个仿真着重于分析不同算法的效率, 这里收集了每个算法用于所有分子动力学模拟的 CPU 时间. 这些仿真采用具有不同原子数的 NVT 和 NPT 系综. 对于 NVT 系综, 使用下列参数: 密度为 $0.5\sigma^3$, 温度 300 K. 对于 NPT 系综, 剪切应力为 5.0×10^7 Pa, 温度 30 K, 密度 $1.0\sigma^3$. 从图 7(a)中可以看出, 新算法可以很好地处理 NVT 系统, 只是在效率上略低于原始算法. 这种效率上的降低是由于对于固定 h 进行了一些无用计算. 对于 NPT 系综(图 7(b)), 原来的算法已经不能使用. 而新算法却可以很好地提高计算效率. 此外, 联合法比单一算法在效率上提高了 2~20 倍.

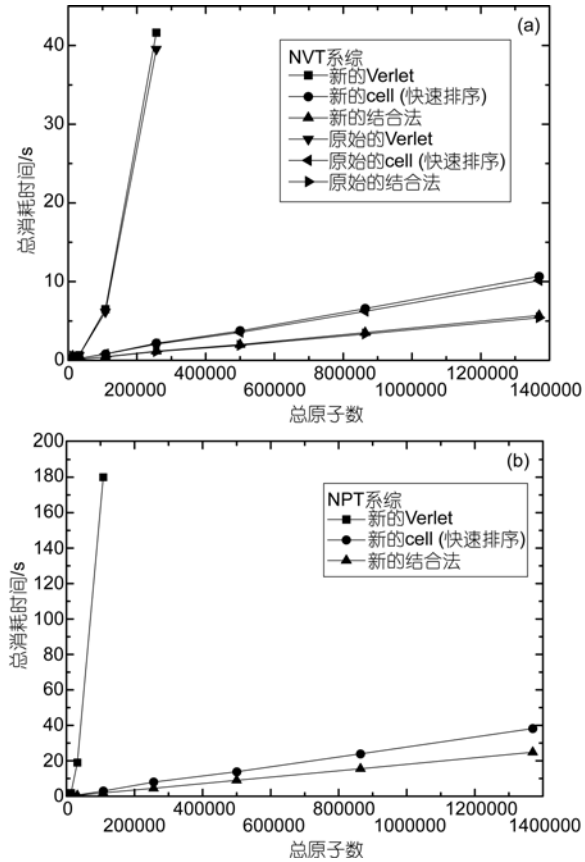

图 7 分子动力学消耗的时间

(a) NVT 系综; (b) NPT 系综

3 结论

传统算法构建邻位列表是基于立方体系统的, 本文将邻位算法扩展到了平行六面体的情况. 联合法是一个与原子数 N 成正比的算法, 许多分子动力学模拟证明了算法的正确性. 此外, 联合法比单一算法在效率上提高了 2~20 倍.

致谢 感谢甄玉宝博士对本文提出了很多宝贵的意见!

参考文献

1 Verlet L. Computer "experiments" on classical fluids. I. Thermodynamical properties of lennard-jones molecules. Phys Rev, 1967, 159(1): 98—103[DOI]

2 Quentrec B, Brot C. New method for searching for neighbors in molecular dynamics computations. J Comput Phys, 1973, 13(3): 430—432[DOI]
3 Boris J. A vectorized near neighbors algorithm of order n using a monotonic logical grid. J Comput Phys, 1986, 66(1): 1—20[DOI]
4 Morales J, Rull L, Toxvaerd S. Efficiency test of the traditonal MD and the link-cell methods. Comput Phys Commun, 1989, 56(2): 129—134[DOI]
5 Chialvo A A, Debenedetti P G. On the performance of an automated verlet neighbor list algorithm for large systems on a vector processor. Comput Phys Commun, 1991, 64(1): 15—18[DOI]
6 Mattson W, Rice B M. Near-neighbor calculations using a modified cell-linked list method. Comput Phys Commun, 1999, 119(2-3): 135—148[DOI]
7 Fang Q, Wang R, Liu C. Movable hash algorithm for search of the neighbor atoms in molecular dynamics simulation. Comp Mater Sci, 2002, 24(4): 453—456[DOI]
8 Petrella R J, Andricioaei I, Brooks B R, et al. An improved method for nonbonded list generation: Rapid determination of near-neighbor pairs. J Comput Chem, 2003, 24(2): 222—231[DOI]
9 Matin M, Daivis P, Todd B. Cell neighbor list method for planar elongational flow: Rheology of a diatomic fluid. Comput Phys Commun, 2003, 151(1): 35—46[DOI]
10 Yao Z, Wang J, Liu G, et al. Improved neighbor list algorithm in molecular simulations using cell decomposition and data sorting method. Comput Phys Commun, 2004, 161(1-2): 27—35[DOI]
11 张宇军, 董光能, 毛军红, 等. 含氢类金刚石膜沉积过程的分子动力学模拟. 科学通报, 2007, 52(23): 2813—2817
12 陈丽群, 王崇愚, 于涛. bcc Fe 中<100>刃型位错上扭折的分子动力学模拟. 科学通报, 2007, 52(6): 720—724
13 李水清, 姚强, 陈冰, 等. 回转筒颗粒表面流的分子动力学模拟和连续理论. 科学通报, 2006, 51(19): 2319—2326
14 Zhao Q, Zhang F, Zhou H. Collision dynamics of He@C60+He@C60 at low energies. Sci China Ser G-Phys Mech Astron, 2008, 51(7): 765—772[DOI]
15 Born M, Huang K. Dynamical Theory of Crystal Lattices. Oxford: Clarendon Press, 1954
16 Parrinello M, Rahman A. Strain fluctuations and elastic constants. J Chem Phys, 1982, 76(5): 2662—2666[DOI]
17 Gusev A A, Zehnder M M, Suter U W. Fluctuation formula for elastic constants. Phys Rev B, 1996, 54(1): 1—4[DOI]
18 Cui Z, Sun Y, Li J, et al. Combination method for the calculation of elastic constants. Phys Rev B, 2007, 75(21): 214101[DOI]
19 Zhang S, Chen N. Molecular dynamics simulations for high-pressure induced b1-b2 transition in nacl by mobius pair potentials. Model Simul Mater Sci Eng, 2003, 11(3): 331—338[DOI]
20 Souza I, Martins J. Metric tensor as the dynamical variable for variablecell-shape molecular dynamics. Phys Rev B, 1997, 55(14): 8733—8742[DOI]
21 Parrinello M, Rahman A. Crystal structure and pair potentials: A molecular-dynamics study. Phys Rev Lett, 1980, 45(14): 1196—1199[DOI]
22 Ciccotti J R G. Molecular dynamics simulation of rigid molecules. Comput Phys Rep, 1986, 4: 345—392
23 Nose S, Klein M. Constant pressure molecular dynamics for molecular system. Mol Phys, 1983, 50(5): 1055—1076[DOI]
24 Allen M, Tildesley D. Computer Simulation of Liquids. Oxford: Oxford University Press, 1987
25 Bekker H. Unification of box shapes in molecular simulations. J Comput Chem, 1997, 18(15): 1930—1942[DOI]
26 Gonnet P. A simple algorithm to accelerate the computation of nonbonded interactions in cell-based molecular dynamics simulations. J Comput Chem, 2007, 28(2): 570—573[DOI]
27 Puetz M, Kolb A. Optimization techniques for parallel molecular dynamics using domain decomposition. Comput Phys Commun, 1998, 113(2-3): 145—167[DOI]
28 Bond S D, Leimkuhler B J, Laird B B. The nose-poincare method for constant temperature molecular dynamics. J Comput Phys, 1999, 151(1): 114—134[DOI]
29 Sun G. Construction of high order symplectic runge-kutta methods. J Comput Math, 1993, 11(3): 250—260