

引言

知之者不如好知者，好之者不如乐之者

生而知之者，上也；学而知之者，次也；困而学之，又其次也；困而不学，民斯为下矣。

I hear and I forget. I see and I remember. I do and I understand. 闻而忘之，见而知之，行而明之

谢辞

没有下面这些人, 这本手册肯定要来的晚一些.

- 录入公式

阮洋: 第四章 1–2节, 第六章

杨宇: 第八章

- 翻译

严立京: 第五章 5.1–5.2, 第三章 3.13–3.18

杨旭云: 第四章 4.6–4.10

姚闯: 第三章 3.4.1–3.4.7

那些没有出现名字的人, 我们仍然也感谢他们.

因了所有这些人, 这个世界, 至少好像变得可爱多一点.

译宣言

2014-11-03 22:15:04

考虑再三, 我决定征集人手将GROMACS的手册翻译成中文.

总看到有人说翻译这些科技资料没有意义, 要直接去看英文, 英文的更准确. 这些说辞当然有它的道理, 对初学者却没有太多帮助. 一个GROMACS的初学者, 他需要的是快速入门. 等入了门, 再去看英文的手册, 英文的资料也能更有信心, 不再那么害怕了. 而能快速入门的途径之一, 就是读用他的母语写成的手册或资料. 可以想象一下, 如果不对化学, 生物学中的那些名称进行翻译, 而是直接使用其英文或其音译, 什么奥克斯真, 哈埃兆真, 毛赖克犹, 埃斯得一股脑全上, 对大多数人来说, 无疑如同天书, 和欧阳锋看到的《九阴真经》最后一章差不多. 我怕是很多学生直接就被吓跑了.

当然, 我不否认有些人天生异禀, 直接去读英文资料可能更好. 可惜, 我们大部分人并不是那样, 也没有那个造化. 颜之推曾说: 上智不教而成, 下愚虽教无益, 中庸之人, 不教不知也. 我们大部分都是中等之人吧, 可教可学. 所以有些母语的资料总归还是有帮助的.

我将GROMACS 5.0.2版本的手册分成了24份, 除却无须翻译的目录, 参考文献, 索引, 要翻译的共有21份, 每份15页左右. 清单如下:

1. GROMACS Manual 5.0.2 第0章 目录.pdf 无须翻译
2. GROMACS_Manual_5.0.2_第1章_共6页 已完成
3. GROMACS_Manual_5.0.2_第2章_共3页 已完成
4. GROMACS_Manual_5.0.2_第3章_3.1-3.3_共5页 已完成
5. GROMACS_Manual_5.0.2_第3章_3.4.1-3.4.7_共17页 已完成
6. GROMACS_Manual_5.0.2_第3章_3.4.8-3.4.11_共14页 已完成
7. GROMACS_Manual_5.0.2_第3章_3.5-3.12_共12页 已完成
8. GROMACS_Manual_5.0.2_第3章_3.13-3.18_共12页 已完成
9. GROMACS_Manual_5.0.2_第4章_4.1-4.2_共20页 已完成
10. GROMACS_Manual_5.0.2_第4章_4.3-4.5_共16页 已完成
11. GROMACS_Manual_5.0.2_第4章_4.6-4.10_共17页 已完成
12. GROMACS_Manual_5.0.2_第5章_5.1-5.6_共18页 已完成
13. GROMACS_Manual_5.0.2_第5章_5.7-5.8_共17页 已完成

14. GROMACS_Manual_5.0.2_第6章_6.1-6.5_共17页 已完成
15. GROMACS_Manual_5.0.2_第6章_6.6-6.14_共21页 已完成
16. GROMACS_Manual_5.0.2_第7章_7.1-7.3.15_共20页 已完成
17. GROMACS_Manual_5.0.2_第7章_7.3.16-7.3.29_共22页 已完成
18. GROMACS_Manual_5.0.2_第8章_8.1-8.6_共10页 已完成
19. GROMACS_Manual_5.0.2_第8章_8.7-8.14_共12页 已完成
20. GROMACS_Manual_5.0.2_附录_A_共9页 已完成
21. GROMACS_Manual_5.0.2_附录_B_共9页 已完成
22. GROMACS_Manual_5.0.2_附录_C_共5页 已完成
23. GROMACS Manual 5.0.2_附录_参考文献.pdf 无须翻译
24. GROMACS Manual 5.0.2_附录_索引.pdf 无须翻译

愿意翻译的同学, 可以加入群GROMACS中文组, 群号132266540, 共同商量. 不愿加入群的同学也可以到下载一份, 翻译完成后再发到我邮箱 Jerkwin@Gmail.com.

等手册全部翻译完成, 我再进行统稿, 制作成pdf版和在线版, 放到网上供人下载查阅.

翻译要求:

1. 翻译时请使用微软的Word程序, 将翻译好的文字保存为Word 2003的doc格式, 文件名与pdf文件名一致
2. 原手册中的图片无须添加到翻译稿中
3. 手册中的公式请用Word自带的公式编辑器或MathType软件或LaTeX格式书写

我为网络, 网络为我. 网络力量, 有我一份.

译后记

2015-05-28 21:22:05

短短续续地, 终于译完了GROMACS手册.

这是我第一次翻译这么长的文档, 希望这不是最后一次.

现在回头看来, 自从2014年11月开始动手翻译以来, 时间已经过去了半年多.

白驹过隙, 马齿徒增.

这个翻译工作权且作为我生命中这段时间的一个注记吧.

翻译这么长的文档, 也着实不易.

不记得有几次, 我直直觉得自己无法再译下去了, 也是不可能完成的了.

我甚至都已为自己想好了理由, 便对自己说,

不必再做了, 你做这些努力, 你翻译这些, 为的哪般呢?

聪慧的, 自不必看; 上进的, 自去看英文; 愚笨的, 自不会去看.

那, 你告诉我, 还有谁会来读你翻译的呢?

我无言以答, 却隐隐不甘心, 便对自己说,

可能会有的吧, 那些鲁钝的呢?

是的, 可能会有的吧.

我终不能以必知的无否定了未知的有.

正如面对真空, 有人见了无, 也有人见了有.

那, 我还可以说这是因了梦么?

这样说, 会有人信么?

纵然无法欺人, 也可当作一个自欺的理由吧.

多少次, 我曾经以为丢掉了他.

其实他一直在那里, 伴随着我.

我不知道他会停留到什么时候.

虽则他已经着了厚厚一层俗尘,

却从未从这个世界上消失,

不时熠熠而现, 让我着迷不已.

他的不时隐隐

是在等待长风

第一章 简介

1.1 计算化学与分子建模

GROMACS是一个用于分子动力学模拟和能量最小化的计算引擎. 分子动力学模拟和能量最小化是计算化学和分子建模领域众多技术中的两种. **计算化学** 是计算技术在化学中应用, 涉及范围从分子的量子力学到复杂大分子聚集体的动力学. **分子建模** 是用实际的原子模型描述复杂化学体系的一般方法, 其目的是以原子尺度的详细知识为基础, 理解和预测物质的宏观性质. 通常, 分子建模被用于设计新的材料, 因为在这过程中需要对实际体系的物理性质进行准确的预测.

宏观的物理性质可以分为 (a) 静态平衡性质, 如一种酶与其一种抑制剂的结合常数, 系统的平均势能, 或液体的径向分布函数 ; (b) 动态或非平衡性质, 如液体的粘度, 膜中的扩散过程, 相变的动力学, 反应动力学, 或在晶体中缺陷的动力学. 计算方法的选择取决于要回答的问题以及该方法在当前技术发展水平下得到可靠结果的可行性. 理想情况下, (相对论)含时薛定谔方程能以很高的精度描述分子体系的性质, 但这种 **从头算** 的方法水平只能处理几个原子的平衡态计算. 因此, 采用近似是必要的. 体系的复杂度越高, 待研究过程的时间跨度越长, 计算所需的近似度就越高. 到某个阶段(远早于人们所希望的), 从头算方法必须借助于 **经验** 的参数化模型, 或被其完全取代. 当体系的复杂度导致基于原子间相互作用的物理原理的模拟仍然失败时, 分子建模就完全建立在对已知结构和化学数据进行相似性分析的基础之上. QSAR方法(定量结构-活性关系)和许多基于同源蛋白的蛋白质结构预测就属于这类方法.

宏观性质始终是分子体系某种代表性统计系综(平衡或非平衡)的系综平均值. 对分子建模, 这有两个重要的推论:

- 仅有单一结构的信息是远远不够的, 即使这个结构是全局能量最小点. 为计算宏观性质, 必须产生指定温度下体系的代表性系综. 但对于计算那些与自由能相关的热力学平衡性质, 如相平衡, 结合常数, 溶解度, 分子构象的相对稳定性等等, 这仍然不够. 自由能和热力学势的计算需要对分子模拟技术进行特殊地推广.
- 原则上, 分子模拟提供了体系结构和运动的原子尺度的细节, 然而, 这些细节常常与那些感兴趣的宏观性质无关. 这为简化对相互作用的描述并对无关细节进行统计平均提供了可能. 统计力学为这些简化提供了理论框架, 并可以使用一系列不同等级的方法, 涉及到的方法从将一组原子视为一个单元, 以少量的集合坐标描述运动, 对溶剂分子利用平均力势和随机动力学进行统计平均[7], 一直到 **介观动力学**. 介观动力学直接描述密度而不是原子, 并以描述流(作为对热力学梯度的响应)取代描述速度或加速度(作为对力的响应)[8].

有两种方法可用于产生一个具有代表性的平衡系综: (a) 蒙特卡洛(Monte Carlo, MC)模拟 和(b) 分子动力学(Molecular Dynamics, MD)模拟. 对于非平衡系综的产生和动态事件的分析, 只有第二种方法是适用的. 与MD相比, 蒙特卡洛模拟更简单(不需要计算力), 然而, 在一定的计算时间内, MC并不能得到比MD明显更好的统计结果. 因此, MD是更通用的技术方法. 如果初始构型离平衡态非常远, 体系的受力可能过大, 以致MD模拟失败. 在这种情况下, 需要先对体系进行稳健的 **能量最小化**. 另一个进行能量最小化的原因是去除体系的所有动能: 如果要对动力学模拟过程中几个“快照”进行比较, 能量最小化能够减少结构和势能中的热噪声, 因此比较结果可能更好.

1.2 分子动力学模拟

对含 N 个相互作用原子的系统, MD模拟求解其牛顿运动方程:

$$m_i \frac{\partial^2 \mathbf{r}_i}{\partial t^2} = \mathbf{F}_i, i = 1, \dots, N \quad (1.1)$$

原子所受的力是一个势能函数导数的负值:

$$\mathbf{F}_i = -\frac{\partial V}{\partial \mathbf{r}_i} \quad (1.2)$$

模拟时, 会以很小的时间步长同时求解这些方程, 使系统演化一段时间, 同时小心地使温度和压力维持在设定值, 并以一定的时间间隔将原子坐标写入输出文件中.

表1.1 分子及含氢键液体中的典型振动频率(波数). 300 K时, $kT/h = 200 \text{ cm}^{-1}$

键的类型	振动类型	波数(cm^{-1})
C-H, O-H, N-H	伸缩	3000-3500
C=C, C=O	伸缩	1700-2000
HOH	弯曲	1600
C-C	伸缩	1400-1600
H ₂ CX	剪式, 摆摆	1000-1500
CCC	弯曲	800-1000
O-H...O	摆动	400-700
O-H...O	伸缩	50-200

随着时间变化的坐标代表了体系的一个轨迹. 经过初始的一些变化之后, 体系通常会达到平衡态. 通过对平衡后的轨迹进行平均, 就可以从输出文件中提取出许多宏观性质.

此时, 了解MD模拟的局限性是有益的. 用户应该清楚这些局限性, 并始终通过对已知实验性质进行检查来评估模拟的准确度. 下面列出了MD所采用的近似.

模拟是经典的

使用牛顿运动方程自然意味着我们使用 **经典力学** 描述原子的运动. 对处于正常温度下的大多数原子来说, 这是合适的, 但也有例外: 氢原子质量非常小, 质子的运动有时具有显著的量子力学特性. 例如, 在沿氢键传递过程中, 质子可能通过 **隧穿效应** 越过势垒. 经典力学不可能正确地处理这样的过程! 低温下的液氦是经典力学失效的另一个例子. 或许我们不是很关心氦, 但共价键的高频振动却应该让我们担心! 当共振频率 ν 接近或超过 $k_B T/h$ 时, 经典谐振子与真正的量子谐振子的统计力学明显不同. 室温下当 $h\nu = k_B T$ 时, 波数 $\sigma = 1/\lambda = \nu/c$, 大约为 200 cm^{-1} . 因此, 所有高于, 比如说 100 cm^{-1} 的频率, 在经典模拟中的行为都可能不正确. 这意味着所有键和键角的振动实际上都是可疑的, 甚至氢键键合基团的运动, 比如平动或氢键的摆动都超出了经典的极限(见表1.1). 我们该怎么办呢?

除了进行真正的量子动力学模拟, 我们可以采取下面两种方法中的一种:

- (a) 如果进行MD模拟时对键采用了谐振子近似, 我们应该对总内能 $U = E_{kin} + E_{pot}$ 和比热 C_V (以及熵 S

自由能 A 或 G , 如果要计算它们) 进行校正. 对频率为 ν 的一维振子, 能量和热容的校正分别为[9]:

$$U^{QM} = U^{cl} + kT \left(\frac{1}{2}x - 1 + \frac{x}{e^x - 1} \right) \quad (1.3)$$

$$C_V^{QM} = C_V^{cl} + k \left(\frac{x^2 e^x}{(e^x - 1)^2} - 1 \right) \quad (1.4)$$

其中 $x = h\nu/kT$. 当高频量子振子处于基态时, 具有 $\frac{1}{2}h\nu$ 的零点能, 经典谐振子会吸收过多的能量(kT).

(b) 我们可以将键(和键角)视为运动方程的 **约束**. 其理由是, 与经典振子相比, 处于基态的量子振子与被约束的键更相似. 作这种选择有一个很实用的理由: 当最高频率被移除后, 可以在算法中使用更大的时间步长. 实际中, 相比把键当作谐振子, 当键被约束时的时间步长可以取为原先的4倍[10]. GROMACS对键和键角提供了这种方案. 灵活的键角约束是相当重要的, 它使得运动更加真实, 并能覆盖构型空间[11].

电子处于基态

在MD中, 我们使用 **保守力场**, 它只是原子位置的函数. 这意味电子的运动被忽略了: 我们假定电子能够瞬间调整自己的运动状态以适应原子位置发生的变化(玻恩-奥本海默近似), 并始终处于它们的基态. 这个假定几乎总是正确的, 但显然不能用于电子转移过程和电子激发态, 也不能正确处理化学反应. 当然, 暂时回避化学反应还有着其他原因.

力场是近似的

力场提供了力. 它们实际上不是模拟方法的一部分. 当需要或对体系有了更多了解时, 用户可以修改力场中的参数. 但是, 在一个特定的程序中所能使用的力的形式是有限制的. GROMACS中可用的力场将在第4章进行说明. 在目前的版本中, 力场是对势累加(pair-additive)的(除了长程库仑力), 所以不能包含极化, 也不包含可以微调的键相互作用. 由此还导致了下面列出的一些限制. 除此以外, 力场对水溶液中的生物相关的大分子是相当有用, 相当可靠的!

力场是对势累加的

这意味着, 所有的 **非键** 力来源于非键对势相互作用的加和. 非对势累加相互作用, 其中最重要的是通过原子极化产生的相互作用, 要通过 **有效对势** 进行描述. 这里仅考虑了平均的非对势累加的贡献. 这也意味着, 对相互作用(pair interactions)并不纯粹, 也就是说, 它们对孤立的原子对, 或是与参数化时基于的模型有明显差距的体系, 是无效的. 实际上, 有效对势在实际使用中并没有那么糟糕. 但忽略极化也意味着原子中的电子并不能提供正常的介电常数. 举例来说, 实际的液体烷烃具有略大于2的介电常数, 这会降低(部分)电荷之间的长程静电相互作用. 因此, 模拟将高估远程库仑相互作用. 幸运的是, 下一近似稍稍弥补了这一缺陷.

长程相互作用被截断

在本版本中, GROMACS总是对Lennard-Jones相互作用使用截断半径, 对库伦作用有时也使用截断半径. GROMACS使用的“最小映像约定”要求: 对一个对相互作用(pair interaction)中的每个粒子, 只考虑其在周期性边界条件下的一个映像, 所以截断半径不能超过盒子大小的一半. 对大的体系来说, 这仍然相当大, 但仅仅对于含有带

电粒子的体系会有麻烦。这时可能会发生许多糟糕的事，如电荷会在截断边界处累积，或计算的能量完全错误！对于这样的系统，你应该考虑包含了那些长程静电相互作用的算法，如粒子网格Ewald (particle mesh Ewald, PME) [12, 13].

边界条件是不自然的

由于体系尺寸很小(即便含10000个粒子的体系也还很小)，粒子组成的团簇与其环境(真空)之间具有许多不需要的边界。如果想模拟块状体系，我们必须避免这种情况。对此，我们采用周期性边界条件来避免实际的相边界。由于液体不是晶体，一些不自然情况仍然存在。我们最后提及这一方面是因为它的影响最小。对大体系其误差很小，但对于具有很强内部空间相关性的小体系，周期性的边界可能会增强内部的相关性。在这种情况下，要多加小心并注意测试系统尺寸的影响。当使用格点加和方法计算远程静电相互作用时，这一点尤其重要，因为我们知道这样有时会引入额外的有序。

1.3 能量最小化和搜索方法

在1.1节已经提到，许多情况下需要进行能量最小化。GROMACS提供了许多方法进行局部能量最小化，我们将在3.10节进行详细讨论。

(大)分子体系的势能函数是多维空间中一个超曲面，具有非常复杂的形貌。它有一个最低点，全局极小点 和大量的局部极小点。在这些点上，势能函数对所有坐标的导数为零，对所有坐标的二阶导数都是非负的。二阶导数组成的矩阵，即所谓的 Hessian矩阵，具有非负的本征值；只有对应于孤立分子的平动和转动的集约坐标具有零本征值。在局部极小值之间有 鞍点，其Hessian 矩阵仅有一个负的本征值。通过这些点，体系可以从一个局部极小值到转移到另一个局部极小值。

拥有所有局部极小点，包括全局最小点和所有鞍点的信息，我们就可以描述有关的结构和构象，它们的自由能，以及结构转化的动力学。不幸的是，构型空间的维数和局部极小点的数目非常多，我们不可能进行足够的采样来获得全部的信息。特别是，没有一种最小化方法能够保证在实际可承受的时间范围内找到全局极小点。存在一些不很实用的方法，速度有快有慢[14]。然而，给定一个初始构型，可以找到它的 最近极小点。“最近”在这里并不总是意味着结构意义上的“最近”(即二者坐标差距的平方和最小)，而是指可以通过系统地沿着局部梯度最陡方向往下移动能够到达的极小点。很不幸，GROMACS所能为你做的只是找到这种最近极小！如果你想找到其他极小点，并希望在此过程中找到全局极小点，最好尝试下温度耦合的MD：先在高温下模拟体系一段时间，然后慢慢下降到所需的温度，并不断重复这一过程！如果存在熔点或玻璃化转变温度，可以先在稍低于该温度的条件下模拟一段时间，再使用某些聪明的方案慢慢降温，这样的过程被称为 模拟退火。由于这一过程不需要与物理真实过程相对应，你尽可以发挥你的想象力以加速这一进程。一个经常可行的小技巧是增加氢原子的质量(可增加到10左右)：虽然这将减缓氢原子的快速运动，却几乎不会影响到体系的慢运动，同时能使时间步长增加为原来的3倍或4倍。在搜索过程中你也可以修改势能函数，例如，通过移除势垒(去除二面角函数或以 软核 势取代排斥势[15])，但当慢慢恢复势能函数的时候要小心。允许结构剧烈变化的最好搜索方法是在四维空间中进行漫游[16]，但这超出了GROMACS的标准功能，需要一些额外的编程工作。

有三类可能的能量最小化方法：

- 只需要计算函数值的方法。如单纯形法及其变种。前进的每一步都是在前面计算结果的基础上做出的。如果可以利用导数信息，这类方法要差于那些使用导数信息的方法。

- 使用导数信息的方法. 由于在MD程序中, 势能相对于所有的坐标的偏导数(等于力的负值)都是已知的, 由MD程序稍加修改得到的这类方法非常合适.
- 还使用二阶导数信息的方法. 这些方法在接近极小点时的收敛性非常好: 一个二次势能函数的最小化只需要一步! 问题是, 对于 N 粒子的体系, 必须对 $3N \times 3N$ 的矩阵进行计算, 存储, 求逆. 除非用额外的编码获得二阶导数, 对大多数系统, 这类方法都不可行. 此外, 还有一些实时建立Hessian矩阵的中间方法, 但它们也需要大量的额外存储空间. 因此, GROMACS没有使用这类方法.

GROMACS使用的 **最速下降法** 属于第二类方法. 它只是简单地沿负梯度方向(因此, 在力的方向)前进, 而不考虑前面的已有的任何步骤. 搜索过程中, 步长可以进行调整, 以便使搜索快速进行, 但搜索永远沿着能量减小的方向. 这是一个简单稳健但有些愚蠢的方法: 它的收敛速度可能相当慢, 特别是处于局部极小值附近时! 收敛更快的 **共轭梯度法** (见[17])使用了前面步骤的梯度信息. 一般情况下, 最陡下降法能够很快地接近最近极小点, 而共轭梯度法能够非常接近局部极小点, 但当远离极小点时效果较差. GROMACS也支持L-BFGS最小化, 这是几乎与共轭梯度法相当的方法, 但在某些情况下, 其收敛速度更快.

第二章 定义与单位

2.1 符号标记

对数学符号,在整个文档中都将遵循以下约定:

含义	标注	示例
矢量	粗斜体	\mathbf{r}_i
矢量长度	斜体	r_i

我们用 小写 下标 i, j, k, l 标识粒子: \mathbf{r}_i 为粒子 i 的 位置矢量, 并且使用下面的记法:

$$\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i \quad (2.1)$$

$$r_{ij} = |\mathbf{r}_{ij}| \quad (2.2)$$

粒子 i 受到的力记为 \mathbf{F}_i , 且

$$\mathbf{F}_{ij} = \text{粒子 } j \text{ 施加于粒子 } i \text{ 的力} \quad (2.3)$$

请注意, 自从GROMACS 2.0版本我们已将标记改为 $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$. 因为这是常用的约定. 如果你遇到错误, 请告知我们.

2.2 MD单位

GROMACS使用一系列自治的单位,以便使相关分子性质的计算值接近单位1. 我们称其为 MD单位. 这个系统的基本单位是 nm, ps, K, 电子电荷(e)和原子质量单位(u), 参看表2.1.

与这些单位一致的一套导出单位, 列于表2.2.

静电转换因子 $f = \frac{1}{4\pi\epsilon_0} = 138.935485(9) \text{ kJ mol}^{-1} \text{ nm e}^{-2}$, 它联结了力学量与电学量, 即

$$V = f \frac{q^2}{r} \text{ 或 } F = f \frac{q^2}{r^2} \quad (2.4)$$

表2.1 GROMACS使用的基本单位. 括号中的数字表示精确度.

物理量	符号	单位
长度	r	纳米(nm) = 10^{-9} m
质量	m	原子质量单位(u) = $1.660\ 540\ 2(10) \times 10^{-17} \text{ kg}$ ^{12}C 原子质量的1/12
时间	t	皮秒(ps) = 10^{-12} s
电荷	q	电子电量(e) = $1.602\ 177\ 33(49) \times 10^{-19} \text{ C}$
温度	T	K

表2.2 导出单位

物理量	符号	单位
能量	E, V	kJ mol^{-1}
力	F	$\text{kJ mol}^{-1} \text{nm}^{-1}$
压力	p	$\text{kJ mol}^{-1} \text{nm}^{-3} = 10^{30} / N_{AV} \text{ Pa}$ $1.66054 \times 10^6 \text{ Pa} = 16.6054 \text{ bar}$
速度	v	$\text{nm ps}^{-1} = 1000 \text{ m s}^{-1}$
偶极矩	μ	e nm
电势	Φ	$\text{kJ mol}^{-1} \text{e}^{-1} = 0.010364272(3) \text{ Volt}$
电场强度	E	$\text{kJ mol}^{-1} \text{nm}^{-1} \text{e}^{-1} = 1.0364272(3) \times 10^7 \text{ V/m}$

电势 Φ 和电场强度 E 是计算能量和力的中间量。它们不会出现在GROMACS中。如果在计算中使用它们，可对方程及其相应的单位进行选择。在计算 Φ 和 E 时，我们强烈建议遵循惯例，使用包含转换因子 f 的下列方程：

$$\Phi(\mathbf{r}) = f \sum_j \frac{q_j}{|\mathbf{r} - \mathbf{r}_j|} \quad (2.5)$$

$$\mathbf{E}(\mathbf{r}) = f \sum_j q_j \frac{\mathbf{r} - \mathbf{r}_j}{|\mathbf{r} - \mathbf{r}_j|^3} \quad (2.6)$$

在这种定义下， $q\Phi$ 为能量， $q\mathbf{E}$ 为力。它们的单位由表2.2给出：单位电势大约10 meV。这样，距离一个电子电荷子1 nm的电势能等于 $f \approx 140$ 单位 $\approx 1.4 \text{ V}$ (精确值为1.439965 V)。

注意：这些单位彼此之间是自治的；改变其中的任何一个都可能使得单位系统不再自治，因此 强烈建议不要 修改。特别是，若使用 Å 代替 nm，单位时间将变为0.1 ps；若使用能量单位kcal mol $^{-1}$ 代替kJ mol $^{-1}$ ，单位时间将变为0.488882 ps，且单位温度变为4.184 K。但在这两种情况下，计算的所有电势能都是错误的，因为它们仍然是以kJ mol $^{-1}$ 为单位计算的，并假定nm为长度单位。尽管对电荷重新进行细心的标度可能会得到自治的单位，但很显然，所有这些容易引起混淆的做法都应该被严厉禁止。

常用的物理常数以MD单位表示，其取值可能不同(见表2.3)。所有的物理量都是以mol为单位计算的，而不是以单个分子为单位计算。玻尔兹曼常数 k 和理想气体常数 R 没有区别，它们的值都是0.008 314 51 kJ mol $^{-1}$ K $^{-1}$ 。

表 2.3 部分物理常数

符号	名称	数值
N_{AV}	Avogadro常数	$6.0221367(36) \times 10^{23} \text{ mol}^{-1}$
R	理想气体常数	$8.314510(70) \times 10^{-3} \text{ kJ mol}^{-1} \text{ K}^{-1}$
k_B	Boltzmann常数	
h	Planck常数	$0.39903132(24) \text{ kJ mol}^{-1} \text{ ps}$
\hbar	Dirac常数	$0.063507807(38) \text{ kJ mol}^{-1} \text{ ps}$
c	光速	$299792.458 \text{ nm ps}^{-1}$

表 2.4 LJ势的约化物理量

物理量	符号	与国际单位的关系
长度	r^*	$r\sigma^{-1}$
质量	m^*	mM^{-1}
时间	t^*	$t\sigma^{-1}\sqrt{\epsilon/M}$

温度	T^*	$k_B T \epsilon^{-1}$
能量	E^*	$E \epsilon^{-1}$
力	F^*	$F \sigma \epsilon^{-1}$
压强	P^*	$P \sigma^3 \epsilon^{-1}$
速度	v^*	$v \sqrt{M/\epsilon}$
密度	ρ^*	$N \sigma^3 V^{-1}$

2.3 约化单位

当模拟Lennard-Jones(LJ) 系统时, 使用约化单位(即设某类原子的 $\epsilon_{ii} = \sigma_{ii} = m_i = k_B = 1$)可能会方便, 也是可行的. 当以约化单位输入时, GROMACS将会以约化单位输出. 但 温度 是唯一的例外, 它是以0.008 314 51倍的约化单位表示的. 这是由于在代码中使用玻尔兹曼常数计算温度引起的. 因此, 不是温度 T , 而是 $k_B T$ 表示约化温度. GROMACS的温度 $T = 1$ 意味着约化温度为0.008...单位; 如果约化温度是1, GROMACS的温度应该是120.2717.

表2.4 给出对应于LJ势

$$V_{LJ} = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (2.7)$$

的约化物理量.

第三章 算法

3.1 简介

在本章中, 我们首先介绍GROMACS用到的一些基本概念: 周期性边界条件 (3.2节)和 组 (3.3节). MD算法将在3.4节介绍: 首先给出算法的整体形式, 然后在后面的小节中进行详细说明. (简单的)EM(Energy minimization, 能量最小化)算法在3.10节说明. 用于特殊目的动力学的一些其他算法在此后进行说明.

有几个问题是人们普遍关心的. 在所有情况下, 必须定义由分子组成的 体系. 而分子又是由定义了相互作用形式的粒子组成. 关于分子拓扑结构 和 力场 的详细描述以及力的计算将在第四章讨论. 在本章中, 我们只讨论算法的其他方面, 如配对列表(pair list) 的生成, 速度和位置的更新, 与外部温度和压力的耦合, 约束的保持. 对MD模拟数据的 分析 将在第八章讨论.

3.2 周期性边界条件

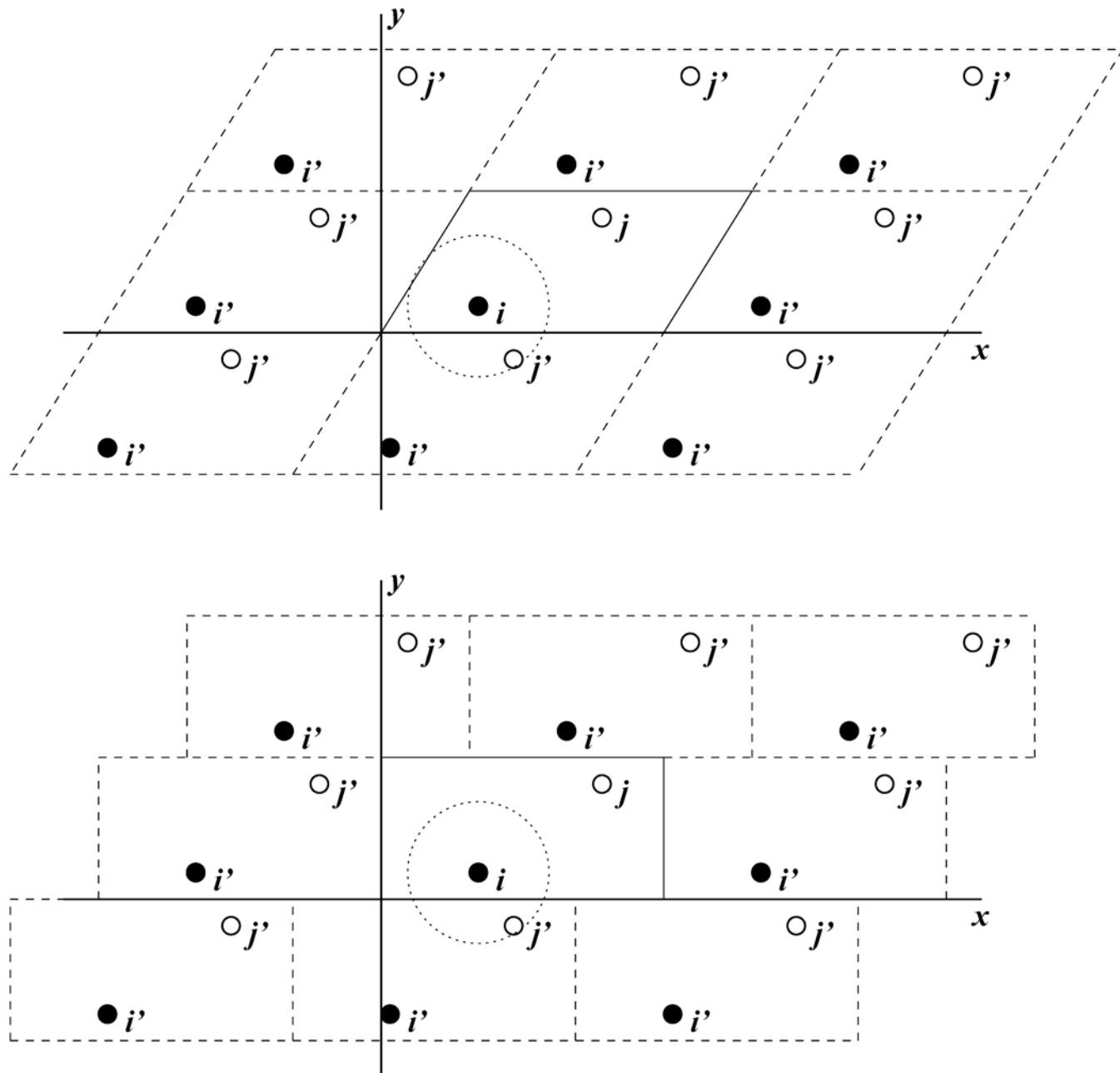


图3.1 二维的周期性边界条件

对有限的体系, 减小边缘效应的经典方法是应用 **周期性边界条件**. 将待模拟体系的原子放到一个空间填充(space-filling) 的盒子中, 周围是体系自身平移后的副本(图 3.1). 因此, 体系不存在边界, 由孤立团簇的多余边界导致的问题现在被由周期性条件引起的问题所取代. 如果体系具有晶体结构, 这样的边界条件符合预期(尽管运动自然被限制于周期性运动, 且其波长与盒子适应). 如果希望模拟非周期性体系, 如液体或溶液, 周期性就会带来误差. 此误差可以通过比较不同尺寸体系的模拟结果进行评估. 比起由不自然的真空边界引起的误差, 周期性边界引起的误差更小一些.

用于空间填充的元胞的有几种可能的形状. 其中的一些, 如 **菱形十二面体** 与 **截角八面体** [18]比立方体更接近于球形, 并因此更适合用于研究溶液中近似球形的大分子, 这是由于对给定的大分子映像的最小距离, 填充这两种形状的盒子需要的溶剂分子更少. 同时, 菱形十二面体和截角八面体是 **三斜晶胞** 的特殊情况, 而三斜晶胞是最一般的空间填充元胞, 组成了所有可能的空间填充形状 [19]. 出于这个原因, GROMACS是基于三斜晶胞的.

GROMACS使用周期性边界条件, 并结合了 **最小映像约定**: 对短程非键相互作用, 每个粒子只考虑它的一个最近的映像. 对长程静电相互作用, 这有时不够准确, 因此GROMACS还采用了点阵加和的方法, 如Ewald 加和, PME和PPPM.

GROMACS支持任何形状的三斜盒子. 模拟盒(元胞)由3个盒矢量 **a, b, c** 定义. 盒矢量必须满足以下条件:

$$a_y = a_z = b_z = 0 \quad (3.1)$$

$$a_x > 0, b_y > 0, c_z > 0 \quad (3.2)$$

$$|b_x| \leq \frac{1}{2}a_x, |c_x| \leq \frac{1}{2}a_x, |c_y| \leq \frac{1}{2}b_y \quad (3.3)$$

通过旋转盒子, 总可以使得等式3.1成立, 不等式(3.2)和(3.3)也总可以通过增减盒矢量使其成立.

即便使用三斜盒子进行模拟, 为了提高计算效率, GROMAC始终将粒子放在长方体空间内, 对二维体系的图示见图3.1. 因此, 从输出的轨迹来看, 模拟好像是在长方体盒子中进行的. `trjconv` 程序可用于转换轨迹, 以便在不同的元胞中显示.

模拟时也可不使用周期性边界条件, 但当模拟孤立的分子团簇时, 使用一个很大的周期性盒子通常效率更高, 这是因为快速的格点搜索只能在周期性体系中使用.

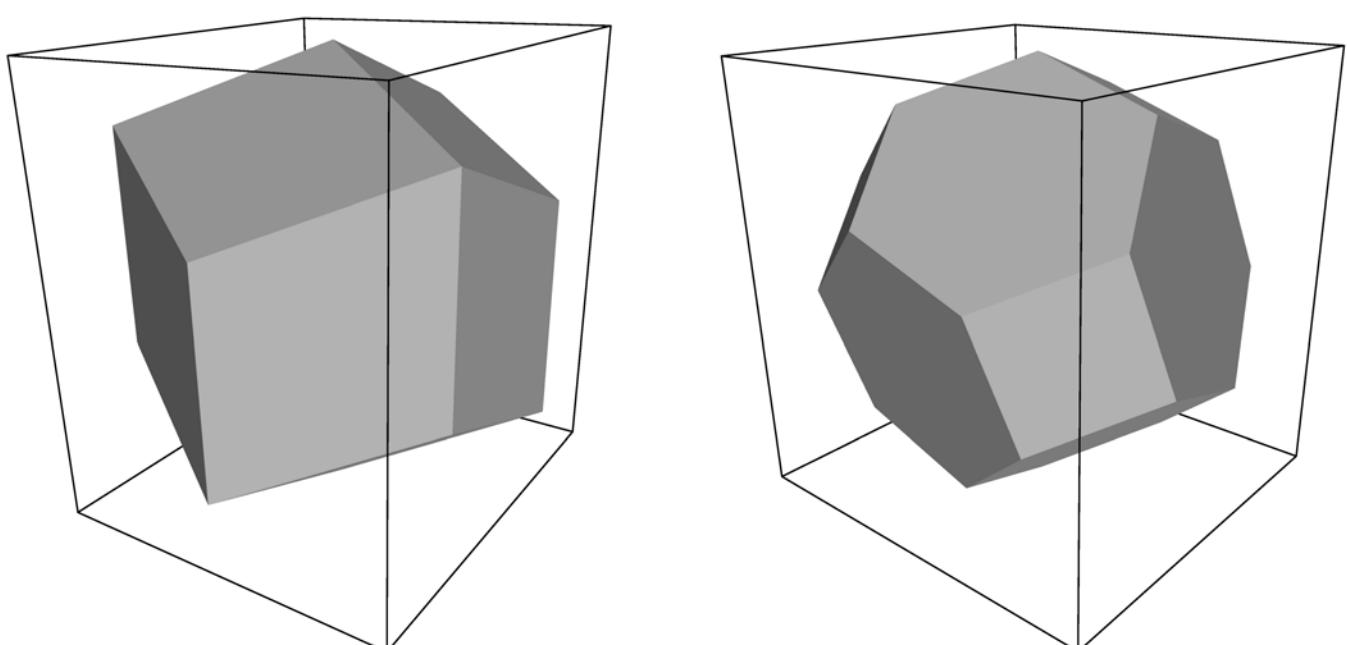


图3.2 菱形十二面体与截角八面体(任意取向)
表3.1: 立方盒, 菱形十二面体和截角八面体

盒子类型	映像距离	盒子体积	盒矢量 abc	盒矢量夹角
			$d \ 0 \ 0$	

立方		d^3	$\begin{matrix} 0 & d & 0 \\ 0 & 0 & d \end{matrix}$	90° 90° 90°
菱形十二面体(XY-正方)	d	$\frac{1}{2}\sqrt{2}d^3$ $0.707d^3$	$d \ 0 \ \frac{1}{2}d$ $0 \ d \ \frac{1}{2}d$ $0 \ 0 \ \frac{1}{2}\sqrt{2}d$	60° 60° 90°
菱形十二面体(XY-六角)			$d \ \frac{1}{2}d \ \frac{1}{2}d$ $0 \ \frac{1}{2}\sqrt{3}d \ \frac{1}{6}\sqrt{3}d$ $0 \ 0 \ \frac{1}{3}\sqrt{6}d$	
截角八面体			$\frac{4}{9}\sqrt{3}d^3$ $0.770d^3$	$d \ \frac{1}{3}d \ -\frac{1}{3}d$ $0 \ \frac{2}{3}\sqrt{2}d \ \frac{1}{3}\sqrt{2}d$ $0 \ 0 \ \frac{1}{3}\sqrt{6}d$

3.2.1 一些有用的盒子类型

表3.1列出了用于模拟溶液体系的三种最常用的盒子类型. 菱形十二面体(图3.2)是最小, 最规则的空间填充元胞. 12个映像元胞的每一个具有相同距离, 其体积是具有相同映像距离的立方体体积的71%. 当模拟溶液中球形或柔性的分子时, 这样可以节省约29%的CPU时间. 有两种不同取向的菱形十二面体满足方程3.1, 3.2和3.3. 程序editconf生成与xy平面有正方截面的菱形十二面. 之所以选择这个取向, 是因为该取向的菱形十二面的前两个盒矢量与x轴和y轴重合, 因此容易理解. 另一取向可用于膜蛋白的模拟. 这时它在xy平面的截面是一个六边形, 其面积比具有相同的映像距离的正方形的面积小14%. 为获得最佳间距, 可以改变盒子的高度(c_z). 这种盒子形状不仅可以节省CPU时间, 也可以使蛋白质的分布更加均匀.

3.2.2 截断限制

最小映像约定意味着, 用于截断非键相互作用的截断半径不能超过最短盒矢量的一半:

$$R_c < \frac{1}{2}\min(\|\mathbf{a}\|, \|\mathbf{b}\|, \|\mathbf{c}\|) \quad (3.4)$$

否则将会有多个映像出现在力的截断距离内. 当研究溶液中的大分子——如蛋白质——时, 仅考虑这个限制是不够的: 原则上, 单个的溶剂分子应该无法同时“看到”大分子的两侧. 这意味着, 每个盒矢量的长度都必须超过大分子在该方向的长度 再加上 两倍的截断半径 R_c . 但人们经常不严格遵守这个限制, 以便稍微减少溶剂层从而节约计算成本. 出于效率的考虑, 三斜盒子的截断更受限制. 对格点搜索的额外限制弱一些:

$$R_c < \min(a_x, b_y, c_z) \quad (3.5)$$

对简单搜索的额外限制更强:

$$R_c < \frac{1}{2}\min(a_x, b_y, c_z) \quad (3.6)$$

每个元胞(立方, 长方或三斜)都被26个平移的映像所包围. 因此一个特定的映像始终可以用指向27个 平移矢量 之一的索引来指认, 并通过对索引矢量进行平移来构建(见3.4.3). 约束式(3.5)保证了只需要考虑26个映像.

3.3 组的概念

GROMACS的MD和分析程序可对用户自定义的原子组 进行一些操作. 组的最大数目为256, 但每个原子最多只能属于六类不同的组. 这六类组如下:

温度耦合组 对每个温度耦合组可单独定义温度耦合参数(参考温度, 时间常数, 自由度数目, 参见3.4.4). 例如, 在高分子溶液中, 相比高分子, 溶剂分子(力和积分的误差使其容易产生更多热效应)可以使用更短的时间常数与热浴耦合, 又如, 可以将表面的温度维持得低于比吸附分子的温度. 可以定义许多不同的温度耦合组. 也请参看下面质心组.

冻结组 属于冻结组的原子在模拟过程中始终保持静止. 在平衡体系的过程中这是很有用的, 例如, 可避免不当放置的溶剂分子对蛋白质的原子产生不合理的碰撞, 尽管对受保护原子添加约束势可以得到同样的效果. 如果需要, 冻结选项可仅仅用于原子的一个或两个坐标, 从而将原子冻结在一个平面或一条直线上. 当一个原子被部分冻结时, 它所受约束仍然可以使它移动, 即便是在冻结方向上. 一个被完全冻结的原子不能被它所受的约束移动. 可定义许多冻结组. 冻结坐标不受压力缩放影响; 在某些情况下, 这可能会产生无用的结果, 尤其是与约束同用时(在这种情况下, 你会得到非常大的压力). 因此, 我们建议避免将冻结组与约束和压力耦合混合使用. 为了平衡体系, 可以先在使用冻结进行等体积模拟时, 然后再使用位置约束与恒压模拟.

加速组 加速组的每个原子上会被加上一个加速度 a^g , 这等同于受到一个外力. 利用这个特性可驱使体系进入非平衡态, 并进行非平衡MD模拟以计算输运性质.

能量监测组 在模拟中, 所有能量监测组之间的交叉相互作用都会被考虑, 且对Lennard-Jones 项和库仑项的计算是分开进行的. 原则上, 最多可定义256组, 但这样将会计算 256×256 项相互作用! 最好保守地使用.

能量监测组之间的所有非键相互作用都可以被排除在外(参见7.3节). 在能量监测组中被排除的粒子对不会被放入配对列表, 当不需要计算体系中的某些相互作用时, 这样做可以显著提高模拟速度.

质心组 GROMACS可以移除质心(center of mass, COM) 的运动, 无论是整个体系的质心还是或原子组的质心. 后者(移除原子组的质心)是有用的, 比如, 对于存在有限的阻止质心运动的摩擦的体系(如气体体系). 对温度耦合与质心运动移除使用相同的组是合理的.

压缩位置输出组 为了进一步减小压缩轨迹文件(.xtc或.tng)的大小, 可以仅仅存储一部分粒子的轨迹. 所有标记为压缩组的会被保存, 其他则不会. 如果没有指定这样的输出组, 所有的原子坐标都将被保存到压缩轨迹文件.

GROMACS工具中组的使用将在8.1节详述.

3.4 分子动力学

MD的整个流程见图3.3. 每次运行MD或EM都需要体系中所有粒子一组初始坐标和(可选的)初始速度作为输入. 本章不介绍如何获得它们; 对于如何设置并实际运行MD, 请参考www.gromacs.org 上的在线手册.

MD 算法流程概览

1. 输入初始条件

作为原子位置函数的势能相互作用 V
 体系中所有原子的位置 \mathbf{r}
 体系中所有原子的速度 \mathbf{v}



对所需计算的步数重复2,3,4步

2. 计算力

任一原子的受力

$$\mathbf{F}_i = -\frac{\partial V}{\partial \mathbf{r}_i}$$

等于原子之间的非键相互作用力

$$\mathbf{F}_i = \sum_j \mathbf{F}_{ij}$$

加上原子之间的键合相互作用力(可能与1, 2, 3或4个原子相关), 再加上约束和/或外力
 可能需要计算势能, 动能和压力张量



3. 更新构型

通过数值求解牛顿运动方程模拟原子的运动

$$\frac{d^2 \mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m_i}$$

or

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i; \quad \frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{F}_i}{m_i}$$



4. 如果需要, 输出单步结果

输出位置, 速度, 能量, 温度, 压力等

图3.3: MD算法流程概览

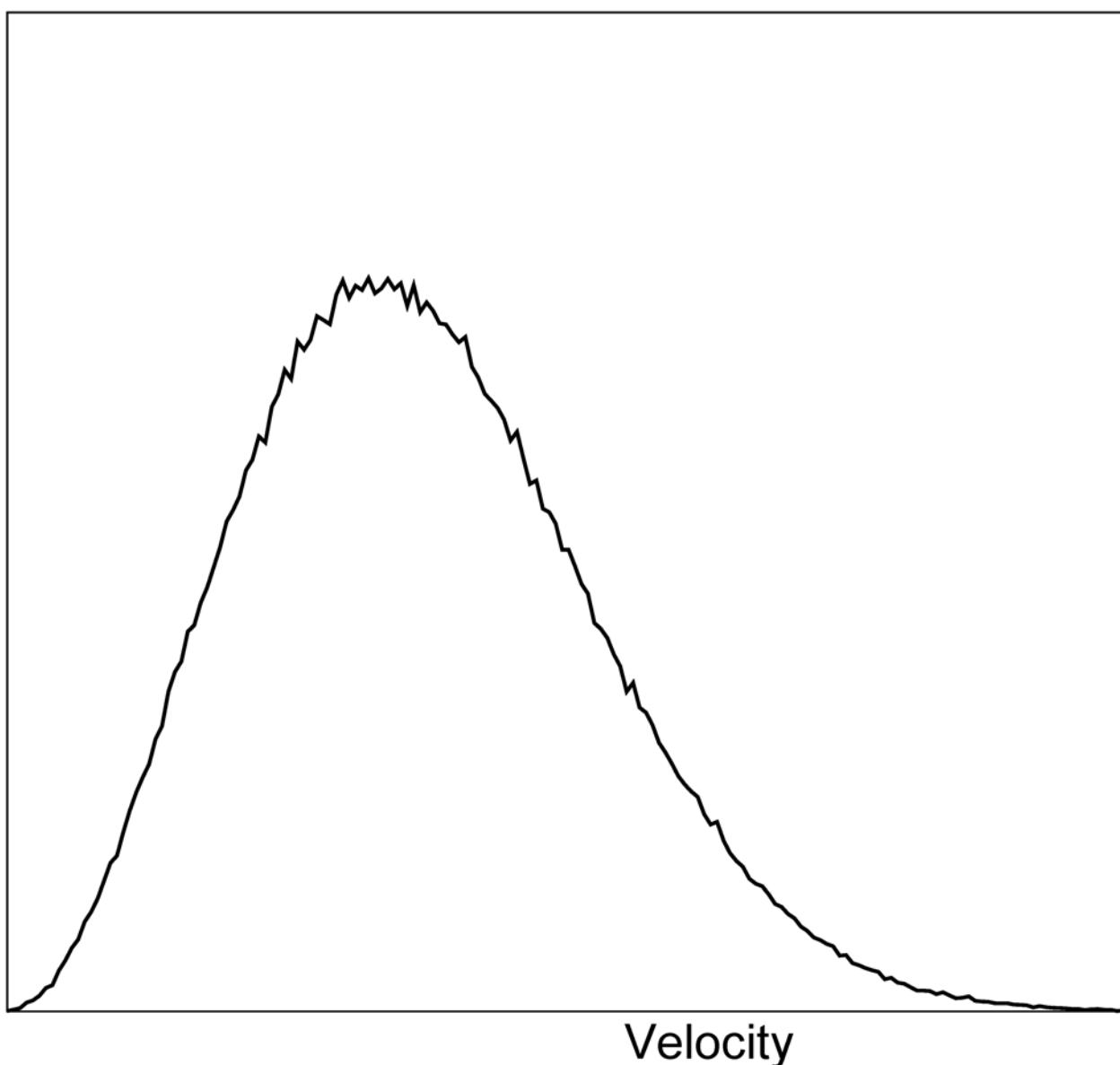


图3.4: 麦克斯韦-玻尔兹曼速度分布, 通过随机数生成

3.4.1 初始条件

拓扑和力场

必须读入体系的拓扑, 它包含了对力场的说明. 力场和拓扑将分别在第四章和第五章进行说明. 所有这些信息都是固定的, 在运行过程中始终不变.

坐标和速度

然后, 在开始运行前, 需要知道盒子的尺寸, 所有粒子的坐标和速度. 盒子的尺寸和形状是由三个矢量(9个数字) $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ 决定的, 它们代表了周期性盒子的三个基矢量.

如果从 $t = t_0$ 开始运行, 必须知道 $t = t_0$ 时刻的坐标. 如果以 Δt 为时间步长, 使用默认的 蛙跳式算法 更新时间步(参见3.4.4节), 还需要知道 $t = t_0 - \frac{1}{2} \Delta t$ 时刻的速度. 如果没有可用的速度, 程序可以根据给定的绝对温度 T 产生所需要的初始原子速度 $v_i, i = 1 \dots 3N$ (图3.4):

$$p(v_i) = \sqrt{\frac{m_i}{2\pi kT}} \exp\left(-\frac{m_i v_i^2}{2kT}\right) \quad (3.7)$$

其中 k 为玻耳兹曼常数(见第二章). 为此, 会使用产生12个随机数 $R_k, 0 \leq R_k < 1$ 并从其和中减去6的方法来产生服从正态分布

的随机数. 结果会乘上速度分布的标准偏差 $\sqrt{kT/m_i}$. 因为由此得到的总能量不会精确地对应于所需的温度 T , 需要进行校正: 首先移除质心的运动, 然后再对所有的速度进行缩放, 以使总能量恰好对应于 T (参见公式3.18).

质心运动

正常情况下, 在每一步中都会将质心的速度设置为零, 体系(通常)不受净的合外力作用, 其质心速度应该保持不变. 然而在实际中, 更新算法会导致质心速度发生缓慢的变化, 并因此使得体系的总动能发生变化, 特别是使用温度耦合时. 如果不对这种变化进行处理, 长时间后质心会有明显的运动, 温度也出现显著异常. 体系的整体转动也会导致同样的问题, 但只出现在孤立团簇的模拟中. 在填满盒子的周期性体系中, 整体转动会与其它自由度耦合, 不会导致这个问题.

3.4.2 近邻搜索

正如在第四章提到的, 内力或者由固定(静态)列表产生, 或者由动态列表产生. 后者包含了任意粒子对之间的非键相互作用. 当计算非键力时, 将所有粒子置于长方体盒子中计算更容易. 如图3.1所示, 可以通过一定方法将三斜盒转换为长方盒. 所以GROMACS的输出坐标总是处于一个长方盒中, 即使模拟时使用了十二面体盒子或三斜盒子. 方程3.1保证了我们可以通过如下方法重置粒子使其处于长方盒中: 先沿盒矢量 c 移动, 再沿 b 移动, 最后沿 a 移动. 根据三个限制性方程3.3, 3.4和3.5, 我们可以利用不超过一倍盒矢量的线性组合找到14个最近的三斜盒子映象.

配对列表的生成

只需要对一些粒子对 i, j 之间的非键配对力进行计算, 在这些粒子对中, 粒子 i 和 j 的最近映象之间的距离 r_{ij} 小于给定的截断半径 R_c . 如果彼此之间的相互作用已完全被键合作用所考虑, 一些满足这一条件的粒子对仍然会被排除. GROMACS使用了一个配对列表, 其中包含了那些必须计算彼此之间非键力的粒子对. 这个列表中包含原子 i , 原子 i 的位移向量, 距离原子 i 的这个特殊映象 `rlist` 范围内的所有粒子 j . 该列表每 `nstlist` 步更新一次, `nstlist` 的典型值为10. 有一个选项可用来计算每个粒子所受到的总的非键力, 这些力来源于围绕列表截断值, 即距离在 `rlist` 和 `rlistlong` 之间的壳层中的所有粒子. 在更新配对列表时, 会计算这些力, 并在随后的 `nstlist` 中保持不变.

为创建邻区列表, 必须找到与给定粒子相近(即在邻居列表截断内)的所有粒子. 这种搜索通常被称为邻区搜索(NS, neighbor search)或对搜索(pair search), 涉及到周期性边界条件和映象的确定(参见3.2节). 搜索算法的复杂度为 $O(N)$, 尽管更简单的 $O(N^2)$ 算法在一定条件下仍然可用.

邻区截断方案: 原子组与Verlet缓冲

从4.6版本开始, GROMACS支持两种不同的截断方案设置: 最初的基于原子组的方案和使用Verlet 缓冲区的方案. 它们之间存在一些非常重要的区别, 这些区别可能会影响计算结果, 计算性能和某些功能的支持情况. 组方案(几乎)可以像Verlet方案一样运行, 但这将导致性能降低. 对在模拟中常用的水分子, 组方案特别快, 但在最近的x86处理器中, 这种优势消失了, 因为可以在Verlet方案的实现中使用更好的指令级并行. 在5.0版本中已经不再提倡使用组方案了, 将来的版本中将会删除此方案.

在组方案中, 近邻列表由至少含一个原子的原子对构成. 这些原子组最初是电荷组(参见3.4.2节), 但对长程静电进行了妥善处理, 性能是它们唯一的优势. 几何中心在截断距离内的两个组会被放入近邻列表中. 在一定数目的MD模拟步中, 会计算所有原子对(每个电荷组一个)之间的相互作用力, 直到下一次更新近邻列表. 这种设置高效, 因为近邻搜索只检查电荷组对, 而不是原子对(对三原子水模型可节约 $3 \times 3 = 9$ 倍), 而且还可以优化非键合力的计算内核, 比如说一个水分子“组”. 没有明确的缓冲时, 由于一些原子对处于截断距离之内却没有相互作用, 而一些原子对处于截断距离之外却有相互作用, 这种设置会导致能量的漂移. 可能有以下原因:

- 原子在两次近邻搜索之间穿过了截断区域, 和/或
- 对包含一个以上原子的电荷组, 当电荷组的几何中心移动到截断距离以外/内时, 会导致一部分原子对移动到截断距离以内/外.

显式地为近邻列表增加一个缓冲会消除这些问题, 但这需要较高的计算成本. 问题的严重程度取决于体系, 你感兴趣的性质, 以及截

断距离的设置.

Verlet截断方案默认使用缓冲对列表. 它也使用了原子团簇, 但这些不像在组方案中是静态的. 相反, 团簇以空间定义, 包含4个或8个原子, 使用如SSE, AVX和GPU的CUDA等可方便地对此进行流计算. 在近邻搜索步骤中, 使用Verlet缓冲创建对列表, 即对列表的截断距离大于相互作用的截断距离. 在计算非键力的内核中, 只有当一个原子对在特定时间步处于截断距离之内时, 这个力才会被加入到近邻列表中. 当原子在两次对搜索步骤中移动时, 这确保了几乎所有处于截断距离内的原子之间的力都会被计算. 我们说, 几乎所有的原子, 是因为GROMACS使用了一个固定的对列表更新频率以提高效率. 一个处于截断距离外的原子对, 在这样固定的步数中, 可能移动得足够多以致于处于截断距离之内. 这种小概率事件会导致小的能量漂移, 而且概率的大小取决于温度. 当使用温度耦合时, 给定能量漂移的一定容差, 可以自动确定缓冲大小.

在`mdp`文件中Verlet方案的设置为:

```
cutoff-scheme      = Verlet
Verlet-buffer-tolerance = 0.005
```

Verlet缓冲的大小由后一选项决定, 这里使用了默认设置: 在一对原子中每个原子的能量误差为0.005 kJ/mol/ps. 需要注意的是, 在配对能量的误差会相互抵消, 这对总能量漂移的影响通常至少比容差小一个数量级. 此外, 总能量的漂移受许多其他因素影响, 通常, 来自约束算法的贡献占主导地位.

对等能量(NVE)模拟, 缓冲大小由对应于速度(无论由何种合适的方法产生, 或由输入文件提供)的温度来决定. 此外, 容差可以设置为-1, 并手动指定缓冲: `rlist > max(rcoulomb, rvdw)`. 得到合理缓冲大小的最简单方法是: 利用NVT的`mdp`文件并将目标温度设定为NVE模拟中你希望使用的温度, 然后将`grompp`打印出的缓冲大小移植到NVE的`mdp`文件中.

基于粒子团簇实现的Verlet截断方案非常高效. 最简单的例子是包含4个粒子的团簇. 然后基于团簇对构建对列表. 团簇对搜索比基于粒子对的搜索快得多, 因为一次可以将 $4 \times 4 = 16$ 个粒子置于列表中. 然后就可以一次计算这16个粒子对之间的非键相互作用, 这可以很好地映射到SIMD单元, 它一次可以执行多个浮点运算(例如基于SSE, AVX, GPU的CUDA, BlueGene FPUs). 在绝大多数类型的体系中, 这种非键相互作用的计算内核要比组方案的内核快很多, 除了在短SIMD宽度的处理器上, 对水分子不使用缓冲对列表时. 后者在(生物)分子模拟中很常见, 所以为了达到最快速度, 值得比较两种方案的性能.

由于Verlet截断方案在4.6版本的GROMACS中才推出, 所以它并不支持组方案中的所有功能. Verlet截断方案支持一些组方案不支持的功能, 表3.2给出了两种截断方案中不(完全)支持的功能.

表3.2: 组方案与Verlet截断方案之间(仅)在非键功能上支持的区别

非键相互作用特性	组	Verlet
无缓冲截断方式	✓	非默认
精确截断	移位/切换	✓
移位相互作用	力+能量	能量
切换势能	✓	✓
切换力	✓	✓
非周期性体系	✓	Z+墙
隐式溶剂	✓	
自由能微扰的非键相互作用	✓	✓
组能量贡献	✓	CPU(而非GPU)
能量组排除	✓	
AdResS多尺度	✓	
OpenMP多线程	仅PME	✓
GPU原生支持		✓

能能源漂移与对列表缓冲

对于一个正则系综(NVT), 由有限的Verlet 缓冲大小所造成的平均能量误差, 可以由截断处的原子位移和势能形状来确定. 在温度 T 下, 一个质量为 m 的自由粒子, 其一维方向上的位移在时间 t 内的分布为高斯分布, 均值为0, 方差为 $\sigma^2 = tk_B T / m$. 对两个原子之间的距离, 其方差变为 $\sigma^2 = \sigma_{12}^2 = tk_B T (1/m_1 + 1/m_2)$. 注意, 在实际过程中, 时间 t 内粒子通常还会与其他粒子相互作用, 所以实际的位移分布会窄得多. 给定非键相互作用的截断距离 r_c , 对列表截断距离 $r_l = r_c + r_b$. 当类型为1的粒子被数密度为 ρ_2 的粒子类型2包围, 粒子间距从 r_0 变到 r_t 时, 我们可以得到在时间 t 之后1和2两种类型的粒子之间的相互作用的平均能量误差:

$$\langle \Delta V \rangle = \int_0^{r_c} \int_{r_l}^{\infty} 4\pi r_0^2 \rho_2 V(r_t) G\left(\frac{r_t - r_0}{\sigma}\right) dr_0 dr_t \quad (3.8)$$

$$\approx \int_{-\infty}^{r_c} \int_{r_l}^{\infty} 4\pi r_0^2 \rho_2 \left[V'(r_c)(r_t - r_c) + V''(r_c) \frac{1}{2}(r_t - r_c)^2 \right] G\left(\frac{r_t - r_0}{\sigma}\right) dr_0 dr_t \quad (3.9)$$

$$\approx 4\pi(r_l + \sigma)^2 \rho_2 \int_{-\infty}^{r_c} \int_{r_l}^{\infty} \left[V'(r_c)(r_t - r_c) + V''(r_c) \frac{1}{2}(r_t - r_c)^2 + V'''(r_c) \frac{1}{6}(r_t - r_c)^3 \right] G\left(\frac{r_t - r_0}{\sigma}\right) dr_0 dr_t \quad (3.10)$$

$$= 4\pi(r_l + \sigma)^2 \rho_2 \left\{ \frac{1}{2} V'(r_c) \left[r_b \sigma G\left(\frac{r_b}{\sigma}\right) - (r_b^2 + \sigma^2) E\left(\frac{r_b}{\sigma}\right) \right] + \frac{1}{6} V''(r_c) \left[\sigma(r_b^2 + 2\sigma^2) G\left(\frac{r_b}{\sigma}\right) - r_b(r_b^2 + 3\sigma^2) E\left(\frac{r_b}{\sigma}\right) \right] + \frac{1}{24} V'''(r_c) \left[r_b \sigma(r_b^2 + 5\sigma^2) G\left(\frac{r_b}{\sigma}\right) - (r_b^4 + 6r_b^2\sigma^2 + 3\sigma^2) E\left(\frac{r_b}{\sigma}\right) \right] \right\} \quad (3.11)$$

其中 G 为零均值单位方差的高斯分布, $E(x) = \frac{1}{2} \operatorname{erfc}(x/\sqrt{2})$. 我们总是希望得到的误差很小, 所以 σ 比 r_c 和 r_l 都小, 这样上述方程中的近似很好, 因为高斯分布衰减很快. 能量误差需要对所有粒子对类型进行平均, 并对粒子数进行加权. 在GROMACS中我们不允许粒子对类型之间的误差抵消, 所以我们对绝对值进行平均. 为获得单位时间的平均能量误差, 需要将平均能量误差除以近邻列表时间长度 $t = (\text{nslist} - 1) \times \text{dt}$. 此函数不能解析地得到, 因此对给定的目标漂移, 我们使用二分法来获得缓冲大小 r_b . 我们还注意到, 在实际中误差通常比这里估计的小很多, 如凝聚相中粒子的位移会比自由移动粒子的位移小得多, 这里使用了这个假定.

当存在(键)约束时, 一些粒子的自由度会变少. 这会减少能量误差. 一个具有两个自由度的粒子任意方向上的位移并不是高斯分布, 而是服从余误差函数:

$$\frac{\sqrt{\pi}}{2\sqrt{2}\sigma} \operatorname{erfc}\left(\frac{|r|}{\sqrt{2}\sigma}\right) \quad (3.12)$$

其中 σ^2 为 $k_B T / m$. 没有办法解析地对此分布进行积分以获得能量误差. 但我们可以使用缩放和移位高斯分布来得到紧上界(这里未给出). 这种高斯分布可用于计算能量误差, 如上所述. 现在我们来考虑约束粒子, 即具有2个或更少自由度的粒子, 它们通过约束连接到其他粒子上, 其他粒子的总质量至少为约束粒子本身质量1.5倍. 对具有单个约束的粒子, 沿约束方向的总质量至少是2.5倍, 这样沿约束方向的位移方差至少减少为原来的1/6.25. 由于高斯分布衰减非常迅速, 这有效地移除了来自位移的一个自由度. 多重约束会减少更多位移, 但因为会变得非常复杂, 我们将那些粒子视为只有2个自由度的粒子.

还有一个重要的实现细节, 可减少因有限的Verlet 缓冲列表大小造成的能力误差. 上述推导假定了粒子对列表. 然而, GROMACS实现时使用了团簇对列表以提高效率. 在大多数情况下, 对列表包含4个粒子组成的团簇之间的对, 也称为 4×4 列表, 但也可以是

4×8 (GPU CUDA内核和AVX 256位单精度内核)或 4×2 (SSE双精度内核).这意味着对列表比相应的 1×1 列表高效得多.这样稍微超出对列表截断时,列表中仍然会有一大部分粒子.这些粒子所占的总粒子的分数可以在模拟中确定,并精确地根据一些合理的假设进行估计.随着对列表范围的增加,这些粒子分数减小,意味着可以使用较小的缓冲.对于典型的全原子模拟,使用0.9 nm截断,这个分数大约是0.9,可使能量误差减少到原来的1/10.在自动Verlet的缓冲计算时考虑了这种减少,这样得到的缓冲尺寸较小.

由图3.5可以看出,对小的缓冲大小,由于误差抵消,总能量漂移比对能量误差容差小得多.对于较大的缓冲大小,误差估计比总能量漂移高6倍,或者缓冲估计大0.024 nm.这是因为质子自由移动不能超过18 fs,但振动可以.

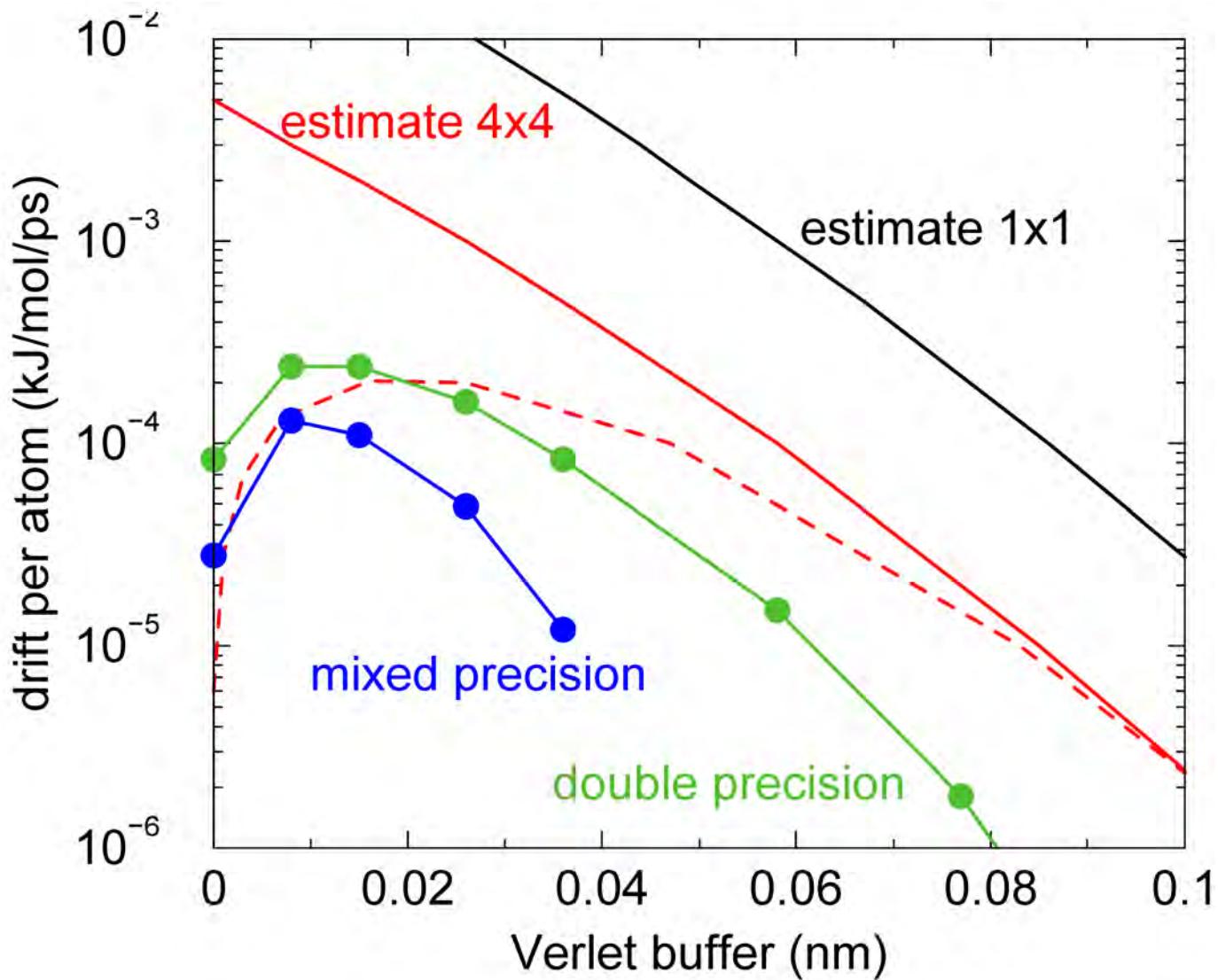


图3.5: 温度 300 K时, SPC/E水体系中每个原子的能量漂移, 模拟使用的时间步长为2 fs, 对列表更新周期为10步(对列表更新时间长度: 18 fs). 使用了PME方法, `ewald-rtol`设置为 10^{-5} ; 该参数影响截断处的势能形状. 图中显示了因有限Verlet 缓冲大小, 对 1×1 和 4×4 原子对列表, 不使用和使用(虚线)正负误差抵消时引起的误差估计. 图中显示的实际能源漂移在模拟时使用了双精度和混合精度. 使用单精度时, 由SETTLE约束算法的舍入误差引起的漂移在缓冲尺寸很大时变为负值. 注意, 缓冲大小为零时, 实际漂移很小, 因为正的(H-H)和负的(O-H)能量误差会相互抵消.

截断假象与切换相互作用

使用Verlet方案时, 可以平移对势使其在截断处为零, 以保证势能等于力的积分. 在组方案中, 只有当势能满足在截断距离处的值为零时, 才可以保证势能等于力的积分. 然而, 当力在截断处不为零时仍会引起能量漂移. 这种影响非常小, 通常觉察不到, 因为其它积分误差(如来自约束的)可能占主导地位. 为完全避免截断假象, 可以将非键力在到达近邻列表截断前精确地切换至零(在GROMACS中有几种方式完成, 见4.1.5节). 这样就形成一个大小等于近邻列表截断, 小于最长相互作用截断的缓冲.

使用组截断方案, 也可以选择让`mdrun`只在需要时才更新近邻列表, 也就是当一个或多个粒子从它们所属的电荷组(见3.4.2节)的几何中心移动的距离超过半个缓冲大小时, 中心由前一步的近邻搜索决定. 此选项可保证没有截断假象, 注意, 对较大的体系, 这种计算的代价很高, 因为近邻列表更新频率将由移动稍微超过半个缓冲长度的一两个粒子决定(这甚至都不意味着近邻列表无效), 而

99.99%的粒子近邻列表依然有效.

简单搜索

由于方程3.1和3.6, 连接处于截断距离 R_c 内的映象的向量 \mathbf{r}_{ij} 可通过下面方法获得:

$$\mathbf{r}''' = \mathbf{r}_j - \mathbf{r}_i \quad (3.13)$$

$$\mathbf{r}'' = \mathbf{r}''' - \mathbf{c} * \text{round}(r_z''' / c_z) \quad (3.14)$$

$$\mathbf{r}' = \mathbf{r}'' - \mathbf{b} * \text{round}(r_y'' / b_y) \quad (3.15)$$

$$\mathbf{r}_{ij} = \mathbf{r}' - \mathbf{a} * \text{round}(r_x' / a_x) \quad (3.16)$$

当需要计算三斜盒子中两粒子间的距离时, 它们不服从方程3.1, 需要将许多盒向量的组合进行移位, 以找到最近的映象.

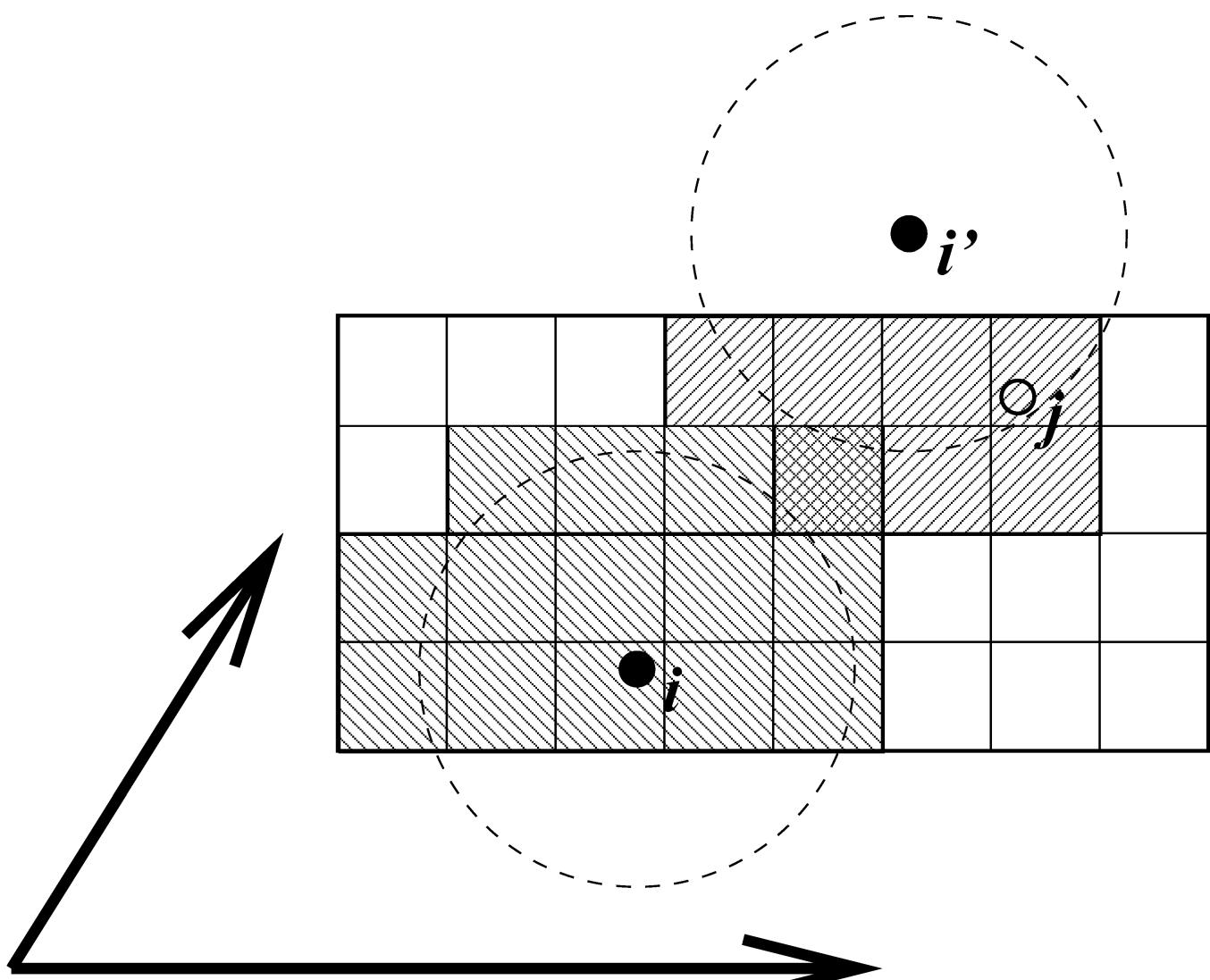


图3.6: 二维格点搜索. 箭头为盒向量.

格点搜索

格点搜索示意图见图3.6. 所有粒子被置于NS格点上, 每一方向上的最小间距 $\geq R_c/2$. 在每一盒向量方向上, 粒子 i 有三个映象. 对每一方向映象可能是-1, 0或1, 相应于平移-1, 0或+1个盒向量. 对 i 的近邻, 我们不是先搜索周围的NS格点单元再计算映象, 而是先构造映象再搜索对应 i 的映象的近邻. 如图3.6所示, 对 i 的不同映象, 一些格点单元的搜索次数会查过一. 这并不是一个问题, 因为由于最小映象约定, 最多只有一个映象能够“看到”粒子 j . 对每个粒子, 搜索的近邻单元少于 $125(5^3)$ 个. 因此, 算法对粒子数是线性标度的. 虽然前因子很大, 当粒子数超过几百时, 标度行为使得算法远优于标准的 $O(N^2)$ 算法. 格点搜索对长方盒子与三斜盒子同样快. 这样对大多数蛋白质和多肽模拟, 使用菱形十二面体盒子更好.

电荷组

最初引入电荷组是为了减少库仑相互作用的截断假象。使用普通截断时，当带(部分)电荷的原子移进移出截断半径时，势能与力会出现明显的跃变。当所有化学基团的净电荷为零时，通过移动净电荷为零的原子组进出近邻列表可以减小这些跃变。这些原子组被称为电荷组。这种方法将截断效应从电荷电荷水平降低到衰减快得多的偶极偶极水平。随着全范围静电方法的出现，如粒子网格Ewald方法(见4.8.2节)，不再需要使用电荷组来提高精度了。依赖于如何创建近邻列表，如何计算相互作用，电荷组甚至可能对精度和效率有小的负面影响。

但仍然有一个重要原因来使用“电荷组”：组截断方案的效率。当适用时，近邻搜索基于在分子拓扑中定义的电荷组进行。若两个电荷组原子的 几何中心 之间的最近映象距离小于截断半径，对列表中会包含电荷组之间的所有原子对。例如，对水体系的近邻搜索，当每个分子被视为一个电荷组时 会快 $3^2=9$ 倍。另外，计算水中原子受力的高度优化的循环只能用于水分子中的所有原子形成单个电荷组的情况。目前使用 近邻搜索组 可能更合适，但由于历史原因仍保留了名称电荷组。当发展新的力场时，建议使用3到4个原子的电荷组以获得最佳性能。对全原子力场这相对容易，因为可以简单地把氢原子，一些情况下的氧原子，置于与其相连的重原子相同的电荷组中，如CH₃, CH₂, CH, NH₂, NH, OH, CO₂, CO.

使用Verlet 截断方案时，电荷组被忽略。

3.4.3 计算力

势能

当计算力时，同时会计算每项相互作用的势能。总势能是所有项的加和，如Lennard-Jones 项，库仑项和键合项。也可以计算 能量监测组 对这些项的贡献，组成能量监测组的原子可以单独定义(参见3.3节)。

动能和温度

对含有 N 个粒子的体系，温度由总动能给出：

$$E_{\text{kin}} = \frac{1}{2} \sum_{i=1}^N m_i v_i^2 \quad (3.17)$$

由此可计算绝对温度 T :

$$\frac{1}{2} N_{\text{df}} k T = E_{\text{kin}} \quad (3.18)$$

其中 k 为玻尔兹曼常数， N_{df} 为自由度数，可根据下式计算：

$$N_{\text{df}} = 3N - N_c - N_{\text{com}} \quad (3.19)$$

这里 N_c 为施加在体系上的 约束 的数目。在进行分子动力学模拟时，必须去除 $N_{\text{com}} = 3$ 个额外的自由度，因为三个质心速度是运动常量，通常设置为零。当在真空中进行模拟时，绕质心的转动也可以被去除，这种情况下 $N_{\text{com}} = 6$ 。当使用一个以上的温度耦合组时，第 i 组的自由度数目为：

$$N_{\text{df}}^i = (3N^i - N_c^i) \frac{3N - N_c - N_{\text{com}}}{3N - N_c} \quad (3.20)$$

当计算三斜体系的压力，或者体系受到剪切力作用时，将动能写成张量形式能带来方便

$$\mathbf{E}_{\text{kin}} = \frac{1}{2} \sum_i^N m_i \mathbf{v}_i \otimes \mathbf{v}_i \quad (3.21)$$

压力和维里

压力张量 \mathbf{P} 可以通过动能 E_{kin} 和维里 Ξ 的差值来计算:

$$\mathbf{P} = \frac{2}{V} (\mathbf{E}_{\text{kin}} - \Xi) \quad (3.22)$$

其中 V 为盒子的体积. 在各向同性体系中被用于压力耦合的标量压力 P 为:

$$P = \text{trace}(\mathbf{P})/3 \quad (3.23)$$

维里张量 Ξ 定义为:

$$\Xi = -\frac{1}{2} \sum_{i < j} \mathbf{r}_{ij} \otimes \mathbf{F}_{ij} \quad (3.24)$$

GROMACS计算维里的实现方法请参考B.1节.

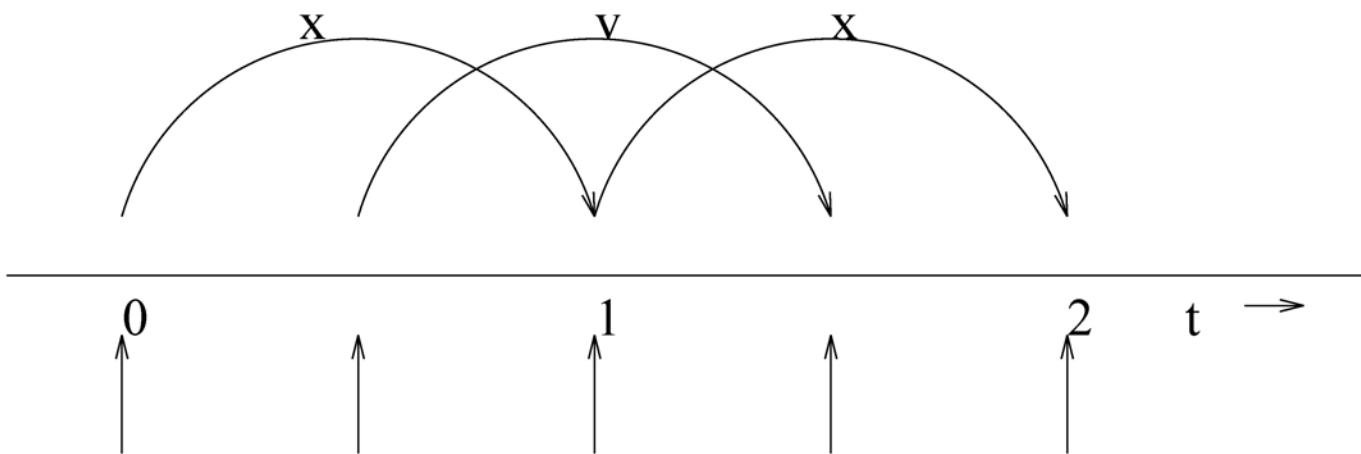


图3.7: 蛙跳式积分方法. 该算法被称为蛙跳是因为 r 和 v 好像青蛙一样在彼此的背上跳跃.

3.4.4 蛙跳式积分方法

GROMACS中默认的MD积分方法是所谓的 **蛙跳式算法** [20], 用于积分运动方程. 当需要非常精确的积分方法并采用温度和/或压力耦合时, 使用速度Verlet积分方法可能更好(参见3.4.5节). 蛙跳式算法使用了 t 时刻的位置 \mathbf{r} 和 $t - \frac{1}{2}\Delta t$ 时刻的速度 \mathbf{v} , 根据 t 时刻的位置计算出力 $\mathbf{F}(t)$, 并利用下面的方法更新位置和速度

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t - \frac{1}{2}\Delta t) + \frac{\Delta t}{m} \mathbf{F}(t) \quad (3.25)$$

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \mathbf{v}(t + \frac{1}{2}\Delta t) \quad (3.26)$$

图3.7是蛙跳式算法的形象说明. 它产生的轨迹与Verlet算法[21]完全一样, 它更新位置的方法是

$$\mathbf{r}(t + \Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) + \frac{1}{m} \mathbf{F}(t) \Delta t^2 + O(\Delta t^4) \quad (3.27)$$

蛙跳式算法对 \mathbf{r} 具有三阶精度, 并且时间可逆. 关于此算法的优点以及与其他时间积分算法的比较, 请参看文献[22].

为考虑温度耦合和压力耦合并扩展以包含约束守恒, 我们修改了运动方程, 详细情况将在后面进行讨论.

3.4.5 速度verlet积分方法

GROMACS也实现了速度Verlet算法[23], 尽管它与所有选项的整合还没有完全好. 在速度Verlet方法中, 利用 t 时刻的位置 \mathbf{r} 和速

度 \mathbf{v} 积分运动方程, 并不需要前半步时刻的速度.

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t) + \frac{\Delta t}{2m} \mathbf{F}(t) \quad (3.28)$$

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \mathbf{v}(t + \frac{1}{2}\Delta t) \quad (3.29)$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t + \frac{1}{2}\Delta t) + \frac{\Delta t}{2m} \mathbf{F}(t + \Delta t) \quad (3.30)$$

或者, 等价地,

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t \mathbf{v} + \frac{\Delta t^2}{2m} \mathbf{F}(t) \quad (3.31)$$

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\Delta t}{2m} [\mathbf{F}(t) + \mathbf{F}(t + \Delta t)] \quad (3.32)$$

不使用温度或压力耦合, 并从 对应 的起始点出发, 蛙跳式方法和速度Verlet方法会给出完全相同的轨迹, 你可以容易地利用上面给出的方程证明这一点. 给定一个具有 相同 起始点 $\mathbf{x}(0)$ 和 $\mathbf{v}(0)$ 的起始文件, 蛙跳式方法和速度Verlet方法 不会 给出相同的轨迹, 因为蛙跳式方法的速度对应于 $t = -\frac{1}{2}\Delta t$ 时刻, 而速度Verlet方法会认为它们对应于 $t = 0$ 时刻.

3.4.6 理解可逆的积分方法: Trotter分解

为进一步理解速度Verlet和蛙跳式积分方法之间的关系, 我们引入动力学中的可逆Trotter公式, 它对于理解GROMACS的热浴和压浴的实现也是非常有用的.

对一个耦合一阶微分方程系统, 可利用演化算符用得到体系从 $t = 0$ 时刻到 t 时刻的演化

$$\begin{aligned} \Gamma(t) &= \exp(iLt)\Gamma(0) \\ iL &= \dot{\Gamma} \cdot \nabla_{\Gamma} \end{aligned} \quad (3.33)$$

其中 L 为Liouville 算符, Γ 为自变量(位置和速度)的多维向量. 将精确算符的一个短时间近似, 在 $\Delta t = t/P$ 时刻是精确的, 连续应用 P 次可使体系演化为

$$\Gamma(t) = \prod_{i=1}^P \exp(iL\Delta t)\Gamma(0) \quad (3.34)$$

对NVE力学, Liouville 算子为

$$iL = \sum_{i=1}^N \mathbf{v}_i \cdot \nabla_{\mathbf{r}_i} + \sum_{i=1}^N \frac{1}{m_i} \mathbf{F}(\mathbf{r}_i) \cdot \nabla_{\mathbf{v}_i} \quad (3.35)$$

它可以被分解为两个算符的加和

$$\begin{aligned} iL_1 &= \sum_{i=1}^N \frac{1}{m_i} \mathbf{F}(\mathbf{r}_i) \cdot \nabla_{\mathbf{v}_i} \\ iL_2 &= \sum_{i=1}^N \mathbf{v}_i \cdot \nabla_{\mathbf{r}_i} \end{aligned} \quad (3.36)$$

这样, 真正力学在短时间内, 对称可逆的近似为

$$\exp(iL\Delta t) = \exp(iL_2 \frac{1}{2}\Delta t) \exp(iL_1 \Delta t) \exp(iL_2 \frac{1}{2}\Delta t) + O(\Delta t^3) \quad (3.37)$$

这对应于速度Verlet积分方法. 第一个 $\frac{1}{2}\Delta t$ 指数项对应于半个速度步, 第二个 Δt 指数项对应于一个完整的速度步, 最后的 $\frac{1}{2}\Delta t$

指数项是最终的半个速度步. 对于未来时刻 $t = n\Delta t$, 上式变为

$$\begin{aligned}\exp(iLn\Delta t) &\approx \left(\exp(iL_2 \frac{1}{2}\Delta t) \exp(iL_1 \Delta t) \exp(iL_2 \frac{1}{2}\Delta t) \right)^n \\ &\approx \exp(iL_2 \frac{1}{2}\Delta t) (\exp(iL_1 \Delta t) \exp(iL_2 \Delta t))^{n-1} \\ &\quad \exp(iL_1 \Delta t) \exp(iL_2 \frac{1}{2}\Delta t)\end{aligned}\tag{3.38}$$

利用这种形式, 我们可以很容易看到的不同Verlet积分方法之间的差异. 蛙跳式积分方法可以看作使用方程3.37的 $\exp(iL_1 \Delta t)$ 项代替半步速度项启动, 得到

$$\exp(iLn\Delta t) = \exp(iL_1 \Delta t) \exp(iL_2 \Delta t) + O(\Delta t^3)\tag{3.39}$$

这里, 速度的整步处于 $t - \frac{1}{2}\Delta t$ 和 $t + \frac{1}{2}\Delta t$ 之间, 因为它是速度Verlet中速度半步的组合. 对未来的时刻 $t = n\Delta t$, 此式变为

$$\exp(iLn\Delta t) \approx (\exp(iL_1 \Delta t) \exp(iL_2 \Delta t))^n\tag{3.40}$$

虽然上式初看起来不对称, 只要整个速度步处于 $t - \frac{1}{2}\Delta t$ 和 $t + \frac{1}{2}\Delta t$ 之间, 这就是在循环中的不同地方启动速度Verlet的一种简单方法.

尽管蛙跳式方法和速度Verlet方法得到的轨迹和势能完全相同, 动能和温度并不一定相同. 标准的速度Verlet方法使用 t 时刻的速度计算动能, 因而只有 t 时刻的温度, 动能是对所有粒子的加和

$$\begin{aligned}KE_{\text{full}}(t) &= \sum_i \left(\frac{1}{2m_i} \mathbf{v}_i(t) \right)^2 \\ &= \sum_i \frac{1}{2m_i} \left(\frac{1}{2} \mathbf{v}_i(t - \frac{1}{2}\Delta t) + \frac{1}{2} \mathbf{v}_i(t + \frac{1}{2}\Delta t) \right)^2\end{aligned}\tag{3.41}$$

平方出现于平均的 外部. 标准的蛙跳式方法基于时间步 $t + \frac{1}{2}\Delta t$ 和 $t - \frac{1}{2}\Delta t$ 的平均动能来计算 t 时刻的动能, 或对所有粒子的加和

$$KE_{\text{average}}(t) = \sum_i \frac{1}{2m_i} \left(\frac{1}{2} \mathbf{v}_i(t - \frac{1}{2}\Delta t)^2 + \frac{1}{2} \mathbf{v}_i(t + \frac{1}{2}\Delta t)^2 \right)\tag{3.42}$$

其中平方处于平均的 内部.

速度Verlet方法的非标准形式会对动能 $KE(t + \frac{1}{2}\Delta t)$ 和 $KE(t - \frac{1}{2}\Delta t)$ 进行平均, 与蛙跳式方法相同. GROMACS现在也实现了这种方法(利用 .mdp 文件中的 **md-vv-avek** 选项). 无温度耦合和压力耦合时, 半步动能平均的速度Verlet方法和蛙跳式方法在数值精度内是完全相同的. 但是, 对于控温和控压方案, 半步动能平均的速度Verlet方法和蛙跳式方法会有所不同, 我们将在控温器和控压器部分对此进行讨论.

对于给定的步长, 半步平均的动能和温度稍微准确一些; 比起整步动能(使用 **md-vv**), 采用半步平均的动能(**md** 和 **md-vv-avek**)得到的平均动能的差异更接近于小步长极限情况下获得的动能. 对NVE模拟, 这种差别通常不明显, 因为粒子的位置与速度仍然完全相同; 差别在于模拟温度含义的 解释, 而 不 在于所产生的轨迹. 虽然半步平均方法得到的动能更精确, 意味着当时间步长变大时它受的影响小些, 但这种方法的噪声更大. 与整步动能方法相比, 半步平均动能方法得到的体系总能量(动能和势能的总和)的RMS偏差更大(在大多数情况下约高一倍). 漂移仍然是相同的, 并且轨迹也完全相同.

对NVT模拟, 两种方法 将 会有差异, 正如控温部分所讨论的, 因为会调节粒子的速度以使得模拟的动能, 以任何一种方式计算, 都能达到与设定温度相应的分布. 在这种情况下, 三种方法不会给出完全相同的结果.

由于速度和位置都定义于同一时刻 t , 速度Verlet积分方法可用于一些方法中, 特别是那些严格正确的控压方法。实际上这些控压方法不能使用蛙跳式积分方法。相比蛙跳式积分方法, 速度Verlet方法多耗费的时间可以忽略不计, 但目前需要的通讯调用多一倍。在大多数情况下, 尤其是对于大的体系, 其通讯速度对并行化非常重要。热力学系综之间的差异以 $1/N$ 极限消失。当只需要NVT系综时, 蛙跳式可能是首选的积分方法。对控压模拟, 热力学细节非常重要, 只有速度Verlet方法能给出真正的系综。在任一情况下, 都可能需要采用双精度进行模拟以得到正确的热力学校正细节。

3.4.7 双程截断

为了节省计算时间, 相比变化迅速的力, 对变化缓慢的力计算时所用的频率可以低些。在GROMACS中, 可以对短程和长程非键相互作用采用这样的多重时间步长分解。直到GROMACS 4.0版本, 对此都采用了一种不可逆的积分方案, 这种方案也被用于GROMOS模拟包: 每 n 步计算一次长程力, 随后将这些值(不加修改)用于方程3.25中的后面 $n - 1$ 积分步。这种不可逆的方案能量守恒性不好, 取样可能不充分。从4.5版本开始, 采用了可逆Trotter分解方案的蛙跳式方法[24]。在这种积分方法中, 每 n 步计算一次长程力, 然后使用时间步长 $\Delta t_{LR} = n\Delta t$ 将其应用到方程3.25中的速度,

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \begin{cases} \mathbf{v}(t - \frac{1}{2}\Delta t) + \frac{1}{m}[\mathbf{F}_{SR}(t) + n\mathbf{F}_{LR}(t)\Delta t] & , \quad \text{step \% } n = 0 \\ \mathbf{v}(t - \frac{1}{2}\Delta t) + \frac{1}{m}\mathbf{F}_{SR}(t)\Delta t & , \quad \text{step \% } n \neq 0 \end{cases} \quad (3.43)$$

参数 n 等于近邻列表的更新频率。在4.5版本中, 多重时间步的速度Verlet方案进尚未完全实现。

一些其他的模拟程序对键合力和/PME网格力使用多重时间步长方法。在GROMACS中, 我们(还)没有实现这个方法, 因为我们有着不同的想法。键可以被约束(这也是对物理量子振子的更佳近似), 这允许将最小时间步长增加至更大值。这不仅将计算力的次数减半, 也将更新计算的次数减半。对于更大的时间步长, 涉及氢原子的键角振动可以使用虚拟相互作用位点移除(见6.8节), 这使得最短时间步达到多时间步方案中PME网格的更新频率。

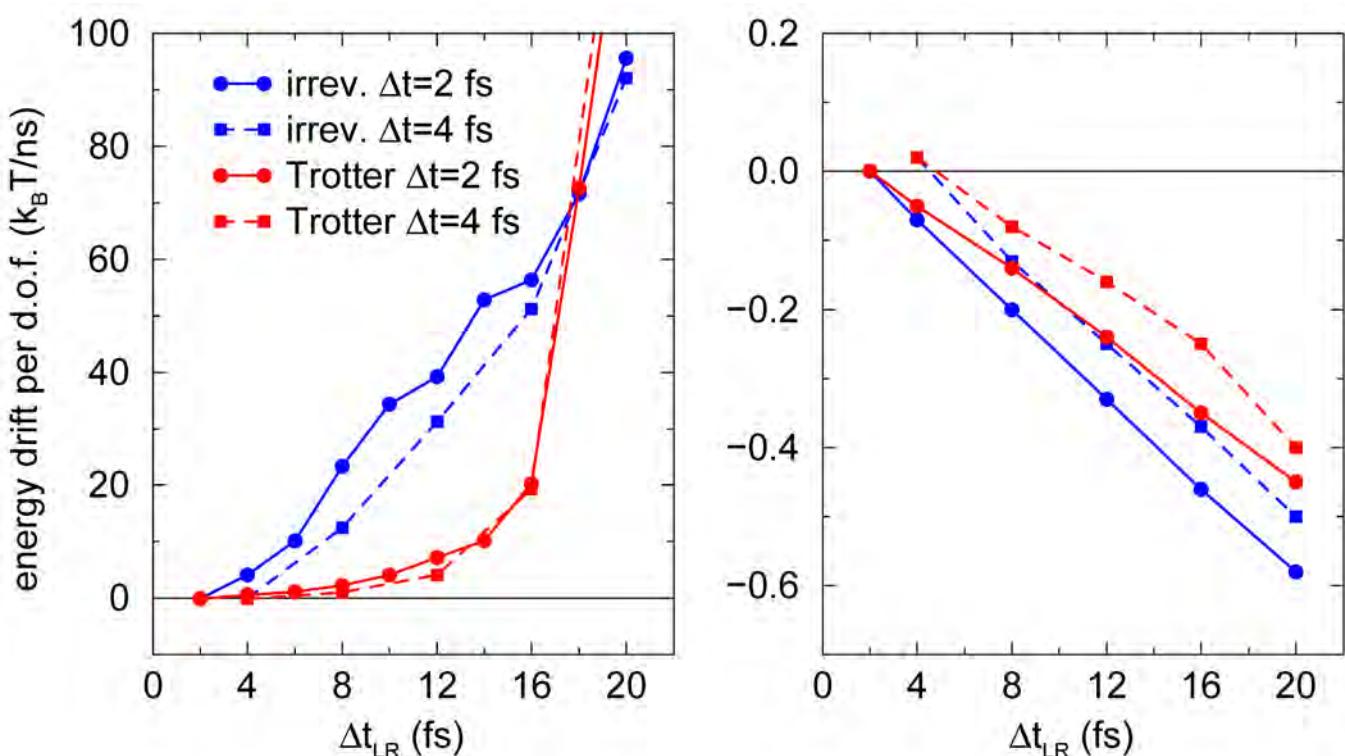


图3.8: SPC/E水模型模拟中每个自由度的能量漂移和长程时间步长的关系, 使用双程截断时, 不可逆的“GROMOS”方案和可逆Trotter方案的对比: (左)反应场; (右) Lennard-Jones 相互作用。

作为一个例子, 我们在如3.8中展示了积分300 K温度下SPC/E水体系运动方程的能量守恒性。为避免截断效应, 使用了反应场, $\epsilon_{RF} = \infty$ 和移位Lennard-Jones 相互作用。二者都使用了缓冲。长程相互作用在1.0和1.4 nm之间进行计算。从图3.7可以看到, 对静电相互作用, 直到 $\Delta t_{LR} = 16$ fs, Trotter 方案的能量守恒性都要好一个数量级。静电相互作用强烈地依赖于水分子变化很快的取向。对Lennard-Jones 相互作用, 能量漂移与 Δt_{LR} 呈线性关系, 大致比静电小两个数量级。Lennard-Jones 力比库仑力小, 它们

主要受水分子的平动影响, 而不是转动.

3.4.8 温度耦合

直接使用的分子动力学模拟对应于NVE系综(粒子数不变, 体积不变, 能量恒定的系综), 我们希望计算的大多数物理量实际上对应于等温(NVT)系综, 也被称为正则系综. GROMACS可以使用Berendsen 的 弱耦合 方案[25], 通过Andersen 恒温器的随机化[26], 扩展系综的Nose-Hoover 方案[27, 28], 或速度重缩放方案[29]来模拟恒温过程, 这些方案各自的优缺点将在下面介绍.

还有其他一些原因使得我们需要对体系的温度进行调控(平衡过程中的漂移, 力截断和积分误差引起的漂移, 外力或摩擦力导致的加热), 但从热力学的观点来看, 这并不是完全正确的. 在某些情况下, 这些做法只是掩盖了问题(增加了体系的温度), 而没有从根本上解决问题(动力学与真实物理过程之间存在的偏差). 对于大的体系, 系综平均引起的误差, 用于消除温度缓慢漂移的控温对结构性质的影响几乎可以忽略. 但如果进行完全的综合比较, 在解释模拟结果时必须谨慎.

Berendsen温度耦合

Berendsen 算法模仿了与给定温度 T_0 的外部热浴相连, 并具有一级动力学特征的弱耦合. 参考文献[30]将它与Nose-Hoover 方案进行了比较. 该算法的效果是, 根据下式慢慢校正体系温度对 T_0 的偏差:

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau} \quad (3.44)$$

这意味着温度偏差指数衰减, 其时间常数为 τ . 这种耦合方法的优点是, 耦合的强度可以改变以适应用户的需要: 对于以平衡为目的模拟可采取很小的时间常数(如0.01 ps); 但对于可靠的平衡模拟, 可采用更大的时间常数(如0.5 ps). 在这种情况下, 它几乎不会影响动力学的守恒性.

Berendsen 恒温可降低动能的涨落. 这意味着, 它不会产生正确的正则系综, 所以严格来说, 采样是不正确的. 引入误差的标度为 $1/N$, 因此对于非常大的体系, 大部分系综平均值不会受到显著影响, 除了动能本身的分布. 然而, 涨落性质, 如热容量, 将受到影响. 后面将要描述的速度重缩放恒温器[29]是一种与此类似的控温方法, 但能产生正确的系综.

流入或流出体系的热量受到对每个粒子的速度进行重新缩放的影响, 缩放每步或每 n_{tc} 步进行一次, 使用一个与时间相关的因子 λ ,

$$\lambda = \left[1 + \frac{n_{TC} \Delta t}{\tau_T} \left\{ \frac{T_0}{T(t - \frac{1}{2} \Delta t)} - 1 \right\} \right] \quad (3.45)$$

参数 τ_T 接近但不完全等于温度耦合的时间常数 τ (方程3.44):

$$\tau = 2C_V \tau_T / N_{df} k \quad (.46)$$

其中 C_V 为体系总的热容, k 为玻尔兹曼常数, N_{df} 为总的自由度数. $\tau \neq \tau_T$ 的原因在于, 缩放速度引起的动能变化会部分地在动能和势能之间进行重新分布, 因此, 温度的变化小于缩放能量. 实际上, 比值 τ/τ_T 的范围从1(气体)到2(简谐固体)到3(水). 当我们使用术语“温度耦合时间常数”时, 我们指的是参数 τ . 注意 缩放因子 λ 在实际中被限制在 $0.8 \leq \lambda \leq 1.25$ 的范围内, 以避免当其过大时导致模拟体系崩溃. 在正常使用中, λ 总是会更接近1.0.

速度重缩放温度耦合

速度重缩放恒温器[29]本质上是一个Berendsen 恒温器(见上文), 并附加了一个随机项, 以保证能够给出正确的动能分布, 使用时根据下面的公式对动能进行修改:

$$dK = (K_O - K) \frac{dt}{\tau_T} + 2 \sqrt{\frac{KK_0}{Nf}} \frac{dW}{\sqrt{\tau_T}} \quad (3.47)$$

其中 K 为动能, N_f 为自由度数, dW 为Wiener过程. 除了随机数种子外, 没有另外的参数. 这种温控器能够产生正确的正则系综, 仍然具有Berendsen 温控器的优点: 温度偏差一阶衰减, 无振荡. 当使用 NVT 系综时, 守恒的能量会被写入到能量文件和log文件.

Andersen温控器

保持恒温系综的一种简单的方式如下, 使用 NVE 积分方法, 并周期性地从麦克斯韦-玻尔兹曼分布中重新选择粒子的速度. [26] 这可以通过每 $\tau_T / \Delta t$ 步同时(大规模碰撞)将所有粒子的速度进行随机化(**andersen-massive**), 或通过每步以小概率, 等于 $\Delta t / \tau$, 随机化每个粒子(**andersen**). 在这两种情况下, Δt 为时间步长, τ_T 为特征的耦合时间尺度. 由于约束的操作方式, 同一约束组中所有粒子的速度必须同时随机化. 由于并行问题, **andersen** 选项目前(5.0)还不能用于含有约束的体系. **andersen-massive** 的使用不受约束限制. 此温控器也是目前唯一可与速度Verlet 算法一同使用的, 因为它在每个时间步的直接对速度进行操作.

这个算法完全避免了其他恒温算法具有的一些遍历性问题, 像能量不能在体系的能量去耦合组分间来回流动, 如在速度缩放运动中. 但是, 它可以通过随机化体系的相关运动降低体系的动能, 包括减慢采样, 当 τ_T 处于中等水平时(小于10 ps). 因此, 该算法通常不能用于考察体系的动力学或运输性质. [31]

Nose-Hoover温度耦合

Berendsen 弱耦合算法可以非常高效地将体系弛豫到到目标温度, 但是, 一旦系统达到平衡, 维持正确的正则系综更重要. 很可惜, 弱耦合方案并不能满足这一点.

为了使用正则系综模拟, GROMACS还支持扩展系综方法. 这种方法首先由Nose提出[27], 后经Hoover 修改[28]. 通过在运动方程中引入一个热容器和一个摩擦项, 方法对体系的哈密顿量进行了扩展. 摩擦力正比于每个粒子的速度和摩擦参数 ξ 的乘积. 这个摩擦参数(或“热浴”变量)是一个完全的动力学量, 有着自己是动量(p_ξ)和运动方程; 其时间导数由当前动能和参考温度之间的差值来计算.

在这个公式中, 图3.3中粒子的运动方程被替换为:

$$\frac{d^2 \mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m_i} - \frac{p_\xi}{Q} \frac{d\mathbf{r}_i}{dt} \quad (3.48)$$

其中热浴参数 ξ 的运动方程为:

$$\frac{dp_\xi}{dt} = (T_0 - T) \quad (3.49)$$

参考温度以 T_0 表示, 而 T 为体系当前的瞬时温度. 耦合强度由常数 Q (通常被称为容器的“质量参数”)和参考温度共同决定[1]

Nose-Hoover 运动方程的守恒量并不是总能量, 而是

$$H = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) + \frac{p_\xi^2}{2Q} + N_f kT\xi \quad (3.50)$$

其中 N_f 为总的自由度数.

依我们看, 利用质量参数来描述耦合强度让人有点难以理解, 特别是它还依赖于参考温度(在一些实现中 Q 的定义还和体系的自由度数有关). 为了维持耦合强度, Q 的改变必须正比于参考温度的改变. 基于这个原因, 我们更愿意让GROMACS用户使用体系和容器之间动能的振荡周期 τ_T 来代替 Q , 它通过下式直接与 Q 和 T_0 相关:

$$Q = \frac{\tau_T^2 T_0}{4\pi^2} \quad (3.51)$$

这为选择Nose-Hoover 耦合强度(弱耦合弛豫与此类似)提供了一个更为直观的方式, 并且 τ_T 与体系的大小和参考温度无关.

然而, 要特别注意弱耦合方案和Nose-Hoover 算法的区别: 使用弱耦合你将得到一个具有强烈阻尼的 指数弛豫, 而Nose-Hoover 方法产生 振荡弛豫. Nose-Hoover 耦合需要的实际弛豫时间 比你选择的振荡周期要大几倍. 这些振荡(相比于指数弛豫)也意味着时间常数通常应该比弱耦合所使用的弛豫时间大4–5倍, 但你可以根据情况改变.

Nose-Hoover 动力学在简单的体系, 如谐振子集合中, 可能是非遍历的, 这意味着即使模拟运行了无限长时间, 也只能对相空间的一小部分进行采样. 因此, 人们发展了Nose-Hoover 链方法, 其中每个的Nose-Hoover 控温器都有自己的Nose-Hoover 恒温器来控制其温度. 当恒温器链的长度趋向无限时, 可以保证过程是遍历的. 只使用几个链可以大大提高遍历性, 但最近的研究表明, 体系仍然是非遍历的, 并且仍然不完全清楚其实际影响[32]. 目前, 链的默认数目为10, 但用户可以更改. 在使用Nose-Hoover 链的情况下, 方程修改为下面的形式以包含恒温粒子链[33]:

$$\begin{aligned} \frac{d^2 \mathbf{r}_i}{dt^2} &= \frac{\mathbf{F}_i}{m_i} - \frac{p_{\xi_1}}{Q_1} \frac{d\mathbf{r}_i}{dt} \\ \frac{dp_{\xi_1}}{dt} &= (T - T_0) - p_{\xi_1} \frac{p_{\xi_2}}{\xi_2} \\ \frac{dp_{\xi_{i-2\dots N}}}{dt} &= \left(\frac{p_{\xi_{i-1}}^2}{Q_{i-1}} - kT \right) - p_{\xi_i} \frac{p_{\xi_{i+1}}}{Q_{i+1}} \\ \frac{dp_{\xi_N}}{dt} &= \left(\frac{p_{\xi_{N-1}}^2}{Q_{N-1}} - kT \right) \end{aligned} \quad (3.52)$$

Nose-Hoover 链的守恒量为

$$H = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) + \sum_{k=1}^M \frac{p_{\xi_k}^2}{2Q'_k} + N_f kT \xi_1 + kT \sum_{k=2}^M \xi_k \quad (3.53)$$

Nose-Hoover 恒温器变量的值与速度一般不包含在输出中, 因为它们会占用相当大的空间并且通常对于分析模拟结果不是很重要. 但是, 可以通过定义环境变量 `GMX_NOSEHOOVER_CHAINS`, 将链中所有Nose-Hoover 粒子的位置和速度输出到 `.edr` 文件中. 在目前的版本中, 蛙跳式积分方法只能使用长度为1的Nose-Hoover 链, 但在以后的版本中可能会支持更长的链.

如在积分方法那节讲的, 对于温度耦合, 速度Verlet方法和蛙跳式算法在试图匹配参考温度时, 对温度的计算有所不同. 速度Verlet(`md-vv`)使用了整步动能, 而蛙跳式缩放和 `md-vv-avek` 使用了半步平均的动能.

通过再次考查Trotter 分解, 我们可以更好地理解这些等温积分方法之间的差异. 对Nose-Hoover 动力学(为简单起见, 使用的链长 $N = 1$, 更多细节请参考文献[34]), 将Liouville 算子劈分为

$$iL = iL_1 + iL_2 + iL_{\text{NHC}} \quad (3.54)$$

其中,

$$\begin{aligned} iL_1 &= \sum_{i=1}^N \left[\frac{\mathbf{p}_i}{m_i} \right] \cdot \frac{\partial}{\partial \mathbf{r}_i} \\ iL_2 &= \sum_{i=1}^N \mathbf{F}_i \cdot \frac{\partial}{\partial \mathbf{p}_i} \\ iL_{\text{NHC}} &= \sum_{i=1}^N -\frac{p_\xi}{Q} \mathbf{v}_i \cdot \nabla \mathbf{v}_i + \frac{p_\xi}{Q} \frac{\partial}{\partial \xi} + (T - T_0) \frac{\partial}{\partial p_\xi} \end{aligned} \quad (3.55)$$

对于使用Nose-Hoover 温度控制的标准速度Verlet方法, 上式变为

$$\begin{aligned}\exp(iL\Delta t) &= \exp(iL_{\text{NHC}}\Delta t/2)\exp(iL_2\Delta t/2) \\ &\quad \exp(iL_1\Delta t)\exp(iL_2\Delta t/2)\exp(iL_{\text{NHC}}\Delta t/2)+O(\Delta t^3)\end{aligned}\quad (3.56)$$

对于使用 **md-vv-avek** 的半步平均温度控制, 这种分解不适用, 因为我们在第二速度步完成前不可能得到整步的温度. 然而, 我们可以通过交换分解中NHC和速度部分的位置构造另一个分解, 它仍然是可逆的,

$$\begin{aligned}\exp(iL\Delta t) &= \exp(iL_2\Delta t/2)\exp(iL_{\text{NHC}}\Delta t/2)\exp(iL_1\Delta t) \\ &\quad \exp(iL_{\text{NHC}}\Delta t/2)\exp(iL_2\Delta t/2)+O(\Delta t^3)\end{aligned}\quad (3.57)$$

利用这种形式我们可以很容易地看出各种速度Verlet积分方法之间的差异. 蛙跳式积分方法可视为以方程3.57 $\exp(iL_1\Delta t)$ 之前的项开始, 得到:

$$\begin{aligned}\exp(iL\Delta t) &= \exp(iL_1\Delta t)\exp(iL_{\text{NHC}}\Delta t/2) \\ &\quad \exp(iL_2\Delta t)\exp(iL_{\text{NHC}}\Delta t/2)+O(\Delta t^3)\end{aligned}\quad (3.58)$$

然后使用了一些代数技巧来求解, 因为有一些量在实际计算之前就需要知道[35].

组内温度耦合

在GROMACS中, 温度耦合可用于原子组, 通常是蛋白质和溶剂. 引入这种算法的原因是, 由于各种不同的效应, 包括截断等等, 不同组分之间的能量交换并不完美. 如果现在整个体系耦合到一个热浴, 水(受到的截断噪声最大)将倾向于升温, 而蛋白质会冷却. 这通常会导致二者之间有100 K的温度差异. 通过使用合适的静电方法(PME), 这些差异会小很多, 但仍然不可忽视. 组内温度耦合的参数在 **mdp** 文件中给出. 最近的研究表明, 蛋白质和水之间细微的温度差异, 实际上可能是虚假的, 是由时间步长有限时计算温度的方式导致的, 非常大的温度差异很可能是体系严重出错的标志, 应该仔细检查[36].

应该提到一种特殊情况: 可以只对体系的一部分进行温度耦合, 而其它部分不进行温度耦合. 这通过对那些不进行恒温的组指定-1的时间常数 τ_T 来实现. 如果只有体系的一部分进行恒温, 体系最终仍然会收敛到一个NVT体系. 实际上, 有一个建议可尽量降低离散时间步长所造成的温度误差: 如果对水使用了约束, 那么应该只对水的自由度进行恒温, 而不是蛋白质的自由度, 因为蛋白质更高频率的模式可能会导致温度与“真实”温度有更大的偏差, 这里的真实温度指的是使用小的时间步长得到的温度[36].

3.4.9 压力耦合

与温度耦合类似, 体系也可以耦合到一个“压力浴”. GROMACS 既支持每步重新缩放坐标与盒矢量的Berendsen 算法[25], 扩展系统的Parrinello-Rahman 方法[37, 38], 也支持速度Verlet的一种变形, Martyna-Tuckerman-Tobias-Klein (MTTK) 方法的压力控制[34]. Parrinello-Rahman 和Berendsen 方法可以与任何上述的温度耦合方法联用; MTTK 只能与Nose-Hoover 温度控制方法联用.

Berendsen压力耦合

Berendsen 算法会利用矩阵 μ 重新缩放坐标和盒矢量, 每步或每 n_{PC} 步一次, 压力向给定参考压力 P_0 的弛豫符合一级动力学特征

$$\frac{d\mathbf{P}}{dt} = \frac{\mathbf{P}_0 - \mathbf{P}}{\tau_p} \quad (3.59)$$

缩放矩阵 μ 由下式给出

$$\mu_{ij} = \delta_{ij} - \frac{n_{\text{PC}}\Delta t}{3\tau_p}\beta_{ij}\{P_{0ij} - P_{ij}(t)\} \quad (3.60)$$

这里, β 为体系的等温压缩系数. 大多数情况下, 它是一个对角矩阵, 且对角线上的值相等, 这个值通常是未知的, 但完全可以采取一

个粗略的估计值, 因为 β 的值仅影响压力弛豫的非临界时间常数, 而不影响平均压力本身. 1个标准大气压, 300 K条件下, 水的 $\beta = 4.6 \times 10^{-10} \text{ Pa}^{-1} = 4.6 \times 10^{-5} \text{ bar}^{-1}$, 相应于 7.6×10^{-4} MD单位(参见第二章). 大多数其它的液体具有与此相近的值. 当缩放具有完全的各向异性时, 必须旋转体系使其服从方程 3.1. 缩放时, 旋转采用一阶近似, 通常小于 10^{-4} . 实际的缩放矩阵 μ' 为

$$\mu' = \begin{pmatrix} \mu_{xx} & \mu_{xy} + \mu_{yx} & \mu_{xz} + \mu_{zx} \\ 0 & \mu_{yy} & \mu_{yz} + \mu_{zy} \\ 0 & 0 & \mu_{zz} \end{pmatrix} \quad (3.61)$$

速度既不缩放, 也不旋转.

在GROMACS中, Berendsen 缩放也可以各向同性地进行, 这意味着, 使用对角元素为 $\text{trace}(\mathbf{P})/3$ 的对角矩阵代替 \mathbf{P} . 对存在界面的体系, 半各向同性缩放可能有用. 在这种情况下, 在 x/y 方向上进行各向同性缩放, 而在 z 方向上进行独立的缩放. 将 x/y 或 z 方向上的压缩系数可以设置为零, 以便只在另一方向上进行缩放.

如果允许完全的各向异性变形并使用约束, 你可能必须使用更缓慢的缩放, 或是减少时间步长以避免约束算法导致的误差. 需要注意的是, 尽管Berendsen 控压算法在模拟时能产生正确的平均压力, 但它不能得到精确的NPT系综, 而且目前尚不完全清楚这种方法可能导致的误差.

Parrinello-Rahman压力耦合

若压力或体积的涨落就其自身而言非常重要(例如, 计算热力学性质), 特别是对于小的体系, 弱耦合方案可能存在一个问题, 这种方案没有很好地定义精确的系综, 模拟的并不是真正的NPT系综.

GROMACS也支持使用Parrinello-Rahman 方法[37, 38]的等压模拟, 这种方法类似于Nose-Hoover 温控方法, 理论上能给出真正NPT系综. 使用Parrinello-Rahman 恒压器, 以矩阵 \mathbf{b} 表示的盒矢量服从矩阵运动方程[2]

$$\frac{d\mathbf{b}^2}{dt^2} = V\mathbf{W}^{-1}\mathbf{b}'^{-1}(\mathbf{P} - \mathbf{P}_{ref}) \quad (3.62)$$

其中 V 代表盒子的体积, \mathbf{W} 为决定耦合强度的矩阵参数. 矩阵 \mathbf{P} 和 \mathbf{P}_{ref} 分别是当前压力和参考压力.

和Nose-Hoover 耦合一样, 粒子的运动方程也改变了. 在大多数情况下, 你将联合使用Parrinello-Rahman 恒压器与Nose-Hoover 恒温器, 为简单起见, 我们这里只给出Parrinello-Rahman 导致的变化:

$$\frac{d\mathbf{r}_i^2}{dt^2} = \frac{\mathbf{F}_i}{m_i} - M \frac{d\mathbf{r}_i}{dt} \quad (3.63)$$

$$\mathbf{M} = \mathbf{b}^{-1} \left[\mathbf{b} \frac{d\mathbf{b}'}{dt} + \frac{d\mathbf{b}}{dt} \mathbf{b}' \right] \mathbf{b}'^{-1} \quad (3.64)$$

(逆)质量参数矩阵 \mathbf{W}^{-1} 决定了耦合的强度, 以及盒子如何变形. 如果 \mathbf{W}^{-1} 的对应元素为0, 盒子的限制条件(3.1)会自动满足. 由于耦合强度也取决于盒子的大小, 我们更愿意在GROMACS中自动计算盒子的大小. 你只需要在输入文中提供近似的等温压缩系数 β 和压力的时间常数 τ_P (L 为盒矩阵元素的最大值):

$$(\mathbf{W}^{-1})_{ij} = \frac{4\pi^2 \beta_{ij}}{3\tau_p^2 L} \quad (3.65)$$

正如在Nose-Hoover 恒温器, 你应该认识到, Parrinello-Rahman 时间常数并不等同于Berendsen 压力耦合算法中使用的弛豫时间. 大多数情况下, 对于Parrinello-Rahman 耦合, 你需要使用4–5倍大的时间常数. 如果压力离平衡态非常远, Parrinello-Rahman 耦合会导致盒子产生非常大的振荡甚至最终导致运行崩溃. 在这种情况下, 你可能必须增加时间常数, 或(更好的办法)先使用弱耦合

方案达到目标压力, 等系统平衡后再切换到Parrinello-Rahman 耦合. 此外, 使用蛙跳式算法时, 在整个时间步骤完成之前不能得到 t 时刻的压力, 因此必须使用上一步中的压力, 这使得算法不是直接可逆的, 并且可能不适合进行高精度的热力学计算.

表面张力耦合

当周期性体系中包含一个以上的相, 且这些相被平行于 xy 表面的表面所隔离时, 表面张力和压力的 z 分量会与压力沿相耦合. 目前, 这只适用于GROMACS中的Berendsen 压力耦合算法. 平均表面张力 $\gamma(t)$ 可以通过法向压力与横向压力之间的差值来计算

$$\gamma(t) = \frac{1}{n} \int_0^{L_z} \left\{ P_{zz}(z, t) - \frac{P_{xx}(z, t) + P_{yy}(z, t)}{2} \right\} dz \quad (3.66)$$

$$= \frac{L_z}{n} \left\{ P_{zz}(t) - \frac{P_{xx}(t) + P_{yy}(t)}{2} \right\} \quad (3.67)$$

其中 L_z 为盒子的高度, n 为表面的数目. z 方向的压力通过使用 μ_{zz} 缩放盒子的高度进行校正

$$\Delta P_{zz} = \frac{\Delta t}{\tau_p} \{ P_{0zz} - P_{zz}(t) \} \quad (3.68)$$

$$\mu_{zz} = 1 + \beta_{zz} \Delta P_{zz} \quad (3.69)$$

除缺少 $1/3$ 这个因子外, 法向压力耦合的校正与此类似. z 方向上的压力校正, 被用于得到表面张力相对于参考值 γ_0 的正确收敛值. x/y 方向盒子长度的校正因子为

$$\mu_{x/y} = 1 + \frac{\Delta t}{2\tau_p} \beta_{x/y} \left(\frac{n\gamma_0}{\mu_{zz} L_z} - \left\{ P_{zz}(t) + \Delta P_{zz} - \frac{P_{xx}(t) + P_{yy}(t)}{2} \right\} \right) \quad (3.70)$$

相比法向压力耦合, β_{zz} 的值更关键. 正常情况下不正确的压缩系数仅仅缩放 τ_p , 但对表面张力耦合还会影响表面张力的收敛性. 当 b_{zz} 设为零时(盒子的高度不变), ΔP_{zz} 也设为零, 以获得正确的表面张力.

MTTK压力控制算法

正如上一节所讲, 蛙跳式积分方法一个缺点在于恒压模拟, 因为计算压力需要整个时间步内的维里和动能, 对蛙跳式积分方法, 这些信息直到整个时间步完成之后才能得到.

速度Verlet方法确实可以计算, 但需要额外的全局通讯作为代价, 并计算真正的NPT系综, 忽略任何积分误差.

综合了压力耦合和温度耦合的完整方程, 这里称其为MTTK方程(Martyna-Tuckerman-Tobias-Klein), 取自Martyna等[34]和Tuckerman[39]的论文, 为方便, 我们引入 $\epsilon = (1/3) \ln(V/V_0)$, 其中 V_0 为参考体积. ϵ 的动量为 $v_\epsilon = p_\epsilon/W = \dot{\epsilon} = \dot{V}/3V$, 并定义 $\alpha = 1 + 3/N_{dof}$ (参见参考文献[39])

等压方程为

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} + \frac{p_\epsilon}{W} \mathbf{r}_i \quad (3.71)$$

$$\frac{\dot{\mathbf{p}}_i}{m_i} = \frac{1}{m_i} \mathbf{F}_i - \alpha \frac{p_\epsilon}{W} \frac{\mathbf{p}_i}{m_i} \quad (3.72)$$

$$\dot{\epsilon} = \frac{p_\epsilon}{W} \quad (3.72)$$

$$\frac{\dot{p}_\epsilon}{W} = \frac{3V}{W} (P_{\text{kin}} - P) + (\alpha - 1) \left(\sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} \right)$$

其中

$$P_{\text{int}} = P_{\text{kin}} - P_{\text{vir}} = \frac{1}{3V} \left[\sum_{i=1}^N \left(\frac{\mathbf{p}_i^2}{2m_i} - \mathbf{r}_i \cdot \mathbf{F}_i \right) \right] \quad (3.73)$$

包含 α 的项可保证相空间不可压缩[39]. ϵ 加速度项可以改写为

$$\frac{\dot{p}_\epsilon}{W} = \frac{3V}{W} (\alpha P_{\text{kin}} - P_{\text{vir}} - P) \quad (3.74)$$

对速度项, 这些方程变为

$$\begin{aligned} \dot{\mathbf{r}}_i &= \mathbf{v}_i + v_\epsilon \mathbf{r}_i \\ \dot{\mathbf{v}}_i &= \frac{1}{m_i} \mathbf{F}_i - \alpha v_\epsilon \mathbf{v}_i \\ \dot{\epsilon} &= v_\epsilon \\ \dot{v}_\epsilon &= \frac{3V}{W} (P_{\text{int}} - P) + (\alpha - 1) \left(\sum_{i=1}^N \frac{1}{2} m_i \mathbf{v}_i^2 \right) \\ P_{\text{int}} &= P_{\text{kin}} - P_{\text{vir}} = \frac{1}{3V} \left[\sum_{i=1}^N \left(\frac{1}{2} m_i \mathbf{v}_i^2 - \mathbf{r}_i \cdot \mathbf{F}_i \right) \right] \end{aligned} \quad (3.75)$$

对于这些方程, 守恒量为

$$H = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) + \frac{p_\epsilon}{2W} + PV \quad (3.76)$$

下一步是添加温度控制, 添加Nose-Hoover 链, 并将其添加到恒压器自由度. 令 η 为恒压器的Nose-Hoover 变量, Q' 为恒压器的恒温器常数, 我们得到

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} + \frac{p_\epsilon}{W} \mathbf{r}_i \\ \dot{\mathbf{p}}_i &= \frac{1}{m_i} \mathbf{F}_i - \alpha \frac{p_\epsilon}{W} \frac{\mathbf{p}_i}{m_i} - \frac{p_{\xi_1}}{Q_1} \frac{\mathbf{p}_i}{m_i} \\ \dot{\epsilon} &= \frac{p_\epsilon}{W} \\ \frac{\dot{p}_\epsilon}{W} &= \frac{3V}{W} (\alpha P_{\text{kin}} - P_{\text{vir}} - P) - \frac{p_{\eta_1}}{Q'_1} p_\epsilon \\ \dot{\xi}_k &= \frac{p_{\xi_k}}{Q_k} \\ \dot{\eta}_k &= \frac{p_{\eta_k}}{Q'_k} \\ \dot{p}_{\xi_k} &= G_k - \frac{p_{\xi_{k+1}}}{Q_{k+1}} \quad k = 1, \dots, M-1 \\ \dot{p}_{\eta_k} &= G'_k - \frac{p_{\eta_{k+1}}}{Q'_{k+1}} \quad k = 1, \dots, M-1 \\ \dot{p}_{\xi_M} &= G_M \\ \dot{p}_{\eta_M} &= G'_M \end{aligned} \quad (3.77)$$

其中

$$\begin{aligned}
P_{\text{int}} &= P_{\text{kin}} - P_{\text{vir}} = \frac{1}{3V} \left[\sum_{i=1}^N \left(\frac{\mathbf{p}_i^2}{2m_i} - \mathbf{r}_i \cdot \mathbf{F}_i \right) \right] \\
G_1 &= \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} - N_f kT \\
G_k &= \frac{p_{\xi_{k-1}}^2}{2Q_{k-1}} - kT \quad k = 1, \dots, M \\
G'_1 &= \frac{p_\epsilon^2}{2W} - kT \\
G'_k &= \frac{p_{\eta_{k-1}}^2}{2Q'_{k-1}} - kT \quad k = 1, \dots, M
\end{aligned} \tag{3.78}$$

现在的守恒量为

$$\begin{aligned}
H &= \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + U(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) + \frac{p_\epsilon^2}{2W} + PV \\
&\quad + \sum_{k=1}^M \frac{p_{\xi_k}^2}{2Q_k} + \sum_{k=1}^M \frac{p_{\eta_k}^2}{2Q'_k} + N_f kT \xi_1 + kT \sum_{k=2}^M \xi_k + \sum_{k=2}^M \eta_k
\end{aligned} \tag{3.79}$$

回到Trotter 分解形式, 对压力控制和温度控制[34]我们得到:

$$iL = iL_1 + iL_2 + iL_{\epsilon,1} + iL_{\epsilon,2} + iL_{\text{NHC-baro}} + iL_{\text{NHC}} \tag{3.80}$$

其中, “NHC-baro” 对应于恒压器的Nose-Hoover 链, NHC对应于粒子的NHC

$$iL_1 = \sum_{i=1}^N \left[\frac{\mathbf{p}_i}{m_i} + \frac{p_\epsilon}{W} \mathbf{r}_i \right] \cdot \frac{\partial}{\partial \mathbf{r}_i} \tag{3.81}$$

$$iL_2 = \sum_{i=1}^N \mathbf{F}_i - \alpha \frac{p_\epsilon}{W} \mathbf{p}_i \cdot \frac{\partial}{\partial \mathbf{p}_i} \tag{3.82}$$

$$iL_{\epsilon,1} = \frac{p_\epsilon}{W} \frac{\partial}{\partial \epsilon} \tag{3.83}$$

$$iL_{\epsilon,2} = G_\epsilon \frac{\partial}{\partial p_\epsilon} \tag{3.84}$$

其中

$$G_\epsilon = 3V(\alpha P_{\text{kin}} - P_{\text{vir}} - P) \tag{3.85}$$

用Trotter 分解, 我们得到

$$\begin{aligned}
\exp(iL\Delta t) &= \exp(iL_{\text{NHC-baro}}\Delta t/2) \exp(iL_{\text{NHC}}\Delta t/2) \\
&\quad \exp(iL_{\epsilon,2}\Delta t/2) \exp(iL_2\Delta t/2) \\
&\quad \exp(iL_{\epsilon,1}\Delta t) \exp(iL_1\Delta t) \\
&\quad \exp(iL_2\Delta t/2) \exp(iL_{\epsilon,2}\Delta t/2) \\
&\quad \exp(iL_{\text{NHC}}\Delta t/2) \exp(iL_{\text{NHC-baro}}\Delta t/2) + O(\Delta t^3)
\end{aligned} \tag{3.86}$$

$\exp(iL_1\Delta t)$ 作用来自于微分方程 $\dot{\mathbf{r}}_i = \mathbf{v}_i + v_\epsilon \mathbf{r}_i$ 的解, 其中 $\mathbf{v}_i = \mathbf{p}_i/m_i$, v_ϵ 为常数, 初始条件为 $\mathbf{r}_i(0)$, 在 $t = \Delta t$ 时刻进行计算. 这样可以得到

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i(0)e^{v_\epsilon \Delta t} + \Delta t \mathbf{v}_i(0)e^{v_\epsilon \Delta t/2} \frac{\sinh(v_\epsilon \Delta t/2)}{v_\epsilon \Delta t/2} \tag{3.87}$$

$\exp(iL_2 \Delta t / 2)$ 作用来自于微分方程 $\dot{\mathbf{v}}_i = \frac{\mathbf{F}_i}{m_i} - \alpha v_\epsilon \mathbf{v}_i$ 的解, 得到

$$\mathbf{v}_i(\Delta t / 2) = \mathbf{v}_i(0) e^{-\alpha v_\epsilon \Delta t / 2} + \frac{\Delta t}{2m_i} \mathbf{F}_i(0) e^{-\alpha v_\epsilon \Delta t / 4} \frac{\sinh(\alpha v_\epsilon \Delta t / 4)}{\alpha v_\epsilon \Delta t / 4} \quad (3.88)$$

md-vv-avek 使用完整步的动能来确定压力控制的压力, 但使用半步平均动能来确定温度, 这可以写为 Trotter 分解

$$\begin{aligned} \exp(iL\Delta t) &= \exp(iL_{\text{NHC-baro}} \Delta t / 2) \exp(iL_{\epsilon,2} \Delta t / 2) \exp(iL_2 \Delta t / 2) \\ &\quad \exp(iL_{\text{NHC}} \Delta t / 2) \exp(iL_{\epsilon,1} \Delta t) \exp(iL_1 \Delta t) \exp(iL_{\text{NHC}} \Delta t / 2) \\ &\quad \exp(iL_2 \Delta t / 2) \exp(iL_{\epsilon,2} \Delta t / 2) \exp(iL_{\text{NHC-baro}} \Delta t / 2) + O(\Delta t^3) \end{aligned} \quad (3.89)$$

使用约束时, 这些方程会变得更为复杂, 对约束力需要迭代求解每个方程. 对迭代细节的讨论超出了本手册的范围; 我们鼓励读者参看文献[40]中的实现.

温度和压力耦合的不频繁计算

温度和压力的控制需要全局通信来计算的动能和维里, 对大的体系, 如果在每一步都进行会使计算变得十分耗时. 我们可以重排 Trotter 分解, 给出另一种可逆的辛积分方法, 每 n 步进行一次而不是每步一次. 当耦合时间步长过大时, 这些新的积分方法将发散, 因为辅助变量积分不收敛. 然而, 在大多数情况下, 长的耦合时间更合适, 因为他们对动力学的扰动小些[34].

使用 Nose-Hoover 温度控制时, 标准的速度 Verlet 方法有如下的 Trotter 展开

$$\begin{aligned} \exp(iL\Delta t) &\approx \exp(iL_{\text{NHC}} \Delta t / 2) \exp(iL_2 \Delta t / 2) \\ &\quad \exp(iL_1 \Delta t) \exp(iL_2 \Delta t / 2) \exp(iL_{\text{NHC}} \Delta t / 2) \end{aligned} \quad (3.90)$$

如果相对于系统运动, Nose-Hoover 链的运动足够慢, 对速度 Verlet 方法, 我们可以写出一个进行 n 步的积分方法

$$\begin{aligned} \exp(iL\Delta t) &\approx \exp(iL_{\text{NHC}}(n\Delta t / 2)) [\exp(iL_2 \Delta t / 2) \\ &\quad \exp(iL_1 \Delta t) \exp(iL_2 \Delta t / 2)]^n \exp(iL_{\text{NHC}}(n\Delta t / 2)) \end{aligned} \quad (3.91)$$

对于压力控制, 这成为

$$\begin{aligned} \exp(iL\Delta t) &\approx \exp(iL_{\text{NHC-baro}}(n\Delta t / 2)) \exp(iL_{\text{NHC}}(n\Delta t / 2)) \\ &\quad \exp(iL_{\epsilon,2}(n\Delta t / 2)) [\exp(iL_2 \Delta t / 2) \\ &\quad \exp(iL_{\epsilon,1} \Delta t) \exp(iL_1 \Delta t) \\ &\quad \exp(iL_2 \Delta t / 2)]^n \exp(iL_{\epsilon,2}(n\Delta t / 2)) \\ &\quad \exp(iL_{\text{NHC}}(n\Delta t / 2)) \exp(iL_{\text{NHC-baro}}(n\Delta t / 2)) \end{aligned} \quad (3.92)$$

其中对盒子体积的积分每步都进行, 但对辅助变量的积分每 n 步一次.

3.4.10 完整的更新算法

使用蛙跳式积分方法时, 更新速度和坐标的完整算法见图3.9. 步骤4的SHAKE 算法在下面进行说明.

GROMACS具有“冻结”(防止运行)选中粒子的规定, 这些粒子必须被定义为“冻结组”. 这是使用 冻结因子 f_g 实现的, 冻结因子时一个向量, 对每个冷冻组都不相同(参见3.3节). 此向量只包含零(冻结)或1(不冻结). 当我们考虑冻结因子和外部加速度 a_h 时, 速度的更新算法变为

$$\mathbf{v}(t + \frac{\Delta t}{2}) = f_g * \lambda * \left[\mathbf{v}(t - \frac{\Delta t}{2} + \frac{\mathbf{F}(t)}{m} \Delta t + a_h \Delta t) \right] \quad (3.93)$$

其中 g 和 h 为组索引, 每个原子都不同.

更新算法

给出:

t 时刻所有原子的位置 r

$t - \frac{1}{2}\Delta t$ 时刻所有原子的速度 v

t 时刻所有原子上的加速的 F/m

(计算力时忽略所有约束)

$t - \Delta t$ 时刻总的动能和维里



1. 根据方程3.45和3.60计算缩放因子 λ 和 μ



2. 更新和缩放速度: $v' = \lambda(v + a\Delta t)$



3. 计算未约束的新坐标: $r' = r + v'\Delta t$



4. 对坐标使用约束算法: $\text{constrain}(r' \rightarrow r''); r''$



5. 校正约束速度: $v = (r'' - r) / \Delta t$



6. 缩放坐标和盒子: $r = \mu r''; b = \mu b$

图3.9: 蛙跳式积分方法对应的MD更新算法

3.4.11 输出步

MD运行的最重要输出是 轨迹文件, 它包含了规则间隔时刻粒子的坐标和(可选的)速度. 轨迹文件包含许多帧, 每帧可能包括位置, 速度和/或力, 以及模拟体积尺寸, 积分步, 积分时间等信息. 时间的解释与所选择的积分方法有关, 如上所述. 对速度Verlet积分方法, 标记时刻 t 的速度是那个时刻的速度. 对于其他积分方法(如蛙跳式, 随机动力学), 标记时刻 t 的速度是 $t - \frac{1}{2}\Delta t$ 时刻的速度.

由于轨迹文件很长, 不要每一步都保存! 为保留所有信息, 每15步输出一帧就够了, 因为对体系中最大频率的每一周期至少使用了30步, 并且Shannon 的采样定理指出, 对带限信号, 最大频率的每一周期进行两次采样就包含了所有可用的信息. 但这样做仍然会得到很大的文件! 所以, 如果对最高频率不感兴趣, 每ps进行10或20此采样就够了. 注意由频闪效应导致的高频运动失真, 这被称为混叠: 较高的频率相对采样频率出现镜像, 显得频率较低.

GROMACS也可以将模拟体系一部分坐标的精度进行降低, 并写入到特殊的压缩格式的轨迹文件. 所有其他的工具可以读取和写入这种格式. 参见7.3节关于如何设置 `.mdp` 文件以便使用 `mdrun` 此功能的详细信息.

3.5 壳层分子动力学

GROMACS可以使用Dick 和Overhauser 的壳层模型来模拟极化率[41]. 在这个模型中, 表示电子自由度的壳层粒子通过弹性势连接

到原子核上. 在模拟的每一步, 都会对相对于壳层位置的势能进行最小化(见后面). GROMACS中的壳层模型已经成功用于N₂[42]和水[43].

3.5.1 优化壳层位置

壳层粒子 S 上的力 \mathbf{F}_S 可分解为两部分,

$$\mathbf{F}_S = \mathbf{F}_{bond} + \mathbf{F}_{nb} \quad (3.94)$$

其中 \mathbf{F}_{bond} 代表表示极化能量的分量, 通常以简谐势代替, \mathbf{F}_{nb} 为库仑与van der Waals 相互作用的加和. 如果我们假定 \mathbf{F}_{nb} 几乎为常数, 就可以解析地导出壳层的最佳位置, 即 $b_i F_S = 0$ 的位置. 若壳层S连接到原子A:

$$\mathbf{F}_{bond} = k_b(\mathbf{x}_S - \mathbf{x}_A) \quad (3.95)$$

在迭代求解过程中, 设 n 次迭代的位置为 $\mathbf{x}_S(n)$, 则在第 n 次迭代中,

$$\mathbf{F}_{nb} = \mathbf{F}_S - k_b(\mathbf{x}_S(n) - \mathbf{x}_A) \quad (3.96)$$

由此壳层的最佳位置符合下面的关系

$$\mathbf{F}_S - k_b(\mathbf{x}_S(n) - \mathbf{x}_A) + k_b(\mathbf{x}_S(n+1) - \mathbf{x}_A) = 0 \quad (3.97)$$

若记

$$\Delta\mathbf{x}_S = \mathbf{x}_S(n+1) - \mathbf{x}_S(n) \quad (3.98)$$

最终得到

$$\Delta\mathbf{x}_S = \mathbf{F}_S / k_b \quad (3.99)$$

这样就得到了优化壳层位置时计算下一步尝试的算法

$$\mathbf{x}_S(n+1) = \mathbf{x}_S(n) + \mathbf{F}_S / k_b \quad (3.100)$$

3.6 约束算法

在GROMACS中可以使用LINCS(默认)或传统的SHAKE方法来施加约束.

3.6.1 SHAKE

SHAKE算法[44]将一组不受约束的坐标 \mathbf{r}' 转变为一组坐标 \mathbf{r}'' , 使其满足一系列距离约束条件, 使用一组 \mathbf{r} 作为参考,

$$\text{SHAKE}(\mathbf{r}' \rightarrow \mathbf{r}''; \mathbf{r}) \quad (3.101)$$

算法所起的作用与求解一组约束运动方程的Lagrange 乘子相同. SHAKE需要指定 相对容差. 算法会继续直到所有的约束都处于相对容差之内. 若由于偏差太大, SHAKE不能重置坐标, 或迭代次数超过指定数目, 会给出错误信息.

假定运动方程必须满足 K 个完整可积分约束, 表述为

$$\sigma_k(\mathbf{r}_1 \dots \mathbf{r}_N) = 0; k = 1 \dots K \quad (3.102)$$

例如 $(\mathbf{r}_1 - \mathbf{r}_2)^2 - b^2 = 0$. 力的定义为:

$$-\frac{\partial}{\partial \mathbf{r}_i} \left(V + \sum_{k=1}^K \lambda_k \sigma_k \right) \quad (3.103)$$

其中 λ_k 为Lagrange 乘子, 必须求解它们以满足约束方程. 上面求和的第二部分决定了 约束力 \mathbf{G}_i , 定义为:

$$\mathbf{G}_i = - \sum_{k=1}^K \lambda_k \frac{\partial \sigma_k}{\partial \mathbf{r}_i} \quad (3.104)$$

在蛙跳式或Verlet 算法中, 由约束力引起的位移为 $(\mathbf{G}_i/m_i)(\Delta t)^2$. 求解Lagrange 乘子(进而位移)需要求解一组二度耦合的方程, 这可使用SHAKE方法迭代求解. 对于刚性水分子的特殊情况, 这种模拟构成了超过了80%的模拟体系, 我们已经实现了SETTLE算法(见5.5节)[45].

对速度Verlet, 必须施加额外的约束, 以约束第二个半速度步的速度, 并去除平行于键向量的任何速度分量, 这一步骤被称为 RATTLE, Andersen 的论文中对其有更详细的论述[46].

3.6.2 LINCS

LINCS算法

在未约束的一步更新之后, LINCS算法会将键重置为它们的正确长度. 这个方法不进行迭代, 总是包含两个步骤. 尽管LINCS方法基于矩阵, 但并不需要进行矩阵乘法. LINCS方法比SHAKE方法更稳定, 也更快, 但只能用于键约束和孤立的键角约束, 如OH的质子键角. 由于其稳定性, LINCS特别适用于Brown 动力学. LINCS有两个参数, 在后文的参数节中将会解释. 并行版本的LINCS, P-LINCS, 将在3.17.3节介绍.

LINCS公式

考虑 N 个粒子的体系, 粒子位置由 $3N$ 向量 $\mathbf{r}(t)$ 给出. 分子动力学中的运动方程由牛顿定律给出:

$$\frac{d^2 \mathbf{r}}{dt^2} = \mathbf{M}^{-1} \mathbf{F} \quad (3.105)$$

其中 \mathbf{F} 为 $3N$ 个力向量, \mathbf{M} 为 $3N \times 3N$ 的对角矩阵, 包含了粒子的质量. 体系受到 K 个与时间无关的约束方程

$$g_i(\mathbf{r}) = |\mathbf{r}_{i_1} - \mathbf{r}_{i_2}| - d_i = 0 \quad i = 0, \dots, K \quad (3.106)$$

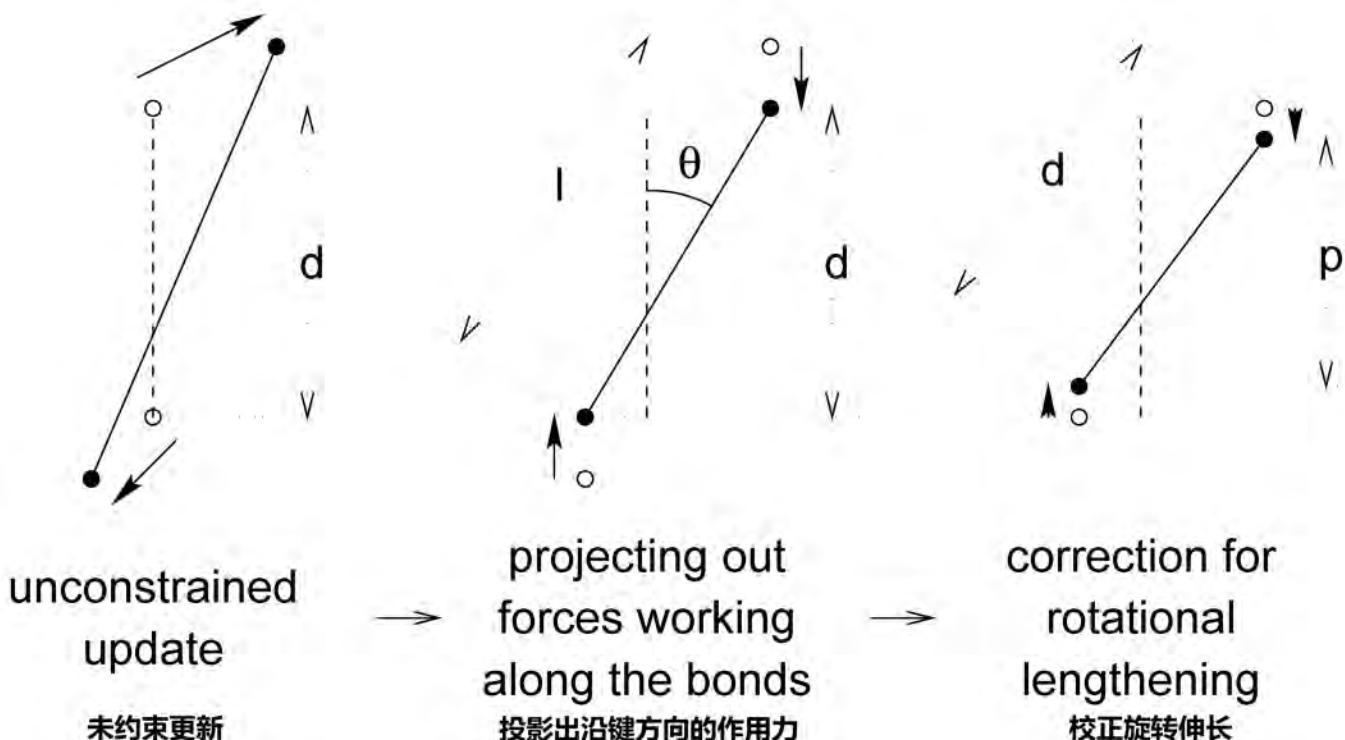


图 3.10: 一个时间步中进行的三次位置更新. 虚线为原先的键, 长度为 d , 实线为新的键. $l = d \cos \theta$, $p = (2d^2 - l^2)^{\frac{1}{2}}$

在数值积分方案中, LINCS会在一个未约束的更新之后施加, 如SHAKE一样. LINCS算法分两步工作(见图3.10). 在第一步中, 新键到旧键的投影被设置为零, 在第二步中, 对因旋转导致的键伸长进行校正. 第一步与第二步的计算非常相似. 算法的完整推导见[47], 这里只给出第一步的一个简短说明.

对约束方程的梯度矩阵使用新的记号, 记为

$$B_{hi} = \frac{\partial g_H}{\partial r_i} \quad (3.107)$$

注意 \mathbf{B} 为 $K \times 3N$ 的矩阵, 包含了约束的方向. 下列方程显示了新的约束坐标 \mathbf{r}_{n+1} 与未约束坐标 \mathbf{r}_{n+1}^{unc} 的关系

$$\begin{aligned} \mathbf{r}_{n+1} &= (\mathbf{I} - \mathbf{T}_n \mathbf{B}_n) \mathbf{r}_{n+1}^{unc} + \mathbf{T}_n \mathbf{d} = \\ \mathbf{r}_{n+1}^{unc} &- \mathbf{M}^{-1} \mathbf{B}_n (\mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T)^{-1} (\mathbf{B}_n \mathbf{r}_{n+1}^{unc} - \mathbf{d}) \end{aligned} \quad (3.108)$$

其中 $\mathbf{T} = \mathbf{M}^{-1} \mathbf{B}^T (\mathbf{B} \mathbf{M}^{-1} \mathbf{B}^T)^{-1}$. 由方程3.105, 3.106推导此方程的过程可参见[47].

第一步并未将实际键的长度设置为需要的长度, 而是将新键投影到旧键的方向上. 为校正键 i 的旋转, 键在原先方向上的投影 p_i 被设置为

$$p_i = \sqrt{2d_i^2 - l_i^2} \quad (3.109)$$

其中 l_i 为经第一步投影后键的长度. 校正后的位置为

$$\mathbf{r}_{n+1}^* = (\mathbf{I} - \mathbf{T}_n \mathbf{B}_n) \mathbf{r}_{n+1} + \mathbf{T}_n \mathbf{p} \quad (3.110)$$

对旋转效应的校正实际上是一个迭代过程, 但在MD过程中只进行了一步. 对每一约束, 经过这一校正后, 约束的相对偏差将小于 0.0001. 在能量最小化中, 这可能不够精确, 因此, 迭代次数被设置为展开的阶数(见后文).

约一半的CPU时间被用于对约束耦合矩阵 $\mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T$ 进行求逆, 而且必须每步进行一次. $K \times K$ 矩阵的对角元素为 $1/m_{i_1} + 1/m_{i_2}$. 只有当两个键相连时非对角元素才不为零, 而是 $\cos \phi / m_c$, 其中 m_c 为连接两个键的原子质量, ϕ 为键之间的

夹角.

使用幂展开方法对矩阵 \mathbf{T} 求逆. 为此引入 $K \times K$ 的矩阵 \mathbf{S} , 其元素为矩阵 $\mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T$ 对角元素平方根的倒数. 这一矩阵可用于将耦合矩阵的对角元素变换为1:

$$\begin{aligned} (\mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T)^{-1} &= \mathbf{S} \mathbf{S}^{-1} (\mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T)^{-1} \mathbf{S}^{-1} \mathbf{S} \\ &= \mathbf{S} (\mathbf{S} \mathbf{B}_n \mathbf{M}^{-1} \mathbf{B}_n^T \mathbf{S})^{-1} \mathbf{S} = \mathbf{S} (\mathbf{I} - \mathbf{A}_n)^{-1} \mathbf{S} \end{aligned} \quad (3.111)$$

矩阵 \mathbf{A}_n 是对称的稀疏矩阵, 对角元素中含有零, 因此可以使用一个简单的技巧来计算其逆矩阵

$$(\mathbf{I} - \mathbf{A}_n)^{-1} = \mathbf{I} + \mathbf{A}_n + \mathbf{A}_n^2 + \mathbf{A}_n^3 + \dots \quad (3.112)$$

这种求逆方法只有当所有 \mathbf{A}_n 本征值的绝对值都小于1时才使用. 对只含有键约束的分子, 其连接度非常低, 这个条件总是成立, 即使有环状结构出现也是一样. 对含有键角约束的分子, 可能会有问题. 使用额外的距离约束来约束键角, 可以引入多个小的环状结构. 这样分子的连接度变高, 本征值变大. 因此LINCS方法不能应用于耦合键角约束.

对所有键都进行约束的分子, \mathbf{A}_n 的本征值约为0.4, 这意味着在展开方程3.112中, 每增大一个阶数, 偏差会下降为原来的0.4. 但对相对孤立的三角约束, 最大的本征值大约为0.7. 当使用额外的角约束移除醇类基团中含氢的键角振动, 或当使用LINCS约束水分子, 如使用柔性约束时, 会导致这些三角约束出现. 这些三角约束的收敛速度比其他约束慢两倍. 因此, 从GROMACS 4.0版本开始, 一些额外的项被添加到这些三角约束的展开中

$$(\mathbf{I} - \mathbf{A}_n)^{-1} \approx \mathbf{I} + \mathbf{A}_n + \dots + \mathbf{A}_n^{N_i} + \left(\mathbf{A}_n^* + \dots + \mathbf{A}_n^{*N_i} \right) \mathbf{A}_n^{N_i} \quad (3.113)$$

其中 N_i 为正常的展开阶数, \mathbf{A}^* 只包含 \mathbf{A} 中处于刚性三角耦合约束内的元素, 所有其他元素为零. 使用这种方法, 角约束的精度接近其他约束的精度, 同时确定展开所需要的一系列矩阵矢量乘法只需要对几个约束耦合进行. 这一过程在P-LINCS的论文[48]中有说明.

LINCS参数

LINCS精度取决于展开方程3.112中矩阵的数目. 对MD计算四阶展开就足够了; 对Brown 动力学, 使用大的时间步长时, 可能需要八阶展开. 展开阶数是*.mdp文件中的一个参数. LINCS的实现方式使得其算法不可能失败. 即使不可能重置约束时, LINCS也将创建一个尽可能满足约束构型. 然而, 当在某一步中键的旋转超过了预先定义的角度时, LINCS会给警告. 这一角度用户可以在*.mdp文件中设置.

3.7 模拟退火

GROMACS支持众所周知的模拟退火方法. 你甚至可以独立地将多个原子组和任意数目的参考温度进行耦合, 参考温度在模拟中会改变. 对温度耦合中的每一组, 退火是通过简单地改变当前的参考温度实现的. 因此, 实际的弛豫和耦合性质取决于你使用的热浴类型与耦合强度. 由于参考温度会变化, 请务必记住体系温度 不会 瞬时达到参考值–你必须考虑到耦合算法内在的弛豫时间. 如果退火参考温度的改变快于温度弛豫, 当它们之间相差过大时, 模拟可能会崩溃.

退火策略是对每个组指定一系列相应的时间和参考温度. 你还可以选择只使用单个序列(退火完成后温度将耦合到最后的参考值), 或是使用周期性退火, 当退火序列完成后重新从第一个参考温度开始. 在模拟中, 你可以混合配合使用两种退火类型和不退火的组.

3.8 随机动力学

随机或速度Langevin 动力学在牛顿运动方程中增加了摩擦项和噪声项:

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = -m_i \gamma_i \frac{d\mathbf{r}_i}{dt} + \mathbf{F}_i(\mathbf{r}) + \dot{\mathbf{r}}_i \quad (3.114)$$

其中 γ_i 为摩擦常数[1/ps], $\dot{\mathbf{r}}_i$ 为噪声过程, 满足 $\langle \dot{\mathbf{r}}_i(t) \dot{\mathbf{r}}_j(t+s) \rangle = 2m_i \gamma_i k_B T \delta(s) \delta_{ij}$. 当与体系的时间尺度相比, $1/\gamma_i$ 较大时, 随机动力学可视为具有随机温度耦合的分子动力学. 与使用Berendsen 温度耦合的MD相比, SD的优势在于它产生的系综是已知的. 模拟真空中的体系时, SD还有另外一个优点: 对整个平动转动自由度不存在误差累积. 当 $1/\gamma_i$ 小于体系的时间尺度时, 动力学与MD完全不同, 但取样仍然正确.

在GROMACS中有两种算法可用于积分方程3.114: 简单有效的方法和更复杂的蛙跳式算法[49]. 后一方法现在已经废弃. 这两种积分方法的精度等价于常规MD的蛙跳式积分方法与速度Verlet 积分方法的精度, 不同在于存在约束时复杂SD积分方法的采样温度稍微过高(尽管误差小于使用整步速度动能项的速度Verlet 积分方法). 简单积分方法几乎与等同于常用的离散Langevin 方程方法, 但以脉冲方式施加摩擦项与速度项[50]. 简单积分方法如下:

$$\mathbf{v}' = \mathbf{v}(t - \frac{1}{2}\Delta t) + \frac{1}{m} \mathbf{F}(t) \Delta t \quad (3.115)$$

$$\Delta \mathbf{v} = -\alpha \mathbf{v}'(t + \frac{1}{2}\Delta t) + \sqrt{\frac{k_B T}{m}(1-\alpha^2)} \mathbf{r}_i^G \quad (3.116)$$

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \left(\mathbf{v}' + \frac{1}{2}\Delta t \right) \Delta t \quad (3.117)$$

$$\mathbf{r}(t + \frac{1}{2}\Delta t) = \mathbf{v}' + \Delta \mathbf{v} \quad (3.118)$$

$$\alpha = 1 - e^{-\gamma \Delta t} \quad (3.119)$$

其中 \mathbf{r}_i^G 为服从高斯分布的噪声, 其 $\mu = 0, \sigma = 1$. 算法首先在无摩擦与噪声的情况下更新整个时间步的速度, 获得 \mathbf{v}' , 这完全与蛙跳式的常规更新相同. 然后在 $t + \Delta t$ 步以脉冲方式施加摩擦与噪声. 这种方案的优点在于依赖速度的项在整个时间步中都起作用, 由于对力的积分取决于坐标与速度, 这样就使得方案对力进行正确积分很简单, 如在约束和耗散粒子动力学(DPD, dissipative particle dynamics, 还未实现)情况下. 有约束时, 坐标更新方程3.117会被划分为常规的蛙跳式更新与 $\Delta \mathbf{v}$. 这些更新都完成后, 再对坐标和速度施加约束.

在已废弃的复杂方法中, 在每个积分步对每个自由度需要四个高斯随机数, 有约束时, 每个积分步坐标需要约束两次. 取决于力的计算代价, 这可能会占用显著一部分模拟时间. 精确地对随机动力学进行继续运行是不可能的, 因为在模拟过程中没有存储随机数产生器的状态.

当使用SD控温时, γ 的合适值可取0.5 ps⁻¹, 因为这样得到的摩擦低于水的内摩擦, 同时仍能提供有效的控温.

3.9 Brown动力学

在高摩擦的极限情况下, 随机动力学退化为Brown 动力学, 也被称为位置Langevin 动力学. 它施加于过阻尼体系, 即惯性效应可忽略的体系. 方程为

$$\frac{d\mathbf{r}_i}{dt} = \frac{1}{\gamma_i} \mathbf{F}_i(\mathbf{r}) + \dot{\mathbf{r}}_i \quad (3.120)$$

其中 γ_i 为摩擦系数[amu/ps], $\dot{\mathbf{r}}_i(t)$ 为噪声过程, 满足 $\langle \dot{\mathbf{r}}_i(t) \dot{\mathbf{r}}_j(t+s) \rangle = 2\delta(s) \delta_{ij} k_B T / \gamma_i$. 在GROMACS中, 使用简单的显示方案进行上面的方程进行积分:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \frac{\Delta t}{\gamma_i} \mathbf{F}_i(\mathbf{r}(t)) + \sqrt{2k_B T \frac{\Delta t}{\gamma_i}} \mathbf{r}_i^G \quad (3.121)$$

其中 \mathbf{r}_i^G 为服从高斯分布的噪声, 其 $\mu = 0, \sigma = 1$. 摩擦系数 γ_i 可以取为对所有粒子相同, 或者取为 $\gamma_i = m_i \gamma_i$, 其中的摩擦系数 γ_i 对不同组的原子可取不同值. 由于假定体系处于过阻尼状态, 因此可以使用大的时间步长. 应使用LINCS方法进行约束, 因为

SHAKE方法对大的原子位移不收敛. BD是 `mdrun` 程序的一个选项.

3.10 能量最小化

GROMACS中的能量最小化可以使用最速下降, 共轭梯度, 或L-BFGS(有限内存的Broyden-Fletcher-Goldfarb-Shanno 准牛顿最小化, 我们倾向于使用简称). EM只是 `mdrun` 程序的一个选项.

3.10.1 最速下降

尽管最速下降方法确实不是最有效的搜索算法, 它很稳健, 并且易于实现.

定义代表所有 $3N$ 个坐标的向量 \mathbf{r} . 初始时, 必须给出的最大位移值 h_0 (例如0.01 nm).

首先计算力 \mathbf{F} 和势能. 新位置的计算方法如下:

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \frac{\mathbf{F}_n}{\max(|\mathbf{F}_n|)} h_n \quad (3.122)$$

其中 h_n 为最大位移, \mathbf{F}_n 为力, 也即势能 V 的负梯度. $\max(|\mathbf{F}_n|)$ 表示力分量绝对值的最大值. 再次计算新位置的力与能量,

- 若 $V_{n+1} < V_n$, 则接受新位置, 令 $h_{n+1} = 1.2h_n$
- 若 $V_{n+1} \geq V_n$, 则拒绝新位置, 令 $h_{n+1} = 0.2h_n$

当力的计算次数达到用户指定的数目(如100), 或力(梯度)分量绝对值的最大值小于设定值 ϵ 后, 算法停止. 由于力的截断会使能量计算产生一些噪声, 停止条件不应该设得过高, 以免算法迭代无法结束. ϵ 的合理值可以根据简谐振子在温度 T 下的受到的根均方力 f 来估计, 其值为

$$f = 2\pi\nu\sqrt{2mkT} \quad (3.123)$$

其中 ν 为振子频率, m 为(约化)质量, k 为Boltzmann 常数. 对波数为100 cm⁻¹, 质量为10 原子单位, 温度 1 K时的弱振子, $f = 7.7$ kJ mol⁻¹ nm⁻¹. ϵ 的值在1到10之间是可以接受的.

3.10.2 共轭梯度

共轭梯度在最小化早期慢于最速下降, 但靠近能量最低点时更高效. 方法的参数和停止条件与最速下降相同. 在GROMACS中, 共轭梯度不能与约束一起使用, 包括用于刚性水的SETTLE算法[45], 因为还未实现. 若体系中存在水分子, 必须使用柔性水模型, 这可以在 `*.mdp` 文件中利用 `define = -DFLEXIBLE` 指定.

这真的不是一个限制, 因为只有在简正模式分析前的能量最小化中才需要共轭梯度的精度, 简正分析时不能使用约束. 对大多数其他目的, 最速下降足够高效.

3.10.3 L-BFGSD

原始的BFGS算法会逐步产生更好的逆Hess矩阵的近似, 并将体系移动到当前的估计的最小点. 这种方法需要的内存正比于粒子数的平方, 因此实用中不能用于大体系如生物分子. 作为替代, 我们使用了Nocedal 的L-BFGS算法[51, 52]. 它使用来自前一步的固定数目的校正得到近似的逆Hess矩阵. 这种滑动窗技术几乎与原始方法一样有效, 但所需内存要少得多—正比于粒子数与校正步数的乘积. 实用中, 我们发现它比共轭梯度收敛得更快, 但由于存在校正步, 此方法还未并行化. 也需要指出, 切换或移位相互作用通常可以提高收敛性, 因为明显的截断意味着当前坐标下的势能函数与前一步用于构建逆Hess矩阵近似的势能函数稍有不同.

3.11 简正模式分析

GROMACS可以进行简正模式分析[53, 54, 55], 计算是通过对角化质量加权的Hess矩阵 H 进行的:

$$R^T M^{-1/2} H M^{-1/2} R = \text{diag}(\lambda_1, \dots, \lambda_{3N}) \quad (3.124)$$

$$\lambda_i = (2\pi\omega_i)^2 \quad (3.125)$$

其中 M 包含了原子质量, R 的列为本征向量, λ_i 为本征值, ω_i 为相应的频率.

首先需要计算Hess矩阵, 它是一个 $3N \times 3N$ 的矩阵, 其中 N 为原子数,

$$H_{ij} = \frac{\partial^2 V}{\partial x_i \partial x_j} \quad (3.126)$$

其中 x_i 和 x_j 代表原子的x, y或z坐标. 实际计算时并不使用上面的方程, 而是根据得到的力, 利用数值方法计算Hess矩阵

$$H_{ij} = -\frac{f_i(\mathbf{x} + h\mathbf{e}_j) - f_i(\mathbf{x} - h\mathbf{e}_j)}{2h} \quad (3.127)$$

$$f_i = -\frac{\partial V}{\partial x_i} \quad (3.128)$$

其中 \mathbf{e}_j 为 j 方向的单位向量. 应当指出, 对通常的简正模式计算, 在计算Hess矩阵前需要进行彻底的能量最小化, 需要的能量容差取决于体系的类型, 粗略的指示值为 $0.001 \text{ kJ mol}^{-1}$. 应当使用共轭梯度或L-BFGS方法进行能量最小化, 并使用双精度版本.

在这些计算中会涉及多个GROMACS程序. 首先应该使用 `mdrun` 进行能量最小化, 其次再使用 `mdrun` 计算Hess矩阵. 注意, 创建运行输入文件时, 应使用来自全精度轨迹文件中的最小化的构型, 因为结构文件不够精确. `g_nmeig` 程序进行对角化计算, 并将简正模式按其频率进行排序. `mdrun` 和 `g_nmeig` 都应当使用双精度版本. 可以使用 `g_anaeig` 程序分析简正模式. 要创建任意温度下的结构系综中的任意一组简正模式, 可使用 `g_nmens` 程序. 关于简正模式分析及其相关的主成分分析(见8.10节)的概述可参见[56].

3.12 自由能计算

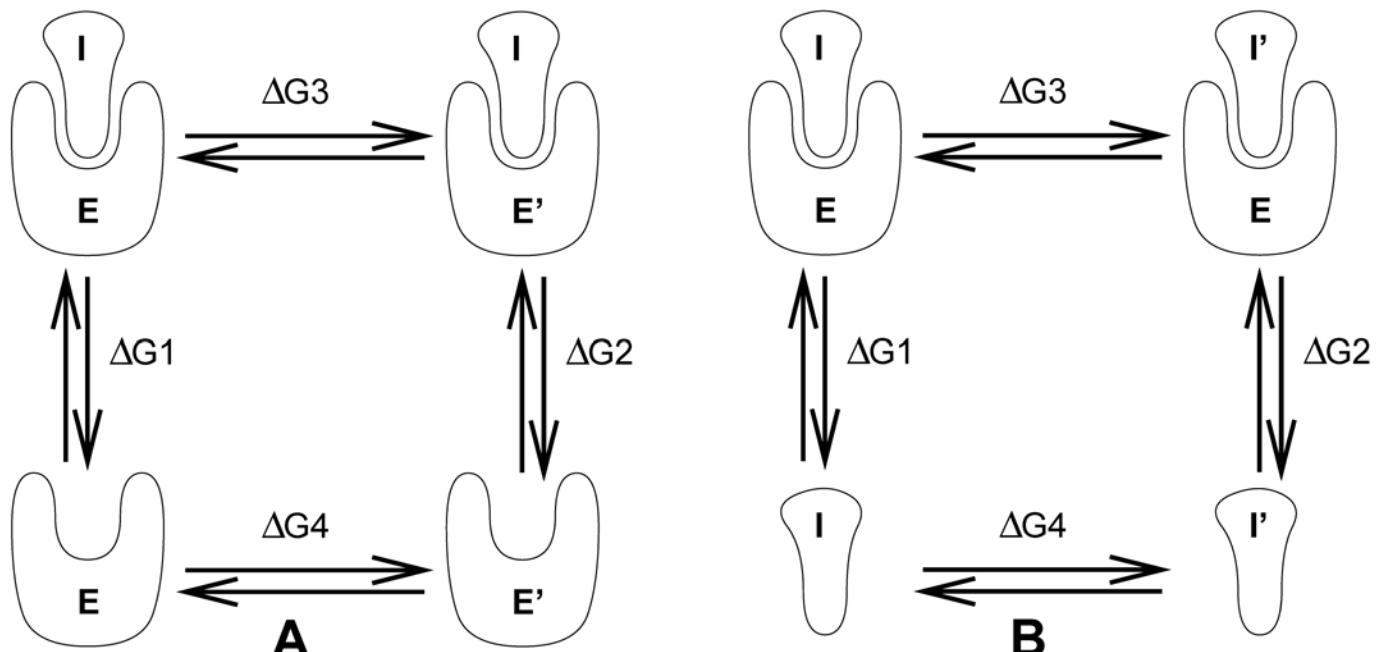


图3.11: 自由能循环. A: 计算 ΔG_{12} , 抑制剂I分别与酶E, E'结合时的自由能差值. B: 计算 ΔG_{12} , 抑制剂I, I'分别与酶结合时的自由能差值.

3.12.1 慢增长方法

GROMACS可使用几种不同的方法计算自由能, 其中包括“慢增长”方法. 自由能计算的一个例子是计算抑制剂I分别与酶E, 突变后的

酶E'结合时的自由能差值. 对这样的大体系, 利用计算机模拟来进行对接计算是不可行的, 即便是在合理的计算时间内以合理的计算精度模拟抑制剂从酶中释放的过程, 也是不可行的. 然而, 如果我们考虑图3.11 A中的热力学循环, 可得到

$$\Delta G_1 - \Delta G_2 = \Delta G_3 - \Delta G_4 \quad (3.129)$$

若对左边的项感兴趣, 我们也可以同样好地计算右边的项.

如果我们要计算两个抑制剂I和I'与酶E'结合时的自由能差值(图3.11 B), 我们仍然可以使用方程3.129来计算所需要的性质.

在GROMACS中可以利用“慢增长”方法计算两个分子物种之间的自由能差值. 不同分子物种之间的这种自由能差值没有物理意义, 但利用热力学循环, 可由它们获得有意义的量. 这种方法要求在模拟中体系的哈密顿量缓慢地从描述一个体系(A)转变到描述另一个体系(B). 变化必须足够缓慢, 以保证体系在这个过程中仍保持平衡状态. 若能满足这些要求, 变化是可逆的, 并且从B到A的慢增长模拟与从A到B的慢增长模拟会得到同样的结果(但符号相反). 这是一个有用的核查方法, 但用户应当心, 正向与反向增长结果的相等并不能确保结果的正确性.

对哈密顿量所需的修改是通过将 H 作为耦合参数 λ 的函数实现的: $H = H(p, q; \lambda)$. 这样 $\lambda = 0$ 描述体系A, $\lambda = 1$ 描述体系B:

$$H(p, q; 0) = H^A(p, q); \quad H(p, q; 1) = H^B(p, q) \quad (3.130)$$

在GROMACS中, λ 依赖的函数形式对各种力场贡献是不同的, 对此的说明请参看4.5节.

Helmholtz 自由能 A 与 N, V, T 系综的配分函数 Q 有关, 这一系综被认为是等容等温MD模拟产生的平衡系综. 一般更有用的 Gibbs 自由能 G 与 N, P, T 系综的配分函数 Δ 有关, 这一系综被认为是等压等温MD模拟产生的平衡系综.

$$A(\lambda) = -k_B T \ln Q \quad (3.131)$$

$$Q = c \iint \exp[-\beta H(p, q; \lambda)] dp dq \quad (3.132)$$

$$G(\lambda) = -k_B T \ln \Delta \quad (3.133)$$

$$\Delta = c \iiint \exp[-\beta H(p, q; \lambda) - \beta pV] dp dq dV \quad (3.134)$$

$$G = A + pV \quad (3.135)$$

其中 $\beta = 1/(k_B T)$, $c = (N!h^{3N})^{-1}$. 这些对相空间的积分不能由模拟得到, 但可以计算它们对 λ 的导数作为系综平均:

$$\frac{dA}{d\lambda} = \frac{\iint (\partial H / \partial \lambda) \exp[-\beta H(p, q; \lambda)] dp dq}{\iint \exp[-\beta H(p, q; \lambda)] dp dq} = \left\langle \frac{\partial H}{\partial \lambda} \right\rangle_{NVT; \lambda} \quad (3.136)$$

在 N, p, T 系综中的关系与此类似. A和B之间的自由能差值可通过积分对 λ 的导数获得:

$$A^B(V, T) - A^A(V, T) = \int_0^1 \left\langle \frac{\partial H}{\partial \lambda} \right\rangle_{NVT; \lambda} d\lambda \quad (3.137)$$

$$G^B(p, T) - G^A(p, T) = \int_0^1 \left\langle \frac{\partial H}{\partial \lambda} \right\rangle_{NpT; \lambda} d\lambda \quad (3.138)$$

若要计算 $G^B(p, T) - G^A(p, T)$, 自然的选择是等压模拟. 然而, 也可以从等容的慢增长模拟计算这个量, 模拟由压力为 p 体积为 V 的体系A开始, 到压力为 p_B 的体系B结束, 并进行下面的小(但原则上精确)校正:

$$G^B(p) - G^A(p) = A^B(V) - A^A(V) - \int_p^{p^B} [V^B(p') - V] dp' \quad (3.139)$$

这里的记号中忽略了温度 T . 这一校正粗略地等于 $-\frac{1}{2}(p^B - p)\Delta V = (\Delta V)^2/(2\kappa V)$, 其中 ΔV 为 p 时的体积变化, κ 为等

温压缩率. 这一校正通常很小, 例如, 等容情况下, 在1000个水分子的体系中凭空产生一个水分子增加的额外压力最大为22 bar, 但对Helmholtz 自由能的校正只有 -1 kJ mol^{-1} .

在直角坐标系中, 哈密顿量的动能项只取决于动量, 可独立进行积分, 并且实际上可从方程中去除. 当质量不变时, 动能对自由能完全没有贡献. 当质量变化时, 对自由能的累计贡献为 $-\frac{3}{2}k_B T \ln(m^B/m^A)$. 注意 这只对无约束的情况适用.

3.12.2 热力学积分

GROMACS也可以在一个模拟中对方程3.137或3.138从A到B的整个范围内进行积分. 然而, 若变化太大, 采样可能不充分, 用户可能更倾向于在仔细选择的中间的 λ 值上精确地确定 $\langle dG / d\lambda \rangle$ 的值. 通过将步长`delta_lambda`设定为零, 这很容易实现. 每一模拟都可以先进行平衡, 由 $\partial H / \partial \lambda$ 的涨落可以得到每个 $\langle dG / d\lambda \rangle$ 值的合适的误差估计. 然后, 可以通过适当的数值积分获得总的自由能变化.

GROMACS现在也支持使用`g_bar`程序计算从状态A转变到状态B的 ΔG 值, 计算时利用BAR(Bennett's Acceptance Ratio) 方法. 相同的数据也可用MBAR[58]方法来计算自由能, 但目前分析时需要来自外部`pymbar`程序包中的工具, 可从<https://SimTK.org/home/pymbar> 下载.

力场贡献与 λ 的依赖关系将在4.5节详细讨论.

3.13 副本交换动力学

副本交换分子动力学(REMD, replica exchange molecular dynamics) 是一种加速抽样的方法, 可用于任何类型的模拟, 特别适用于不同构象之间存在较高能量势垒的情况. 它对同一体系处于不同温度下的多个副本进行模拟, 并以一定的时间间隔随机地完全交换两个副本的状态, 交换的概率为:

$$P(1 \leftrightarrow 2) = \min \left(1, \exp \left[\left(\frac{1}{k_B T_1} - \frac{1}{k_B T_2} \right) (U_1 - U_2) \right] \right) \quad (3.140)$$

其中 T_1 和 T_2 为参考温度, U_1 和 U_2 分别为副本1和2的瞬时势能. 交换后, 使用 $(T_1/T_2)^{\pm 0.5}$ 对速度进行缩放, 并在下一步进行近邻搜索. 这方法将快速采样和最高温度频繁跨垒方法结合起来, 在所有不同温度下都使用了正确的玻耳兹曼抽样[59, 60]. 由于随着温差的增大概率下降非常迅速, 我们只会尝试交换相邻的温度. 在一步中我们不应尝试对所有可能的副本对进行交换. 例如, 如果交换副本1和2, 交换副本2和3的机会不仅取决于副本2和3的能量, 还取决于副本1的能量. 在GROMACS中这个问题的解决方法是, 在“奇数”次尝试试着交换所有的“奇数”副本对, 在“偶数”次尝试试着交换所有的“偶数”副本对. 如果我们有四个按温度排序的副本: 0, 1, 2和3, 并且每1000步尝试进行一次交换, 那么将会在1000, 3000等步尝试对副本对0-1和2-3进行交换, 在2000, 4000等步尝试对副本对1-2进行交换.

应该如何选择温度呢? 能量差可以写成:

$$U_1 - U_2 = N_{df} \frac{c}{2} k_B (T_1 - T_2) \quad (3.141)$$

其中 N_{df} 为一个副本的总自由度数, 对简谐势 c 为1, 对蛋白质/水体系 c 为2左右. 如果 $T_2 = (1 + \epsilon)T_1$ 则概率变为:

$$P(1 \leftrightarrow 2) = \exp \left(-\frac{\epsilon^2 c N_{df}}{2(1 + \epsilon)} \right) \approx \exp \left(-\epsilon^2 \frac{c}{2} N_{df} \right) \quad (3.142)$$

这样, 对于大小为 $e^{-2} \approx 0.135$ 的概率, 可得到 $\epsilon \approx 2/\sqrt{c N_{df}}$. 对所有键都进行约束时, $N_{df} \approx 2N_{atoms}$, 这样对于 $c = 2$ 我们应选择的 ϵ 为 $1/\sqrt{N_{atoms}}$. 然而, 当我们使用压力耦合时会出现一个问题. 更高温时密度会下降, 并导致能量增加[61], 这些情况我们需要考虑到. GROMACS网站提供了一个被称为“REMD calculator”的计算器, 它可以根据你输入的温度范围和原子数给出一组温度.

Okabe等人提出了等压等温系综的REMD扩展[62]. 他们将交换概率修改为:

$$P(1 \leftrightarrow 2) = \min \left(1, \exp \left[\left(\frac{1}{k_B T_1} - \frac{1}{k_B T_2} \right) (U_1 - U_2) + \left(\frac{P_1}{k_B T_1} - \frac{P_2}{k_B T_2} \right) (V_1 - V_2) \right] \right) \quad (3.143)$$

其中 P_1 和 P_2 为各自的参考压力, V_1 和 V_2 为模拟中各自的瞬时体积. 在大多数情况下, 体积的差别很小以至于可以忽略第二项. 只有在 P_1 和 P_2 之间相差很大或者存在相变的情况下第二项才起作用.

GROMACS也支持哈密顿量副本交换. 在哈密顿量副本交换中, 每个副本的哈密顿量不同, 并根据模拟指定的自由能途径进行定义. 用来维持正确系综概率的交换概率为:

$$P(1 \leftrightarrow 2) = \min \left(1, \exp \left[\left(\frac{1}{k_B T_1} - \frac{1}{k_B t_2} \right) ((U_1(x_2) - U_1(x_1)) + (U_2(x_1) - U_2(x_2))) \right] \right) \quad (3.144)$$

各自的哈密顿量由GROMCAS内置的自由能函数定义, 并在不同 λ 值之间进行交换, λ 在mdp文件中定义.

也可以同时进行哈密顿量副本交换和温度副本交换, 使用的接受标准为:

$$P(1 \leftrightarrow 2) = \min \left(1, \exp \left[\left(\frac{1}{k_B T_1} - \right) \left(\frac{U_1(x_2) - U_1(x_1)}{k_B T} + \frac{U_2(x_1) - U_2(x_2)}{k_B T_2} \right) \right] \right) \quad (3.145)$$

GROMACS还实现了吉布斯抽样副本交换[63]. 在吉布斯抽样副本交换中, 会测试所有可能交换的副本对, 并允许交换不相邻的副本.

吉布斯抽样副本交换不需要额外的势能计算. 然而, 在吉布斯抽样副本交换中存在额外的通讯成本, 对一些排列, 肯定会发生超过一轮的交换. 在某些情况下, 这些额外的通讯成本可能会影响效率.

`mdrun` 程序的包含了所有不同副本交换方法的选项. 由于算法中内在的并行化, 只有安装了MPI时候才能运行副本交换模拟. 为提高效率, 每个副本可以在一个独立的队列上运行. 想了解如何使用这些多节点功能, 可以参考 `mdrun` 的帮助文档.

3.14 主成分动力学抽样

蛋白质主成分动力学(参见8.10节)的结果可用于指导MD模拟. 基本思想是, 我们可以从初始MD模拟(或源自其他途径)中获得拥有最大振幅的集约涨落. 对沿一个或多个这些集约模式的那些位置, 在接下来的(第二次)MD模拟中可以使用约束, 约束时可根据不同的目的使用不同的方法. 例如, 可以固定沿一个特定模式的位置以监测在此位置坐标上的平均力(自由能梯度). 另外一个应用是提高相对于普通MD的抽样效率[64, 65]. 在这种情况下, 与蛋白质通常采用的类扩散途径相比, 对体系构型空间的采样更系统.

另外一个提高抽样效率的可能方法是洪泛方法, 将洪泛势添加到一些(集约)自由度上, 以将体系排除到在某些相空间区域之外[66].

主成分动力学抽样或洪泛方法的步骤如下. 首先使用协方差分析(`g_covar`)或者简正模式分析(`g_nmeig`)确定特征向量和特征值. 然后将这些信息用于 `make_edi`, 这个命令有很多用于选择向量和设定参数的选项, 请参考 `gmx make_edi -h`. 最后将生成的 `edi` 输入文件传递给 `mdrun`.

3.15 扩展系综动力学

在扩展系综模拟[67]中, 坐标和热力学系综都可以被视为构型变量进行抽样. 任何给定状态的概率可写为:

$$P(\vec{x}, k) \propto \exp(-\beta_k U_k + g_k) \quad (3.146)$$

其中 $\beta_k = \frac{1}{k_B T_k}$ 为对应于第 k 个热力学状态 β , g_k 为用户定义的对应于第 k 个状态的权重因子. 因此, 这一空间是混合, 通用,

或扩展的系综,是对多个热力学系综同时抽样得到的. g_k 用于为扩展系综中的每个子系综指定权重,既可以是固定的,也可以由迭代程序确定. 通常选择的 g_k 集合对每个热力学系综会给出相同的概率,在这种情况下, g_k 等于无量纲的自由能,但根据需要也可以将它们设定为任意值. 在mdp选项列表中有好几种算法可以用来平衡这些权重.

在GROMACS中,这个空间通过交替地在 k 和 \vec{x} 方向上采样得到. \vec{x} 方向的采样是由标准的分子动力学采样完成;不同热力学状态之间的采样由Monte Carlo 完成,且支持几种不同的Monte Carlo 移动方法. k 状态可以根据温度的不同进行定义,或根据自由能 λ 变量的选择进行定义,或者同时根据两者进行定义. 因此扩展系综模拟是系列化的副本交换形式,单次模拟就能探索多个热力学状态.

3.16 并行

通过在多个核上并行地进行计算,可以缩短模拟所需要的CPU时间. 理想情况下, 我们能够得到线性标度: 在 N 个核上运算就能使计算速度加快 N 倍. 但实际上, 只有当CPU核数目很少时才能达到这种效果. 计算的标度行为与所使用的算法关系很大. 此外, 不同算法对原子间的相互作用范围有不同的限制.

3.17 区域分解

由于分子模拟中的大部分相互作用都是局部的, 区域分解是分解体系的自然方法. 在区域分解中, 每个队列分配一个空间区域, 然后对当前处于局部区域内粒子的运动方程进行积分. 使用区域分解, 你必须要做两个选择: 将单位晶胞分成多个区域和将力分配给各个区域. 绝大部分分子模拟软件都是使用半壳层方法来分配力. 但还有另外两种需要通讯更少的方法: 八壳层方法[68]和中点法[69]. GROMACS目前使用的是八壳层法, 但对于某些体系或者硬件架构来说, 中点法可能更合适. 因此, 将来我们可能会实现中点法. 绝大部分区域分解的技术细节可以在GROMACS 4的论文[5]中找到.

3.17.1 通讯坐标和力

对三斜单元晶胞的最一般情况, 我们用1D, 2D或3D格点将空间分成平行六边体, 并称其为区域分解晶胞. 每个晶胞被分配到一个粒子-粒子队列. 在每个MD步的开始, 整个体系被分到多个队列, 并在那里执行近邻搜索. 由于近邻搜索基于电荷组, 因此电荷组也是区域分解的基本单位. 电荷组被分配到其几何中心所在的晶胞. 在计算力之前, 需要对一些来自近邻晶胞的坐标进行通讯, 在力的计算完成之后, 需要将力以相反的方向进行通讯. 通讯和力的分配基于能覆盖一个或多个晶胞的区域. 图3.12展示了一个区域设置的例子.

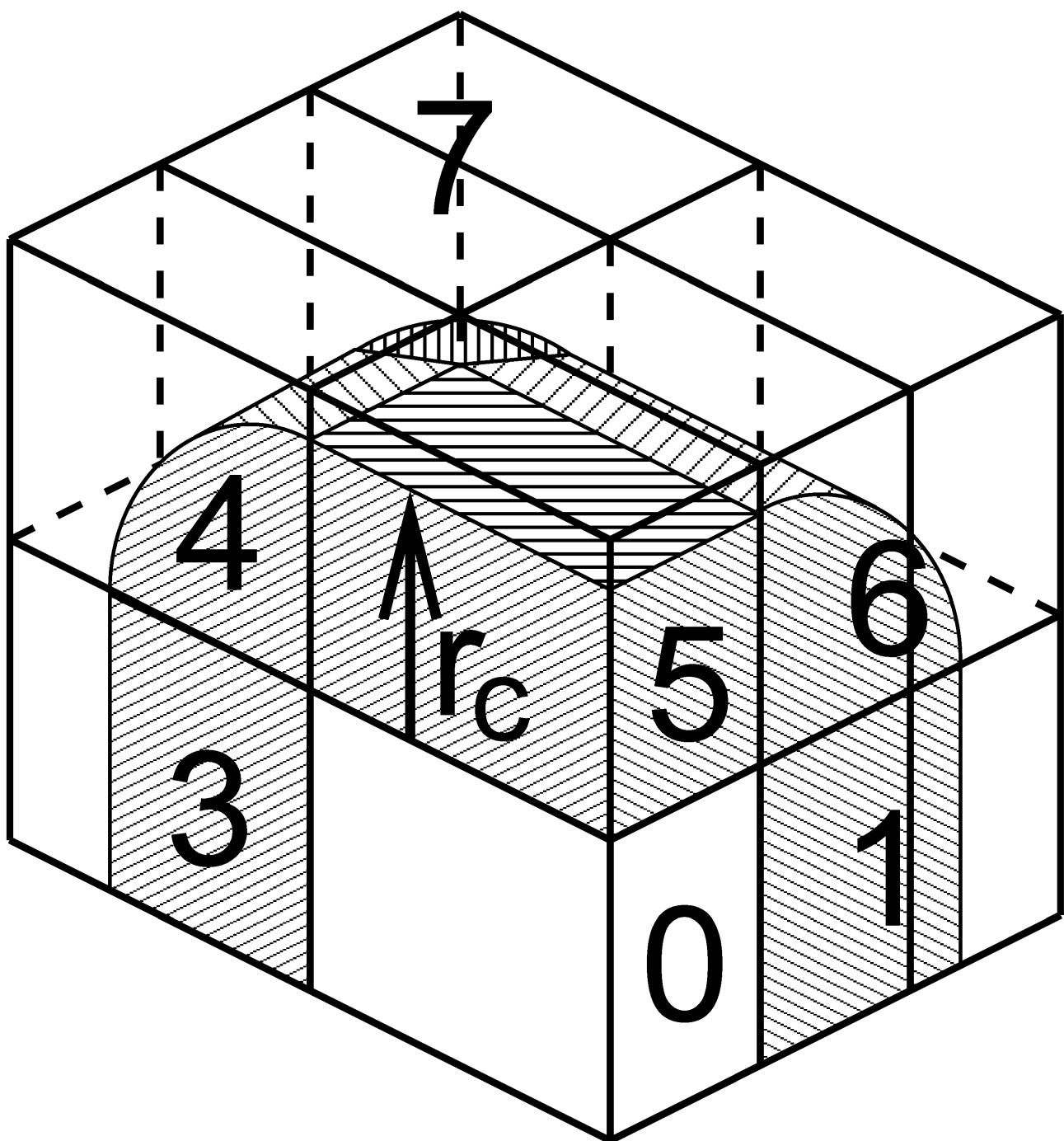


图3.12: $3 \times 2 \times 2$ 晶胞无交错的区域分解格点. 区域1到7中的坐标被通讯到区域0中拥有自己粒子的边角晶胞. r_c 为截断半径.

坐标通讯是通过将数据沿 x , y 或 z 的“负”方向移动到下一个近邻完成的. 这可以在一个或多个脉冲内完成. 在图3.12中, x 方向需要两个脉冲, 然后 y 方向需要1个脉冲, 然后 z 方向需要1个脉冲. 而力的通讯步骤则与此过程相反. 要想了解确定哪些非键力和键合力需要在哪个队列上进行计算的细节, 请参考GROMACS 4论文[5].

3.17.2 动态负载均衡

当不同队列的计算负载不同(负载失衡)时, 所有队列都必须等待最耗时的那个队列. 这是我们极力要避免的情况. 负载失衡可能由以下三个原因造成:

- 不均匀的粒子分布
- 不均匀的相互作用计算时间分布(带电/不带电, 由GROMACS内部的水计算循环造成的水/非水计算)
- 统计涨落(仅当粒子数目较少时)

因此我们需要一个动态负载均衡算法, 它能够 独立地 调整每个区域分解晶胞的体积. 为了达到这个目的, 需要交错2D或3D区域分解的格点. 图3.13显示了2D中的最一般情况. 由于交错, 我们可能需要两个距离检查, 以决定一个电荷组是否需要通讯: 非键距离检查和键合距离检查.

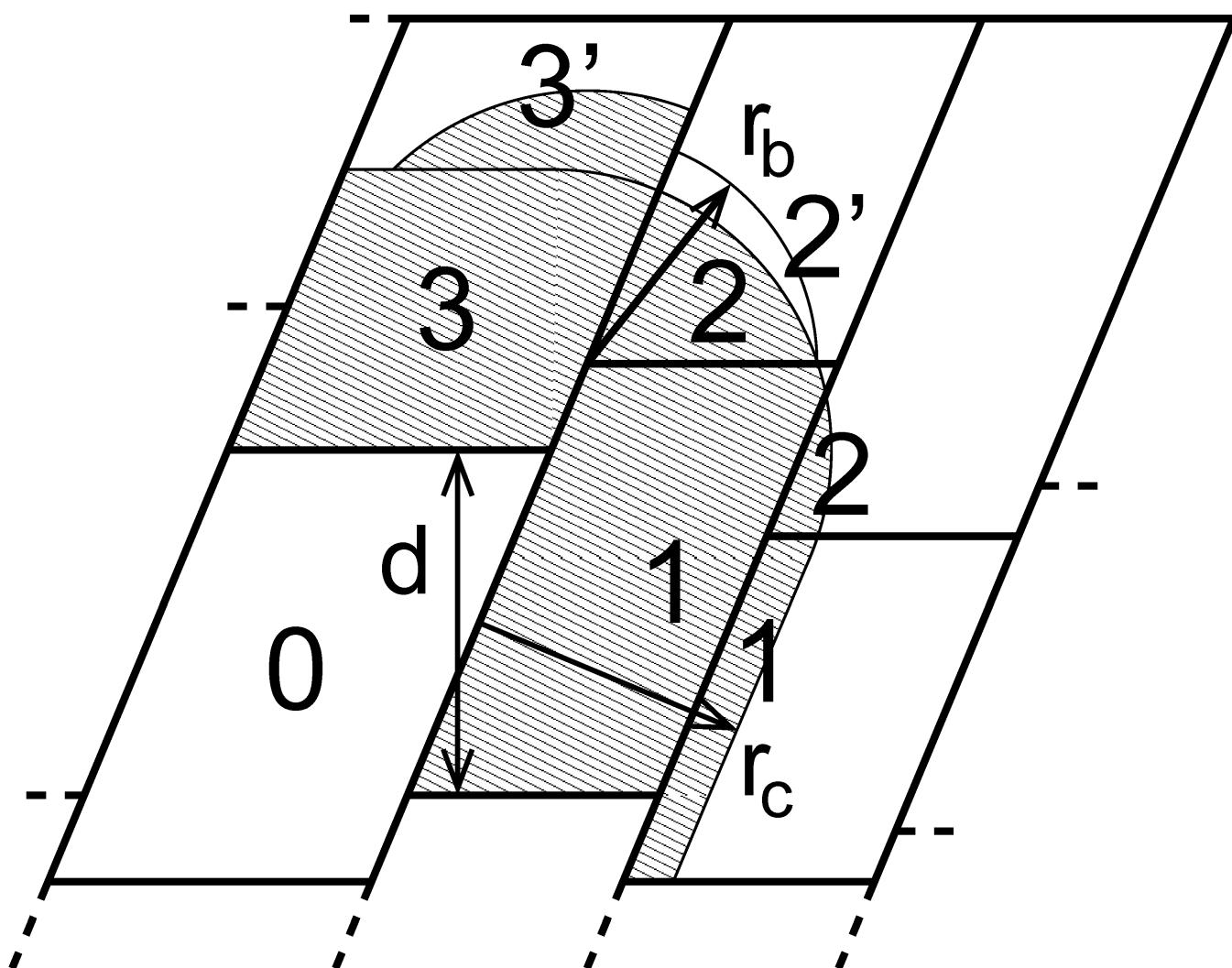


图3.13: 通讯到区域0队列的各个区域, 详情见正文. r_c 和 r_b 分别为非键截断半径和键合截断半径, d 为晶胞交错边界之间的距离.

默认情况下, 在模拟中, 当由力计算失衡造成的总体性能损失达到5%或更多时, `mdrun` 会自动开启动态负载均衡. 注意, 由于在一个积分步中, 力的计算只是所需要工作的一部分, 所以报告中力的负载失衡数据可能会偏高. 负载失衡会记录日志文件中, 输出频率由设定的日志输出步决定, 当使用 `-v` 选项时, 也会在屏幕上输出负载失衡. 平均负载失衡和由负载失衡导致的总性能损失会记录在日志文件的结尾处.

最小允许缩放比例是动态负载均衡的一项重要参数. 默认情况下, 在每个方向上区域分解晶胞至少能够缩小为原来的0.8. 对于3D区域分解, 这相当于允许晶胞的体积缩小为原来的0.5, 补偿100%的负载失衡. 这个最小允许缩放比例可以通过 `mdrun` 的 `-dds` 选项进行调整.

3.17.3 并行中的约束

由于使用区域分解时, 分子的各个部分可以处于不同的队列上, 键约束可以跨越晶胞的边界. 因此需要一个并行的约束算法.

GROMACS使用了P-LINCS算法[48], 它LINCS算法[47]的并行版(参见3.6.2节). P-LINCS算法过程的示意见图3.14. 当分子跨越晶胞边界时, 会在晶胞边界中对此分子中相邻直到(`lincs_order + 1`)键远的原子进行通讯. 然后, 会对局部的键和通讯得到的键施加正常的LINCS算法. 经过这个步骤之后, 已经正确地被约束了局部的键, 尽管额外通讯得到的键还没有. LINCS的初始步需要一个坐标通讯步. 不需要对力进行通讯.

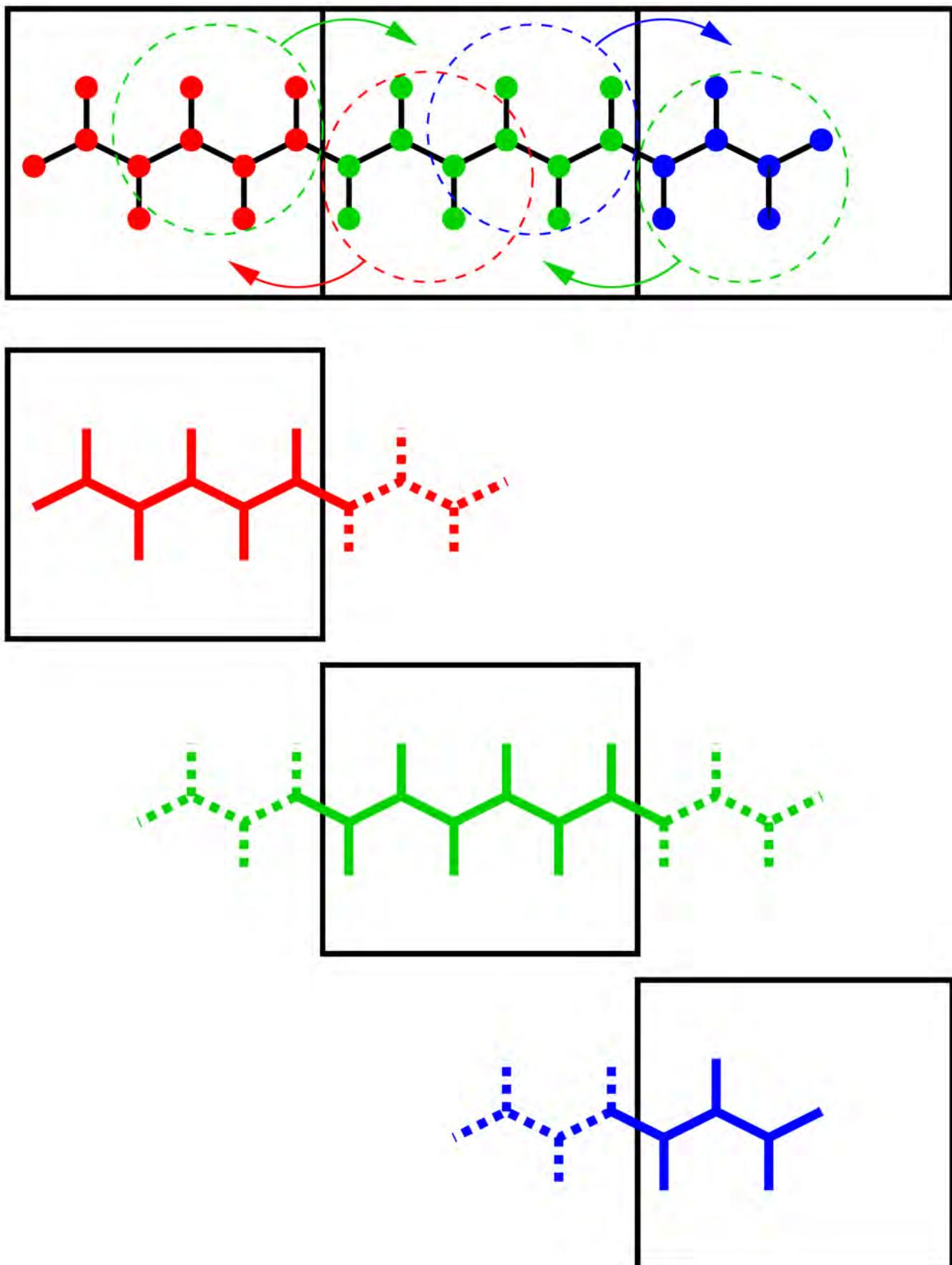


图3.14: P-LINCS算法并行设置示例.一个分子被划分到3个区域分解晶胞中,使用的矩阵展开阶数为3.上面部分展示了需要通讯的原子坐标及其要通讯到的晶胞.下面部分展示了三个晶胞中各自的局部约束(实线)和非局部约束(虚线).

3.17.4 相互作用范围

区域分解利用了相互作用的局域性.这就意味着对相互作用的范围有所限制.默认情况下, `mdrun` 程序会试着找到相互作用范围和效率之间的最佳平衡.但也有可能发生模拟停止并给出缺失相互作用的错误信息,或者当使用更短的相互作用范围时, 模拟运行稍微快

一些的情况. 相互作用范围及其默认值都列在表3.3中.

表3.3: 区域分解的相互作用范围

相互作用	范围	选项	默认
非键	$r_c = \max(r_{\text{list}}, r_{\text{VdW}}, r_{\text{Coul}})$	<code>mdp</code> 文件	
双体键合	$\max(r_{\text{mb}}, r_c)$	<code>mdrun -rdd</code>	初始构型 + 10%
多体键合	r_{mb}	<code>mdrun -rdd</code>	初始构型 + 10%
约束	r_{con}	<code>mdrun -rcon</code>	由键长估计
虚拟位点	r_{con}	<code>mdrun -rcon</code>	0

在大多数情况下, `mdrun` 的默认设置不会导致模拟停止并给出缺失相互作用的出错信息. 键合相互作用的范围由初始构型中键合的电荷组之间的距离, 再加上10%确定. 对约束, r_{con} 的值由(`lincs_order + 1`)个键所能覆盖的最大距离决定, 并且所有这些键以120度的角相连接. 实际的约束通讯并不受 r_{con} 限制, 而是受最小晶胞尺寸 L_C 限制:

$$L_C \geq \max(r_{\text{mb}}, r_{\text{com}}) \quad (3.147)$$

不使用动态负载均衡, 并使用压力缩放时, 实际上允许体系的缩放超过这个限制. 注意, 对于三斜盒子, L_C 并不是简单地取盒子对角分量与此方向晶胞数目的比值, 而是三斜晶胞边界之间的最小距离. 对于菱形十二面体盒子, 此距离是沿 x 和 y 方向长度的 $\sqrt{3}/2$.

当 $r_{\text{mb}} > r_c$ 时, `mdrun` 会采用一个灵巧的算法以减少通讯. 如果只是简单地通讯 r_{mb} 之内的所有电荷组, 这将大大增加通讯次数. 因此只会对那些与不在局部出现的电荷组之间存在键合相互作用连接的电荷组进行通讯. 这将导致很少的额外通讯, 也会稍微增加区域分解设置的成本. 在某些情况下, 例如粗粒化模拟中使用非常短的截断, 可能需要手动设置 r_{mb} 以减小这个成本.

3.17.5 多程序多数据PME并行

静电相互作用是长程的, 因此需要使用特殊的算法来避免对许多原子对进行加和. 在GROMACS中通常使用的方法是PME(参见4.8.2节). 由于在PME算法中, 所有粒子彼此之间都存在相互作用, 因此需要全局通讯. 这通常就是区域分解标度行为的限制因素. 为了减小这个问题带来的影响, 我们提出了多程序多数据(Multiple-Program, Multiple-Data) 方法[5]. 在这个方法中, 我们选择一些队列只进行PME网格计算, 而其他被称为粒子-粒子(PP)队列的队列则用于完成所有剩余的其他任务. 对于长方体盒子, PP队列与PME队列的最佳比例通常是3:1, 而对于菱形十二面体盒子最佳比例是2:1. 当PME队列的数量减少为原来的1/4时, 通讯调用的次数大约减少到原来的1/16. 或者换句话说, 现在我们可以使用4倍多的队列. 另外, 对于网络中4核或8核的现代设备来说, PME的有效网络带宽将提高4倍, 因为在PME计算中, 每个机器上只有四分之一的核在使用网络连接.

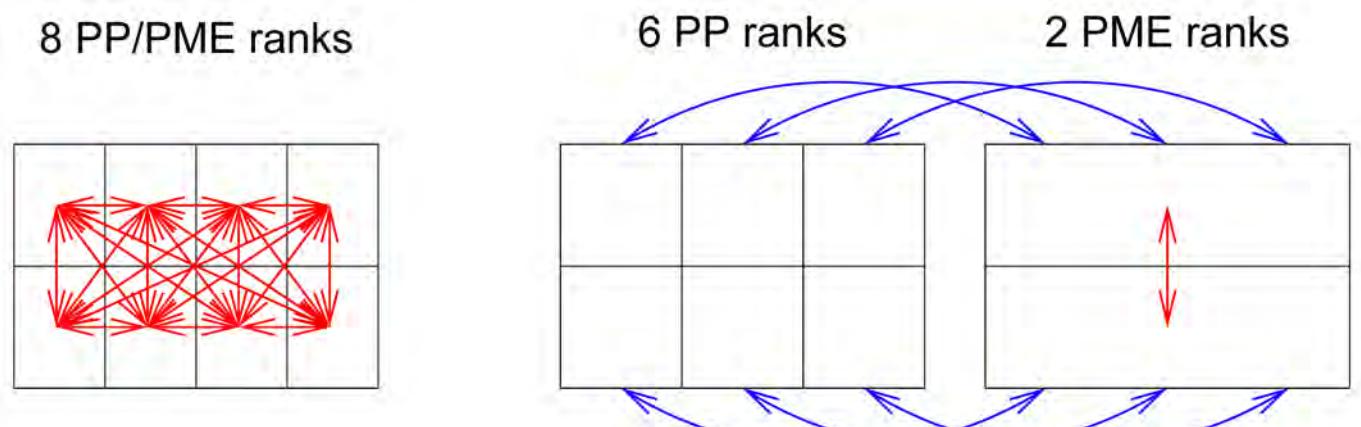


图3.15: 8个队列不使用MPMD(左边)和使用MPMD(右边)的例子. 左边的PME通讯(红箭头)比右边的多得多. 对MPMD算法, 虽然需要额外的PP-PME坐标和力通讯(蓝箭头), 但是通讯复杂度更低.

`mdrun` 默认会交错运行PP和PME队列. 如果设备中的队列没有依次编号, 可以使用 `mdrun -ddorder pp_pme` 命令. 对带有真正的

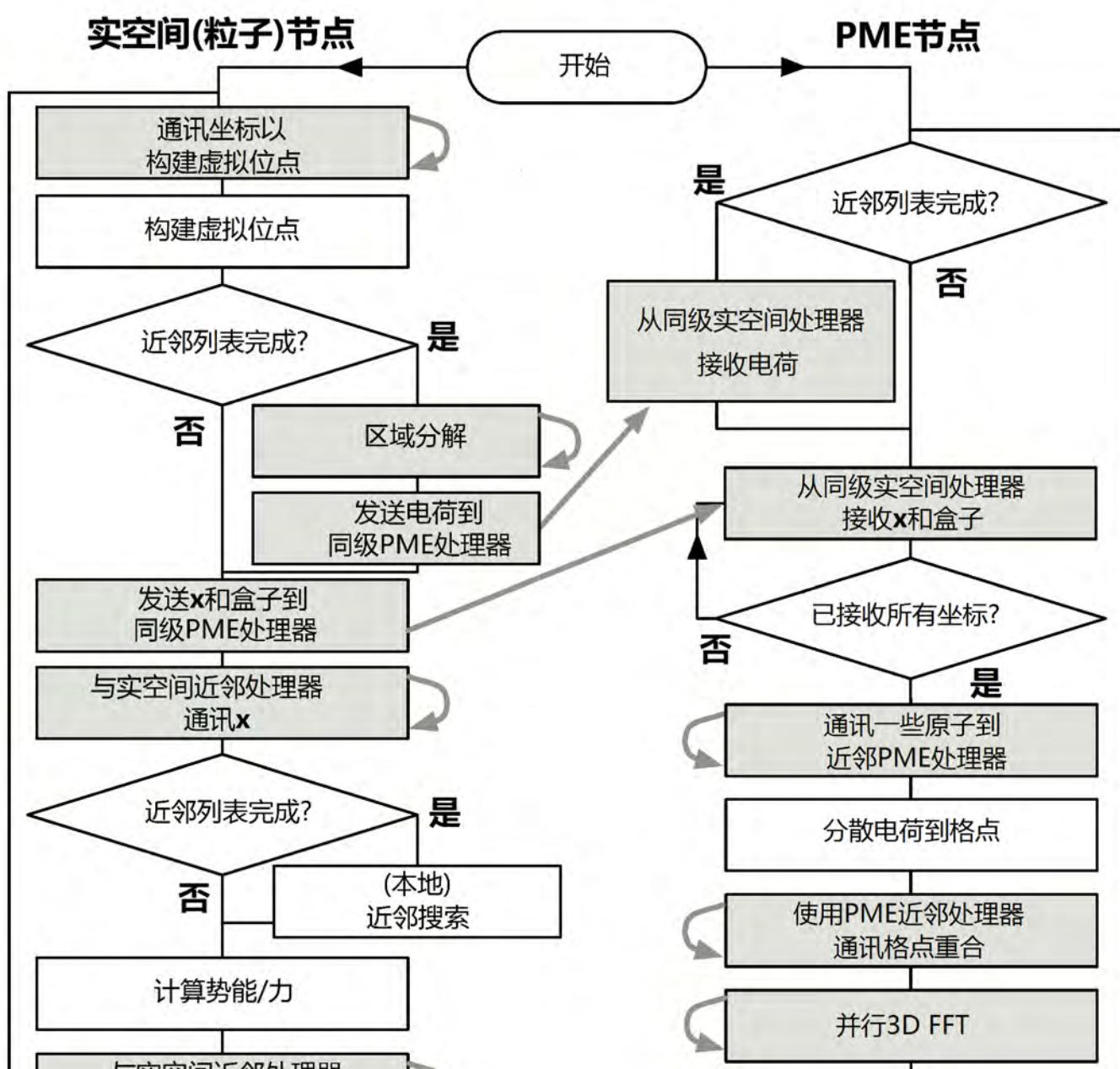
三维环及相应的合适的队列分配通讯软件的设备, 我们应该使用 `mdrun -ddorder cartesian` 命令.

为了优化性能, 我们通常应该设置截断和PME格点使得PME负载达到总计算负载的25%到33%. `grompp`会在末尾输出此负载的估计值, 并且 `mdrun` 也会计算相同的估计值以确定使用的PME队列的最佳数目. 对高度并行化计算, 利用 `mdp` 设置和/或由 `mdrun` 的 `-npme` 选项设定的PME队列数目来优化PME负载是值得的. 对于改变静电设置, 了解当对库仑截断和PME格点间距使用相同的比例进行缩放时, 静电的精度几乎保持不变, 对我们是很有用的. 注意, 由于PME队列数目比PP队列数目的小, 所以通常在我们估算PME队列数目的时候最好多算一些, 这样可以减少总的等待时间.

PME区域分解可以是沿 x 和/或 y 方向的1D或2D分解. 因为在高度并行化中, 2D分解的区域形状和铅笔很相似, 所以2D分解也被称为铅笔分解. 只有当PP分解沿 x 方向有唯一一个区域时, 才能使用沿 y 方向的1D分解. 2D PME分解沿 x 方向的区域数目必须等于PP分解数目. `mdrun`会自动选择1D或2D PME分解(当可以使用给定的全部队列数时), 基于PME中坐标再分布通讯加上格点重叠和转置通讯的最小数目. 为了避免PP和PME队列之间坐标和力的多余通讯, 理想情况下 x 方向上的DD晶胞数目应等于PME队列的数目或其倍数. 默认情况下, `mdrun`会考虑这些情况.

3.17.6 区域分解流程图

流程图3.16显示了区域分解中不同算法所有可能的通讯. 对于简单点的模拟, 流程图也一样, 只是不包含未使用的算法及与其相应的通讯.



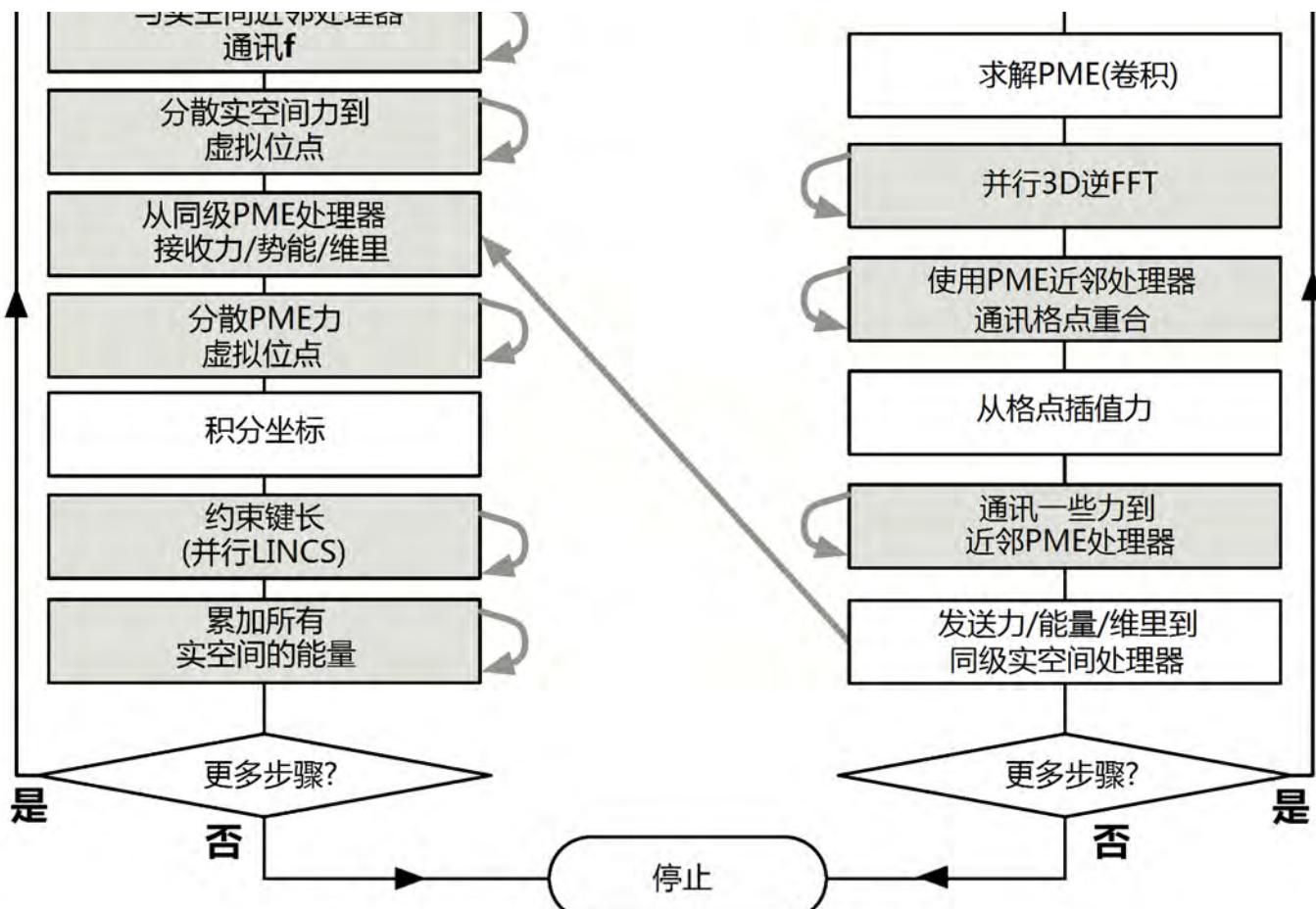


图 3.16: 标准MD模拟中算法和通讯(箭头)的流程图, 模拟使用了虚拟位点, 约束和独立PME网格队列.

3.18 隐式溶剂模型

隐式溶剂模型提供了一种表示溶剂分子静电效应的有效方式, 同时在分子动力学模拟中避免了大量计算, 这些计算涉及对周围水分子进行精确的, 溶剂化的描述. 相比于显式溶剂模型, 隐式溶剂模型有以下几个优点: 不需要对溶质周围的水分子进行平衡; 不存在粘度, 蛋白质能够更快地探索构型空间.

GROMACS中的隐式溶剂计算利用广义Born 公式, Born 半径的计算可使用Still[70], HCT[71] 和OBC[72]模型.

溶剂化自由能 G_{solv} 是三项的加和, 溶剂-溶剂空穴项(G_{cav}), 溶剂-溶质范德华项(G_{vdw}), 还有溶剂-溶质静电极化项(G_{pol}).

G_{cav} 和 G_{vdw} 的总和对应于移除所有电荷的分子的(未极化)溶剂化自由能, 通常称作 G_{np} , 等于总的溶剂可及表面积乘上表面张力. 因此溶剂化自由能的总表达式可写为:

$$G_{\text{solv}} = G_{\text{np}} + G_{\text{pol}} \quad (3.148)$$

对于广义Born 模型, G_{pol} 可由广义Born 方程计算[70]:

$$G_{\text{pol}} = \left(1 - \frac{1}{\epsilon}\right) \sum_{i=1}^n \sum_{j>i}^n \frac{q_i q_j}{\sqrt{r_{ij}^2 + b_i b_j \exp\left(\frac{-r_{ij}^2}{4b_i b_j}\right)}} \quad (3.149)$$

在GROMCAS中, 我们引入了代换[73]:

$$c_i = \frac{1}{\sqrt{b_i}} \quad (3.150)$$

这样当计算每项相互作用时, 可通过简单变换得到新变量 x :

$$x = \frac{r_{ij}}{\sqrt{b_i b_j}} = r_{ij} c_i c_j \quad (3.151)$$

最后, 改写后3.149的完整的公式如下:

$$G_{\text{pol}} = \left(1 - \frac{1}{\epsilon}\right) \sum_{i=1}^n \sum_{j < i}^n \frac{q_i q_j}{\sqrt{b_i b_j}} \xi(x) = \left(1 - \frac{1}{\epsilon}\right) \sum_{i=1}^n q_i c_i \sum_{j < i}^n q_j c_j \xi(x) \quad (3.152)$$

方程3.148的非极化部分(G_{np})直接根据每个原子的Born 半径计算, 计算时使用了Schaefer 等人的简单ACE类型近似[74], 包含了对所有原子的一个简单循环. 这只需要一个额外的溶剂化参数, 与原子类型无关, 但三种Born 半径模型之间有稍微的差别.

-
1. 注意, 在一些推导, 会使用另一种符号 $\xi_{\text{alt}} = v_\xi = p_\xi / Q$. ↪
 2. GROMACS中的盒矩阵表示 \mathbf{b} 对应于Nose和Klein 论文中盒矩阵表示 \mathbf{h} 的转置. 因此, 我们的一些方程看起来与原论文中的略有不同. ↪

第四章 相互作用函数和力场

为了能够使用一些流行力场中采用的势能函数(见4.10节), GROMACS提供了一些势能函数, 包括非键相互作用以及二面角的相互作用. 我们将在适当的小节中对它们予以讨论.

势能函数可以被划分成三个部分:

1. 非键项: Lennard-Jones 或Buckingham, 以及库仑或修正的库仑. 非键相互作用的计算基于已移除了排除项的邻区列表(一定半径内非键原子的列表).
2. 键合项: 共价键伸缩, 键角弯曲, 反常二面角, 和正常二面角. 这些项的计算是基于固定的列表.
3. 约束项: 位置约束, 角度约束, 距离约束, 方向约束和二面角约束, 均基于固定的列表进行计算.

4.1 非键相互作用

GROMACS的非键相互作用是对势累加的(pair-additive), 并满足中心对称:

$$V(\mathbf{r}_1, \dots, \mathbf{r}_N) = \sum_{i < j} V_{ij}(\mathbf{r}_{ij}) \quad (4.1)$$

$$\mathbf{F}_i = - \sum_j \frac{dV_{ij}(r_{ij})}{dr_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}} = -\mathbf{F}_j \quad (4.2)$$

非键相互作用包含排斥项, 色散项和库仑项. 组合起来的排斥项和色散项可以取Lennard-Jones(或6-12相互作用)或 Buckingham(或exp-6势)形式. 此外, (部分)带电的原子之间的非键相互作用通过库仑项表达.

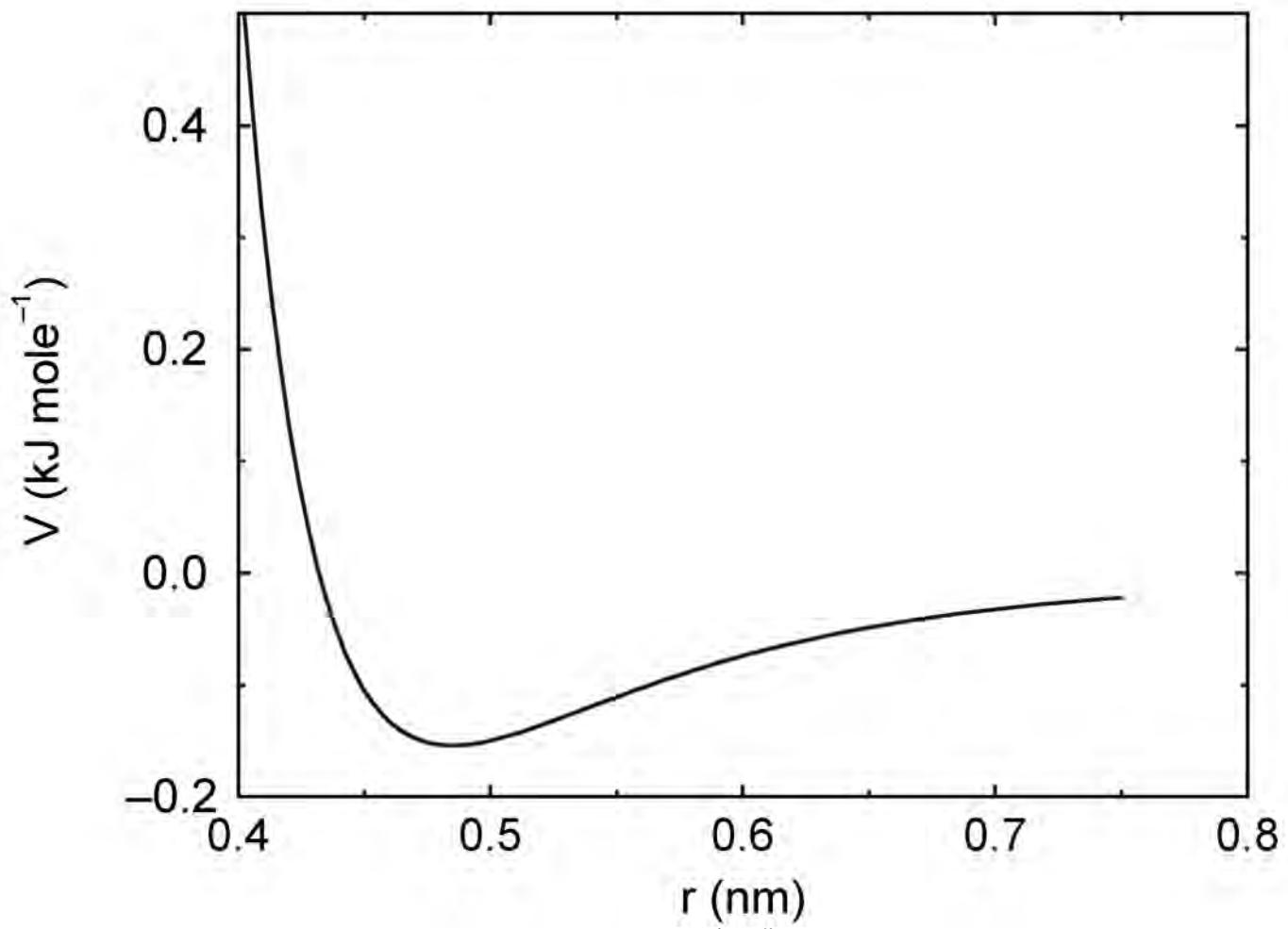


图4.1: Lennard-Jones 相互作用

4.1.1 Lennard-Jones相互作用

两个原子之间Lennard-Jones 势 $V_{LJ}(r_{ij})$ 等于:

$$V_{LJ}(r_{ij}) = \frac{C_{ij}^{(12)}}{r_{ij}^{12}} - \frac{C_{ij}^{(6)}}{r_{ij}^6} \quad (4.3)$$

其函数曲线如图4.1所示. 参数 $C_{ij}^{(12)}$ 和 $C_{ij}^{(6)}$ 取决于配对的 原子类型; 因此来源于LJ-参数矩阵. 在Verlet截断方案中, 通过将势能移动一个恒量, 可使其在截断距离处为零.

此势能函数的力是:

$$\mathbf{F}_i(r_{ij}) = \left(12 \frac{C_{ij}^{(12)}}{r_{ij}^{13}} - 6 \frac{C_{ij}^{(6)}}{r_{ij}^7} \right) \frac{\mathbf{r}_{ij}}{r_{ij}} \quad (4.4)$$

LJ势也可以写成以下形式:

$$V_{LJ}(r_{ij}) = 4\varepsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right) \quad (4.5)$$

在构造非键LJ-参数的参数矩阵时, GROMACS可使用两种不同的组合规则, 只采用几何平均值 (力场文件输入部分的类型1) :

$$\begin{aligned} C_{ij}^{(6)} &= \left(C_{ii}^{(6)} C_{jj}^{(6)} \right)^{1/2} \\ C_{ij}^{(12)} &= \left(C_{ii}^{(12)} C_{jj}^{(12)} \right)^{1/2} \end{aligned} \quad (4.6)$$

或采用Lorentz-Berthelot 规则. 此规则利用算术平均计算 σ_{ij} , 利用几何平均计算 ε_{ij} (类型2) :

$$\begin{aligned} \sigma_{ij} &= \frac{1}{2} (\sigma_{ii} + \sigma_{jj}) \\ \varepsilon_{ij} &= (\varepsilon_{ii} \varepsilon_{jj})^{1/2} \end{aligned} \quad (4.7)$$

最后, 可以对两个参数都采用几何平均值 (类型3) :

$$\begin{aligned} \sigma_{ij} &= (\sigma_{ii} \sigma_{jj})^{1/2} \\ \varepsilon_{ij} &= (\varepsilon_{ii} \varepsilon_{jj})^{1/2} \end{aligned} \quad (4.8)$$

OPLS力场就采用这个规则.

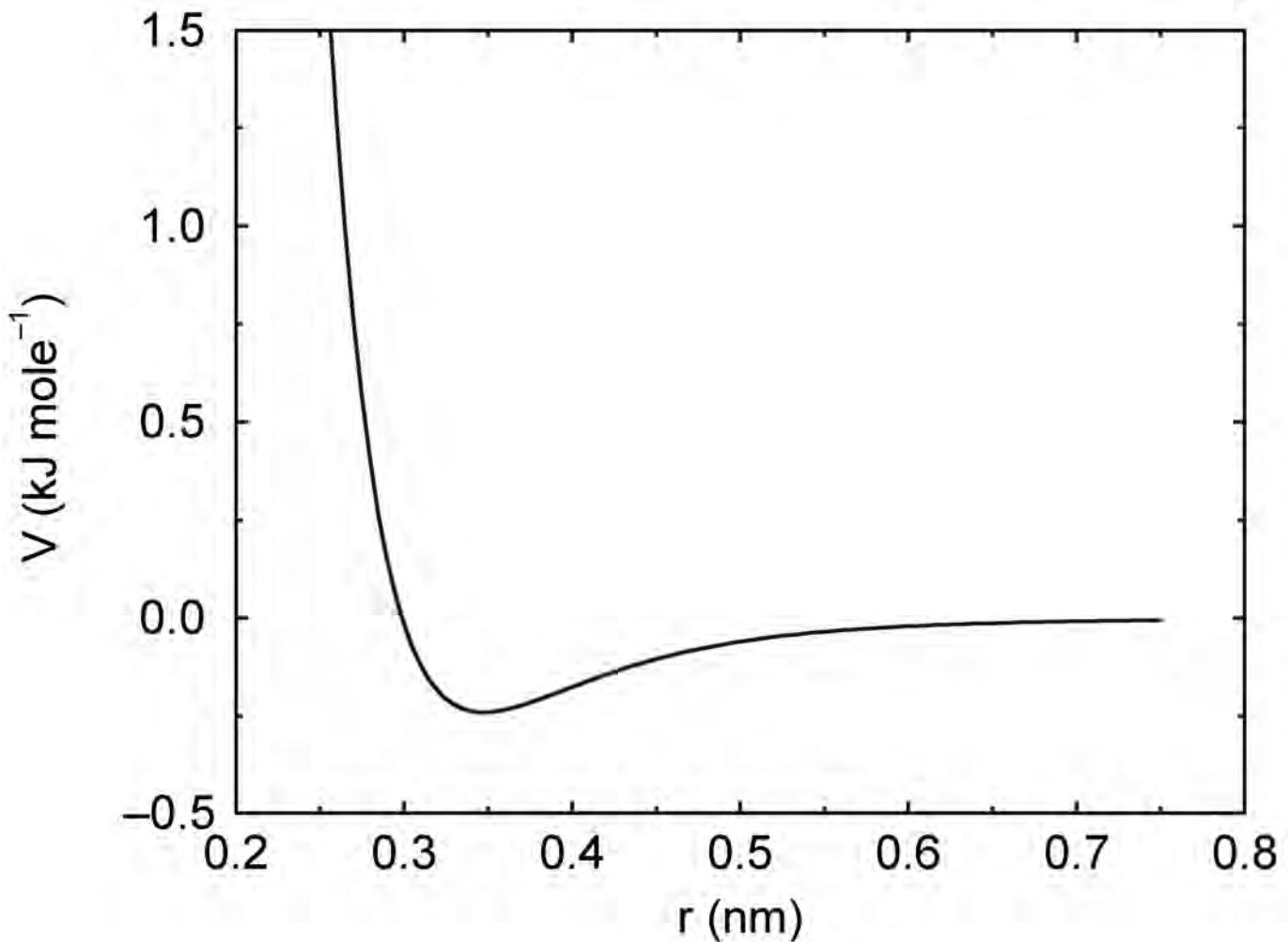


图4.2: Buckingham 相互作用

4.1.2 Buckingham势

与Lennard-Jones 势相比, Buckingham 势的排斥项更加灵活, 贴近实际, 但其计算也更昂贵. Buckingham 势的形式是:

$$V_{bh}(r_{ij}) = A_{ij} \exp(-B_{ij}r_{ij}) - \frac{C_{ij}}{r_{ij}^6} \quad (4.9)$$

请参见图4.2. 此势能函数的力为:

$$\mathbf{F}_i(r_{ij}) = \left[A_{ij} B_{ij} \exp(-B_{ij} r_{ij}) - 6 \frac{C_{ij}}{r_{ij}^7} \right] \frac{\mathbf{r}_{ij}}{r_{ij}} \quad (4.10)$$

4.1.3 库仑相互作用

两个带电粒子之间的库仑相互作用由下式给出:

$$V_c(r_{ij}) = f \frac{q_i q_j}{\varepsilon_r r_{ij}} \quad (4.11)$$

参见图4.3, 其中 $f = \frac{1}{4\pi\varepsilon_0} = 138.935485$ (见第2章)

此势对应的力为:

$$\mathbf{F}_i(r_{ij}) = f \frac{q_i q_j}{\varepsilon_r r_{ij}^2} \frac{\mathbf{r}_{ij}}{r_{ij}} \quad (4.12)$$

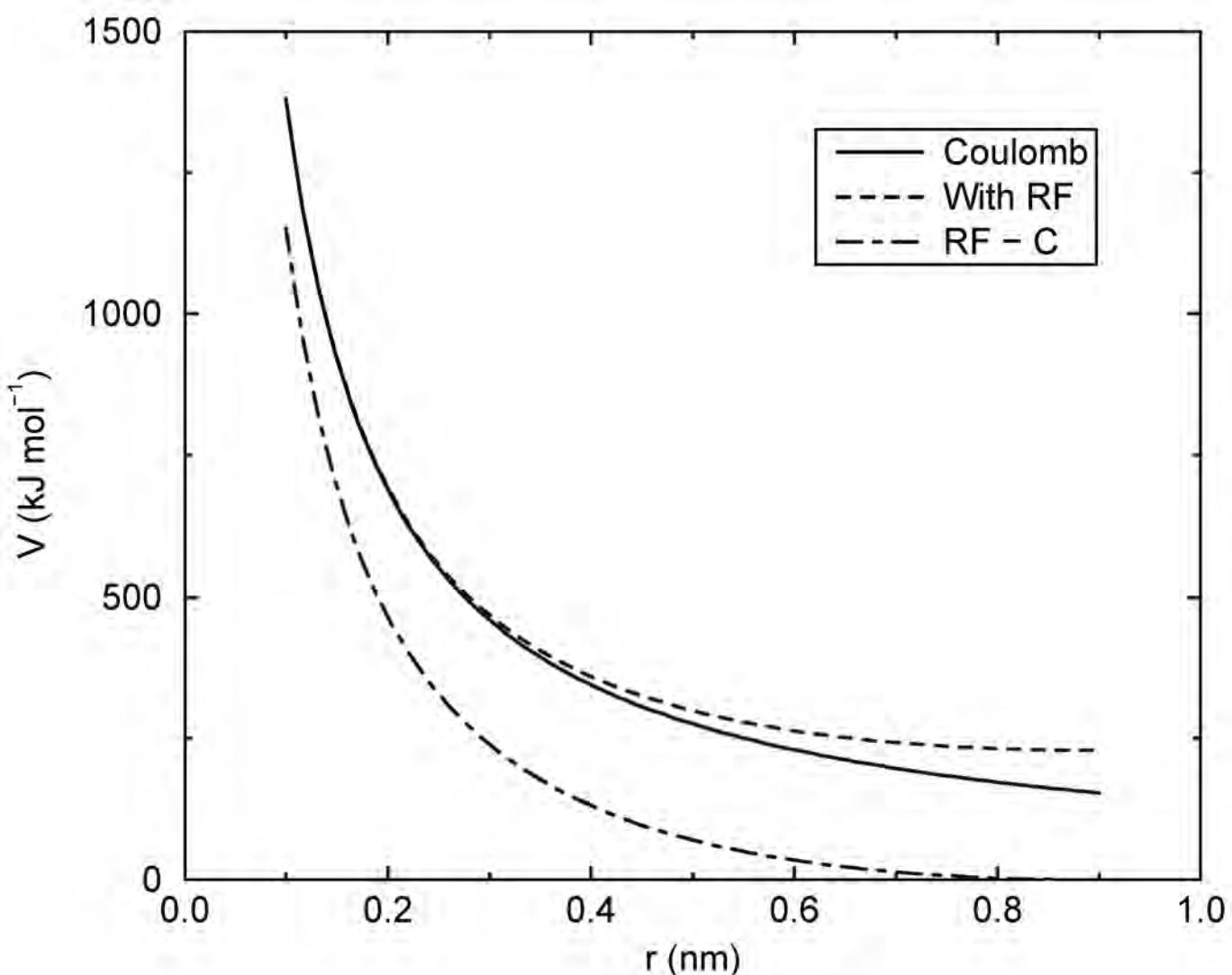


图4.3: 反应场与非反应场的库仑相互作用 (带同种电荷的粒子之间). 在非反应场的情况下 ε_r 为1, ε_{rf} 为78, r_c 为0.9纳米. 除相差一个常数外, 点划线与虚线完全相同.

使用普通的库仑相互作用时要么不能截断, 要么所有粒子对都处于截断半径范围内, 因为力在截断处存在跃变, 而且变化很大. 如果你确实需要使用截断, 可以将势能进行移动以使势能等于力的积分. 在基于组的截断方案中, 这种移动只适用于非排除对. 在Verlet截断方案中, 移动也适用于排除对和自相互作用, 这使得势能等同于 $\varepsilon_{rf} = 1$ 的反应场 (见下文).

GROMACS中相对介电常数 ε_r 可在 **grompp** 的输入中设置.

4.1.4 反应场库仑作用

对均相体系, 可通过假定截断半径 r_c 以外的粒子周围是介电常数为 ε_{rf} 的恒定介电环境, 对库仑相互作用进行校正. 此时, 库仑相互作用可写为:

$$V_{crf} = f \frac{q_i q_j}{\varepsilon_r r_{ij}} \left[1 + \frac{\varepsilon_{rf} - \varepsilon_r}{2\varepsilon_{rf} + \varepsilon_r} \frac{r_{ij}^3}{r_c^3} \right] - f \frac{q_i q_j}{\varepsilon_r r_c} \frac{3\varepsilon_{rf}}{2\varepsilon_{rf} + \varepsilon_r} \quad (4.13)$$

上式右边的常量表达式使得在截断处势能为零. 对带电的截断球体, 上式对应于均匀背景电荷的中性化. 我们可以将方程4.13简写为

$$V_{crf} = f \frac{q_i q_j}{\varepsilon_r} \left[\frac{1}{r_{ij}} + k_{rf} r_{ij}^2 - c_{rf} \right] \quad (4.14)$$

其中,

$$k_{rf} = \frac{1}{r_c^3} \frac{\varepsilon_{rf} - \varepsilon_r}{(2\varepsilon_{rf} + \varepsilon_r)} \quad (4.15)$$

$$c_{rf} = \frac{1}{r_c} + k_{rf} r_c^2 = \frac{1}{r_c} \frac{3\varepsilon_{rf}}{(2\varepsilon_{rf} + \varepsilon_r)} \quad (4.16)$$

对大的 ε_{rf} , k_{rf} 趋近 $r_c^{-3}/2$, 而当 $\varepsilon_{rf} = \varepsilon_r$ 时校正消失. 图4.3中画出了校正的相互作用, 很显然, 对 r_{ij} 的导数(即负的力)在截断距离处为零. 这种势函数对应的力为:

$$\mathbf{F}_i(\mathbf{r}_{ij}) = f \frac{q_i q_j}{\varepsilon_r} \left[\frac{1}{r_{ij}^2} - 2k_{rf} r_{ij} \right] \frac{\mathbf{r}_{ij}}{r_{ij}} \quad (4.17)$$

反应场校正也可以用于所有的排除原子对, 包括自身对, 在这种情况下, 方程4.13和4.17中不出现正常的库仑项.

Tironi 等引入了更一般的反应场理论, 其截断半径 r_c 以外的电介质也具有离子强度 I [75]. 这种情况下, 我们可以利用逆 Debye 屏蔽长度 κ , 将常数 k_{rf} 和 c_{rf} 重写为

$$\kappa^2 = \frac{2I F^2}{\varepsilon_0 \varepsilon_{rf} R T} = \frac{F^2}{\varepsilon_0 \varepsilon_{rf} R T} \sum_{i=1}^K c_i z_i^2 \quad (4.18)$$

$$k_{rf} = \frac{1}{r_c^3} \frac{(\varepsilon_{rf} - \varepsilon_r)(1 + \kappa r_c) + \frac{1}{2}\varepsilon_{rf}(\kappa r_c)^2}{(2\varepsilon_{rf} + \varepsilon_r)(1 + \kappa r_c) + \varepsilon_{rf}(\kappa r_c)^2} \quad (4.19)$$

$$c_{rf} = \frac{1}{r_c} \frac{3\varepsilon_{rf}(1 + \kappa r_c + \frac{1}{2}(\kappa r_c)^2)}{(2\varepsilon_{rf} + \varepsilon_r)(1 + \kappa r_c) + \varepsilon_{rf}(\kappa r_c)^2} \quad (4.20)$$

其中, F 是法拉第常数, R 是理想气体常数, T 是绝对温度, c_i 是物种 i 的摩尔浓度, z_i 是物种 i 的电荷数, 共有 K 种不同的物种. 在离子强度为零的极限情况下, 方程4.19和4.20分别退化为简单形式的方程4.15和4.16.

4.1.5 修正的非键相互作用

在GROMACS中, 可以利用移位函数对非键势能函数进行修改, 其目的在于用连续的力取代截断的力, 且使其在截断半径处具有连续的导数. 使用这种力时, 每步积分引入的误差更小, 且避免了一些复杂情况, 比如在截断过程中由偶极产生的电荷. 实际上,

使用移位力时, 不需要对电荷组创建邻近列表. 然而, 移位函数使库仑势产生了一个相当大的变形. 除非对“丢失”的远程势进行了正确的计算以及加和 (通过使用PPPM, Ewald 或PME), 否则必须仔细评估这些修改的效果. 对Lennard-Jones 色散项和排斥项的修改较小, 但它确实消除了截断效应引起的噪声.

切换函数 (将势能乘上一个函数) 和移位函数 (将势能或力加上一个函数) 之间 没有 本质区别[76]. 切换函数是移位函数的一种特殊情况. 将切换函数应用于与粒子 j 作用在粒子 i 上的静电力或范德华力有关的 力函数 $F(r)$:

$$\mathbf{F}_i = cF(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}} \quad (4.21)$$

对于纯的库仑或Lennard-Jones 相互作用, $F(r) = F_\alpha(r) = r^{-\alpha+1}$. 移位力 $F_s(r)$ 一般可写为:

$$\begin{aligned} F_s(r) &= F_\alpha(r) & r < r_1 \\ F_s(r) &= F_\alpha(r) + S(r) & r_1 \leq r \leq r_c \\ F_s(r) &= 0 & r_c \leq r \end{aligned} \quad (4.22)$$

当 $r_1 = 0$ 时, 这是一个通常的移动函数, 否则它就是一个切换函数. 相应的移位库仑势为:

$$V_s(r_{ij}) = f\Phi_s(r_{ij})q_i q_j \quad (4.23)$$

其中, $\Phi(r)$ 是势函数

$$\Phi_s(r) = \int_r^\infty F_s(x)dx \quad (4.24)$$

GROMACS的移位函数在边界处应该是平滑的, 因此移位函数应当满足以下边界条件:

$$\begin{aligned} S(r_1) &= 0 \\ S'(r_1) &= 0 \\ S(r_c) &= -F_\alpha(r_c) \\ S'(r_c) &= -F'_\alpha(r_c) \end{aligned} \quad (4.25)$$

三阶多项式形式

$$S(r) = A(r - r_1)^2 + B(r - r_1)^3 \quad (4.26)$$

即满足这些要求. 常数 A 和 B 可利用 r_c 处的边界条件求出

$$\begin{aligned} A &= -\frac{(\alpha + 4)r_c - (\alpha + 1)r_1}{r_c^{\alpha+2}(r_c - r_1)^2} \\ B &= \frac{(\alpha + 3)r_c - (\alpha + 1)r_1}{r_c^{\alpha+2}(r_c - r_1)^3} \end{aligned} \quad (4.27)$$

因此, 整个力函数为:

$$F_s(r) = \frac{\alpha}{r^{\alpha+1}} + A(r - r_1)^2 + B(r - r_1)^3 \quad (4.28)$$

相应的势函数为:

$$\Phi(r) = \frac{1}{r^\alpha} - \frac{A}{3}(r - r_1)^3 - \frac{B}{4}(r - r_1)^4 - C \quad (4.29)$$

其中

$$C = \frac{1}{r_c^\alpha} - \frac{A}{3}(r_c - r_1)^3 - \frac{B}{4}(r_c - r_1)^4 \quad (4.30)$$

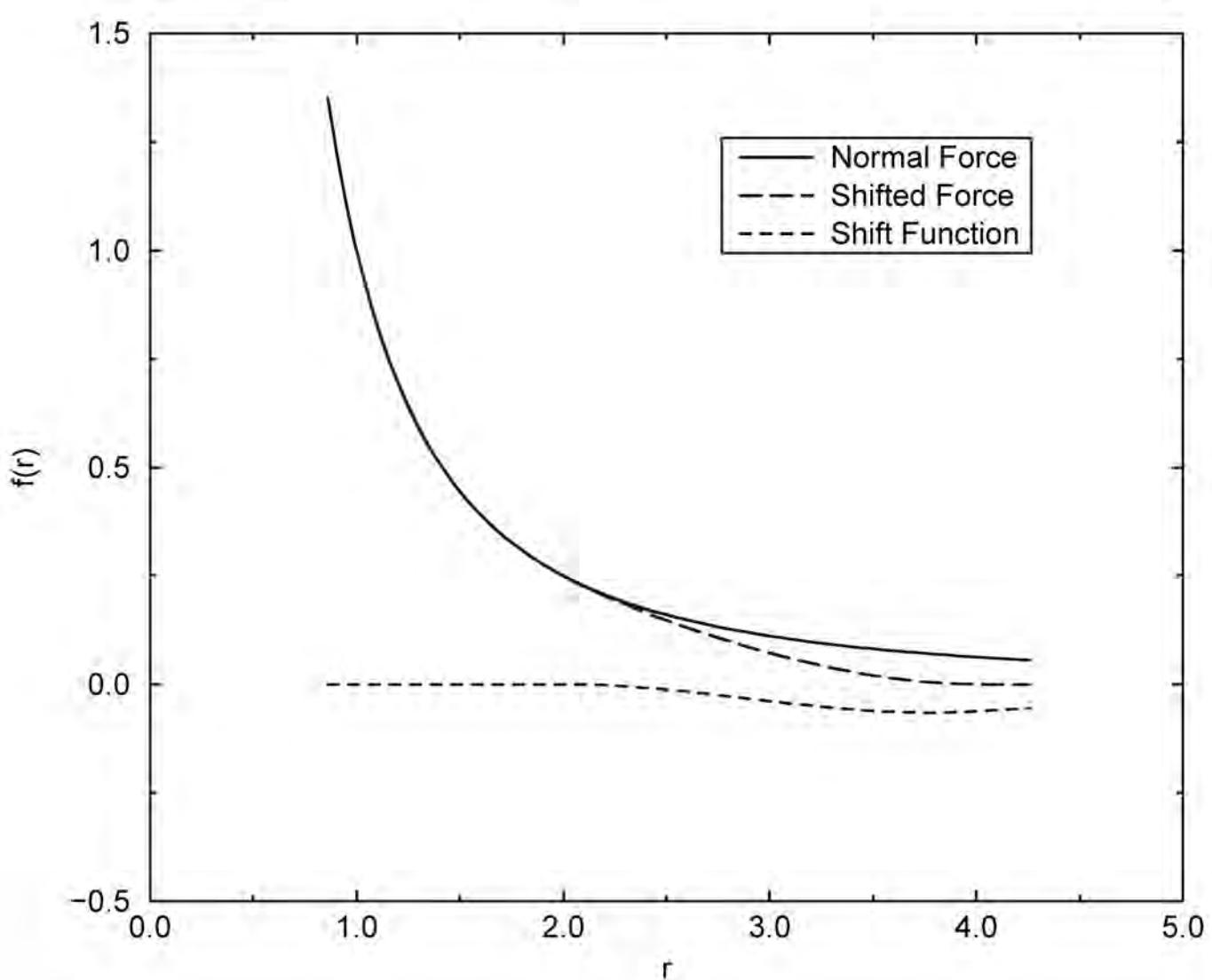


图4.4: 库仑力, 移位力与移位函数 $S(r)$, 其中 $r_1 = 2, r_c = 4$.

当 $r_1 = 0$ 时, 修正的库仑力函数

$$F_s(r) = \frac{1}{r^2} - \frac{5r^2}{r_c^4} + \frac{4r^3}{r_c^5} \quad (4.31)$$

与推荐使用的短程函数加上长程部分的泊松解是相同的[77]. 修正的库仑势函数为:

$$\Phi(r) = \frac{1}{r} - \frac{5}{3r_c} + \frac{5r^3}{3r_c^4} - \frac{r^4}{r_c^5} \quad (4.32)$$

参见图 4.4.

4.1.6 修改的Ewald求和短程相互作用

当利用Ewald求和方法或粒子网格Ewald方法计算长程相互作用时, 短程库仑势必须进行修改, 这与上面切换函数的情形类似. 在这种情况下, 短程势能由下式给出:

$$V(r) = f \frac{\operatorname{erfc}(\beta r_{ij})}{r_{ij}} q_i q_j \quad (4.33)$$

其中, β 参数用于决定直接空间和与倒易空间和之间的相对比重, $\operatorname{erfc}(x)$ 是余误差函数. 关于长程静电相互作用的详细讨论见4.8节.

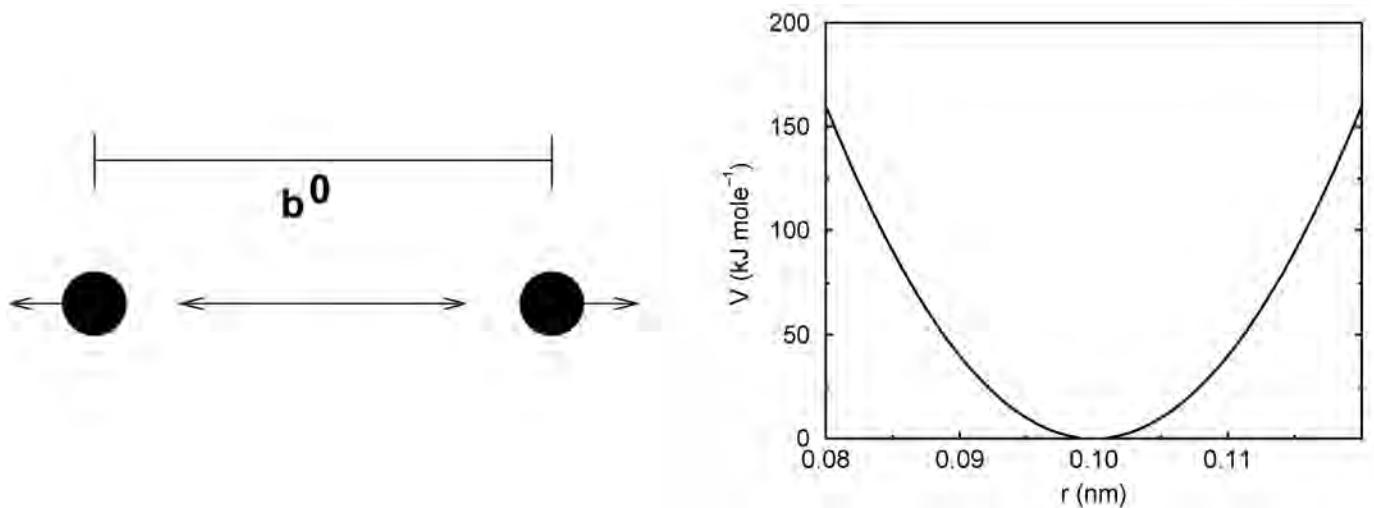


图4.5: 键伸缩的原理(左)以及键伸缩的势能(右).

4.2 键合相互作用

键合相互作用基于固定的原子列表. 它们不仅包括对相互作用(pair interactions), 也包括三体和四体相互作用. 键合相互作用包括键伸缩(双体), 键角(三体)和二面角(四体)相互作用. 一种特殊类型的二面角相互作用(称为异常二面角)可用于将原子保持在同一平面上, 或是避免将分子转变为相反手性(镜像)的结构.

4.2.1 键伸缩

简谐势

两个共价键合原子 i 和 j 之间的键伸缩可利用简谐势进行描述

$$V_b(r_{ij}) = \frac{1}{2} k_{ij}^b (r_{ij} - b_{ij})^2 \quad (4.34)$$

参见图4.5, 相应的力如下:

$$\mathbf{F}_i(\mathbf{r}_{ij}) = k_{ij}^b (r_{ij} - b_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}} \quad (4.35)$$

四次幂势

在GROMOS-96力场[78]中, 为提高计算效率, 共价键势记为以下形式:

$$V_b(r_{ij}) = \frac{1}{4} k_{ij}^b (r_{ij}^2 - b_{ij}^2)^2 \quad (4.36)$$

相应的力为:

$$\mathbf{F}_i(\mathbf{r}_{ij}) = k_{ij}^b (r_{ij}^2 - b_{ij}^2) \mathbf{r}_{ij} \quad (4.37)$$

对这种形式的势函数, 其力常数与通常的简谐力常数 $k^{b,\text{harm}}$ (4.2.1节) 存在如下关系

$$2k^b b_{ij}^2 = k^{b,\text{harm}} \quad (4.38)$$

力常数大多是由GROMOS-87使用的简谐力常数导出[79]. 尽管计算时这种形式效率更高(因为不需要计算平方根), 但它在概念上比较复杂. 它还存在一个特别的缺点, 由于不是简谐势形式, 单个键的平均能量不等于 $\frac{1}{2} kT$, 而在正常的简谐势情况下是 $\frac{1}{2} kT$.

4.2.2 Morse势键伸缩

对需要非简谐键伸缩势的一些体系, GROMACS提供了两个原子 i 和 j 之间的Morse势[80]. 这个势能函数与简谐势的不同在于, 它具有一个不对称的势阱, 且无限远处的力为零. 函数形式是:

$$V_{morse}(r_{ij}) = D_{ij}[1 - \exp(-\beta_{ij}(r_{ij} - b_{ij}))]^2 \quad (4.39)$$

参见图4.6, 相应的力是:

$$\begin{aligned} \mathbf{F}_{morse}(\mathbf{r}_{ij}) &= 2D_{ij}\beta_{ij}r_{ij} \exp(-\beta_{ij}(r_{ij} - b_{ij})) * \\ &[1 - \exp(-\beta_{ij}(r_{ij} - b_{ij}))] \frac{\mathbf{r}_{ij}}{r_{ij}} \end{aligned} \quad (4.40)$$

其中 D_{ij} 是势阱深度, 以 kJ/mol 为单位, β_{ij} 定义了势阱的陡度(以 nm^{-1} 为单位), b_{ij} 为平衡距离(以 nm 为单位). 陡度参数 β_{ij} 可用原子 i 和 j 的约化质量、基本振动频率 μ_{ij} 和势阱深度 D_{ij} 表示:

$$\beta_{ij} = \omega_{ij} \sqrt{\frac{\mu_{ij}}{2D_{ij}}} \quad (4.41)$$

且因为 $\omega = \sqrt{\frac{k}{\mu}}$, 可将 β_{ij} 改用简谐力常数 k_{ij} 表示:

$$\beta_{ij} = \sqrt{\frac{k_{ij}}{2D_{ij}}} \quad (4.42)$$

对于小的偏差 $(r_{ij} - b_{ij})$, 可利用泰勒展开, 取指数项的一阶近似:

$$\exp(-x) \approx 1 - x \quad (4.43)$$

将方程 4.42 和 4.43 代入到函数中:

$$\begin{aligned} V_{morse}(r_{ij}) &= D_{ij}[1 - \exp(-\beta_{ij}(r_{ij} - b_{ij}))]^2 \\ &= D_{ij}[1 - (1 - \sqrt{\frac{k_{ij}}{2D_{ij}}}(r_{ij} - b_{ij}))]^2 \\ &= \frac{1}{2}k_{ij}(r_{ij} - b_{ij})^2 \end{aligned} \quad (4.44)$$

就得到了简谐键伸缩势.

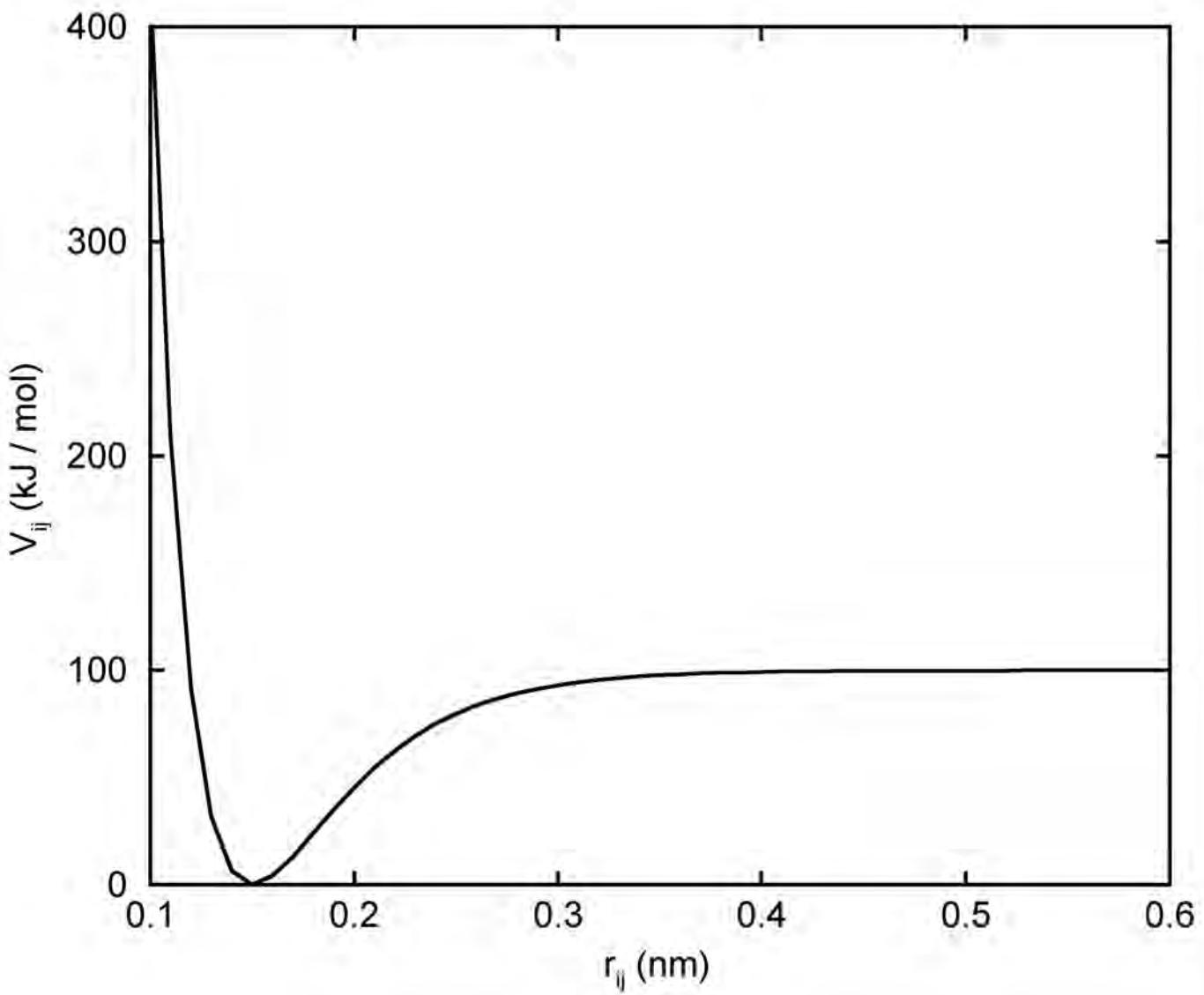


图4.6: Morse势阱, 键长为0.15nm.

4.2.3 立方键伸缩势

另一种比Morse势稍简单的非简谐键伸缩势是在简单的简谐形式上增加了距离的一个立方项:

$$V_b(r_{ij}) = k_{ij}^b (r_{ij} - b_{ij})^2 + k_{ij}^b k_{ij}^{cub} (r_{ij} - b_{ij})^3 \quad (4.45)$$

在Ferguson[82]发展的一个柔性水模型(基于SPC水模型[81])中, OH键使用了立方键伸缩势。该模型给出了合理的红外光谱。GROMACS库中提供了Ferguson水模型([flexwat-ferguson.itp](#))。应当指出,这种势函数是不对称的:过分拉伸导致能量无限降低。因此积分的时间步长被限制为1 fs。

该势对应的力为:

$$\mathbf{F}_i(r_{ij}) = 2k_{ij}^b (r_{ij} - b_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}} + 3k_{ij}^b k_{ij}^{cub} (r_{ij} - b_{ij})^2 \frac{\mathbf{r}_{ij}}{r_{ij}} \quad (4.46)$$

4.2.4 FENE键伸缩势

在聚合物的粗粒化模拟中,珠子之间通常由FENE(finitely extensible nonlinear elastic, 有限扩展非线弹性)势[83]连接:

$$V_{\text{FENE}}(r_{ij}) = -\frac{1}{2} k_{ij}^b b_{ij}^2 \log \left(1 - \frac{r_{ij}^2}{b_{ij}^2} \right) \quad (4.47)$$

这种势能函数看起来很复杂, 但力的表达式更简单:

$$\mathbf{F}_{\text{FENE}}(r_{ij}) = -k_{ij}^b \left(1 - \frac{r_{ij}^2}{b_{ij}^2} \right)^{-1} \mathbf{r}_{ij} \quad (4.48)$$

在短距离处此势不对称地接近力常数为 k^b 的简谐势, 而在距离为 b 时发散.

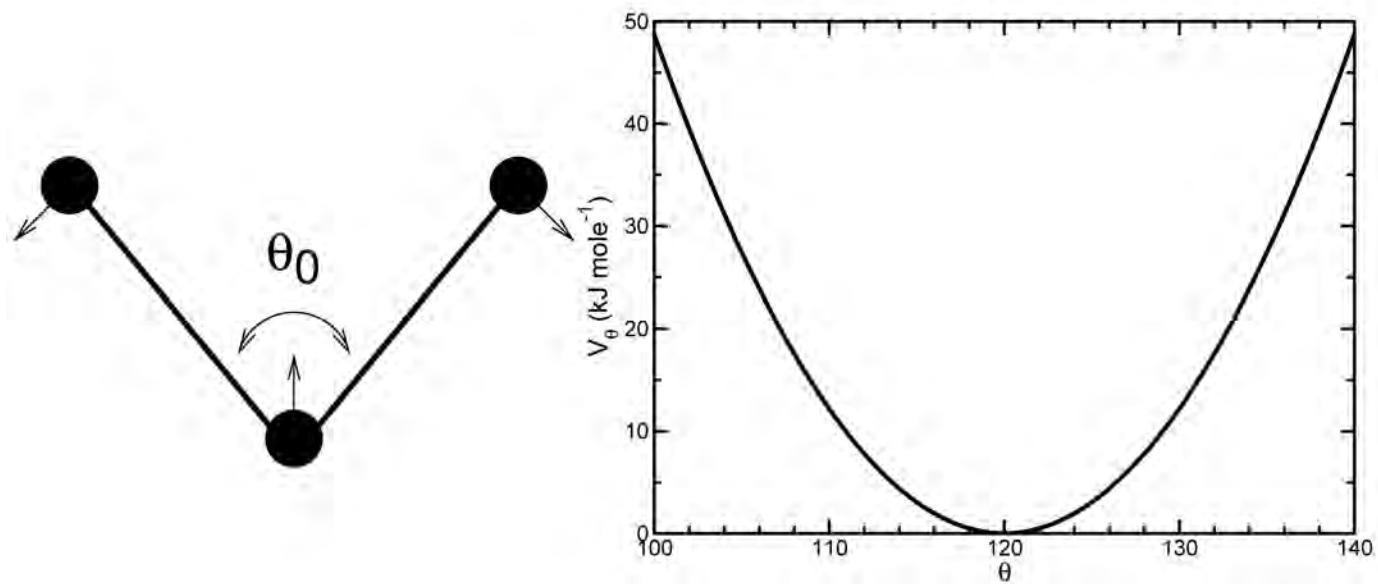


图4.7: 角振动原理 (左) 和键角势 (右).

4.2.5 简谐角势

三个原子 $i - j - k$ 之间的键角振动也可以用键角 θ_{ijk} 的简谐势描述

$$V_a(\theta_{ijk}) = \frac{1}{2} k_{ijk}^\theta (\theta_{ijk} - \theta_{ijk}^0)^2 \quad (4.49)$$

由于键角的振动由简谐势描述, 其形式与键伸缩是一样的 (图4.5) .

利用链式规则, 可得到力的方程:

$$\begin{aligned} \mathbf{F}_i &= -\frac{dV_a(\theta_{ijk})}{dr_i} \\ \mathbf{F}_k &= -\frac{dV_a(\theta_{ijk})}{dr_k} \quad \text{其中 } \theta_{ijk} = \arccos \frac{(\mathbf{r}_{ij} \cdot \mathbf{r}_{kj})}{r_{ij} r_{kj}} \\ \mathbf{F}_j &= -\mathbf{F}_i - \mathbf{F}_k \end{aligned} \quad (4.50)$$

编号 i, j, k 是共价键原子的序列. 原子 j 处于中间; 原子 i 和 k 处于顶端 (见图4.7) . 注意, 在拓扑文件的输入中, 键角以度为单位, 力常数以 $\text{kJ/mol}/\{\text{rad}\}^2$ 为单位.

4.2.6 余弦键角势

GROMOS-96力场利用一个简化的函数表示角振动:

$$V_a(\theta_{ijk}) = \frac{1}{2} k_{ijk}^\theta \left(\cos(\theta_{ijk}) - \cos(\theta_{ijk}^0) \right)^2 \quad (4.51)$$

其中

$$\cos(\theta_{ijk}) = \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{kj}}{r_{ij} r_{kj}} \quad (4.52)$$

相应的力可通过对原子位置的偏微分推导出来。这个函数中的力常数与简谐势的力常数 $k^{\theta,\text{harm}}$ (4.2.5) 存在如下关系

$$k^\theta \sin^2(\theta_{ijk}^0) = k^{\theta,\text{harm}} \quad (4.53)$$

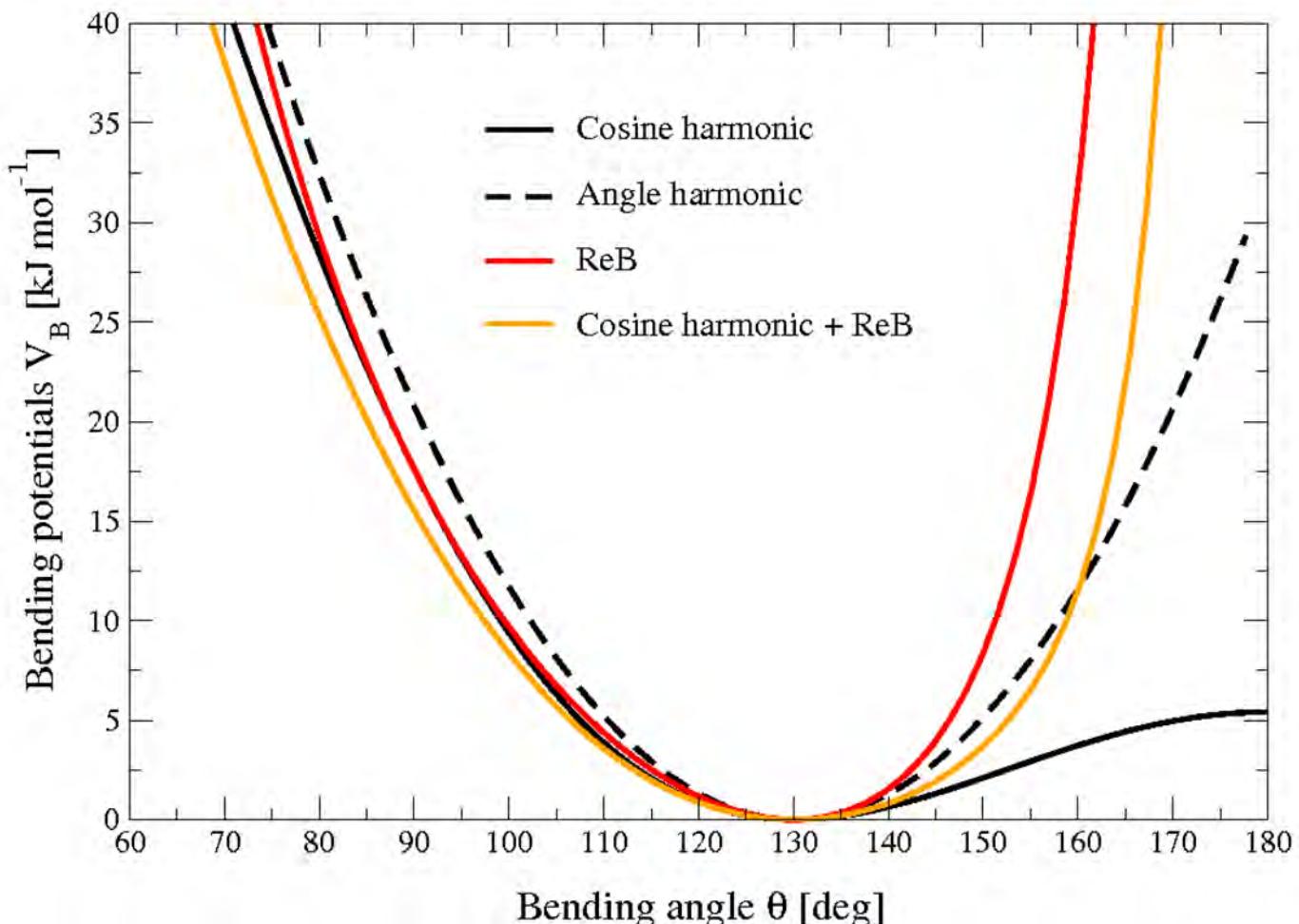


图4.8: 弯曲键角势: 余弦简谐(黑实线), 简谐角(虚黑线)和限制弯曲(红色), 三者具有相同的弯曲常数 $k_\theta = 85 \text{ kJ mol}^{-1}$ 和平衡角 $\theta_0 = 130^\circ$. 橙色线表示余弦简谐($k = 50 \text{ kJ mol}^{-1}$)与限制弯曲($k = 25 \text{ kJ mol}^{-1}$)的和, 二者的 $\theta_0 = 130^\circ$.

在GROMOS-96手册中有一个更加复杂的, 与温度相关的转换公式。这些公式在0 K时是等效的, 300 K时的差异大约为0.1%到0.2%。注意 在拓扑文件的输入中, 角度的单位为度, 力常数的单位为kJ/mol。

4.2.7 受限弯曲势

受限弯曲(ReB)势[84]可防止弯曲角度 θ 达到 180° 。这样, 在进行粗粒分子动力学模拟时, 避免了计算扭转角和势能带来的数值不稳定性。

为了系统地防止弯曲角度达到 180° , 在弯曲势4.51的分母项中引入了 $\sin^2 \theta$ 因子

$$V_{\text{ReB}}(\theta_i) = \frac{1}{2} k_\theta \frac{(\cos \theta_i - \cos \theta_0)^2}{\sin^2 \theta_i} \quad (4.54)$$

图4.8显示了ReB势4.54和标准势4.51之间的比较。ReB势的壁垒在靠近 180° 的区域非常排斥, 因此, 弯曲角能够保持在一定的安全区间内, 远离不稳定性。之所以在分母中使用 $\sin \theta$ 的二次方项是为了保证这种行为, 并且能给出优雅的微分表达式:

$$F_{\text{ReB}}(\theta_i) = \frac{2k_\theta}{\sin^4 \theta_i} (\cos \theta_i - \cos \theta_0)(1 - \cos \theta_i \cos \theta_0) \frac{\partial \cos \theta_i}{\partial \vec{r}_k} \quad (4.55)$$

因其构造, 限制弯曲势不能用于平衡的 θ_0 值太接近 0° 或 180° 的情形 (根据经验, 推荐至少有 10° 的差异). 重要的是, 在初始构型中, 所有的弯曲角度都必须处在安全的区间内, 以避免初始的不稳定性. 这种弯曲势可以与任何形式的扭转势联合使用. 使用它将总是能够避免三个连续的粒子变成共线结构, 因此, 任何扭转势都不会出现奇点. 也可以将它加入到标准的弯曲势中, 以处理约 180° 的角, 但在极小点附近保持它的原始形式 (参见图4.8中的橙色曲线).

4.2.8 Urey-Bradley势

三个原子 $i - j - k$ 之间的Urey-Bradley 键角振动用角 θ_{ijk} 的简谐势和原子 i 和 k 之间的距离的简谐校正项来描述. 尽管可以很容易地将它写成两项的简单加和, 但方便的做法是把它作为拓扑文件中的单独一项, 并在输出文件中作为单独的能量项. 这种势能主要用在CHARMM力场[85]中. 其能量由下式给出:

$$V_a(\theta_{ijk}) = \frac{1}{2} k_{ijk}^\theta (\theta_{ijk} - \theta_{ijk}^0)^2 + \frac{1}{2} k_{ijk}^{UB} (r_{ik} - r_{ik}^0)^2 \quad (4.56)$$

力的方程可从4.2.1和4.2.5节推导出来.

4.2.9 键键交叉项

形成键 $i - j$ 和 $k - j$ 的三个粒子 i, j, k 之间的键键交叉项由下式给出[86]:

$$V_{rr'} = k_{rr'} (|\mathbf{r}_i - \mathbf{r}_j| - r_{1e})(|\mathbf{r}_k - \mathbf{r}_j| - r_{2e}) \quad (4.57)$$

其中 $k_{rr'}$ 是力常数, r_{1e} 和 r_{2e} 分别是键 $i - j$ 和 $j - k$ 的平衡键长. 此势对应的粒子 i 受到的力是:

$$\mathbf{F}_i = -k_{rr'} (|\mathbf{r}_k - \mathbf{r}_j| - r_{2e}) \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|} \quad (4.58)$$

原子 k 受到的力可以通过交换上述方程中 i 和 k 得到. 最后, 原子 j 受到的力遵循内力之和应该为零的事实:

$$\mathbf{F}_j = -\mathbf{F}_i - \mathbf{F}_k.$$

4.2.10 键-角交叉项

为形成键 $i - j$ 和 $k - j$ 的三个粒子 i, j, k 之间的键-角交叉项由下式给出[86]:

$$V_{k\theta} = k_{r\theta} (|\mathbf{r}_i - \mathbf{r}_k| - r_{3e})(|\mathbf{r}_i - \mathbf{r}_j| - r_{1e} + |\mathbf{r}_k - \mathbf{r}_j| - r_{2e}) \quad (4.59)$$

其中 $k_{r\theta}$ 为力常数, r_{3e} 为 $i - k$ 之间的距离, 其他常数的含义与方程4.57中相同. 原子 i 受到的力为:

$$\mathbf{F}_i = -k_{r\theta} \left[(|\mathbf{r}_i - \mathbf{r}_k| - r_{3e}) \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|} + (|\mathbf{r}_i - \mathbf{r}_j| - r_{1e} + |\mathbf{r}_k - \mathbf{r}_j| - r_{2e}) \frac{\mathbf{r}_i - \mathbf{r}_k}{|\mathbf{r}_i - \mathbf{r}_k|} \right] \quad (4.60)$$

4.2.11 四次键角势

键角势可以使用四阶多项式以满足特殊目的

$$V_q(\theta_{ijk}) = \sum_{n=0}^5 C_n (\theta_{ijk} - \theta_{ijk}^0)^n \quad (4.61)$$

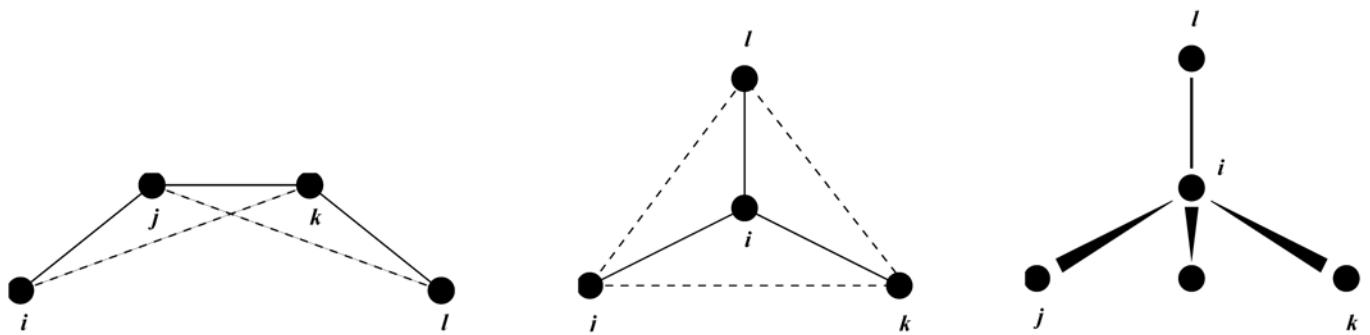


图4.9: 异常二面角原理. 环的面外弯曲(左), 环的取代(中), 四面体偏离(右). 在所有情况下, 异常二面角 ξ 被定义为平面 (i, j, k) 和 (j, k, l) 之间的夹角.

4.2.12 异常二面角

异常二面角是为了保持平面基团(例如芳环)处于平面内, 或者为了防止分子翻转成其镜像分子, 见图4.9.

异常二面角: 简谐型

最简单的异常二面角势是简谐势, 曲线如图4.10.

$$V_{id}(\xi_{ijkl}) = \frac{1}{2} k_\xi (\xi_{ijkl} - \xi_0)^2 \quad (4.62)$$

简谐势函数不连续, 但由于不连续点选择在距离 ξ_0 为 180° 的位置, 这种不连续性永远不会引出麻烦. 注意, 在拓扑输入文件中, 角度的单位为度, 而力常数的单位为 $\text{kJ/mol}/\text{rad}^2$.

异常二面角: 周期型

这种势函数等同于周期性的正常二面角(见下文). 将此种二面角类型(类型4)单独分类只是为了在参数部分和输出中将异常二面角和正常二面角区分开来.

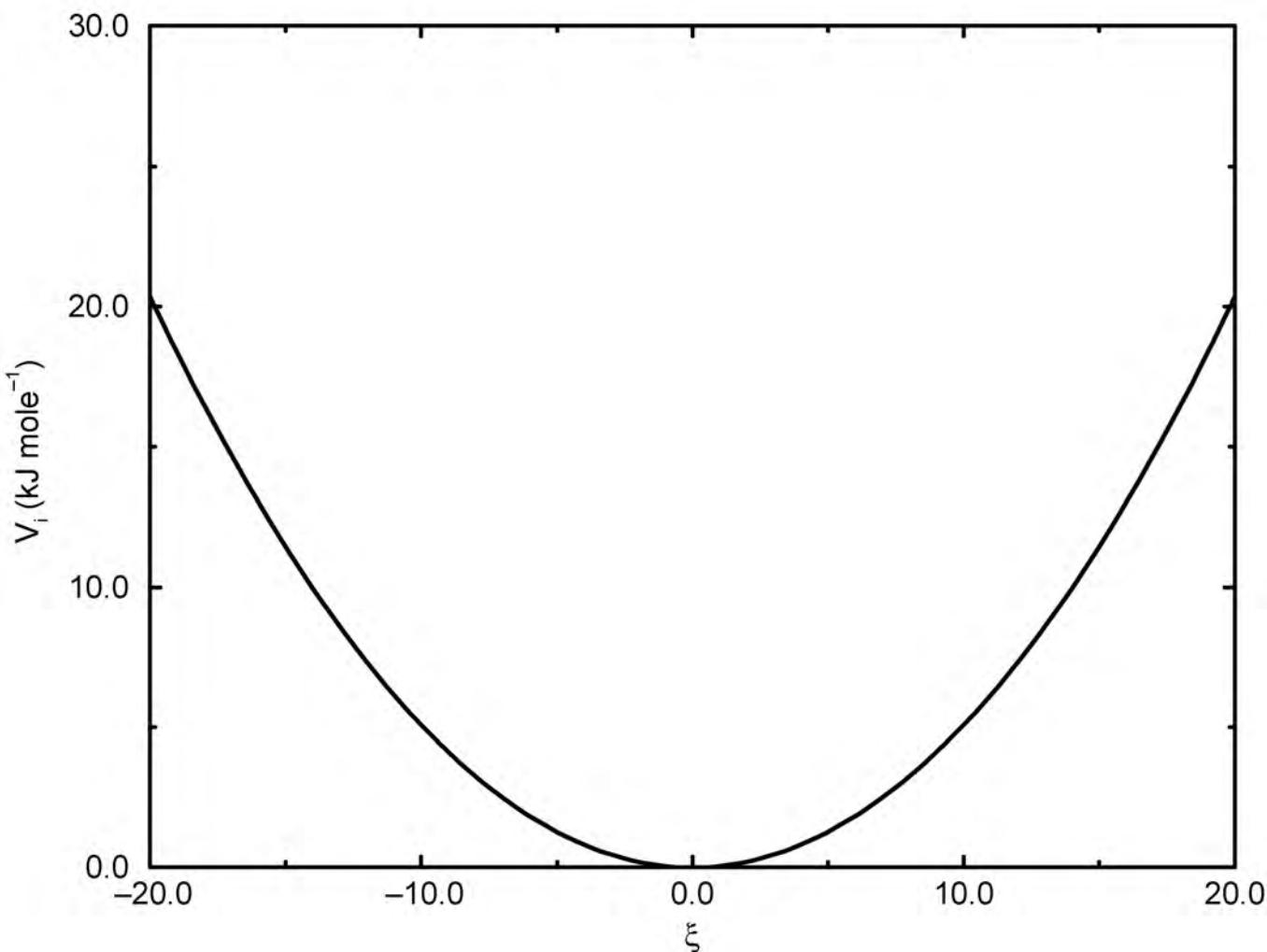


图4.10: 异常二面角的势能

4.2.13 正常二面角

对于正常的二面角相互作用, 可以选择GROMOS的周期性函数或基于 $\cos \phi$ 的幂的多项式展开函数 (被称为Ryckaert-Bellemans 势) . 这种选择使得二面角四个原子中的第一和第四个原子之间的特殊相互作用也被包含在内.

对周期性的GROMOS势, 必须包含特殊的1–4 LJ 相互作用 ; 对 烷烃 的Ryckaert-Bellemans 势, 1–4相互作用必须从非键列表中排除. 注意: Ryckaert-Bellemans 势也可在OPLS等力场中与1–4相互作用联合使用. 因此, 在这种情况下你不应该修改 `pdb2gmx` 生成的拓扑文件.

正常二面角: 周期型

正常二面角的定义遵从IUPAC/IUB约定, 其中, ϕ 是面 ijk 和面 jkl 之间的夹角, 零 相应于 顺式(cis) 构型(i 和 l 处于面的同一侧). GROMACS拓扑文件中有两类二面角函数类型. 标准类型1的行为和其他的键相互作用类似. 对某些力场, 类型9很有用. 当在 `[dihedraltypes]` 部分对同一原子类型定义了多个参数时, 类型9可使多个势能函数自动用于在 `[dihedral]` 部分的一个单一的二面角上.

这种类型的势函数为

$$V_d(\phi_{ijkl}) = k_\phi (1 + \cos(n\phi - \phi_s)) \quad (4.63)$$

正常二面角: Ryckaert-Bellemans势函数

对烷烃, 经常使用以下的正常二面角势 (见图4.12) :

$$V_{rb}(\phi_{ijkl}) = \sum_{n=0}^5 C_n (\cos(\psi))^n \quad (4.64)$$

其中, $\psi = \phi - 180^\circ$.

注意: 从一个约定到另一个约定的转换可以通过将每个系数 C_n 乘上 -1^n .

常数C的一个示例在表4.1中给出.

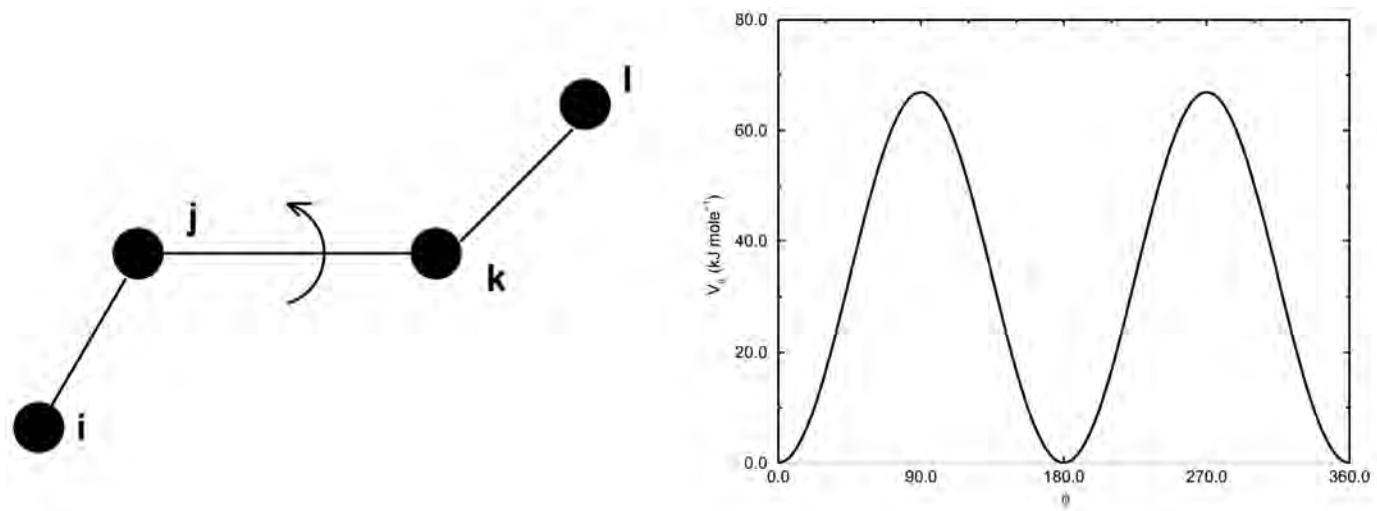


图4.11: 正常二面角的原理 (左, 反式trans构型) 和二面角势 (右)

表4.1: Ryckaert-Bellemaans势的常数值(kJ mol^{-1})

C0 9.28	C2 -13.12	C4 26.24
C1 12.16	C3 -3.06	C5 -31.5

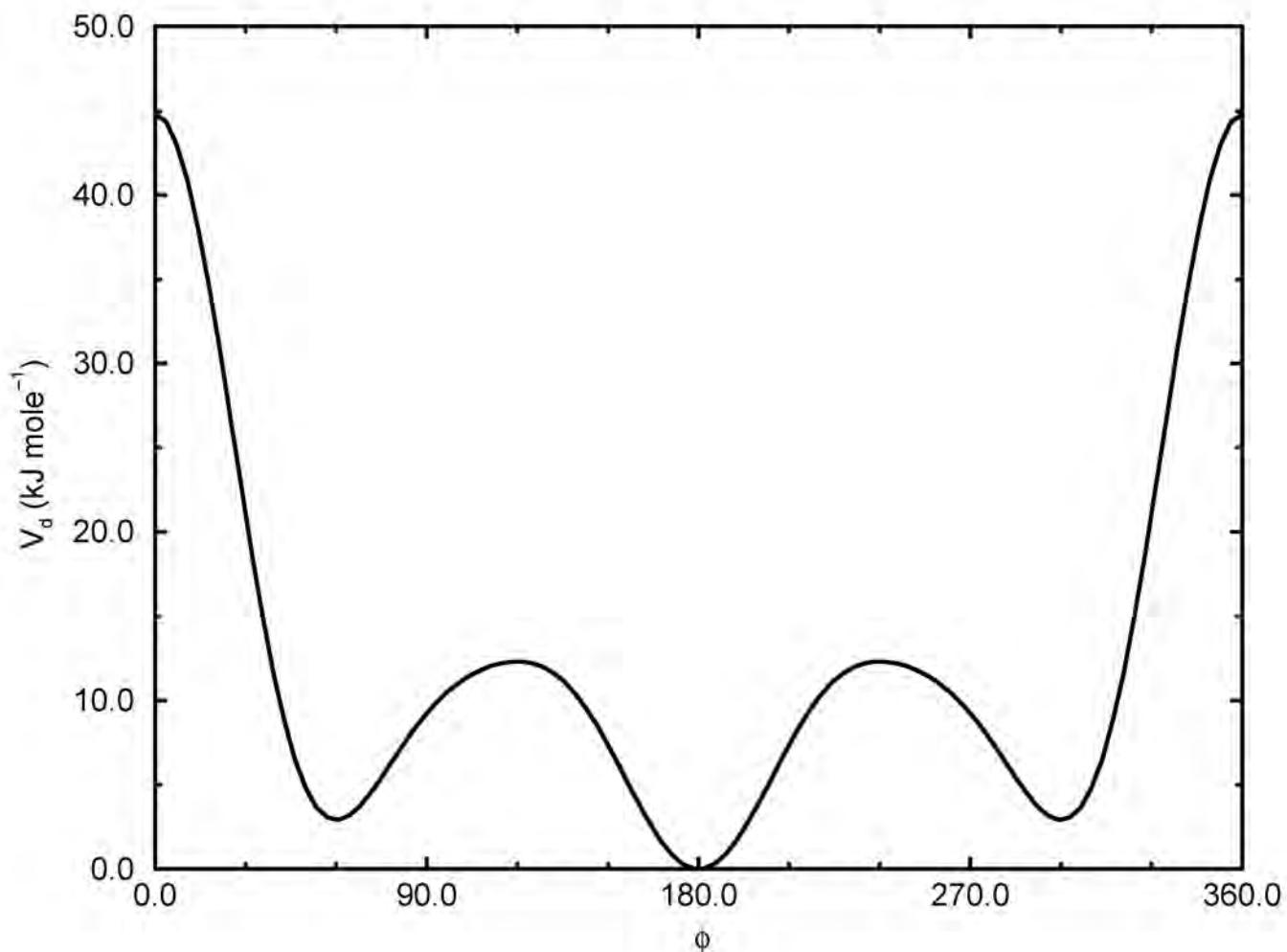


图4.12: Ryckaert-Bellemans 二面角势

(注意: 使用这种势函数意味着排除了组成二面角的第一个和最后一个原子之间的LJ相互作用, 并且 ψ 的定义遵从“聚合物约定”($\psi_{trans} = 0$).

RB二面角函数也可用于包含傅立叶二面角(见下文):

$$V_{rb}(\phi_{ijkl}) = \frac{1}{2} [F_1(1 + \cos(\phi)) + F_2(1 - \cos(2\phi)) + F_3(1 + \cos(3\phi)) + F_4(1 - \cos(4\phi))] \quad (4.65)$$

由于存在等式 $\cos(2\phi) = 2\cos^2(\phi) - 1$, $\cos(3\phi) = 4\cos^3(\phi) - 3\cos(\phi)$ 和 $\cos(4\phi) = 8\cos^4(\phi) - 8\cos^2(\phi) + 1$, 可以利用下面的方程将OPLS参数变换为Ryckaert-Bellemans参数

$$\begin{aligned} C_0 &= F_2 + \frac{1}{2}(F_1 + F_3) \\ C_1 &= \frac{1}{2}(-F_1 + 3F_3) \\ C_2 &= -F_2 + 4F_4 \\ C_3 &= -2F_3 \\ C_4 &= -4F_4 \\ C_5 &= 0 \end{aligned} \quad (4.66)$$

OPLS参数遵从蛋白质约定, 而RB参数遵从聚合物约定(这导致 $\cos(\phi)$ 奇数次项前面的负号).

注意: 记得将文献中OPLS和RB参数的单位 kcal mol^{-1} 转化为GROMACS中的单位 kcal mol^{-1} .

正常二面角: 傅立叶函数

OPLS势能函数为傅立叶级数的前三或四次余弦项[87]. GROMACS中, 四项函数为:

$$V_F(\phi_{ijkl}) = \frac{1}{2} [C_1(1 + \cos(\phi)) + C_2(1 - \cos(2\phi)) + C_3(1 + \cos(3\phi)) + C_4(1 + \cos(4\phi))] \quad (4.67)$$

GROMACS在内部使用Ryckaert-Bellemans 代码来计算傅立叶二面角 (见上文), 因为计算效率更高.

注意: 记得将文献中OPLS参数的单位 kcal mol^{-1} 转化为GROMACS中的单位 kcal mol^{-1} .

正常二面角: 限制扭转势

非常类似于限制弯曲势 (见4.2.7), 限制扭转/二面角势的函数为:

$$V_{\text{ReT}}(\phi_i) = \frac{1}{2} k_\phi \frac{(\cos \phi_i - \cos \phi_0)^2}{\sin^2 \phi_i} \quad (4.68)$$

这类函数具有 $\cos \phi$ 函数的优点 (对 $\sin \phi$ 求导时不存在问题), 并且能保持扭转角处于仅有的一个最小值. 在这种情况下, 因子 $\sin^2 \phi$ 不会让二面角从 $[-180^\circ : 0]$ 区间移到 $[0 : 180^\circ]$ 区间内, 即, 函数不能同时在 $-\phi_0$ 和 $+\phi_0$ 处取极大值, 而只能在其中之一取极大值. 由于这个原因, 初始构型中所有的二面角的角度值都必须处于所需的角度区间内, 并且平衡的 ϕ_0 值不应太靠近区间的端点 (与4.2.7中所讲的限制弯曲势类似, 推荐至少 10° 的差异).

正常二面角: 组合弯曲-扭转势

当形成二面角的四个粒子共线时 (这种情况永远不会发生在原子尺度的模拟中, 但可能出现在粗粒模拟中), 扭转角及其势函数的计算会引起数值不稳定性. 避免这一问题的一种方法是使用限制弯曲势 (见4.2.7), 以防止二面角达到 180° .

另一种方式是忽略二面角不合理定义时的任何效应, 通过将扭转势 (余弦形式) 与相邻弯曲角的弯曲势耦合为一个特殊形式, 保持二面角的力和势函数的计算在整个角度范围内连续:

$$V_{\text{CBT}}(\theta_{i-1}, \theta_i, \phi_i) = k_\phi \sin^3 \theta_{i-1} \sin^3 \theta_i \sum_{n=0}^4 a_n \cos^n \phi_i \quad (4.69)$$

这种组合的弯曲扭转 (combined bending-torsion, CBT) 势函数由文献[88]提出, 用于聚合物熔融的模拟. 文献[84]对此有详细的描述.

这种势函数有两个主要的优点:

- 它不仅取决于(第 $i-2, i-1, i$ 和 $i+1$ 号珠子之间的)二面角 ϕ_i , 而且也取决于由三个相邻的珠子(分别是 $i-2, i-1, i$ 和 $i-1, i, i+1$)形成的弯曲角 θ_{i-1} 和 θ_i . 当两个弯曲角中的任意一个接近 180° 时, 由文献[89]尝试性地提出, 并由文献[90]从理论上进行了讨论的两个 $\sin^3 \theta$ 前因子抵消了扭转势和力.
- 通过表达为 $\cos \phi_i$ 的多项式形式, 它也取决于 ϕ_i , 这避免了 $\phi = 0^\circ$ 或 $\phi = 180^\circ$ 时计算扭转力的奇异性.

这两种特性使得CBT势用于弯曲角存在弱约束的MD模拟时表现良好, 甚至对于弯曲角和扭转角变化剧烈的拉伸/非平衡MD模拟也是如此. 当使用CBT势时, 相邻的 θ_{i-1} 和 θ_i 的弯曲势可以取任意形式, 也可完全省略这两个角弯曲项(θ_{i-1} 和 θ_i). 图4.13显示了包含和不含 $\sin^3 \theta$ 因子 (分别为蓝色和灰色的曲线) 的扭转势之间的区别. 此外, V_{CBT} 对笛卡尔变量的导数直接易懂:

$$\frac{\partial V_{\text{CBT}}(\theta_{i-1}, \theta_i \phi_i)}{\partial \vec{r}_l} = \frac{\partial V_{\text{CBT}}}{\partial \theta_{i-1}} \frac{\partial \theta_{i-1}}{\partial \vec{r}_l} + \frac{\partial V_{\text{CBT}}}{\partial \theta_i} \frac{\partial \theta_i}{\partial \vec{r}_l} + \frac{\partial V_{\text{CBT}}}{\partial \phi_i} \frac{\partial \phi_i}{\partial \vec{r}_l} \quad (4.70)$$

CBT基于无多重性的余弦形式, 因此它只在 0° 左右对称. 为获得一个不对称的二面角分布 (如在 $[-180^\circ : 180^\circ]$ 区间内只有一个最大值), 应该使用标准的扭转势, 如简谐角势或周期性的余弦势来代替CBT势. 然而, 这两种形式对于力的微分 ($1/\sin \phi$) 和珠子的直线排列 (θ_i 或 $\theta_{i-1} = 0^\circ, 180^\circ$) 都不方便. 非 $\cos \phi$ 势与 $\sin^3 \theta$ 因子的耦合并不能提高模拟的稳定性, 因为在某些情况下 θ 和 ϕ 同时为 180° . 在此步的积分是可能的 (由于扭转势的抵消), 但下一步积分将是病态的 (θ 不是 180° 而 ϕ 非常接近 180°).

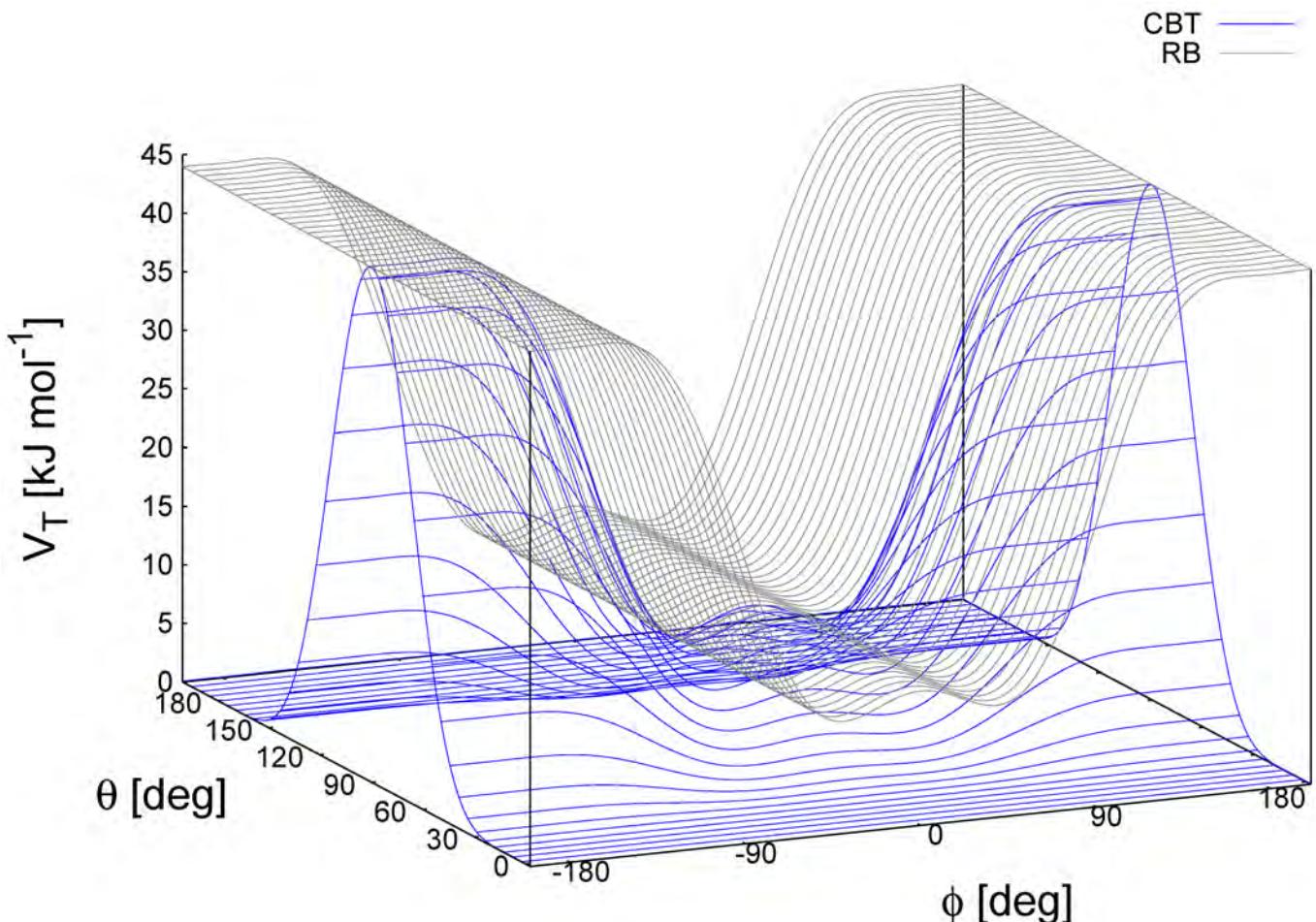


图4.13: 蓝色: 组合弯曲扭转势的表面图(方程4.69, 其中 $k = 10 \text{ kJ mol}^{-1}$, $a_0 = 2.41$, $a_1 = -2.95$, $a_2 = 0.36$, $a_3 = 1.33$), 为简单起见, 弯曲角相同($\theta_1 = \theta_2 = \theta$). 灰色: 同样的扭转势函数但不含 $\sin^3 \theta$ 项 (Ryckaert-Bellemans 型). ϕ 为二面角.

4.2.14 表格式键合相互作用

为了有充分的灵活性, 可通过用户提供的表格函数形式, 对键, 键角和二面角使用任何函数. 它们包括:

$$V_b(r_{ij}) = kf_n^b(r_{ij}) \quad (4.71)$$

$$V_a(\theta_{ijk}) = kf_n^a(\theta_{ijk}) \quad (4.72)$$

$$V_d(\phi_{ijkl}) = kf_n^d(\phi_{ijkl}) \quad (4.73)$$

其中, k 是以能量为单位的力常数, f 是三次样条函数; 详细信息见6.10.1. 对每种相互作用, 在拓扑信息中指定力常数 k 和表格编号 n . 有两种不同类型的键, 一种产生排除 (类型8), 一种不产生排除 (类型9). 详情见表5.5. 表格文件会被提供给

mdrun程序. 表格文件名和下划线后, 加入字母“b”表示键, “a”表示角或“d”表示二面角和表格编号. 例如, 对一个 $n = 0$ 的键 (并使用默认的表格文件名), 将从文件 **table_b0.xvg** 中读取表格. 如拓扑信息 (表5.5) 中具体介绍的那样, 可简单地通过使用不同的 n 值提供多个表格, 并应用到恰当的键. 表格文件的格式是分别为 x , $f(x)$, $-f'(x)$ 的三列, 其中 x 必须是等间隔的. 拓扑信息的输入要求列于表5.5. 这些表格的设置如下:

键: x 是以纳米(nm)为单位的距离. 若模拟时距离超出表格长度, `mdrun` 将显示错误消息并退出.

角度: x 是以度为单位的角度. 表格应从0开始直到并包括180度 ; 导数以度为单位.

二面角: x 是以度为单位的二面角. 表格应从-180直到并包括180度, 使用IUPAC/IUB约定, 即零为顺式, 导数以度为单位.

4.3 限制

可以使用一些特殊的势能函数对体系的运动施加限制, 这样可以避免模拟中出现灾难性的偏差, 或是可以包含来自实验数据的知识. 无论哪种情况, 限制都不是力场的一部分, 并且所用参数的合理性也不重要. 提到GROMACS中实现的势能形式只是为了保持完整性. 在GROMACS中限制和约束所用的算法差异很大.

4.3.1 位置限制

位置限制用于将粒子限制在固定的参考位置 \mathbf{R}_i . 在平衡过程中可利用它们来避免体系关键部分的位置产生剧烈的变化(例如, 当溶剂还未达到平衡时, 蛋白质会受到来自溶剂分子的很大的力, 这时就可以限制蛋白质中的一些运行). 位置限制的另一个应用是将粒子限制在围绕某个区域的壳层中, 以便对此区域进行细致的模拟. 壳层只是近似的, 因为它们缺少由壳层外面缺失粒子引起的适当的相互作用. 位置限制只是为了维持体系内部的完整性. 对球形壳层, 更好的做法是让力常数随半径变化, 由内部边界从零开始增加, 一直到外部边界上的很大值. 但这一功能尚未在GROMACS中实现.

位置限制使用下面的势能形式

$$V_{pr}(\mathbf{r}_i) = \frac{1}{2}k_{pr}|\mathbf{r}_i - \mathbf{R}_i|^2 \quad (4.74)$$

此势能的图形见图4.14.

不失一般性, 可将势能改写为:

$$V_{pr}(\mathbf{r}_i) = \frac{1}{2} \left[k_{pr}^x (x_i - X_i)^2 \hat{x} + k_{pr}^y (y_i - Y_i)^2 \hat{y} + k_{pr}^z (z_i - Z_i)^2 \hat{z} \right] \quad (4.75)$$

相应的力为

$$\begin{aligned} F_i^x &= -k_{pr}^x (x_i - X_i) \\ F_i^y &= -k_{pr}^y (y_i - Y_i) \\ F_i^z &= -k_{pr}^z (z_i - Z_i) \end{aligned} \quad (4.76)$$

使用三个不同的力常数就可以在每个空间方向上打开或关闭位置限制, 这意味着原子可以简谐地限制于平面或直线上. 位置限制施加于指定的固定原子列表上, 这一列表通常使用 `pdb2gmx` 程序产生.

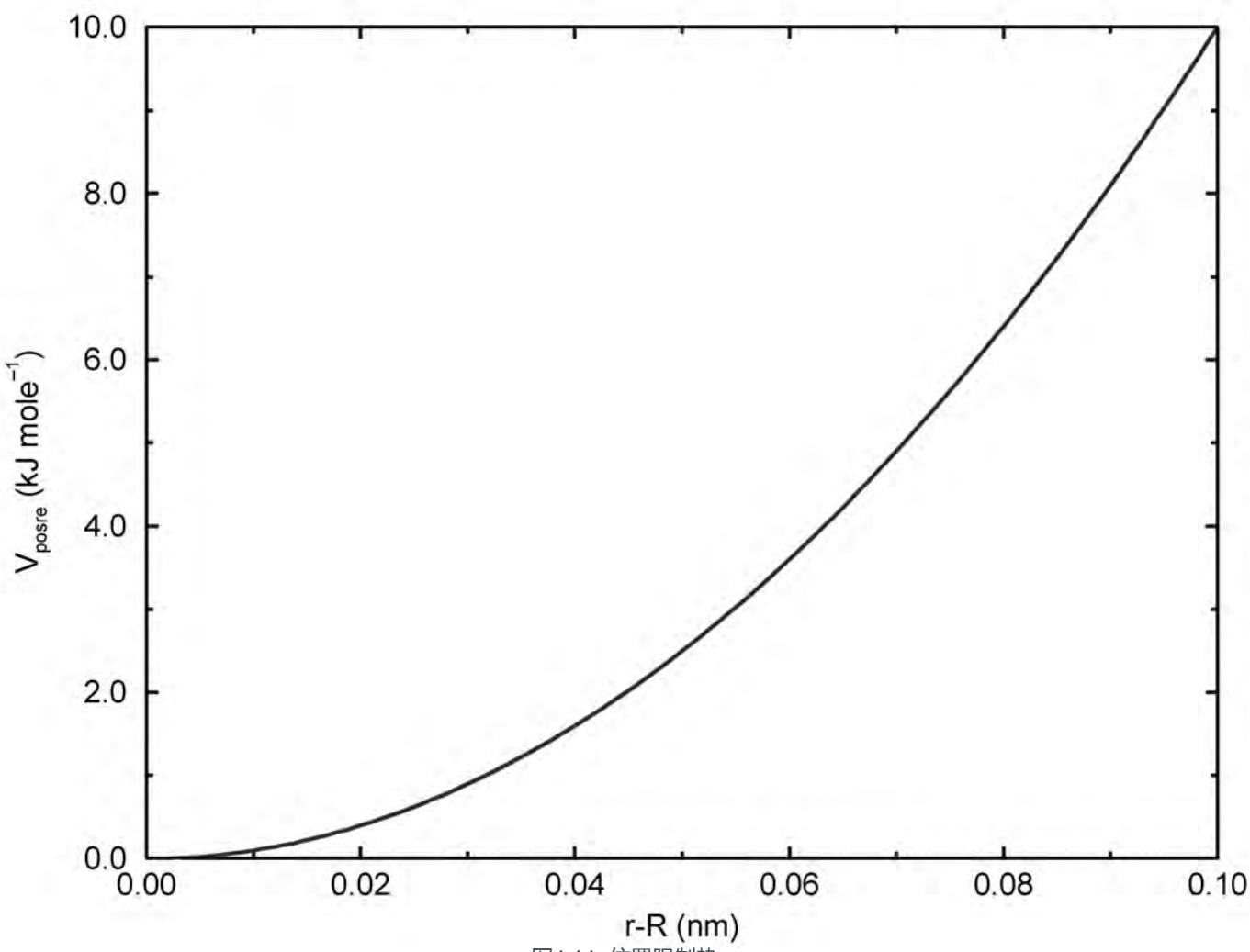


图4.14: 位置限制势

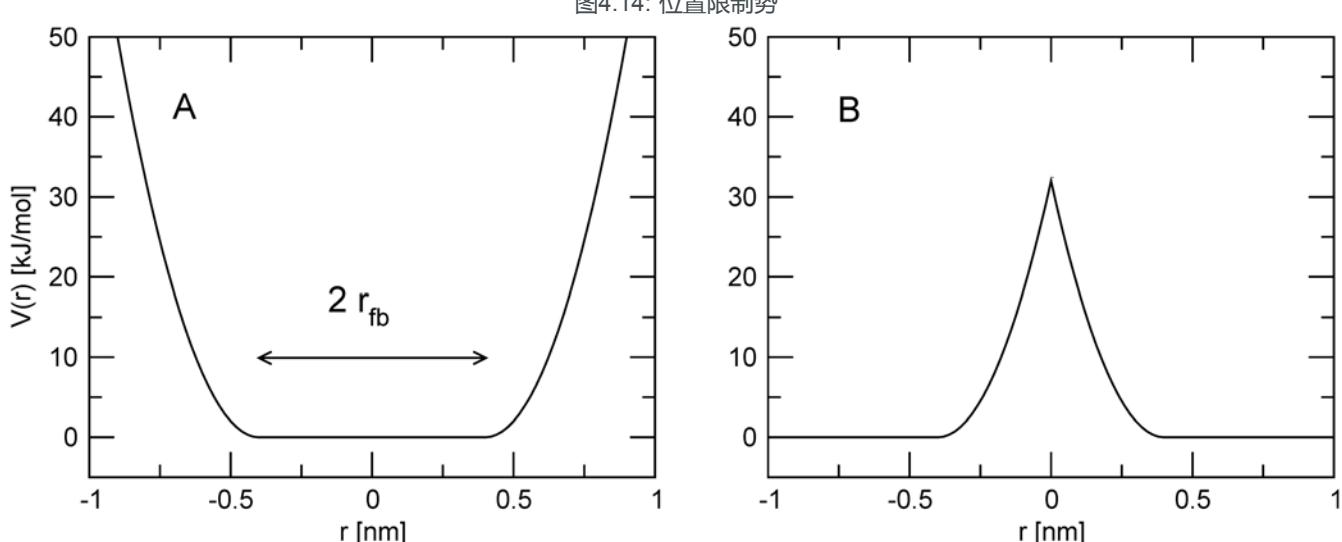


图4.15: 平底位置限制势. (A) 未反转, (B) 反转

4.3.2 平底位置限制势

平底位置限制可用于将粒子限制在模拟体积的某一部分中。在势能的平底区域中限制粒子不受力的作用，但当粒子处于平底区域之外时会受到一个简谐力，这个力会驱使粒子向平底区域运动。对同一粒子可以同时施加常规的位置限制和平底的位置限制（但只用于相同的参考位置 \mathbf{R}_i ）。平底位置限制使用下面的通用势能函数（图4.15 A）：

$$V_{\text{fb}}(\mathbf{r}_i) = \frac{1}{2} k_{\text{fb}} [d_g(\mathbf{r}_i; \mathbf{R}_i) - r_{\text{fb}}]^2 H[d_g(\mathbf{r}_i; \mathbf{R}_i) - r_{\text{fb}}] \quad (4.77)$$

其中 \mathbf{R}_i 为参考位置， r_{fb} 为平势离中心的距离， k_{fb} 为力常数， H 为Heaviside 阶梯函数。到参考位置的距离 $d_g(\mathbf{r}_i; \mathbf{R}_i)$ 取

决于平底势的几何构造 g .

对平底势, GROMACS支持下面几种几何构造:

球 ($g = 1$): 粒子被限制于给定半径的球中. 所受平底势的力指向球的中心, 距离的计算方法如下:

$$d_g(\mathbf{r}_i; \mathbf{R}_i) = |\mathbf{r}_i - \mathbf{R}_i| \quad (4.78)$$

圆柱 ($g = 2$): 粒子被限制于给定半径的圆柱中. 圆柱的轴平行于 z 轴, 所受平底势的力指向圆柱的轴, 力的 z 分量为零. 距离的计算方法如下:

$$d_g(\mathbf{r}_i; \mathbf{R}_i) = \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2} \quad (4.79)$$

层 ($g = 3, 4, 5$): 粒子限制于层中. 层由其厚度与法线定义, 法线可平行于 x, y 或 z 轴, 作用力平行于层的法线. 距离的计算方法如下:

$$d_g(\mathbf{r}_i; \mathbf{R}_i) = |x_i - X_i| \text{ 或 } d_g(\mathbf{r}_i; \mathbf{R}_i) = |y_i - Y_i| \quad d_g(\mathbf{r}_i; \mathbf{R}_i) = |z_i - Z_i| \quad (4.80)$$

对同一粒子可施加多个独立的, 具有不同几何构造的平底位置限制. 例如, 沿 z 方向同时施加圆柱和层限制可使粒子限制于一个圆盘中; 在 x, y, z 方向施加三个层限制可使粒子限制于立方体中.

此外, 可以反转限制区域和未限制区域, 这样得到的势能会使粒子处于 \mathbf{R}_i, g 和 r_{fb} 定义的体积之外. 在拓扑中定义负值的 r_{fb} 可启用此功能, 使用下面的势能:

$$V_{fb}^{\text{inv}} = \frac{1}{2} k_{fb} [d_g(\mathbf{r}_i; \mathbf{R}_i) - |r_{fb}|]^2 H[-(d_g(\mathbf{r}_i; \mathbf{R}_i) - |r_{fb}|)] \quad (4.81)$$

4.3.3 角限制

角限制用于限制两对粒子间或一对粒子与 z 轴之间的角度. 所用的函数形式类似于恰当二面角, 对两对原子之间的角:

$$V_{ar}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k, \mathbf{r}_l) = k_{ar} (1 - \cos(n(\theta - \theta_0))), \quad \theta = \arccos \left(\frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|} \cdot \frac{\mathbf{r}_l - \mathbf{r}_k}{\|\mathbf{r}_l - \mathbf{r}_k\|} \right) \quad (4.82)$$

对一对原子与 z 轴之间的角:

$$V_{ar}(\mathbf{r}_i, \mathbf{r}_j) = k_{ar} (1 - \cos(n(\theta - \theta_0))), \quad \theta = \arccos \left(\frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right) \quad (4.83)$$

当不需要区分平行与反平行向量时, 多重性 n 可设为2. 当多重性为1时, 平衡角度 θ 应处于0度和180度之间; 当多重性为2时, θ 应处于0度和90度之间.

4.3.4 二面角限制

二面角限制用于限制由四个粒子定义的二面角, 如不当二面角(4.2.12节), 但使用了稍加修改的势能:

$$\phi' = (\phi - \phi_0) \text{ MOD } 2\pi \quad (4.84)$$

其中 ϕ_0 为参考角度, 势能的定义为:

$$V_{dihr}(\phi') = \begin{cases} \frac{1}{2} k_{dihr} (\phi' - \phi_0 - \Delta\phi)^2 & \text{当 } \phi' > \Delta\phi \\ 0 & \text{当 } \phi' \leq \Delta\phi \end{cases} \quad (4.85)$$

其中 $\Delta\phi$ 为用户定义的角度, k_{dihr} 为力常数. 注意, 在拓扑文件的输入中, 角度的单位为度, 力常数的单位为kJ/mol/rad².

4.3.5 距离限制

当指定的一对原子之间的距离超过一个阈值时, 距离限制会给势能增加一个惩罚项. 它们通常用于施加来自实验的限制, 例如核磁共振(NMR, nuclear magnetic resonance) 实验中对体系运动的限制. 因此, MD可用于NMR数据的结构精修. 在GROMACS中, 有三种方式对原子对施加距离限制:

- 简单的简谐限制: 使用 [bonds] 类型6. (见5.4节)
- 分段线性/简谐限制: [bonds] 类型10.
- 复杂的NMR距离限制, 可选的原子对, 时间和/或系综平均

下面将叙述后面两个选项.

当低于指定的下界或处于指定的两个上界之间时, 距离限制的势能形式为二次; 当超过最大边界时, 势能为线性(见图4.16).

$$V_{dr}(r_{ij}) = \begin{cases} \frac{1}{2} k_{dr} (r_{ij} - r_0)^2 & \text{当 } r_{ij} < r_0 \\ 0 & \text{当 } r_0 \leq r_{ij} < r_1 \\ \frac{1}{2} k_{dr} (r_{ij} - r_1)^2 & \text{当 } r_1 \leq r_{ij} < r_0 \\ \frac{1}{2} k_{dr} (r_2 - r_1)(2r_{ij} - r_2 - r_1) & \text{当 } r_2 \leq r_{ij} \end{cases} \quad (4.86)$$

相应的力为

$$F_i = \begin{cases} -k_{dr}(r_{ij} - r_0) \frac{r_{ij}}{r_{ij}} & \text{当 } r_{ij} < r_0 \\ 0 & \text{当 } r_0 \leq r_{ij} < r_1 \\ -k_{dr}(r_{ij} - r_1) \frac{r_{ij}}{r_{ij}} & \text{当 } r_1 \leq r_{ij} < r_0 \\ -k_{dr}(r_2 - r_1) \frac{r_{ij}}{r_{ij}} & \text{当 } r_2 \leq r_{ij} \end{cases} \quad (4.87)$$

对不是由NMR数据导出的限制, 这个函数通常足够了, [bonds] 类型10的段可用于在原子对之间施加独立的限制, 见5.7.1节. 对由NMR测量导出的限制, 可能需要更复杂的函数, 这可通过 [distance_restraints] 段提供, 具体将在下面论述.

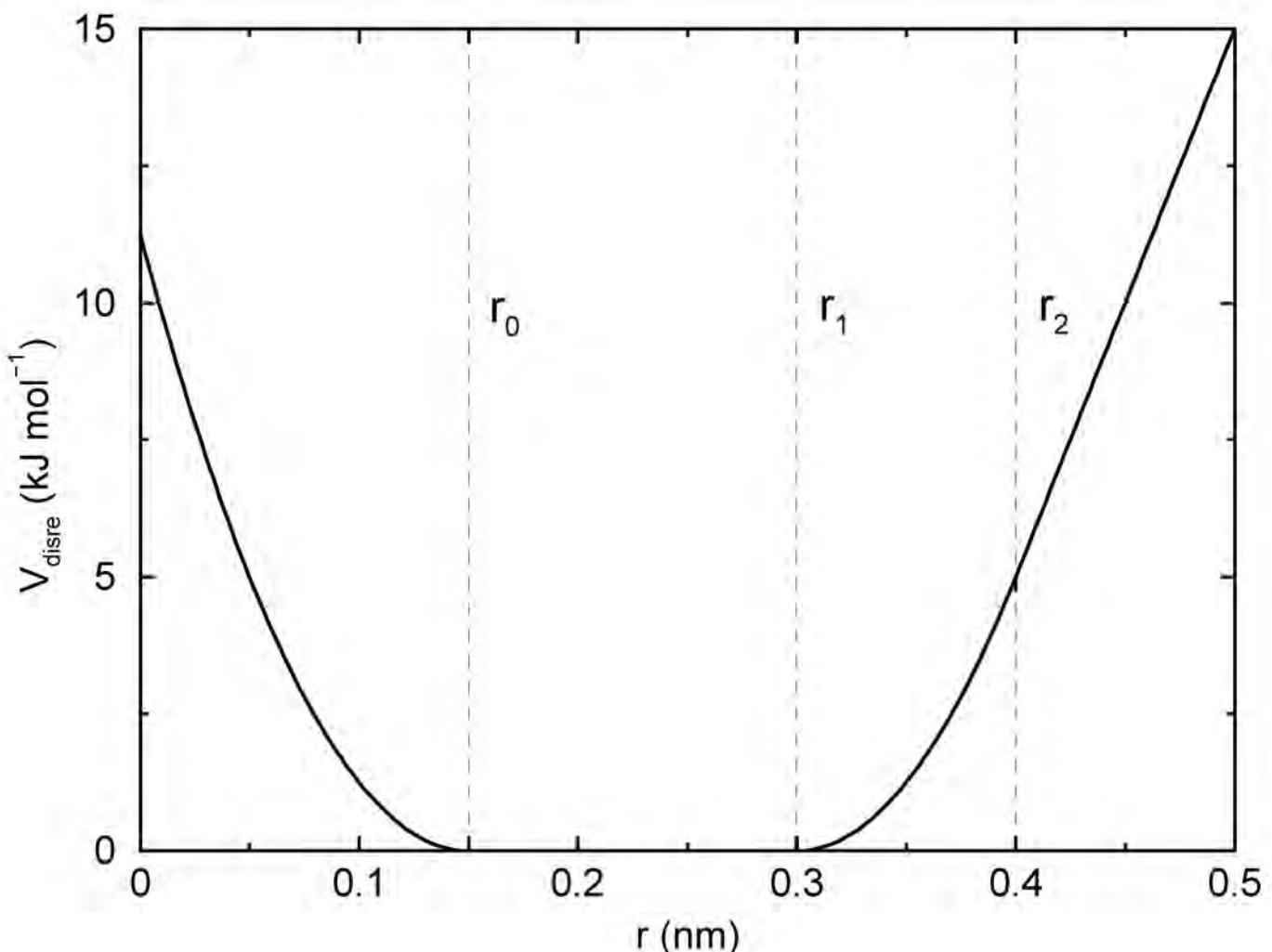


图4.16: 距离限制势

时间平均

基于瞬时距离的距离限制可能会显著地减弱分子内的涨落. 采用 **时间平均** 的限制距离可以解决这个问题[91]. 时间平均的力为

$$F_i = \begin{cases} -k_{dr}^a (\bar{r}_{ij} - r_0) \frac{r_{ij}}{r_{ij}} & \text{当 } \bar{r}_{ij} < r_0 \\ 0 & \text{当 } r_0 \leq \bar{r}_{ij} < r_1 \\ -k_{dr}^a (\bar{r}_{ij} - r_1) \frac{r_{ij}}{r_{ij}} & \text{当 } r_1 \leq \bar{r}_{ij} < r_0 \\ -k_{dr}^a (r_2 - r_1) \frac{r_{ij}}{r_{ij}} & \text{当 } r_2 \leq \bar{r}_{ij} \end{cases} \quad (4.88)$$

其中 \bar{r}_{ij} 由衰减时间为 τ 的指数运行平均给出:

$$\bar{r}_{ij} = \langle r_{ij}^{-3} \rangle^{-1/3} \quad (4.89)$$

缓慢打开力常数 k_{dr}^a , 以弥补模拟开始时缺少历史数据的问题:

$$k_{dr}^a = k_{dr} \left(1 - \exp \left(-\frac{t}{\tau} \right) \right) \quad (4.90)$$

由于时间平均, 我们无法再定义距离限制势了.

采用这种方式, 一个原子可以通过在两个位置之间移动 平均地 满足两个不兼容的距离限制. 例如围绕 χ 二面角旋转的氨基酸支链会因此接近其他各种基团. 这样的活动支链可导致多个NOE, 不能由单一结构满足.

mdrun 程序计算时间平均距离的方法如下:

$$\begin{aligned}\overline{r^{-3}}_{ij}(0) &= r_{ij}(0)^{-3} \\ \overline{r^{-3}}_{ij}(t) &= \overline{r^{-3}}_{ij}(t - \Delta t) \exp\left(-\frac{\Delta t}{\tau}\right) + r_{ij}(t)^{-3} \left[1 - \exp\left(-\frac{\Delta t}{\tau}\right)\right]\end{aligned}\quad (4.91)$$

当原子对处于边界之内时可能仍会受到力的作用, 因为时间平均的距离可能仍然超过了边界. 为防止质子彼此靠得太近, 可以使用一种混合的方法. 在这种方法中, 当瞬时距离处于边界之内时, 惩罚项为零, 在其他情况下惩罚项为瞬时偏离与时间平均偏离乘积的平方根:

$$\mathbf{F}_i = \begin{cases} k_{dr}^a \sqrt{(r_{ij} - r_0)(\bar{r}_{ij} - r_0)} \frac{r_{ij}}{r_{ij}} & \text{当 } r_{ij} < r_0 \text{ 且 } \bar{r}_{ij} < r_0 \\ -k_{dr}^a \min\left(\sqrt{(r_{ij} - r_1)(\bar{r}_{ij} - r_1)}, r_2 - r_1\right) \frac{r_{ij}}{r_{ij}} & \text{当 } r_{ij} > r_1 \text{ 且 } \bar{r}_{ij} > r_1 \\ 0 & \text{其他情况}\end{cases}\quad (4.92)$$

多对平均

根据实验数据, 有时不清楚是哪些原子对导致了单个NOE, 在其他一些情况下, 由于体系的对称性, 很显然地有多余一对的原子有贡献, 例如含有三个质子的甲基. 对这样一个基团, 不可能区分不同的质子, 因此当计算此甲基与其他质子(或一组质子)之间的距离时, 所有三个质子都要考虑. 由于磁共振的物理本性, NOE信号的强度与原子间距离的6次方成反比, 因此, 当联合原子对时, N 个限制的固定列表要放在一起, 表现出的“距离”为:

$$r_N(t) = \left[\sum_{n=1}^N \bar{r}_n(t)^{-6} \right]^{-1/6}\quad (4.93)$$

其中对 \bar{r}_n 我们可以使用 r_{ij} 或方程4.89. 瞬时距离和时间平均的距离 r_N 可以联合起来施加混合约束, 如上所述. 由于更多的质子对对相同的NOE信号有贡献, 其强度会增大, 由于是倒数加和, 累加的“距离”会小于距离的任何一个分量.

将力分布到原子对上时, 有两个选项. 对第一个守恒选项, 力定义为限制势对坐标的导数, 当不使用时间平均时, 得到的势能是守恒的. 分布到原子对上的力与 r^{-6} 成正比, 这意味着互相靠近的原子对比互相远离的原子对受到更大的力, 这可能导致分子“过于刚性”. 另一个选项是等力分配. 这种情况下, 每对原子受到的力为限制势对 r_N 导数的 $1/N$. 这种方法的优势在于可以对更多的构型进行抽样, 但不守恒的力可能导致对质子的局部加热.

也可以使用多个(蛋白质)分子的 系综平均. 在这种情况下, 界限应当降低:

$$\begin{aligned}r_1 &= r_1 * M^{-1/6} \\ r_2 &= r_2 * M^{-1/6}\end{aligned}\quad (4.94)$$

其中 M 为分子数. 当给出适当的选项时, GROMACS预处理器 **grompp** 可自动进行这些处理. 得到的“距离”用于计算标量力:

$$\mathbf{F}_i = \begin{cases} 0 & r_N < r_1 \\ k_{dr}(r_N - r_1) \frac{r_{ij}}{r_{ij}} & r_1 \leq r_N < r_2 \\ k_{dr}(r_2 - r_1) \frac{r_{ij}}{r_{ij}} & r_N \geq r_2 \end{cases}\quad (4.95)$$

其中 i, j 代表所有对NOE信号有贡献的原子对.

使用距离限制

在拓扑文件中可以将基于NOE数据的距离限制添加到分子定义中, 如下面的示例:

[distance_restraints]								
ai	aj	type	index	type'	low	up1	up2	fac
10	16	1	0	1	0.0	0.3	0.4	1.0
10	28	1	1	1	0.0	0.3	0.4	1.0
10	46	1	1	1	0.0	0.3	0.4	1.0
16	22	1	2	1	0.0	0.3	0.4	2.5
16	34	1	3	1	0.0	0.5	0.6	1.0

在这个例子中, 可以发现很多特性. **ai** 列与 **aj** 列为限制粒子的原子编号. **type** 列总是为 1. 如在 4.3.5 节解释的那样, 多个距离可以对单个 NOE 信号有贡献. 在拓扑中, 这可使用 **index** 列指定. 在我们的例子中, 限制 10–28 和 10–64 的索引都是 1, 因此它们会被同时处理. 一起处理的一个额外要求是几个限制必须位于连续的行中, 中间不能有其他限制. **type'** 列通常为 1, 但也可设为 2, 以便不对距离限制进行时间平均与系综平均, 这适用于限制氢键. 列 **low**, **up1** 和 **up2** 分别为方程 4.86 中的 r_0 , r_1 和 r_2 值. 在一些情况下, 对一些限制使用不同的力常数可能有用, 这可通过 **fac** 列进行控制. 对每个限制, 参数文件中的力常数会乘上 **fac** 列中的值.

4.3.6 取向限制

本节论述在 MD 模拟中如何计算和限制向量间的取向, 如一些 NMR 实验中测量的. 这里论述的精修方法, 以及使用与不使用时间平均与系综平均对结果的影响已经发表 [92].

理论

在 NMR 实验中, 当一个分子在溶剂中的翻滚并非完全各向同性时可测量矢量间的取向. 这种取向测量的两个例子是残基偶极耦合(两个原子核之间)或化学位移各向异性. 向量 \mathbf{r} 的可观测量可写为:

$$\delta_i = \frac{2}{3} \text{tr}(\mathbf{SD}_i) \quad (4.96)$$

其中 \mathbf{S} 为无量纲的分子序张量, 张量 \mathbf{D}_i 为

$$\mathbf{D}_i = \frac{c_i}{\|\mathbf{r}_i\|^\alpha} \begin{pmatrix} 3xx - 1 & 3xy & 3xz \\ 3xy & 3yy - 1 & 3yz \\ 3xz & 3yz & 3zz - 1 \end{pmatrix} \quad (4.97)$$

其中

$$x = \frac{r_{i,x}}{\|\mathbf{r}_i\|}, y = \frac{r_{i,y}}{\|\mathbf{r}_i\|}, z = \frac{r_{i,z}}{\|\mathbf{r}_i\|} \quad (4.98)$$

对一个偶极耦合 \mathbf{r}_i 为连接两个原子核间的向量, $\alpha = 3$, 常数 c_i 为:

$$c_i = \frac{\mu_0}{4\pi} \gamma_1^i \gamma_2^i \frac{\hbar}{4\pi} \quad (4.99)$$

其中为 γ_1^i 和 γ_2^i 为两个核的旋磁比.

序张量是对称的, 迹为零. 使用旋转矩阵 \mathbf{T} 可将其变换为下面的形式:

$$\mathbf{T}^T \mathbf{S} \mathbf{T} = s \begin{pmatrix} -\frac{1}{2}(1-\eta) & 0 & 0 \\ 0 & -\frac{1}{2}(1+\eta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.100)$$

其中 $-1 \leq s \leq 1$, $0 \leq \eta \leq 1$. s 称为序参数, η 称为序张量 \mathbf{S} 的非对称性. 当分子在溶剂中的翻滚各向同性时, s 为零, 无法观测到取向效应, 因为所有的 δ_i 都是零.

模拟中取向的计算

由于下面将要解释的原因, \mathbf{D} 矩阵是相对于分子的参考取向计算的. 取向由旋转矩阵 \mathbf{R} 定义, 计算它时需要使用最小二乘方法将选定的一组原子的当前坐标拟合到参考构型, 参考构型为模拟的初始构型. 对后面将要处理的系综平均, 参考结构来自第一个子体系. \mathbf{D}_i^c 矩阵的计算方法为:

$$\mathbf{D}_i^c(t) = \mathbf{R}(t) \mathbf{D}_i(t) \mathbf{R}^T(t) \quad (4.101)$$

向量 i 的取向为:

$$\delta_i^c(t) = \frac{2}{3} \text{tr}(\mathbf{S}(t) \mathbf{D}_i^c(t)) \quad (4.102)$$

序张量 $\mathbf{S}(t)$ 通常是未知的, 对它的一个合理选择是能够使计算与观测到的取向的(加权)平均方差最小的张量:

$$MSD(t) = \left(\sum_{i=1}^N w_i \right)^{-1} \sum_{i=1}^N w_i (\delta_i^c(t) - \delta_i^{exp})^2 \quad (4.103)$$

为正确地联合不同类型的测量, w_i 的单位应使得所有项都是无量纲的. 这意味着 w_i 单位的次数为 δ_i 单位的次数-2. 注意 使用常数因子缩放所有 w_i 并不影响序张量.

时间平均

由于张量 \mathbf{D} 随时间迅速涨落, 比能在实验中观测到的快得多, 在模拟中它们应当对时间进行平均. 然而, 在模拟中时间与分子数都是有限的, 通常不能得到对分子所有取向收敛的 \mathbf{D}_i 张量的平均值. 如果假定分子内 \mathbf{r}_i 向量平均取向的收敛比分子的翻滚时间快得多, 这一張量可以在随分子旋转的坐标系中进行平均, 如方程4.101所示. 时间平均的张量使用指数衰减记忆函数进行计算:

$$\mathbf{D}_i^a(t) = \frac{\int_{u=t_0}^t \mathbf{D}_i^c(u) \exp\left(-\frac{t-u}{\tau}\right) du}{\int_{u=t_0}^t \exp\left(-\frac{t-u}{\tau}\right) du} \quad (4.104)$$

假定序张量 \mathbf{S} 的涨落慢于 \mathbf{D}_i , 时间平均的取向为:

$$\delta_i^a(t) = \frac{2}{3} \text{tr}(\mathbf{S}(t) \mathbf{D}_i^a(t)) \quad (4.105)$$

其中序张量 \mathbf{S} 使用表达式4.103进行计算, 但将其中的 $\delta_i^c(t)$ 替换为 $\delta_i^a(t)$.

限制

可以对模拟结构施加力进行限制, 施加的力正比于计算与实验取向之间差异. 无时间平均时, 恰当的势能可定义为:

$$V = \frac{1}{2} k \sum_{i=1}^N w_i (\delta_i^c(t) - \delta_i^{exp})^2 \quad (4.106)$$

其中 k 的单位为能量单位, 这样限制 i 的有效力常数为 kw_i . 力是势能 V 的梯度的负值. 作用于向量 \mathbf{r}_i 上的力 \mathbf{F}_i 为:

$$\begin{aligned} \mathbf{F}_i(t) &= -\frac{dV}{d\mathbf{r}_i} \\ &= -kw_i(\delta_i^c(t) - \delta_i^{exp}) \frac{d\delta_i(t)}{d\mathbf{r}_i} \\ &= -kw_i(\delta_i^c(t) - \delta_i^{exp}) \frac{2c_i}{\|\mathbf{r}\|^{2+\alpha}} \left(2\mathbf{R}^T \mathbf{S} \mathbf{R} \mathbf{r}_i - \frac{2+\alpha}{\|\mathbf{r}\|^2} \text{tr}(R^T \mathbf{S} \mathbf{R} \mathbf{r}_i \mathbf{r}_i^T) \mathbf{r}_i \right) \end{aligned}$$

系综平均

系综平均可通过模拟一个具有 M 个子体系的体系来实现, 其中每个子体系包含了完全相同的取向限制. 体系只通过取向限制势进行相互作用:

$$V = M \frac{1}{2} k \sum_{i=1}^N w_i < \delta_i^c(t) - \delta_i^{exp} >^2 \quad (4.107)$$

子体系 m 中向量 $\mathbf{r}_{i,m}$ 上的力为:

$$\mathbf{F}_{i,m}(t) = -\frac{dV}{d\mathbf{r}_{i,m}} = -kw_i < \delta_i^c(t) - \delta_i^{exp} > \frac{d\delta_{i,m}^c(t)}{d\mathbf{r}_{i,m}} \quad (4.108)$$

时间平均

当使用时间平均时, 无法定义势能. 但我们仍可以定义一个量, 它粗略地表示了限制的能量:

$$V = M \frac{1}{2} k^a \sum_{i=1}^N w_i < \delta_i^a(t) - \delta_i^{exp} >^2 \quad (4.109)$$

缓慢打开力常数 k^a , 以弥补时间接近 t_0 时缺少历史数据的问题. 它精确地正比于已积累的平均量:

$$k^a = k \frac{1}{\tau} \int_{u=t_0}^t \exp \left(-\frac{t-u}{\tau} \right) du \quad (4.110)$$

真正关键的是力的定义, 它正比于时间平均与瞬时偏差乘积的平方根. 只使用时间平均的偏差会导致大的振荡. 力的计算方式如下:

$$\begin{aligned} \mathbf{F}_{i,m}(t) &= \begin{cases} 0 & \text{当 } ab \leq 0 \\ k^a w_i \frac{a}{|a|} \sqrt{ab} \frac{d\delta_{i,m}^c(t)}{d\mathbf{r}_{i,m}} & \text{当 } ab > 0 \end{cases} \\ a &= < \delta_i^a(t) - \delta_i^{exp} > \\ b &= < \delta_i^c(t) - \delta_i^{exp} > \end{aligned} \quad (4.111)$$

使用取向限制

在拓扑文件的 [orientation_restraints] 段中可以将取向限制添加到分子的定义中. 这里我们给出一个示例段, 包含了五个N-H残基的偶极耦合限制:

[orientation_restraints]									
;	ai	aj	type	exp.	label	alpha	const.	obs.	weight
;						Hz	nm^3	Hz	Hz^-2
31	32	1	1	3	3	6.083	-6.73	1.0	
43	44	1	1	4	3	6.083	-7.87	1.0	
55	56	1	1	5	3	6.083	-7.13	1.0	
65	66	1	1	6	3	6.083	-2.57	1.0	
73	74	1	1	7	3	6.083	-2.10	1.0	

观测量的单位为Hz, 但也可以选择任何其他单位. **ai**列与**aj**列为限制粒子的原子编号. **type**列总是为1. **exp.**列代表实验编号, 从1开始. 对每次实验, 会优化单独的序张量 **S**. 对每个限制, 其标签应为大于零的唯一数字. **alpha**列包含了方程4.97中用于计算取向的次数 α . **const.**列包含相同方程中使用的常数 c_i . 这些常数的单位应为观测量乘上 nm $^\alpha$. **obs.**列包含了观测量, 单位任意. 最后一列包含权重 w_i , 其单位为观测量单位平方的倒数.

取向限制的一些参数可在 **grompp.mdp** 文件中设定. 不同力常数, 平均时间和系综平均对结果的影响, 请参考论文[92].

4.4 极化

GROMACS可处理极化, 采用的方法是将壳层(Drude)粒子附着到原子和/或虚拟位点上. 然后在每个时间步对壳层粒子的能量进行最小化, 以维持体系处于Born-Oppenheimer 势能面上.

4.4.1 简单极化

只是平衡距离为零的简谐势.

4.4.2 水极化

用于水的特殊势能, 允许单个壳层粒子的各向异性极化[43].

4.4.3 Thole极化

基于Thole的早期工作[93], Roux 及其合作者实现了对乙醇这样分子的势能[94, 95, 96]. 在这些分子中, 壳层粒子之间存在分子内的相互作用, 然而, 必须对这些相互作用进行屏蔽, 因为完全的库仑作用太强了. 两个壳层粒子 i 和 j 之间的势能为:

$$V_{thole} = \frac{q_i q_j}{r_{ij}} \left[1 - \left(1 + \frac{\bar{r}_{ij}}{2} \right) \exp(-\bar{r}_{ij}) \right] \quad (4.112)$$

注意, 在Noskov 等人的文章中[96], 方程1有一个符号错误:

$$\bar{r}_{ij} = a \frac{r_{ij}}{(\alpha_i \alpha_j)^{1/6}} \quad (4.113)$$

其中 a 为一个魔术常数(无量纲), 通常取2.6[96]; α_i 和 α_j 分别为壳层粒子的极化率.

4.5 自由能相互作用

本节叙述自由能计算中, 势能对 λ 的依赖关系(见3.12节). 所有常见的势能类型与约束类型都可以在状态A($\lambda = 0$)和状态B($\lambda = 1$)之间光滑地进行插值, 反之亦然. 所有键合相互作用都利用对相互作用参数的线性插值进行插值, 非键合相互作用可以使用线性插值或软核相互作用.

从GROMACS4.6开始, λ 是一个向量, 允许自由能转变的不同分量以不同速率进行. 库仑, Lennard-Jones, 键合和限制项都可

以使用 **.mdp** 文件中的选项独立地进行控制.

简谐势

这里给出的例子针对键势能, 在GROMACS中使用的是简谐势. 然而, 这些方程也同样适用于键角势与不当二面角势.

$$V_b = \frac{1}{2} \left[(1 - \lambda)k_b^A + \lambda k_b^B \right] \left[b - (1 - \lambda)b_0^A - \lambda b_0^B \right]^2 \quad (4.114)$$

$$\begin{aligned} \frac{\partial V_B}{\partial \lambda} = & \frac{1}{2} (k_b^B - k_b^A) \left[b - (1 - \lambda)b_0^A + \lambda b_0^B \right]^2 + \\ & (b_0^A - b_0^B) \left[b - (1 - \lambda)b_0^A - \lambda b_0^B \right] \left[(1 - \lambda)k_b^A + \lambda k_b^B \right] \end{aligned} \quad (4.115)$$

GROMOS-96键和键角

对四次键伸缩势与基于余弦的键角势, 通过对力常数与平衡位置的线性插值进行插值, 这里就不给出相应的公式了.

恰当二面角

对恰当二面角, 这些方程更复杂一些:

$$V_d = \left[(1 - \lambda)k_d^A + \lambda k_d^B \right] \left(1 + \cos \left[n_\phi \phi - (1 - \lambda)\phi_s^A - \lambda \phi_s^B \right] \right) \quad (4.116)$$

$$\begin{aligned} \frac{\partial V_d}{\partial \lambda} = & (k_d^B - k_d^A) \left(1 + \cos \left[n_\phi \phi - (1 - \lambda)b_s^A - \lambda \phi_s^B \right] \right) + \\ & (\phi_s^B - \phi_s^A) \left[(1 - \lambda)k_d^A - \lambda k_d^B \right] \sin \left[n_\phi \phi - (1 - \lambda)\phi_s^A - \lambda \phi_s^B \right] \end{aligned} \quad (4.117)$$

注意, 多重性 n_ϕ 不能参数化, 因为函数在 $[0, 2\pi]$ 区间内应当保持周期性.

表格键相互作用

对表格键合相互作用, 只对力常数进行插值:

$$V = \left((1 - \lambda)k^A + \lambda k^B \right) f \quad (4.118)$$

$$\frac{\partial V}{\partial \lambda} = (k^B - k^A) f \quad (4.119)$$

库仑相互作用

对两个粒子之间的库仑相互作用, 粒子电荷随 λ 的变化关系如下:

$$V_c = \frac{f}{\varepsilon_{rf} r_{ij}} \left[(1 - \lambda)q_i^A q_j^A + \lambda q_i^B q_j^B \right] \quad (4.120)$$

$$\frac{\partial V_c}{\partial \lambda} = \frac{f}{\varepsilon_{rf} r_{ij}} \left[-q_i^A q_j^A + \lambda q_i^B q_j^B \right] \quad (4.121)$$

其中 $f = \frac{1}{4\pi\varepsilon_0} = 138.935\ 485$ (见第二章).

反应场库仑相互作用

对两个粒子间包含反应场的库仑相互作用, 粒子电荷随 λ 的变化关系如下:

$$V_c = f \left[\frac{1}{r_{ij}} + k_{rf} r_{ij}^2 - c_{rf} \right] \left[(1-\lambda) q_i^A q_j^A + \lambda q_i^B q_j^B \right] \quad (4.122)$$

$$\frac{\partial V_c}{\partial \lambda} = f \left[\frac{1}{r_{ij}} + k_{rf} r_{ij}^2 - c_{rf} \right] \left[-q_i^A q_j^A + q_i^B q_j^B \right] \quad (4.123)$$

注意, 常数 k_{rf} 和 c_{rf} 是利用介质的介电常数 ϵ_{rf} 进行定义的(参看4.1.4节).

Lennard-Jones相互作用

对两个粒子间的Lennard-Jones 相互作用, 原子类型 随 λ 的变化关系可写为:

$$V_{LJ} = \frac{(1-\lambda)C_{12}^A + \lambda C_{12}^B}{r_{ij}^{12}} - \frac{(1-\lambda)C_6^A + \lambda C_6^B}{r_{ij}^6} \quad (1.124)$$

$$\frac{\partial V_{LJ}}{\partial \lambda} = \frac{C_{12}^B - C_{12}^A}{r_{ij}^{12}} - \frac{C_6^B - C_6^A}{r_{ij}^6} \quad (1.125)$$

应当指出, 从状态A到状态B的途径也可以利用 σ 和 ϵ 进行表达(见方程4.5). 看起来改变力场参数 σ 和 ϵ 而不是导出参数 C_{12}, C_6 物理上可能更合理, 但在参数空间中这两种途径之间的差别并不大, 而且自由能自身与途径无关, 因此我们使用了上面的简单公式.

动能

当粒子的质量改变时, 动能对自由能也有贡献(注意, 我们不能将动量 \mathbf{p} 写为 $m\mathbf{v}$, 因为那样会导致 $\frac{\partial E_k}{\partial \lambda}$ 的符号错误[97]):

$$E_k = \frac{1}{2} \frac{\mathbf{p}^2}{(1-\lambda)m^A + \lambda m^B} \quad (4.126)$$

$$\frac{\partial E_k}{\partial \lambda} = -\frac{1}{2} \frac{\mathbf{p}^2(m^B - m^A)}{((1-\lambda)m^A + \lambda m^B)^2} \quad (4.127)$$

求导后, 可以代入 $\mathbf{p} = m\mathbf{v}$, 得到

$$\frac{\partial E_k}{\partial \lambda} = -\frac{1}{2} \mathbf{v}^2 (m^B - m^A) \quad (4.128)$$

约束

约束是哈密顿量正式的一部分, 因此对自由能也有贡献. 在GROMACS中这可以使用LINCS或SHAKE算法进行计算. 如果我们有一些约束方程 g_k :

$$g_k = \mathbf{r}_k - d_k \quad (4.129)$$

其中 \mathbf{r}_k 为两粒子间的距离向量, d_k 为两粒子间的约束距离. 利用距离与 λ 的关系, 我们可将方程改写为

$$g_k = \mathbf{r}_k - \left((1-\lambda)d_k^A + \lambda d_k^B \right) \quad (4.130)$$

使用Lagrange 乘子, C_λ 对哈密顿量的贡献

$$C_\lambda = \sum_k \lambda_k g_k \quad (4.131)$$

$$\frac{\partial C_\lambda}{\partial \lambda} = \sum_k \lambda_k (d_k^B - d_k^A) \quad (4.132)$$

4.5.1 软核相互作用

在自由能计算中, 当有粒子出现或消失时, 对Lennard-Jones 势和库仑势使用方程4.125和4.123中的简单线性插值, 可能收敛很差. 当粒子几乎消失或接近消失时(λ 接近0或1), 相互作用能会变得足够弱, 粒子彼此可能非常接近, 致使得到的 $\partial V / \partial \lambda$ 值的涨落非常大(因为简单线性插值同时取决于 λ 两端点的值).

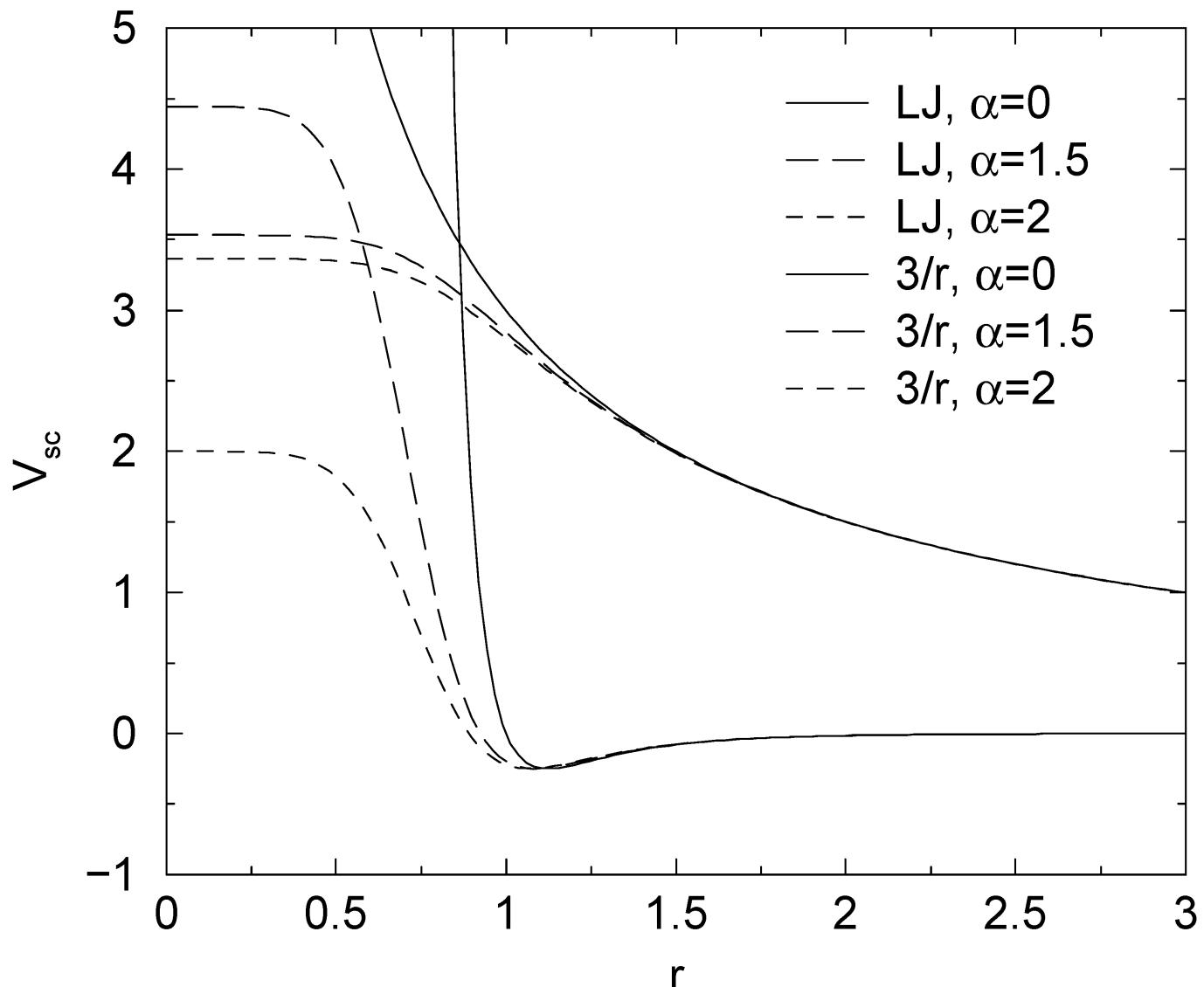


图4.17: $\lambda = 0.5$ 时的软核作用势, 其中 $p = 2$, $C_6^A = C_{12}^A = C_6^B = C_{12}^B = 1$.

为避免这些问题, 需要去除势能的奇异性. 这可以通过利用“软核”势修改常规的Lennard-Jones 势和库仑势来实现, 软核势限制了 λ 值处于0和1之间的能量和力, 但不改变 $\lambda = 0$ 或1时的势能和力.

在GROMACS中, 软核势是常规势能的偏移, 因此势能及其导数的奇点完全不可能出现在 $r = 0$ 处:

$$V_{sc} = (1 - \lambda)V^A(r_A) + \lambda V^B(r_B) \quad (4.133)$$

$$r_A = (\alpha \sigma_A^6 \lambda^p + r^6)^{\frac{1}{6}} \quad (4.134)$$

$$r_B = (\alpha \sigma_B^6 (1 - \lambda)^p + r^6)^{\frac{1}{6}} \quad (4.135)$$

其中 V^A 和 V^B 分别为状态A($\lambda = 0$)和状态B($\lambda = 1$)中正常的“硬核”Van der Waals 势或静电势, α 为软核参数(在 **.mdp** 文件中利用 **sc_alpha** 进行设置), p 为软核 λ 的次数(使用 **sc_power** 进行设置), σ 为相互作用半径, 其值为 $(C_{12}/C_6)^{1/6}$, 当 C_6 或 C_{12} 为零时, σ 是一个输入参数(由 **sc_sigma** 设置).

对中等大小的 λ , r_A 和 r_B 对 $r > \alpha^{1/6}\sigma$ 范围的势能改变很小, 对更小的 r 它们可使软核相互作用快速地变为一个常数(图 4.17). 相应的力为:

$$F_{sc}(r) = -\frac{\partial V_{sc}(r)}{\partial r} = (1-\lambda)F^A(r_A)\left(\frac{r}{r_A}\right)^5 + \lambda F^B(r_B)\left(\frac{r}{r_B}\right)^5 \quad (4.136)$$

其中 F^A 和 F^B 为“硬核”力. 对自由能导数的贡献为:

$$\begin{aligned} \frac{\partial V_{sc}(r)}{\partial \lambda} &= V^B(r_B) - V^A(r_A) + (1-\lambda) \frac{\partial V^A(r_A)}{\partial r_A} \frac{\partial r_A}{\partial \lambda} + \lambda \frac{\partial V^B(r_B)}{\partial r_B} \frac{\partial r_B}{\partial \lambda} \\ &= V^B(r_B) - V^A(r_A) + \\ &\quad \frac{p\alpha}{6} \left[\lambda F^B(r_B) r_B^{-5} \sigma_B^6 (1-\lambda)^{p-1} - (1-\lambda) F^A(r_A) r_A^{-5} \sigma_A^6 \lambda^{p-1} \right] \end{aligned} \quad (4.137)$$

GROMACS原始的Lennard-Jones 软核函数使用的 $p = 2$, 但 $p = 1$ 给出的 $\partial H/\partial \lambda$ 曲线更光滑.

另一个应当考虑的事情是, 软核对那些无Lennard-Jones 相互作用的氢原子的影响. 它们的软核参数 σ 使用 **.mdp** 文件中的 **sc-sigma** 选项设置. 对 $p = 1$, 当 $\lambda = 0$ 和/或1时, 或对 $p = 2$, 当 λ 接近0或1时, 这些氢原子会使 $\partial H/\partial \lambda$ 曲线产生峰值. 降低 **sc-sigma** 的值可减小这种效应, 但相对于软核状态中其他的相互作用, 这种做法会增加与氢原子的相互作用.

当选择软核势时(设置 **sc-alpha**>0), 依次打开和关闭库仑和Lennard-Jones, 然后线性地关闭库仑相互作用, 而不使用软核相互作用, 在大多数情况下, 这种做法的统计噪声会更小. 可将mdp选项 **sc-coul** 设为 **yes** 来取消这种行为. 此外, 软核相互作用只用于当A或B状态具有零相互作用的情况. 若A和B状态都具有非零的相互作用势, 会使用上文所说的线性缩放方法. 当同时关闭库仑和Lennard-Jones 相互作用时, 会使用软核势, 并引入或删除一个氢原子, 将sigma参数设置为 **sc-sigma-min**, 其自身默认值为 **sc-sigma-default**.

最近, 有人推导了软核方法一个新公式, 与上面所述的标准软核途径相比, 在大多数情况下, 它给出的统计方差更低[99, 100].

$$V_{sc}(r) = (1-\lambda)v^A(r_A) + \lambda V^B(r_B) \quad (4.138)$$

$$r_A = \left(\alpha \sigma_A^{48} \lambda^p + r^{48} \right)^{\frac{1}{48}} \quad (4.139)$$

$$r_B = \left(\alpha \sigma_B^{48} (1-\lambda)^p + r^{48} \right)^{\frac{1}{48}} \quad (4.140)$$

GROMACS也实现了这种“1-1-48”途径. 注意对这种途径, 软核参数应满足 $0.001 < \alpha < 0.003$, 而不是 $\alpha \approx 0.5$.

4.6 方法

4.6.1 排除和1-4相互作用

一个分子链中互相邻近的原子, 即那些以共价键结合, 或者通过一个或两个原子相连的原子分别被称为 第一相邻原子, 第二相邻原子 和 第三相邻原子(见图4.18). 由于原子 i 与原子 $i+1$ 和 $i+2$ 之间的相互作用主要是量子力学的, 因此不能用 Lennard-Jones 势进行描述. 作为替代, 我们假定这些相互作用可以利用简谐键项或约束($i, i+1$)和简谐键角项($i, i+2$)进行充分的描述. 因此, 将第一和第二相邻原子(原子 $i+1, i+2$)从原子 i 的Lennard-Jones 相互作用列表中 排除. 原子 $i+1, i+2$ 被称为原子 i 的排除原子.

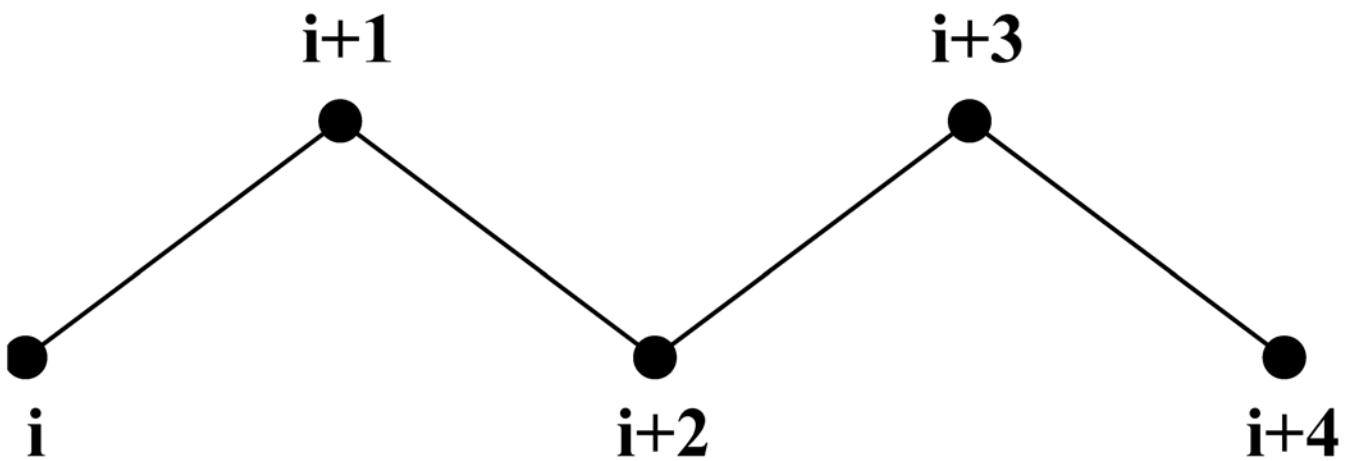


图4.18: 沿烷烃链的原子

对第三个相邻原子, 正常的Lennard-Jones 排斥作用有时仍然过强, 这意味着当将其应用于分子时, 分子会因内部张力发生变形或者破碎, 对顺式构象的碳-碳相互作用情况尤其如此(如顺丁烷). 因此, 对一些这样的相互作用, 其Lennard-Jones 排斥在 GROMACS力场中被减弱了, 这是通过维持一个单独的1–4作用列表和正常的Lennard-Jones 参数列表实现的. 在其他力场中, 如OPLS[101], 标准的Lennard-Jones 参数减少为原来的1/2, 但在这种情况下, 也会对色散(r^{-6})和库伦相互作用进行相应的缩放. GROMACS可以使用这些方法中的任何一种.

4.6.2 电荷组

原则上, MD中力的计算复杂度是 $O(N^2)$. 因此, 对非键力(NBF, non-bonded force) 的计算我们使用了截断: 只有彼此间的距离在一定范围内的粒子才会相互作用. 这将NBF计算的复杂度减少为 $O(N)$ (典型的是 $100N$ 到 $200N$), 但同时也引入了误差. 在大多数情况下, 误差是可以接受的, 除了使用截断会导致电荷产生时. 在这种情况下, 应该考虑使用GROMACS提供的格点加和方法.

让我们考虑水分子与另一个原子的相互作用. 如果在原子-原子的基础上使用普通的截断, 我们可能包括了原子和氧之间的相互作用(电荷为-0.82), 而没有包括质子的平衡电荷, 结果将导致体系中出现了一个很大的偶极矩. 因此, 我们必须保证原子组的总电荷为0. 这些组被称为 电荷组. 注意, 如果对长程静电相互作用使用了好的处理方法(如粒子网格Ewald方法, 见4.8.2节), 就不需要将电荷组维持在一起.

4.6.3 组方案中截断的处理

GROMACS在处理截断时相当灵活, 这意味有大量的参数可供设置. 这些参数在 `grompp` 的输入文件中设置. 影响到截断相互作用的参数有两类: 你可以选择在每种情况下使用哪种相互作用, 在邻区搜索中使用哪种截断.

对于库伦和范德华相互作用有相应的相互作用类型选择词(`vdwtype` 和 `coulombtype` 项)和两个参数, 一共六个非键相互作用参数. 参见7.3节对这些参数的完整描述.

邻区搜索(NS, neighbor searching) 可使用单程或双程方法进行. 由于前者只是后者的特例, 我们将只讨论更一般的双程方法. 在这种情况下, NS会使用两个半径: `rlist` 和 `max(rcoulomb, rvdw)`. 通常每10个时间步或每20 fs(参数 `nstlist`)重新构建一次邻区列表. 在邻区列表中, 会存储所有落在 `rlist` 范围内的相互作用对. 此外, 对于没有落在 `rlist` 范围内但落在 `max(rcoulomb, rvdw)` 范围内的粒子对, 它们之间的相互作用在邻区搜索过程中也会被计算. 力和能量会分开存储, 在连续的NS之间, 在每个积分步会将它们添加到短程力中. 如果 `rlist` = `max(rcoulomb, rvdw)`, 生成邻区列表时不会计算力. 根据短程力与长程力的总和来计算维里, 这意味着在非NS步中维里可能有稍微的不对称. 当使用混合精度的 `mdrun` 时, 维里几乎总是不对称的, 因为非对角元素与加和中的每个元素相差不大. 在大多数情况下, 这些都不是真正的问题, 因为维里的涨落可以比平均值大两个数量级.

除了普通截断, 由于需要使用电荷组, 表4.2中所有的相互作用函数都要求使用比为其形式指定的 r_c 更大的半径进行邻区搜索. 额外半径的典型值是0.25 nm(粗略地等于电荷组中两原子之间的最大距离加上电荷组在邻区列表更新中可扩散的距离).

表4.2: 非键相互作用不同函数形式的参数

类型	参数
库仑	普通截断 r_c, ϵ_r
	反应场 r_c, ϵ_{rf}
	移位函数 r_1, r_c, ϵ_r
	切换函数 r_1, r_c, ϵ_r
VdW	普通截断 r_c
	移位函数 r_1, r_c
	切换函数 r_1, r_c

4.7 虚拟相互作用位点

在GROMACS中可以各种方式使用虚拟相互作用位点(在GROMACS 3.3以前的版本中被称为哑原子). 我们将虚拟位点的位置 \mathbf{r}_s 写为其他粒子位置 \mathbf{r}_i 的函数: $\mathbf{r}_s = f(\mathbf{r}_1 \dots \mathbf{r}_n)$, 这样就可以在力的计算中使用虚拟位点了, 它可能带有电荷或者涉及其他相互作用. 作用于虚拟位点上的力必须以一种恰当的方式被重新分配到有质量的粒子上. 在参考文献[102]中可以找到一个较好的方. 我们可以将势能写为:

$$V = V(\mathbf{r}_s, \mathbf{r}_1, \dots, \mathbf{r}_n) = V^*(\mathbf{r}_1, \dots, \mathbf{r}_n) \quad (4.141)$$

作用于粒子 i 上的力为:

$$\mathbf{F}_i = -\frac{\partial V^*}{\partial \mathbf{r}_i} = -\frac{\partial V}{\partial \mathbf{r}_i} - \frac{\partial V}{\partial \mathbf{r}_s} \frac{\partial \mathbf{r}_s}{\partial \mathbf{r}_i} = \mathbf{F}_i^{direct} + \mathbf{F}'_i \quad (4.142)$$

第一项是正常的力, 第二项是因虚拟位点引起的作用于粒子 i 上的力, 可以将它写为张量形式:

$$\mathbf{F}'_i = \begin{bmatrix} \frac{\partial x_s}{\partial x_i} & \frac{\partial y_s}{\partial x_i} & \frac{\partial z_s}{\partial x_i} \\ \frac{\partial x_s}{\partial y_i} & \frac{\partial y_s}{\partial y_i} & \frac{\partial z_s}{\partial y_i} \\ \frac{\partial x_s}{\partial z_i} & \frac{\partial y_s}{\partial z_i} & \frac{\partial z_s}{\partial z_i} \end{bmatrix} \mathbf{F}_s \quad (4.143)$$

其中 \mathbf{F}_s 为虚拟位点上的力, x_s, y_s 和 z_s 为虚拟位点的坐标. 这样, 合力和总的力矩都是守恒的.

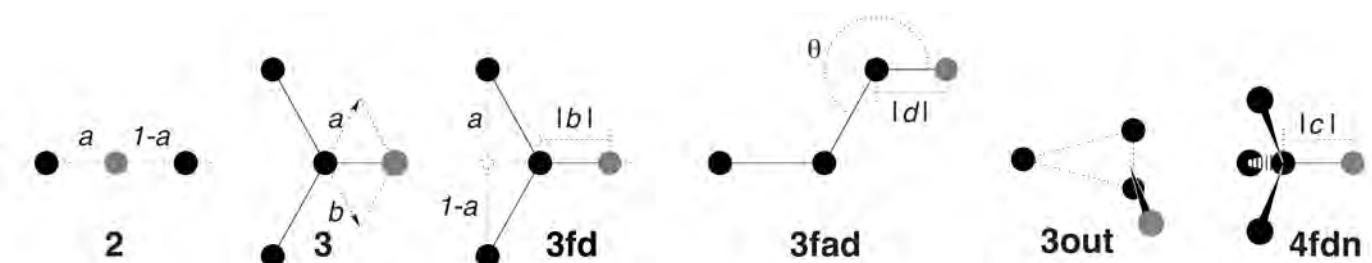


图4.19: GROMACS中六种不同类型虚拟位点的构建方式. 黑色为构建原子, 灰色为虚拟位点.

当使用虚拟位点时, 维里的计算(方程3.24)并不简单. 由于维里涉及到所有原子的加和(而不仅虚拟位点), 在计算维里之前, 虚拟位点上的力必须被重新分配到原子上(使用方程4.143). 在一些特殊的情况下, 原子上的力可以写为虚拟位点上的力的线性组

合(下面的类型2和类型3), 计算维里时是否重新分配力没有区别. 然而, 一般情况下, 应该首先进行力的重新分配.

在GROMACS中, 从周围原子构建虚拟位点的方法共有六种, 我们按构建原子的数目对其进行分类. 注意, 所有提到的位点类型都可以利用类型3fd(归一化的, 平面内)和3out(为归一化, 平面外)进行构造. 然而, 涉及到的计算量按列表顺序迅速增大, 因此我们强烈推荐使用第一个适合的虚拟位点类型, 只要它能满足一定的目的. 图4.19给出了6种可用的虚拟位点构建方式. 概念上最简单的构建类型是线性组合:

$$\mathbf{r}_s = \sum_{i=1}^N \omega_i \mathbf{r}_i \quad (4.144)$$

使用相同的权重分重新分配力:

$$\mathbf{F}'_i = \omega_i \mathbf{F}_s \quad (4.145)$$

GROMACS支持的虚拟位点类型将在下面列出. 虚拟位点中的构建原子也可以是虚拟位点自身, 但仅当它们处于列表中的较高位置时, 即虚拟位点可以利用比虚拟位点简单的“粒子”进行构建.

2. 两个原子的线性组合(图4.19 2):

$$\omega_i = 1 - a, \omega_j = a \quad (4.146)$$

这种情况下虚拟位点位于通过原子 i 和 j 的线上.

3. 三个原子的线性组合(图4.19 3):

$$\omega_i = 1 - a - b, \omega_j = a, \omega_k = b \quad (4.147)$$

这种情况下虚拟位点位于另外三个原子构成的平面上.

3fd. 在三个原子构成的平面中, 具有固定的距离(图4.19 3fd):

$$\mathbf{r}_s = \mathbf{r}_i + b \frac{\mathbf{r}_{ij} + a\mathbf{r}_{jk}}{|\mathbf{r}_{ij} + a\mathbf{r}_{jk}|} \quad (4.148)$$

这种情况下虚拟位点位于其他三个粒子所构成的平面中, 与原子 i 的距离为 $|b|$. 由虚拟位点上的力引起的粒子 i , j 和 k 上的力为:

$$\begin{aligned} \mathbf{F}'_i &= \mathbf{F}_s - \gamma(\mathbf{F}_s - \mathbf{p}) \\ \mathbf{F}'_j &= (1 - a)\gamma(\mathbf{F}_s - \mathbf{p}) \\ \mathbf{F}'_k &= a\gamma(\mathbf{F}_s - \mathbf{p}) \\ \gamma &= \frac{b}{|\mathbf{r}_{ij} + a\mathbf{r}_{jk}|} \\ \mathbf{p} &= \frac{\mathbf{r}_{is} \cdot \mathbf{F}_s}{\mathbf{r}_{is} \cdot \mathbf{r}_{is}} \mathbf{r}_{is} \end{aligned} \quad (4.149)$$

3fad. 在三个原子所构成的平面中, 具有固定的角度和距离(图 4.19 3fad):

$$\begin{aligned} \mathbf{r}_s &= \mathbf{r}_i + d \cos \theta \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} + d \cos \theta \frac{\mathbf{r}_\perp}{|\mathbf{r}_\perp|} \\ \mathbf{r}_\perp &= \mathbf{r}_{jk} - \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{r}_{ij} \end{aligned} \quad (4.150)$$

这种情况下虚拟位点位于其他三个粒子所构成的平面中, 与原子 i 的距离为 $|d|$, 与 \mathbf{r}_{ij} 所成的角度为 α . 原子 k 定义了平面以及角度的方向. 注意, 在这种情况下必须明确指定 b 和 α 的, 以代替 a 和 b (也可参见5.2.2节). 由虚拟位点上的力引起的粒子 i, j 和 k 上的力为(\mathbf{r}_\perp 的定义如方程4.150):

$$\begin{aligned}
 \mathbf{F}'_i &= \mathbf{F}_s - \frac{d \cos \theta}{|\mathbf{r}_{ij}|} \mathbf{F}_1 + \frac{d \sin \theta}{|\mathbf{r}_\perp|} \left(\frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{F}_2 + \mathbf{F}_3 \right) \\
 \mathbf{F}'_j &= \frac{d \cos \theta}{|\mathbf{r}_{ij}|} \mathbf{F}_1 - \frac{d \sin \theta}{|\mathbf{r}_\perp|} \left(\mathbf{F}_2 + \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{F}_2 + \mathbf{F}_3 \right) \\
 \mathbf{F}'_k &= \frac{d \sin \theta}{|\mathbf{r}_\perp|} \mathbf{F}_2 \\
 \mathbf{F}_1 &= \mathbf{F}_s - \frac{\mathbf{r}_{ij} \cdot \mathbf{F}_s}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{r}_{ij} \\
 \mathbf{F}_2 &= \mathbf{F}_1 - \frac{\mathbf{r}_\perp \cdot \mathbf{F}_s}{\mathbf{r}_\perp \cdot \mathbf{r}_\perp} \mathbf{r}_\perp \\
 \mathbf{F}_3 &= \frac{\mathbf{r}_{ij} \cdot \mathbf{F}_s}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}} \mathbf{r}_\perp
 \end{aligned} \tag{4.151}$$

3out. 三个原子的非线性组合, 平面外(图4.19 3out):

$$\mathbf{r}_s = \mathbf{r}_i + a\mathbf{r}_{ij} + b\mathbf{r}_{ik} + c(\mathbf{r}_{ij} \times \mathbf{r}_{ik}) \tag{4.152}$$

构建的虚拟位点位于其他原子构成的平面外. 由虚拟位点上的力引起的粒子 i, j 和 k 上的力为:

$$\begin{aligned}
 \mathbf{F}'_j &= \begin{bmatrix} a & -cz_{ik} & cy_{ik} \\ cz_{ik} & a & -cx_{ik} \\ -cy_{ik} & cx_{ik} & a \end{bmatrix} \mathbf{F}_s \\
 \mathbf{F}'_k &= \begin{bmatrix} b & cz_{ij} & -cy_{ij} \\ -cz_{ij} & b & cx_{ij} \\ cy_{ij} & -cx_{ij} & b \end{bmatrix} \mathbf{F}_s \\
 \mathbf{F}'_i &= \mathbf{F}_s - \mathbf{F}'_j - \mathbf{F}'_k
 \end{aligned} \tag{4.153}$$

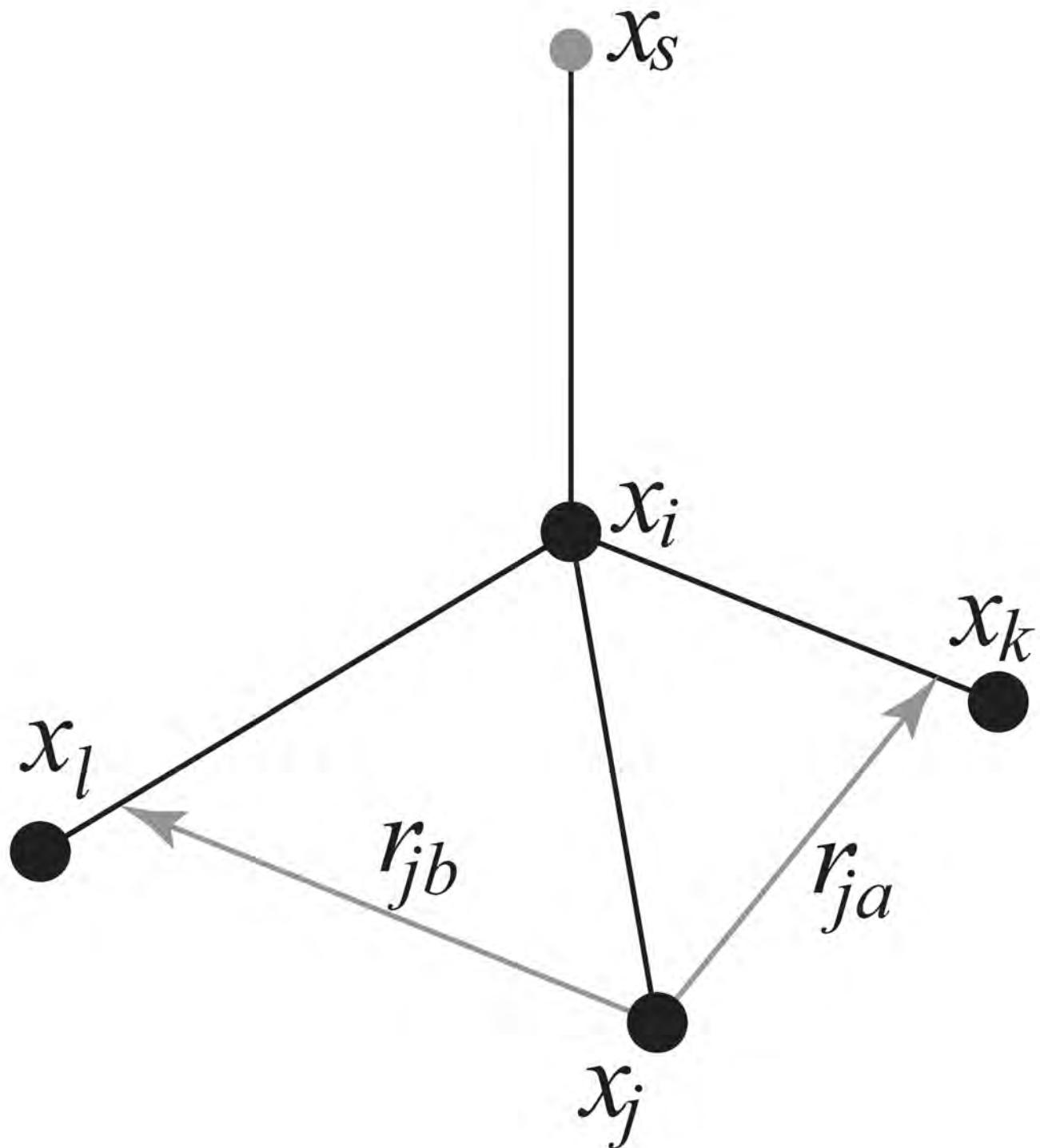


图4.20: 新的4fdn 虚拟位点的构建方法. 即便所有构建原子都处于同一平面内时, 这种虚拟位点也是稳定的.

4fdn. 由四个原子构建, 具有固定的距离, 见图4.20. 这种构建有点复杂. 由于之前的构建类型(4fd)不稳定, 这迫使我们引入了这种更加复杂的构建:

$$\begin{aligned}
 \mathbf{r}_{ja} &= a\mathbf{r}_{ik} - \mathbf{r}_{ij} = a(\mathbf{x}_k - \mathbf{x}_i) - (\mathbf{x}_j - \mathbf{x}_i) \\
 \mathbf{r}_{jb} &= b\mathbf{r}_{il} - \mathbf{r}_{ij} = b(\mathbf{x}_l - \mathbf{x}_i) - (\mathbf{x}_j - \mathbf{x}_i) \\
 \mathbf{r}_m &= \mathbf{r}_{ja} \times \mathbf{r}_{jb} \\
 \mathbf{x}_s &= \mathbf{x}_i + c \frac{\mathbf{r}_m}{|\mathbf{r}_m|}
 \end{aligned} \tag{4.154}$$

这种情况下虚拟位点与原子 i 的距离为 $|c|$, a 和 b 是参数. 注意, 为节省浮点运算的时间, 没有对矢量 \mathbf{r}_{ik} 和 \mathbf{r}_{ij} 进行归一化. 由虚拟位点上的力引起的粒子 i , j 和 k 上的力可通过对构造表达式使用链式规则求导计算. 这种构建精确并保持能量守恒, 但确实导致了相对冗长的表达式, 在这里我们没有给出所有表达式(超过200次浮点运算). 感兴趣的读者可以阅读 `vsite.c` 中的代码. 幸运的是, 这种虚拟位点类型通常只用于手性中心, 例如蛋白质中的 C_α 原子.

新的4fdn构造在拓扑中的'type' 识别值为2. GROMACS内部依然支持以前的4fd类型('type' 值为1), 但不应将其用于新的模拟. 所有目前的GROMACS程序都会自动创建4fdn 类型作为替代.

N. N 个原子的线性组合, 相对权重为 a_i . 原子 i 的权重为:

$$\omega_i = a_i \left(\sum_{j=1}^N a_j \right)^{-1} \quad (4.155)$$

设置权重时有三个选项:

- COG 几何中心: 等权重
- COM 质量中心: a_i 为原子 i 的质量. 在自由能模拟中, 原子的质量会改变, 只有A状态的质量会用于权重
- COW 权重中心: a_i 由用户定义

4.8 长程静电作用

4.8.1 Ewald 加和

N 个粒子及其周期映象的总静电能由下式给出:

$$V = \frac{f}{2} \sum_{n_x} \sum_{n_y} \sum_{n_z*} \sum_i^N \sum_j^N \frac{q_i q_j}{\mathbf{r}_{ij,n}} \quad (4.156)$$

$(n_x, n_y, n_z) = \mathbf{n}$ 为盒子的索引矢量, 星号表示当 $(n_x, n_y, n_z) = (0, 0, 0)$ 时, 应忽略 $i = j$ 项. 距离 \mathbf{r}_{ij} 为电荷之间的真距离, 而不是最小映象之间的距离. 此加和条件收敛, 但收敛非常缓慢.

首次引入时, Ewald 加和被用于计算晶体中周期映象的长程相互作用[103]. 采取的方法是将方程4.156中收敛缓慢的单个加和转换为两个快速收敛的项与一个常数项:

$$V = V_{\text{dir}} + V_{\text{rec}} + V_0 \quad (4.157)$$

$$V_{\text{dir}} = \frac{f}{2} \sum_{i,j}^N \sum_{n_x} \sum_{n_y} \sum_{n_z*} q_i q_j \frac{\text{erfc}(\beta r_{ij,n})}{r_{ij,n}} \quad (4.158)$$

$$V_{\text{rec}} = \frac{f}{2\pi V} \sum_{i,j}^N q_i q_j \sum_{m_x} \sum_{m_y} \sum_{m_z*} \frac{\exp(-(\pi \mathbf{m}/\beta)^2 + 2\pi i \mathbf{m} \cdot (\mathbf{r}_i - \mathbf{r}_j))}{\mathbf{m}^2} \quad (4.159)$$

$$V_0 = -\frac{f\beta}{\sqrt{\pi}} \sum_i^N q_i^2 \quad (4.160)$$

其中 β 参数用于决定直接空间加和与倒易空间加和之间的相对权重, $\mathbf{m} = (m_x, m_y, m_z)$. 这样, 我们可以对直接空间加和使用较短的截断(数量级为1 nm), 对倒易空间加和也可以使用较短的截断(例如每个方向10个波矢). 不幸的是, 倒易空间加和的计算量以 N_2 增加(或 $N^{3/2}$, 如果使用更好一点的算法), 因此用于大的体系是不现实的.

使用Ewald方法

不要使用Ewald 方法, 除非你绝对确定你需要使用. 几乎对所有的情况, 下面的PME方法表现都更好. 如果你依然坚持使用精度的Ewald 加和方法, 在你的 `.mdp` 文件中输入以下内容, 如果盒子的边长3 nm左右:

```
coulombtype      = Ewald
rvdw             = 0.9
rlist            = 0.9
rcoulomb         = 0.9
fourierspacing  = 0.6
ewald-rtol       = 1e-5
```

盒子尺寸的比例和 `fourierspacing` 参数决定了每个方向使用的波矢 m_x, m_y, m_z 的最大振幅. 例子中对3 nm的立方盒子每个方向将使用11个波矢(从-5到5). `ewald-rtol` 参数为在截断处静电相互作用的相对强度. 减少此值可以得到更加精确的直接空间加和, 但倒易空间加和的精度会略微降低.

4.8.2 PME

Tom Darden 提出的粒子网格Ewald 方法[12]可提高倒易空间加和的计算速度. 这种方法不直接对波矢进行加和, 而是使用内插方法将电荷分配到网格上. GROMACS实现的PME方法使用了基数B样条插值[13], 通常被称为平滑PME(SPME). 先使用3D FFT算法对格点进行傅里叶变换, 在k空间中利用对格点的单个加和就可以得到倒易空间的能量项.

格点上的势能可利用逆变换进行计算, 通过使用内插因子就可以得到每个原子上的力.

PME算法的复杂度为 $N \log(N)$, 对中等或大的体系计算速度远远高于普通的Ewald 加和方法. 对非常小的体系, 使用Ewald 方法可能更好, 因为可以避免建立格点和进行变换的过程. PME的并行化可参见MPMD PME部分(3.17.5).

使用Verlet 截断方案时, PME直接空间的势能被移动了一个常数, 使得在截断处的势能为零. 这种移动很小, 由于体系的电荷接近于零, 总的移动非常小, 这与Lennard-Jones 势能的情况不同, 那种情况下所有的移动都会被累加起来. 我们使用了移动, 这样力是势能的精确积分.

使用PME

作为在GROMACS中使用PME加和方法的例子, 在你的 `.mdp` 文件中指定如下几行:

```
coulombtype      = PME
rvdw             = 0.9
rlist            = 0.9
rcoulomb         = 0.9
fourierspacing  = 0.12
pme-order        = 4
ewald-rtol       = 1e-5
```

在这个例子中, `fourierspacing` 参数决定了FFT格点的最大间距(即, 格点数的最小值), `pme-order` 控制插值的阶数. 使用四阶(立方)内插与指定的间距, 静电能应该能精确到 $5 \cdot 10^{-3}$. 由于Lennard-Jones 能量的精确度更低一些, 还可以轻微地增加间距.

PME可用于压力缩放, 但当心, 在一些体系中各项异性缩放可能会导致虚假的有序结构.

4.8.3 P3M-AD

在GROMACS中, 也可使用Hockney 和Eastwood 发展的粒子-粒子-网格方法来处理长程静电相互作用[104]. 动力学计算中, 尽管P3M方法是第一个用于分子模拟的高效长程静电方法, 但在原子模拟中, P3M方法很大程度上已经被平滑PME方法(SPME)所取代. 原始P3M方法计算时有一个缺点: 它需要3次3D-FFT后变换来获得粒子上的力. 但P3M并不需要这样, 可以通过对势能进行数值微分得到所需的力, 像在PME中一样. 这样的方法被称为P3M-AD. P3M-AD和PME唯一的不同在于: PME对晶

格Green影响函数进行了优化以尽量减小误差. 然而, 在2012年有研究表明, 可以修改SPME影响函数得到P3M[105]. 这意味着P3M-AD误差最小化的优点在PME中可以同样的计算代价和同样的代码获得, 只需增加几行代码修改影响函数. 然而, 对最佳参数设置, P3M-AD误差最小化的效果小于10%. 对于隔行(也称为交错)的格点, P3M-AD确实展现了很大的精度提升, 但GROMACS(目前)并不支持这种处理.

在GROMACS中使用P3M时, 所用的选项与PME完全相同, 只需要选择不同的静电类型即可.

coulombtype = P3M-AD

4.8.4 优化傅里叶变换和PME计算

建议对计算静电相互作用的参数进行优化, 如PME格点尺寸和截断半径. 在运行长的模拟之前, 这尤其重要.

GROMACS中有一个特殊的工具, **g_tune_pme**, 可用于自动选择格点的最佳大小和PME节点的数目.

4.9 长程范德华相互作用

4.9.1 色散校正

在这一节中, 我们将推导长程校正, 这是由于对Lennard-Jones 或Buckingham 相互作用使用截断而引起的. 我们假定截断足够大以至于可以安全地忽略斥力项, 因此只须考虑色散项的校正. 由于色散相互作用的本质(我们截断了一个正比于 $-r^6$ 的势能), 能量和压力的校正都是负的. 尽管能量的校正通常很小, 对自由能计算却可能很重要, 因计算时需要考虑两种不同哈密顿量之间的差异. 与此相反, 压力的校正很大, 当需要正确的压力时, 在任何情况下都不能忽略, 尤其是对于NPT模拟. 虽然原则上在力场参数化时可以不经校正就使得压力接近期望的实验值, 但这种方法使得参数依赖于截断, 因此不够理想.

能量

如果我们假定超出截断距离 r_c 后体系变为均相, 色散相互作用对维里的长程贡献就能够解析得到. 两个粒子之间的色散能可写为

$$V(r_{ij}) = -C_6 r_{ij}^{-6} \quad (4.161)$$

相应的力为:

$$\mathbf{F}_{ij} = -6C_6 r_{ij}^{-8} \mathbf{r}_{ij} \quad (4.162)$$

在周期性体系中计算全部的势能不容易, 所以通常会使用间断的或光滑的截断. 记截断处的势能和力分别为 V_c 和 \mathbf{F}_c , 对含有 N 个粒子, 粒子密度 $\rho = N/V$ 的体系, 色散能的长程贡献为:

$$V_{lr} = \frac{1}{2} N \rho \int_0^\infty 4\pi r^2 g(r) (V(r) - V_c(r)) dr \quad (4.163)$$

我们将使用移位函数对上式进行积分, 它是GROMACS中VdW相互作用的最一般形式. 从0至 r_1 , 移位函数具有恒定的差值 S , 超过截断距离 r_c 后, 移位函数为零. 积分方程4.163, 并假定在半径为 r_1 的球体内, 粒子密度等于整体密度, 径向分布函数 $g(r)$ 在 r 超过 r_1 后为1:

$$\begin{aligned}
V_{lr} &= \frac{1}{2}N \left(\rho \int_0^{r_1} 4\pi r^2 g(r) C_6 S dr + \rho \int_{r_1}^{r_c} 4\pi r^2 (V(r) - V_c(r)) dr + \rho \int_{r_c}^{\infty} 4\pi r^2 V(r) dr \right) \\
&= \frac{1}{2}N \left(\left(\frac{4}{3}\pi\rho r_1^3 - 1 \right) C_6 S + \rho \int_{r_1}^{r_c} 4\pi r^2 (V(r) - V_c(r)) dr - \frac{4}{3}\pi N \rho C_6 r_c^{-3} \right)
\end{aligned} \quad (4.164)$$

其中, -1 项为自相互作用的校正, 对普通截断, 我们只需要假定超过 r_c 后 $g(r)$ 等于1, 这样校正可化简为[106]:

$$V_{lr} = -\frac{2}{3}\pi N \rho C_6 r_c^{-3} \quad (4.165)$$

举例来说, 如果我们考虑一个只含水分子的模拟盒子, 截断半径为0.9 nm, 密度为1 g cm⁻³, 上面的校正对每个水分子大约是-0.75 kJ mol⁻¹.

对均相的混合物, 我们需要定义一个 平均色散常数

$$\langle C_6 \rangle = \frac{2}{N(N-1)} \sum_i^N \sum_{j>i}^N C_6(i,j) \quad (4.166)$$

在GROMACS中, 计算此平均值时不考虑被排除的原子对.

对非均相模拟体系, 如脂质界面系统, 若两部分的 $\langle C_6 \rangle$ 比较接近, 能量校正也可以适用

维里和压力

体系的标量维里可通过两粒子 i 和 j 之间的色散相互作用计算:

$$\Xi = \frac{1}{2} \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} = 3C_6 r_{ij}^{-6} \quad (4.167)$$

相应的压力由下式给出:

$$P = \frac{2}{3V} (E_{kin} - \Xi) \quad (4.168)$$

对维里的长程校正为:

$$\Xi_{lr} = \frac{1}{2}N\rho \int_0^{\infty} 4\pi r^2 g(r)(\Xi - \Xi_c) dr \quad (4.169)$$

假定超过 r_1 后 $g(r)$ 为1, 积分得到维里的长程校正:

$$\begin{aligned}
\Xi_{lr} &= \frac{1}{2}N\rho \left(\int_{r_1}^{r_c} 4\pi r^2 (\Xi - \Xi_c) dr + \int_{r_c}^{\infty} 4\pi r^2 3C_6 r_{ij}^{-6} dr \right) \\
&= \frac{1}{2}N\rho \left(\int_{r_1}^{r_c} 4\pi r^2 (\Xi - \Xi_c) dr + 4\pi C_6 r_c^{-3} \right)
\end{aligned} \quad (4.170)$$

对普通截断, 压力的校正值为:

$$P_{lr} = -\frac{4}{3}\pi C_6 \rho^2 r_c^{-3} \quad (4.171)$$

对前面水盒子的例子, 每个分子的维里校正是0.75 kJ mol⁻¹, 对SPC水模型相应的压力校正大约是-280 bar.

对于均相混合物, 我们同样可以使用平均色散常数 $\langle C_6 \rangle$ (方程4.166):

$$P_{lr} = -\frac{4}{3}\pi \langle C_6 \rangle \rho^2 r_c^{-3} \quad (4.172)$$

对于非均相体系, 使用方程4.172时的限制与在能量的情况下相同(参见4.9.1节).

4.9.2 Lennard-Jones PME

为了利用Lennard-Jones 势能处理截断以外非均相的体系, 我们可以使用粒子网格Ewald方法, 像上面对静电的讨论一样. 在这种情况下修改的Ewald 方程变为:

$$V = V_{\text{dir}} + V_{\text{rec}} + V_0 \quad (4.173)$$

$$V_{\text{dir}} = \frac{1}{2} \sum_{i,j}^N \sum_{n_x} \sum_{n_y} \sum_{n_z*} \frac{C_6^{ij} g(\beta r_{ij,n})}{r_{ij,n}^6} \quad (4.174)$$

$$V_{\text{rec}} = \frac{\pi^{\frac{3}{2}} \beta^3}{2V} \sum_{m_x} \sum_{m_y} \sum_{m_z*} f(\pi|\mathbf{m}|/\beta) \times \sum_{i,j}^N C_6^{ij} \exp[-2\pi i \mathbf{m} \cdot (\mathbf{r}_i - \mathbf{r}_j)] \quad (4.175)$$

$$V_0 = -\frac{\beta^6}{12} \sum_i^N C_6^{ii} \quad (4.176)$$

(4.1)

其中 $\mathbf{m} = (m_x, m_y, m_z)$, 参数 β 决定了直接空间和倒易空间的权重, C_6^{ij} 为粒子 i 和 j 的组合色散参数. 星号表示当 $(n_x, n_y, n_z) = (0, 0, 0)$ 时, 应忽略 $i = j$ 的项. $\mathbf{r}_{ij,n}$ 为粒子之间的距离. 根据Essmann 的推导[13], 上面引入的函数 f 和 g 的定义为:

$$f(x) = 1/3 \left[(1 - 2x^2) \exp(-x^2) + 2x^3 \sqrt{\pi} \operatorname{erfc}(x) \right] \quad (4.177)$$

$$g(x) = \exp(-x^2)(1 + x^2 + \frac{x^4}{2}) \quad (4.178)$$

只要色散参数可以像静电电荷一样按几何规则进行组合(方程4.6), 上面的方法就适用

$$C_{6,\text{geom}}^{ij} \left(C_6^{ii} C_6^{jj} \right) \quad (4.179)$$

对Lorentz-Berthelot 组合规则(方程4.7), 倒易部分的加和必须计算七次, 因为色散参数的拆分如下:

$$C_{6,\text{L-B}}^{ij} = (\sigma_i + \sigma_j)^6 = \sum_{n=0}^6 P_n \sigma_i^n \sigma_j^{(6-n)} \quad (4.180)$$

P_n 为Pascal 三角系数. 对倒易部分, 这导致了不可忽略计算代价, 需要七个独立的FFT, 因此这成为了以前实现LJ-PME的限制因素. 解决这个问题的一个方法是, 使用几何组合规则计算势能倒易部分的近似的相互作用参数, 得到总的相互作用:

$$\begin{aligned} V(r < r_c) &= \underbrace{C_6^{\text{dir}} g(\beta r) r^{-6}}_{\text{直接空间}} + \underbrace{C_{6,\text{geom}}^{\text{recip}} [1 - g(\beta r)] r^{-6}}_{\text{倒易空间}} \\ &= C_{6,\text{geom}}^{\text{recip}} r^{-6} + \left(C_6^{\text{dir}} - C_{6,\text{geom}}^{\text{recip}} \right) g(\beta r) r^{-6} \end{aligned} \quad (4.181)$$

$$V(r > r_c) = \underbrace{C_{6,\text{geom}}^{\text{recip}} [1 - g(\beta r)] r^{-6}}_{\text{倒易空间}} \quad (4.182)$$

这样可以很好的定义哈密顿量, 且可以显著提高模拟速度. 这个近似确实引入了一些误差, 但由于差异在于倒易空间中计算的相互作用, 与总的相互作用能相比, 影响非常小. 在双脂层模拟中, 使用1.0 nm的截断, 总色散能的相对误差小于0.5%. 更完整的的讨论可参看[107].

在GROMACS中, 我们现在已经能够正确地计算这项相互作用, 方法是从直接空间相互作用中减去倒易空间中使用的近似势能的贡献:

$$V_{\text{dir}} = C_6^{\text{dir}} r^{-6} - C_6^{\text{recip}} [1 - g(\beta r)] r^{-6} \quad (4.183)$$

当 $C_6^{\text{dir}} = C_6^{\text{recip}}$ 时, 这项势能将简化为方程4.174中的表达式, 总的相互作用为:

$$V(r < r_c) = \underbrace{C_6^{\text{dir}} r^{-6} - C_6^{\text{recip}} [1 - g(\beta r)] r^{-6}}_{\text{直接空间}} + \underbrace{C_6^{\text{recip}} [1 - g(\beta r)] r^{-6}}_{\text{倒易空间}} \quad (4.184)$$

$$V(r > r_c) = C_6^{\text{recip}} [1 - g(\beta r)] r^{-6} \quad (4.185)$$

当 $C_6^{\text{dir}} \neq C_6^{\text{recip}}$ 时, 直到截断处都会存在未经修改的LJ力, 误差比模拟时直接空间相互作用未考虑倒易空间中使用的近似而导致的误差小一个数量级. 当采用移位的范德华相互作用时, 常数

$$\left(-C_6^{\text{dir}} + C_6^{\text{recip}} [1 - g(\beta r_c)] \right) r_c^{-6} \quad (4.1)$$

会加到方程4.184中以确保势能在截断处连续. 注意, 当 $C_6^{\text{dir}} = C_6^{\text{recip}}$ 时上式退化为预期的 $-C_6 g(\beta r_c) r_c^{-6}$, 如方程4.183类似. 除此之外, 可以利用长程色散校正来校正在倒易空间中使用组合规则导致的近似. 这种校正假定, 对截断LJ势能, 粒子分布是均匀的. 但由于组合规则近似导致的误差非常小, 在大多数情况下, 没有必要进行长程校正. 同样注意, 这样的均相校正并不能校正表面张力, 它是非均相性质.

使用LJ-PME

作为在GROMACS中使用使用粒子网格Ewald加和方法计算Lennard-Jones 相互作用的例子, 在你的 `.mdp` 文件中指定如下几行:

```
vdwtype      = PME
rvdw         = 0.9
vdw-modifier = Potential-Shift
rlist        = 0.9
rcoulomb    = 0.9
fourierspacing = 0.12
pme-order   = 4
ewald-rtol-lj = 0.001
lj-pme-comb-rule = geometric
```

如果同时启用LJ-PME和静电PME, 应使用同样的傅里叶格点和插值阶数, 因此 `fourierspacing` 和 `pme-order` 的设置对二者相同. `ewald-rtol-lj` 控制直接空间和倒易空间之间的划分, 方式与 `ewald-rtol` 相同. 除此之外, 倒易空间使用的组合规则由 `lj-pme-comb-rule` 决定. 如果当前的力场使用Lorentz-Berthelot 组合规则, 设置 `lj-pme-comb-rule = geometric` 可获得显著的性能提升, 同时精度会略微降低. 有关此方法的细节可在前面的章节中找到.

注意, 使用完全的长程色散校正意味着, 当与库伦PME联用时, `rvdw` 是一个自由参数, 而不必受限于力场的参数化方案. 这样, 为了提高精度和计算速度, 可以对截断, 间距, 阶数和容差项进行优化.

自然, 使用LJ-PME而不是LJ截断会增加倒易空间部分的计算和通信. 因此, 为了使用单纯PME队列并行模拟的最佳负载均衡, 应该使用更多的队列. 这可能改进 `mdrun` 使用的自动负载均衡.

4.10 力场

一个力场由两个明显不同的两部分组成:

- 用于产生势能及其导数, 也即力的方程组(称为s). 这些在前边的章节已有详细的论述.
- 用于方程组的参数. 本手册未给出这些参数, 但可以在GROMACS程序的相关数据文件中找到.

在一组方程之内, 可使用各种不同的参数集. 必须当心, 方程与参数的组合应自洽. 一般来说, 后验地改变一些参数是很危险的, 因为对总力的各种贡献是相互关联的. 这意味着原则上在使用之前, 应该记录每一项改动, 通过与实验数据比较进行验证, 发表在预审的杂志上.

GROMACS 5.0.2包括了一些力场, 网上还有额外的一些力场可供使用. 如果你不清楚应该选择哪个力场, 我们推荐GROMOS-96用于联合原子设置, OPLS-AA/L 用于全原子参数. 我们将对可用的选项进行详细描述.

全氢力场

基于GROMOS-87的全氢力场几乎等同于正常的GROMOS-87力场, 因为额外的氢原子没有Lennard-Jones 相互作用并且电荷为零. 唯一的区别在于键角项和异常二面角项. 此力场仅仅适用于需要精确的氢原子位置的情况, 例如来自NMR测量的距离限制. 当要使用此力场时, 请阅读前面的段落.

4.10.1 GROMOS-96

GROMACS支持GROMOS-96力场[78], 并包含了43A1, 43A2(发展, 改进了烷烃二面角), 45A3, 53A5和53A6参数集中的所有参数. 此外还包括了所有的标准构建单元, 并可以利用 `pdb2gmx` 程序自动创建拓扑.

GROMOS-96力场是GROMOS-87力场的进一步发展, 对GROMOS-87力场的蛋白质和小分子进行了改进. 注意, 53A6中的糖类参数确实对应于2004年发表的参数[108], 与45A4中的不同, GROMACS此次为包含这些参数. 45A4参数集对应于这些参数的后一个版本. 然而, 我们不推荐将GROMOS-96力场用于长的烷烃与脂类分子. GROMOS-96力场与GROMOS-87力场的不同之处在于:

- 力场参数
- 键合相互作用参数并未与原子类型相关联
- 四次键伸缩势(4.2.1)
- 基于键角余弦的键角势(4.2.6)

GROMACS和GROMOS-96的实现有两个不同, 当使用两个程序来模拟同样的体系时, 这可能导致结果有少许的不同:

- GROMOS-96中对溶剂的邻区搜索根据溶剂分子的第一个原子进行. GROMACS没有使用这种方法, 但使用电荷组中心进行搜索产生的差别很小
- GROMOS-96中的维里是基于分子的. GROMACS没有使用这种方法, 而是使用了原子的维里

GROMOS-96力场参数化时对Lennard-Jones 使用的截断值为1.4 nm, 因此请确保使用的Lennard-Jones 截断([rvdw](#))至少为1.4. 也可以使用更大的截断, 因为超多1.4 nm后Lennard-Jones 势能和力几乎为零.

GROMOS-96文件

GROMACS可以读写GROMOS-96的坐标文件和轨迹文件. 这些文件的扩展名应为[.g96](#), 文件可以是GROMOS-96的初始/最终构型文件, 坐标轨迹文件或二者的结合. 文件是固定格式的, 所有浮点数的输出格式为15.9, 因而文件可能会变得非常大. 按给定的顺序, GROMACS支持以下的数据块:

- Header block 文件头数据块:

TITLE (mandatory) 标题(必需)

- Frame blocks 帧数据块:

TIMESTEP (optional) 时间步(可选)

POSITION/POSITIONRED (mandatory) 位置(必需)

VELOCITY/VELOCITYRED (optional) 速度(可选)

BOX (optional) 盒子(可选)

参看GROMOS-96手册中对数据块的详细描述[78]. 注意 所有GROMACS程序都可以读取压缩(.Z)或gzip压缩(.gz)的文件.

4.10.2 OPLS/AA

4.10.3 AMBER

GROMACS原生支持以下AMBER力场:

- AMBER94 [109]
- AMBER96 [110]
- AMBER99 [111]
- AMBER99SB [112]
- AMBER99SB-ILDN [113]
- AMBER03 [114]
- AMBERGS [115]

4.10.4 CHARMM

GROMACS支持CHARMM力场, 这些力场可用于蛋白质[116, 117], 脂类[118]和核酸[119, 120]. 其中的蛋白质参数(一定程度上的脂类和核酸参数)进行过彻底的验证, 验证时, 会计算势能并将其与利用CHARMM分子模拟包中的标准参数集计算的势能进行比较, 也会考察蛋白质力场在GROMACS特定技术中的表现, 如虚拟位点(支持较大的时间步长)和最近实现的快速隐式溶剂[73]. 验证的细节及其结果可参看Bjelkmar 等人的论文[121]. 核酸及HEME相关参数的转换和测试由Michel Cuendet 完成.

当在[pdb2gmx](#) 中选择CHARMM力场时, 默认的选项是使用CMAP(用于扭转校正映射). 若想排除CMAP, 请使用[-nocmap](#) 选项. GROMACS中实现的CMAP项的基本形式为骨架扭转角 ϕ 和 ψ 的函数. CMAP项在[.rtp](#) 文件中定义, 对每个支持CMAP的残

基, 在结尾处使用 [**cmap**] 语句定义CMAP. 接着的五个原子名称定义了两个扭转角. 原子1–4定义 ϕ , 原子2–5定义 ψ . 相应的原子类型会被匹配到 **cmap.itp** 文件中正确的CMAP类型, 它们包含了校正映射.

GROMACS中使用的CHARMM36力场可在http://mackerell.umaryland.edu/charmm_ff.shtml#gromacs 找到.

4.10.5 粗粒化力场

粗粒化是减少体系自由度数的一种系统方法. 为此, 典型的做法是以单个珠子代表整个原子组, 粗粒化的力场用于描述珠子之间的有效相互作用. 根据参数化方法的选择, 这些相互作用的函数形式可以很复杂, 且通常使用表格势能.

粗粒模型被设计用于重现参考体系的某些性质, 可以是全原子模型的性质甚至是实验数据. 依据重现的性质, 可使用不同的方法来得到这样的力场. 以下是各种方法的一个不完全列表:

- 守恒自由能
 - 单纯形方法
 - MARTINI力场(参看下一节)
- 守恒分布(如径向分布函数), 所谓的基于结构的粗粒化
 - (迭代)Boltzmann 逆
 - 逆Monte Carlo
- 守恒力
 - 力匹配

注意, 粗粒势能依赖于状态(例如温度, 密度等), 应该根据体系的特点和模拟条件进行重新参数化. 你可以利用, 例如Versatile Object-oriented Toolkit for Coarse-Graining Applications(VOTCA) [122], 来进行重新参数化. 这个软件包被设计用于协助进行系统的粗粒化, 实现了上面提到的大多数算法, 并且具有经过充分测试的GROMACS界面. 它是开源软件, 更多信息请浏览 www.votca.org .

4.10.6 MARTINI

MARTINI力场是一组粗粒化参数, 可用于构建许多体系, 包括蛋白质和膜.

4.10.7 PLUM

PLUM力场[123]无溶剂蛋白-膜模型的一个例子, 其中的膜由基于结构的粗粒化导出[124]. 可以在code.google.com/p/plumx 找到GROMACS的实现.

第五章 拓扑文件

5.1 简介

GROMACS运行时需要知道哪些原子及其组合对势能函数有贡献(参见第四章). 此外, 它还需要知道对于不同函数必需的参数. 所有这些都利用 **拓扑文件** `*.top` 进行描述, 它列出了每个原子的 **固定属性**. 原子类型比元素种类多得多, 但力场只对存在于生物系统中的原子类型, 外加一些金属, 离子和硅进行了参数化. 键合相互作用和一些特殊的相互作用由拓扑文件中的固定配对所决定, 必须排除某些非键相互作用(第一和第二近邻), 因为它们已经在键合相互作用中处理过了. 另外, 原子还具有 **动态属性**, 如位置, 速度和力. 严格地说, 这些不属于分子拓扑, 而是存储在坐标文件 `*.gro` (位置和速度)或轨迹文件 `*.trr` (位置, 速度, 力).

本章将描述拓扑文件, `*.top` 文件和数据库文件的设置: 各个参数代表什么, 如果需要如何/到哪里修改它们. 首先, 会解释了所有的文件格式. 然后, 在5.8.1小节说明在每个力场的文件组织形式.

注意: 如果你创建了自己的拓扑文件, 我们诚挚地希望你能够将其上传到位于www.gromacs.org 的拓扑文件存档. 想一想, 如果在开始计算之前, 那里就已经存有你需要的拓扑文件了, 你肯定会感谢那些贡献的人. 如果你创建了新的力场或者对标准力场进行了修改, 我们同样希望你能够将其上传到力场存档.

5.2 粒子类型

在GROMACS中, 共有三种类型的粒子, 参见表 5.1. GROMACS只使用了常规原子和虚拟相互作用位点, 壳层只对极化模型, 如壳层水模型, 才是需要的[43].

表 5.1: GROMACS的粒子类型

粒子	符号
原子	A
壳层	S
虚拟相互作用位点	V(或D)

5.2.1 原子类型

每个力场都定义了一组原子类型, 它们具有特征名称或编号, 质量(以a.m.u为单位). 这些原子类型的列表可以在 `atomtypes.atp` (.atp = atom type parameter, 原子类型参数)文件中找到. 因此, 你可以从这个文件开始修改和/或增加原子类型. 下面的示例原子类型来自于 `gromos43a1.ff` 力场

```

O 15.99940 ; carbonyl oxygen (C=O)
OM 15.99940 ; carboxyl oxygen (CO-)
OA 15.99940 ; hydroxyl, sugar or ester oxygen
OW 15.99940 ; water oxygen
N 14.00670 ; peptide nitrogen (N or NH)
NT 14.00670 ; terminal nitrogen (NH2)
NL 14.00670 ; terminal nitrogen (NH3)
NR 14.00670 ; aromatic nitrogen
NZ 14.00670 ; Arg NH (NH2)

```

```

NE 14.00670 ; Arg NE (NH)
C 12.01100 ; bare carbon
CH1 13.01900 ; aliphatic or sugar CH-group
CH2 14.02700 ; aliphatic or sugar CH2-group
CH3 15.03500 ; aliphatic CH3-group

```

注意: GROMACS以名称来使用原子类型, 而不是编号(参考GROMOS的例子)

5.2.2 虚拟位点

一些力场会使用虚拟相互作用位点(由其他粒子的位置构建的相互作用位点), 并为这些位点设置一些相互作用(例如, 对苯环可产生正确的四极矩), 这些将在4.7中详细介绍.

为了在你的体系中生成虚位点, 你需要在拓扑文件中包含 [virtual_sites?] 段(为向后兼容也可以使用老版本中的名称 [dummmies?]), 其中 ? 代表构成虚拟位点的粒子数目, 对类型2为 2, 对类型3, 3fd, 3fad 和3out 为 3, 对类型4fdn 为 4. 类型4fdn 取代了旧的4fd 类型('type' 值为1), 因为它有时会不稳定. 尽管程序内部仍然支持4fd 类型, 不过最好不要在新的输入文件中使用. 不同类型将在4.7节解释.

类型2的参数看起来应该像这样:

```

[ virtual_sites2 ]
; Site   from     funct    a
5       1 2       1      0.7439756

```

类型3的像这样:

```

[ virtual_sites3 ]
; Site   from     funct    a        b
5       1 2 3     1      0.7439756  0.128012

```

类型3fd的像这样:

```

[ virtual_sites3 ]
; Site   from     funct    a        d
5       1 2 3     2      0.5      -0.105

```

类型3fad的像这样:

```

[ virtual_sites3 ]
; Site   from     funct    theta    d
5       1 2 3     3      120      0.5

```

类型3out的像这样:

```

; Site   from     funct    a        b        c

```

5 1 2 3 4 -0.4 -0.4 6.9281

类型4fdn 的像这样:

```
[ virtual_sites4 ]
; Site   from      funct    a        b        c
5       1 2 3 4     2        1.0      0.9      0.105
```

这就构成了虚拟位点, 编号5(第一列 **Site**)基于其他原子的位置, 这些原子的索引号为1和2, 或者1, 2和3, 或者1, 2, 3和4(接下来的第2, 3或4列 **from**), 构建时所遵循的规则由函数类型编号(下一列 **func**)以及具体的参数(最后1, 2或3列 **a b c**)决定。显然, 原子数目(包括虚拟位点数)取决于分子。研究一下GROMCAS自带的TIP4P或TIP5P水模型的拓扑文件对理解这些可能很有帮助。

注意, 如果在虚拟位点和/或普通原子之间定义了任何固定的键合相互作用, **grompp** 命令将会移除它们(除非使用 **-normvsbds** 选项)。键合相互作用的移除是在生成排除之后, 因为生成排除是以“化学”的键相互作用为基础。

可以使用更普遍的方法构建虚拟位点, 利用的指令是 **[virtual_sustesn]**。所需要的参数列在表5.5中。下面是将一组给定原子的几何中心定义为虚拟位点的例子:

```
[ virtual_sustesn ]
; Site   funct      from
5       1           1 2 3 4
```

表 5.2: GROMACS中的静态原子类型性质

性质	符号	单位
类型	-	-
质量	m	a.m.u
电荷	q	e
epsilon	ε	kJ/mol
sigma	σ	nm

5.3 参数文件

5.3.1 原子

原子类型 静态性质 (参见表 5.2)的指定基于几个地方的数据。质量来源于 **atomtypes.apt** 文件(参见5.2.1节), 电荷来源于 ***.rtp** (.rtp = residue topology parameter, 残基拓扑参数, 参见5.6.1节)文件。这就意味着只对构建氨基酸, 核酸的基本单元定义了电荷, 对其他的分子, 用户需要自己定义电荷。当使用 **pdb2gmx** 程序生成一个拓扑文件 (***.top**)时, 来自这些文件的信息将被整合在一起。

5.3.2 非键参数

非键参数包括van der Waals 参数V(**c6** 或 σ , 由组合规则决定)和W(**c12** 或 ϵ), 它们列在 **ffnonbonded.itp** 文件中,

其中的 **ptype** 是粒子类型(参见表 5.1). **[*type]** 指令中的条目和键合参数会被应用它们在拓扑文件中的相应部分. 除了将在5.3.4节中提到的那些, 缺少参数将导致警告.

```
[ atomtypes ]
; name      at.num      mass      charge      ptype      V(c6)      W(c12)
    O          8      15.99940     0.000        A      0.22617E-02     0.74158E-06
    OM         8      15.99940     0.000        A      0.22617E-02     0.74158E-06
    .....

[ nonbond_params ]
; i      j      func      V(c6)      W(c12)
    O      O          1      0.22617E-02     0.74158E-06
    O      OA         1      0.22617E-02     0.13807E-05
    .....
```

注意: GROMACS所包含的大部分力场也带有 **at.num.** 列, 但相同的信息位于OPLS_AA **bond_type** 列. 参数V与W的含义取决于拓扑文件 **[defaults]** 段中选择的组合规则(参见5.7.1节).

对组合规则1:

$$\begin{aligned} V_{ii} &= C_i^{(6)} = 4\epsilon_i \sigma_i^6 \quad [\text{kJ mol}^{-1} \text{nm}^6] \\ W_{ii} &= C_i^{(12)} = 4\epsilon_i \sigma_i^{12} \quad [\text{kJ mol}^{-1} \text{nm}^{12}] \end{aligned} \quad (5.1)$$

对组合规则2和3:

$$\begin{aligned} V_{ii} &= \sigma_i \quad [\text{nm}] \\ W_{ii} &= \epsilon_i \quad [\text{kJ mol}^{-1}] \end{aligned} \quad (5.2)$$

对不同原子类型间的一些或所有组合都可以在 **[nonbond_params]** 段给出, 参数V与W的定义同上. 对于其他没有给出的任何原子组合, 将根据组合原则, 利用相应原子类型的参数进行计算.

对组合规则1和3:

$$\begin{aligned} C_{ij}^{(6)} &= \sqrt{C_i^{(6)} C_j^{(6)}} \\ C_{ij}^{(12)} &= \sqrt{C_i^{(12)} C_j^{(12)}} \end{aligned} \quad (5.3)$$

对组合规则2:

$$\begin{aligned} \sigma_{ij} &= \frac{1}{2}(\sigma_i + \sigma_j) \\ \epsilon_{ij} &= \sqrt{\epsilon_i \epsilon_j} \end{aligned} \quad (5.4)$$

当需要提供 σ 和 ϵ 时(规则2和3), 看起来不可能使用非零的 C^{12} 与零值的 C^6 参数进行组合. 然而, 提供负值的 σ 恰好会这样做, C^6 被设为零, C^{12} 正常计算. 这只是代表了读入 σ 值的一种特殊情况, 没有其他的.

对Buckingham 势只有一种组合规则:

$$A_{ij} = \sqrt{A_{ii} A_{jj}}$$

$$B_{ij} = 2 / \left(\frac{1}{B_{ii}} + \frac{1}{B_{jj}} \right)$$

$$C_{ij} = \sqrt{C_{ii} C_{jj}}$$
(5.5)

5.5.3 键参数

键参数(如, 键长, 键角, 异常/正常二面角)都列在 `ffbonded.itp` 文件中. 这个数据库中的条目分别给出了相互作用的原子类型, 相互作用的类型, 与相互作用有关的参数. 在处理拓扑文件时, `grompp` 程序会读取这些参数, 然后应用到相关的键参数中. 例如将 `bondtypes` 应用到 `[bonds]` 指令中的条目, 等等. 从相关的 `[*type]` 指令中缺失的任何键参数都会导致致命错误. 相互作用的类型列于表 5.5. 下面是从这些文件中摘录的例子:

```
[ bondtypes ]
; i   j     func      b0      kb
C   O       1    0.12300  502080.
C   OM      1    0.12500  418400.
.....
[ angletypes ]
; i   j   k     func      th0      cth
HO  OA   C       1    109.500  397.480
HO  OA   CH1     1    109.500  397.480
.....
[ dihedraltypes ]
; i   l     func      q0      cq
NR5* NR5      2    0.000  167.360
NR5* NR5*     2    0.000  167.360
.....
[ dihedraltypes ]
; j   k     func      phi0     cp      mult
C   OA       1    180.000  16.736    2
C   N        1    180.000  33.472    2
.....
[ dihedraltypes ]
;
; Ryckaert-Bellemans Dihedrals
;
; aj   ak   funct
CP2 CP2      3    9.2789  12.156  -13.120  -3.0597  26.240  -31.495
```

在 `ffbonded.itp` 文件中, 你可以添加键参数. 如果你想为新的原子类型增加参数, 请确保你已经在 `atomtypes.atp` 中定义了它们.

5.3.4 分子内的对相互作用

分子中原子对之间额外的Lennard-Jones 和静电相互作用可以添加到分子定义部分的 `[pairs]` 段中. 这些相互作用的参数可以独立于非键相互作用参数进行设置. 在GROMOS力场中, `[pairs]` 仅仅用于修改1-4相互作用(相隔3条键的两个原子之间的相互作用). 在这些力场中, 1-4相互作用并不包括在非键相互作用中(参见5.4节).

`[pairtypes]`

```
; i      j    func          cs6          cs12 ; THESE ARE 1-4 INTERACTIONS
0      0      1    0.22617E-02    0.74158E-06
0      OM     1    0.22617E-02    0.74158E-06
....
```

`ffnonbonded.itp`文件中原子类型的对相互作用参数位于`[pairtypes]`段。GROMOS力场显式地列出了所有这些相互作用的参数，但对于OPLS这样的力场这一段可能是空的，因为这些力场通过统一地缩放参数来计算1-4相互作用。对于那些不在`[pairtypes]`段出现的对参数，只能当`forcefield.itp`文件中`[defaults]`指令的`gen-pair`设置为“yes”时才能生成（参见5.7.1节）。当`gen-pairs`设置为“no”时，`grompp`程序会对每个未设定参数的对类型产生警告。

普通对相互作用和1-4相互作用的函数类型为1。函数类型2和`[pairs_nb]`特别用于自由能模拟。当计算水合自由能时，需要将溶质与溶剂去耦合。这可通过添加一个B-状态拓扑（参见3.12节）实现，其中所有溶质的非键参数，即电荷和LJ参数，都被设置为0。然而，A状态和B状态之间自由能的差值并不是总的水合自由能。我们必须通过重新引入真空中的溶质分子内部的库伦和LJ相互作用来增加自由能。当溶质内的库伦和LJ相互作用未修改时，第二步可以与第一步结合起来。为此，引入了对函数类型2，它与函数类型1完全相同，除了B状态参数与A状态参数始终相同。

在`[pairtypes]`段搜索参数，函数类型1和2之间并没有什么差别。对相互作用段`[pair_nb]`用于取代非键相互作用。它使用未缩放的电荷与非键LJ参数，并只使用A状态的参数。注意要为`[pairs_nb]`中列出的所有原子对添加排除，否则这些原子对也将在正常的邻区列表中结束。

作为替代，通过使用`couple-moltypes`，`couple-lambda0`，`couple-lambda1`，和`couple-intramol`关键字，我们可以不修改拓扑文件而获得同样的行为。具体请参考3.12节和6.1节。

所有这三种对类型全都使用普通的库伦作用，即便当使用反应场、PME、Ewald或移位库伦相互作用来计算非键相互作用时。类型1和2的能量会写入能量和日志文件，其中每个能量组对都有单独的“LJ-14”和“Coulomb-14”项。`[pairs_nb]`的能量会添加到“LJ-(SR)”和“Coulomb-(SR)”项中。

5.3.5 隐式溶剂化模型

GROMACS自4.5版本开始支持隐式溶剂。拓扑文件中引入了一个段用于列出参数：

```
[ implicit_genborn_params ]
; Atomtype    sar      st      pi      gbr      hct
NH1        0.155    1      1.028    0.17063   0.79 ; N
N          0.155    1      1      0.155      0.79 ; Proline backbone N
H          0.1      1      1      0.115      0.85 ; H
CT1        0.180    1      1.276    0.190      0.72 ; C
```

在这个例子中，先列出了原子类型，后面有5个数字和注释（分号后面）。

目前并未使用1-3列的值。它们与更精细的表面积算法相关，特别是Qiu等提出的一个[70]。第4列为原子的范德华半径，用于计算波恩半径。电介质偏移在`*.mdp`文件中指定，对不同的波恩半径方法会从输入的范德华半径减去此偏移，如Onufriev等的描述[72]。第5列是HTC和OBC模型的缩放因子，数值来源于Tinker程序实现的HTC成对缩放方法[71]。这种方法被修改过，缩放因子通过HTC算法被调整得使解析的表面积和GB表面积之间的偏差最小。根据

Hawkins[71] 等的建议, 缩放被进一步修改为不是成对的, 而是基于每个原子(而不是每对原子).

5.4 排除

`grompp`程序会对相邻直到一定数目键的原子生成非键相互作用的排除, 如在拓扑文件的 [`moleculename`] 段中定义的那样(参见5.7.1节). 当彼此之间以“化学”键([`bonds`] 类型1到5, 7或8)或约束([`constraints`] 类型1)连接时, 粒子被认为是键合在一起的. 类型5 [`bonds`] 可用于创建两个原子之间无相互作用的连接. 有一种不通过化学键连接原子的简谐相互作用([`bonds`] 类型6), 也有一种不通过化学键连接原子而固定距离的第二类约束类型([`constraints`] 类型2). 所有这些相互作用的完整列表见表 5.5.

分子内额外的排除可以手动在 [`exclusions`] 段中添加. 每行必须以一个原子编号开始, 后面跟着一个或多个原子编号. 第一个原子和其他原子之间的所有非键相互作用都会被排除.

当需要排除原子组内部或彼此之间的非键相互作用时, 使用能量监测组排除会更方便和高效(参见3.3节).

5.5 约束算法

约束的定义在 [`constraints`] 段中, 其格式为两个原子编号, 后面跟着函数类型和约束距离. 函数类型可以为1或2, 它们之间的唯一区别在于, 类型1用于产生排除而类型2不产生排除(参见5.4节). 距离是通过在 `*.mdp` 文件中选择的LINCS或SHAKE算法进行约束的. 在自由能计算中, 通过增加第二个约束距离, 这两种类型的约束都可以进行微扰(参见5.7.5节). `grompp`程序可自动将一些类型的键或键角(参见表 5.5)转变为约束, 在 `*.mdp` 文件中有很多相关的选项.

我们也实现了SETTLE算法[45], 它对SHAKE进行解析求解, 只用于水. 可以在拓扑文件中选择SETTLE. 例如, 请参看SPC水分子的定义:

```
[ moleculename ]
; molname      nrexcl
SOL          1

[ atoms ]
; nr      at type    res nr      ren nm      at nm      cg nr      charge
  1       OW        1        SOL        OW1        1        -0.82
  2       HW        1        SOL        HW2        1         0.41
  3       HW        1        SOL        HW3        1         0.41

[ settles ]
; OW      funct      doh      dhh
  1       1        0.1      0.16333

[ exclusions ]
1     2     3
2     1     3
3     1     2
```

[`settles`] 指令定义了水分子的第一个原子, SETTLE函数类型始终为1, 必须给出O-H和H-H之间的距离. 注意, SETTLE算法也可用于TIP3P和TIP4P水分子模型[125]. TIP3P具有另一个构型, TIP4P具有一个虚拟位点, 但由于是生成的, 不需要抖动(或扰动).

5.6 `pdb2gmx` 输入文件

GROMACS的`pdb2gmx`程序可以根据输入的坐标文件产生拓扑文件, 它支持几种不同格式的坐标文件, 但`*.pdb`是最常用的(也因此程序名称为`pdb2gmx`). `pdb2gmx`程序运行时会在GROMACS `share/top` 目录的子目录和你的工作目录中搜索力场, 并根据扩展名为`.ff`的目录中的`forcefield.itp`文件识别力场. 目录中可能存在`forcefield.doc`文件, 如果存在, `pdb2gmx`会将此文件的第一行作为力场的简单描述显示给用户, 以帮助用户选择力场. 否则, 用户可以使用`pdb2gmx`的命令行参数`-ff xxx`来选择力场, 所选的力场将位于`xxx.ff`目录. 搜索力场时, `pdb2gmx`会首先搜索工作目录, 然后再搜索GROMACS的`share/top`目录, 并使用找到的第一个匹配`xxx.ff`的目录.

`pdb2gmx`会读入两个通用文件: 位于力场目录的原子类型文件(扩展名为`.atp`, 参见5.2.1节), 位于工作目录或GROMACS `share/top` 目录的`residuetypes.dat`文件. `residuetypes.dat`文件决定了哪些残基名称将被视为蛋白质, DNA, RNA, 水和离子.

对不同类型的分子, `pdb2gmx`可以读入一个或多个数据库及拓扑信息. 属于同一个数据库的一组文件应具有相同的基准名称(basename), 基准名称最好能够对分子类型有所说明(如, 氨基酸, rna, dna). 可能的文件如下:

- `<basename>.rtp`
- `<basename>.r2b` (可选)
- `<basename>.arn` (可选)
- `<basename>.hdb` (可选)
- `<basename>.n.tdb` (可选)
- `<basename>.c.tdb` (可选)

只有包含了构建单元拓扑信息的`.rtp`文件是必需的, 来自于其他文件的信息只用于具有相同基准名的`.rtp`文件中的构建单元. 通过在工作目录中放置具有相同基准名的额外文件, 用户也可以为力场添加新的构建单元. 默认只能定义一个额外的构建单元, 但使用`-rtpp`选项调用`pdb2gmx`程序时, 可以使用本地文件中的构建单元来代替力场中默认的构建单元.

5.6.1 残基数据库

残基数据库文件的扩展名为`.rtp`. 这个文件原本包含蛋白质的构建单元(氨基酸), 是GROMACS对GROMOS `rt37c4.dat`文件的解释说明, 因此残基数据库文件包含常用构建单元的信息(键, 电荷, 电荷组, 异常二面角). 最好不要更改这个文件, 因为它是`pdb2gmx`的标准输入. 但如果确实需要修改, 请修改工作目录中的`*.top`文件(参见5.7.1节), 或`.rtp`文件, 像5.6节解释的那样. 通过直接包含`*.itp`拓扑文件定义一个新的小分子的拓扑可能更容易一些, 具体作法将在5.7.2节讨论. 当添加新的蛋白质残基到数据库中时, 别忘了将残基名称添加到`residuetypes.dat`文件中, 这样`grompp`, `make_ndx`和分析程序才能将残基识别为蛋白质残基(参见8.1.1节).

`.rtp`文件只会被`pdb2gmx`程序使用. 如前面提到的, `pdb2gmx`程序仅仅从`.rtp`数据库中读入键, 原子电荷, 电荷组和异常二面角这些额外信息, 其余的信息是从坐标输入文件读入的. 一些蛋白质的坐标文件中包含非标准氨基酸. 你

必须为这些“陌生”残基创建构建单元, 否则你将无法得到‘*.top’文件. 对坐标文件中的一些分子, 如配体, 多原子离子, 晶格化溶剂分子等, 也是一样. 残基数据库可以根据以下方法创建:

```
[ bondedtypes ] ; mandatory 必需
; bonds      angles      dihedrals      impropers
    1          1            1             2 ; mandatory 必需

[ GLY ] ; mandatory 必需

[ atoms ] ; mandatory 必需
; name      type      charge      chargegroup
    N        N       -0.280        0
    H        H        0.280        0
    CA       CH2      0.000        1
    C        C        0.380        2
    O        O       -0.380        2

[ bonds ] ; optional 可选
; atom1     atom2     b0      kb
    N        H
    N        CA
    CA       C
    C        O
    -C       N

[ exclusions ] ; optional 可选
; atom1     atom2

[ angles ] ; optional 可选
; atom1     atom2     atom3     th0      cth

[ dihedrals ] ; optional 可选
; atom1     atom2     atom3     atom4     phi0      cp      mult
; atom1     atom2     atom3     atom4     q0      cq
    N        -C       CA        H
    -C       -CA      N        -O

[ ZN ]

[ atoms ]
    ZN      ZN      2.000      0
```

文件是自由格式, 唯一的限制是每行最多一个条目. 文件中的第一个域为 [**bondedtypes**], 后面跟着四个数字, 分别代表键, 键角, 二面角和异常二面角的相互作用类型. 文件中的残基条目包含原子和(可选的)键, 键角, 二面角, 异常二面角. 电荷组代码代表电荷组的编号, 相同电荷组的原子应该始终按顺序排列. 当使用 **pdb2gmx** 程序及氢数据库添加缺失的氢原子时(参见5.6.4), **.rtp** 条目中定义的原子名称应该精确地对应于氢数据库中使用的名称约定. 键相互作用中的原子名称前可添加减号或加号, 分别代表原子处于残基之前或残基之后. 添加到键, 键角, 二面角和异常二面角的显式参数会覆盖 **.itp** 文件中的标准参数, 但只能用于特殊情况. 除参数外, 也可为每个键相互作用添加字符串. GROMOS-96的 **.rtp** 文件就是这样. 这些字符串会被复制到拓扑文件, 通过使用 **grompp** C预处理器的 **#define** 语句, 可用力场参数替换这些字符串.

pdb2gmx 程序会自动生成所有的键角, 这意味着对大多数力场 [**angles**] 域仅用于覆盖 **.itp** 参数. 对GROMOS-96力场必须指定所有键角的相互作用编号.

pdb2gmx 程序会自动为每个旋转键生成一个正常二面角, 倾向位于重原子上. 当使用 [**dihedrals**] 域时, 不会为对

应于指定二面角的键生成其他二面角. 可以为一条旋转键指定多个二面角函数. 对CHARMM27力场, 使用 `pdb2gmx` 程序默认的 `-cmap` 选项可为二面角增加校正图. 详细信息请参考4.10.4节.

`pdb2gmx` 会设置排除数为3, 这意味着最多以3条键连接的原子之间的相互作用都被排除了. 程序会为相隔三条键的所有原子对生成对相互作用(氢原子除外). 当需要排除更多的相互作用, 或不需要生成一些对相互作用时, 可添加 [`exclusions`] 域, 后面跟着位于不同行上的原子名称对, 这些原子之间的所有非键和对相互作用都将被排除.

5.6.2 残基构建单元数据库

每个力场对残基都有自己的命名约定. 大部分残基都具有一致的命名, 但有一些残基, 特别是那些具有不同质子化状态的残基, 可能具有许多不同的名称. `.r2b` 文件可用于将标准的残基名称转换为力场构建单元名称.

表 5.3 GROMACS内部的残基名称约定

简写	英文名称	含义
ARG	protonated arginine	质子化精氨酸
ARGN	neutral arginine	中性精氨酸
ASP	negatively charged aspartic acid	带负电的天冬氨酸
ASPH	neutral aspartic acid	中性天冬氨酸
CYS	neutral cysteine	中性半胱氨酸
CYS2	cysteine with sulfur bound to another cysteine or a heme	通过硫与另一个半胱氨酸或血红素结合的半胱氨酸
GLU	negatively charged glutamic acid	带负电的谷氨酸
GLUH	neutral glutamic acid	中性谷氨酸
HISD	neutral histidine with N _δ protonated	N _δ 质子化的中性组氨酸
HISE	neutral histidine with N _ε protonated	N _ε 质子化的中性组氨酸
HISH	positive histidine with both N _δ and N _ε protonated	N _δ 和 N _ε 质子化的带正电的组氨酸
HIS1	histidine bound to a heme	结合到血红素的组氨酸
LYSN	neutral lysine	中性赖氨酸
LYS	protonated lysine	质子化赖氨酸
HEME	heme	血红素

如果力场目录中不存在 `.r2b` 文件, 或残基未被列出, 会假定构建单元的名称与残基名称相同. `.r2b` 文件可包含2或5列. 2列格式为, 第一列为残基名称, 第二列为构建单元名称. 5列格式具有3个附加列, 分别为出现在N端, C端和同时出现在两个末端的残基(单个残基分子)的构建单元. 这对一些力场有用, 例如AMBER力场. 如果不存在一个或多个末端, 应在相应的列中输入短划线.

对残基, 存在GROMACS的命名约定, 此约定仅通过 `.r2b` 文件和 `specbond.dat` 文件显现出来(除 `pdb2gmx` 代码外). 只有当你添加残基类型到 `.rtp` 文件时, 这个约定才变得重要. 此约定列在表5.3中. 对于特殊的键, 如与血红素基团相连的键, GROMACS命名约定通过 `specbond.dat` 引入(参见5.6.7节), 如果需要, 此约定随后可以利用 `.r2b` 文件进行翻译.

5.6.3 原子重命名数据库

力场中使用的原子名称经常不遵循IUPAC或PDB约定。.arn数据库用于将坐标文件中的原子名称转换为力场中的名称。未列出的原子会保持原有名称。.arn文件有三列，分别为构建单元名称，旧的原子名称和新的原子名称。残基名称支持问号通配符，用以匹配单个字符。

share/top 目录下还存在一个通用的原子重命名文件 **xlateat.dat**，可以将坐标文件中常见的非标准原子名称转换为IUPAC/PDB约定名称。因此，当编写力场文件时，你可以使用标准的原子名称，除了将其翻译为力场名称外，不需要进一步的翻译。

5.6.4 氢数据库

氢数据库储存在 .hdb 文件中，它包含了关于 **pdb2gmx** 程序如何将氢原子连接到已有原子的信息。在GROMACS 3.3 版本以前的数据库中，根据连接的原子对氢原子进行命名：把连接原子名称的首字母用'H'代替。从3.3版本开始，必须明确列出氢原子，因为以前的做法仅适用于蛋白质，因而不能推广用于其它分子。如果有一个以上的氢原子连接到相同的原子，氢原子名称的末尾将添加一个数字。例如，添加两个氢原子到 ND2(天冬酰胺)，氢原子将被命名为 HD21 和 HD22。这很重要，因为 .rtp 文件(参见5.6.1节)中的命名必须相同。氢数据库的格式如下：

```
; res      # additions
# H add type    H   i   j   k
ALA      1
        1      H   N   -C   CA
ARG      4
        1      H   N   CA   C
        1      HE  NE  CD   CZ
        2      HH1 NH1 CZ   NE
        2      HH2 NH2 CZ   NE
```

第一行为残基名称(ALA或ARG)以及氢原子的类型数，这些氢原子可根据氢数据库添加到残基中。后面的行每行对应于一个氢原子的添加：

- 添加的氢原子数
- 添加氢原子的方法，可能的方法如下：

1 单个平面氢原子，如环或肽键

产生一个氢原子(n)，置于原子(i,j,k)形成的平面内，位于角(j-i-k)的平分线上，距离原子i 0.1 nm，并使角(n-i-j)和(n-i-k)大于90度。

2 单个氢原子，例如羟基

产生一个氢原子(n)，距离原子i 0.1 nm，并使角(n-i-j)为109.5度，二面角(n-i-j-k)为反式。

3 两个平面氢原子，如乙烯 -C=CH₂或酰胺 -C(=O)NH₂

产生两个氢原子(n1,n2)，距离原子i 0.1 nm，并使角(n1-i-j)与(n2-i-j)为120度，二面角(n1-i-j-k)为顺式，(n2-i-j-k)为反式，这样命名符合IUPAC标准[126]。

4 两个或三个四面体氢原子，例如-CH₃

产生三个(n1,n2,n3)或两个(n1,n2)氢原子，距离原子i 0.1 nm，并使角(n1-i-j), (n2-i-j), (n3-i-j) 都为109.47

度, 二面角($n_1-i-j-k$)为反式, ($n_2-i-j-k$)为反式+120度, ($n_3-i-j-k$)=反式+240度.

5 单个四面体氢原子, 例如C₃CH

产生一个氢原子(n'), 距离原子*i* 0.1 nm, 处于四面体构型, 角($n'-i-j$), ($n'-i-k$) 和($n'-i-l$) 都为109.47度.

6 两个四面体氢原子, 如C-CH₂-C

产生两个氢原子(n_1, n_2), 距离原子*i* 0.1 nm, 处于四面体构型, 位于平分角*j-i-k*的平面上, 并且角(n_1-i-n_2), (n_1-i-j)和(n_1-i-k)都为109.47度.

7 两个水中的氢原子

根据SPC[81]水模型的几何构型在原子*i*周围产生两个氢原子. 对称轴交替位于三个坐标轴两个方向之间.

10 三个水中的“氢原子”

根据SPC[81]水模型的几何构型在原子*i*周围产生两个氢原子. 对称轴交替位于三个坐标轴两个方向之间. 此外, 在氧原子位置产生一个额外的粒子, 并将其名称的第一个字母以'M'替代. 此方法用于四位点的水模型, 如TIP4P[125].

11 四个水中的“氢原子”

同上, 但会在氧原子位置产生两个额外的粒子, 名字分别为'LP1'和'LP2'. 此方法用于五位点的水模型, 如TIP5P[127].

- 新的氢原子的名称(或其前缀, 如HD2, 对前面天冬酰胺的例子).
- 三或四个控制原子(i, j, k, l), 其中的第一个始终为与氢原子相连的原子, 另外两个或三个取决于所选的代码. 对于水, 只有一个控制原子.

对一些非常特殊的情况, 可以利用上面的方法近似地解决, 并进行适当的能量最小化, 这样得到的构型用于MD模拟的初始构型已经足够好了. 例如对仲胺氢, 亚硝酰基氢(C=NH)甚至乙炔氢都可利用上面方法2羟基氢进行近似的添加.

5.6.5 末端数据库

末端数据库储存在[aminoacids.n.tdb](#)和[aminoacids.c.tdb](#)文件中, 分别对应于N端和C端. 文件包含了关于[pdb2gmx](#)程序如何将原子连接到已有原子, 应该删除或更改哪些原子, 应该添加哪些键相互作用的信息. 文件的格式如下(来源于[gromos43a1.ff/aminoacids.c.tdb](#)):

```
[ None ]
[ COO- ]
[ replace ]
C      C      C    12.011     0.27
O      O1     OM   15.9994   -0.635
OXT    O2     OM   15.9994   -0.635
[ add ]
2     8     0     C     CA     N
OM    15.9994  -0.635
[ bonds ]
C  O1  gb_5
C  O2  gb_5
```

```
[ angles ]
O1 C 02 ga_37
CA C 01 ga_21
CA C 02 ga_21
[ dihedrals ]
N CA C 02 gd_20
[ impropers ]
C CA 02 01 gi_1
```

该文件以块为组织, 每块的标题指定了块的名称. 这些块对应于可添加到分子的不同类型的末端. 在本例中 [coo-] 为第一块, 对应于将末端碳原子更改为去质子化的羧基. [None] 为第二末端类型, 对应于维持分子原样的末端. 块名称不能取以下的任何一种: replace, add, delete, bonds, angles, dihedrals, impropers. 否则会干扰块的参数, 并可能令读者十分迷惑.

每个块可使用以下选项:

- [replace]

将一个已有原子替换为具有不同原子类型, 原子名称, 电荷和/或质量的原子. 此项可用于替换输入坐标文件和 .rtp 数据库中的原子, 但也可用于仅仅重命名输入坐标中的原子, 以便它匹配力场中的名称. 对后一种情况, 也应该存在相应的 [add] 段, 用于指示如何添加相同的原子, 这样才能知道序列中的位置和成键. 这种原子可出现在输入坐标中并保持, 或不出现在输入坐标中而是通过 pdb2gmx 程序构建. 对于每一个要替换的原子, 应该输入具有以下域的一行:

- 要替换原子的名称
- 新的原子名称(可选)
- 新的原子类型
- 新的质量
- 新的电荷

- [add]

添加新的原子. 对于每一个(组)要添加的原子需要两行输入. 第一行包含与氢数据库相同的域(新原子的名称, 原子数目, 添加类型, 控制原子, 参见5.6.4节), 但增加了两个只用于C端的添加类型

8 双羧基氧原子, -COO⁻

根据规则3产生两个氧原子(n1,n2), 距离原子i 0.136 nm, 角(n1-i-j)和(n2-i-j)为117度

9 羧基氧原子和氢原子, -COOH

根据规则3产生两个氧原子(n1,n2), n1 和n2分别距离原子i 0.123 nm 和0.125 nm, 角(n1-i-j)为121度, (n2-i-j)为115度. 根据规则2在n2周围产生一个氢原子(n'), 其中n'-n2-i 和n'-n2-i-j 应分别对应于n-i-j 和n-i-j-k.

此行之后, 下面的一行指定了添加原子的细节, 与替换原子的方式相同, 即:

- 原子类型
- 质量
- 电荷
- 电荷组(可选)

如氢数据库(参见5.6.1节)一样, 当一个以上的原子连接到一个已有原子时, 原子名称的末尾会追加一个数字. 注意, 像在氢数据库中一样, 原子名称现在与控制原子处于同一行, 而在GROMACS 3.3版本之前它位于第二行的开始. 当电荷组域空白时, 添加的原子和它连接的原子将具有相同的电荷组编号.

- **[delete]**

删除已有的原子. 每个原子名称一行.

- **[bonds]**, **[angles]**, **[dihedrals]** 和 **[impropers]**

添加额外的键参数. 格式与 ***.rtp** 文件中使用的相同, 参见5.6.1节.

5.6.6 虚拟位点数据库

由于不能依赖输入文件中氢的位置, 我们需要一个特殊的输入文件以确定要添加的虚拟氢位点的几何构型和参数. 为创建更复杂的虚拟位点(例如当保持整个芳香侧链刚性时), 我们还需要关于平衡键长和侧链中所有原子角度的信息. 对每个力场, 这些信息在 **.vsd** 文件中指定. 与末端类似, 对 **.rtp** 文件中每个残基类型都有一个这样的文件.

虚拟位点数据库真的不是一个非常简单的信息列表. 它开始的两段指定了用于CH₃, NH₃和NH₂基团的质量中心(通常称为MCH₃/MNH₃). 根据氢原子和重原子之间的平衡键长和键角, 我们需要在这些质量中心之间施加略有不同的约束距离. 注意, 我们不需要指定实际的参数(会自动生成), 而只需要指定使用的质量中心类型. 为此, 有三个段名称 **[CH3]**, **[NH3]** 和 **[NH2]**. 对每个段我们需要三列. 第一列为连接到2/3氢原子的原子类型, 第二列为连接的下一个重原子的类型, 第三列为使用的质量中心类型. 作为一种特殊情况, 在 **[NH2]** 段的第二列也可以指定为 **planar**, 代表将使用不同的构造方法, 而不使用质量中心. 对于NH₂基团是否应为平面, 目前的一些力场观点不一, 但我们力图保持力场默认的平衡参数.

虚拟位点数据库的第二部分明确地包含芳香族侧链的原子对/三联组的平衡键长和键角. 目前, 虚拟位点生成代码的特定例程会读入这些项, 因此如果你想扩展它, 如扩展到核酸, 你需要编写新的代码. 这些段根据氨基酸的短名称进行命名(**[PHE]**, **[TYR]**, **[TRP]**, **[HID]**, **[HIE]**, **[HIP]**), 简单地包含2或3列原子名称, 接着是指定键长(以nm为单位)或键角(以度为单位)的数字. 注意, 对整个分子的平衡几何构型有所近似, 如果分子未处于自然状态, 其单个键长/键角的值可能与平衡值不同.

5.6.7 特殊键

pdb2gmx 程序生成残基间化学键时使用的主要机制为, 从头到尾连接不同残基中的骨架原子进而形成大分子. 在某些情况下(如二硫键, 血红素基团, 支化聚合物), 有必要创建非骨架残基间的化学键. **specbond.dat** 文件即用于此功能. 残基必须属于相同的 **[moleculetype]**. 当操纵不同链之间的特殊残基间的化学键时, **pdb2gmx** 程序的 **-merge**

和 `-chainsep` 功能很有用.

`specbond.dat` 文件的第一行表示文件中的条目数. 如果你添加了一个新的条目, 请确保增加此数字. 文件的其余行提供创建键的具体说明. 行的格式如下:

```
resA atomA nbondsA resB atomB nbondsB length newresA newresB
```

分别代表:

1. `resA` 参与成键的残基A的名称.
2. `atomA` 残基A中形成键的原子的名称.
3. `nbondsA` `atomA`可以形成的键的总数.
4. `resB` 参与成键的残基B的名称.
5. `atomB` 残基B中形成键的原子的名称.
6. `nbondsB` `atomB`可以形成的键的总数.
7. `length` 键的参考长度. 在提供给 `pdb2gmx` 程序的坐标文件中, 若 `atomA` 和 `atomB` 之间的距离不在 `length` $\pm 10\%$ 范围内, 它们之间不会形成键.
8. `newresA` 残基A的新名称, 如果必要. 有些力场对与二硫键或血红素相连的半胱氨酸使用了CYS2之类的名称.
9. `newresB` 残基B的新名称, 同上.

5.7 文件格式

5.7.1 拓扑文件

拓扑文件是根据GROMACS对分子拓扑的具体说明建立的. 可利用 `pdb2gmx` 程序生成 `*.top` 文件. 拓扑文件中所有可能的输入项都列于表 5.4 和表 5.5 中. 表中还列有: 所有参数的单位, 哪些相互作用可被微扰以用于自由能计算, `grompp` 可使用哪些键合相互作用生成排除, `grompp` 可将哪些键合相互作用转化为约束.

表5.4: 拓扑(`*.top`)文件

参数					
相互作用类型	指令	原子数	函数类型	参数	自由能
必需	<code>defaults</code>			非键函数类型; 组合规则 <code>cr</code> ; 生成对(<code>no/yes</code>); <code>fudge LJ()</code> ; <code>fudge QQ()</code>	
必需	<code>atomtypes</code>			原子类型; 质量 <code>m(u)</code> ; 电荷 <code>q(e)</code> ; 粒子类型; <code>V^{cr}</code> <code>W^{cr}</code>	
	<code>bondtypes</code>			(参考表 5.5 <code>bonds</code> 指令)	
	<code>pairtypes</code>			(参考表 5.5 <code>pairs</code> 指令)	
	<code>angletypes</code>			(参考表 5.5 <code>angles</code> 指令)	
	<code>dihedraltypes</code> *			(参考表 5.5 <code>dihedrals</code> 指令)	

	constrainttypes	(参考表 5.5 constraints 指令)				
LJ	nonbond_params	2	1	$V^{cr}; W^{cr}$		
Buckingham	nonbond_params	2	2	$a(kJ mol^{-1}); b(nm^{-1}); c6(kJ mol^{-1} nm^6)$		
分子定义						
相互作用类型	指令	原子数	函数类型	参数	自由能	
必需	moleculatype			分子名称; n_{ex}^{nrexcl}		
必需	atoms	1		原子类型; 残基编号; 残基名称; 原子名称; 电荷组 编号; 电荷q(e); 质量m(u)	类型, q, m	
分子内相互作用和几何构型定义见表 5.5						
系统						
相互作用类型	指令	原子数	函数类型	参数	自由能	
必需	system			系统名称		
必需	molecules			分子名称; 分子数		

- 原子数: 对每个指令需要的原子类型的数目
- 函数类型: 要选择的函数类型
- 自由能: 表示自由能计算时这种相互作用的哪些参数可以进行插值
- $^{cr^*}$: 组合规则确定LJ参数的类型, 参见5.3.2节
- ** : 对dihedraltypes可指定4个原子或两个内原子(对异常二面角则设定两个外原子)
- $^{nrexcl^*}$: 对非键相互作用, 排除相邻的 n_{ex} 个键
- 对于自由能计算, 在正常参数之后, 类型, 电荷q和质量m或无参数添加到同一行上的拓扑'B'($\lambda=1$).

表 5.5: [moleculatype] 指令详解

相互作用名称	拓扑文件指令	原子数*	函数类型†	参数顺序及其单位	用于自由能计算?‡	交叉参考(节)
键	bonds §,¶	2	1	$b_0(nm); k_b(kJ mol^{-1} nm^{-2})$	所有	4.2.1
G96键	bonds §,¶	2	2	$b_0(nm); k_b(kJ mol^{-1} nm^{-4})$	所有	4.2.1
Morse键	bonds §,¶	2	3	$b_0(nm); D(kJ mol^{-1}); \beta(nm^{-1})$	所有	4.2.2
立方键	bonds §,¶	2	4	$b_0(nm); C_{i=2,3}(kJ mol^{-1} nm^{-i})$		4.2.3
连接	bonds §	2	5			5.4
简谐势	bonds	2	6	$b_0(nm); k_b(kJ mol^{-1} nm^{-2})$	所有	4.2.1, 5.4
FENE键	bonds §	2	7	$b_m(nm); k_b(kJ mol^{-1} nm^{-2})$		4.2.4
表格键	bonds §	2	8	表号(≥ 0); $k(kJ mol^{-1})$	k	4.2.14

表格键	bonds	2	9	表号(≥ 0); $k(\text{kJ mol}^{-1})$	k	4.2.14, 5.4
约束势	bonds	2	10	下限,上限1,上限2(nm); $k_{dr}(\text{kJ mol}^{-1} \text{nm}^{-2})$	所有	4.3.5
额外LJ或库仑	pairs	2	1	V^{**} ; W^{**}	所有	5.3.4
额外LJ或库仑	pairs	2	2	fudge $QQ(\cdot)$; $q_i, q_j(e)$; $V^{**}; W^{**}$		5.3.4
额外LJ或库仑	pairs_nb	2	1	$q_i, q_j(e)$; V^{**} ; W^{**}		5.3.4
键角	angles [¶]	3	1	$\theta_0(\text{deg})$; $k_\theta(\text{kJ mol}^{-1} \text{rad}^{-2})$	所有	4.2.5
G96键角	angles [¶]	3	2	$\theta_0(\text{deg})$; $k_\theta(\text{kJ mol}^{-1})$	所有	4.2.6
键-键交叉项	angles	3	3	$r_{1e}, r_{2e}(\text{nm})$; $k_{rr'}(\text{kJ mol}^{-1} \text{nm}^{-2})$		4.2.9
键-角交叉项	angles	3	4	$r_{1e}, r_{2e}, r_{3e}(\text{nm})$; $k_{r\theta}(\text{kJ mol}^{-1} \text{nm}^{-2})$		4.2.10
Urey-Bradley键角	angles [¶]	3	5	$\theta_0(\text{deg})$; $k_\theta(\text{kJ mol}^{-1} \text{rad}^{-2})$; $r_{13}(\text{nm})$; $k_{UB}(\text{kJ mol}^{-1} \text{nm}^{-2})$	所有	4.2.8
四次键角	angles [¶]	3	6	$\theta_0(\text{deg})$; $C_{i=0,1,2,3,4}(\text{kJ mol}^{-1} \text{rad}^{-4})$		4.2.11
表格角	angles	3	8	表号(≥ 0); $k(\text{kJ mol}^{-1})$	k	4.2.14
限制弯曲势	angles	3	10	$\theta_0(\text{deg})$; $k_\theta(\text{kJ mol}^{-1})$		4.2.7
正常二面角	dihedrals	4	1	$\phi_s(\text{deg})$; $k_\phi(\text{kJ mol}^{-1})$; 多重数	ϕ, k	4.2.13
异常二面角	dihedrals	4	2	$\xi_0(\text{deg})$; $k_\xi(\text{kJ mol}^{-1} \text{rad}^{-2})$	所有	4.2.12
Ryckaert-Belleman二面角	dihedrals	4	3	$C_0, C_1, C_2, C_3, C_4, C_5(\text{kJ mol}^{-1})$	所有	4.2.13
周期异常二面角	dihedrals	4	4	$\phi_s(\text{deg})$; $k_\phi(\text{kJ mol}^{-1})$; 多重数	ϕ, k	4.2.12
傅立叶二面角	dihedrals	4	5	$C_0, C_1, C_2, C_3, C_4(\text{kJ mol}^{-1})$	所有	4.2.13
表格二面角	dihedrals	4	8	表号(≥ 0); $k(\text{kJ mol}^{-1})$	k	4.2.14
正常二面角(多重)	dihedrals	4	9	$\phi_s(\text{deg})$; $k_\phi(\text{kJ mol}^{-1})$; 多重数	ϕ, k	4.2.13
限制二面角	dihedrals	4	11	$\phi_0(\text{deg})$; $k_\phi(\text{kJ mol}^{-1})$		4.2.13
结合弯区扭转势	dihedrals	4	10	$a_0, a_1, a_2, a_3, a_4(\text{kJ mol}^{-1})$		4.2.13
排除	exclusions	1		一个或多个原子索引号		5.4
约束	constraints [§]	2	1	$b_0(\text{nm})$	所有	4.5, 5.5
约束	constraints	2	2	$b_0(\text{nm})$	所有	4.5, 5.5, 5.4
SETTLE	settles	1	1	$d_{\text{oh}}, d_{\text{HH}}(\text{nm})$		3.6.1, 5.5
二体虚拟站点	virtual_sites2	3	1	$a()$		4.7
三体虚拟站点	virtual_sites3	4	1	$a, b()$		4.7
三体虚拟站点		4	2	$a()$; $d(\text{nm})$		4.7

(fd)	<code>virtual_sites3</code>					
三体虚拟站点 (fad)	<code>virtual_sites3</code>	4	3	$\theta(\text{deg})$; $d(\text{nm})$		4.7
三体虚拟站点 (out)	<code>virtual_sites3</code>	4	4	$a, b()$; $c(\text{nm}^{-1})$		4.7
四体虚拟站点 (fdn)	<code>virtual_sites4</code>	5	2	$a, b()$; $c(\text{nm})$		4.7
N体虚拟站点 (COG)	<code>virtual_sitesn</code>	1	1	一个或多个构建原子索引号		4.7
N体虚拟站点 (COM)	<code>virtual_sitesn</code>	1	2	一个或多个构建原子索引号		4.7
N体虚拟站点 (COW)	<code>virtual_sitesn</code>	1	3	一对或多对构建原子的索引号与权重		4.7
位置约束	<code>position_restraints</code>	1	1	$k_x, k_y, k_z (\text{kJ mol}^{-1} \text{nm}^{-2})$	所有	4.3.1
平底位置约束	<code>position_restraints</code>	1	2	$g, r(\text{nm}), k(\text{kJ mol}^{-1} \text{nm}^{-2})$		4.3.2
距离约束	<code>distance_restraints</code>	2	1	类型; 标签; 下限, 上限1, 上限2(nm); 权重()		4.3.5
二面角约束	<code>dihedral_restraints</code>	4	1	$\phi_0(\text{deg})$; $\Delta\phi(\text{deg})$	所有	4.3.4
方向约束	<code>orientation_restraints</code>	2	1	$\exp.$; 标签; α ; $c(U \text{ nm}^\alpha)$; $\text{obs.}(U)$; 权重(U^{-1})		4.3.6
角度约束	<code>angle_restraints</code>	4	1	$\theta_0(\text{deg})$; $k_c(\text{kJ mol}^{-1})$; 多重数	θ, k	4.3.3
角度约束(z)	<code>angle_restraints_z</code>	2	1	$\theta_0(\text{deg})$; $k_c(\text{kJ mol}^{-1})$; 多重数	θ, k	4.3.3

- *: 指令所需的原子数目
- †: 用于选择的函数类型编号
- ‡: 表示自由能计算时这种相互作用的哪些参数可以进行插值
- §: `grompp`会利用这种相互作用类型产生排除
- ¶: `grompp`会将这种相互作用类型转换为约束
- **: 组合规则确定LJ参数的类型, 参见5.3.2节
- ||: 无连接, 因此不会为这种相互作用产生排除

文件布局说明:

- 分号(;)和换行符之间为注释
- 忽略\后的换行符, 即\为续行符
- [和] 之间为指令
- 拓扑等级(必须遵循)包含三个层次:
 - 参数级别, 定义了一些力场的设定(见表 5.4)

- 分子级别, 应包含一个或多个分子的定义(见表 5.5)
- 体系级别, 只包含体系的特定信息([**system**]及[**molecules**])
- 每一项应用空格或制表符分格, 而不是逗号
- 分子中的原子应从1开始连续编号
- 相同电荷组中的原子必须连续列出
- 文件只解析一次, 这意味着不会处理向前引用: 每一项在使用前必须进行定义
- 可以从键生成排除, 或手动覆盖
- 可以从原子类型生成键合力的类型, 或对每条键覆盖
- 在同一个原子上可以使用多个相同类型的键合相互作用
- 强烈推荐使用描述性的注释行和空行
- 自GROMACS 3.1.3版本开始, 参数级别的所有指令可使用多次, 并且没有顺序限制, 除了在其它参数定义中, 原子类型在使用前需要进行定义
- 对相同的原子类型组合, 如果某个相互作用的参数被定义了多次, 会使用最后定义的参数; 自GROMACS 3.1.3版本开始, 参数若存在不同值的重定义, **grompp**会生成警告
- 在[**moleculetype**]之前使用[**atoms**], [**bonds**], [**pairs**], [**angles**]等指令毫无意义, 会导致警告
- 在[**system**]之前使用[**molecules**]指令毫无意义, 会导致警告
- 在[**system**]之后唯一可以使用的指令是[**molecules**]
- 在[]中使用未知的字符串会导致直到下一个指令的所有数据被忽略, 并产生警告

下面是拓扑文件的一个例子, **urea.top**:

```

;
;           Example topology file
;
; The force-field files to be included
#include "amber99.ff/forcefield.itp"

[ moleculetype ]
; name nrexcl
Urea      3

[ atoms ]
  1  C   1  URE    C    1      0.880229    12.01000    ; amber C type
  2  O   1  URE    O    2     -0.613359   16.00000    ; amber O type

```

```

3 N 1 URE N1 3 -0.923545 14.01000 ; amber N type
4 H 1 URE H11 4 0.395055 1.00800 ; amber H type
5 H 1 URE H12 5 0.395055 1.00800 ; amber H type
6 N 1 URE N2 6 -0.923545 14.01000 ; amber N type
7 H 1 URE H21 7 0.395055 1.00800 ; amber H type
8 H 1 URE H22 8 0.395055 1.00800 ; amber H type

[ bonds ]
 1 2
 1 3
 1 6
 3 4
 3 5
 6 7
 6 8

[ dihedrals ]
; ai aj ak al funct definition
 2 1 3 4 9
 2 1 3 5 9
 2 1 6 7 9
 2 1 6 8 9
 3 1 6 7 9
 3 1 6 8 9
 6 1 3 4 9
 6 1 3 5 9

[ dihedrals ]
 3 6 1 2 4
 1 4 3 5 4
 1 7 6 8 4

[ position_restraints ]
; you wouldn't normally use this for a molecule like Urea,
; but we include it here for didactic purposes
; ai funct fc
 1 1 1000 1000 1000 ; Restrain to a point
 2 1 1000 0 1000 ; Restrain to a line (Y-axis)
 3 1 1000 0 0 ; Restrain to a plane (Y-Z-plane)

[ dihedral_restraints ]
; ai aj ak al type label phi dphi kfac power
 3 6 1 2 1 1 180 0 1 2
 1 4 3 5 1 1 180 0 1 2

; Include TIP3P water topology
#include "amber99/tip3p.itp"

[ system ]
Urea in Water

[ molecules ]
;molecule name nr.
Urea 1
SOL 1000

```

下面是此文件的解释说明.

`#include "amber99.ff/forcefield.itp"`: 包含你正使用的力场的信息, 包括键和非键参数. 本例使用了 AMBER99 力场, 但你进行模拟时可使用不同的力场. `grompp` 会自动找到这个文件并复制粘贴它的内容. 你可以在 `share/top/amber99.ff/forcefield.itp` 文件中看到它的内容

```

#define _FF_AMBER
#define _FF_AMBER99

```

```
[ defaults ]
; nbfunc      comb-rule     gen-pairs      fudgeLJ      fudgeQQ
1             2                 yes            0.5           0.8333

#include "ffnonbonded.itp"
#include "ffbonded.itp"
#include "gbsa.itp"
```

两个`#define`语句设置了条件, 这样拓扑文件后面的部分可以知道使用了AMBER 99力场.

[defaults]:

- `nbfunc`为非键函数类型, 取1(Lennard-Jones) 或2(Buckingham)
- `comb-rule`为组合规则的编号(参见5.3.2节).
- `gen-pairs`用于对的生成. 默认为'no', 即得到对类型列表的1–4的参数. 当列表中不存在参数时, 会导致致命错误并停止运行. 设置为'yes' 会根据使用`fudgeLJ`的正常Lennard-Jones 参数, 生成不存在于配对列表中的1–4参数
- `fudgeLJ`为Lennard-Jones 1–4 相互作用的修正因子, 默认为1
- `fudgeQQ`为静电1–4相互作用的修正因子, 默认为1
- N为6-N势中排斥项的次数(只用于非键Lennard-Jones), 自GROMACS 4.5版本开始, `mdrun`会读取和使用N, 当N不等于12时, 使用表格相互作用函数(在旧版本中, 你必须使用用户自定义的表格相互作用).

注意, `gen-pairs`, `fudgeLJ`, `fudgeQQ`和N是可选的. 仅当`gen-pairs`被设置为'yes' 时才会使用`fudgeLJ`, 而总会使用`fudgeQQ`. 然而, 如果你想指定N, 你也需要给出其他参数的值.

后面的其他一些`#include`语句用于添加描述力场其余部分所需的大量数据. 我们将跳过这些并回到`urea.top`文件. 在那里, 我们会看到

[moleculetype]: 定义此`*.top`文件中分子的名称, `nrexcl=3`代表排除不远于三条键的原子之间的非键相互作用.

[atoms]: 定义分子, 其中`nr`和`type`是固定的, 其余的由用户自定义. 因此`atom`可以随意命名, `cgnr`可大可小(如果可能, 一个电荷组的总电荷应为零), 这里的电荷也可以更改.

[bonds]: 不注释.

[pairs]: LJ和库仑的1–4相互作用

[angles]: 不注释.

[dihedrals]: 在此情况下, 有9个正常二面角(`funct=1`), 3 个异常二面角(`funct=4`), 没有Ryckaert-Bellemans 型的二面角. 如果你想在拓扑文件中包含Ryckaert-Bellemans 型的二面角, 请遵照下列的示例(例如, 对癸烷):

[dihedrals] ; ai aj ak al funct c0 c1 c2 1 2 3 4 3 2 3 4 5 3

在烷烃势能的原始实现中[128], 没有使用1–4相互作用, 这意味着为了使用这个特定的力场, 你需要从拓扑文件的 [**pairs**] 段中移除1–4相互作用. 对大多数的现代力场, 如OPLS/AA或Amber, 其规则是不同的, Ryckaert-Bellemans势是作为余弦序列与1–4相互作用一起使用的.

[**position_restraints**]: 利用简谐势将选中的粒子约束在参考位置(参见4.3.1节). **grompp**会从由单独的坐标文件中读取参考位置.

[**dihedral_restraints**]: 将选定的二面角约束在参考值. 二面角约束的实现方法参见本手册4.3.4节的论述. [**dihedral_restraints**]指令中设定的参数如下:

- **type** 只有一个可能的值, 1
- **label** 未使用, 并被从代码删除.
- **phi** 为本手册方程4.84和4.85中的 ϕ_0 值.
- **dphi** 为本手册方程4.85中的 $\Delta\phi$ 值.
- **kfac** 类似于距离约束中的 **fac**, 是力常数的修正因子. 通过它, 可以利用不同的力常数维持不同的约束.
- **power** 未使用, 并被从代码删除.

#include "tip3p.itp": 包括一个已经构建好的拓扑文件(参见5.7.2节).

[**system**]: 体系的标题, 用户自定义

[**molecules**]: 定义体系中分子(亚分子)的总数, 这些分子已经在此 ***.top** 中进行了定义. 在这个示例文件中, 它代表1个尿素分子溶于1000个水分子中. 分子类型SOL是在 **tip3p.itp** 文件中定义的. 这里的每个名称必须对应于拓扑文件前面的 [**moleculename**] 中给出的名称. 分子类型块的顺序, 这些分子的数目必须与坐标文件匹配, 坐标文件和拓扑文件会一起提供给 **grompp**. 分子块不需要连续, 但某些工具(如 **genion**)可能只对一个特定分子类型的第一或最后一块起作用. 另外, 这些块与组的定义无关(参见3.3节和8.1节).

5.7.2 Molecule.itp文件

如果你构建了一个拓扑文件并且经常使用(像水分子, 已经构建好的 **tip3p.itp**), 将它做成 **molecule.itp** 文件更好一些. 此文件中只列出了一个特定分子的信息, 你可以在多个体系中重用 [**moleculename**], 而无需重新调用 **pdb2gmx** 或手动复制粘贴. 例子 **urea.itp** 如下:

```
[ moleculename ]
; molname nrexcl
URE 3

[ atoms ]
  1  C  1  URE    C    1  0.880229  12.01000 ; amber C type
...
  8  H  1  URE   H22    8  0.395055  1.00800 ; amber H type
```

```
[ bonds ]
  1 2
...
  6 8

[ dihedrals ]
; ai aj ak al funct definition
  2   1   3   4    9
...
  6   1   3   5    9

[ dihedrals ]
  3   6   1   2    4
  1   4   3   5    4
  1   7   6   8    4
```

使用`*.itp`文件会使`*.top`文件明显变短:

```
;           Example topology file
;
; The force field files to be included
#include "amber99.ff/forcefield.itp"

#include "urea.itp"

; Include TIP3P water topology
#include "amber99/tip3p.itp"

[ system ]
Urea in Water

[ molecules ]
;molecule name      nr.
Urea                  1
SOL                   1000
```

5.7.3 `#ifdef`语句

GROMACS有一个非常强大的功能, 你可以在`*.top`文件中使用`#ifdef`语句. 通过使用这条语句以及相关的`#define`语句, 像在前面的`amber99.ff/forcefield.itp`文件中那样, 在同一个`*.top`文件中可以为分子使用不同的参数. 下面给出TFE的一个例子, 其中的一个选项对原子使用了不同的电荷: De Loof 等人给出的电荷[129]或Van Buuren 和Berendsen 给出的电荷[130]. 实际上, 你可以使用C预处理器`cpp`的许多功能, 因为`grompp`包含了类似的预处理函数以扫描文件. 使用`#ifdef`选项的方式如下:

- 在`*.mdp`文件(包含`grompp`输入参数)中使用选项`define = -DDeLoof`, 或在`*.top`中早些使用`#define DeLoof`, 并且
- 将`#ifdef`语句放在`*.top`中, 如所示:

```
...
[ atoms ]
; nr type resnr residu atom cgnr charge mass
#ifndef DeLoof
; Use Charges from DeLoof
```

```

1      C      1      TFE      C      1      0.74
2      F      1      TFE      F      1      -0.25
3      F      1      TFE      F      1      -0.25
4      F      1      TFE      F      1      -0.25
5      CH2     1      TFE      CH2     1      0.25
6      OA      1      TFE      OA      1      -0.65
7      HO      1      TFE      HO      1      0.41
#else
; Use Charges from VanBuuren
1      C      1      TFE      C      1      0.59
2      F      1      TFE      F      1      -0.2
3      F      1      TFE      F      1      -0.2
4      F      1      TFE      F      1      -0.2
5      CH2     1      TFE      CH2     1      0.26
6      OA      1      TFE      OA      1      -0.55
7      HO      1      TFE      HO      1      0.3
#endif

[ bonds ]
; ai    aj   funct        c0          c1
6      7      1 1.000000e-01 3.138000e+05
1      2      1 1.360000e-01 4.184000e+05
1      3      1 1.360000e-01 4.184000e+05
1      4      1 1.360000e-01 4.184000e+05
1      5      1 1.530000e-01 3.347000e+05
5      6      1 1.430000e-01 3.347000e+05
...

```

`pdb2gmx` 使用这个机制来实现可选的位置约束(4.3.1节), 具体是通过使用`#include`包含`.itp`文件, 而此文件的只有设置了特定的`#define`(并且拼写正确!)时才有意义.

5.7.4 用于自由能计算的拓扑文件

两个体系A和B之间的自由能差值, 其计算方法见3.12节的论述. 描述体系A和B的拓扑具有相同数目的分子, 分子的原子数也相同. 通过在`[atoms]`指令下添加B参数, 可以对质量和非键相互作用进行微扰. 键合相互作用的微扰可以通过添加B参数到键合类型或键合相互作用来完成. 能够进行微扰的参数列于表 5.4和表 5.5. 相互作用的 λ 依赖性见4.5节的论述. 使用的键参数(位于定义键相互作用的行, 或通过对键合类型列表的原子类型进行查找得到)在表 5.6中进行了解释. 大多数情况下, 处理方式都很直观. 当键合相互作用中A和B的原子类型不完全相同, 且B状态的参数不存在时, 无论是位于行还是键合类型, `grompp`都会使用A状态的参数, 并发出警告. 对自由能计算, 拓扑B($\lambda = 1$)的所有参数或者添加在同一行, 或者不添加, 添加时位于正常参数之后, 顺序与正常参数相同. 自GROMACS 4.6版本起, 如果 λ 被视为矢量, 那么`bonded-lambdas`分量控制所有未明确标记为约束的键合项. 约束项通过`restraint-lambdas`分量控制.

表 5.6: 用于自由能拓扑的键合参数, 位于定义键合相互作用的行, 或基于原子类型在键类型中查找. A和B表示分别用于状态A和B的参数, +和-表示拓扑中(不)存在的参数, x表示存在没有影响.

B状态与A状态的原子类型完全相同?	行参数 A B	A原子类型的键合类型参数 A B	B原子类型的键合类型参数 A B	信息
yes	+AB -	x x		错误
	+A +B	x x		
	- -	- -		
	- -	+AB -		
no	+AB -	x x	x x	警告
	+A +B	x x	x x	错误 警告
	- -	- -	x x	
	- -	+AB -	- -	警告 警告
	- -	+A +B	- -	
	- -	+A x	+B -	

- -

+A x

+ +B

下面是一个拓扑文件的例子, 将200个丙醇分子转变为200个戊烷分子, 使用了GROMOS-96力场.

```
; Include force field parameters
#include "gromos43a1.ff/forcefield.itp"
[ moleculetype ]
; Name      nrexcl
PropPent    3

[ atoms ]
; nr      type   resnr   residue  atom   cgnr   charge   mass     typeB   chargeB   massB
  1       H        1        PROP     PH      1      0.398    1.008    CH3      0.0      15.035
  2       OA       1        PROP     PO      1     -0.548   15.9994   CH2      0.0      14.027
  3       CH2      1        PROP     PC1     1      0.150    14.027   CH2      0.0      14.027
  4       CH2      1        PROP     PC2     2      0.000    14.027
  5       CH3      1        PROP     PC3     2      0.000    15.035

[ bonds ]
; ai      aj      funct   par_A    par_B
  1      2          2      gb_1     gb_26
  2      3          2      gb_17    gb_26
  3      4          2      gb_26    gb_26
  4      5          2      gb_26    gb_26

[ pairs ]
; ai      aj      funct
  1      4          1
  2      5          1

[ angles ]
; ai      aj      ak      funct   par_A    par_B
  1      2      3          2      ga_11    ga_14
  2      3      4          2      ga_14    ga_14
  3      4      5          2      ga_14    ga_14

[ dihedrals ]
; ai      aj      ak      al      funct   par_A    par_B
  1      2      3      4          1      gd_12    gd_17
  2      3      4      5          1      gd_17    gd_17

[ system ]
; Name
Propanol to Pentane

[ molecules ]
; Compound #mols
PropPent   200
```

未微扰的 **PC2** 和 **PC3** 原子不需要指定B状态参数, 因为所用B参数将从A参数复制得来. 原子间不微扰的键合相互作用不需要指定B参数, 如示例拓扑中的最后一条键. 使用OPLS/AA力场的拓扑根本不需要键合参数, 因为A和B参数都由原子类型决定. 涉及一个或两个微扰原子的非键相互作用将使用自由能微扰函数形式, 两个非微扰原子间的非键相互作用使用正常的函数形式. 这意味着, 例如, 当只对粒子的电荷进行微扰时, 若lambda 不等于0或1, 其Lennard-Jones 相互作用也会受到影响.

注意, 此拓扑文件使用了GROMOS-96力场, 其中的键合相互作用不是由原子类型决定的. 键合相互作用的字符串由C预处理器转换而来. 力场参数文件包含这样的行:

```
#define gb_26 0.1530 7.1500e+06
#define gd_17 0.000 5.86 3
```

5.7.5 约束力

一个分子中两个原子之间的约束力, 可以使用自由能微扰代码进行计算, 计算时在两个原子之间添加约束, 约束对拓扑A和B具有不同长度. 当B的长度比A大1 nm时, lambda 保持为零, 哈密顿量对lambda 的导数就是约束力. 对分子之间的约束, 可以使用拉扯代码, 参见6.4节. 下面是一个具体的例子, 通过组合两个甲烷成一个“分子”, 计算水中的两个甲烷分子距离0.7 nm时的约束力. 注意, GROMACS中“分子”的定义不一定对应于分子的化学定义. 在GROMACS中, 一个“分子”可被定义为希望同时考虑的任意一组原子. 添加的约束函数类型为2, 意味着它不被用于生成排除(参见5.4节). 注意约束自由能项被包含在导数项中, 并且具体被包含在**bonded-lambdas**部分中. 然而, 改变约束的自由能没有被包含在用于BAR和MBAR的势能差值中, 因为这需要对每个约束分量重新计算能量. 我们计划在以后的版本中实现此功能.

```
; Include force-field parameters
#include "gromos43a1.ff/forcefield.itp"

[ moleculetype ]
; Name          nrexcl
Methanes        1

[ atoms ]
; nr   type    resnr    residu    atom   cgnr  charge    mass
  1    CH4      1         CH4       C1     1      0    16.043
  2    CH4      1         CH4       C2     2      0    16.043
[ constraints ]
; ai      aj    funct    length_A    length_B
  1      2      2           0.7          1.7

#include "gromos43a1.ff/spc.itp"
[ system ]
; Name
Methanes in Water

[ molecules ]
; Compound #mols
Methanes      1
SOL          2002
```

5.7.6 坐标文件

扩展名为**.gro**的文件包含了GROMOS-87格式的分子结构. 这种文件的一个示例片段如下:

```
MD of 2 waters, reformat step, PA aug-91
 6
 1WATER OW1   1   0.126   1.624   1.679   0.1227  -0.0580   0.0434
 1WATER HW2   2   0.190   1.661   1.747   0.8085   0.3191  -0.7791
 1WATER HW3   3   0.177   1.568   1.613  -0.9045  -2.6469   1.3180
 2WATER OW1   4   1.275   0.053   0.622   0.2519   0.3140  -0.1734
 2WATER HW2   5   1.337   0.002   0.680  -1.0641  -1.1349   0.0257
 2WATER HW3   6   1.326   0.120   0.568   1.9427  -0.8216  -0.0244
1.82060 1.82060 1.82060
```

格式是固定, 即, 所有列的位置都固定. 如果你想在自己的程序中读取这样的文件, 而不使用GROMACS库, 你可以使用以下格式:

C格式: "%5i%5s%5s%5i%8.3f%8.3f%8.4f%8.4f%8.4f"

或更准确一点, 包含标题等内容, 应该像这样:

```
%s\n", Title
"%d\n", natoms
for (i=0; (i<natoms); i++) {
    "%d-%s%5s%5d%8.3f%8.3f%8.4f%8.4f%8.4f\n",
    residuenr, residuename, atomname, atomnr, x, y, z, vx, vy, vz
}
"%10.5f%10.5f%10.5f%10.5f%10.5f%10.5f%10.5f%10.5f\n",
    box[X][X], box[Y][Y], box[Z][Z],
    box[X][Y], box[X][Z], box[Y][X], box[Y][Z], box[Z][X], box[Z][Y]
```

Fortran格式: (i5, 2a5, i5, 3f8.3, 3f8.4)

因此 `confin.gro` 是GROMACS坐标文件, 几乎与GROMOS-87文件(对GROMOS用户: 当使用 `ntx=7` 时)相同. 唯一的区别是, GROMACS对盒子使用了张量, 而不是向量.

5.8 力场的组织

5.8.1 力场文件

默认提供了许多力场, 并根据 `<name>.ff` 目录的存在进行检测, 位于 `$GMXLIB/share/gromacs/top` 的子目录和/或工作目录中. `pdb2gmx` 会打印出力场文件的位置信息, 因此你可以很容易地知道调用了哪个版本的力场, 以防你在一个或另一个位置进行了修改. GROMACS包含了以下力场:

- AMBER03蛋白质, 核酸 AMBER94(Duan et al., J. Comp. Chem. 24, 1999–2012, 2003)
.....
- AMBER94力场(Cornell et al., JACS 117, 5179–5197, 1995)
....
- AMBER96蛋白质, 核酸AMBER94(Kollman et al., Acc. Chem. Res. 29, 461–469, 1996)
.....
- AMBER99蛋白质, 核酸AMBER94((Wang et al., J. Comp. Chem. 21 , 1049–1074, 2000)
.....
- AMBER99SB 蛋白质, 核酸AMBER94(Hornak et al., Proteins 65, 712–725, 2006)
.....
- AMBER99SB-ILDN 蛋白质, 核酸AMBER94(Lindorff-Larsen et al., Proteins 78 , 1950–58, 2010)
.....
- AMBERGS力场(Garcia & Sanbonmatsu, PNAS 99, 2782–2787, 2002)
....
- CHARMM27全原子力场(CHARM22与CMAP的蛋白质力场)
- GROMOS96 43a1力场
- GROMOS96 43a2力场(改进了烷烃二面角)

- GROMOS96 45a3力场((Schuler JCC 2001 22 1205)
...)
- GROMOS96 53a5力场(JCC 2004 vol 25 pag 1656)
...)
- GROMOS96 53a6力场(JCC 2004 vol 25 pag 1656)
...)
- GROMOS96 54a7力场(Eur. Biophys. J. (2011), 40, 843–856, DOI: 10.1007/s00249–011–0700–9)
...)
- OPLS-AA/L 全原子力场(2001氨基酸二面角)

力场被包含在拓扑文件的开始, 使用`#include`语句, 后面跟着`<name>.ff/forcefield.itp`. 此语句包含力场文件, 反过来, 被包含的力场文件也可包含其它力场文件. 所有的力场以同样方式进行组织. 力场的一个例子`amber99.ff/forcefield.itp`见5.7.1节.

对每个力场, 有一些文件只被`pdb2gmx`使用. 它们是: 残基数据库(`.rtp`, 参见5.6.1节), 氢数据库(`.hdb`, 参见5.6.4节), 两个末端数据库(`.n.tdb`和`.c.tdb`, 参见5.6.5节), 只包含质量的原子类型数据库(`.atp`, 参见5.2.1节). 其他可选的文件参见5.6节的论述.

5.8.2 改变化力场参数

如果你想更改分子中少量的键相互作用的参数, 最容易的方法是, 直接在`*.top`文件`[moleculetype]`段(参见5.7.1节的格式和单位说明)的键相互作用的定义之后输入参数. 如果想更改一种相互作用所有实例中的参数, 你可以在力场文件更改, 或者在包含力场之后添加一个新的`[???types]`段. 如果一种相互作用参数被定义了多次, 会使用最后的定义. 对GROMACS 3.1.3版本, 当使用不同的值重定义参数时, 会导致警告. 建议不要更改原子类型的Lennard-Jones 参数, 因为在GROMOS力场中一些原子类型组合的Lennard-Jones 参数并不是根据标准组合规则生成的. 这些组合(以及其他可能遵循组合规则的)在`[nonbond_params]`段进行定义, 更改原子类型的Lennard-Jones 参数对这些组合没有影响.

5.8.3 添加原子类型

对GROMACS 3.1.3版本, 在包含正常力场之后, 可以使用额外的`[atomtypes]`段添加原子类型. 在新的原子类型定义之后, 可以定义另外的非键参数和对参数. 对3.1.3版本以前的GROMACS, 新的原子类型需要添加到力场文件的`[atomtypes]`段, 因为在最后的`[atomtypes]`段之上的所有非键参数都会使用标准的组合规则覆盖.

第六章 专题

6.1 自由能计算的实现

计算自由能时, 必须指定两个条件: 结束状态以及连接结束状态的途径. 指定结束状态的方式有两种, 最直接的方法是在拓扑中指定结束状态. 大多数势能函数形式既支持A状态也支持B状态. 每当指定两种状态时, A状态对应于初始自由能态, 而B状态对应于最终的状态.

在一些情况下, 也可以仅仅通过 `.mdp` 文件来定义结束状态, 而无须改变拓扑, 这可以通过使用 `couple-moltypes`, `couple-lambda0`, `couple-lambda1`, `couple-intramol` 等 `mdp` 关键词完成. `couple-moltypes` 所选定的任何分子类型会自动获得一个B状态(与一个重新定义的A状态), B状态是根据 `couple-lambda` 关键词隐式地构建的, `couple-lambda0` 和 `couple-lambda1` 定义了出现于A状态(`couple-lambda0`)和B状态(`couple-lambda1`)中的非键参数, 可用选项有'q', 'vdw' 和'vdw-q', 它们分别代表了库仑, van der Waals 或二者的参数, 这些参数在各自的状态中可以被关闭.

一旦定义了结束状态, 就必须定义结束状态之间的途径, 途径只需要在 `.mdp` 文件中定义. 从4.6版本开始, λ 是一个向量, 它的库仑, van der Waals, 键合, 约束, 质量分量都可以独立地进行调整. 这样我们在同一模拟中就可以线性地关闭库仑项, 然后是使用软核势的van der Waals 项. 对于副本交换或扩展系综模拟, 这尤其有用. 因为对这两种模拟, 为改进采样, 在同一模拟中对相互作用从有到无的所有途径进行采样非常重要.

`fep-lambdas` 是 λ 的默认数组, 其值的范围从0到1. 如果没有指定, 其他所有 λ 数组都会使用此数组中的值. 以前使用单个 λ 变量控制途径的做法, 可以通过仅使用 `fep-lambdas` 定义途径的方法.

例如, 如果你想首先改变库仑项, 然后是van der Waals 项, 同时以与van der Waals 项同样的速率改变键合项, 并在整个模拟的前三分之二改变约束项, 那么你可以使用下面的 λ 向量:

```
coul-lambdas      = 0.0  0.2  0.5  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
vdw-lambdas      = 0.0  0.0  0.0  0.0  0.4  0.5  0.6  0.7  0.8  1.0
bonded-lambdas   = 0.0  0.0  0.0  0.0  0.4  0.5  0.6  0.7  0.8  1.0
restraint-lambdas = 0.0  0.0  0.1  0.2  0.3  0.5  0.7  1.0  1.0  1.0  1.0
```

这等价于下面的设置:

```
fep-lambdas      = 0.0  0.0  0.0  0.0  0.4  0.5  0.6  0.7  0.8  1.0
coul-lambdas     = 0.0  0.2  0.5  1.0  1.0  1.0  1.0  1.0  1.0  1.0
restraint-lambdas = 0.0  0.0  0.1  0.2  0.3  0.5  0.7  1.0  1.0  1.0  1.0
```

在这种情况下, `fep-lambda` 数组作为默认值被填充到键合与van der Waals 相互作用的 λ 数组中. 通常, 最好时明确列出所有的数组, 以确保进行了正确的赋值.

如果你只想打开从A到B的约束, 那么可以使用下面的设置:

```
restraint-lambdas = 0.0 0.1 0.2 0.4 0.6 1.0
```

这样, λ 向量的其他所有分量都将保持A状态的值.

使用热力学积分计算与向量 λ 对应的自由能时, TI方程变为矢量方程:

$$\Delta F = \int \langle \nabla H \rangle \cdot d\vec{\lambda} \quad (6.1)$$

或对有限差分有:

$$\Delta F \approx \int \sum \langle \nabla H \rangle \cdot d\lambda \quad (6.2)$$

从<https://SimTK.org/home/pymbar> 下载的外部脚本 **pymbar** 可以计算根据GROMACS的dhdl.xvg 输出自动计算此积分.

6.2 平均力势

平均力势(PMF, potential of mean force) 是对构型系综的平均力进行积分得到的势能. 在GROMACS中, 有几种不同的方法可用于计算平均力. 每一种方法都有其局限性, 如下所述.

- **牵引代码:** 用于分子或分子组质心之间.
- **使用简谐键或约束的自由能代码:** 单个原子之间.
- **位置约束的自由能代码:** 改变相对不动原子组的构型.
- **有限情况的牵引代码:** 原子组之间, 它们是较大分子的一部分, 其中的键使用了SHAKE或LINCS约束. 如果牵引组较大, 会使用牵引代码.

下面两节中更详细地讨论牵引和自由能代码.

熵效应

当两个原子或两个组的质心之间的距离被约束或束缚时, 由于两个组的旋转, PMF会含有单纯由熵引起贡献[131]. 对于有两个非相互作用质量的体系, 平均力势为:

$$V_{pmf}(r) = -(n_c - 1)k_B T \log(r) \quad (6.3)$$

其中 n_c 为约束的维数(例如: 对正常的约束 $n_c = 3$, 仅在 z 方向存在约束时, $n_c = 1$). 是否需要校正这一贡献取决于你需要的PMF. 当你想要将底物牵引到蛋白质中时, 这个熵对应的项确实对要做的功有贡献. 但当计算溶剂中两个溶质之间的PMF以便用于无溶剂的模拟时, 应该移除熵的贡献. 注意, 此项可能显著; 300 K时, 若距离减半, 贡献为 3.5 kJ mol^{-1} .

6.3 非均衡牵引

当连续改变两个组之间的距离时, 会对体系做功, 这意味着体系不再处于平衡状态. 尽管在缓慢牵引的极限情况下, 体系会处于平衡状态, 但对于许多体系, 这种极限情况在可行的计算时间内部无法达到. 然而, 使用Jarzynski 关系[132], 从两个距离之间的许多非平衡模拟, 可以得到平衡的自由能变化 ΔG

$$\Delta G_{AB} = -k_B T \log \left\langle e^{-\beta W_{AB}} \right\rangle_A \quad (6.4)$$

其中 W_{AB} 为强制体系从状态A沿某路径到达状态B所需的功, 尖括号表示对初始状态A的正则系综进行平均, $\beta = 1/k_B T$.

6.4 牵引代码

牵引代码会在一对或多对原子组的质量的中心之间施加力或约束. 每个牵引的反应坐标被称为“坐标”, 它作用于两个牵引组. 一个牵引组可以是一个或多个牵引坐标的一部分. 此外, 坐标还可以作用于单个组或空间中的绝对参考位置. 两个组之间的距离可根据1, 2或3个维度来计算, 或者沿着用户定义的向量进行计算. 参考距离可以是恒定的, 或者可以随时间线性变化. 正常情况下, 所有的原子都根据质量进行加权, 但也可以使用额外的加权因子.

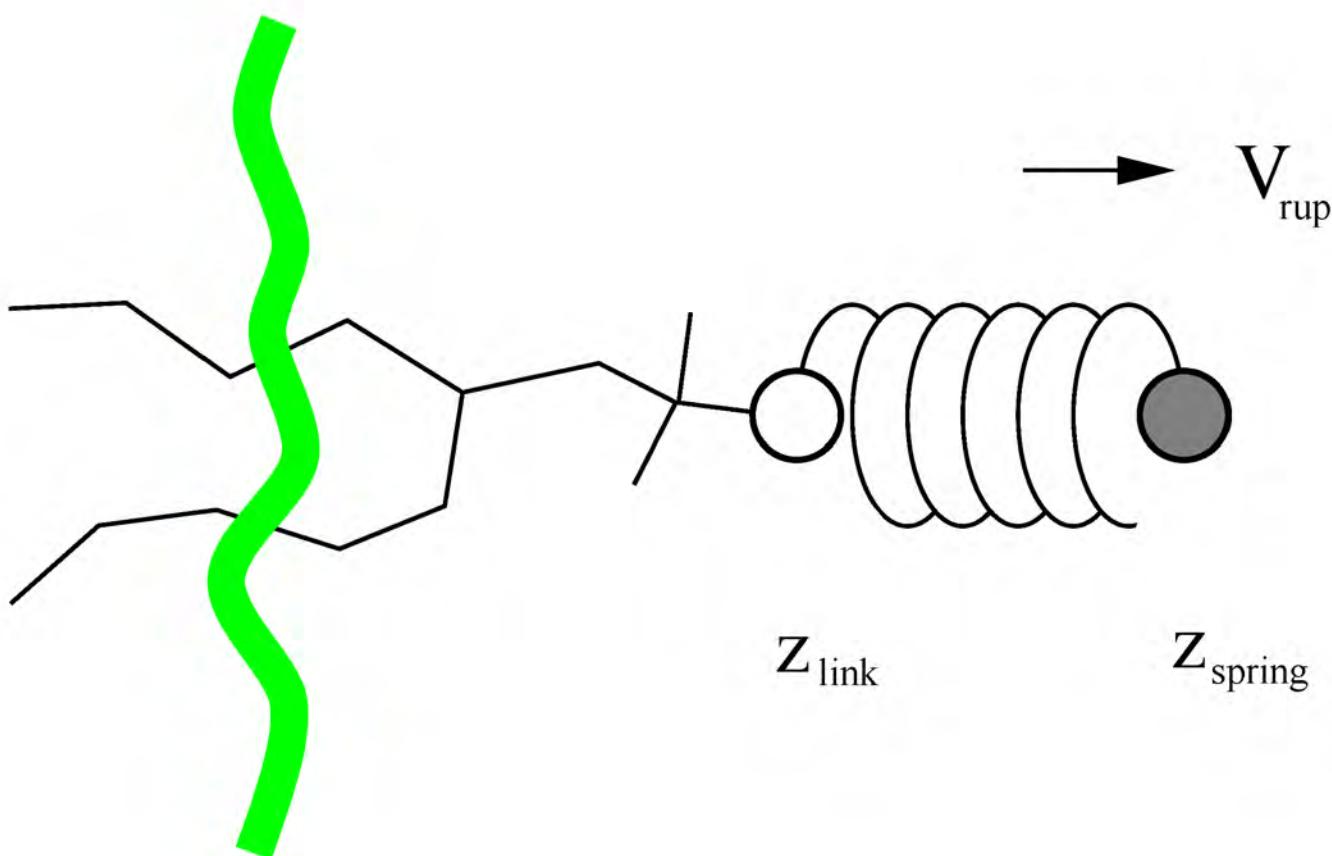


图6.1: 使用伞形牵引将脂分子拉出双脂层的示意图. V_{rup} 为弹簧收缩速度, Z_{link} 为弹簧附着的原子, Z_{spring} 为弹簧的位置.

GMX支持三种不同类型的牵引计算, 对所有类型的牵引, 参考距离可以是恒定, 或是随时间线性变化.

1. 伞形牵引 在两个组的质心之间施加简谐势. 因此, 力正比于位移.
2. 约束牵引 约束两组质心之间的距离. 约束力可以写入到文件中. 此方法使用SHAKE算法, 但当仅约束两个组时,

为精确只需要1次迭代.

3. 恒力牵引 在两个质心之间施加恒定的力. 因此, 势能是线性的. 在这种情况下, 牵引速度没有参考距离.

质量中心的定义

在GROMACS中, 有三种方法来定义的一组原子的质量中心. 标准方式是使用“普通”质心, 可能带有附加的权重因子. 使用周期性边界条件时, 就无法再唯一地定义一组原子的质量中心. 因此, 会使用一个参考原子. 为确定质量中心, 对于组中其他所有的原子, 会使用到参考原子最近的周期影象. 这样唯一地定义了质量中心. 默认情况下, 会使用中间的原子(根据拓扑中的顺序)作为参考原子, 但用户也可以选择使用任何其原子, 如果它更接近于组的中心.

对于多层次系统, 例如双脂层, 可能需要计算脂分子的PMF, 它是脂分子到整个双层的距离的函数. 在这种情况下, 整个双层可作为参考组, 但也可以为PMF定义更局部的反应坐标. `.mdp` 选项 `pull_geometry = cylinder` 并不使用参考组中的所有原子, 而是动态地仅仅使用那些位于圆柱内的原子. 圆柱的半径为 `r_1`, 围绕着通过牵引组的牵引向量. 这仅适用于一维定义的距离, 并且圆柱的长轴沿着定义距离的方向. 也可以使用 `r_0` 定义第二个圆柱, 并且使用一个线性切换函数对距离位于 `r_0` 和 `r_1` 之间的原子进行加权. 这可以平滑原子进出圆柱的影响(导致牵引力出现跃变).

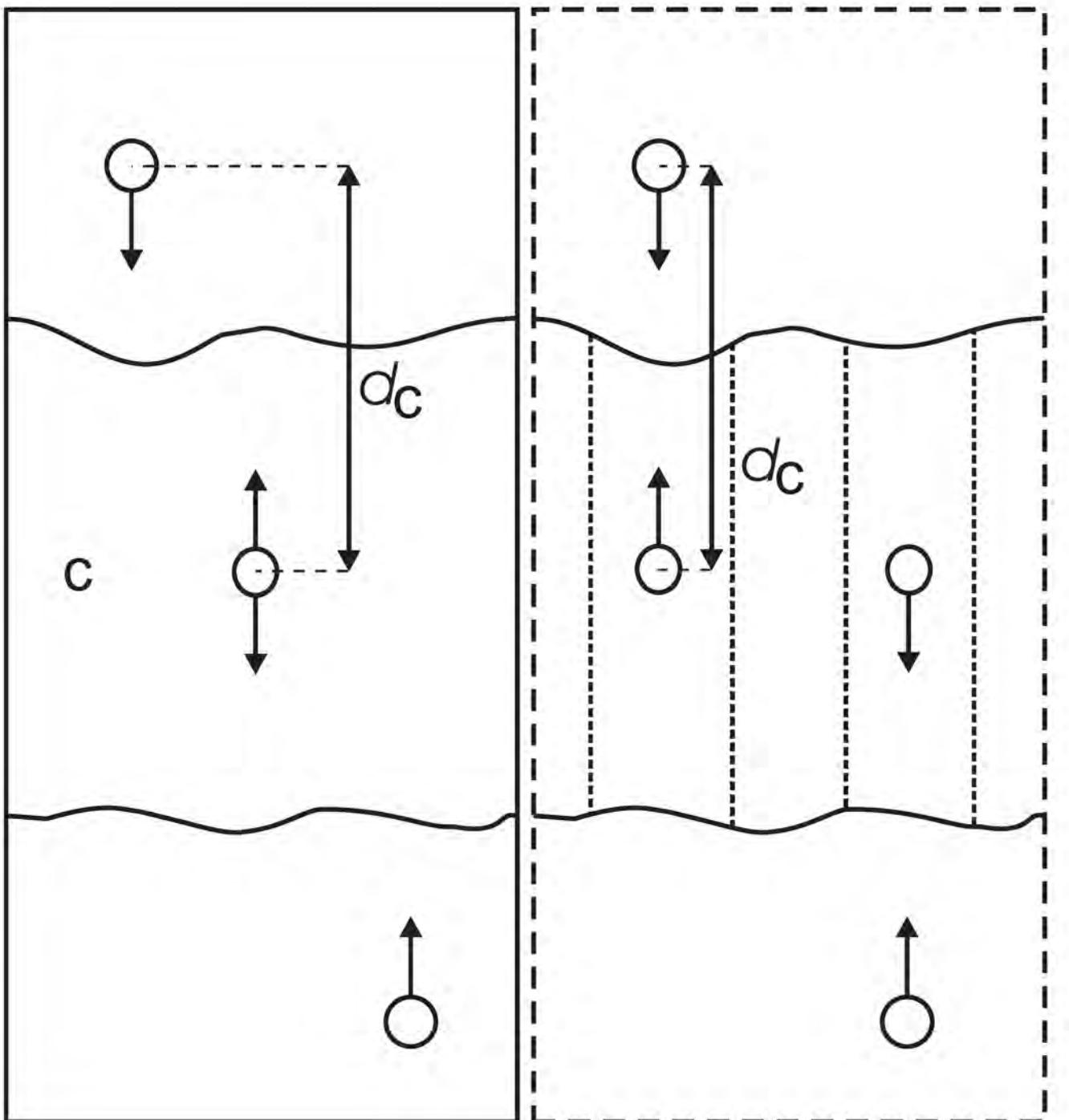


图6.2: 比较普通质心参考组和用于界面体系的圆柱参考组. C为参考组. 圆圈代表两个组的质心加上参考组, d_c 为参考距离.

对周期性的体系中的一组分子, 可能无法明确地定义一个普通的参考组. 例如在 x 和 y 方向周期性相连, 但在 z 方向具有两个液-气界面的水板坯. 在这样的设置中, 水分子可以从液体中蒸发, 并穿过蒸汽, 通过周期性边界, 到达另一个界面. 这种体系固有的周期性使得我们无法合适地定义一个沿着 z 方向的“普通”质心. 一个合适的解决方法对参考组中的所有原子使用一个余弦形的加权剖面. 此剖面单元格胞中的单周期余弦, 并对其相位进行了优化, 使得能够给出权重的最大加和, 包括质量权重. 当所有原子都位于单元格胞长度一半的范围内时, 这种方法提供了唯一和连续的参考位置, 几乎等同于普通的质心位置. 请参看文献[133]了解详细信息.

当在计算过程中使用相对权重 ω_i 时, 无论是输入中提供的权重, 还是因圆柱几何或余弦加权引入的权重, 都需要对权重进行缩放, 以保证动量守恒:

$$\omega'_i = \omega_i \sum_{j=1}^N \omega_j m_j \left/ \sum_{j=1}^N \omega_j^2 m_j \right. \quad (6.5)$$

其中 m_i 为组中原子 j 的质量. 计算约束力时所需要的组的质量为:

$$M = \sum_{i=1}^N \omega'_i m_i \quad (6.6)$$

加权的质量中心的定义为:

$$\mathbf{r}_{com} = \sum_{i=1}^N \omega'_i m_i \mathbf{r}_i \left/ M \right. \quad (6.7)$$

根据质心, 可计算作用在每组上的AFM, 约束或伞形力 \mathbf{F}_{com} . 作用于组质心上的力会按下式重新分配到原子上:

$$\mathbf{F}_i = \frac{\omega'_i m_i}{M} \mathbf{F}_{com} \quad (6.8)$$

局限性

有一个理论局限性: 严格的来说, 只能计算某些组之间的约束力, 这些组没有通过约束连接到体系的其余部分. 如果一个组包含的一部分分子的键长被约束, 应该同时迭代牵引约束和LINCS或SHAKE的键约束算法. GROMACS还无法做到这点. 这意味着, 若模拟时在 `.mdp` 文件中使用了 `constraints = all-bonds` 选项, 严格地说, 牵引仅限于整个分子或分子组. 在一些情况下, 通过使用自由能代码可以避免此限制, 参见6.7节. 在实践中, 当牵引组包含大量原子和/或牵引力很小时, 因不对两种约束算法同时进行迭代导致的误差可忽略不计. 在这些情况下, 相比于键长, 牵引组的约束校正唯一很小.

6.5 强制旋转

这个模块可用于强制一组原子进行旋转, 例如一个蛋白质亚基的旋转. 有多种旋转势能, 其中复杂的类型, 在模拟过程中允许灵活地适应旋转亚基以及局部旋转轴. 一个示例应用可以在文献[134]中找到.

6.5.1 固定轴旋转

具有各向同性势的固定轴

在固定轴方法中(参见图6.3 B), 对一组 N 个, 位置为 \mathbf{x}_i 的原子(记为“旋转组”), 作用于其上的力矩是通过旋转一组参考原子位置-通常是他们的初始位置 \mathbf{y}_i^0 -施加的, 旋转以恒定角速度 ω 绕着由方向向量 $\hat{\mathbf{v}}$ 和支点 \mathbf{u} 定义的轴线进行. 为此, 每个位置为 \mathbf{x}_i 的原子利用“虚拟”弹簧势能附着到它移动的参考位置 $\mathbf{y}_i = \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u})$, 其中 $\boldsymbol{\Omega}(t)$ 是一个矩阵, 描述了绕轴的旋转. 对最简单的情况, “弹簧”是利用简谐势描述的,

$$V^{\text{iso}} = \frac{k}{2} \sum_{i=1}^N \omega_i [\boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u}) - (\mathbf{x}_i - \mathbf{u})]^2 \quad (6.9)$$

并具有可选的质量加权前因子 $\omega_i = Nm_i/M$, 总质量 $M = \sum_{i=1}^N m_i$. 旋转矩阵 $\boldsymbol{\Omega}(t)$ 为

$$\boldsymbol{\Omega}(t) = \begin{pmatrix} \cos \omega t + v_x^2 \xi & v_x v_y \xi - v_z \sin \omega t & v_x v_z \xi + v_y \sin \omega t \\ v_x v_y \xi + v_z \sin \omega t & \cos \omega t + v_y^2 \xi & v_y v_z \xi - v_x \sin \omega t \\ v_x v_z \xi - v_y \sin \omega t & v_y v_z \xi + v_x \sin \omega t & \cos \omega t + v_z^2 \xi \end{pmatrix}$$

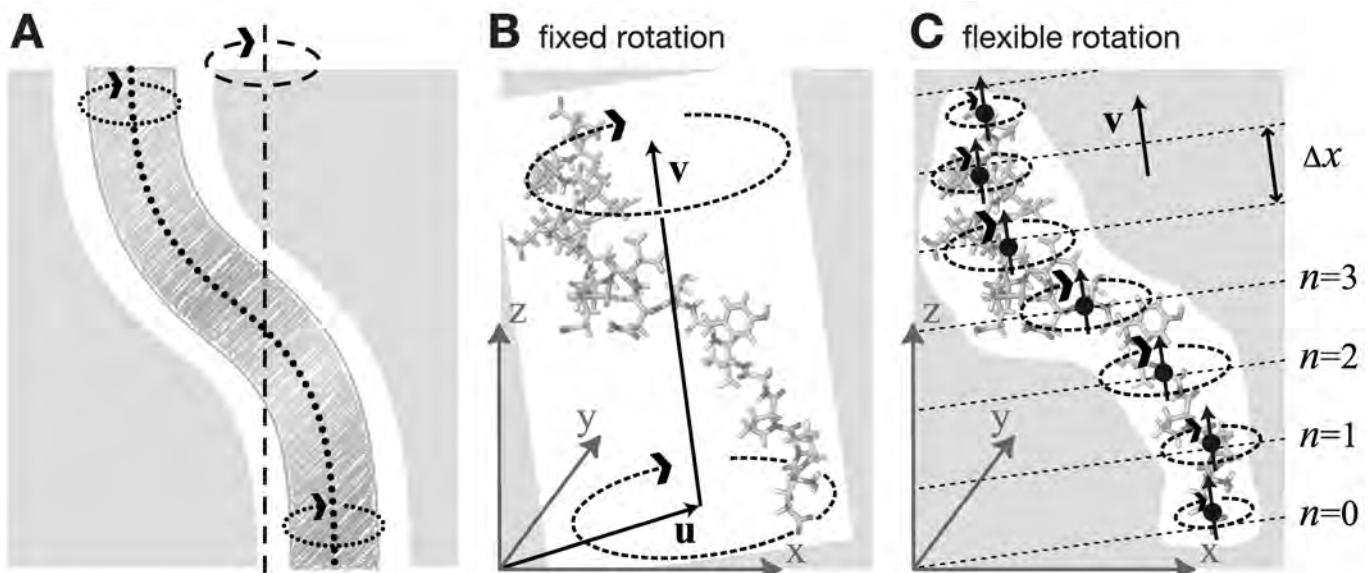


图6.3: 固定轴旋转和柔性轴旋转的比较. A: 当使用固定转轴(虚线)时, 旋转白色管状空腔内的形状可产生假象. 更实际的, 形状会围绕成像一个轴承(灰色)内部的柔性的管道清洁器(虚线). B: 围绕轴 \mathbf{v} 与由向量 \mathbf{u} 指定的支点进行的固定旋转. C: 将旋转片段细分为板坯, 每个板坯具有独立的旋转轴(\uparrow)和支点(\bullet)以满足灵活性. 具有索引 n 和 $n + 1$ 的两个板坯之间的距离为 Δx .

其中 v_x, v_y, v_z 为归一化的旋转向量 \hat{v} 的分量, $\xi := 1 - \cos(\omega t)$. 如图6.4A所示, 对单个原子 j , 旋转矩阵 $\boldsymbol{\Omega}(t)$ 作用于 $t = t_0$ 时刻原子 j 的初始参考位置 $\mathbf{y}_j^0 = \mathbf{x}_j(t_0)$. 在其后的时刻 t , 参考位置已旋转远离其初始位置(沿蓝色虚线), 从而产生力

$$\mathbf{F}_j^{\text{iso}} = -\nabla_j V^{\text{iso}} = k\omega_j \left[\boldsymbol{\Omega}(t)(\mathbf{y}_j^0 - \mathbf{u}) - (\mathbf{x}_j - \mathbf{u}) \right] \quad (6.10)$$

此力指向参考位置.

无支点各向同性势

不使用固定的支点矢量 \mathbf{u} , 该势能使用旋转组的质量中心 \mathbf{x}_c 作为旋转轴的支点,

$$\mathbf{x}_c = \frac{1}{M} \sum_{i=1}^N m_i \mathbf{x}_i, \quad \mathbf{y}_c^0 = \frac{1}{M} \sum_{i=1}^N m_i \mathbf{y}_i^0 \quad (6.11)$$

这就得到了“无支点”的各向同性势

$$V^{\text{iso-pf}} = \frac{k}{2} \sum_{i=1}^N \omega_i [\boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{y}_c^0) - (\mathbf{x}_i - \mathbf{x}_c)]^2 \quad (6.12)$$

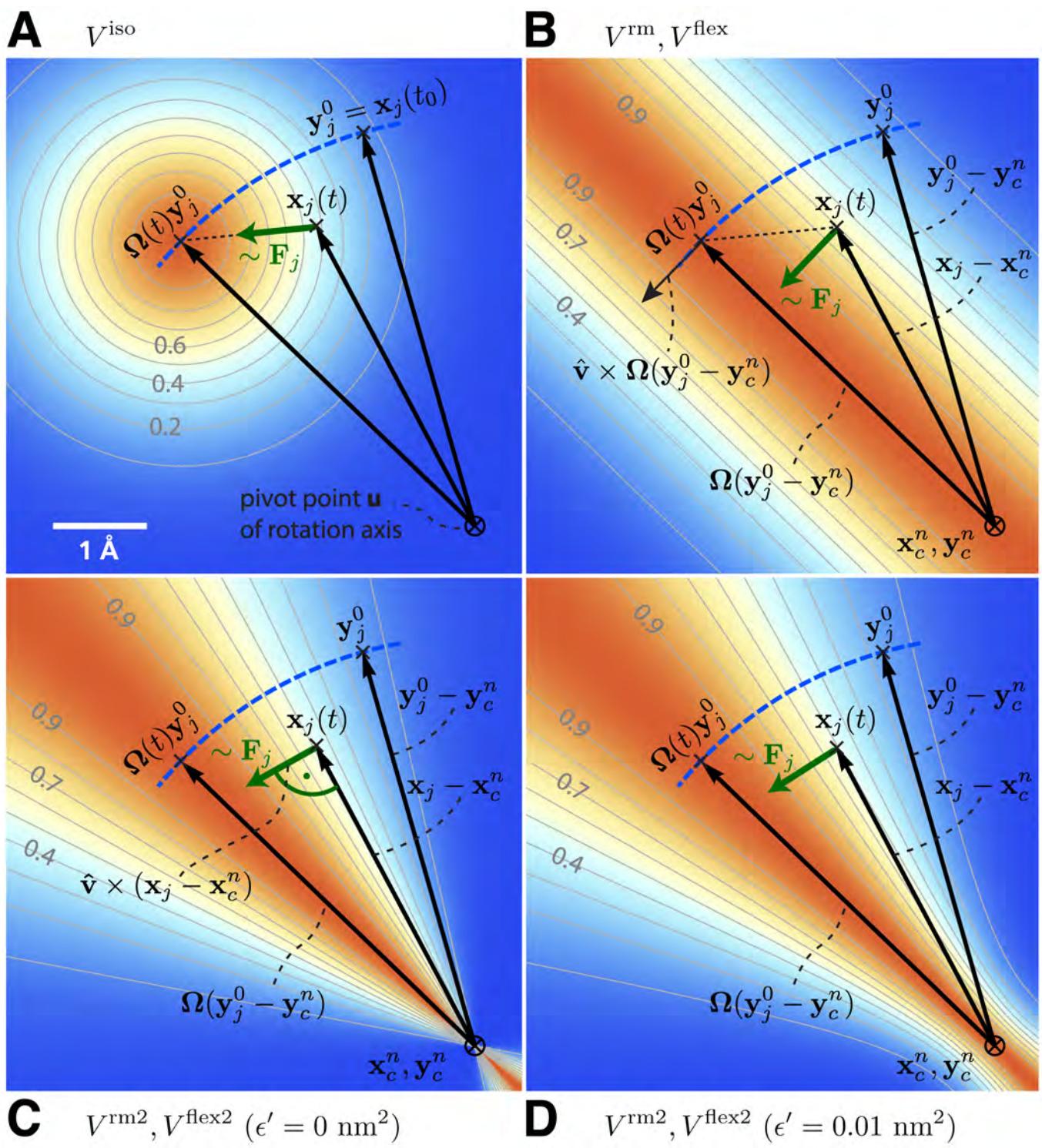


图6.4: 不同的旋转势的选择和符号的定义. 对于 $x_j(t)$ 的单个原子显示了所有四个势能函数 V (彩色编码). A: 各向同性势 V^{iso} , B: 径向运动势 V^{rm} 和柔性势 V^{flex} , C-D: 径向运动二型势 $V^{\text{rm}2}$ 和柔性二型势 $V^{\text{flex}2}$, 其中 ϵ' 分布为 0 nm^2 (C)和 0.01 nm^2 (D). 旋转轴垂直于平面并标记为 \otimes . 浅灰色等值线表示 x_j 平面内的波尔兹曼因子 $e^{-V/(k_B T)}$, $T = 300 \text{ K}$, $k = 200 \text{ kJ}/(\text{mol}\cdot\text{nm}^2)$. 绿色箭头显示了作用于原子 j 上的力 F_j 的方向; 蓝色虚线表示参考位置的运动.

相应的力为

$$F^{\text{iso-pf}} = k\omega_j \left[\Omega(t)(y_j^0 - y_c^0) - (x_j - x_c) \right] \quad (6.13)$$

如果不使用质量加权, 支点 x_c 为组的几何中心.

平行运动势的变体

由各向同性势产生的力(方程6.9和6.12)还含有平行于旋转轴的成分, 从而限制了沿整个旋转组(对 V^{iso})轴线或旋转

组内(对 $V^{\text{iso-pf}}$)轴线的运动. 如果倾向于不限制沿轴线的运动, 我们已经通过消除势能所有平行于旋转轴的分量实现了“平行运动”的一种变体. 通过投影参考和实际位置之间的距离向量

$$\mathbf{r}_i = \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u}) - (\mathbf{x}_i - \mathbf{u}) \quad (6.14)$$

到垂直于旋转矢量的平面上,

$$\mathbf{r}_i^\perp := \mathbf{r}_i - (\mathbf{r}_i \cdot \hat{\mathbf{v}})\hat{\mathbf{v}} \quad (6.15)$$

得到

$$\begin{aligned} V^{\text{pm}} &= \frac{k}{2} \sum_{i=1}^N \omega_i (\mathbf{r}_i^\perp)^2 \\ &= \frac{k}{2} \sum_{i=1}^N \omega_i \{ \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u}) - (\mathbf{x}_i - \mathbf{u}) \\ &\quad - \{ [\boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u}) - (\mathbf{x}_i - \mathbf{u})] \cdot \hat{\mathbf{v}} \} \hat{\mathbf{v}} \}^2 \end{aligned} \quad (6.16)$$

同样的

$$\mathbf{F}_j^{\text{pm}} = k\omega_j \mathbf{r}_j^\perp \quad (6.17)$$

无支点的平行运动势

将方程6.16中的固定支点 \mathbf{u} 替换为质量中心 \mathbf{x}_c 就得到了平行运动势无支点的变体形式. 对

$$\mathbf{s}_i = \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{y}_c^0) - (\mathbf{x}_i - \mathbf{x}_c) \quad (6.18)$$

各自的势能和力为

$$V^{\text{pm-pf}} = \frac{k}{2} \sum_{i=1}^N \omega_i (\mathbf{s}_i^\perp)^2 \quad (6.19)$$

$$\mathbf{F}_j^{\text{pm-pf}} = k\omega_j \mathbf{s}_j^\perp \quad (6.20)$$

径向运动势

在上述变体中, 转动势能的最小点位于参考位置 \mathbf{y}_i 上的一个点(对各向同性势), 或者是通过 \mathbf{y}_i 平行于旋转轴的一条线(对平行运动势). 结果, 径向力限制了原子的径向运动. 接下来的两类旋转势, V^{rm} 和 V^{rm2} , 大大减少甚至消除了这种效应. 第一类变体势, V^{rm} (图6.4B), 消除了所有平行于连接参考原子和旋转轴向量的力的分量,

$$V^{\text{rm}} = \frac{k}{2} \sum_{i=1}^N \omega_i [\mathbf{p}_i \cdot (\mathbf{x}_i - \mathbf{u})]^2 \quad (6.21)$$

其中

$$\mathbf{p}_i := \frac{\hat{\mathbf{v}} \times \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u})}{\|\hat{\mathbf{v}} \times \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u})\|} \quad (6.22)$$

该变体只取决于原子 i 到由 $\hat{\mathbf{v}}$ 和 $\boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u})$ 张成的平面的距离 $\mathbf{p}_i \cdot (\mathbf{x}_i - \mathbf{u})$. 相应的力为

$$\mathbf{F}_j^{\text{rm}} = -k\omega_j \left[\mathbf{p}_j \cdot (\mathbf{x}_j - \mathbf{u}) \right] \mathbf{p}_j \quad (6.23)$$

无支点的径向运动势

类似于无支点的各向同性势, 我们可以得到上面势能无支点的形式. 对

$$\mathbf{q}_i := \frac{\hat{\mathbf{v}} \times \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u})}{\|\hat{\mathbf{v}} \times \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u})\|} \quad (6.24)$$

对这种径向运动势无支点的变体形式, 势能和力分别为

$$V^{\text{rm-pf}} = \frac{k}{2} \sum_{i=1}^N \omega_i [\mathbf{q}_i \cdot (\mathbf{x}_i - \mathbf{x}_c)]^2 \quad (6.25)$$

$$\mathbf{F}_j^{\text{rm-pf}} = -k\omega_j \left[\mathbf{q}_j \cdot (\mathbf{x}_j - \mathbf{x}_c) \right] \mathbf{q}_j + k \frac{m_j}{M} \sum_{i=1}^N \omega_i [\mathbf{q}_i \cdot (\mathbf{x}_i - \mathbf{x}_c)] \mathbf{q}_i \quad (6.26)$$

二型径向运动势

如图6.4B, 由 V^{rm} 产生的力仍包含小的二阶径向分量. 大多数情况下, 这种扰动可以容忍; 如果不能容忍, 可以使用备选的势能函数 V^{rm2} , 它完全消除了力径向部分的贡献, 如图6.4C所示,

$$V^{\text{rm2}} = \frac{k}{2} \sum_{i=1}^N \omega_i \frac{[(\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{u}) \cdot \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u})]^2}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{u})\|^2 + \epsilon'} \quad (6.27)$$

其中, 引入了小参数 ϵ' 以避免奇异性. 对 $\epsilon' = 0 \text{ nm}^2$, 等势能平面由 $\mathbf{x}_i - \mathbf{u}$ 和 $\hat{\mathbf{v}}$ 张成, 得到的力垂直于 $\mathbf{x}_i - \mathbf{u}$, 因此不会再压缩会扩展远离或朝向旋转轴的结构部分.

在方程6.27的分母中使用一个小的正值的 ϵ' (例如, $\epsilon' = 0.01 \text{ nm}^2$, 图6.4D)会得到一个定义良好的势能函数, 并且连续的力也靠近旋转轴, 但 $\epsilon' = 0 \text{ nm}^2$ 时(图6.4C)不是这样. 对

$$\mathbf{r}_i := \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{u}) \quad (6.28)$$

$$\mathbf{s}_i := \frac{\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{u})}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{u})\|} \equiv \Psi_i \hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{u}) \quad (6.29)$$

$$\Psi_i^* := \frac{1}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{u})\|^2 + \epsilon'} \quad (6.30)$$

作用在原子 j 上的力为

$$\mathbf{F}_j^{\text{rm2}} = -k \left\{ \omega_j (\mathbf{s}_j \cdot \mathbf{r}_j) \left[\frac{\Psi_j^*}{\Psi_j} \mathbf{r}_j - \frac{\Psi_j^{*2}}{\Psi_j^3} (\mathbf{s}_j \cdot \mathbf{r}_j) \mathbf{s}_j \right] \right\} \times \hat{\mathbf{v}} \quad (6.31)$$

二型无支点径向运动势

上面势能的无支点变体形式为

$$V^{\text{rm2-pf}} = \frac{k}{2} \sum_{i=1}^N \omega_i \frac{[(\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c)) \cdot \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{y}_c)]^2}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c)\|^2 + \epsilon'} \quad (6.32)$$

其中

$$\mathbf{r}_i := \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{y}_c) \quad (6.33)$$

$$\mathbf{s}_i := \frac{\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c)}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c)\|} \equiv \Psi_i \hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c) \quad (6.34)$$

$$\Psi_i^* := \frac{1}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c)\|^2 + \epsilon'} \quad (6.35)$$

作用于原子 j 上的力为

$$\begin{aligned} \mathbf{F}_j^{\text{rm2-pf}} = & -k \left\{ \omega_j (\mathbf{s}_j \cdot \mathbf{r}_j) \left[\frac{\Psi_j^*}{\Psi_j} \mathbf{r}_j - \frac{\Psi_j^{*2}}{\Psi_j^3} (\mathbf{s}_j \cdot \mathbf{r}_j) \mathbf{s}_j \right] \right\} \times \hat{\mathbf{v}} \\ & + k \frac{m_j}{M} \left\{ \sum_{i=1}^N \omega_i (\mathbf{s}_i \cdot \mathbf{r}_i) \left[\frac{\Psi_i^*}{\Psi_i} \mathbf{r}_i - \frac{\Psi_i^{*2}}{\Psi_i^3} (\mathbf{s}_i \cdot \mathbf{r}_i) \mathbf{s}_i \right] \right\} \times \hat{\mathbf{v}} \end{aligned} \quad (6.36)$$

6.5.2 柔性轴旋转

如图6.3A-B所示, 固定轴旋转方案的刚体行为对于许多应用而言是一个缺点. 特别是, 当平衡的原子位置直接依赖于参考位置时, 旋转组的变形会被抑制. 为避免这种限制, 现在将方程6.26和6.32推广到图6.3C示意的“柔性轴”. 具体做法是将旋转组划分为一些列等间距的板坯, 每个板坯垂直于旋转向量, 并对每个板坯施加独立的旋转势. 图6.3C中点线表示板坯的中间平面, 大的黑点表示板坯的中心.

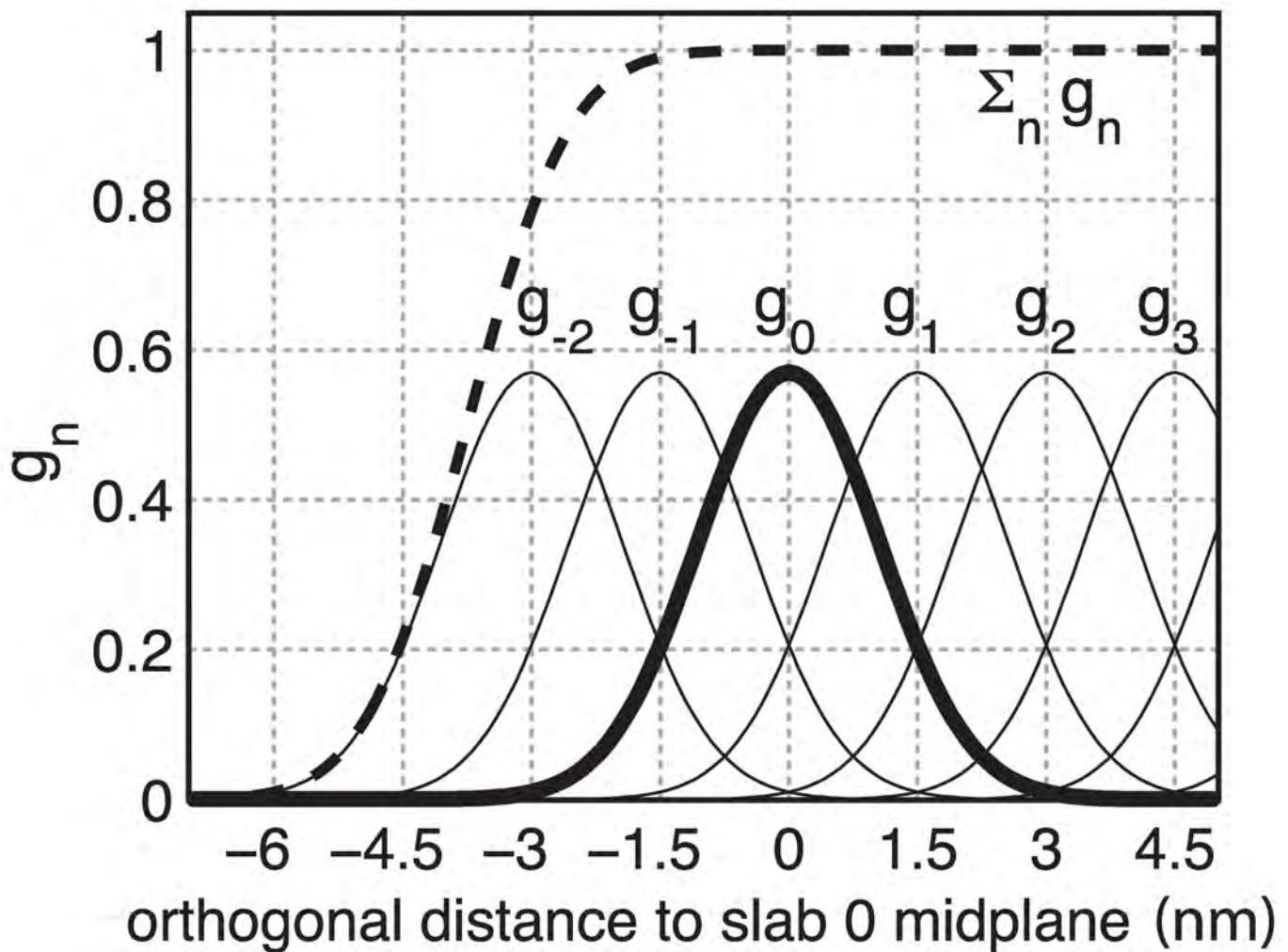


图6.5: 中心位于 $n\Delta x$ 的高斯函数 g_n , 板坯距离 $\Delta x = 1.5 \text{ nm}$, $n \geq -2$. 高斯函数 g_0 以粗体突出显示; 虚线表示所显示的高斯函数的总和.

为避免势能与力的不连续性, 我们定义了“软板坯”, 根据每个板坯 n 对总势能函数 V^{flex} 的贡献进行加权, 加权时使用高斯函数

$$g_n(\mathbf{x}_i) = \Gamma \exp\left(-\frac{\beta_n^2(\mathbf{x}_i)}{2\sigma^2}\right) \quad (6.37)$$

其中中心位于第 n 个板坯中间的平面上. 这里的 σ 为高斯函数的宽度, Δx 为相邻板坯间的距离,

$$\beta_n(\mathbf{x}_i) := \mathbf{x}_i \cdot \hat{\mathbf{v}} - n\Delta x \quad (6.38)$$

最方便的一个选择是 $\sigma = 0.7\Delta x$, 并且

$$1/\Gamma = \sum_{n \in Z} \exp\left(-\frac{(n - \frac{1}{4})^2}{2 \cdot 0.7^2}\right) \approx 1.75464$$

这样得到的总和几乎是恒定的, 基本上与 \mathbf{x}_i 无关(图6.5中虚线所示), 即

$$\sum_{n \in Z} g_n(\mathbf{x}_i) = 1 + \epsilon(\mathbf{x}_i) \quad (6.39)$$

其中 $|\epsilon(\mathbf{x}_i)| < 1.3 \cdot 10^{-4}$. 这一选择也意味着板坯对力的各自贡献加起来等于1, 这样就不再需要进一步的归一化

了.

对每个板坯中心 \mathbf{x}_c , 所有原子的贡献来源于其高斯加权的(也可选择使用质量加权)位置向量 $g_n(\mathbf{x}_i)\mathbf{x}_i$. 瞬时板坯中心 \mathbf{x}_c^n 根据当前位置 \mathbf{x}_i 进行计算

$$\mathbf{x}_c^n = \frac{\sum_{i=1}^N g_n(\mathbf{x}_i)m_i \mathbf{x}_i}{\sum_{i=1}^N g_n(\mathbf{x}_i)m_i} \quad (6.40)$$

而参考中心 \mathbf{y}_c^n 根据参考位置 \mathbf{y}_i^0 进行计算

$$\mathbf{y}_c^n = \frac{\sum_{i=1}^N g_n(\mathbf{y}_i^0)m_i \mathbf{y}_i^0}{\sum_{i=1}^N g_n(\mathbf{y}_i^0)m_i} \quad (6.41)$$

由于 g_n 衰减很快, 每个板坯将主要涉及距板坯中心 $\approx 3\Delta x$ 内的原子的贡献.

柔性轴势

我们考虑两种柔性轴的变体. 在第一种变体中, 将高斯加权的板坯划分过程应用到径向运动势中(方程6.26/图6.4B)中, 就得到了板坯 n 的贡献

$$V^n = \frac{k}{2} \sum_{i=1}^N \omega_i g_n(\mathbf{x}_i) [\mathbf{q}_i^n \cdot (\mathbf{x}_i - \mathbf{x}_c^n)]^2$$

总的势能函数

$$V^{\text{flex}} = \sum_n V^n \quad (6.42)$$

注意, 方程6.26中使用的全局质心 \mathbf{x}_c 现在被 \mathbf{x}_c^n 代替, 它是板坯的质心. 对

$$\mathbf{q}_i^n := \frac{\hat{\mathbf{v}} \times \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{y}_c^n)}{\|\hat{\mathbf{v}} \times \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{y}_c^n)\|} \quad (6.43)$$

$$b_i^n := \mathbf{q}_i^n \cdot (\mathbf{x}_i - \mathbf{x}_c^n) \quad (6.44)$$

作用在原子 j 上的力为

$$\begin{aligned} \mathbf{F}_j^{\text{flex}} = & -k\omega_j \sum_n g_n(\mathbf{x}_j) b_j^n \left\{ \mathbf{q}_j^n - b_j^n \frac{\beta_n(\mathbf{x}_j)}{2\sigma^2} \hat{\mathbf{v}} \right\} \\ & + km_j \sum_n \frac{g_n(\mathbf{x}_j)}{\sum_h g_n(\mathbf{x}_h)} \sum_{i=1}^N \omega_i g_n(\mathbf{x}_i) b_i^n \left\{ \mathbf{q}_i^n - \frac{\beta_n(\mathbf{x}_j)}{\sigma^2} [\mathbf{q}_i^n \cdot (\mathbf{x}_j - \mathbf{x}_c^n)] \hat{\mathbf{v}} \right\} \end{aligned} \quad (6.45)$$

注意, 对定义的 V^{flex} , 板坯在空间中是固定的, 参考中心 \mathbf{y}_c^n 也一样. 如果在模拟中, 旋转组沿 \mathbf{v} 方向移动得太远, 它可能会进入一个没有定义参考板坯中心的区域-由于缺少附近的参考位置-这样就不可能对势能进行计算. 因此, 对这个势能, 我们给出了一个略加修改的版本, 它能够避免前面所说的问题, 具体做法是将板坯 $n = 0$ 的中间平面附着到旋转组的中心, 这样得到的板坯会与旋转组一起移动. 通过从位置中减去组的质心 \mathbf{x}_c ,

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x}_c, \tilde{\mathbf{y}}_i^0 = \mathbf{y}_i^0 - \mathbf{y}_c^0 \quad (6.46)$$

这样

$$V^{\text{flex2}} = \frac{k}{2} \sum_n \sum_{i=1}^N \omega_i g_n(\tilde{\mathbf{x}}_i) \left[\frac{(\hat{\mathbf{v}} \times \boldsymbol{\Omega}(t)(\tilde{\mathbf{y}}_i^0 - \tilde{\mathbf{y}}_c^n)) \cdot (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_c^n)}{\|\hat{\mathbf{v}} \times \boldsymbol{\Omega}(t)(\tilde{\mathbf{y}}_i^0 - \tilde{\mathbf{y}}_c^n)\|} \right]^2 \quad (6.47)$$

为了简化力的导数, 以及出于效率的考虑, 我们在这里假定 \mathbf{x}_c 是恒定的, 因此 $\partial \mathbf{x}_c / \partial x = \partial \mathbf{x}_c / \partial y = \partial \mathbf{x}_c / \partial z = 0$, 这种得到的力误差很小(数量级为 $O(1/N)$ 或 $O(m_j/M)$, 如果使用质量加权的话), 因此可以容忍. 利用这一假定, 力 $\mathbf{F}^{\text{flex-t}}$ 与方程6.45具有相同的形式.

二型柔性轴势

第二种变体是将势 V^{rm2} (方程6.32)用于划分的板坯, 得到了无径向力贡献的柔性轴(图6.4C)

$$V^{\text{flex2}} = \frac{k}{2} \sum_{i=1}^N \sum_n \omega_i g_n(\mathbf{x}_i) \frac{[(\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c^n)) \cdot \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{y}_c^n)]^2}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c^n)\|^2 + \epsilon'} \quad (6.48)$$

其中

$$\mathbf{r}_i^n := \boldsymbol{\Omega}(t)(\mathbf{y}_i^0 - \mathbf{y}_c^n) \quad (6.49)$$

$$\mathbf{s}_i^n := \frac{\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c^n)}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c^n)\|} \equiv \psi_i \hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c^n) \quad (6.50)$$

$$\psi_i^* := \frac{1}{\|\hat{\mathbf{v}} \times (\mathbf{x}_i - \mathbf{x}_c^n)\|^2 + \epsilon'} \quad (6.51)$$

$$W_j^n := \frac{g_n(\mathbf{x}_j)m_j}{\sigma_h g_n(\mathbf{x}_h)m_h} \quad (6.52)$$

$$\mathbf{S}^n := \sum_{i=1}^N \omega_i g_n(\mathbf{x}_i) (\mathbf{s}_i^n \cdot \mathbf{r}_j^n) \left[\frac{\psi_i^*}{\psi_i} \mathbf{r}_i^n - \frac{\psi_i^{*2}}{\psi_i^3} (\mathbf{s}_i^n \cdot \mathbf{r}_i^n) \mathbf{s}_i^n \right] \quad (6.53)$$

作用于原子 j 上的力为

$$\begin{aligned} \mathbf{F}_j^{\text{flex2}} = & -k \left\{ \sum_n \omega_j g_n(\mathbf{x}_j) (\mathbf{s}_j^n \cdot \mathbf{r}_j^n) \left[\frac{\psi_j^*}{\psi_j} \mathbf{r}_j^n - \frac{\psi_j^{*2}}{\psi_j^3} (\mathbf{s}_j^n \cdot \mathbf{r}_j^n) \mathbf{s}_j^n \right] \right\} \times \hat{\mathbf{v}} \\ & + k \left\{ \sum_n W_j^n \mathbf{S}^n \right\} \times \hat{\mathbf{v}} - k \left\{ \sum_n W_j^n \frac{\beta_n(\mathbf{x}_j)}{\sigma^2} \frac{1}{\psi_j} \mathbf{s}_j^n \cdot \mathbf{S}^n \right\} \hat{\mathbf{v}} \\ & + \frac{k}{2} \left\{ \sum_n \omega_j g_n(\mathbf{x}_j) \frac{\beta_n(\mathbf{x}_j)}{\sigma^2} \frac{\psi_j^*}{\psi_j^2} (\mathbf{s}_j^n \cdot \mathbf{r}_j^n)^2 \right\} \hat{\mathbf{v}} \end{aligned} \quad (6.54)$$

应用变换(6.46)可得到二型柔性势“耐平动”的形式, $V^{\text{flex2-t}}$. 再次, 假定 $\partial \mathbf{x}_c / \partial x, \partial \mathbf{x}_c / \partial y, \partial \mathbf{x}_c / \partial z$ 很小, 得到的方程 $V^{\text{flex2-t}}$ 和 $\mathbf{F}^{\text{flex2-t}}$ 分别类似于 V^{flex2} 和 $\mathbf{F}^{\text{flex2}}$.

6.5.3 用法

为使用应用强制旋转, 可通过 `.mdp` 输入文件中的 `rot_group0`, `rot_group1` 等索引组定义受到旋转势能作用的粒子

i. 参考位置 \mathbf{y}_i^0 由提供给 `grompp` 的特殊 `.trr` 文件中读取。如果发现这样的文件，会使用 $\mathbf{x}_i(t=0)$ 作为参考位置，并写入 `.trr`，这样它们就可以用于随后的设置。势能的所有参数，如 k , ϵ' 等(表 6.1)。由 `.mdp` 参数提供；`rot_type` 选择势能的类型。选项 `rot_massw` 选择是否使用质量加权平均。对柔性势能，截断值 g_n^{\min} (典型值为 0.001)，可确保只计算对 V 和 F 有显著贡献的部分，即那些 $g_n(\mathbf{x}) < g_n^{\min}$ 的项被省略。表6.2总结了写入到附加输出文件的一些量，将在下面讨论它们。

表6.1：各种旋转势能所用的参数。 X 表示给定的势能实际使用的参数。

参数		k	$\hat{\mathbf{v}}$	\mathbf{u}	ω	ϵ'	Δx	g_n^{\min}
.mdp 输入变量名		k	vec	pivot	rate	eps	slab_dist	min_gauss
单位		kJ mol nm ⁻²	-	nm	°/ps	nm ²	nm	-
固定轴势能：	方程							
各向同性 V^{iso}	(6.9)	X	X	X	X	-	-	-
-无支点 $V^{\text{iso-pf}}$	(6.12)	X	X	-	X	-	-	-
平行运动 V^{pm}	(6.16)	X	X	X	X	-	-	-
-无支点 $V^{\text{pm-pf}}$	(6.20)	X	X	-	X	-	-	-
径向运动 V^{rm}	(6.21)	X	X	X	X	-	-	-
-无支点 $V^{\text{rm-pf}}$	(6.26)	X	X	-	X	-	-	-
二型径向 V^{rm2}	(6.27)	X	X	X	X	X	-	-
-无支点 $V^{\text{rm2-pf}}$	(6.32)	X	X	-	X	X	-	-
柔性轴势能：	方程							
柔性 V^{flex}	(6.42)	X	X	-	X	-	X	X
-耐平动 $V^{\text{flex-t}}$	(6.47)	X	X	-	X	-	X	X
二型柔性势 V^{flex2}	(6.48)	X	X	-	X	X	X	X
-耐平动 $V^{\text{flex2-t}}$	-	X	X	-	X	X	X	X

表 6.2：强制旋转模拟时记录在输出文件中的物理量。所有板坯数据每 `nstout` 步输出一次，其他旋转数据每 `nstrout` 步输出一次。

物理量	单位	方程	输出文件	固定	可变
$V(t)$	kJ/mol	参见6.1	<code>rotation</code>	X	X
$\theta_{\text{ref}}(t)$	度	$\theta_{\text{ref}}(t)=\omega t$	<code>rotation</code>	X	X
$\theta_{\text{av}}(t)$	度	(6.55)	<code>rotation</code>	X	-
$\theta_{\text{fit}}(t), \theta_{\text{fit}}(t, n)$	度	(6.57)	<code>rotangles</code>	-	X
$\mathbf{y}_0(n), \mathbf{x}_0(t, n)$	nm	(6.40, 6.41)	<code>rotslabs</code>	-	X
$\tau(t)$	kJ/mol	(6.58)	<code>rotation</code>	X	-
$\tau(t, n)$	kJ/mol	(6.58)	<code>rottorque</code>	-	X

旋转组的角度: 固定轴

对固定轴旋转，组相对于参考组的平均角度 $\theta_{\text{av}}(t)$ ，是通过所有旋转组原子到其参考位置的距离加权的角度偏差来确定的，

$$\theta_{\text{av}} = \sum_{i=1}^N r_i \theta_i / \sum_{i=1}^N r_i \quad (6.55)$$

这里 r_i 为参考位置到旋转轴的距离, 偏差角 θ_i 由原子位置决定, 投射到垂直于旋转轴并通过支点 \mathbf{u} 的平面(参考方程6.15中 \perp 的定义),

$$\cos \theta_i = \frac{(\mathbf{y}_i - \mathbf{u})^\perp \cdot (\mathbf{x}_i - \mathbf{u})^\perp}{\|(\mathbf{y}_i - \mathbf{u})^\perp \cdot (\mathbf{x}_i - \mathbf{u})^\perp\|} \quad (6.56)$$

如果实际结构的旋转在参考前, θ_{av} 的符号选择使得 $\theta_{\text{av}} > 0$.

旋转组角度: 柔性轴

对于柔性轴旋转, 会提供两个输出, 整个旋转组的角度和板坯中片段的分开角度. 确定整个旋转组的角度时, 会使用 RMSD匹配将 \mathbf{x}_i 匹配到 $t = 0$ 时刻的参考位置 \mathbf{y}_i^0 , 得到的 θ_{fit} 是参考为达到最佳匹配必须绕 $\hat{\mathbf{v}}$ 旋转的角度,

$$\text{RMSD}(\mathbf{x}_i, \Omega(\theta_{\text{fit}}) \mathbf{y}_i^0) \stackrel{!}{=} \min \quad (6.57)$$

为了确定每个板坯 n 的局部角度, 会使用板坯 n 的高斯函数对参考和实际位置进行加权, 方程6.57计算的 $\theta_{\text{fit}}(t, n)$ 来自于高斯加权的位置.

对于所有的角度, `.mdp` 输入选项 `rot_fit_method` 控制是进行正常的RMSD匹配, 还是在匹配时将每个位置 \mathbf{x}_i 置于距旋转轴相同距离作为其参考对应 \mathbf{y}_i^0 . 对后一种情况, RMSD只测量了角度差异, 不包含径向部分.

通过搜索最小能量确定角度

作为替代方法, 对 `rot_fit_method = potential`, 旋转组的角度取为旋转势能最小时对应的角度. 因此, 需要对围绕当前参考角度的一系列角度计算其对应的旋转势能. 在这种情况下, `rotangles.log` 输出文件包含了所选角度对应的旋转势能值, `rotation.xvg` 列出了最小势能对应的角度.

力矩

对固定轴转动, 旋转势对应的力矩 $\boldsymbol{\tau}(t)$ 如下

$$\boldsymbol{\tau}(t) = \sum_{i=1}^N \mathbf{r}_i(t) \times \mathbf{f}_i^\perp(t) \quad (6.58)$$

其中 $\mathbf{r}_i(t)$ 为旋转轴到 $\mathbf{x}_i(t)$ 的距离向量, $\mathbf{f}_i^\perp(t)$ 为垂直于 $\mathbf{r}_i(t)$ 和 $\hat{\mathbf{v}}$ 的力分量. 对柔性轴旋转, 会使用板坯的局部旋转轴和高斯加权的位置计算每个板坯的 $\boldsymbol{\tau}_n$.

6.6 计算电生理学

计算电生理学方法(CompEL, Computational Electrophysiology)[135] 可用于模拟穿过膜通道的离子流, 这些离子流是由跨膜电位或离子浓度梯度驱动的. 就像在真正细胞中那样, CompEL 中的跨膜电位是通过对穿过膜的电荷维持一个小小的不平衡 Δq 实现的. Δq 和膜电容给出了电位差:

$$\Delta U = \Delta q / C_{membrane} \quad (6.59)$$

跨膜电场与浓度梯度由 `.mdp` 选项控制, 允许用户设置膜每侧的参考离子数. 如果离子的实际数目与参考数目不同, 并且这种情况持续的时间超过了用户指定的时间, 程序会交换两侧的一些离子对或水对, 直至离子数目恢复到参考数目. 通过计算通道电导性与离子选择性, CompEL 模拟也可用于确定通道的逆电位, 它是电生理实验中获得的重要表征参数.

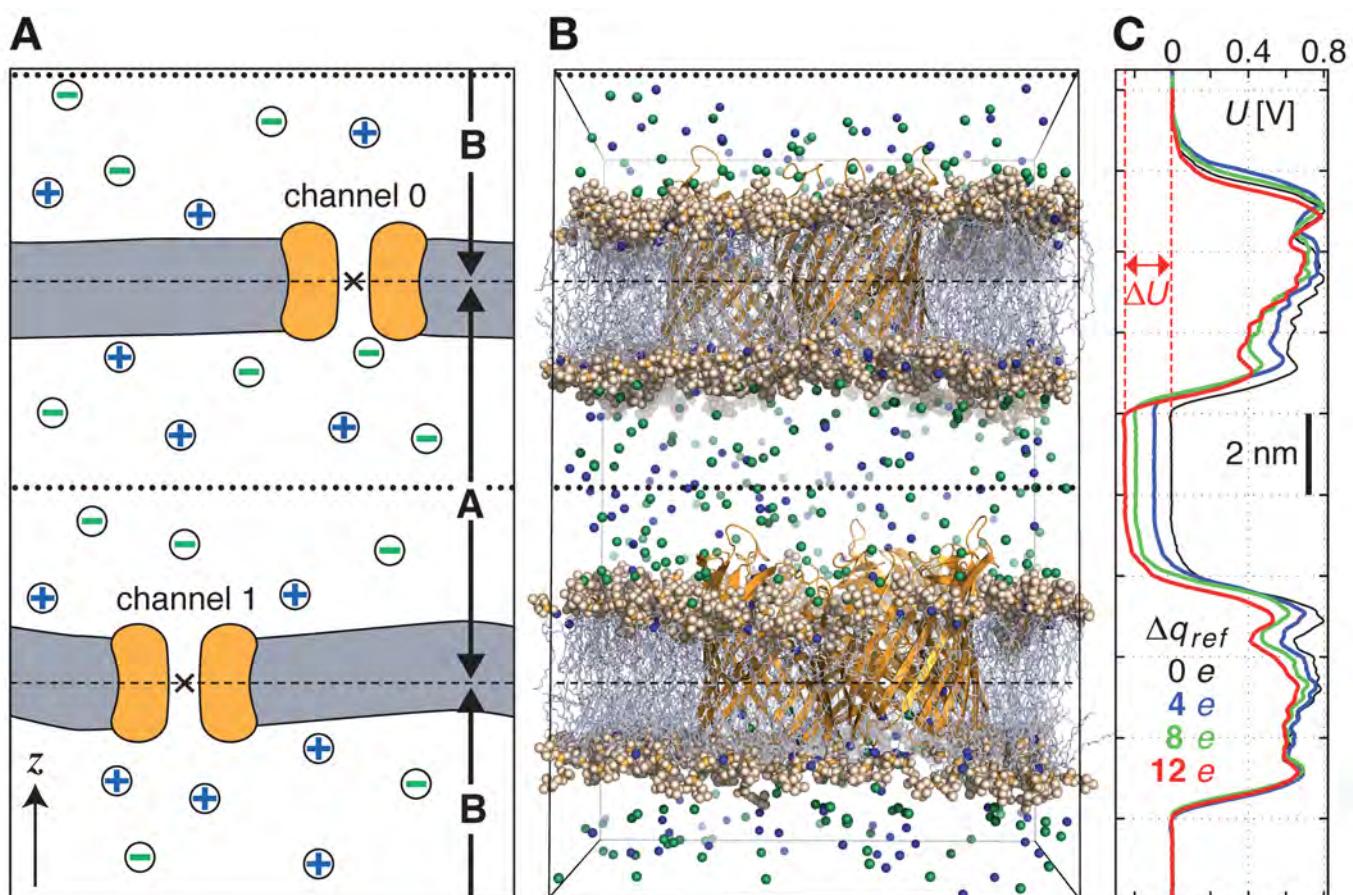


图6.6: CompEL 模拟双膜的典型设置(A, B). 图(C)显示了由选择的两侧之间的电荷不平衡 Δq_{ref} 导致的电位差.

设置CompEL 时, 模拟体系被划分为两个部分, A和B, 它们的离子浓度是独立的, 互不相关. 构建体系的最好方法是利用两个双层体系, 每个都包含感兴趣的通道/孔(图6.6 A, B). 若它们通道的轴方向相同, 这样就可以在正电位和负电位情况下同时观察到通道中的离子流, 这对一些研究, 如通道整流非常重要.

使用 `gmx potential` 工具可以很容易地计算跨膜电位差 Δu . 这样在模拟的每个时间段都可以精确地得到沿 z 轴或孔轴的电势降低(图6.6 C). 在模拟中, 穿过通道的电荷为 q_i 的离子 i 的类型与数目都会写入 `swapiions.xvg` 输出文件中. 利用此输出文件, 可确定在每个时间段 Δt 内的平均通道电导值 G

$$G = \frac{\sum_i n_i q_i}{\Delta t \Delta U} \quad (6.60)$$

离子选择性可通过不同种类离子数流量的比值来计算. 要想得到最好的结果, 需要对许多重合时间段内的值进行平均.

逆电位的计算最好是利用一系列小的模拟, 每个模拟中使用较小但强度变化的离子不平衡来产生给定的跨膜浓度梯度. 例如, 若一侧含有1 M浓度的盐, 另一侧的浓度为0.1 M, 其余部分保持电荷中性, 可使用 $\Delta q = 0e$, $\Delta q = 2e$, $\Delta q = 4e$ 的一系列模拟. 对获得的零电流附近的所有 $I - U$ 对, 利用电流电压关系进行线性拟合就可以得到 U_{rev} .

6.6.1 使用

可利用如下的 `.mdp` 选项控制CompEL 计算:

```
swapcoords      = Z    ; 交换位置 Swap positions: no, X, Y, Z
swap_frequency = 100   ; 尝试交换的频率 Swap attempt frequency
```

若你的膜位于 xy 平面内, 请选择 `Z`(图6.6 A, B). 这样两侧之间的离子交换只依赖于 z 的位置. `swap_frequency` 决定尝试交换的频繁程度, 进行交换时需要在并行队列之间通讯离子, 溶剂和交换组位置的信息, 若交换频率过高, 会降低运算速度.

```
split_group0 = channel0 ; 定义一侧边界 Defines compartment boundary
split_group1 = channel1 ; 定义另一侧边界 Defines other compartment boundary
massw_split0 = no       ; 使用质量权重中心 use mass-weighted center?
massw_split1 = no
```

`split_group0` 和 `split_group1` 为两个索引组, 用于定义两侧之间的边界, 通常是通道的中心. 若 `massw_split0` 或 `massw_split1` 设置为 `yes`, 会使用每个索引组的质心作为边界, 在这里是 z 方向的质心. 否则将使用几何中心(图6.6中A中的 x). 如果像这里一样, 一个膜通道被选作划分组, 其中心将定义两侧之间的分隔平面(图中的虚横线). 所有索引组都必须在索引文件中进行定义.

```
swap_group      = NA+_CL- ; 交换离子 Ions to be included in exchange
solvent_group   = SOL     ; 溶剂分子的组名称 Group name of solvent molecules
cyl0_r          = 5.0     ; 划分组孔半径 Split cylinder 0: pore radius (nm)
cyl0_up         = 0.75    ; 上面延伸 Split cylinder 0 upper extension (nm)
cyl0_down       = 0.75    ; 下面延伸 Split cylinder 0 lower extension (nm)
cyl1_r          = 5.0     ; 其他通道设置 same for other channel
cyl1_up         = 0.75
cyl1_down       = 0.75
coupl_steps     = 10      ; 两次平均之间的交换次数 Average over these many swap steps
threshold       = 1        ; 若小于此值不进行交换 Do not swap if < threshold
```

`swap_group` 设置了流和交换循环中所涉及离子的索引组, `solvent_group` 定义了与离子互换的溶剂组. 圆柱选项只影响离子的计数, 即, 穿过通道0或1的离子会被计数, 计数时依据(通道)圆柱相对于各个划分组位置的半径的定义, 上下扩展. 这并不影响实际的流和交换, 却能提供离子穿过每个通道的次数. 注意, 只有当一个离子在互换步骤中处于定义的划分圆柱范围内时, 才能认为离子通过了某个特定的通道, 若 `swap_frequency` 太高, 特定的离子可能在一个互换步骤中处于A侧, 在下一步骤中却处于B侧, 因此就无法清楚地确定离子到底是穿过了哪个通道.

`coupl_steps` 设置了尝试互换的步骤数, 也就是每侧中实际离子数与参考离子数之间的差距, 在实际交换发生以前必须尝试的次数. 若 `coupl_steps` 设置为1, 那么离子交换与否由瞬时离子分布决定, 而 `coupl_steps` > 1 时则由尝试步骤过程中时间平均的离子分布决定. 这适用于一些情况, 例如, 当离子扩散至两侧的边界时, 会导致大量无效的离子交换, `threshold` 为1意味着进行交换时一侧的平均离子数与设定值至少相差1. 更高的阈值将导致更大容许差距. 离子将交换直至达到设定的数目 ± 阈值.

```
anionsA = -1 ; A侧离子的参考数目 Reference count of anions in A
```

```
cationsA = -1 ; ... of cations in A
anionsB = -1 ; ... of anions in B
cationsB = -1 ; ... of cations in B
```

这些选项设定两侧需要的阴离子数与阳离子数. `-1` 意味着它们的数目由0时间步的值决定, 并保持不变, 注意, 这些数目应加到互换组的总离子数中.

注意, CompEL 模拟的双层体系, 可以很容易地使用下列方法创建. 利用 `gmx editconf -translate 0 0 <l_z>` 命令沿膜的法线方法(通常为 z)重复已有的膜/通道MD体系, 其中 `l_z` 为重复方向的盒子长度. 如果你已经定义了单层体系通道的索引组, 使用 `gmx make_ndx -n index.ndx -twin` 可得到双层体系的索引组.

为抑制膜沿互换方向的大的涨落, 采用伞形牵引(参见6.4节)在两个通道和/或双层中心之间施加简谐势(仅作用于交换方向), 可能会有帮助.

多通道

如果划分组包含的分子数大于1, 必须正确地选择所有分子相对于彼此的PBC映象, 这样才能正确地确定通道中心.

GROMACS假定 `.tpr` 文件中初始结构的PBC表示是正确的. 设定下面的环境变量可检查是否正确

- `GMX_COMPELDUMP`: 分子完整后将它的初始结构输出到 `.pdb` 文件.

6.7 使用自由能代码计算PMF

自由能耦合参数方法(参见3.12节)提供了几种计算平均力势的方式. 通过使用简谐势或约束连接两个原子, 可计算它们之间的平均力势. 为达到这个目的, 一些特殊的势能函数可避免产生额外的排除, 参见5.4节. 当状态B中的最小或约束长度比状态A中的大1 nm时, 限制力或约束力为 $\partial H / \partial \lambda$. 通过设置 `.mdp` 文件中的 `delta-lambda` 可改变原子间的距离, 它是 λ 和时间的函数. 所得到的结果应当与使用伞形抽样或约束抽样得到的结果完全相同(尽管由于实现方法不同数值上不可能完全相同). 与牵引代码不同, 自由能代码也可用于处理约束连接的原子.

也可利用位置限制来计算平均力势. 采用位置限制时, 原子连接到空间中的某一位置, 二者之间具有简谐势(参见4.3.1节). 这些位置可以是耦合参数 λ 的函数. 可利用 `grompp` 的 `-r` 和 `-rb` 选项分别设置A状态和B状态的位置. 可以利用这种方法进行靶向MD. 注意, 我们并不鼓励对蛋白质使用靶向MD. 将这些构象作为状态A和B的位置限制坐标, 就可以强制蛋白质从一个构型转换到另一个. 你可以在0和1之间缓慢地改变 λ 的值. 这种方法的主要缺点在于, 蛋白质的构型自由度受到位置限制的严重制约, 而与从状态A到B的变化无关. 此外, 蛋白质由状态A到B的强制转变几乎处于一条直线上, 而实际的路径可能大为不同. 更适合这种方法的体系是固体体系或限制于边界或墙之间的液体. 你可以用这种方法来测量改变与边界或墙之间的距离时所需要的力. 因为边界(或墙)必须要固定. 位置限制并不会限制体系的抽样.

6.8 移除最快的自由度

MD模拟的最大时间步长由模拟体系中的最小振动周期决定. 键伸缩振动处于量子力学基态, 因此利用约束进行描述比简谐势更好.

对其余的自由度, 最短振动周期(模拟所得)是13 fs, 涉及氢原子键角的振动. 作为一项准则, 使用Verlet(蛙跳式)积分

方案时, 对一个简谐振子, 每个周期内至少需要执行5次数值积分步才能达到合理的精度, 相应的最大时间步长约为3 fs. 忽略这些非常快的周期为13 fs的振动, 下一个最短的周期约为20 fs, 相应的最大时间步长约为4 fs.

从氢原子中移除键角自由度的最好方法是将它们定义为虚拟作用位点而不是常规原子. 常规原子通过键, 键角, 二面角连接到分子, 而虚拟位点的位置则根据已定义好的方式, 由三个邻近重原子的位置计算得到(参看4.7节). 对水, 羟基, 疏基或氨基中的氢原子, 不能移除其自由度, 因为需要保留旋转自由度. 除此之外, 唯一可降低这些快速运动的方法是增加氢原子的质量, 但同时也需要增加与氢原子相连的重原子的质量. 这会增大水分子, 羟基, 疏基或氨基的转动惯量, 但不会影响体系的平衡性质, 对体系动力学性质的影响也不会太大. 这些方法将在6.8.1节进行简短的讨论, 以前的论文中有详细的讨论[136].

使用虚拟位点并同时修改质量, 下一瓶颈可能是由不当二面角(用于维持分子基团的平面性或手性)和肽的二面角引起. 不可能更改肽的二面角而不影响蛋白质的物理行为. 维持平面性的不当二面角主要用于处理芳香残基, 这些残基中的键, 键角, 二面角也可以替换为稍复杂点的虚拟位点.

本节所讨论的所有修改都可以使用GROMACS的拓扑创建工具 `pdb2gmx` 来完成, 其中有单独的选项用于增加氢原子的质量, 虚拟化所有氢原子或虚拟化所有芳香残基. 注意, 当全部氢原子都被虚拟化后, 那些处于芳香残基内部的氢原子也将被虚拟化, 即, 对芳香残基中氢原子的处理各不相同, 而是取决于对芳香残基的处理.

当处理拓扑文件时, `grompp` 会直接从力场参数(键长和键角)中获取氢原子虚拟位点的构建参数. 对芳香残基, 这些构建方法基于力场中结构的键长, 键角, 但这些参数是直接写入 `pdb2gmx` 源代码中的, 因为对整个芳香基团的构建非常复杂.

6.8.1 氢原子的键-键角振动

构建虚拟位点

将氢原子定义为虚拟位点的目的在于去除所有的高频自由度. 在一些情况下, 不应该去除所有的氢原子自由度, 例如对羟基或氨基应当保留氢原子的旋转自由度. 必须注意构建虚拟位点时不能引入不需要的相关, 例如构建原子间的键-键角振动可能会转变为氢原子的键振动. 另外, 由于虚拟位点没有质量, 根据定义, 为了维持体系的总质量守恒, 被视为虚拟位点的氢原子的质量应当加到与其相连接的重原子上.

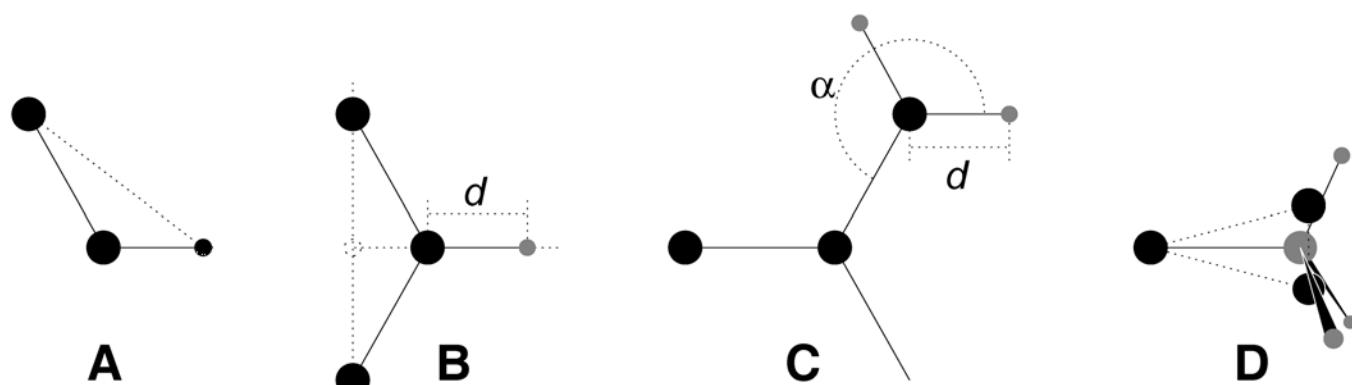


图6.7: 氢原子不同类型虚拟位点的构建方式. 黑点为用于构建虚拟位点的原子, 灰点为虚拟位点, 氢原子小于重原子. A. 固定的键角, 注意这种情况下氢原子并不是虚拟位点; B. 位于三原子平面内, 距离固定; C. 位于三原子平面内, 键角与距离固定; D. 氨基的构建方法(-NH_2 或 -NH_3^+), 详情见下文.

考虑到这些, 蛋白质中的氢原子可以自然地分为几类, 每一类需要不同的构建方法(参见图6.7).

- 羟基(-OH)或巯基(-SH)氢原子:** 羟基中唯一可约束的内部自由度为C-O-H键角。通过定义一个额外的具有适当长度的键可对此键角进行固定，见图6.7 A. 这样可以去除键角弯曲的高频振动，同时保留了二面角的旋转自由度。对巯基情况相同。注意，在这种情况下氢原子并不被视为虚拟位点。
- 单个氨基或酰胺(-NH-)和芳香氢原子(-CH-):** 不能利用键矢量的线性组合来构建这些氢原子的位置，因为重原子之间具有柔性的键角。作为替代，可将氢原子置于与重原子距离固定的位置，位于通过成键重原子与两个第二键合原子连线上一点的直线上，见图6.7 B.
- 平面氨基(-NH₂)氢原子:** 用于单个酰胺氢原子的方法用于平面氨基时效果不好，因为找不到两个合适的重原子来定义氢原子的方向。作为替代，氢原子被置于距N原子一定距离的位置，与C原子所成的键角固定，处于其他重原子定义的平面上，键图6.7 C.
- 氨基(伞形 -NH₂或-NH₃⁺)氢原子:** 不能由与其相连的重原子来构建具有旋转自由度的氨基氢原子的虚拟位点，因为这样做会失去氨基的旋转自由度。为保留旋转自由度同时去除氢原子的键-键角自由度，可构建两个“哑质点”，它们具有与氨基相同的总质量，转动惯量(绕C-N键转动)和质心。这些哑质点原子与任何其他原子都不存在相互作用，它们会彼此相连，且与碳原子相连，形成刚性三角形。由这三个粒子，可根据两个碳原子的矢量及其外积的线性组合来构建N和氢原子的位置，这样得到的氨基具有旋转自由度，但不含其他内部自由度，见图6.7 D.

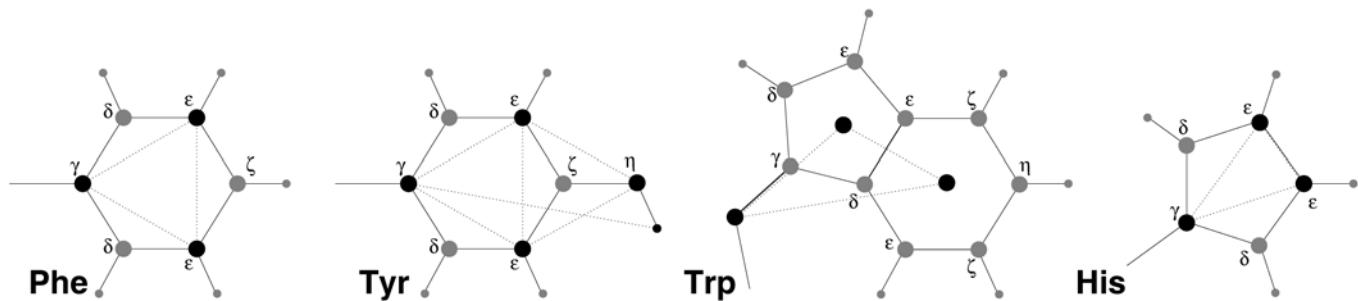


图6.8: 芳香残基不同类型虚拟位点的构建方法。黑点为用于构建虚拟位点的原子，灰点为虚拟位点，氢原子小于重原子。A: 苯丙氨酸; B: 络氨酸(注意氢原子并不是虚拟位点); C: 色氨酸; D: 组氨酸

6.8.2 芳香基团的面外振动

芳香残基支链中的平面结构致使它自身就是一个完美的虚拟位点的构建方式，具有完美的平面基团，没有本质上不稳定的，用于维持正常原子处于平面内所需要的约束。构建的基本方法是定义三个原子或哑质点，利用三者相互之间的约束来固定几何结构，并将其余原子作为简单的虚拟位点类型(参见4.7节)。不同的芳香残基需要不同的方法：

- 苯丙氨酸:** C_γ, C_{ε1}, C_{ε2} 作为正常原子，但每个的质量为苯基的1/3，见图6.8 A.
- 络氨酸:** 对苯环的处理与苯丙氨酸相同。另外，在 C_{ε1}, C_{ε2} 和 C_η 之间定义约束。原始的不当二面角可将两个三角形(一个为苯环的，一个与 C_η 有关)维持在平面内，但由于转动惯量更大，约束更稳定。羟基中的键-键角使用 C_γ 和 C_η 之间的约束进行约束。注意，氢原子并不视为虚拟位点进行处理，见图6.8 B.
- 色氨酸:** 将 C_β 视为常规原子，在每个环的质心创建两个哑质点，每个的质量等于相应环的总质量(C_{δ2} 和 C_{ε2} 对每个环的贡献为1/2)。这样做可以保持总质量和转动惯量几乎等于原本的值，见图6.8 C.
- 组氨酸:** 将 C_γ, C_{ε1}, N_{ε2} 视为正常原子，但其质量会重新分配，这样环的质心保持不变，见图6.8 D.

6.9 粘度计算

剪切粘度是液体的一种性质, 实验上很容易测量. 它可用于力场的参数化, 因为它是一个动力学性质, 而用于力场参数化的大多数其他性质都是热力学性质. 粘度也是一种重要的性质, 因为它会影响液体中溶剂化分子的构象变化的速率.

利用爱因斯坦关系式可以从平衡模拟计算粘度:

$$\eta = \frac{1}{2} \frac{V}{k_B T} \lim_{x \rightarrow \infty} \frac{d}{dt} \left\langle \left(\int_{t_0}^{t_0+t} P_{xz}(t') dt' \right)^2 \right\rangle_{t_0} \quad (6.61)$$

这可以利用 `g_energy` 进行计算. 这种方法收敛十分缓慢[137], 用于确定粘度的精确值时, 即便是纳秒尺度的模拟也可能不够长. 此外, 计算结果非常依赖于对静电的处理. 使用(短的)截断会导致压力的非对角元素出现很大噪声, 可能使计算的粘度值增大一个数量级.

GROMACS也提供了计算粘度的非平衡方法[137]. 其原理基于以下事实, 通过外力输入体系的能量会通过粘性摩擦而耗散, 产生的热会通过耦合的热浴移除. 对牛顿流体, 施加一个小的力会导致速度梯度, 其方程如下:

$$a_x(z) + \frac{\eta}{\rho} \frac{\partial^2 v_x(z)}{\partial z^2} = 0 \quad (6.62)$$

这里我们在 x 方向施加了加速度 $a_x(z)$, 它是坐标 z 的函数. 在GROMACS中加速的剖面为:

$$a_x(z) = A \cos\left(\frac{2\pi z}{l_z}\right) \quad (6.63)$$

其中为 l_z 为盒子的高度. 产生的速度剖面为:

$$v_x(z) = V \cos\left(\frac{2\pi z}{l_z}\right) \quad (6.64)$$

$$V = A \frac{\rho}{\eta} \left(\frac{l_z}{2\pi}\right)^2 \quad (6.65)$$

可利用 A 和 V 来计算粘度:

$$\eta = \frac{A}{V} \rho \left(\frac{l_z}{2\pi}\right)^2 \quad (6.66)$$

模拟中 V 的定义为

$$V = \frac{\sum_{i=1}^N m_i v_{i,x}^2 \cos\left(\frac{2\pi z}{l_z}\right)}{\sum_{i=1}^N m_i} \quad (6.67)$$

产生的速度剖面并不耦合到热浴. 此外计算体系动能时会排除速度剖面. 你可能希望使用尽可能大的 V 以得到好的统计, 但剪切速率不能太高, 以免使体系过远地远离平衡态. 当余弦为零时, 剪切速率最大,

$$\text{sh}_{\max} = \max_z \left| \frac{\partial v_x(z)}{\partial z} \right| = A \frac{\rho}{\eta} \frac{l_z}{2\pi} \quad (6.68)$$

对 $\eta = 10^{-3}$ [kg m⁻¹ s⁻¹], $\rho = 10^3$ [kg m⁻³], $l_z = 2\pi$ [nm] 的模拟, $\text{sh}_{\max} = 1$ [ps nm⁻¹] A . 这个剪切速率应小于体系中最长相关时间的倒数. 对大多数液体, 转动的相关时间最长, 大约为 10 ps. 对上面的情况, A 应小于 0.1 [nm ps⁻²]. 若剪切速率过高, 观察到的粘度会过低. 由于 V 正比于盒子高度的平方, 盒子最好沿 z 轴方向延伸. 一般而言, 要得到粘度的精确值, 100 ps 的模拟长度足够了.

由粘性摩擦产生的热会被耦合热浴移除. 由于这种耦合并不是瞬时的, 液体的真正温度会稍稍低于观察到的温度. Berendsen 推导了这种温度偏移[30], 它与剪切速率的关系为:

$$T_s = \frac{\eta\tau}{2\rho C_v} \text{sh}_{\max}^2 \quad (6.69)$$

其中 τ 为 Berendsen 控温器的耦合时间, C_v 为热容. 使用上面例子中的值, $\tau = 10^{-13}$ [s], $C_v = 2 \cdot 10^3$ [J kg⁻¹K⁻¹], 可得到 $T_s = 25$ [K ps⁻²] sh_{\max}^2 , 如果需要的剪切速率小于 1/10 [ps⁻¹], T_s 小于 0.25 [K], 可忽略不计.

注意, 当从平衡态开始时, 体系必须要构建速度剖面, 构造所需时间的数量级与液体的相关时间一致.

根据方程 6.66 得到的 V 和 $1/\eta$ 这两个量及其平均值与涨落会写入到能量文件.

6.10 表格相互作用函数

6.10.1 势能三次样条插值

在 GROMACS 的一些内部循环中, 计算势能与力时会使用查表的方法, 并利用三次样条算法对这些表格进行插值. 对静电, 色散, 排斥相互作用会使用独立的表格, 但为了提高缓存性能, 这些项被合并到一个单一的数组中. 对 $x_i \leq x < x_{i+1}$ 的三次样条插值为:

$$V_s(x) = A_0 + A_1\epsilon + A_2\epsilon^2 + A_3\epsilon^3 \quad (6.70)$$

其中表格间距 h 与分数 ϵ 为:

$$h = x_{i+1} - x_i \quad (6.71)$$

$$\epsilon = (x - x_i)/h \quad (6.72)$$

因此 $0 \leq \epsilon < 1$. 由此, 我们可以计算导数得到力:

$$-V'_s(x) = -\frac{dV_s(x)}{d\epsilon} \frac{d\epsilon}{dx} = -(A_1 + 2A_2\epsilon + 3A_3\epsilon^2)/h \quad (6.73)$$

四个系数可以从四个条件得到: V_s 和 $-V'_s$ 在每个区间的两端应精确地等于势能 V 与力 $-V'$ 的值. 这样得到了每个区间上的误差

$$|V_s - V|_{\max} = V''' \frac{h^4}{384} + O(h^5) \quad (6.74)$$

$$|V'_s - V'|_{\max} = V''' \frac{h^3}{72\sqrt{3}} + O(h^4) \quad (6.75)$$

$$|V''_s - V''|_{\max} = V''' \frac{h^2}{12} + O(h^3) \quad (6.76)$$

V 和 V' 连续, 但 V'' 一阶不连续. 对混合精度与双精度的GROMACS, 每纳米的点数分别为500和2000. 这意味着势能与力的误差通常小于混合精度的精确度.

GROMACS会存储 A_0 , A_1 , A_2 和 A_3 . 计算力的子程序会获得一个具有这些参数与缩放因子 s 的表格, s 等于每nm的点数(注意 h 的单位为 s^{-1}). 算法的过程大致如下:

1. 计算距离向量(\mathbf{r}_{ij})与距离 r_{ij}
2. 将 r_{ij} 乘上 s , 截断到整数值 n_0 以获得表格中的索引号
3. 计算分数部分($\epsilon = sr_{ij} - n_0$)与 ϵ^2
4. 利用插值计算势能 V 与标量力 f
5. 将 f 乘上 \mathbf{r}_{ij} , 得到矢量力 \mathbf{F}

注意 查表明显慢于大部分Lennard-Jones 与库仑相互作用的计算. 然而, 它比PPPM方法中使用的移位库仑函数要快很多. 最后, 修改一个势能的表格(得到它的图形表示)要比修改MD程序中的内部循环简单得多.

6.10.2 用户自定义势能函数

你也可以使用自己的势能函数而无需编辑GROMACS的源代码. 势能函数应符合下面的方程

$$V(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0} f(r_{ij}) + C_6 g(r_{ij}) + C_{12} h(r_{ij}) \quad (6.77)$$

其中 f , g , h 为用户自定义的函数. 注意 若 g 代表正常的色散相互作用, $g(r)$ 应 <0. C_6 , C_{12} 和电荷由拓扑读入. 也要注意组合规则只支持Lennard-Jones 与Buckingham 势能函数, 你的表格应匹配二进制拓扑中的参数.

当在你的 `.mdp` 文件中添加下面的行时,

```
rlist      = 1.0
coulombtype = User
rcoulomb   = 1.0
vdwtype    = User
rvdw       = 1.0
```

`mdrun` 将将读入一个非键相互作用表格文件, 若设置了 `energygrp-table` 则将读入多个非键相互作用文件(参见下面的说明). 文件的名称可使用 `mdrun` 的选项 `-table` 设置. 表格文件应包含7列用于查询的数据, 依次为: x , $f(x)$, $-f'(x)$, $g(x)$, $-g'(x)$, $h(x)$, $-f'(x)$. x 的范围从0到 $r_c + 1$ (`table_extension` 的值可在 `.mdp` 文件中更改). 可以选择你喜欢的间距. 对标准表格, GROMACS对混合精度与双精度分别使用了0.002与0.0005 nm. 在这里, r_c 表

示两个截断 `rvdw` 和 `rcoulomb` 的最大值(参见上文). 这些变量不需要相同(也不需要为1.0). 一些用作势能的函数在 $x = 0$ 处存在奇点, 但由于原子彼此之间的距离通常不会小于0.1 nm, $x = 0$ 处的函数值并不重要. 最后, 也可以将标准的库仑和修改的LJ势组合在一起(反之亦然), 只须指定, 例如 `coulombtype = Cut-off` 或 `coulombtype = PME`, `vdwtype = User`. 但表格文件必须始终包含7列, 并且所有列中必须为有意义的数据(不能为零). 你可以在 `GMXLIB` 目录中找到一些创建好的表格文件, 它们可用于6–8, 6–9, 6–10, 6–11和6–12 Lennard-Jones 势, 并使用了正常的库仑相互作用.

如果想对不同原子组使用不同的函数形式, 可利用能量组进行设置. 利用 `.mdp` 的 `energygrp-table` 选项(参见7.3节)可以对不同的两个能量组之间的非键相互作用使用不同的表格. 具有不同相互作用势的原子应放于不同能量组中. 对于列于 `energygrp-table` 中的能量组对, 会使用正常的表格. 当需要对许多原子类型使用不同的函数形式时, 这种作法更简单一些,

6.11 混合量子经典模拟技术

在分子力学(MM)力场中电子的影响是利用经验参数进行描述的. 这些参数是基于实验数据得到的, 或是基于高水平的量子化学计算得到的, 对给定共价结构的基态适用. MM近似对基态过程通常足够精确, 在这些过程中原子间所有的键合连接都不会改变. 然而对那些键合连接会改变的过程, 例如化学反应过程或涉及多个量子态的过程, 如光化学转换, 不可忽略电子, 因此需要利用量子力学描述体系, 或是至少需要对体系中反应的那部分用量子力学进行描述.

模拟溶液中或酶中化学反应的一种方法是将量子力学(QM)与分子力学(MM)组合起来, 反应部分使用量子力学方法描述, 同时其余部分以力场描述. 目前的GROMACS版本中提供了几个流行量子化学程序的接口(MOPAC[138], GAMESS-UK[139], Gaussian[140] and CPMD[141]).

在GROMACS中两个体系之间的相互作用可以Field等人的方法[142]或Morokuma及其合作者的QM/MM方法[143, 144]进行描述.

6.11.1 概述

在本版本的GROMACS中可采用两种方法描述QM和MM两子体系之间的相互作用:

1. 电子嵌入 QM子体系的哈密顿量包含了QM区域的电子与MM原子之间, QM原子核与MM原子之间的静电相互作用:

$$H^{QM/MM} = H_e^{QM} - \sum_i^n \sum_J^M \frac{e^2 Q_J}{4\pi\epsilon_0 r_{iJ}} + \sum_A^N \sum_J^M \frac{e^2 Z_A Q_J}{e\pi\epsilon_0 R_{AJ}} \quad (6.78)$$

其中 n 和 N 分别为QM区域中的电子数与原子核数, M 为带电的MM原子数. 上式右边的第一项为孤立QM体系原始的哈密顿量. 两个加和项的第一个为QM电子与MM原子之间总的静电相互作用, 第二个为QM原子核与MM原子之间总的静电相互作用. QM原子与MM原子之间的键合相互作用在MM水平以合适的力场项进行描述. 连接两个子体系之间的化学键使用氢原子进行封端, 以保证QM区域的原子价层完整. 这类原子上的力只出现于QM区域, 但会分配到成键的两个原子上. 封端原子通常被称为链接原子.

2. ONIOM 在这种方法中, 首先会以需要的从头算水平计算孤立QM子体系的能量与梯度, 接下来会使用MM

力场方法计算包含QM区域在内的整个体系的能量与力, 并将其加到前一步计算得到的能量与力中. 最后, 因为对QM区域内的相互作用考虑了两次, 为进行校正, 会使用分子力学对孤立的QM子体系进行计算, 并从上一步的结果中减去所得的能量与力. 总的QM/MM能量的计算表达式如下(梯度类似):

$$E_{tot} = E_I^{QM} + E_{I+II}^{MM} - E_I^{MM} \quad (6.79)$$

其中下标I和II分别代表QM和MM子体系. 上标代表了计算能量的水平. ONIOM方案的优点在于它并不限于两层QM/MM描述, 而是很容易推广到多层, 每一层都可以使用不同水平的方法进行描述.

6.11.2 使用方法

要使用GROMACS的QM/MM功能, 你需要:

1. 如果需要, 引入QM/MM边界上的链接原子;
2. 指定哪些原子需要在QM水平进行处理;
3. 指定QM水平, 基组, QM界面的类型等等.

添加链接原子

对连接QM和MM子体系的键需要引入链接原子. 在GROMACS中, 链接原子具有特殊的原子类型, LA. 在QM计算中, LA原子类型被当作氢原子进行处理, 在力场计算中则被视为虚拟位点. 如果存在链接原子, 它们是体系的一部分, 但与其他任何原子都没有相互作用, 除了作用于其上的力会分配到成键的两个原子上. 在拓扑中, 链接原子(LA)因此被定义为虚拟位点原子.

```
[ virtual_sites2 ]
LA QMatom MMatom 1 0.65
```

关于虚拟位点的处理细节请参看5.2.2节. 在模拟的每一步, 链接原子会被取代.

另外, 键本身也会被约束代替:

```
[ constraints ]
QMatom MMatom 2 0.153
```

注意, 因为在我们的体系中, QM/MM键是碳碳键(0.153 nm), 我们使用的约束长度为0.153 nm, 咂位置为0.65. 后者为理想碳氢键长与理想碳碳键长之间的比例. 采用这一比例, 链接原子始终距离QMatom 0.1 nm, 与碳氢键长相符. 如果QM和MM子体系由不同类型的键相连接, 需要使用与键类型匹配的不同约束与呀位置.

指定QM原子

需要在QM水平处理的原子, 包括链接原子, 要添加到索引文件中. 此外, QM区域中原子之间的化学键需要在拓扑文件中定义为连接键(键类型5):

```
[ bonds ]
QMatom1 QMatom2 5
QMatom2 QMatom3 5
```

指定QM/MM模拟参数

在 `.mdp` 文件中下列参数控制QM/MM模拟的运行.

`QMMM = no`

如果设为 `yes`, 进行QM/MM模拟. 可以使用不同水平的QM单独对几个组进行描述, 在 `QMMM-grps` 域中指定这些组, 彼此之间以空格隔开. 各个组使用的从头算方法的水平在 `QMmethod` 和 `QMbasis` 域中指定. 使用不同水平的方法对组进行描述只能与ONIOM QM/MM一起使用, 由 `QMMSscheme` 指定.

`QMMM-grps =`

使用QM水平描述的组

`QMMSscheme = normal`

可用选项为 `normal` 和 `ONIOM`, 用于选择QM/MM接口. `normal` 意味着QM子体系电子化的嵌入MM子体系中, 只能对一个 `QMMM-grps` 使用从头算方法进行描述, 方法的水平通过 `QMmethod` 和 `QMbasis` 指定. 体系的其余部分处于MM水平. QM和MM两个子体系的相互作用如下: MM部分的点电荷包含在QM部分的单电子哈密顿量中, 所有的Lennard-Jones 相互作用都在MM水平进行描述. 如果选择 `ONIOM`, 使用Morokuma 及其同事发展的ONIOM方法对子体系之间的相互作用进行描述. 可以有一个以上的 `QMMM-grps`, 每个组可以使用不同级别的QM(`QMmethod` 和 `QMbasis`)进行描述.

`QMmethod =`

用于计算QM原子的能量和梯度的方法. 可用的方法包括AM1, PM3, RHF, UHF, DFT, B3LYP, MP2, CASSCF, MMVB和CPMD. 对CASSCF, 电子数和活化空间的轨道数分别由 `CASelectrons` 和 `CASorbitals` 指定. 对CPMD, 平面波截断由关键词 `planewavecutoff` 指定.

`QMbasis =`

用于展开电子波函数的高斯基组. 目前只可使用高斯基组, 即STO-3G, 3-21G, 3-21G, 3-21+G, 6-21G, 6-31G, 6-31G, 6-31+G和6-311G. CPMD使用平面波展开而不是以原子为中心的基函数, `planewavecutoff` 关键词控制平面波展开.

`QMcharge =`

`QMMM-grps` 的总电荷数, 以 `e` 为单位. 在有一个以上 `QMMM-grps` 的情况下, 需要单独指定每个ONIOM层的总电荷.

`QMmult =`

`QMMM-grps` 的多重度. 在有一个以上 `QMMM-grps` 的情况下, 需要单独指定每个ONIOM层的多重度.

`CASorbitals =`

进行CASSCF计算时包含在活化空间中的轨道数.

CASElectrons =

进行CASSCF计算时包含在活化空间中的电子数.

SH = no

若设为 yes, 在激发态势能面进行QM/MM的MD模拟, 在模拟过程中, 当体系碰到锥形交叉线时, 强制非绝热跳跃到基态. 此选项只能与CASSCF方法联合使用.

6.11.3 输出

QM计算所得的能量与力会加到GROMACS计算的能量与力中, 在 .edr 文件中会有总QM能量的选择项.

6.11.4 未来发展

为增加QM/MM接口的精度, 一些功能目前正在开发中. 其中的一个功能是在QM计算中使用离域化的MM电荷. 使用这种模糊电荷的最大好处是库仑势在原子间距离时具有有限值. 在点电荷表示中, 靠近QM区域的部分带电的MM原子倾向于“过极化”QM体系, 在计算中会导致假象.

此外, 还需要发展一个过渡态优化方法.

6.12 自适应分辨率方案

自适应分辨率方案(AdResS)[145, 146] 采用力插值方法将不同分辨率的两个部分耦合起来. 与前一节的混合经典量子模拟技术相反, 在AdResS 中, 高分辨粒子的数目不固定, 可随着模拟时间而变化..

下面我们讨论的AdResS 对相同体系采用双分辨描述(原子与粗粒), 示意图见图6.9. 本节的详细实现细节已发表于 [147, 148].

每个分子需要一个定义良好的映射点(通常为质心), 但也可以使用粒子坐标的任何其他线性组合. 在拓扑中映射点被定义为虚拟位点. 粗粒化区域的力只是映射点位置的函数. 在这个实现中, 利用电荷组或电荷组的集合对分子进行建模, 这实际上允许一个分子有多个映射点. 对更大的分子, 如聚合物, 这种方法可能会有帮助. 在那种情况下, 必须扩展AdResS 以描述键合相互作用[149]. 将来的GROMACS版本中会实现这种功能.

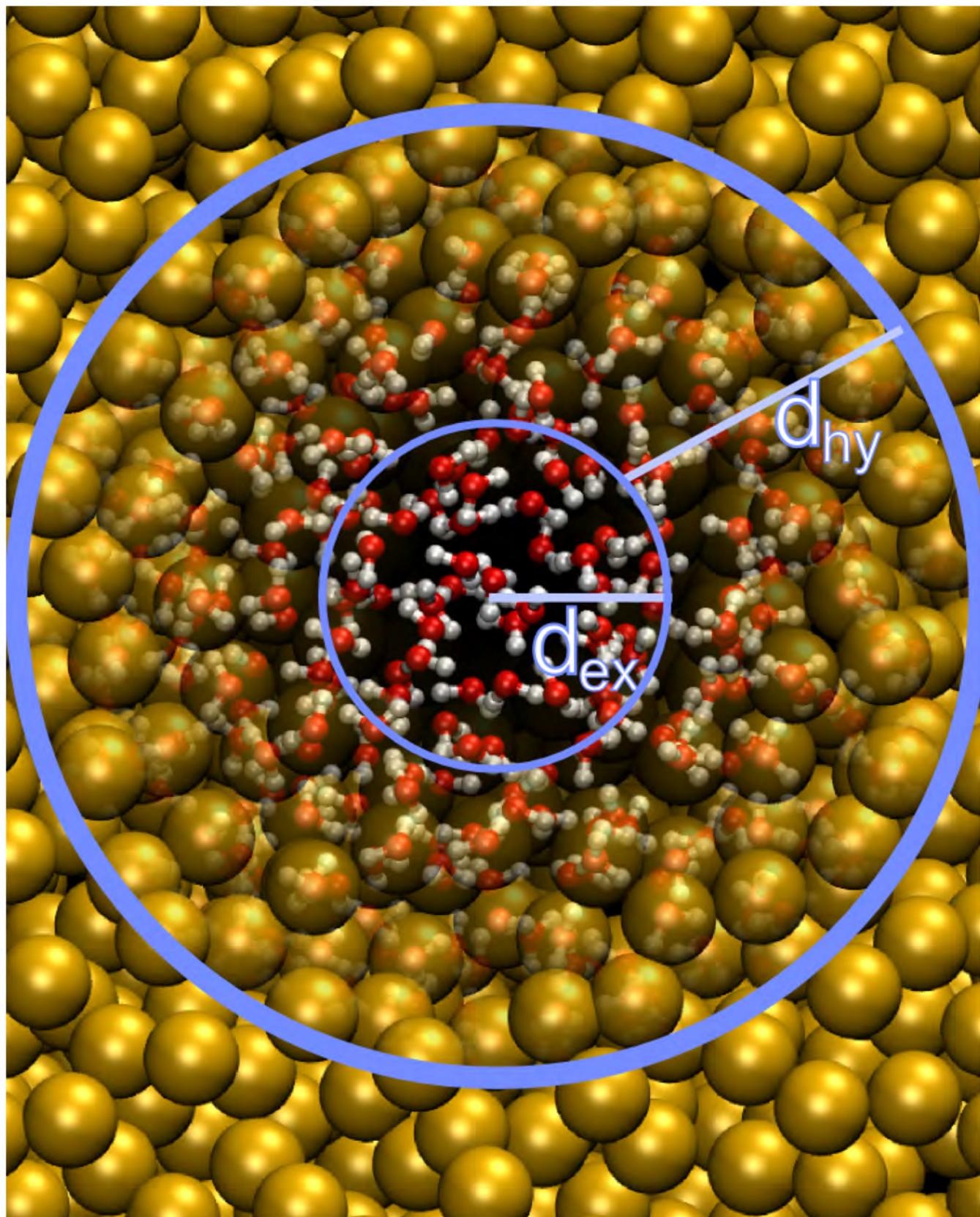


图6.9: 水分子AdResS方法的示意图

两个分子间的力为[145][1]:

$$\vec{F}_{\alpha\beta} = w_\alpha w_\beta \vec{F}_{\alpha\beta}^{\text{ex,mol}} + [1 - w_\alpha w_\beta] \vec{F}_{\alpha\beta}^{\text{cg,mol}} \quad (6.80)$$

其中 α 和 β 标记两个分子, w_α , w_β 为两个分子的自适应权重.

上式第一部分代表了分子显式的相互作用, 可写为

$$\vec{F}_{\alpha\beta}^{\text{ex,mol}} = \sum_{i \in \alpha} \sum_{j \in \beta} \vec{F}_{ij}^{\text{ex}} \quad (6.81)$$

其中 \vec{F}_{ij}^{ex} 为第 α 个分子中的第 i 个原子与第 β 个分子中第 j 个原子之间的力, 由显式的力场给出. 方程6.80的第二部分来自分子的粗粒化相互作用. GROMACS中的实现稍有扩展:

$$\vec{F}_{\alpha\beta} = \sum_{i \in \alpha} \sum_{j \in \beta} w_i w_j \vec{F}_{ij}^{\text{ex}} + [1 - w_\alpha w_\beta] \vec{F}_{\alpha\beta}^{\text{cg,mol}} \quad (6.82)$$

其中 w_i 和 w_j 为基于原子的权重, 由 **adress-site** 选项决定. 若 **adress-site** 为质心, 原子 i 具有其电荷组质量中心的权重, 分子 α 的权重 w_α 由粗粒化粒子的位置决定, 而粗粒化粒子由拓扑中指定的原子化粒子的虚拟位点构建. 这种推广可以实现各种AdResS方法, 同时普通的方法可利用拓扑中的质心虚拟位点, 设定 **adress-site=COM** 并把一个分子的所有原子(除代表粗粒化相互作用的虚拟位点外)放于同一电荷组来实现. 对大的分子, 使用基于原子的权重有时更好, 这可利用设置 **adress-site=atomperatom** 或将每个原子置于独立的电荷组(一个原子的电荷组质心为其自身)来实现.

粗粒化力场 \vec{F}^{cg} 通常利用基于结构的粗粒化方法由原子化体系导出(参见4.10.5节). 为指定哪些原子属于一个粗粒化表示, 可使用能量组. 每个粗粒化相互作用必须与单独的能量组相联系, 这就是代表粗粒相互作用的虚拟位点也必须处于不同电荷组的原因. 被视为粗粒化相互作用进行处理的电荷组, 列于 **adress_cg_grp_names**. 插值的最重要内容(参见方程6.80和方程6.82)是自适应权重函数(示意图见图6.9):

$$w(x) = \begin{cases} 1 & : \text{原子化/显式区域 atomistic/explicit region} \\ 0 < w < 1 & : \text{杂合区域 hydrid region} \\ 0 & : \text{粗粒化区域 coarse-grained region} \end{cases} \quad (6.83)$$

它的值处于0和1之间. w 的定义在显式区域是纯的显式力, 在粗粒化区域则是单纯的粗粒化力, 因此方程6.80中基本上只在杂合区域具有混合的相互作用, 这在标准模拟中是不会出现的. 在GROMACS中, 类似于 \cos^2 函数被用于权重函数:

$$w(x) = \begin{cases} 0 & : x > d_{\text{ex}} + d_{\text{hy}} \\ \cos^2 \left(\frac{\pi}{2d_{\text{hy}}} (x - d_{\text{ex}}) \right) & : d_{\text{ex}} + d_{\text{hy}} > x > d_{\text{ex}} \\ 1 & : d_{\text{ex}} > x \end{cases} \quad (6.84)$$

其中 d_{ex} 和 d_{hy} 分别为显式与杂合区域的大小. 根据研究体系的物理特点, 你也可以使用其他函数, 只要它们满足下面的边界条件: 1) 连续, 2) 单调, 3) 边界处导数为零. 已经实现了对模拟盒子的球形划分和一维划分(**adress-type** 选项), 取决于此选项, 到显式中心的距离 x 的计算方法为:

$$x = \begin{cases} |(\vec{R}_\alpha - \vec{R}_{\text{ct}}) \cdot \hat{e}| & : \text{沿} \hat{e} \text{方向划分 splitting in } \hat{e} \text{ direction} \\ |\vec{R}_\alpha - \vec{R}_{\text{ct}}| & : \text{球形划分 spherical splitting} \end{cases} \quad (6.85)$$

其中 R_{ct} 为显式区域的中心(由 **adress-reference-coords** 选项定义), \vec{R}_α 为第 α 个分子的映射点. 对质心映射,

$$R_\alpha = \frac{\sum_{i \in \alpha} m_i r_i}{\sum_{i \in \alpha} m_i} \quad (6.86)$$

注意, 权重函数的值唯一地取决于分子的映射. 力的插值(见方程6.82)可导致密度的非均匀性并影响杂合区域的体系结构.

减小密度不均匀性的一种方法是使用所谓的热力学力(TF, thermodynamic force)[151]. 这个力包含了空间相关的外场, 施加到杂合区域中每个分子的粗粒化位点上. 可以为体系中的每个物种指定这种力. TF补偿了在均相密度剖面中出现的压力剖面[152], 因此可以校正杂合区域中的局部密度不均匀性, 并允许原子化和粗粒化的部分之间相互耦合, 它们在构建时在目标密度具有不同的压力. 场的强度可以使用迭代过程确定, 详细说明见VOTCA软件包的[手册](#)[122]. 将 `adress-interface-correction` 设为 `thermoforce` 可启用TF校正, `adress-tf-grp-names` 定义了校正作用的能量值.

6.12.1 例: 水的自适应分辨率模拟

本节将解释说明水的自适应分辨率模拟的设置, 模拟时原子化的SPC水模型[181]和它的粗粒化表示进行了耦合(如[152]中所用的). 为进行模拟需要下面一些步骤:

- 执行参考的全原子模拟
- 创建粗粒化表示并保存为表格相互作用函数
- 创建SPC水的杂合拓扑
- 修改原子坐标文件以包含粗粒化表示
- 在 `grompp` 输入文件中定义分辨率模拟的构型
- 创建索引文件

相互作用粗粒化表示被存储为表格相互作用函数, 这种函数的说明见6.10.2节. 约定是使用 $C^{(12)}$ 列, 且系数 $C^{(12)}$ 设为1, 其他所有列都为零. VOTCA手册中关于如何使用各种技术对SPC水进行粗粒化有详细的说明及教程. 这里, 我们将粗粒化相互作用命名为CG, 因此相应的表格文件为 `table(CG,CG).xvg`. 要创建拓扑文件, 你可以从原子的拓扑文件开始,(例如 `share/gromacs/top/oplsaa.ff/spc.itp`). 我们在这里假定使用刚性水模型. 在VOTCA教程中文件被命名为 `hybrid_spc.itp`. 与原子化拓扑的唯一不同在于额外的粗粒化虚拟位点:

```
[ moleculetype ]
; molname      nrexcl
SOL          2

[ atoms ]
; nr   type    resnr   residue  atom   cgnr   charge   mass
  1  opls_116     1       SOL      OW      1    -0.82
  2  opls_117     1       SOL      HW1     1     0.41
  3  opls_117     1       SOL      HW2     1     0.41
  4    CG         1       SOL      CG      2      0

[ settles ]
; OW   funct    doh    dhh
  1      1      0.1    0.16330

[ exclusions ]
  1  2  3
  2  1  3
  3  1  2
```

```
[ virtual_sites3 ]
; Site from funct a d
4 1 2 3 1 0.05595E+00 0.05595E+00
```

根据虚拟位点类型3及指定的系数, 虚拟位点位于分子的质心(对大分子必须使用 `virtual sitesn`). 我们现在需要在拓扑文件中包含修改后的水模型并定义 `CG` 类型, 在 `topol.top` 中:

```
#include "ffoplsaa.itp"
[ atomtypes ]
;name      mass       charge     ptype      sigma      epsilon
CG        0.0000    0.0000      V          1          0.25

#include "hybrid_spc.itp"

[ system ]
Adaptive water

[ molecules ]
SOL 8507
```

σ 和 ϵ 的值分别对应于 $C_6 = 1$ 和 $C_{12} = 1$, 因此表格文件应该在 C_6 或 C_{12} 列包含粗粒相互作用. 在此例中我们使用了OPLS力场, 并指定了 ϵ 和 σ . 注意, 对那些直接使用 C_6 和 C_{12} 项定义原子类型的力场, 可以简单地设置 $C_6 = 0$ 和 $C_{12} = 0$. 关于表格相互作用的更多细节请参考6.10.2节. 由于水分子具有一个虚拟位点, 坐标文件也需要包括虚拟位点.

```
adaptive water coordinates
34028
 1SOL    OW    1  0.283  0.886  0.647
 1SOL    HW1   2  0.359  0.884  0.711
 1SOL    HW2   3  0.308  0.938  0.566
 1SOL    CG    4  0.289  0.889  0.646
 1SOL    OW    5  1.848  0.918  0.082
 1SOL    HW1   6  1.760  0.930  0.129
 1SOL    HW2   7  1.921  0.912  0.150
 1SOL    CG    8  1.847  0.918  0.088
(...)
```

此文件可手动创建或使用VOTCA工具 `csg_map` 的 `--hybrid` 选项创建.

在 `grompp` 输入文件中, 需要启用AdResS 功能, 并定义构型.

```
(...)
; AdResS relevant options
energygrps           = CG
energygrp_table       = CG CG
; Method for doing Van der Waals
vdw-type              = user
adress                = yes
adress_type            = xsplit
adress_ex_width       = 1.5
adress_hy_width       = 1.5
adress_interface_correction = off
adress_reference_coords = 8 0 0
adress_cg_grp_names   = CG
```

在这里我们定义了能量组 `CG`, 它包含了粗粒化的虚拟位点. 如前所述, 粗粒化相互作用通常使用表格, 这就要求将 `vdw-type` 设置为 `user`. 若对多组分体系进行粗粒化, 必须为每一个组分定义一个能量组. 注意, 定义粗粒化表示的所有能量组, 都必须列于 `adress_cg_grp_names` 中, 以便与常规能量组区分开来.

必须更新索引文件, 并添加CG组, 其中包含所有粗粒化虚拟位点, 使用GROMACS的工具 `make_ndx` 很容易完成.

6.13 VMD插件用于轨迹文件输入/输出

GROMACS可以使用已经安装的VMD插件来读写非GROMACS格式的轨迹文件. 例如, 你可以将AMBER DCD格式的轨迹文件直接提供给GROMACS工具.

要使用上述功能, VMD的版本不能低于1.8, 还需要你的系统提供dlopen 函数, 这样才能在运行时确定存在哪些插件, 构建GROMACS时是否构建了共享库. CMake将会在你的路径下查找VMD可执行程序, 根据这个路径, 或者在配置或运行时的环境变量 `VMDDIR` 来定位插件. 作为替代, 运行时可利用 `VMD_PLUGIN_PATH` 来指定插件的路径. 注意, 这些插件是二进制形式的, 格式必须匹配插件所在机器的架构.

6.14 交互式分子动力学

GROMACS支持交互式的分子动力学(IMD, interactive molecular dynamics) 协议, VMD可利用此协议控制NAMD中运行的模拟. IMD允许从VMD客户端监测正在运行的GROMACS模拟, 此外, 通过使用鼠标或力反馈设备拉扯原子, 残基或片段, 用户可以与模拟进行互动. 关于GROMACS实现及示例GROMACS IMD系统的额外的信息可以在这个[主页](#)上找到.

6.14.1 准备模拟输入

GROMACS的实现只允许与运行模拟的一部分进行传送与交互, 例如, 当没有水分子被传送或牵引的情况下. 通常使用 `.mdp` 文件的 `IMD-group` 选项指定组. 当 `IMD-group` 为空时, IMD协议被禁用, 不能通过 `mdrun` 的开关来启用. 要与整个体系进行互动, `IMD-group` 可设为 `System`. 当使用 `grompp` 时, 作为VMD输入的 `.gro` 文件会写出(`grompp` 的 `-imd` 开关).

6.14.2 启动模拟

VMD与GROMACS之间的通讯可利用TCP套接字完成, 因而可控制本地机器或远程集群上的 `mdrun`. 可利用 `mdrun` 的 `-imdport` 选项指定连接端口, 默认为8888. 如果指定的端口号为0或更小, GROMACS会自动为IMD分配一个可用的端口.

`mdrun` 客户端每 N 步接收一次VMD中的力, 并将新的位置发送到客户端. VMD允许以交互方式增加或减小通讯频率. 默认情况下, 即使没有连接IMD客户端, 模拟也会启动并运行. 这种行为可利用 `mdrun` 的 `-imdwait` 开关选项更改. 启动后, 每当客户端断开连接时, 积分会停止直至重新连接客户端. 当使用 `-imdterm` 开关时, 可以通过按下VMD中的 `stop` 按钮来终止模拟, 默认未启用此功能. 最后, 为了与模拟进行互动, (即从VMD中牵引)必须使用 `-imdpull` 选项. 因此, 当未设置 `-imdwait`, `-imdterm` 或 `-imdpull` 时, 只可以监测模拟, 而不能从VMD客户端影响模拟. 然而, 由于 IMD协议无须认证, 在不安全环境直接可连的主机上运行模拟是不明智的. TCP的安全shell 转发可用于连接不能从活

动主机直接可达的模拟。注意, `mdrun` 的IMD命令行开关默认是隐藏的, 只有使用 `gmx mdrun -h -hidden` 才能显示于帮助文本中。

6.14.3 从VMD中连接

在VMD中, 首先必须加载与IMD组对应的结构(File → New Molecule), 然后必须使用IMD的连接窗口(Extensions → Simulation → IMD Connect (NAMD))。在IMD连接窗口中, 必须指定主机名与端口, 并点按下 connect 按钮, Detach Sim 允许断开连接, 同时不中断模拟, Stop Sim 会在下一次近邻搜索前终止模拟(若 `-imdterm` 运行)。

时间步传输速率用于调整模拟与IMD客户端之间的通讯频率。当接收到新的帧时, 设置保持速率每 N 帧加载到VMD 一次而不是忽略它们。显示的能量单位是国际单位, 与NAMD不同。

-
1. 注意, 此方程服从牛顿第三定律, 而其他一些实现方案未必服从[150]. [←](#)

第七章 运行参数和程序

7.1 在线和HTML手册

本章所有的信息都可以在GROMACS数据目录下的HTML文件中找到。文件的路径取决于文件的安装位置，默认位置是`/usr/local/gromacs/share/gromacs/html/online.html`。如果你使用Linux安装包进行安装，通常可以在`/usr/share/gromacs/html/online.html`找到。你也可以使用GROMACS网站的在线手册，其网址为`http://manual.gromacs.org/current`。

此外，我们为所有程序都安装了标准的UNIX手册页。如果你用`source`命令执行了主机上GROMACS二进制目录下的`GMXRC`脚本，这些手册页应该已经出现在`MANPATH`环境变量中了，而且你应该能够通过键入如`man gmx-grompp`之类的命令进行查看。你也可以通过在命令行中使用`-h`标识（如`gmx grompp -h`）看到同样的信息，你还可以使用`gmx help grompp`之类的命令。所有程序的列表可从`gmx help`获得。

7.2 文件类型

表7.1列出了GROMACS使用的文件类型及其简短说明，你可以在HTML参考手册或其在线版本中找到每类文件更加详细的说明。如果系统的XDR库中存在配置脚本，GROMACS 1.6或更高版本可在任何架构上读取XDR格式的GROMACS文件，这些文件应该永远存放于UNIX系统下，因为它们必须要有NFS的支持。

表7.1: GROMACS的文件类型

默认名称.扩展名	类型	默认选项	说明
atomtp.atp	Asc		pdb2gm使用的原子类型文件
eiwit.brk	Asc	-f	Brookhaven资料库文件
state.cpt	xdr		检查点文件
nnniced.dat	Asc		通用数据文件
user.dlg	Asc		ngmx对话框数据
sam.edi	Asc		ED抽样输入
sam.edo	Asc		ED抽样输出
ener.edr			通用能量: edr ene
ener.edr	xdr		便携xdr格式的能量文件
ener.ene	Bin		能量文件
eiwit.ent	Asc	-f	蛋白质数据库条目
plot.eps	Asc		EPS(tm)文件
conf.esp	Asc	-c	ESPResSo格式的坐标文件
conf.g96	Asc	-c	Gromos-96格式的坐标文件
conf.gro	Asc	-c	Gromos-87格式的坐标文件
conf.gro		-c	结构: gro g96 pdb esp tpr tpb tpa
out.gro		-o	结构: gro g96 pdb esp`
polar.hdb	Asc		氢数据库

topinc.itp	Asc		拓扑结构头文件
run.log	Asc	-l	日志文件
ps.m2p	Asc		mat2ps输入文件
ss.map	Asc		矩阵数据到颜色的映射文件
ss.mat	Asc		矩阵数据文件
gromppmdp	Asc	-f	含MD参数的grompp输入文件
hessian mtx	Bin	-m	Hessian矩阵
index.ndx	Asc	-n	索引文件
hello.out	Asc	-o	通用输出文件
eiwit.pdb	Asc	-f	蛋白质数据库文件
residue.rtp	Asc		pdb2gmx使用的残基类型文件
doc.tex	Asc	-o	LaTeX文件
topol.top	Asc	-p	拓扑结构文件
topol.tpb	Bin	-s	二进制运行的输入文件
topol.tpr		-s	通用运行输入文件: tpr tpb tpa
topol.tpr		-s	结构+质量(db): tpr tpb tpa gro g96 pdb
topol.tpr	xdr	-s	便携式xdr运行输入文件
traj.trj	Bin		轨迹文件(特定机器架构下的)
traj.trr			全精度轨迹: trr trj cpt
traj.trr	xdr		便携xdr格式的轨迹
root.xpm	Asc		X PixMap兼容的矩阵文件
traj.xtc		-f	轨迹, 输入: xtc trr trj cpt gro g96 pdb
traj.xtc		-f	轨迹, 输出: xtc trr trj gro g96 pdb
traj.xtc	xdr		压缩的轨迹(便携xdr格式)
graph.xvg	Asc	-o	xvgr/xmgr文件

7.3 运行参数

7.3.1 通用参数

参数的默认值在括号中给出。列表中的第一个选项始终是默认选项。单位在方括号中给出，破折号和下划线没有区别。

有一份示例的 `.mdp` 文件，可用于开始一个正常的模拟。你也可以编辑它以满足特定需求。

7.3.2 预处理

`include`:

在你的拓扑信息中包含特定的目录。

格式: `-I/home/john/mylib -I../otherlib`

`define`:

传递给预处理器的预定义，默认不存在预定义。你可以在自己定制的拓扑文件中使用任何定义来控制选项。有以下默认可用的选项：

- **-DFLEXIBLE**

告诉 `grompp` 在拓扑信息中使用柔性的水模型而不是刚性的水模型, 对简正模式分析有用.

- **-DPOSRES**

告诉 `grompp` 在拓扑信息中包括 `posre.itp`, 用于位置限制.

7.3.3 运行控制

integrator: (尽管名字的意思是积分方法, 但此参数包括了实际上不属于积分的算法. `steep` 及其下的所有条目都属于此类)

- **md**

蛙跳式算法积分牛顿运动方程.

- **md-vv**

速度Verlet 算法积分牛顿运动方程. 对从同样轨迹的对应点开始的恒NVE模拟, 此选项将产生与 `md` 选项解析等同的轨迹, 但不是二进制完全等同. 由于动能是从整个积分步的速度计算所得, 因此会稍微过高. 这种积分方法的优点是更精确, 使用了基于Trotter 展开的可逆的Nose-Hoover 和Parrinello-Rahman 耦合积分, 以及(略过小)整步的速度输出. 这些优点的代价是额外的计算量, 特别是当使用约束和并行的额外通讯时. 注意, 对几乎所有的模拟 `md` 积分方法都足够精确.

- **md-vv-avek**

等同于 `md-vv` 的速度Verlet 算法, 除了动能是由 `md` 积分方法的两半步动能的平均值决定以外, 因而此方法更精确. 当与Nose-Hoover 和/Parrinello-Rahman 耦合联用时, 此选项会略微增加计算成本.

- **sd**

准确, 高效的蛙跳式随机动力学积分方法. 使用约束时, 在每积分步, 坐标需要被约束两次. 取决于力的计算成本, 这可能会占用模拟的大部分时间. 一组或多组原子(`tc-grps`)的温度可通过 `ref-t` [K] 设定, 每组的逆摩擦常数通过 `tau-t` [ps] 设定. 参数 `tcoupl` 被忽略. 随机数发生器使用 `1d-seed` 初始化. 当作为恒温器使用时, 2 ps 是 `tau-t` 的一个适当值, 因为这时的摩擦比水的内摩擦低, 同时摩擦也高到足以除去多余的热量.

注意: 温度偏差的衰减要比使用相同 `tau-t` 的Berendsen 恒温器快一倍.

- **sd2**

曾经是默认的SD积分方法, 但现在已过时. 每个坐标每一步需要四个高斯随机数. 使用约束时, 温度会稍微过高.

- **bd**

Brown 或位置Langevin 动力学的欧拉积分方法, 粒子速度为所受力与摩擦系数(`bd-fric` [amu ps⁻¹])的比值, 再加上随机热噪声(`ref-t`). 当 `bd-fric=0` 时, 每个粒子的摩擦系数由mass/ `tau-t` 计算, 与 `sd` 积分方法类似. 随机发生器利用 `1d-seed` 初始化.

- **steep**

能量最小化的最陡下降算法. 最大步长为 `emstep` [nm], 容差为 `emtol` [kJ mol⁻¹ nm⁻¹].

- **cg**

能量最小化的共轭梯度算法, 容差为 `emtol` [$\text{kJ mol}^{-1} \text{nm}^{-1}$]. 当由 `nstcgsteep` 决定的最速下降步骤完成后, 使用CG会更高效. 简正模式分析前进行的最小化需要非常高的精度, 为此, 编译GROMACS时应使用双精度.

- **1-bfgs**

能量最小化的准牛顿算法, 基于低内存需求的Broyden-Fletcher-Goldfarb-Shanno 方法, 在实用中, 似乎比共轭梯度法收敛更快, 但是由于必须的校正步骤, 此方法(暂且)不能并行.

- **nm**

对 `tpr` 文件中的结构进行简正模式分析. 应该使用双精度编译的GROMACS.

- **tpi**

测试粒子插入. 拓扑结构中的最后一个分子是测试粒子. 需要 `mdrun` 的 `-rerun` 选项并提供一条轨迹. 这条轨迹不能包含待插入的分子. 每帧中会进行 `nsteps` 次插入操作, 插入时分子的位置和取向都是随机的.

当 `nstlist` 大于1时, 围绕相同随机位置半径为 `rtpi` 的球体进行 `nstlist` 次插入, 并使用相同的邻区列表 (和相同的长程能量, 若 `rwdw>rlist` 或 `rcoulomb>rlist`, 只适用于单原子分子). 因为构造邻区列表很昂贵, 你可以使用同一个列表几乎免费地进行若干次额外的插入. 随机种子由 `ld-seed` 设定. 玻尔兹曼加权的温度由 `ref-t` 设定, 它应与原始轨迹的模拟温度匹配. `tpi` 正确地考虑了色散校正. 所有相关的量都会写入到由 `mdrun` 的 `-tpi` 选项指定的文件中. 插入能量的分布会写入到由 `mdrun` 的 `-tpid` 选项指定的文件中. 不产生轨迹或能量文件. 并行`tpi` 与单节点`tpi`的结果完全相同. 对于带电分子, 使用密格点的PME最准确, 也很有效, 因为对每一帧系统的势能只需要计算一次.

- **tpic**

测试粒子插入到一个预定义的空腔位置. 步骤与`tpi`相同, 除了需要从轨迹中读入一个额外的坐标, 作为插入位置. 要被插入分子的中心应位于0,0,0. GROMACS不会为你完成这点, 因为针对不同的情况, 采用不同的方式定心可能更好. 同时, `rtpi` 设置围绕此处球体的半径. 对每一帧只进行一次邻区搜索, 不会使用 `nstlist`. 并行`tpic` 与单节点`tpic`的结果完全相同.

tinit: (0) [ps]

运行的起始时间(只对 `md`, `sd` 和 `bd` 积分方法有意义)

dt: (0.001) [ps]

积分时间步长(只对 `md`, `sd` 和 `bd` 积分器有意义)

nsteps: (0)

积分或最小化的最大步数, -1意味着不限制步数

init-step: (0)

起始步. 第*i*步的运行时间的计算公式为: $t = t_{\text{init}} + dt * (\text{init-step} + i)$. 自由能计算中lambda的计算公式为: $\text{lambda} = \text{init-lambda} + \text{delta-lambda} * (\text{init-step} + i)$. 非平衡态MD的参数也依赖于步数. 因此, 为了准确的重新启动或重做部分模拟, 可能需要将 `init-step` 设置为重启帧的步数. `gmx convert-tpr` 可自动执行此操作.

comm-mode

- **Linear**
去除质心的平动
- **Angular**
去除质心的平动以及围绕质心的转动
- **None**
不限制质心的运动

nstcomm: (100) [步]

去除质心运行的频率

comm-grps:

需要去除质心运动的组, 默认是整个体系

7.3.4 Langevin动力学

bd-fric: (0) [amu ps⁻¹]Brown 动力学摩擦系数. 当**bd-fric=0**时, 每个粒子的摩擦系数为其质量与**tau-t**的比值.**ld-seed**: (-1) [整数]用于初始化随机和Brown 动力学的热噪声随机数发生器. 当**ld-seed**设置为-1时, 将使用伪随机种子. 当在多个处理器上运行BD或SD时, 每个处理器使用的种子等于**ld-seed**加上处理器的编号.

7.3.5 能量最小化

emtol: (10.0) [kJ mol⁻¹ nm⁻¹]

当最大的力比这个值小时就认为最小化收敛

emstep: (0.01) [nm]

初始步长大小

nstcgsteep: (1000) [步]

用共轭梯度进行能量最小化时执行一步最速下降的频率

nbfgscorr: (10)

L-BFGS 最小化的校正步数. 数值越高(至少理论上)越准确, 但速度更慢.

7.3.6 壳层分子动力学

当体系中存在壳层或柔性约束时, 会在每个时间步对壳层的位置和柔性约束的长度进行优化, 直到在壳层和约束上的力的RMS值小于**emtol**, 或达到迭代的最大次数(**niter**)

emtol: (10.0) [kJ mol⁻¹ nm⁻¹]

当最大的力比这个值小时就认为最小化收敛. 对壳层MD这个值最大是1.0, 但由于此变量也用于能量最小化, 所以其默认值为10.0.

niter: (20)

优化壳层位置和柔性约束的最大迭代次数.

fcstep: (0) [ps²]

优化柔性约束的步长. 应设置为 $\mu/(d^2V/dq^2)$, 其中 μ 为柔性约束中两粒子的约化质量, d^2V/dq^2 是势能在约束方向上的二阶导数. 这个数字对不同的柔性约束不要相差太大, 因为迭代次数进而运行时间对**fcstep**非常敏感. 多尝试几个值!

7.3.7 测试粒子插入

rtpi: (0.05) [nm]

测试粒子插入半径, 请参考积分方法**tpi**和**tpic**

7.3.8 输出控制

nstxout: (0) [步]

将坐标写入输出轨迹文件的间隔步数, 最后一步的坐标始终会写入

nstvout: (0) [步]

将速度写入输出轨迹文件的间隔步数, 最后一步的速度始终会写入

nstfout: (0) [步]

将力写入输出轨迹文件的间隔步数

nstlog: (1000) [步]

将能量写入log文件的间隔步数, 最后一步的能量始终会写入

nstcalcenergy: (100)

两次能量计算之间的间隔步数, 0表示从不计算. 此选项仅与动力学有关. 使用双程截断时**nstcalcenergy**应等于**nstlist**或其倍数. 此选项会影响并行模拟的性能, 因为计算能量需要所有进程之间的全局通讯, 在高度并行的情况下, 这可能会成为计算瓶颈.

nstenergy: (1000) [步]

将能量写入能量文件的间隔步数, 最后一步的能量始终会写入, 应是**nstcalcenergy**的倍数. 注意, 对所有MD步数与**nstcalcenergy**的模, 精确的能量总和及其涨落都存储在能量文件中, 所以当**nstenergy>1**时, **g_energy**仍可以给出精确的能量平均值及其涨落

nstxout-compressed: (0) [步]

利用有损压缩输出位置坐标的间隔步数

compressed-x-precision: (1000) [实数]

压缩轨迹文件的精度

compressed-x-grps:

写入到压缩轨迹文件的组, 默认写入整个体系(如果 **nstxout-compressed**>0)

energygrps:

写入能量文件的组

7.3.9 邻区搜索

cutoff-scheme:

- **Verlet**

利用缓冲生成粒子的配对列表. 缓冲区的大小会根据 **verlet-buffer-tolerance** 自动进行设置, 除非将其设置为-1, 在这种情况下会使用 **rlist**. 此选项在 **rwdw=rcoulomb** 处存在一个明显, 准确的截断. 目前仅支持截断, 反应场, PME静电和普通LJ. **Verlet** 方案不支持 **mdrun** 的一些功能, 但 **grompp** 会对此进行检查. 只有 **Verlet** 支持原生GPU加速. 使用GPU加速的PME或单独的PME队列时, **mdrun** 会通过缩放 **rcoulomb** 和格点间距自动调整CPU/GPU负载平衡. 可以使用 **-notunepme** 关闭此功能. 当体系中不含水分子, 或 **group** 使用配对列表缓冲维持能量守恒时, **Verlet** 会比 **group** 快.

- **group**

为原子组生成配对列表. 这些组对应于拓扑信息中的电荷组. 这是4.6版本之前唯一的截断处理方案. 配对列表没有显式的缓冲. 这使得计算水的力时效率很高, 但只有显式地加入缓冲时, 能量才守恒.

nstlist: (10) [步]

- **>0**

更新邻区列表(以及远程力, 当使用双程截断时)的频率. 当该值为0时, 邻区列表只生成一次. 对能量最小化, 当 **nstlist>0** 时, 每次计算能量都将更新邻区列表. 设置了 **cutoff-scheme=Verlet** 和 **verlet-buffer-tolerance**, **nstlist** 实际是最小值且 **mdrun** 可能会增大它的值, 除非将其设置为1. 对GPU上的并行模拟和(或)非键合力计算, 取20或40往往能得到最佳性能.

使用 **cutoff-scheme=Group** 和非准确的截断, **nstlist** 会影响模拟的精确性, 不能随意选择.

- **0**

邻区列表仅构建一次, 且不再更新. 这主要用于真空中模拟, 这种情况下所有粒子彼此间都能看到对方.

- **-1**

自动更新频率, 仅支持 **cutoff-scheme=group**. 此选项只能用于切换, 移位或用户定制的势能函数, 其截断可以小于 **rlist**. 缓冲的大小为 **rlist** 减去最长的截断值. 以根据前一步的邻区搜索确定的电荷组几何中心开始, 只有当一个或多个粒子移动的距离超过缓冲大小的一半时, 邻区列表才会更新. 更新时会考虑因压力耦合或 **deform** 选项引起的坐标缩放. 此选项可确保没有截断虚假, 但对于较大的体系, 计算成本可能会很高, 因为邻区列表的更新频率仅由一或两个粒子决定, 它们移动的距离稍微超出缓冲长度的一半(这并不一

定意味着邻区列表是无效的), 而99.99%的粒子都不需要更新.

`nstcalclr`: (-1) [步]

使用组截断方案时, 控制远程力的计算周期.

- **1**
在每一步都计算长程力. 当静电和范德华相互作用使用单独的带缓冲的邻区列表时, 此选项有用, 特别地, 它支持范德华的截断距离大于静电的(例如与PME联用时有用). 然而, 对这两种相互作用使用相同的长程截断值, 并在每一步更新它们是没有意义的—此时将一切都放在短程列表中会稍快一些.
- **> 1**
每 `nstcalclr` 步计算一次长程力, 并使用多时间步积分方法将它们组合起来. 现在可以使用超过 `nstlist` 的频率进行计算, 因为会保存列表. 这可能是一个好主意, 例如, 对于变化比静电慢的范德华相互作用.
- **-1**
当进行邻区搜索时计算长程力. 虽然这是默认值, 你可能要考虑更频繁地更新长程力.
注意, PP-PME负载均衡可能会自动导致双程力(twin-range force) 的计算, 这是为了维持所选择的范德华相互作用半径, 即便负载均衡正在改变静电截断. 如果 `.mdp` 文件已经指定了双程相互作用(比如每2–3步使用比PME静电更长的截断计算Lennard-Jones 相互作用), 负载均衡对Lennard-Jones 也会有小的影响, 因为短程截断(在其内部每步都计算力)改变了.

`ns-type`:

- **grid**
在盒子中生成格点, 且在每 `nstlist` 步构建新的邻区列表时才检查在相邻格点区域内的原子. 对大的体系, 格点搜索比简单搜索快得多.
- **simple**
每 `nstlist` 步构建新的邻区列表时, 检查盒子中的每个原子(仅与 `cutoff-scheme=group` 联用).

`pbc`:

- **xyz**
在所有方向上使用周期性边界条件.
- **no**
不使用周期性边界条件, 忽略盒子. 不使用截断的模拟, 可将所有的截断设为0, 并且 `nstlist=0`. 对单个MPI 队列, 不使用截断时, 使用 `nstlist=0`, `ns-type=simple` 可获得最佳性能.
- **xy**
只在x和y方向上使用周期性边界条件. 仅适用于 `ns-type=grid`, 可与 `walls` 联用. 没有墙或只有一个墙时, 体系在z方向上大小是无限的, 因此不能使用压力耦合或Ewald 加和方法. 当使用两面墙时没有这些缺点.

periodic-molecules:

- **no**

分子是有限的, 可以使用快速的分子PBC

- **yes**

用于含有通过周期性边界条件与自身耦合的分子的体系, 需要较慢的PBC算法, 分子在输出中不保持完整

verlet-buffer-tolerance: (0.005) [kJ/mol/ps]

只对 **cutoff-scheme=Verlet** 有用. 此选项设置由 Verlet 缓冲引起的每个粒子配对相互作用的最大允许误差, 间接设置了 **rlist**. 若 **nstlist** 和 Verlet 缓冲大小都固定(出于性能原因), 不在配对列表中的粒子对在 **nstlist**-1 步内能够不时地进入截断距离内. 这将导致非常小的能量跳跃. 对等温系综, 对于给定的截断和 **rlist** 可估算出这些非常小的能量跳跃. 估算时假定均相的粒子分布, 因此对多相体系可能会略微低估误差. 对于较长的配对列表寿命(**nstlist**-1)*dt, 缓冲会被高估, 因为忽略了粒子之间的相互作用.

由于误差抵消, 总能量的实际漂移幅度通常小一到两个数量级. 注意, 与基于简单粒子配对的列表相比, GROMACS 的配对列表设置导致漂移降低为原来的1/10, 生成的缓冲大小考虑了这一点影响. 不使用动力学(能量最小化等)时, 缓冲为截断的5%. 对NVE模拟, 会使用初始温度, 除非初始温度为零, 此时使用10%的缓冲. 对NVE模拟通常需要降低容差以便在纳秒的时间尺度达到适当能量守恒. 要覆盖自动缓冲设置, 可使用 **verlet-buffer-tolerance=-1**, 并手动设置 **rlist**.

rlist: (1) [nm]

短程邻区列表的截断距离. 使用 **cutoff-scheme=Verlet** 时, 默认由 **verlet-buffer-tolerance** 选项设置, 并忽略 **rlist** 值.

rlistlong: (-1) [nm]

长程邻区列表的截断距离. 此参数仅适用于切换势的双程截断设置. 在这种情况下, 需要一个缓冲区域以考虑电荷组的大小. 在所有其他情况下, 该参数自动设置为最长的截断距离.

7.3.10 静电

coulombtype:

- **Cut-off**

双程截断, 其中邻区截断距离为 **rlist**, 库仑截断距离为 **rcoulomb**, 且 **rcoulomb≥rlist**.

- **Ewald**

经典Ewald加和方法. 实空间的截断距离 **rcoulomb** 应等于 **rlist**. 例如, 使用 **rlist=0.9,** **rcoulomb=0.9**. 倒易空间使用的波矢的最大振幅由 **fourierspacing** 控制. 直接/倒易空间的相对精度由 **ewald-rtol** 控制.

注: Ewald 算法的复杂度为 $O(N^{3/2})$, 因此对于大的体系非常慢. 包含这个方法主要是为了作为参考—在大多数情况下PME方法的性能都好得多.

- **PME**

快速平滑粒子网格Ewald(SPME)静电方法. 直接空间类似于Ewald加和方法, 而倒易空间部分使用FFT进行计算. 格点尺寸由 `fourierspacing` 控制, 内插的阶数由 `pme-order` 控制. 使用0.1 nm格点间距的三次内插方法时, 静电力的计算精度为 $2\text{--}3 \times 10^{-4}$. 由于VDW截断导致的误差大于此, 你可以尝试使用0.15 nm的格点间距. 当并行运行时, 内插的并行性能优于FFT, 因此可以试着减小格点尺寸, 同时增加内插.

- **P3M-AD**

粒子粒子粒子网格算法, 具有长程静电相互作用的解析梯度. 除影响函数对格点进行了优化外, 方法和代码与SPME完全相同, 优化使在计算精度略有提高.

- **Reaction-Field electrostatics**

库仑截断距离为 `rcoulomb` 的反应场, 其中 `rcoulomb≥rlist`. 超过截断距离的介电常数为 `epsilon-rf`. 当 `epsilon-rf=0` 时, 介电常数无穷大.

- **Generalized-Reaction-Field**

库仑截断距离为 `rcoulomb` 的广义反应场, 其中 `rcoulomb≥rlist`. 超过截断距离的介电常数为 `epsilon-rf`. 离子强度由带电的(即非零电荷)电荷组计算. GRF势的温度通过 `ref-t` [K] 设定.

- **Reaction-Field-zero**

在GROMACS中, 使用 `cutoff-scheme=group` 时, 正常的反应场静电方法会导致能量守恒性很差.

`Reaction-Field-zero` 通过将超出截断距离的势能设为零解决了这个问题. 这种方法只适用于介电常数无穷大(`epsilon-rf=0`)的情况, 因为只有这样力在截断距离处才能消失. `rlist` 应比 `rcoulomb` 大0.1至0.3 nm, 以考虑电荷组的大小以及更新邻区对扩散的影响. 这一点以及使用查表代替解析函数使得 `Reaction-Field-zero` 的计算比正常反应场更耗时.

- **Reaction-Field-nec**

与 `Reaction-Field` 相同, 但是GROMACS 3.3以前版本中的实现. 没有使用反应场校正排除原子对和自身对的影响. 使用反应场计算1-4相互作用. 因排除不具有1-4相互作用的粒子对而缺少的校正达到总静电能的百分之几, 并导致力和压力有微小的差别.

- **Shift**

类似于 `vdwtype` 的 `Shift`. 你可能想使用 `Reaction-Field-zero` 代替, 它具有类似的势能形状, 但具有物理意义, 并且含有排除校正项, 计算的能量更好.

- **Encad-Shift**

库仑势在整个范围内降低, 使用Encad 模拟包中的定义

- **Switch**

类似于 `vdwtype` 的 `Switch`. 切换库仑势可导致严重的假象, 建议: 使用 `Reaction-Field-zero` 代替.

- **User**

`mdrun` 需要使用文件 `table.xvg`, 里面包含用户定义的势能函数, 包括排斥, 色散和库仑相互作用. 当存在对相互作用时, `mdrun` 也需要描述对相互作用的文件 `tablep.xvg`. 当非键和对相互作用需要使用相同的相互作用时, 用户可以为两个表格文件指定相同的文件名. 这些文件应包含7列: `x` 值, `f(x)`, `-f'(x)`, `g(x)`,

$-g'(x)$, $h(x)$, $-h'(x)$, 其中 $f(x)$ 为库仑函数, $g(x)$ 为色散函数, $h(x)$ 为排斥函数. 当 $vdwtype$ 不为 $User$ 时, 会忽略 g , $-g'$, h 和 $-h'$. 对非键相互作用, x 的取值应该从 0 到最大截断距离 + $table-extension$, 彼此的间距应均匀. 对于对相互作用会使用文件中的表格长度. 当使用混合精度时, 非用户自定义表格的最佳间距为 0.002 [nm], 使用双精度时, 最佳值为 0.0005 [nm]. 在 $x=0$ 处的函数值并不重要. 更多信息请参考印刷手册.

- **PME-Switch**

PME 和对直接空间部分切换函数的组合(参见上文). $rcoulomb$ 可以小于 $rlist$. 主要用于等能量模拟(注意, **PME** 与 **cutoff-scheme=Verlet** 联用会更有效).

- **PME-User**

PME 和用户表格的组合(参见上文). $rcoulomb$ 可以小于 $rlist$. $mdrun$ 会从用户表格中减去 PME 网格的贡献. 因为这个扣除, 用户表格应包含大约 10 个十进制的位置.

- **PME-User-Switch**

PME-User 和切换函数的组合(参见上文). 对最终粒子之间的相互作用使用切换函数, 即, 同时对用户提供的函数和 PME 网格校正部分使用切换函数.

coulomb-modifier

- **Potential-shift-Verlet**

选择 **Potential-shift** 与 Verlet 截断方案, 因为它(几乎)不增加计算量; 选择 **None** 与组截断方案.

- **Potential-shift**

对库仑势进行固定的移位以使其在截断处为零. 这使得势能可由力的积分得到. 注意这并不影响力或采样.

- **None**

使用未经修改的库仑势. 与组方案连用时, 这意味着没有使用准确的截断, 会计算邻区列表中的所有粒子对之间的能量和力.

rcoulomb-switch: (0) [nm]

从何处开始切换库仑势, 只适用于应用力或势能的切换时.

rcoulomb: (1) [nm]

库伦截断距离

epsilon-r: (1)

相对介电常数. 为 0 时意味着无穷大.

epsilon-rf: (0)

反应场的相对介电常数, 仅与反应场静电方法一起使用. 为 0 时意味着无穷大.

7.3.11 VdW

vdwtype:

- **Cut-off**

双程截断, 邻区列表的截断距离为 `rlist`, VdW截断距离为 `rvdw`, 其中 `rvdw≥rlist`.

- **PME**

使用快速平滑粒子网格Ewald(SPME)方法计算VdW相互作用. 与静电计算类似, 格点大小由 `fourierspacing` 控制. 内插的阶数由 `pme-order` 控制. 直接/倒易空间的相对精度由 `ewald-rtol-lj` 控制, 倒易程序部分使用的特定组合规则由 `lj-pme-comb-rule` 设置.

- **Shift**

此功能已废弃, 并被 `vdw-modifier = Force-switch` 取代. LJ(不包括Buckingham) 势在整个范围内降低, 相应的力在 `rwdw-switch` 和 `rwdw` 之间平滑地衰减到零. 邻区搜索的截断距离 `rlist` 应该比 `rwdw` 大0.1至0.3 nm以考虑电荷组大小及邻区列表更新时扩散的影响.

- **Switch**

此功能已废弃, 并被 `vdw-modifier = Potential-switch` 取代. LJ(不包括Buckingham) 势在 `rwdw-switch` 之内是正常的, 之后被逐渐降低, 并在 `rwdw` 处达到零. 势能和力函数都是连续平滑的. 但需要注意, 所有的切换函数都会导致力的突起(增加, 因为我们切换了势能). 邻区搜索的截断距离 `rlist` 应该比 `rwdw` 大0.1至0.3 nm以考虑电荷组大小及邻区列表更新时扩散的影响.

- **Encad-Shift**

LJ(不包括Buckingham) 势在整个范围内降低, 使用Encad 模拟包中的定义.

- **User**

参看 `coulombtype` 的 `user` 选项. `x=0` 处的函数值并不重要. 如果你想使用LJ校正, 请确保 `rwdw` 对应于用户定义函数的截断距离. 若 `coulombtype` 没有设置为 `User`, 会忽略 `f` 和 `-f'`.

vdw-modifier:

- **Potential-shift-Verlet**

选择 `Potential-shift` 与Verlet 截断方案, 因为它(几乎)不增加计算量; 选择 `None` 与组截断方案.

- **Potential-shift**

对VdW势进行固定的移位以使其在截断处为零. 这使得势能可由力的积分得到. 注意这并不影响力或采样.

- **None**

使用未经修改的VdW势. 与组方案连用时, 这意味着没有使用准确的截断, 会计算邻区列表中的所有粒子对之间的能量和力.

- **Force-switch**

在 `rwdw-switch` 和 `rwdw` 之间平滑地将力切换至零. 这使得势能在整个范围内移位并在截断处切换至零, 注意, 这种计算方法比普通截断方法更耗时, 并且不要求能量守恒, 因为 `Potential-shift` 的能量守恒性也差

不多.

- **Potential-switch**

在 **rvdw-switch** 和 **rvdw** 之间平滑地将势能切换至零. 注意, 这会导致力的数值在切换区域产生很大的假象, 并且计算也更耗时. 只有当你使用的力场要求使用时, 才能使用此选项.

rvdw-switch: (0) [nm]

从何处开始切换LJ力或势能, 只适用于应用力或势能的切换时.

rvdw: (1) [nm]

LJ或Buckingham 的截断距离.

DispCorr:

- **no**

不使用任何修正

- **EnerPres**

对能量和压力进行长程色散校正

- **Ener**

只对能量进行长程色散校正

7.3.12 表格

table-extension:: (1) [nm]

非键势能函数查询表超出最大截断距离后的延伸长度. 该值应足够大以考虑电荷组大小及更新邻区列表时扩散的影响. 不使用用户定义势能时, 对1-4相互作用的查询表会使用相同的表格长度, 此长度与非键相互作用的表格无关.

table-extension 的值决不可能影响 **rlist**, **rcoulomb** 或 **rvdw** 的值.

energygrp-table:

当对静电和/或VdW使用用户表格时, 可以在这里列出能量组之间的配对, 这些配对可以使用单独的用户表格. 两个能量组的名称将被追加到表格的文件名中, 追加时按照它们在 **energygrps** 中定义的顺序, 彼此之间以下划线隔开. 例如, 如果 **energygrps = Na Cl Sol**, **energygrp-table = Na Na Na Cl**, 除常规的 **table.xvg** 外, **mdrun** 还会读取 **table_Na_Na.xvg** 和 **table_Na_Cl.xvg**, **table.xvg** 将被用于所有其它能量组配对.

7.3.13 Ewald

fourierspacing: (0.12) [nm]

对普通的Ewald方法, 盒子尺寸和间距的比值决定了在每个(含符号)方向上使用的波矢数目的下限. 对PME和P3M, 该比率决定了沿每个轴使用的Fourier 空间格点数目的下限. 在所有情况下, 每个方向上的数目都可通过非零的 **fourier_n[xyz]** 重新进行设置. 为优化粒子-粒子相互作用计算和PME网格计算之间的相对负载, 知道下面的事实可能对你会有帮助, 当对库仑截断和PME格点间距使用相同的因子进行缩放时, 静电计算的精确度几乎保持不变.

fourier-nx (0) ; **fourier-ny** (0) ; **fourier-nz**: (0)

使用Ewald方法时, 倒易空间波矢的最高振幅. 使用PME或P3M时, 格点的大小. 这些值会覆盖每个方向的 **fourierspacing** 设置. 最佳的数值为2, 3, 5和7的幂, 避免使用大的素数.

pme-order (4)

PME内插的阶数. 4对应于立方内插. 当并行运行时, 你可以尝试6/8/10, 并同时减少格点尺寸.

ewald-rtol (1e-5)

Ewald移位的实空间势能在 **rcoulomb** 处的相对强度由 **ewald-rtol** 给出. 降低此值会得到更精确的实空间加和, 但计算倒易空间加和时需要更多的波矢.

ewald-rtol-1j (1e-3)

当使用PME计算VdW相互作用时, **ewald-rtol-1j** 用于控制 **rvdw** 处色散能的相对强度, 与 **ewald-rtol** 控制静电能的方式类似.

lj-pme-comb-rule (Geometric)

LJ-PME倒易部分VdW参数的组合规则. 几何规则比Lorentz-Berthelot 规则快得多, 因此通常建议优先选择几何规则, 即便力场的其余部分使用了Lorentz-Berthelot 规则.

- **Geometric**

应用几何组合规则

- **Lorentz-Berthelot**

应用Lorentz-Berthelot 组合规则

ewald-geometry: (3d)

- **3d**

在所有三个维度进行Ewald 加和.

- **3dC**

倒易部分的加和仍然以3D进行, 但对 **z** 方向的力和势能进行校正以产生伪二维的加和. 如果体系在 **x-y** 平面具有板状几何结构, 你可以尝试增加盒子在 **z** 方向的长度(盒子的长度取为板高的3倍通常是可以的), 并使用这个选项.

epsilon-surface: (0)

此选项控制3D Ewald 加和的偶极校正. 默认值零意味着不进行校正. 将此值设置为围绕无穷大体系的假想表面的相对介电常数值, 即可进行校正. 小心, 如果你的体系含有自由移动的电荷, 你就应该使用此选项. 这个值不影响长程校正的板状3DC方法.

7.3.14 温度耦合

tcoupl:

- **no**

不使用温度耦合.

- **berendsen**

通过Berendsen 恒温器与温度为 **ref-t** [K] 的热浴耦合, 时间常数 **tau-t** [ps]. 多个组可以独立耦合, 这些组可在 **tc-grps** 中指定, 彼此之间以空格分开.

- **nose-hoover**

使用扩展系综的Nose-Hoover 温度耦合. 参考温度与耦合组的选择方法同上, 但在这种情况下 **tau-t** [ps] 控制的是平衡时温度涨落的周期, 它与弛豫时间稍有不同. 对NVT模拟, 能量的守恒量会写入能量和日志文件.

- **andersen**

通过在每个时间步对一部分粒子随机化进行温度耦合. 参考温度和耦合组的选择同上. **tau-t** 是每个分子两次随机化之间的平均时间间隔. 此方法对粒子的动力学有一定的抑制作用, 但很少或不出现遍历问题. 目前只能用于速度Verlet, 不能用于约束.

- **andersen-massive**

通过在不频繁的时间步对所有粒子随机化进行温度耦合. 参考温度和耦合组的选择方法同上. **tau-t** 是所有分子两次随机化之间的时间间隔. 此方法对粒子的动力学有一定的抑制作用, 但很少或不出现遍历问题. 目前只能用于速度Verlet,

- **v-rescale**

通过速度缩放与随机项联用的方法进行温度耦合(JCP 126, 014101). 此恒温器类似于Berendsen 耦合, 使用相同的 **tau-t** 进行缩放, 但随机项确保了能够产生正确的正则系综. 随机数种子通过 **ld-seed** 设置. 即便对 **tau-t=0** 该恒温器也工作正常. 对于NVT模拟, 能量的守恒量会写入能量和日志文件.

nstcouple: (-1)

与温度耦合的频率. 默认值-1表示 **nstcouple** 与 **nstlist** 相等, 除非 **nstlist<0**, 此时会使用10. 对速度Verlet积分方法, **nstcouple** 被设置为1.

nh-chain-length (10)

速度Verlet积分方法中Nose-Hoover 恒温链的数目, 蛙跳式 **md** 积分方法只支持1. Nose-Hoover 链变量的数据不会输出到.edr, 但可使用环境变量 **GMX_NOSEHOOVER_CHAINS** 设置输出.

tc-grps:

独立地耦合到温度浴的组

tau-t: [ps]

耦合的时间常数(**tc-grps** 中的每组一个值), -1意味着没有温度耦合

ref-t: [K]

耦合的参考温度(**tc-grps** 中的每组一个值)

7.3.15 压力耦合

pcoupl:

- **no**
不使用压力耦合. 这意味着盒子的大小固定.
- **berendsen**
指数弛豫的压力耦合, 时间常数为 **tau-p** [ps]. 在每个时间步对盒子进行缩放. 有人认为, 这并不能得到正确的热力学系综, 但在模拟的开始阶段这是缩放盒子的最有效方式.
- **Parrinello-Rahman**
扩展系综的压力耦合, 盒矢量服从运动方程. 原子的运动方程也耦合到此方程. 没有瞬时的缩放. 与Nose-Hoover 温度耦合类似, 时间常数 **tau-p** [ps] 是平衡时压力的涨落周期. 当你想要在数据收集过程中施加压力缩放时, 这可能是更好的方法, 但需要当心, 如果从不同的压力开始模拟, 你可能会得到非常大的振荡. 对那些精确涨落非常重要的NPT系综的模拟, 或者如果压力耦合时间非常短, 这种方法可能不合适, 因为前面时间步的压力会被GROMACS用于计算当前时间步的压力.
- **MTTK**
Martyna-Tuckerman-Tobias-Klein 实现, 只用于 **md-vv** 或 **md-vv-avek**, 非常类似于Parrinello-Rahman. 与Nose-Hoover 温度耦合类似, 时间常数 **tau-p** [ps] 是平衡时压力的涨落周期. 当你想要在数据收集过程中施加压力缩放时, 这可能是更好的方法, 但需要当心, 如果从不同的压力开始模拟, 你可能会得到非常大的振荡. 目前仅支持各向同性缩放.

pcoupltype:

- **isotropic**
各向同性压力耦合, 时间常数为 **tau-p** [ps]. 压缩系数和参考压力分布分别通过 **compressibility** [bar⁻¹] 和 **ref-p** [bar] 进行设置, 都只需要一个值.
- **semisotropic**
压力耦合在 **x** 和 **y** 方向各向同性, 但 **z** 方向上的不同. 对膜模拟有用. 需要两个值, 分别对应于 **x/y** 方向和 **z** 方向.
- **anisotropic**
同上, 需要6个值, 分别是 **xx**, **yy**, **zz**, **xy/yx**, **xz/zx** 和 **yz/zy** 分量. 当非对角线压缩系数为零时, 长方体盒子在模拟中仍将保持为长方体. 要注意的是, 各向异性缩放可能会导致模拟盒子的剧烈变形.
- **surface-tension**
表面张力耦合, 表面平行于 **xy** 平面. **z** 方向采用正常的压力耦合, 表面张力耦合到盒子的 **x/y** 维度. **ref-p** 的第一个值是参考表面张力与表面数目的乘积, 单位为 [bar nm], 第二个值是参考的 **z**-压力, 单位为 [bar]. 两个 **compressibility** 的值(单位 [bar⁻¹])分别为 **x/y** 和 **z** 的方向的压缩系数. **z** 方向压缩系数的值应该具有一定的准确度, 因为它会影响表面张力的收敛性, 也可以将其设置为零以保持盒子的高度固定.

nstpcouple: (-1)

压力耦合的频率. 默认值为-1, 表示 **nstpcouple** 等于 **nstlist**, 除非 **nstlist <= 0**, 此时使用10. 对速度Verlet积分方法, **nstpcouple** 被设置为1.

tau-p(1): [ps]

耦合的时间常数

compressibility: [bar⁻¹]

压缩系数(注意: 现在真的是以bar⁻¹为单位了), 处于1个标准大气压, 300 K下的水其压缩系数为4.5e-5 [bar⁻¹].

ref-p: [bar]

耦合的参考压力

refcoord-scaling:

- **no**

不修改用于位置限制的参考坐标. 注意, 使用该选项维里和压力将取决于参考坐标的绝对位置.

- **all**

利用压力耦合缩放矩阵对参考坐标进行缩放.

- **com**

利用压力耦合缩放矩阵对参考坐标的质心进行缩放. 每个参考坐标到质心的矢量不进行缩放. 只使用一个质心, 即便有多个分子存在位置限制. 在计算初始构型参考坐标的质心时, 不考虑周期性边界条件.

7.3.16 模拟退火

在GROMACS中, 对每个温度组的模拟退火是分开控制的. 参考温度是一个分段线性函数, 对每个组可以使用任意数目的点, 并选择单一序列或周期性退火类型. 实际退火是通过动态地改变参考温度进行的, 由于选择的温控算法也使用该温度, 所以要记住, 体系通常不会瞬间达到参考温度!

annealing:

每个温度组的退火类型

- **no**

不进行模拟退火, 只耦合到参考温度.

- **single**

退火点的单一序列. 如果模拟时间比最后一点的时间还长, 当退火序列达到最后的时间点后, 温度将耦合到最后一点的值并保持不变.

- **periodic**

一旦到达最后的参考时间, 退火将从第一个参考点重新开始. 此过程不断重复, 直到模拟结束.

annealing-npoints:

退火参考/控制点数目的列表, 用于每个温度组. 对不退火的组使用0. 此项的数目应等于温度组的数目.

annealing-time:

退火参考/控制点的时间列表, 用于每个温度组. 如果你正使用周期性退火类型, 时间将与最终值取模, 即, 如果时间为0, 5, 10和15, 耦合将会在15 ps, 30 ps, 45 ps 等时间点后以0 ps时值重新启动. 此项的数目应该等于**annealing-npoints**给出的数字的总和.

annealing-temp:

退火参考/控制点的温度列表, 用于每个温度组. 此项的数目应该等于**annealing-npoints**给出的数字的总和.

很迷惑? 好吧, 让我们举个例子. 假设你有两个温度组, 组的选择设置为**annealing = single periodic**, 组的点数设置为**annealing-npoints = 3 4**, 退火时间设置为**annealing-time = 0 3 6 0 2 4 6**, 温度设置为**annealing-temp = 298 280 270 298 320 320 298**. 在0 ps第一组将被耦合到298 K, 但参考温度在3 ps内将线性下降到280 K, 然后在3 ps到6 ps这段时间内, 温度将从280 K线性变化到270 K. 在这之后温度保持270 K不变. 0 ps时第二组被耦合到298 K, 在2 ps内温度线性增加到320 K, 并保持不变直到4 ps. 在4 ps和6 ps之间温度降低到298 K, 然后将以同样的方式重新开始, 即在6 ps和8 ps之间从298 K线性上升到320 K. 如果你不确定, 请检查**grompp**给出的汇总信息!

7.3.17 速度产生

gen-vel:

- **no**

不产生速度. 当输入结构文件中不存在速度时, 速度被设置为零.

- **yes**

grompp根据温度为**gen-temp** [K]的麦克斯韦分布产生速度, 随机数种子为**gen-seed**. 此选项只对**md**积分器有意义.

gen-temp: (300) [K]

麦克斯韦分布的温度

gen-seed: (-1) [整数]

用于初始化产生随机速度的随机数发生器, 当**gen-seed**设置为-1时, 将使用伪随机种子.

7.3.18 键约束

constraints:

- **none**

除在拓扑中明确定义的键外, 不使用任何约束, 即, 利用简谐(或其他)势或Morse势(取决于**morse**的设置)描述键, 利用简谐(或其他)势描述键角.

- **h-bonds**

将含有氢原子的键转换为约束.

- **all-bonds**

将所有的键都转换为约束.

- **h-angles**

将所有涉及氢原子的键和键角都转换为键约束.

- **all-angles**

将所有的键和键角都转换为键约束.

constraint-algorithm:

- **LINCS**

线性约束求解器(LINear Constraint Solver). 区域分解会与并行版本的P-LINCS一起使用. 使用**lincs-order**设置精度, 同时也设置矩阵求逆展开中矩阵的数目. 经过矩阵求逆校正后, 算法会执行一次迭代校正以补偿因旋转导致的增长. 这种迭代的次数可以通过**lincs-iter**控制. 每**nstlog**步, 相对约束的根均方偏差会打印到日志文件. 如果某根键在一步中的旋转超过了**lincs-warnangle**(度), 将打印警告到日志文件和**stderr**. LINCS不能用于耦合键角约束.

- **SHAKE**

与LINCS相比, SHAKE方法稍慢, 且不太稳定, 但能用于键角约束. 相对容差由**shake-tol**设置, 对“正常”的MD, 0.0001是合适的值. SHAKE不支持处于不同节点上的原子之间的约束, 因此当存在电荷组之间的约束时, 它不能与区域分解一起使用. SHAKE不能用于能量最小化.

continuation:

此选项以前的名字为**unconstrained-start**.

- **no**

对初始构型施加约束并复位壳层

- **yes**

不对初始构型施加约束, 不复位壳层, 对准确的延续和重新运行很有用.

shake-tol: (0.0001)

SHAKE相对容差

lincs-order: (4)

约束耦合矩阵展开的最高阶数. 当约束形成三角形时, 在这些三角形约束的正常展开之上会施加一个相同阶的展开. 对“正常”的MD模拟通常4阶就足够了; 对含有虚拟位点或BD并使用大时间步长的模拟, 需要使用6阶; 对精确的能量最小化, 可能需要使用8或更高的阶数. 与区域分解联用时, 原胞的大小由**lincs-order**+1个约束张成的距离决定. 如果想进行超过此限制的缩放, 可以降低**lincs-order**, 增加**lincs-iter**, 因为当 $(1 + \text{lincs-iter})^* \text{lincs-order}$ 保

持不变时, 精度不会变差.

lincs-iter: (1)

LINCS中用于校正旋转增长的迭代次数. 对于正常的运行一步就足够了. 但对于NVE模拟, 如果需要精确的能量守恒或精确的能量最小化, 你可能需要将其增加到2.

lincs-warnangle: (30) [度]

LINCS失效前键能够旋转的最大角度.

morse:

- **no**

使用简谐势描述键

- **yes**

使用Morse势描述键

7.3.19 能量组排除

energygrp-excl:

所有非键相互作用被排除在外的能量组对. 例如: 如果你有两个能量组 **Protein** 和 **SOL**, 指定

```
energygrp-excl = Protein Protein SOL SOL
```

只会给出蛋白质和溶剂之间的非键相互作用. 对于加快 **mdrun -rerun** 的能量计算, 排除冻结组之间的相互作用, 此选项特别有用.

7.3.20 墙

nwall: 0

当设置为 **1** 时, 在 **z=0** 处存在一面墙; 当设置为 **2** 时, 还存在一面位于 **z=z-box** 处的墙. 墙只能用于 **pbc=xy**. 当设置为 **2** 时, 可以使用压力耦合与Ewald加和(通常最好是使用半各向同性的压力耦合, 并将 **x/y** 的压缩率设置为0, 否则表面积会发生变化). 墙会与体系的其余部分进行相互作用, 其 **atomtype** 是可选的. 会自动增加能量组 **wall0** 和 **wall1** (**nwall=2** 时) 以监测能量组与每面墙之间的相互作用. Z方向的质心运动移除将被关闭.

wall-atomtype:

每面墙在力场中的原子类型名称. 通过(例如)在拓扑文件中定义一个特殊的墙原子类型及其组合规则, 可以独立地调整每个原子类型和墙的相互作用.

wall-type:

- **9-3**

对墙后体积进行积分的LJ势: 9-3势

- **10-4**

对墙面进行积分的LJ势: 10-4势

- **12-6**
直接的LJ势, 由与墙的Z距离决定
- **table**
用户定义的势, 根据与墙的Z距离进行索引, 以类似于 **energygrp-table** 的选项读入, 其中的第一个名称为“正常”能量组, 第二名称为 **wall0** 或 **wall1**, 只使用表中的色散和排斥列.

wall-r-linpot: -1 [nm]

与墙的距离在此值以下时, 势能线性连续, 因此力为常数. 当一些原子超过墙时, 将此选项设置为正值对平衡尤其有用. 当此值 ≤ 0 (对 **wall-type=table** 则是 <0) 时, 原子超过墙后会产生致命错误.

wall-density: [nm⁻³/nm⁻²]

每面墙的原子数密度, 适用于类型为9-3和10-4的墙

wall-ewald-zfac: 3

第三个盒矢量的缩放因子, 仅用于Ewald加和, 最小值为2. Ewald加和只能与 **nwall=2** 联用, 并需要使用 **ewald-geometry=3dc**. 盒子中真空层的作用是降低周期性映象之间不合理的库仑相互作用.

7.3.21 质心牵引

pull:

- **no**
不使用质心牵引. 以下所有的牵引选项都将被忽略(选项如果存在于 **.mdp** 文件中会导致警告)
- **umbrella**
使用参考组与一个或多个组之间的伞势牵引质心
- **constraint**
使用参考组与一个或多个组之间的约束牵引质心. 设置与 **umbrella** 完全相同, 除了使用的是刚性约束而不是简谐势.
- **constant-force**
使用线性势牵引质心, 拉力恒定. 此选项没有参考位置, 因此不会使用参数 **pull-init** 和 **pull-rate**.

pull-geometry:

- **distance**
沿着连接两组的矢量进行牵引. 可以使用 **pull-dim** 选择分量.
- **direction**
在 **pull-vec** 方向进行牵引.
- **direction-periodic**

与 `direction` 相同, 但允许距离超过盒长的一半. 使用这种几何设置, 盒子在牵引维度不应该发生变化(例如, 无压力缩放), 不会将拉力添加到维里.

- `cylinder`

用于相对于层的牵引, 参考质心由参考组的一个局部圆柱部分给出. 牵引方向为 `pull-vec`. 使用两个半径从参考组中选择一个圆柱, 圆柱围绕的轴以 `pull-vec` 方向通过牵引组. 半径 `pull-r1` 之内的所有相对权重为1, `pull-r1` 和 `pull-r0` 之间的权重被切换到零. 也会使用质量权重. 注意, 半径应小于盒子长度的一半. 对于倾斜圆柱, 半径应该比盒长一半更小, 因为参考组中的原子与牵引组质心同时具有径向和轴向分量.

`pull-dim`: (Y Y Y)

与牵引几何设置 `distance` 联用的距离分量, 也会设置打印到输出文件的分量.

`pull-r1`: (1) [nm]

牵引几何 `cylinder` 的圆柱内径

`pull-r0`: (1) [nm]

牵引几何 `cylinder` 的圆柱外径

`pull-constr-tol`: (1e-6)

约束牵引的相对约束容差

`pull-start`:

- `no`

不修改 `pull-init`

- `yes`

将初始构型的质心距离添加到 `pull-init`

`pull-print-reference`: (10)

- `no`

不打印每个牵引坐标中第一组的质心

- `yes`

打印每个牵引坐标中第一组的质心

`pull-nstxout`: (10)

所有牵引组质心的输出频率

`pull-nstfout`: (1)

所有牵引组受力的输出频率

`pull-nngroups`: (1)

牵引组的数量, 使用时不包括绝对参考组. 牵引组可以在多个牵引坐标中重复使用. 下面只给出了第1组的牵引选项, 对其他组的选项, 只需简单地增加组编号即可.

pull-ncoords: (1)

牵引坐标的数目. 下面只给出了坐标1的牵引选项, 对其他其他坐标的选项, 只需简单地增加组编号即可.

pull-group1-name:

牵引组的名称, 在索引文件或默认组中查找, 以获得涉及的原子.

pull-group1-weights:

可选的相对权重, 原子质量乘以此值给出质心的总权重. 此值应为0, 意味着所有原子的相对权重为1, 或牵引组中原子的数目.

pull-group1-pbcatom: (0)

处理组内周期性边界条件的参考原子(不影响组间PBC的处理). 此选项仅当牵引组的直径超过最短盒矢量长度的一半时才重要. 为了确定质心, 组中所有原子被置于其最接近**pull-group1-pbcatom**的周期性映象. 此值为0时意味着使用中间原子(编号顺序). 此参数不能用于牵引几何**cylinder**. 此值为-1时启用余弦加权, 这对周期性的体系中的一组分子有用, 例如, 水的平板(参考 Engin et al. J. Chem. Phys. B 2010).

pull-coord1-groups:

给出牵引坐标作用的两个组的编号. 第一个编号可以是0, 在这种情况下使用**pull-coord1-origin**的绝对参考. 使用绝对参考时体系不再具有平移不变性, 你应该考虑如何处理质心的运动.

pull-coord1-origin: (0.0 0.0 0.0)

使用绝对参考时牵引的参考位置.

pull-coord1-vec: (0.0 0.0 0.0)

牵引方向. **grompp**会对此矢量进行归一化.

pull-coord1-init: (0.0) [nm]

t=0时刻的基准距离.

pull-coord1-rate: (0) [nm/ns]

基准位置的变化速率.

pull-coord1-k: (0) [$\text{kJ mol}^{-1} \text{nm}^{-2}$ / $\text{kJ mol}^{-1} \text{nm}^{-1}$]

力常数. 对于伞势牵引, 此值为简谐力常数[$\text{kJ mol}^{-1} \text{nm}^{-2}$]. 对于恒力牵引, 此值为线性势的力常数, 因而为力常数[$\text{kJ mol}^{-1} \text{nm}^{-1}$]的负值(!).

pull-coord1-kB: (pull-k1) [$\text{kJ mol}^{-1} \text{nm}^{-2}$ / $\text{kJ mol}^{-1} \text{nm}^{-1}$]

与**pull-coord1-k**类似, 但用于状态B. 仅当启用**free-energy**时才使用. 力常数为 (1-lambda) **pull-coord1-k** + lambda **pull-coord1-kB**.
 * * * * *

7.3.22 NMR精修

disre:

- **no**
忽略拓扑文件中的距离约束信息
- **simple**
简单(每分子)的距离约束.
- **ensemble**
一个模拟盒中分子系综的距离约束. 正常情况下, 需要对多个子体系进行系综平均, 每个系综处于单独的盒子中, 使用 `mdrun -multi` 提供包含不同坐标和/或速度的 `topol0.tpr`, `topol1.tpr`, ... 环境变量 `GMX_DISRE_ENSEMBLE_SIZE` 设置每个系综中体系的数目(通常等于 `mdrun -multi` 的值).

disre-weighting:

- **equal** (默认)
将约束力平分到约束中的所有原子对上
- **conservative**
约束力为约束势的导数, 将导致原子对的权重为 r^{-7} . 当 `disre-tau` 为零时力是守恒的.

disre-mixed

- **no**
计算约束力时使用时间平均的违反
- **yes**
计算约束力时使用时间平均违反与瞬时违反乘积的平方根

disre-fc: (1000) [$\text{kJ mol}^{-1} \text{nm}^{-2}$]

距离约束的力常数, 对每个约束乘以一个(可能)不同的因子, 约束在拓扑文件中相互作用的 `fac` 列给出.

disre-tau: (0) [ps]

进行距离约束平均的时间常数. 零值关闭时间平均.

nstdisreout: (100) [步]

进行时间平均的间隔步数, 也是约束中涉及的所有原子对之间的瞬时距离写入到能量文件的间隔步数(会使能量文件变得非常大)

orire:

- **no**

忽略拓扑文件中的取向约束信息

- **yes**

使用取向约束, 可以利用 **mdrun -multi** 进行系综平均

orire-fc: (0) [kJ mol]

取向约束的力常数, 对每个约束乘以一个(可能)不同的权重因子. 可设置为零以获得自由模拟的取向.

orire-tau: (0) [ps]

对取向约束进行时间平均的时间常数. 零值关闭时间平均.

orire-fitgrp:

取向约束的叠合组. 此原子组用于确定体系相对于参考取向的旋转矩阵 **R**. 参考取向为第一个子体系的初始构型. 对蛋白质, 主链是合理的选择.

nstorigeout: (100) [步]

进行时间平均的间隔步数, 也是所有约束的瞬时取向和分子序张量写入到能量文件的间隔步数(会使能量文件变得非常大)

7.3.23 自由能计算

free-energy:

- **no**

只使用拓扑A.

- **yes**

在拓扑A($\lambda=0$) 和拓扑B($\lambda=1$) 之间进行内插, 并将哈密顿量对 λ (由 **dhdl-derivatives** 指定)的导数或哈密顿量对其他 λ 值(由 **foreign-lambda** 指定)的差值写入到能量文件和/或 **dhdl.xvg**. 这些文件可用诸如 **g_bar** 等程序进行处理. 对势能, 键长, 键角进行线性内插的方法见本手册中的说明. 当 **sc-alpha** 大于零时, 对LJ和库仑相互作用使用软核势.

- **expanded**

启用扩展系综模拟, 其中转化状态变为动力学变量, 允许不同哈密顿之间的跳跃. 请参考扩展系综的选项, 这些选项控制了如何进行扩展系综模拟. 扩展系综模拟中使用的不同哈密顿由其他自由能选项定义.

init-lambda: (-1) [浮点数]

λ 的起始值. 通常, 只能用于慢增长方法(即 **delta-lambda** 非零). 在其他情况下, 应指定 **init-lambda-state** 来代替. 必须大于或等于0.

delta-lambda: (0)

每个时间步 λ 的增量

init-lambda-state: (-1) [整数]

lambda 状态的起始值. 指定应使用lambda 向量(**coul-lambdas**, **vdw-lambdas**, **bonded-lambdas**, **restraint-lambdas**, **mass-lambdas**, **temperature-lambdas**, **fep-lambdas**)哪一列. 这是一个从零开始的索引: **init-lambda-state** 0表示第一列, 依此类推.

fep-lambdas:()

零, 一个或更多lambda 值, 会计算与其相应的哈密顿差值, 并每隔 **nstdhdl** 步写入到dhdl.xvg 文件. 值必须处于0和1之间. 不同lambda 值之间的自由能差值可以利用 **g_bar** 进行计算. **fep-lambdas** 不同于其他-lambdas 关键词, 因为所有未指定的lambda 向量的分量都将使用 **fep-lambdas** (包括约束的lambdas, 因而也包括牵引代码约束).

coul-lambdas:()

零, 一个或更多lambda 值, 会计算与其相应的哈密顿差值, 并每隔 **nstdhdl** 步写入到dhdl.xvg 文件. 值必须处于0和1之间. 只有静电相互作用由此lambda 向量的分量控制(并且仅当lambda=0 和lambda=1 的状态具有不同的静电相互作用时).

vdw-lambdas:()

零, 一个或更多lambda 值, 会计算与其相应的哈密顿差值, 并每隔 **nstdhdl** 步写入到dhdl.xvg 文件. 值必须处于0和1之间. 只有van der Waals 相互作用由此lambda 向量的分量控制.

bonded-lambdas:()

零, 一个或更多lambda 值, 会计算与其相应的哈密顿差值, 并每隔 **nstdhdl** 步写入到dhdl.xvg 文件. 值必须处于0和1之间. 只有键合相互作用由此lambda 向量的分量控制.

restraint-lambdas:()

零, 一个或更多lambda 值, 会计算与其相应的哈密顿差值, 并每隔 **nstdhdl** 步写入到dhdl.xvg 文件. 值必须处于0和1之间. 只有约束相互作用: 二面角约束和牵引代码约束由此lambda 向量的分量控制.

mass-lambdas:()

零, 一个或更多lambda 值, 会计算与其相应的哈密顿差值, 并每隔 **nstdhdl** 步写入到dhdl.xvg 文件. 值必须处于0和1之间. 只有粒子质量由此lambda 向量的分量控制.

temperature-lambdas:()

零, 一个或更多lambda 值, 会计算与其相应的哈密顿差值, 并每隔 **nstdhdl** 步写入到dhdl.xvg 文件. 值必须处于0和1之间. 只有温度由此lambda 向量的分量控制. 注意这些lambda 不能用于副本交换, 只能用于模拟回火.

calc-lambda-neighbors (1)

如果设置了 **init-lambda-state**, 此选项控制lambda 值的数目, 会计算相应的哈密顿差值并输出. 正值将lambda 点的数目限制为只计算到 **init-lambda-state** 的第n个邻居. 例如, 若 **init-lambda-state** 为5, 此参数值为2, 会计算lambda 点3-7的能量并输出. 此值为-1意味着会输出所有lambda. 对正常的BAR如g_bar, 此值取1就足够了, 而对于MBAR应该使用-1.

sc-alpha: (0)

软核势的 α 参数, 为0时对LJ和库仑相互作用进行线性内插

sc-r-power: (6)

软核势方程中径向项的次数. 可能的值为6和48. 6更标准一些, 为默认值. 当使用48时, **sc-alpha**一般应更小一些(0.001至0.003).

sc-coul: (no)

是否对分子的库仑相互作用使用软核自由能相互作用变换. 默认是不使用, 因为在禁用van der Waals 相互作用前线性地禁用库仑相互作用通常更有效.

sc-power: (0)

软核函数中lambda 的次数, 只支持1和2

sc-sigma: (0.3) [nm]

软核势的sigma 值, 用于那些C6或C12参数为零, 或sigma 小于**sc-sigma**的粒子

couple-moltype:

这里可以设置计算溶剂化或耦合自由能的分子类型(在拓扑中定义). 特殊选项 **system** 用于耦合体系中的所有分子类型, 这对于平衡由(几乎)随机坐标开始的体系有用. 必须启用 **free-energy** 选项. 此分子类型的Van der Waals 相互作用和/或电荷在lambda=0 和lambda=1 之间可以启用或关闭, 取决于 **couple-lambda0** 和 **couple-lambda1** 的设置. 如果想对分子多个副本中的一个去耦合, 你需要在拓扑中复制并重命名分子的定义.

couple-lambda0:

- **vdw-q**

在lambda=0 开启所有相互作用

- **vdw**

在lambda=0 电荷为零(无库仑相互作用)

- **q**

在lambda=0 开启Van der Waals 相互作用. 需要使用软核相互作用避免奇点

- **none**

在lambda=0 关闭Van der Waals 相互作用, 并且电荷为零. 需要使用软核相互作用避免奇点

couple-lambda1:

类似于 **couple-lambda1**, 但用于lambda=1

couple-intramol:

- **no**

对于分子类型 **couple-moltype**, 分子内的所有非键相互作用都被排除, 或以显式的对相互作用代替. 以这种方式, 分子的去耦合状态对应于无周期效应的适当真空状态.

- **yes**

也启用/关闭分子内的Van der Waals 和库仑相互作用. 用于较大分子的配分自由能, 这种情况下分子内的非键相互作用可能导致分子被动力学地局限于真空中的构型. 不会关闭1–4对相互作用.

nstdhdl: (100)

输出dH/dlambda 和可能的哈密顿差值到dhdl.xvg 文件的频率, 0表示不输出, 此值应为 **nstcalcenergy** 的倍数.

dhdl-derivatives: (yes)

如果为yes(默认值), 每 **nstdhdl** 步会输出哈密顿对lambda 的导数. 使用 **g_bar** (尽管使用正确的 **foreign-lambda** 设置也可以做到, 但可能不够灵活)或热力学积分对线性能量差值进行内插时需要这些导数值.

dhdl-print-energy: (no)

在dhdl 文件中包含总能量会势能. 可用选项有'no', 'potential' 或'total'. 如果感兴趣的状态处于不同的温度, 后面进行自由能分析时需要这些信息. 如果所有状态的温度都相同, 不需要这些信息. 当使用 **mdrun -rerun** 产生 **dhdl.xvg** 文件时, 'potential' 选项非常有用. 当从已有轨迹重新运行时, 动能经常是不正确的, 因此必须单独使用势能来计算残余的自由能, 并解析地计算动能分量.

separate-dhdl-file: (yes)

- **yes**

计算出的自由能(在 **foreign-lambda** 和 **dhdl-derivatives** 中设置)写入到一个单独的文件, 默认文件名为 **dhdl.xvg**. **g_bar** 可直接使用此文件.

- **no**

自由能写入到能量输出文件(**ener.edr**, 以累积块的形式, 每 **nstenergy** 步一次), 可使用 **g_energy** 或直接使用 **g_bar** 来提取.

dh-hist-size: (0)

如果为非零值, 指定分格哈密顿差值(**foreign-lambda** 指定)或导数dH/dl 值的直方图的大小, 并写入ener.edr. 当计算自由能差值时, 这样做可以节省磁盘空间. 每个 **foreign lambda** 输出一个直方图, 每个dH/dl 输出两个直方图, 每 **nstenergy** 步一次. 记住, 不正确的直方图设置(尺寸过小或分格太宽)可引入错误. 不要使用直方图, 除非你确定自己需要它.

dh-hist-spacing: (0.1)

指定直方图的分格宽度, 以能量为单位. 与 **dh-hist-size** 结合使用. 此大小限制了自由能计算的精度. 不要使用直方图, 除非你确定自己需要它.

7.3.24 扩展系综计算

nstexpanded

在扩展系综模拟中, 尝试移动之间的积分步数, 移动时会改变体系的哈密顿量. 必须为 **nstcalcenergy** 的倍数, 但可以大于或小于 **nstdhdl**.

lmc-stats:

- **no**
不在状态空间中进行Monte Carlo.
- **metropolis-transition**
使用Metropolis 权重更新每个状态的扩展系综权重. Min1, $\exp(-(\beta_{\text{new}} u_{\text{new}} - \beta_{\text{old}} u_{\text{old}}))$
- **barker-transition**
使用Barker 转移判据更新每个状态i的扩展系综权重, 定义为 $\exp(-\beta_{\text{new}} u_{\text{new}})/[\exp(-\beta_{\text{new}} u_{\text{new}})+\exp(-\beta_{\text{old}} u_{\text{old}})]$
- **wang-landau**
使用Wang-Landau 算法(在状态空间, 而不是能量空间)来更新扩展系综的权重.
- **min-variance**
使用Escobedo 等人的最小方差更新方法来更新扩展系综的权重. 权重将不再是自由能, 但更重视那些需要更多采样以给出不确定度的状态.

lmc-mc-move:

- **no**
不在状态空间中进行Monte Carlo.
- **metropolis-transition**
随机选择一个新的状态, 向上或向下, 然后使用Metropolis 判据来决定接受还是拒绝: Min1, $\exp(-(\beta_{\text{new}} u_{\text{new}} - \beta_{\text{old}} u_{\text{old}}))$
- **barker-transition**
随机选择一个新的状态, 向上或向下, 然后使用Barker 转移判据来决定接受还是拒绝: $\exp(-\beta_{\text{new}} u_{\text{new}})/[\exp(-\beta_{\text{new}} u_{\text{new}})+\exp(-\beta_{\text{old}} u_{\text{old}})]$
- **gibbs**
使用给定坐标状态的条件权重 $(\exp(-\beta_i u_i)/\sum_k \exp(\beta_i u_i))$ 决定转移到哪个状态.
- **metropolized-gibbs**
使用给定坐标状态的条件权重 $(\exp(-\beta_i u_i)/\sum_k \exp(\beta_i u_i))$ 决定转移到哪个状态, 排除当前状态, 然后使用拒绝步骤保证细致平衡. 总是比Gibbs 方法高效, 尽管在许多情况下只略微高效一点, 例如只有当最近的邻区有明显的相空间重叠时.

lmc-seed: (-1)

在状态空间进行Monte Carlo 移动时使用的随机数种子. 当**lmc-seed**设置为-1时, 将使用伪随机数种子.

mc-temperature:

用于接受/拒绝Monte Carlo 移动的温度. 如果未指定, 会使用在第一组**ref_t**中指定的模拟温度.

wl-ratio: (0.8)

要重置的状态占据数直方图的截断值, 自由能增量重置为 $\text{delta} \rightarrow \text{delta}^* \text{wl-scale}$. 如果我们定义 $\text{Nratio} = (\text{每个直方区间的采样数}) / (\text{每个直方区间的平均采样数})$, **wl-ratio**的值为0.8意味着, 只有当所有 $\text{Nratio} > 0.8$ 并且 同时所有 $1/\text{Nratio} > 0.8$ 时. 直方图才被认为时平坦的.

wl-scale: (0.8)

每当直方图被认为很平时, 自由能Wang-Landau 增量的当前值会乘以**wl-scale**. 此值必须介于0和1之间.

init-wl-delta: (1.0)

Wang-Landau 增量的初始值, 以kT为单位. 接近1 kT的值通常最有效, 尽管有时2–3 kT的值更好, 如果自由能差值较大的话.

wl-oneovert: (no)

在大量采样极限情况下, 设置Wang-Landau 增量的缩放为模拟时间的倒数. 有确切证据表明, 这里使用的状态空间中标准的Wang-Landau 算法可导致自由能‘燃烧’到不正确的值, 且依赖于初始状态. 当**wl-oneovert**为yes时, 若增量小于 $1/N$, 其中N为收集的样本数(因此正比于数据收集时间, ‘ $1/t'$), 则将Wang-Lambda 增量设设置为 $1/N$, 每步降低. 一旦发生这种情况, 将忽略**wl-ratio**, 但当达到**lmc-weights-equil**设置的平衡判据后, 权重仍将停止更新.

lmc-repeats: (1)

控制每次迭代中执行每个Monte Carlo 交换类型的次数. 在大量Monte Carlo 重复的极限情况下, 所有方法都收敛到Gibbs 采样方法. 此值通常不需要不同于1.

lmc-gibbsdelta: (-1)

限制Gibbs 采样到选定的相邻状态数. 对于Gibbs 采样方法, 对所有定义的状态都进行采样有时效率比较低. **lmc-gibbsdelta**取正值意味着只有加或减**lmc-gibbsdelta**的状态才能进行上下交换. 值为-1意味着所有状态都可交换. 当状态数少于100时, 包括所有的状态可能并没有那么耗时.

lmc-forced-nstart: (0)

强制在初始状态空间进行采样以产生权重. 为得到合理的初始权重, 此设置允许模拟从开始到最后的lambda 状态进行驱动, 在每个状态, 移动到下一lambda 状态前进行**lmc-forced-nstart**步. 如果**lmc-forced-nstart**足够长(几千步, 也许), 权重就会接近正确值. 然而, 在大多数情况下, 简单地使用标准的权重平衡算法可能更好.

nst-transition-matrix: (-1)

输出扩展系综转移矩阵的频率. 负值表示只在模拟的最后输出.

symmetrized-transition-matrix: (no)

是否对称化经验转移矩阵. 在极限情况下, 矩阵将是对称的, 但在短的时间尺度内由于统计噪声会变得不对称. 通过使用矩阵 $T_{\text{sym}} = 1/2 (T + \text{transpose}(T))$ 强制对称化, 可以避免一些问题, 如(振幅很小的)负本征值.

mininum-var-min: (100)

如果选择了**min-variance**策略(**lmc-stats**)的选项仅适用于大量采样, 如果在每个状态使用的样本太少会被卡住.)
mininum-var-min为每个状态在**min-variance**策略被激活之前允许的最小采样数.

init-lambda-weights:

用于扩展系综状态的初始权重(自由能). 默认为零权重向量. 格式类似于 **fep-lambda** 设置的lambda 向量, 但权重可以为任意浮点数. 单位为kT. 长度必须匹配lambda 向量的长度.

lmc-weights-equil: (no)

- **no**

在整个模拟中连续更新扩展系综权重.

- **yes**

输入的扩展系综权重被视为平衡值, 在整个模拟过程中不更新.

- **wl-delta**

当Wang-Landau 增量低于 **weight-equil-wl-delta** 指定的值时, 停止更新扩展系综权重.

- **number-all-lambda**

当所有lambda 状态的采样数都大于 **weight-equil-number-all-lambda** 指定的值时, 停止更新扩展系综权重.

- **number-steps**

当步数大于 **weight-equil-number-steps** 指定的值时, 停止更新扩展系综权重.

- **number-samples**

当所有lambda 状态的总采样数大于 **weight-equil-number-samples** 指定的值时, 停止更新扩展系综权重.

- **count-ratio**

当最小和最大采样lambda 状态的采样数之间的比值大于 **weight-equil-count-ratio** 指定的值时, 停止更新扩展系综权重.

simulated-tempering: (no)

启用或关闭模拟回火. 模拟回火是通过扩展系综采样实现的, 实现时使用不同的温度代替了不同的哈密顿.

sim-temp-low: (300)

模拟回火的低温值.

sim-temp-high: (300)

模拟回火的高温值.

simulated-tempering-scaling: (linear)

控制从lambda 向量的 **temperature-lambda** 部分计算中间lambda 对应温度的方式.

- **linear**

使用 **temperature-lambda** 的值对温度进行线性内插, 即, 若 **sim-temp-low**=300, **sim-temp-high**=400,

则lambda=0.5 对应的温度为350. 非线性的温度设定总可以通过不均匀间距的lambda 实现.

- **geometric**

在 **sim-temp-low** 和 **sim-temp-high** 之间对温度进行几何内插. 第i个状态的温度为 **sim-temp-low** * $(\text{sim-temp-high} / \text{sim-temp-low})^{(i/(ntemps-1))}$ 次方. 对恒定热容, 这种方法应该给出大致相等的交换, 尽管那些涉及蛋白质折叠的模拟具有很高的热容峰.

- **exponential**

在 **sim-temp-low** 和 **sim-temp-high** 之间对温度进行指数内插. 第i个状态的温度为 **sim-temp-low** + $(\text{sim-temp-high} - \text{sim-temp-low}) * (\exp(\text{temperature-lambdas}[i]) - 1) / (\exp(1.0) - 1)$.

7.3.25 非平衡MD

acc-grps:

具有恒定加速度的组(如 **Protein Sol**), **Protein** 和 **Sol** 组中的所有原子都将具有恒定的加速度, 加速度的值在 **accelerate** 行中指定.

accelerate: (0) [nm ps⁻²]

acc-grps 的加速度, 对每个组有x, y和z三个分量(例如, **0.1 0.0 0.0 -0.1 0.0 0.0** 意味着, 第一组在X方向具有恒定的加速度0.1 nm ps⁻², 第二组的加速度与第一组相反).

freezegrps:

群组将被冻结(即其X, Y, 和/或Z位置不会被更新;如 脂质SOL).freezedim 指定哪个维度的冻结申请.为了避免 虚假 contributions 的维里和压力, 由于之间完全大部队 冷冻原子, 你需要使用能源集团排除, 这也节省了计算时间. 需要注意的是冷冻原子的坐标不被压耦合算法缩放.

freezedim:

尺寸为这组freezegrps 应该被冻结, 指定Y或N为X, Y 与Z和为每个组(例如YYNNNN意味着, 在第一组中的粒子可以仅在Z方向移动.在第二组中的粒子可以以任何方向移动).

cos-acceleration: (0) [nm ps⁻²]

所述加速度曲线的振幅来计算粘度.加速度为 在X方向和大小是COS-加速度COS(2π的z / boxheight). 两 术语被添加到能量的文件: 速度分布和第1 /粘度的振幅.

deform: (0 0 0 0 0) [nm ps⁻¹]

变形对框要素的速度: A(X)B(y) 的C(Z)B(X)C(X)C(Y).每一步 为其变形的框元件是非零的计算公式为: 箱(TS)+(叔 TS)*变形, 非对角元素被用于校正周期性.的坐标变换与符合 多地.冷冻自由度都(故意)也改变.时间Ts设定为t时 第一步, 在步骤在其中x和v被写入轨迹保证精确重新启动. 变形可以与semiisotropic 或各向异性压力耦合时, 可以使用适当的可压缩被设置为零.对角元素可以用来对应变 固体.非对角元素可用于剪切固体或液体.

7.3.26 电场

E-x; **E-y**; **E-z**:

如果你想在某个方向上使用电场, 在适当的 **E-*** 后输入3个数字. 第一个数字: 余弦的数目, 只实现了单个余弦项(频率为0), 所以输入1; 第二个数字: 电场强度, 以 **V nm⁻¹** 为单位; 第三个数字: 余弦的相位, 你可以在这里输入任何数字, 因为频率为零的余弦没有相位.

E-xt; **E-yt**; **E-zt**:

尚未实现

混合量子/经典分子动力学

QMMM:

- **no**
无QM/MM.

- **yes**
QM/MM模拟. 可以使用不同水平的QM单独对几个组进行描述, 在 **QMMM-grps** 域中指定这些组, 彼此之间以空格隔开. 各个组使用的从头算方法的水平在 **QMmethod** 和 **QMbasis** 域中指定. 使用不同水平的方法对组进行描述只能与ONIOM QM/MM一起使用, 由 **QMMMscheme** 指定.

QMMM-grps:

QM水平的组

QMMMscheme:

- **normal**
正常的QM/MM. 只能对一个 **QMMM-grps** 使用从头算方法进行描述, 方法的水平通过 **QMmethod** 和 **QMbasis** 指定. 体系的其余部分处于MM水平. QM和MM两个子体系的相互作用如下: MM部分的点电荷包含在QM部分的单电子哈密顿算符中, 所有的Lennard-Jones 相互作用都在MM水平进行描述.
- **ONIOM**
使用Morokuma 及其同事发展的ONIOM方法对子体系之间的相互作用进行描述. 可以有一个以上的 **QMMM-grps**, 每个组可以使用不同级别的QM(**QMmethod** 和 **QMbasis**)进行描述.

QMmethod: (RHF)

用于计算QM原子的能量和梯度的方法. 可用的方法包括AM1, PM3, RHF, UHF, DFT, B3LYP, MP2, CASSCF 和 MMVB. 对CASSCF, 电子数和活化空间的轨道数分别由 **CASElectrons** 和 **CASorbitals** 指定.

QMbasis: (STO-3G)

用于展开电子波函数的基组. 目前只可使用高斯基组, 即STO-3G, 3-21G, 3-21G, 3-21+G, 6-21G, 6-31G, 6-31G, 6-31+G和6-311G.
.....

QMcharge: (0) [整数]

QMMM-grps 的总电荷数, 以 **e** 为单位. 在有一个以上 **QMMM-grps** 的情况下, 需要单独指定每个ONIOM层的总电荷.

QMmult: (1) [整数]**QMMM-grps**的多重度. 在有一个以上**QMMM-grps**的情况下, 需要单独指定每个ONIOM层的多重度.**CASorbitals**: (0) [整数]

进行CASSCF计算时包含在活化空间中的轨道数.

CASelectrons: (0) [整数]

进行CASSCF计算时包含在活化空间中的电子数.

SH:

- **no**

无势能面跳跃. 体系总处于电子基态..

- **yes**

在激发态势能面进行QM/MM的MD模拟, 在模拟过程中, 当体系碰到锥形交叉线时, 强制 非绝热 跳跃到基态. 此选项只能与CASSCF方法联合使用.

7.3.27 隐式溶剂

implicit-solvent:

- **no**

不使用隐式溶剂

- **GBSA**

使用基于广义Born 公式的隐式溶剂进行模拟. 共有三种不同的方法可用以计算Born 半径: Still, HCT 和 OBC, 模拟时所用的方法可在**gb-algorithm**行中指定. 非极性溶剂化可通过**sa-algorithm**指定.**gb-algorithm**:

- **Still**

用Still 方法计算Born 半径

- **HCT**

使用Hawkins-Cramer-Truhlar 方法计算Born 半径

- **OBC**

使用Onufriev-Bashford-Case 方法计算Born 半径

nstgbradii: (1) [步]

(重新)计算Born 半径的频率. 对于大多数实际模拟, 设置大于1的值会破坏能量守恒并导致轨迹不稳定.

rgbradii: (1.0) [nm]

计算Born半径的截断值. 目前必须等于 **rlist**

gb-epsilon-solvent: (80)

隐式溶剂的介电常数

gb-saltconc: (0) [M]

隐式溶剂模型的盐浓度, 目前并未使用

gb-obc-alpha (1); **gb-obc-beta** (0.8); **gb-obc-gamma** (4.85);

OBC模型的缩放因子. 默认值为OBC(II). OBC(I)的值分别为0.8, 0和2.91.

gb-dielectric-offset: (0.009) [nm]

计算Born半径时介电偏移的距离, 为每个原子的中心与相应原子极化能量中心之间的偏移

sa-algorithm

- **Ace-approximation**

使用Ace类型的近似(默认)

- **None**

不计算非极性溶剂化. 对GBSA只计算极性部分

sa-surface-tension: [kJ mol⁻¹ nm⁻²]

SA算法中表面张力的默认值. 默认值为-1. 注意, 如果不改变此默认值, **grompp** 将会使用与选择的半径算法相应的值覆盖默认值(Still: 0.0049 kcal/mol/Å², HCT/OBC: 0.0054 kcal/mol/Å²). 将此值设置为0, 并使用SA算法意味着不计算非极性部分.

7.3.28 自适应分辨率模拟

adress: (no)

是否开启AdResS功能.

adress-type: (off)

- **Off**

AdResS模拟的权重为1, 相当于一个显式(正常)的MD模拟. 与禁用AdResS的区别在于, 仍会读入并定义AdResS变量.

- **Constant**

AdResS模拟的权重为常数, 权重值由 **adress-const-wf** 定义

- **XSplit**

AdResS模拟时沿x方向劈分模拟盒, 因此基本上权重只是x坐标的函数, 所有距离都只使用x坐标进行测量.

- **Sphere**

AdResS模拟的显式区域为球形.

adress-const-wf: (1)

常权重模拟的权重(**adress-type=Constant**)

adress-ex-width: (0)

显式区域的宽度, 从**adress-reference-coords**开始测量.

adress-hy-width: (0)

混合区域的宽度.

adress-reference-coords: (0,0,0)

显式区域的中心位置. 测量与它的距离时会使用周期性边界条件.

adress-cg-grp-names:

粗粒能量组的名称. 所有其他能量组都被认为是显式的, 它们之间的相互作用会被自动从粗粒组中排除.

adress-site: (COM)

用于计算权重的映射点.

- **COM**

权重由每个电荷组的质心来计算.

- **COG**

权重由每个电荷组的几何中心来计算.

- **Atom**

权重由每个电荷组中第一个原子的位置来计算.

- **AtomPerAtom**

权重由每个单独的原子的位置来计算.

adress-interface-correction: (off)

- **off**

不使用任何界面校正.

- **thermoforce**

使用热力学力的界面校正. 可以使用**mdrun**的**-tabletf**选项指定表格. 该表格应包含(作用在分子上的)势能和力, 它们是**adress-reference-coords**中距离的函数.

adress-tf-grp-names

若启用 `adress-interface-correction` 时, 施加 `thermoforce` 校正的能量组的名称. 如果没有给出组, 会应用到默认的表.

`adress-ex-forcecap`: (0)

在混合区域对力进行平齐, 对大分子有用. 0禁用力平齐.

7.3.29 用户自定义项

`user1-grps`; `user2-grps`:

`userint1` (0); `userint2` (0); `userint3` (0); `userint4` (0)

`userreal1` (0); `userreal2` (0); `userreal3` (0); `userreal4` (0)

如果修改了源代码你就可以使用这些. 你可以传递整数和实数到你的子程序. 检查

`src/include/types/inputrec.h` 中 `inputrec` 的定义.

第八章 分析

本章将对分析模拟轨迹的各种不同方法进行讨论，并给出相应的分析程序的名称。与这些程序相关的输入输出的具体信息，可以利用www.gromacs.org 的在线手册进行查阅。输出文件通常都是Grace/Xmgr 的图形格式。

首先，在8.1节对分析中使用的组的概念进行解释。8.1.2节将解释一个新的概念—动态选择—只有一些分析工具支持这一概念。然后，将对不同的分析工具进行讲解。

8.1 使用组

`gmx make_ndx`, `gmx mk_angndx`, `gmx select`

在第三章，我们解释了如何在 `mdrun` 中使用 原子组 (参看3.3节)。对大多数分析程序，你也必须选择原子组。虽然大多数程序都能产生一些默认的索引组，但这些程序总是可以从索引文件中读入组。让我们考虑一个具体的例子，A和B双组分混合物的模拟。当我们需要计算A对B的径向分布函数(RDF, Radial Distribution Function) $g_{AB}(r)$ 时，我们必须计算

$$4\pi r^2 g_{AB}(r) = V \sum_{i \in A}^{N_A} \sum_{j \in B}^{N_B} P(r) \quad (8.1)$$

其中， V 为体积， $P(r)$ 为距离A原子 r 处发现B原子的概率。

通过让用户在一个简单的文件中定义组A和B的 原子编号，我们就能以最通用的方式计算 g_{AB} ，而无须在RDF程序中对粒子类型做任何假定。

因此，组可以包含一系列 原子编号，但在一些情况下也可以包含 分子编号。我们还可以利用 三联原子编号 指定一系列键角，利用 四联原子编号 指定一系列二面角，利用 原子编号对 指定一系列键或(分子中的)矢量

需要对分析程序指定索引文件的类型。许多程序都可帮你创建这样的索引文件(`index.ndx`)，它们在创建时使用你输入的构型或拓扑信息。要创建包含一系列 原子编号 (如 g_{AB} 例子中的)的索引文件，可以使用 `gmx make_ndx` 或 `gmx select`。要创建键角或二面角的索引文件，可以使用 `gmx mk_angndx`。当然，你也可以手动创建这些索引文件。索引文件的通用格式如下：

```
[ Oxygen ]
1 4 7

[ Hydrogen ]
2 3 5 6
8 9
```

可以看到，组的名称写在方括号中。随后的原子编号从1开始，可以延续任意多行。

每个支持组的工具都会列出所有可选择的组, 供用户通过组的编号或名称进行选择. 实际上, 如果可以与其他的组区分开来, 使用组名称的头几个字母就足够了. 利用Unix shell 可以在命令行中选择组名称, 而不是以通常的交互方式. 请查看www.gromacs.org 给出的建议.

8.1.1 默认组

当没有为分析工具或 `grompp` 提供索引文件时, 会产生一些默认组以供选择:

System: 体系中所有的原子

Protein: 所有的蛋白质原子

Protein-H: 除氢原子外的所有蛋白质原子

C-alpha: C_{α} 原子

Backbone: 蛋白质骨架原子, N, C_{α} 和C

MainChain: 蛋白质主链上的原子: N, C_{α} , C和O, 包括C端的氧原子

MainChain+Cb: 蛋白质主链上的原子, 包含 C_{β}

MainChain+H: 蛋白质主链上的原子, 包括骨架氨基氢原子和N端氢原子

SideChain: 蛋白质支链上的原子, 即除N, C_{α} , C, O, 骨架氨基氢原子, C端氧原子, N端氢原子以外的所有原子

SideChain-H: 除氢原子外的蛋白质支链上的原子

Prot_Masses: 除哑质量(创建NH₃基团和色氨酸支链时使用的虚位点)外的蛋白质原子, 请参看5.2.2节. 只有当与 **Protein** 组不同时, 才包含此组

Non-Protein: 所有的非蛋白质原子

DNA: 所有的DNA原子

RNA: 所有的RNA原子

Water: 水分子(名称如SOL, WAT, HOH等). 参看 `residuetypes.dat` 文件中的完整列表

non-Water: 不属于 **Water** 组的任何粒子

Ion: 匹配 `residuetypes.dat` 文件中离子条目的任何名称

Water_and_Ions: **Water** 和 **Ion** 组的组合

molecule_name: 所有不能被识别为蛋白质, DNA或RNA的残基/分子. 会为每个残基/分子创建一个组

Other: 不属于蛋白质, DNA, RNA 的所有原子

不会创建空的组. 大部分组只包含蛋白质原子. 如果一个原子的残基名称存在于 **residuetypes.dat** 文件中, 并且被视为“蛋白质”, 这个原子就被视为蛋白质原子. 决定DNA原子, RNA原子等的方法类似. 如果需要修改这些分类, 你可以从库文件目录复制一份文件到你的工作目录, 然后编辑这个副本.

8.1.2 选择

gmx select

目前, 一些分析工具支持一个扩展的概念 (动态)选择, 它与传统的索引组主要存在三个区别:

- 通过文本指定选择, 而不是从文件中读入固定的原子编号. 选择使用的语法类似于VMD. 文本可以交互地输入, 也可以过命令行提供, 或是使用文件.
- 选择并不局限于原子, 也可以是分析的对象, 如一组原子的质心位置. 一些工具可能并不支持那些非单个原子的选择, 例如, 当它们需要的信息, 如原子名称或类型之类, 只对单个原子可用时.
- 选择可以是动态的, 即, 对不同的轨迹帧选择不同的原子. 这样就可以只对体系满足一定的几何条件的那部分进行分析.

作为动态选择的一个简单例子, **resname ABC and within 2 of resname DEF** 将选择残基名称为ABC, 且距残基DEF中任何一个原子都不超过2 nm的所有原子.

对支持动态选择的工具也可以使用传统的索引文件, 像以前的工具一样: 可以为工具指定一个 **.ndx** 文件, 通过组编号或组名称直接从索引文件中选择一个组. 索引组也可以作为更复杂选择的一部分使用.

作为开始, 你可以使用单个结构运行 **gmx select**, 利用交互的提示试验不同的选择. 这个工具提供了一些输出选项, **-on** 和 **-ofpdb** 可以分别将选择的原子写入索引文件和 **.pdb** 文件中. 这些选项不允许测试质心位置的选择, 但可以测试其他选择, 并检查结果.

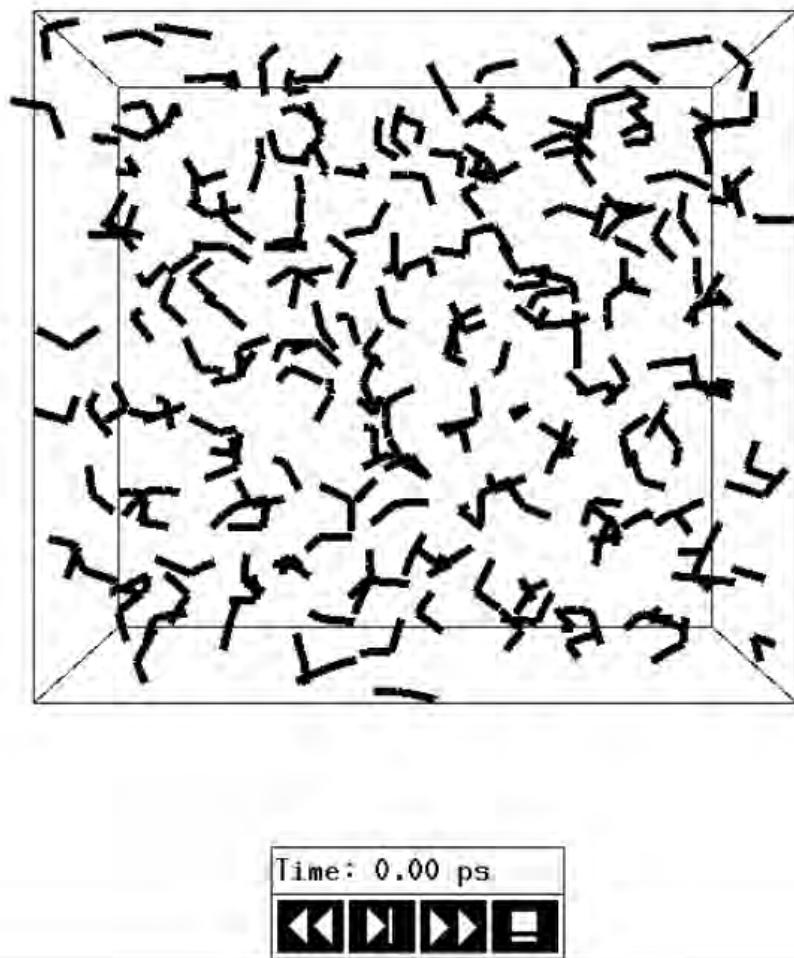
如果想了解选择的详细语法和各个关键词, 对任何支持选择的工具你可以在交互提示下键入 **help**, 或使用 **gmx help selections** 命令. 帮助被划分为不同的子主题, 每个子主题可以通过像 **help syntax / gmx help selections syntax** 这样的命令查看. 一些单独的选择关键词还有其自己的扩展帮助, 可通过像 **help keywords within** 这样的命令进行查看.

目前的交互提示没有提供很多的编辑功能, 如果需要它们, 你可以在 **rlwrap** 下运行程序.

对那些不支持选择语法的工具, 你可以使用 **gmx select -on** 生成静态的索引组并传递给工具. 但是, 这种作法只能使用那些完全支持动态选择的工具所提供的灵活性中的一小部分(仅仅前面三项中的第一项).

为了利用这些选择的优点, 你也可以编写自己的分析工具, 具的例子请参看安装目录下 **share/gromacs/template** 中的 **template.cpp** 文件.

Water: That Was Pretty Cool

图 8.1 显示水盒子的 `gmx view` 窗口

8.2 查看你的轨迹

`gmx view`

在进行分析之前, 先查看一下你的轨迹通常很有帮助. GROMACS提供了一个简单的轨迹查看器 `gmx view`. 这个查看器的优点是它不需要OpenGL的支持, 有些机器—如一些超级计算机—通常不支持OpenGL. 你也可以利用查看器生成EPS格式的轨迹(图 8.1). 如果你需要更快更花哨的查看器, 有一些程序可以读取GROMACS格式的轨迹, 请查看我们[主页](#)上更新的链接.

8.3 通用性质

`gmx energy`, `gmx traj`

如果需要分析部分或全部的 能量 和其他性质, 如 总压力, 压力张量, 密度, 盒子体积 和 盒子尺寸, 请使用 `gmx energy` 程序. 可以从一系列能量列表中进行选择, 如势能, 动能或总能, 也可以从对能量单独贡献的部分中进行

选择, 如Lennard-Jones 或二面角能量.

质心速度定义为

$$\mathbf{V}_{com} = \frac{1}{M} \sum_{i=1}^N m_i \mathbf{v}_i \quad (8.2)$$

其中 $M = \sum_{i=1}^N m_i$ 为体系的总质量. 可以利用程序 `gmx traj -com -ov` 对质心进行监测. 然而我们建议模拟时在每一步都移除质心的速度(参看第三章)!

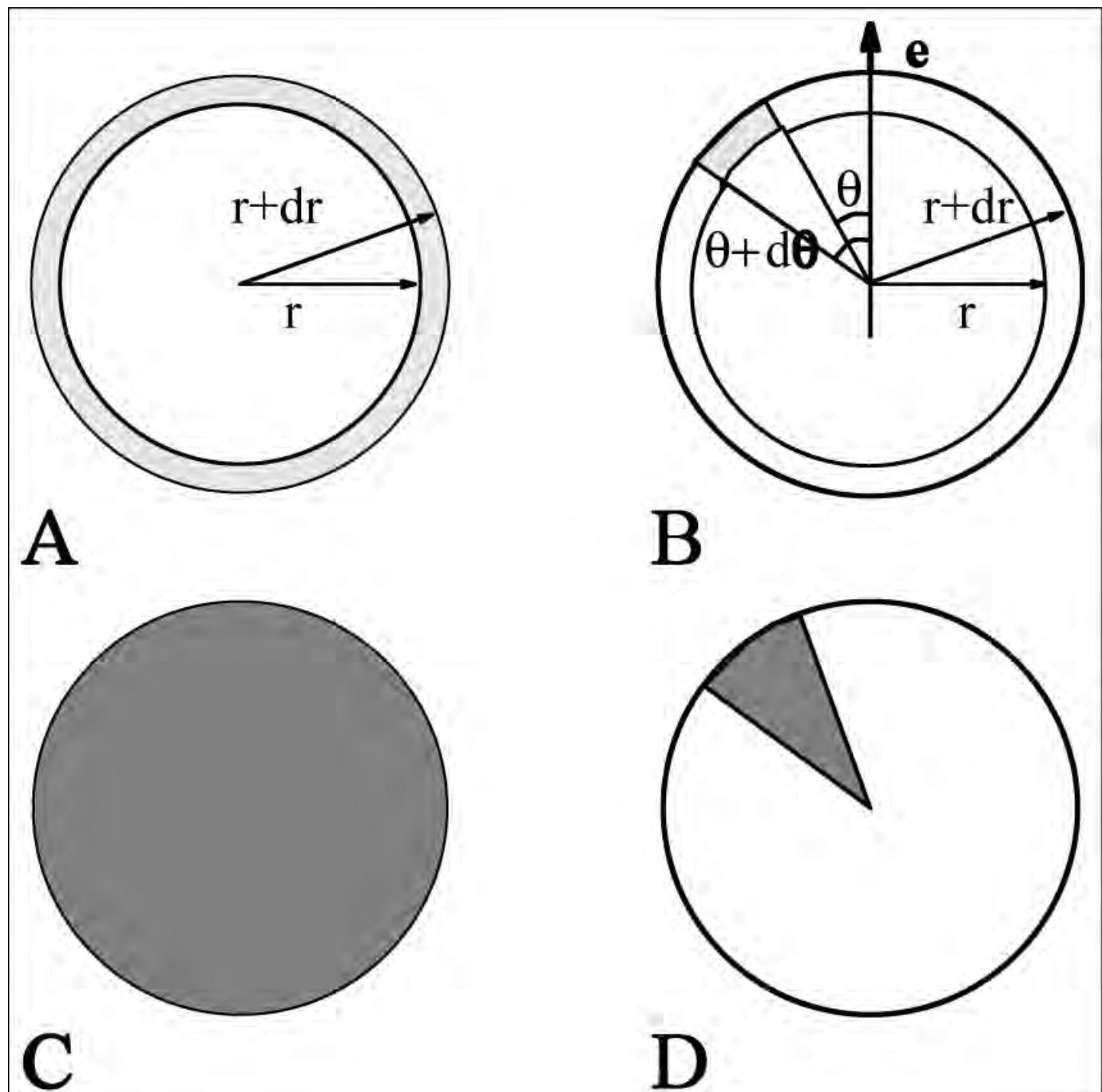


图 8.2: `gmx rdf` 中切片的定义: A. $g_{AB}(r)$. B. $g_{AB}(r, \theta)$. 切片为灰色, C. $\langle \rho_B \rangle_{local}$ 的归一化. D. $\langle \rho_B \rangle_{local,\theta}$ 的归一化. 灰色为归一化体积.

8.4 径向分布函数

`gmx rdf`

A类型粒子与B类型粒子之间的径向分布函数(RDF)或称对关联函数 $g_{AB}(r)$ 的定义为

$$\begin{aligned} g_{AB}(r) &= \frac{\langle \rho_B(r) \rangle}{\langle \rho_B \rangle_{local}} \\ &= \frac{1}{\langle \rho_B \rangle_{local}} \frac{1}{N_A} \sum_{i \in A}^{N_A} \sum_{j \in B}^{N_B} \frac{\delta(r_{ij} - r)}{4\pi r^2} \end{aligned} \quad (8.3)$$

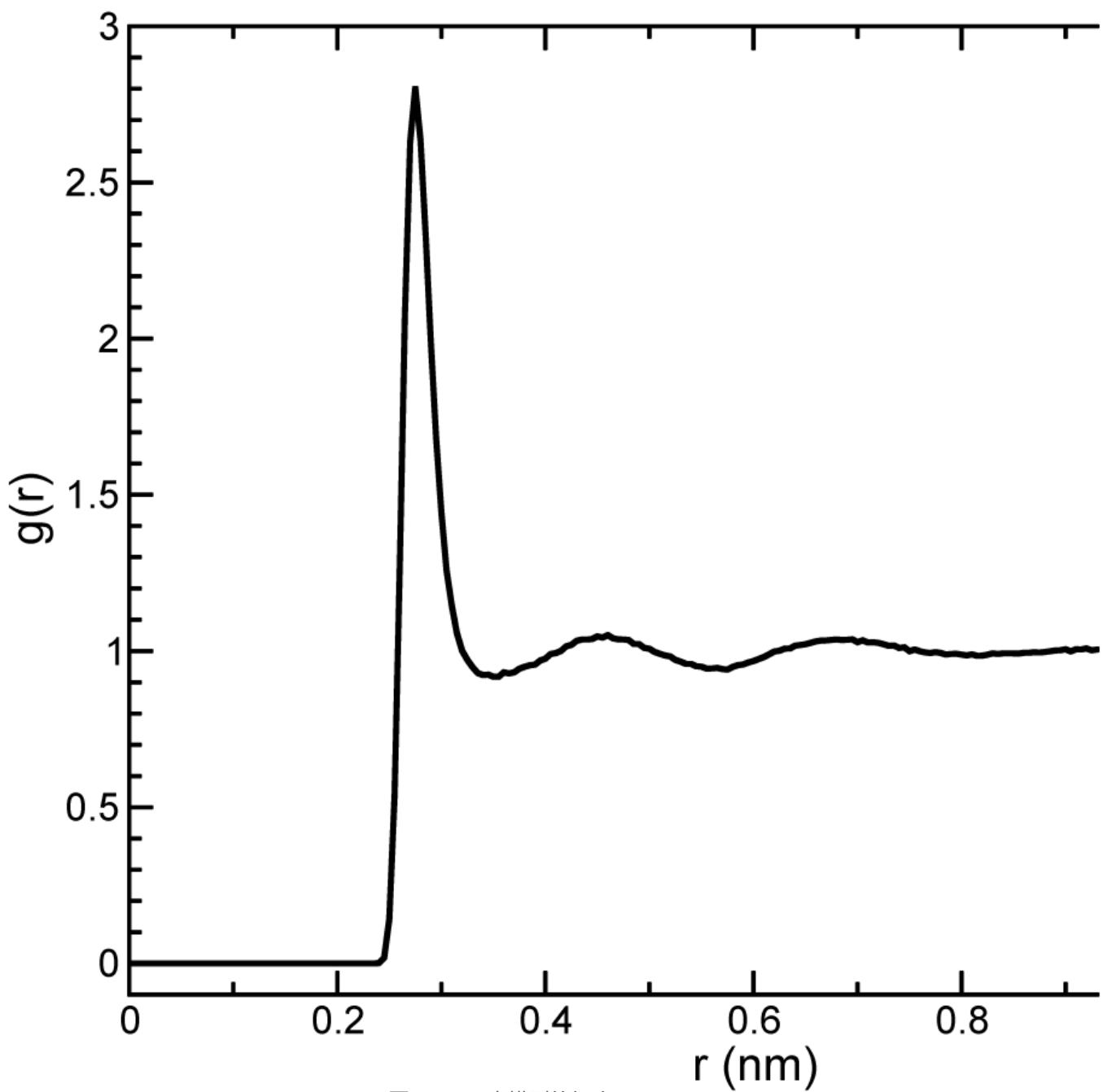
其中, $\langle \rho_B(r) \rangle$ 为距离A粒子 r 处B粒子的密度, $\langle \rho_B(r) \rangle_{local}$ 为所有以A离子为中心, 半径 r_{max} 的壳层内B离子的平均密度(图 8.2C)

通常 r_{max} 的值取为盒子长度的一半. 计算时也会对时间进行平均. 具体计算时, 分析程序 **gmx rdf** 会将体系划分为球形切片(从 r 到 $r + dr$, 图 8.2A), 并生成一个直方图而不是 δ 函数. SPC水模型氧-氧RDF[81]的一个例子如图 8.3.

也可以使用 **gmx rdf** 计算角度相关的RDF $g_{AB}(r, \theta)$, 其中角度 θ 根据特定的实验室坐标轴 e 进行定义, 参看图 8.2B.

$$g_{AB}(r, \theta) = \frac{1}{\langle \rho_B \rangle_{local, \theta}} \frac{1}{N_A} \sum_{i \in A}^{N_A} \sum_{j \in B}^{N_B} \frac{\delta(r_{ij} - r) \delta(\theta_{ij} - \theta)}{2\pi r^2 \sin(\theta)} \quad (8.4)$$

$$\cos(\theta_{ij}) = \frac{\mathbf{r}_{ij} \cdot \mathbf{e}}{\|\mathbf{r}_{ij}\| \|\mathbf{e}\|} \quad (8.5)$$

图8.3 SPC水模型的氧-氧RDF $g_{\text{OO}}(r)$

$g_{AB}(r, \theta)$ 在分析各向异性体系时很有用。注意在这种情况下, 归一化因子 $\langle \rho_B \rangle_{local, \theta}$ 为一直到 r_{max} 的所有从 θ 到 $\theta + d\theta$ 的角度切片的平均密度, 因此与角度相关。参看图 8.2D

8.5 相关函数

8.5.1 相关函数的理论

文献[106]对相关函数的理论作了很好的介绍。我们在这里只介绍GROMACS代码中对相关函数实现的各种细节。性质 $f(t)$ 的自相关函数(ACF) $C_f(t)$ 的定义为

$$C_f(t) = \langle f(\xi) f(\xi + t) \rangle_\xi \quad (8.6)$$

其中, 等号右边的记号表示对 ξ 进行平均, 即对不同时间起点进行平均。对两种性质 $f(t)$ 和 $g(t)$ 也可以计算交叉相关函数

$$C_{fg}(t) = \langle f(\xi)g(\xi + t) \rangle_\xi \quad (8.7)$$

然而, 在GROMACS中不存在计算相关函数的标准方法(注意: 你可以使用xmgr程序计算交叉相关). 相关函数对时间的积分为相关时间 τ_f :

$$\tau_f = \int_0^\infty C_f(t) dt \quad (8.8)$$

具体计算时, 相关函数是根据时间间隔为 Δt 的离散数据的计算的, 因此, MD模拟的ACF为

$$C_f(j\Delta t) = \frac{1}{N-j} \sum_{i=0}^{N-1-j} f(i\Delta t)f((i+j)\Delta t) \quad (8.9)$$

其中, N 为计算时所有可用的时间帧. 所得的ACF明显地只适用于具有相同时间间隔 Δt 的时间点. 对许多应用, 由于需要知道ACF短时间的行为(例如, 开始10 ps), 这常常意味着我们保存数据的时间间隔必须比感兴趣的时间尺度短得多. 方程8.9隐含的另一点是, 理论上我们不可能以相同精度计算所有时间点上的ACF, 因为对 $C_f(\Delta t)$ 我们有 $N - 1$ 个点, 而对 $C_f((N - 1)\Delta t)$ 只有1个点. 然而, 如果我们只计算长度为 $M\Delta t$ 的ACF, 其中 $M \leq N/2$, 我们可以同样的统计精度计算所有的点

$$C_f(j\Delta t) = \frac{1}{M} \sum_{i=0}^{N-1-M} f(i\Delta t)f((i+j)\Delta t) \quad (8.10)$$

当然, 这里的 $j \leq M$. M 有时被视为相关函数的时间延迟. 当决定这样做时, 对非常短的时间间隔($j \ll M$), 我们故意不使用所有的点, 但这使得我们更容易解释结果. 当从模拟轨迹计算ACF的时候, 不能忽略的另一点是, 通常时间起点 ξ (方程8.6)并不是统计无关的, 这可能使得结果存在偏离. 这可以利用块平均方法进行测试.

在这种方法中, 计算时只包含间距最少为时间延迟的时间起点, 例如, 使用 k 乘上间距为 $M\Delta t$ 的起点(其中 $kM \leq N$):

$$C_f(j\Delta t) = \frac{1}{k} \sum_{i=0}^{k-1} f(iM\Delta t)f((iM+j)\Delta t) \quad (8.11)$$

然而, 利用这种方式, 需要非常长的模拟才能得到高精确度的结果, 因为对ACF有贡献的点的个数少得多.

8.5.2 使用FFT计算SCF

根据方8.9计算ACF的代价正比于 N^2 , 计算量很可观. 然而, 通过使用快速Fourier变换计算卷积可以改进计算ACF的代价[106].

8.5.3 ACF的特殊形式

ACF存在一些重要的变形, 例如, 矢量 \mathbf{p} 的ACF

$$C_p(t) = \int_0^\infty P_n(\cos \angle(\mathbf{p}(\xi), \mathbf{p}(\xi + t))) d\xi \quad (8.12)$$

其中 $P_n(x)$ 为 n 阶Legendre 多项式[1]. 这样的相关时间实际上可以NMR或其他弛豫实验中获得. GROMACS可以使用一阶和二阶Legendre 多项式计算相关(方程8.12). 这种方法也可以用于旋转自相关(**gmx rotacf**)和偶极自相关(**gmx dipoles**).

为研究扭转角的动力学, 我们可以定义二面角的自相关函数[153]

$$C(t) = \langle \cos(\theta(\tau)) - \theta(\tau + t) \rangle_{\tau} \quad (8.13)$$

注意 这不是两个函数的乘积, 像通常的相关函数那样, 但可以将它重写成两个乘积加和的形式:

$$C(t) = \langle \cos(\theta(\tau)) \cos(\theta(\tau + t)) + \sin(\theta(\tau)) \sin(\theta(\tau + t)) \rangle_{\tau} \quad (8.14)$$

8.5.4 ACF的一些应用

gmx velacc 程序计算 速度自相关函数

$$C_v(\tau) = \langle \mathbf{v}_i(\tau) \mathbf{v}_i(0) \rangle_{i \in A} \quad (8.15)$$

可以利用Green-Kubo 关系计算自扩散系数[106]

$$D_A = \frac{1}{3} \int_0^{\infty} \langle \mathbf{v}_i(t) \cdot \mathbf{v}_i(0) \rangle_{i \in A} dt \quad (8.16)$$

它是速度自相关函数的积分. 人们普遍相信, 速度自相关函数比均方位移收敛得更快(8.6节), 尽管均方位移也可用于计算扩散常数. 然而, Allen 和Tildesley[106] 警告我们, 不能忽略速度自相关函数的长时间贡献, 因此计算时必须当心.

另一个重要的物理量是偶极相关时间. A类型粒子的 偶极相关函数 可利用**gmx dipoles** 计算, 其公式为

$$C_{\mu}(\tau) = \langle \mu_i(\tau) \mu_i(0) \rangle_{i \in A} \quad (8.17)$$

其中 $\mu_i = \sum_{j \in i} \mathbf{r}_j q_j$. 偶极相关时间可以使用方程8.8 计算. 对此的应用请参看[154].

液体的粘度与压力张量 \mathbf{P} 的相关时间有关联. **gmx energy** 可以计算粘度, 但计算结果并不是很精确[137], 实际上计算值并没有收敛.

8.6 均方位移

gmx msd

为决定A类型粒子的自扩散系数 D_A , 可以使用Einstein 关系式[106]:

$$\lim_{t \rightarrow \infty} \langle \|\mathbf{r}_i(t) - \mathbf{r}_i(0)\|^2 \rangle_{i \in A} = 6 D_A t \quad (8.18)$$

可使用**gmx msd** 程序计算 均方位移 和 D_A . 计算时通常使用一个包含原子编号的索引文件, 计算的MSD对这些原

子进行平均。对含有多个原子的分子, \mathbf{r}_i 可取为分子的质心位置。在这种情况下, 你需要使用包含分子编号的索引文件。但结果几乎与原子平均的结果完全相同。`gmx msd` 程序也可用于计算一维或二维的扩散, 这对研究界面间的水平扩散很有用。

SPC水模型均方位移的一个例子如图8.4所示。

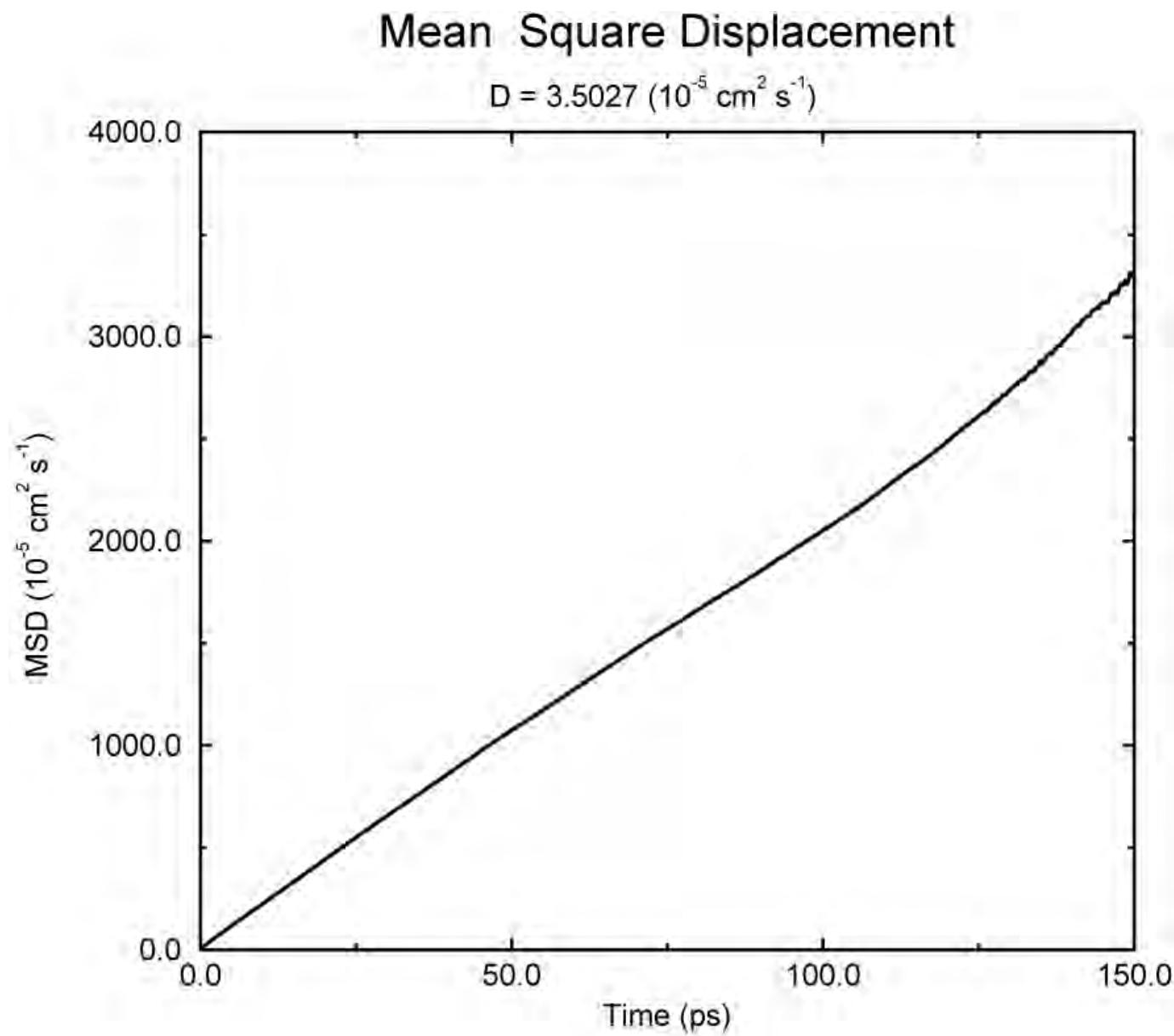


图 8.4 SPC水模型的均方位移

8.7 键/距离, 键角和二面角

`gmx distance`, `gmx angle`, `gmx gangle`

为监测分子中特定的键, 或更一般的说法, 不同点之间的距离, 程序 `gmx distace` 可以计算距离随时间的变化以及距离的分布。使用传统的索引文件时, 组里面应该包含一些原子编号对, 例如:

```
[ bonds_1 ]
1   2
3   4
9   10
```

[bonds_2]
12 13

索引文件也支持选择, 第一行的两个位置定义了第一个距离, 第二个位置对定义了第二个距离, 以此类推. 你可以计算所有残基中CA和CB原子间的距离(假定每个残基或者含有这两类原子, 或者不含有任何一类), 选择的方法是:

name CA CB

选择也允许计算更广义的距离. 例如, 要计算两个残基质心之间的距离, 你可以使用

com of resname AAA plus com of resname BBB

gmx angle 程序计算 键角 和 二面角 的时间分布, 还会给出平均键角或二面角. 计算时索引文件中需要包含三联或是四联原子编号:

```
[ angles ]
1 2 3
2 3 4
3 4 5

[ dihedrals ]
1 2 3
2 3 5
```

对二面角, 你可以使用“生化约定”($\phi = 0 \equiv cis$)或“聚合物决定”($\phi = 0 \equiv trans$), 参考图8.5.

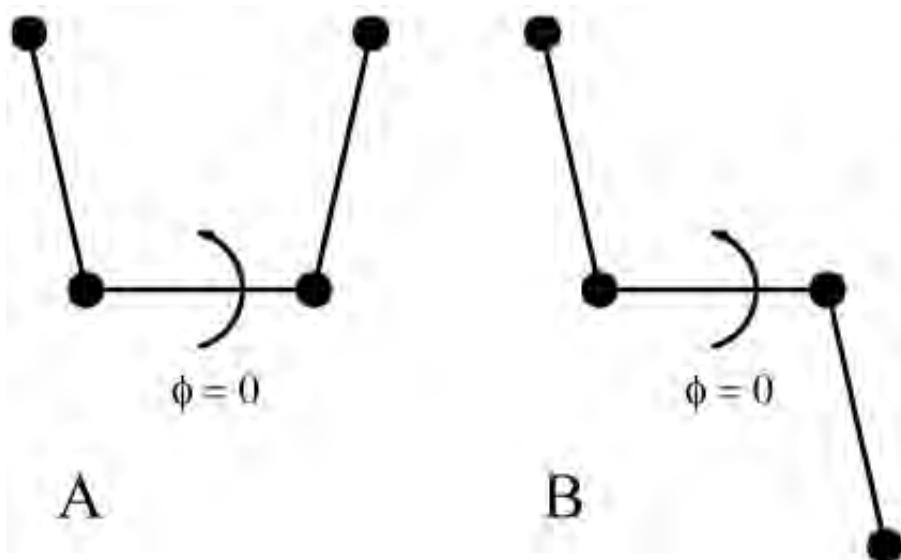


图 8.5 不同的二面角约定: A. “生化约定”. B. “聚合物约定”.

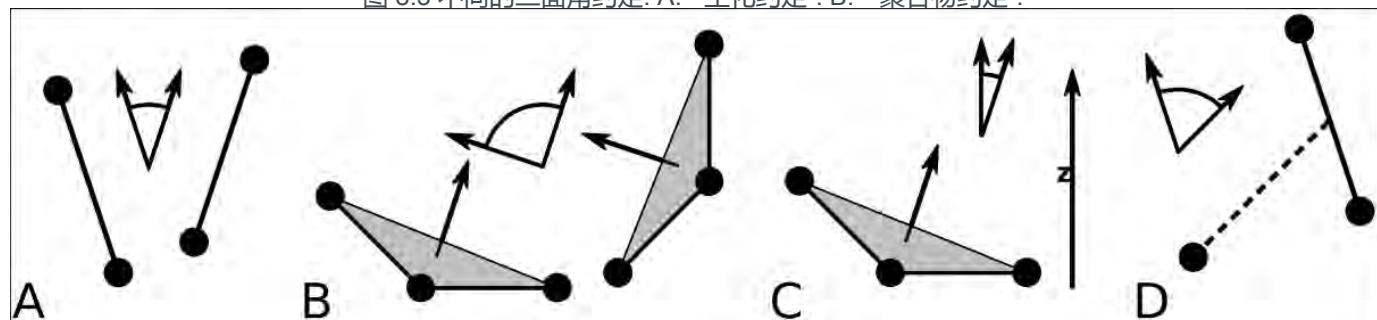


图 8.6 **gmx gangle** 的角度选项: A. 两个矢量间的夹角. B. 两个平面间的夹角. C. 矢量和 z 轴间的夹角. D. 矢量和球的法向间的夹角. 也支持其他组合: 平面和矢量可以交换使用.

gmx gangle 程序计算角度时支持选择. 这个工具也可用于计算角和二面角, 但不能支持 **gmx angle** 的所有选项, 比如自相关或其他时间序列分析. 另外, 它支持两个矢量间的夹角, 矢量与平面, 两个平面(分别由2或3个点定义), 矢量/平面与 z 轴, 或矢量/平面与球的法向(由单一位置决定)间的夹角. 此外还支持矢量/平面相对于其第一帧位置的夹角. 对于平面, **gmx gangle** 使用的法向矢量垂直于平面. 参看图8.6 A, B, C 中的定义.

8.8 回旋半径与距离

gmx gyrate, **gmx distance**, **gmx mindist**, **gmx mdmat**, **gmx xpm2ps**

作为结构密实度的粗略表征, 你可以利用 **gmx gyrate** 程序计算 回旋半径:

$$R_g = \left(\frac{\sum_i \|\mathbf{r}_i\|^2 m_i}{\sum_i m_i} \right)^{1/2} \quad (8.19)$$

其中 m_i 为原子 i 的质量, \mathbf{r}_i 为原子 i 相对于分子质心的位置. 在表征聚合物溶液和蛋白质时, 回旋半径尤其有用.

有时描绘出两个原子间 距离 或是两个原子组(例如: 盐桥中的蛋白质支链)之间 最小 距离的图形很有趣, 一些组之间距离的计算可采用下面的方法:

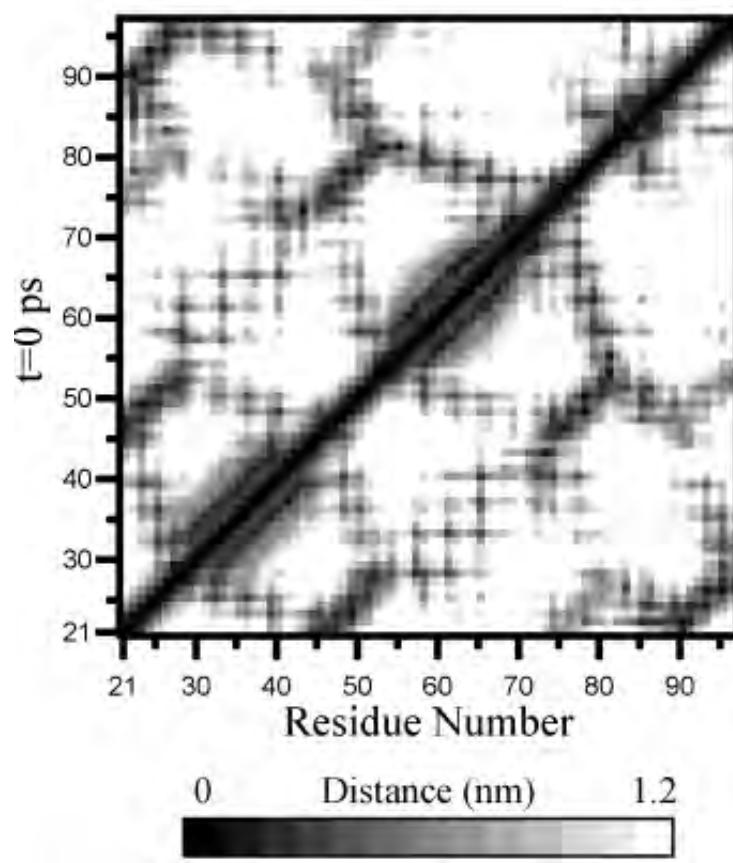


图 8.7:多肽的最小距离矩阵[157]

- 两个组的 几何中心之间的距离 可以使用 **gmx distance** 程序计算, 参看8.7节的论述.
- 不同时刻两个原子组之间的 最小距离 可以使用 **gmx mindist** 计算. 它同时也会计算这些组之间在一定半径 r_{max} 内的 接触数.

- 为监测一个(蛋白质)分子中氨基酸残基之间的最小距离, 你可以使用 `gmx mdmat` 程序. 两个残基 A_i 和 A_j 之间的最小距离的定义为任何一对原子($i \in A_i, j \in A_j$)之间的最小距离. 程序将输出一个所有残基之间最小距离的对称矩阵. 你可以使用 `xv` 等程序对此矩阵进行可视化. 如果想查看坐标轴和图例或者打印矩阵, 你可以使用 `xpm2ps` 将它转换为PS图片, 如图8.7.

画出不同时刻的矩阵, 你就可以分析结构的变化和其他一些信息, 如盐桥的形成.

8.9 结构的根均方偏差

`gmx rms, gmx rmsdist`

一个分子中的某些原子相对于参考结构的 根均方偏差(RMSD) 可以利用 `gmx rms` 程序计算, 计算时先使用最小二乘方法

将结构拟合到参考结构($t_2 = 0$), 再根据下式(方程8.20)计算 RMSD

$$RMSD(t_1, t_2) = \left[\frac{1}{M} \sum_{i=1}^N m_i \| \mathbf{r}_i(t_1) - \mathbf{r}_i(t_2) \|^2 \right]^{1/2} \quad (8.20)$$

其中 $M = \sum_{i=1}^N m_i$, $\mathbf{r}_i(t)$ 为原子 i 在 t 时刻的位置. 注意 计算 RMSD 时拟合并不需要使用相同的原子. 例如, 蛋白质通常利用骨架原子(N, C_α, C)进行拟合, 但 RMSD 可以基于骨架原子计算, 也可以基于整个蛋白质计算.

与把结构与 $t = 0$ 时刻的初始结构(例如一个晶体结构)进行比较不同, 你也可以使用 $t_2 = t_1 - \tau$ 时刻的结构计算方程 8.20. 其结果会给出迁移性与 τ 关系的洞察. 作为 t_1 和 t_2 函数的 RMSD 可以组成一个矩阵, 它给出了一条轨迹的漂亮的图形化解释. 如果轨迹中存在转变, 它们将在矩阵中清楚地显示出来.

作为替代, 也可以使用 `gmx rmsdist` 程序计算不拟合的 RMSD

$$RMSD(t) = \left[\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \| \mathbf{r}_{ij}(t) - \mathbf{r}_{ij}(0) \|^2 \right]^{1/2} \quad (8.21)$$

其中, t 时刻原子间的距离 \mathbf{r}_{ij} 是与0时刻相同原子之间的距离相比

8.10 协方差分析

协方差分析也被称为主成分分析或主成分动力学[158], 利用它可以发现相关的运动. 它使用原子坐标的协方差矩阵 C 进行计算

$$C_{ij} = \langle M_{ii}^{1/2} (x_i - \langle x_i \rangle) M_{jj}^{1/2} (x_j - \langle x_j \rangle) \rangle \quad (8.22)$$

其中 M 为包含原子质量的对角矩阵(质量加权分析)或单位矩阵(非质量加权分析). C 是对称的 $3N \times 3N$ 矩阵, 可以利用正交变换矩阵 R 对其进行对角化

$$R^T C R = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{3N}), \quad \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{3N} \quad (8.23)$$

R 的列就是本征矢量, 也称为主模或基本模. R 定义了一个到新坐标系的变换. 将轨迹投影到主模上便得到主成分 $p_i(t)$

$$\mathbf{p}(t) = R^T M^{1/2} (\mathbf{x}(t) - \langle \mathbf{x} \rangle) \quad (8.24)$$

本征值 λ_i 为主成分 i 的均方涨落, 头几个主模通常描述了体系聚集的整体运动. 可以利用一个(或多个)主模对轨迹进行过滤. 对一个主模 i , 其方式如下

$$\mathbf{x}^f(t) = \langle \mathbf{x} \rangle + M^{-1/2} R_{*i} p_i(t) \quad (8.25)$$

当对大分子进行分析时, 通常想要移除整体的转动和平动以便仅仅观察内部运动. 这可以通过最小二乘拟合到参考结构实现. 需要注意的是, 参考结构对系统需要具有代表性, 因为参考结构的选择会影响协方差矩阵.

始终都应该检查是否已经很好地定义了主模. 如果第一个主成分类似于半个余弦, 第二个主成分类似于整个余弦, 你可能正在过滤噪声(参见下文). 检查头几个主模相关的一个好方式是计算模拟前半部分和后半部分采样的重叠. 注意这只适用于两部分使用相同的参考结构的情况.

文献[159]定义了一个很好的重叠度量. 协方差矩阵的元素正比于位移的平方, 因此需要取矩阵的平方根来检查取样的范围. 可以从本征值 λ_i 和本征矢量计算矩阵的平方根, 本征矢量对应于矩阵 R 的列. 对一个对称, 对角占优, 大小为 $3N \times 3N$ 的矩阵 A , 其平方根为

$$A^{1/2} = R \text{diag}(\lambda_1^{1/2}, \lambda_2^{1/2}, \dots, \lambda_{3N}^{1/2}) R^T \quad (8.26)$$

容易证明这个矩阵及其自身的乘积为矩阵 A . 现在我们可以定义协方差矩阵 A 和 B 之间的差异 d

$$d(A, B) = \sqrt{\text{tr}((A^{1/2} - B^{1/2})^2)} \quad (8.27)$$

$$= \sqrt{\text{tr}(A + B - 2A^{1/2}B^{1/2})} \quad (8.28)$$

$$= \left(\sum_{i=1}^N (\lambda_i^A + \lambda_i^B) - 2 \sum_{i=1}^N \sum_{j=1}^N \sqrt{\lambda_i^A \lambda_j^B} (R_i^A \cdot R_j^B)^2 \right)^{1/2} \quad (8.29)$$

其中, tr 为矩阵的迹. 重叠 s 的定义为

$$s(A, B) = 1 - \frac{d(A, B)}{\sqrt{\text{tr}A + \text{tr}B}} \quad (8.30)$$

当且仅当矩阵 A 和 B 全等时, 重叠为1. 当抽样的子空间完全正交时, 重叠为0.

一个常用的度量是协方差矩阵头几个本征矢量子空间的重叠. m 个正交矢量 $\mathbf{w}_1, \dots, \mathbf{w}_m$ 张成的子空间和 n 个正交矢量 $\mathbf{v}_1, \dots, \mathbf{v}_n$ 张成的参考子空间之间的重叠可定量为

$$\text{overlap}(\mathbf{v}, \mathbf{w}) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m (\mathbf{v}_i \cdot \mathbf{w}_j)^2 \quad (8.31)$$

重叠会随着 m 的增加而增大, 当集合 \mathbf{v} 为集合 \mathbf{w} 的子空间时, 重叠为1. 这个方法的缺点在于没有考虑到本征值. 所有本征矢量的权重都相同, 当存在简并的子空间时(本质值相等), 计算的重叠太小.

另一个有用的检查是余弦含量. 已经证明随机扩散的主成分是余弦, 且其周期数等于主成分索引数的一半[160,159]. 本征值正比于索引数的-2次方. 余弦含量定义为

$$\frac{2}{T} \left(\int_0^T \cos\left(\frac{i\pi t}{T}\right) p_i(t) dt \right)^2 \left(\int_0^T p_i^2(t) dt \right)^{-1} \quad (8.32)$$

当头几个主成分的余弦含量接近1时, 最大的涨落与势能没有联系, 而是对应于随机扩散.

gmx covar 可以构建并对角化协方差矩阵. 可以利用 **gmx anaeig** 对主成分和重叠(还有许多其他量)进行绘图和分析. 余弦含量可以利用 **gmx analyze** 计算.

8.11 二面角主成分分析

gmx angle, gmx covar, gmx anaeig

使用GROMACS可以在二面角空间[161]中进行主成分分析. 首先使用 **gmx mk_angndx** 或其他方法, 在索引文件中定义要计算的二面角, 然后使用 **gmx angle** 程序及其 **-or** 选项产生一个新的 **.trr** 文件, 其中包含了每个二面角分别在两个坐标系下的正弦和余弦值. 即, 在 **.trr** 文件中, 有相应于 $\cos(\phi_1), \sin(\phi_1), \cos(\phi_2), \sin(\phi_2), \dots, \cos(\phi_n), \sin(\phi_n)$ 的一系列数字, 必要时, 会对数字序列补零. 然后, 你可以使用这个 **.trr** 文件作为 **gmx covar** 程序的输入, 像通常一样进行主成分分析. 为此, 你需要生成一个参考文件(**.tpr, .gro, .pdb** 等), 其中包含与新的 **.trr** 文件相同数目的“原子”, 即, 对 n 个二面角, 你需要 $2n/3$ 个原子(如果不是整数就舍去小数部分). 由于参考文件中的坐标与 **.trr** 文件中的信息毫无对应关系, 你需要使用 **gmx covar** 程序的 **-nofit** 选项. 对结果的分析可使用 **gmx anaeig**.

8.12 氢键

gmx hbond

gmx hbond 程序用于分析所有可能的施体D和受体A之间的 氢键 (HB). 分析时使用几何准则决定氢键的存在与否, 参看图 8.8:

$$\begin{aligned} r &\leq r_{HB} = 0.35 \text{ nm} \\ \alpha &\leq \alpha_{HB} = 30^\circ \end{aligned} \quad (8.33)$$

参考值 $r_{HB} = 0.35\text{nm}$ 对应于SPC水模型RDF的第一极小位置(参看图 8.3).

gmx hbond 程序以下面的方式分析两组原子(它们必须相同或没有重叠)或指定的施体-氢-受体之间所有可能存在的氢键:

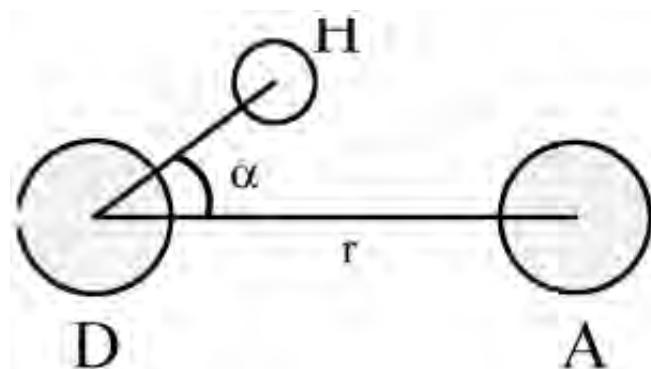


图8.8 氢键的几何准则

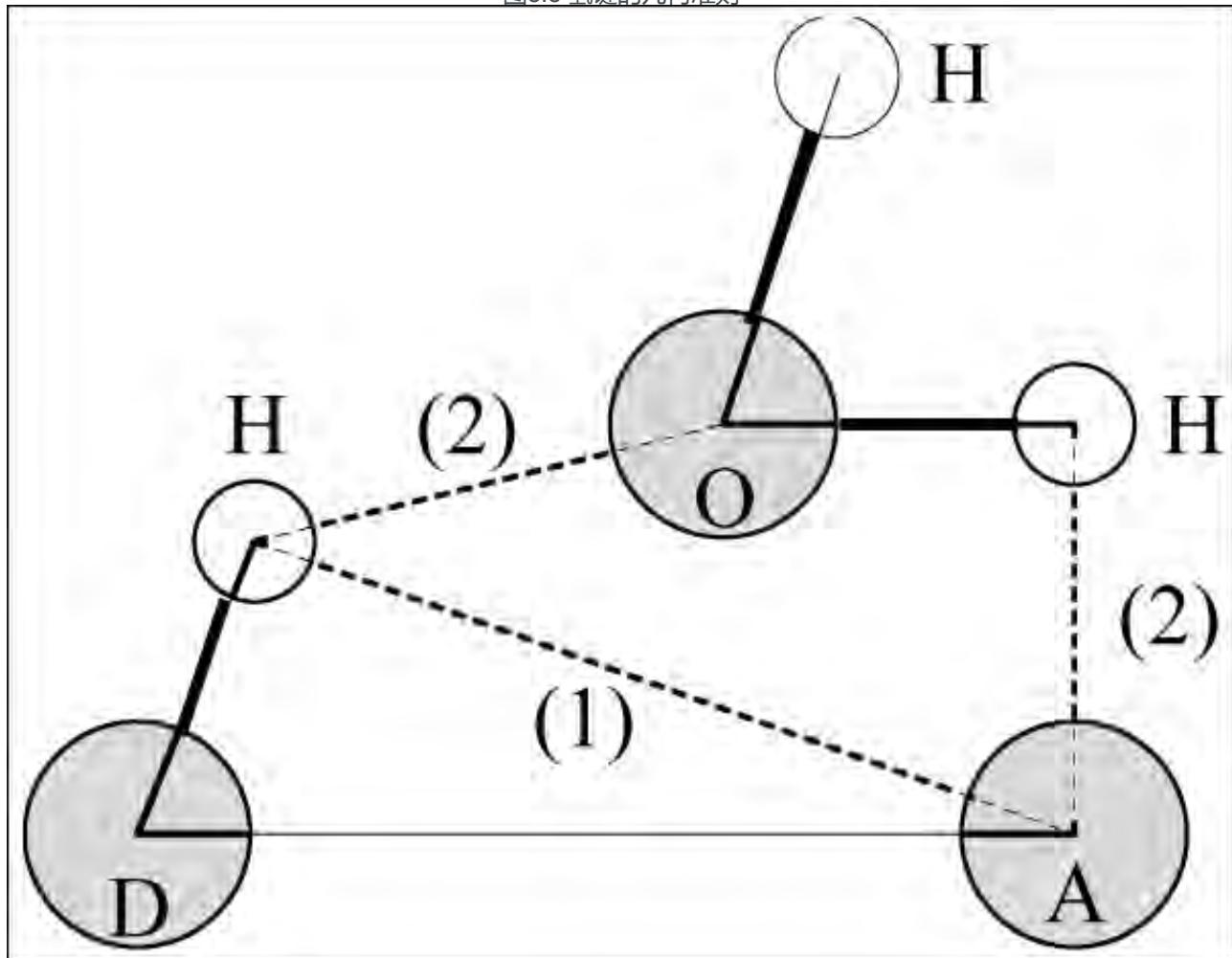


图 8.9: 水对氢键的插入. (1) 两残基间正常的氢键. (2) 通过一个水分子形成的氢键桥.

- 所有氢键中施体-受体距离的分布
- 所有氢键中氢-施体-受体之间夹角的分布
- 每个时间帧中氢键的总数目
- 残基之间实时的氢键数目, 划分到组 $n - n + i$, 其中 n 和 $n + i$ 代表残基编号, i 从0到6. $i = 6$ 的组也包括 $i > 6$ 的所有氢键. 这些组包括 $n - n + 3, n - n + 4$ 和 $n - n + 5$ 氢键, 提供了 α 螺旋, β 转角或 β 股形成的度量.
- 对所有氢键生存函数(0或1)的自相关函数进行平均, 可以计算氢键的寿命

$$C(\tau) = \langle s_i(t)s_i(t + \tau) \rangle \quad (8.34)$$

对 t 时刻的氢键 i , $s_i(t) = \{0, 1\}$. $C(\tau)$ 的积分给出了氢键平均寿命 τ_{HB} 的粗略估计

$$\tau_{HB} = \int_0^\infty C(\tau) d\tau \quad (8.35)$$

程序会输出积分和完整的自相关函数 $C(\tau)$, 以便能够进行更复杂的分析(例如使用多指数拟合)得到 τ 的更佳估计值. 参考文献[162]给出了一个更完整的分析, 一个更花哨的选项是氢键动力学的Luzar 和Chandler分析[163, 164].

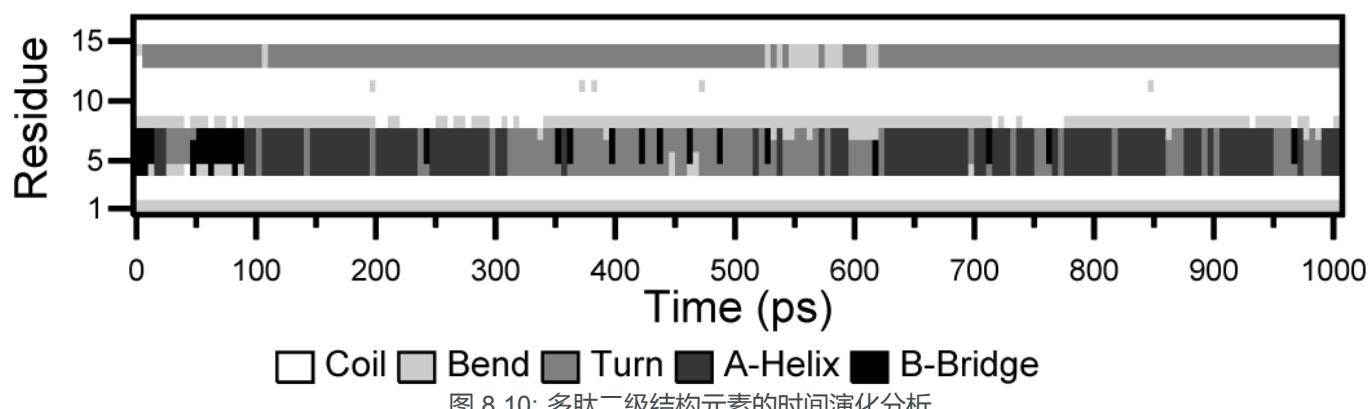
- 生成大小为 $\#HB \times \#frames$ 的氢键存在映射图. 顺序与索引文件中的完全相同(见下文), 但反过来, 这意味着索引文件中的最后一个三联原子编号相应于存在映射图的第一行.
- 输出包含分析组的索引组, 这些组中所有的施体-氢原子对和受体原子, 分析组之间氢键中的施体-氢-受体三联对, 所有插入的溶剂分子.

8.13 与蛋白质相关的项

`gmx do_dssp`, `gmx rama`, `gmx wheel`

为分析蛋白质结构的变化, 你可以计算回旋半径或随时间变化的最小残基距离(参看8.8节), 或计算RMSD(参看8.9节).

在运行中你也可以观察 二级结构元素 的变化, 为此, 你可以使用 `gmx do_dssp` 程序, 它是商业程序 `DSSP` [165] 的接口界面. 更多的信息请参看 `DSSP` 的手册. 图 8.10 给出了一个典型的 `gmx do_dssp` 输出图.



另一个重要的蛋白质分析是所谓的 Ramachandran图, 它是结构在蛋白质骨架的两个二面角 ϕ 和 ψ 形成的平面上的投影. 参看图8.11.

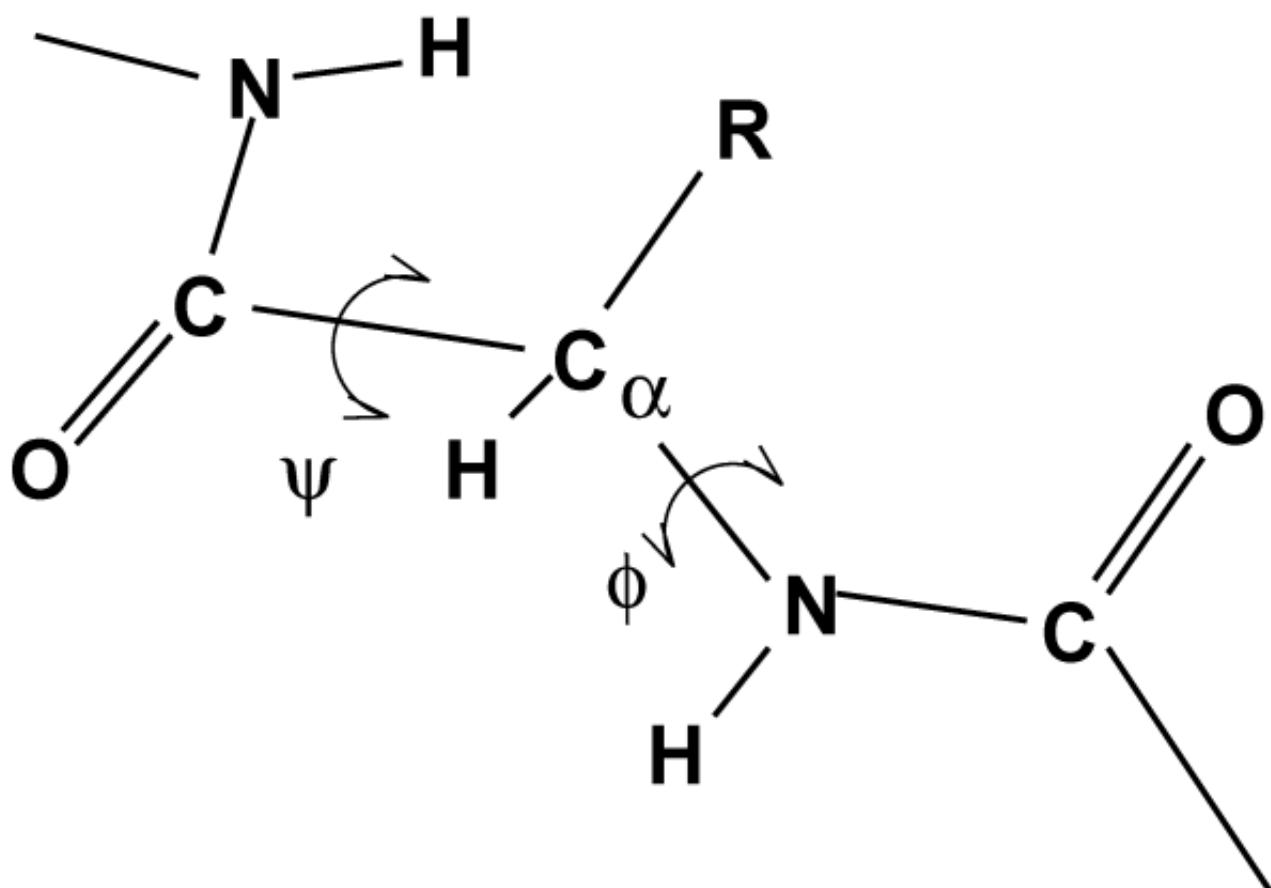


图 8.11: 蛋白质骨架的二面角 ϕ 和 ψ 的定义

你可以使用 `gmx rama` 获得 Ramachandran 图. 图 8.12 是一个典型的输出.

Ramachandran Plot

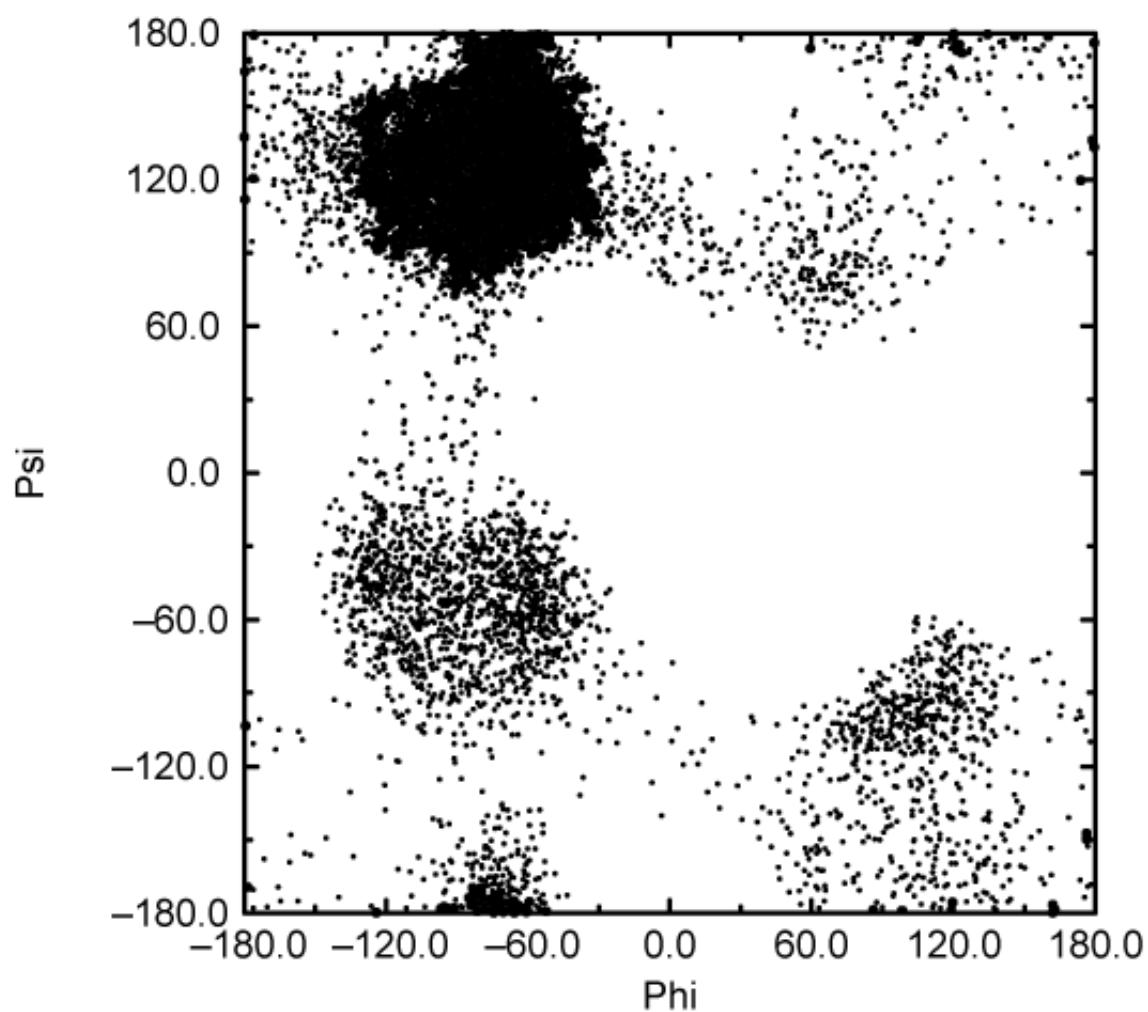


图 8.12: 一个小蛋白质分子的 Ramachandran 图

当研究 α 螺旋时, 多肽的 旋轮投影 很有用, 它可用于查看一个多肽是否是两性的. 你可以使用 `gmx wheel` 程序获得这样的投影图. 图 8.13给出了两个例子.

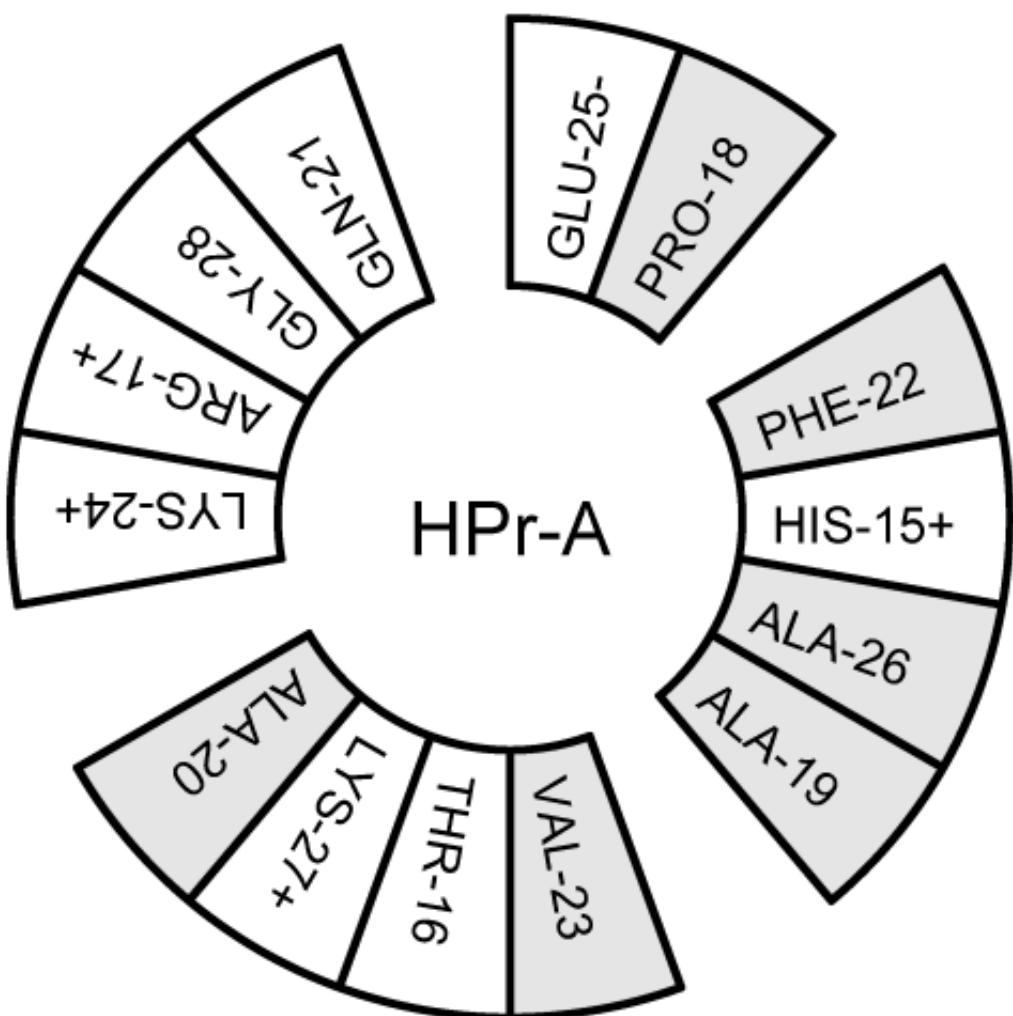


图 8.13: HPr N端螺旋的旋轮投影图

8.14 与界面相关的项

gmx order, gmx density, gmx potential, gmx traj

当模拟具有长碳尾的分子时, 可以计算它们的平均取向. 有许多不同的序参数, 它们大多数是相关的. **gmx order** 程序可利用下面的方程计算序参数

$$S_z = \frac{3}{2} < \cos^2 \theta_z > - \frac{1}{2} \quad (8.36)$$

其中 θ_z 为模拟盒子的 z 轴与待研究的分子轴之间的夹角. 后者被定义为从 C_{n-1} 到 C_{n+1} 的矢量. 可以同样的方式定义参数 S_x 和 S_y . 尖括号隐含着要对时间和分子进行平均. 序参数可以在1(沿界面法向完全有序)和-1/2(完全垂直于界面法向)之间变化, 值为零时代表取向具有各向同性.

程序可以完成两件事. 它可以分别计算每个 CH_2 片段的序参数, 对所有三个轴的序参数, 或将盒子划分为切片并计算一个切片内每个片段序参数的平均值.

第一个方法给出了分子从头到尾如何排列的说明, 第二个方法给出了排列和盒子长度的关系.

从一条轨迹中可以计算出穿过界面的静电势(ψ), 通过电荷密度($\rho(z)$)的双重积分

$$\psi(z) - \psi(-\infty) = - \int_{-\infty}^z dz' \int_{-\infty}^{z'} \rho(z'') dz'' / \epsilon_0 \quad (8.37)$$

其中, 位置 $z = -\infty$ 在体相中足够远以致场强为零. 使用这种方法, 可以将总的势能“划分”为层和水分子的独立贡献. **gmx potential** 程序将盒子划分为切片, 并对每个切片中原子的所有电荷进行加和. 然后程序会积分电荷密度得到电场, 对电场进行积分给出势能. 电荷密度, 电场和势能会写入 **xvgr** 输入文件中.

gmx traj 是一个非常简单的分析程序, 它所做的是打印选中原子的坐标, 速度, 或力. 它也可以计算一个或多个分子的质心, 并将质心坐标打印到三个文件中. 这个分析程序本身可能并不是很有用, 但获得了选择的分子或原子的坐标后很利于进行进一步的分析, 不仅仅对于界面体系是这样.

gmx density 程序计算组的质量密度, 并给出沿盒子轴向的密度分布. 对于观察组的分布或穿过界面的原子, 这很有用.

1. $P_0(x) = 1, P_1(x) = x, P_2(x) = (3x^2 - 1)/2 \leftarrow$

附录A 技术细节

A.1 混合精度或双精度

可以使用混合精度或双精度编译GROMACS。老版本GROMACS的文档将混合精度称为“单精度”，但选择性地使用双精度已实现很多年了。对所有变量都使用单精度将导致精度显著降低。尽管在“混合精度”中，所有的状态向量，即粒子坐标、速度和力，都是单精度的，但关键变量仍是双精度的。后者的一个典型例子是维里，它是对体系中所有力的加和，正负可变。另外，在代码的许多地方，通过改变求和顺序或重新组合数学表达式，我们设法避免对双精度进行算术运算。默认配置使用混合精度，但通过为 `cmake` 添加选项 `-DGMX_DOUBLE=on`，可以很容易地启用双精度。双精度比混合精度慢20%到100%，具体数字取决于运行机器的架构。双精度会使用更多的内存和运行输入，而且能量和全精度轨迹文件的大小几乎是混合模式的两倍。

混合精度的能量精确到最后一位小数，力的最后一一位或两位数字并不重要。维里的精度比力的要差，因为维里只比所有原子加和中的各个元素大一个数量级(参见B.1节)。对大多数情况，这真的不是一个问题，因为维里的涨落可以比其平均值大两个数量级。对库仑相互作用使用截断会导致能量，力和维里出现大的误差。即便使用反应场或晶格加和方法，误差也大于或近于因局部使用单精度导致的误差。由于MD具有混沌性，具有非常相似起始条件的轨迹也将迅速发散，混合精度的发散速度高于双精度。

对大多数模拟，混合精度足够精确。在某些情况下，需要使用双精度以获得合理的结果：

- 简正分析，其中的共轭梯度或l-bfgs 最小化，计算和对角化力常数矩阵
- 长时间的能量守恒，特别是对于大的体系

A.2 环境变量

可以使用环境变量影响GROMACS程序的运行。首先，在 `GMXRC` 文件中设置的变量对运行和编译GROMACS至关重要。以下各节列出了其他一些有用的环境变量。通过在你的shell中设置非空值，大多数环境变量可以起作用。如果需要设置其他的值，参考下面的具体要求。你应该查看自己所用shell的文档以了解如何为当前shell设置环境变量，或者如何为以后的shell设置配置文件。注意，将环境变量输出到批处理控制系统作业中的要求各不相同，你应该查看本地文档的详情说明。

输出控制

1. `GMX_CONSTRAINTVIR`: 打印能量项中的约束维里和力维里。
2. `GMX_MAXBACKUP`: 当尝试写到具有相同名称的新文件时，GROMACS自动备份以前文件的副本。此变量控制备份的最大数量，默认为99。当设置为0时，若输出文件已存在，则无法运行。当设置为-1时，会覆盖所有输出文件，不进行备份。
3. `GMX_NO_QUOTES`: 如果明确设置了此变量，不会在程序结束后打印有趣的引用。

4. **GMX_SUPPRESS_DUMP**: 禁用每步的文件转储, 当(例如)约束算法失效引起体系崩溃时.
5. **GMX_TPI_DUMP**: 将相互作用能小于此环境变量设定值的全部构型转储到一个 **.pdb** 文件.
6. **GMX_VIEW_XPM**: **GMX_VIEW_XVG**, **GMX_VIEW_EPS** 和 **MX_VIEW_PDB**, 分别用于自动查看 **.xvg**, **.xpm**, **.eps** 和 **.pdb** 文件类型的命令, 默认为 **xv**, **xmgrace**, **ghostview** 和 **rasmol**. 设置为空则禁用自动查看特定的文件类型. 命令将被分叉运行于后台, 运行时具有的优先级与GROMACS工具相同(这可能不是你想要的). 注意不要使用阻断终端的命令(例如 **vi**), 因为可能会运行多个实例.
7. **GMX_VIRIAL_TEMPERATURE**: 打印能量项中的维里温度
8. **GMX_LOG_BUFFER**: 用于文件I/O的缓冲区大小. 如果设置为0, 所有文件I/O都不使用缓冲, 因此非常慢. 此变量可以方便调试, 因为它能保证所有文件始终是最新的.
9. **GMX_LOGO_COLOR**: 设置 **ngmx** 标志的显示颜色.
10. **GMX_PRINT_LONGFORMAT**: 当打印十进制数值时使用长浮点格式.
11. **GMX_COMPELDUMP**: 仅用于计算电生理学设置(参见6.6节). 初始结构转储到 **.pdb** 文件, 这样可以检查多聚体通道的PBC表示是否正确.

调试

1. **GMX_PRINT_DEBUG_LINES**: 设置后, 打印调试的行号信息.
2. **GMX_DD_NST_DUMP**: 转储当前DD到PDB文件的间隔步数(默认为0). 仅对区域分解有效, 因此典型的值为0(从不), 1(每DD相)或 **nstlist** 的倍数.
3. **GMX_DD_NST_DUMP_GRID**: 转储当前DD格点到PDB文件的间隔步数(默认为0). 仅对区域分解有效, 因此典型的值为0(从不), 1(每DD相)或 **nstlist** 的倍数.
4. **GMX_DD_DEBUG**: 每个区域分解的通用调试触发器(默认为0, 意味着禁用). 目前只检查全局-局部原子索引映射的一致性.
5. **GMX_DD_NPULSE**: 覆盖所用的DD脉冲数(默认0, 意味着不覆盖). 通常为1或2.

性能和运行控制

1. **GMX_DO_GALACTIC_DYNAMICS**: 设置此环境变量可启用行星模拟(只是为了好玩), 允许在 **.mdp** 文件中设置 **epsilon_r = -1**. 正常情况下, **epsilon_r** 必须大于零, 以防止出现致命错误. 参见 www.gromacs.org 上行星模拟的输入文件的例子.
2. **GMX_ALLOW_CPT_MISMATCH**: 设置后, 如果在 **.tpr** 文件中的系综设置与 **.cpt** 文件中的不匹配, 运行也不会退出.
3. **GMX_CUDA_NB_EWALD_TWINCUT**: 强制使用双程截断内核, 即使PP-PME负载均衡后 **rvdw=rcoulomb**. 会自动切换到双程截断内核, 所以此变量只能用于校准.
4. **GMX_CUDA_NB_ANA_EWALD**: 强制使用解析的Ewald内核. 只能用于校准.

5. **GMX_CUDA_NB_TAB_EWALD**: 强制使用表格的Ewald内核. 只能用于校准.
6. **GMX_CUDA_STREAMSYNC**: 在启用ECC功能的GPU上强制使用cudaStreamSynchronize, 由于在API v5.0 NVIDIA驱动程序(30x.xx以前版本)中存在一个已知的CUDA驱动程序缺陷, 会导致性能下降. 不能与**GMX_NO_CUDA_STREAMSYNC**同时设置.
7. **GMX_CYCLE_ALL**: 运行时对所有的代码进行计时. 不兼容线程.
8. **GMX_CYCLE_BARRIER**: 在每个循环启动/停止调用前调用MPI_Barrier.
9. **GMX_DD_ORDER_ZYX**: 设置构造区域分解格胞的顺序为(z, y, x), 而不是默认的(x, y, z).
10. **GMX_DD_USE_SENDRECV2**: 在约束和vsite通信时, 使用一对**MPI_SendRecv**调用代替两个同步的非阻塞调用(默认为0, 意味着不使用). 对一些MPI实现可能会更快.
11. **GMX_DLBBASED_ON_FLOPS**: 基于flop计数进行区域分解的动态负载均衡, 而不是基于测量到的流逝时间(默认为0, 意味着禁用). 这使得负载均衡可重复, 对调试很有帮助. 值为1时使用flop; 对>1的值将添加(值-1)*5%的噪声到flop, 以增加不均衡性和标度.
12. **GMX_DLBB_MAX_BOX_SCALING**: 每区域分解负载平衡步中所允许的盒子缩放的最大百分比(默认为10)
13. **GMX_DD_RECORD_LOAD**: 运行结束时记录DD负载统计报告(默认为1, 意味着启用)
14. **GMX_DD_NST_SORT_CHARGE_GROUPS**: 重新排序电荷组的间隔步数(默认为1). 只在区域分解中起作用, 所以典型值应为0(从不), 1(每次区域分解), 或**nstlist**的倍数.
15. **GMX_DETAILED_PERF_STATS**: 设置后, 会打印更详细的性能信息到**.log**文件. 输出结果的方式类似于4.5.x版本的性能总结, 因而可能对那些使用脚本解析**.log**文件或标准输出的人有用.
16. **GMX_DISABLE SIMD KERNELS**: 禁用特定架构SIMD优化(SSE2, SSE4.1, AVX等)的非键内核, 因而强制使用普通的C内核.
17. **GMX_DISABLE CUDA TIMING**: 当时间步长较短时, 对异步执行的GPU运算进行计时会有不可忽略的开销. 在这种情况下禁用计时能够提高性能.
18. **GMX_DISABLE GPU DETECTION**: 设置后, 禁用GPU检测, 即使**mdrun**支持GPU.
19. **GMX_DISABLE_PINHT**: 当采用英特尔超线程时, 禁止将连续线程分配到物理内核. 由**mdrun -nopinht**控制, 因而此环境变量可能会被移除.
20. **GMX_DISRE_ENSEMBLE_SIZE**: 距离约束系综平均的体系数目. 整数值.
21. **GMX_EMULATE_GPU**: 不使用GPU加速函数, 而是使用算法等价的CPU引用代码模拟GPU运行. 因为CPU代码较慢, 设置此变量仅用于调试. 如果使用**GMX_NO_NONBONDED**关闭非键计算, 会自动触发, 将不会调用非键计算, 但也会跳过CPU-GPU转移.
22. **GMX_ENX_NO_FATAL**: 当在**.edr**文件中遇到损坏的帧时禁止退出, 允许使用直到损坏的所有帧.
23. **GMX_FORCE_UPDATE**: 调用**mdrun -rerun**时更新力.

24. **GMX_GPU_ID**: 与 **mdrun** 选项 **-gpu_id** 的设置方式相同, **GMX_GPU_ID** 环境变量允许用户指定不同GPU的ID, 用于选择集群中不同计算节点上不同的设备. 不能与 **-gpu_id** 一起使用.
25. **GMX_IGNORE_FSYNC_FAILURE_ENV**: 允许 **mdrun** 继续运行, 即使文件丢失.
26. **GMX_LJCOMB_TOL**: 当设置为浮点值时, 覆盖力场浮点参数的默认容差 $1e-5$.
27. **GMX_MAX_MPI_THREADS**: 设置 **mdrun** 可以使用的最大MPI线程数.
28. **GMX_MAXCONSTRWARN**: 当设置为-1时, 即便产生了很多LINCS警告, **mdrun** 也不会退出.
29. **GMX_NB_GENERIC**: 使用通用的C内核. 如果使用基于组的截断并将 **GMX_NO_Solv_Opt** 设为true, 就应该设置此变量, 从而禁用溶剂优化.
30. **GMX_NB_MIN_CI**: 在GPU上运行时使用的邻居列表平衡参数. 对小的模拟体系, 为了改进多处理器的负载平衡从而提高性能, 设置目标配对列表的最小数目. 必须设置为正整数. 默认值是针对NVIDIA的Fermi和Kepler类型GPU进行优化的, 所以正常使用时无须改变, 但对于未来的架构可能有用.
31. **GMX_NBLISTCG**: 使用基于电荷组的邻居列表和内核.
32. **GMX_NBNXN_CYCLE**: 设置后, 打印详细的邻居搜索循环计数.
33. **GMX_NBNXN_EWALD_ANALYTICAL**: 强制使用解析Ewald的非键内核, 与 **GMX_NBNXN_EWALD_TABLE** 互斥.
34. **GMX_NBNXN_EWALD_TABLE**: 强制使用表格Ewald的非键内核, 与 **GMX_NBNXN_EWALD_ANALYTICAL** 互斥.
35. **GMX_NBNXN SIMD_2XNN**: 强制使用 $2x(N+N)$ SIMD CPU非键内核, 与 **GMX_NBNXN SIMD_4XN** 互斥.
36. **GMX_NBNXN SIMD_4XN**: 强制使用 $4xN$ SIMD CPU非键内核, 与 **GMX_NBNXN SIMD_2XNN** 互斥.
37. **GMX_NO_ALLVSALL**: 禁用优化的 all-vs-all 内核.
38. **GMX_NO_CART_REORDER**: 用于初始化区域分解通信器. 默认等级重排序, 但可利用此环境变量关闭.
39. **GMX_NO_CUDA_STREAMSYNC**: 与 **GMX_CUDA_STREAMSYNC** 相反. 当开启ECC功能的GPU使用V5.0以前版本的CUDA驱动程序API时, 禁用基于标准cudaStreamSynchronize 的GPU等待以提高性能.
40. **GMX_NO_INT**, **GMX_NO_TERM**, **GMX_NO_USR1**: 分布禁用对SIGINT, SIGTERM, SIGUSR1信号进行处理.
41. **GMX_NO_NODECOMM**: 不使用独立的节点间和节点内的通讯器.
42. **GMX_NO_NONBONDED**: 跳过非键计算; 可用于估计增加GPU加速器到当前硬件设置带来的可能的性能增益-假定完成非键计算的速度足够快, 同时CPU进行键合力和PME计算.
43. **GMX_NO_PULLVIR**: 设置后, 不将维里的贡献添加到COM牵引力.
44. **GMX_NOCHARGEGROUPS**: 禁用多原子电荷组, 即, 分配所有非溶剂分子中的每个原子到其自己的电荷组.
45. **GMX_NOPREDICT**: 不预测壳层位置.
46. **GMX_NO_Solv_Opt**: 关闭溶剂优化; 如果启用了 **GMX_NB_GENERIC** 会自动进行.

47. **GMX_NSCELL_NCG**: 每个邻区搜索格胞的理想电荷组数被硬编码为10. 将此环境变量的值设置为任何其它的整数会覆盖硬编码的值.
48. **GMX_PME_NTHREADS**: 设置OpenMP或PME的线程数(覆盖`mdrun`的猜测值).
49. **GMX_PME_P3M**: 使用P3M优化的影响函数, 而不是平滑的PME B样条内插.
50. **GMX_PME_THREAD_DIVISION**: 在三个维度上以“X Y Z”格式对PME线程进行划分. 每一维度上线程的总和必须等于PME线程的总数(在**GMX_PME_NTHREADS**之间设置).
51. **GMX_PMEONEDD**: 如果x和y方向上区域分解格胞的数目都为1, 对PME进行一维分解.
52. **GMX_REQUIRE_SHELL_INIT**: 需要初始化壳层位置.
53. **GMX_REQUIRE_TABLES**: 需要使用表格库仑和van der Waals相互作用.
54. **GMX_SCSIGMA_MIN**: 用于软核 σ 的最小值. 注意, 此值在`.mdp`文件中使用`sc-sigma`关键词进行设置, 而此环境变量可用于重现4.5前的版本对此参数的行为.
55. **GMX_TPIC_MASSES**: 应包含用于测试粒子插入空腔时的多个质量. 最后一个原子的质量中心用于插入空腔.
56. **GMX_USE_GRAPH**: 对键相互作用使用图形.
57. **GMX_VERLET_BUFFER_RES**: Verlet截断方案中缓冲区大小的分辨率. 默认值为0.001, 但可以被此环境变量覆盖.
58. **GMX_VERLET_SCHEME**: 从基于组的方案转换为Verlet截断方案, 即使`.mdp`文件中并没有将`cutoff_scheme`设置为Verlet. 此变量没有必要存在, 因为`mdrun`的`-testverlet`选项具有同样的功能, 保留它只是为了向后兼容.
59. **MPIRUN**: `g_tune_pme`使用的`mpirun`命令.
60. **MDRUN**: `g_tune_pme`使用的`mdrun`命令.
61. **GMX_NSTLIST**: 设置`nstlist`的默认值, 防止在使用Verlet截断方案启动`mdrun`时改变.
62. **GMX_USE_TREEREDUCE**: 对nbnxn力约化使用树约化. 在OpenMP线程数很多时可能更快(如果内存位置很重要).

分析和核心函数

1. **GMX_QM_ACCURACY**: Gaussian 程序L510(MC-SCF)模块的精度.
2. **GMX_QM_ORCA_BASENAME**: `.tpr`文件的前缀, 用于Orca计算的输入和输出文件名.
3. **GMX_QM_CPMSCF**: 当设置为非零值时, Gaussian 时QM计算将迭代求解CP-MCSCF方程.
4. **GMX_QM_MODIFIED_LINKS_DIR**: 修改的Gaussian 链接的位置.
5. **DSSP**: 为`do_dssp`所用, 指向`dssp`可执行程序(不只是其路径).
6. **GMX_QM_GAUSS_DIR**: Gaussian 的安装路径.
7. **GMX_QM_GAUSS_EXE**: Gaussian 可执行程序的名称.

8. **GMX_DIPOLE_SPACING**: `g_dipoles` 所用的间距.
9. **GMX_MAXRESRENUM**: 设置 `grompp` 重新编号的残基的最大数目. -1表示重新编号所有残基.
10. **GMX_FFRTP_TER_RENAME**: 有些力场(例如AMBER), 对N端和C端残基使用特定的名称(NXXX和CXXX), 像那些正常重命名的 `.rtp` 项一样. 设置此环境变量禁用此类重命名.
11. **GMX_PATH_GZIP**: `gunzip` 可执行程序的路径, 为 `g_wham` 所用.
12. **GMX_FONT**: `ngmx` 使用的X11字体的名称.
13. **GMXTIMEUNIT**: 输出文件中使用的时间单位, 可以为fs, ps, ns, us, ms, s, m 或h.
14. **GMX_QM_GAUSSIAN_MEMORY**: Gaussian QM 计算使用的内存.
15. **MULTIPROT**: `multiprot` 可执行程序的名称, 为程序 `do_multiprot` 所用.
16. **NCPUS**: Gaussian QM 计算使用的CPU数
17. **GMX_ORCA_PATH**: Orca 的安装目录.
18. **GMX_QM_SA_STEP**: Gaussian QM 计算的模拟退火步长.
19. **GMX_QM_GROUND_STATE**: 定义Gaussian 表面跳跃计算的态.
20. **GMX_TOTAL**: `total` 可执行程序的名称, 为 `do_shift` 程序所用.
21. **GMX_ENER_VERBOSE**: 让 `g_energy` 和 `eneconv` 输出更多信息.
22. **VMD_PLUGIN_PATH**: VMD插件的安装路径. 需要能够读取VMD插件认可的文件格式.
23. **VMDDIR**: 安装VMD的基准路径.
24. **GMX_USE_XMGR**: 将查看器设置为 `xmgr` (不推荐)以代替 `xmgrace`.

A.3 并行运行GROMACS

默认情况下, 将使用内置的线程MPI库编译GROMACS. 在处理单节点上线程之间的通信时, 使用此函数库比使用外部MPI库更高效. 要在多个节点, 例如一个集群上并行运行GROMACS, 你需要使用外部MPI库配置编译GROMACS. 所有的超级计算机都带有为其特定平台优化的MPI库, 也有几个好的免费MPI实现; OpenMPI通常是一个不错的选择. 注意, MPI和线程MPI支持彼此不兼容.

除MPI并行外, GROMACS还支持通过OpenMP的线程并行. MPI和OpenMP并行可以结合起来使用, 也就是所谓的杂合并行化. 在某些情况下, 杂合并行可提供更好的性能和标度.

请参看www.gromacs.org 上关于不同并行方案使用和性能的详细信息.

A.4 使用GPU运行GROMACS

自4.6版本开始, GROMACS原生支持基于CUDA的GPU. 注意, GROMACS只将计算最密集的部分, 目前也就是非键

相互作用, 分配到GPU上运行, MD计算的所有其他部分都是在CPU上进行的. 对CUDA代码的要求是, Nvidia 计算能力 ≥ 2.0 的GPU, 即至少为Fermi类. 在许多情况下, `cmake` 可自动检测GPU并自动配置GPU支持. 请传递`-DGOMX_GPU=on` 选项给 `cmake`, 并确认已配置GPU支持, . GPU的实际使用与否在 `mdrun` 运行时决定, 取决于是否有(合适的)GPU可用和运行输入设置. 支持GPU的可执行程序也可以只使用CPU进行模拟, 通过使用 `mdrun -nb cpu` 强制仅使用CPU来运行模拟. 只有使用Verlet截断方案的模拟才能在GPU上运行. 为测试以前的tpr文件在GPU上运行的性能, 可以使用 `mdrun` 的 `-testverlet` 选项, 但因为没有进行 `grompp` 的完整参数一致性检查, 你不应该使用此选项进行最终模拟. 在GPU上获得GROMACS的良好性能很容易, 但得到最佳性能很困难. 请查看www.gromacs.org 上关于GPU使用的最新信息.

附录B 一些实现细节

在本章中, 我们将介绍一些实现细节. 本章涉及的内容远远称不上完备, 但我们认为有必要澄清一些事情, 否则将很难理解.

B.1 GROMACS中的单个和维里

维里 Ξ 可以写成全张量形式:

$$\Xi = -\frac{1}{2} \sum_{i < j}^N \mathbf{r}_{ij} \otimes \mathbf{F}_{ij} \quad (\text{B.1})$$

其中 \otimes 表示两个向量的 直积.[1] 当此在MD程序的内部循环中计算直积时, 需要进行9次乘法运算和9加法运算.[2]

这里将展示如何才能将维里计算从内部循环中抽取出来[166].

B.1.1 维里

对一个周期性体系, 计算维里时必须考虑周期性:

$$\Xi = -\frac{1}{2} \sum_{i < j}^N \mathbf{r}_{ij}^n \otimes \mathbf{F}_{ij} \quad (\text{B.2})$$

其中 \mathbf{r}_{ij}^n 表示从原子 j 到原子 i 最近映象 的距离向量. 在这个定义中我们为原子 i 的位置向量 \mathbf{r}_i 添加一个 移位向量 δ_i . 差向量 \mathbf{r}_{ij}^n 因此等于:

$$\mathbf{r}_{ij}^n = \mathbf{r}_i + \delta_i - \mathbf{r}_j \quad (\text{B.3})$$

或简写为:

$$\mathbf{r}_{ij}^n = \mathbf{r}_i^n - \mathbf{r}_j \quad (\text{B.4})$$

对一个三斜体系, i 有27个可能的映象; 当使用截面八面体时, 有15个可能的映象.

B.1.2 非键力的维里

这里给出非键力子程序中单个和维里的推导. 在下面的所有公式中 $i \neq j$.

$$\Xi = -\frac{1}{2} \sum_{i < j}^N \mathbf{r}_{ij}^n \otimes \mathbf{F}_{ij} \quad (\text{B.5})$$

$$= -\frac{1}{4} \sum_{i=1}^N \sum_{j=1}^N (\mathbf{r}_i + \delta_i - \mathbf{r}_j) \otimes \mathbf{F}_{ij} \quad (\text{B.6})$$

$$= -\frac{1}{4} \sum_{i=1}^N \sum_{j=1}^N (\mathbf{r}_i + \delta_i) \otimes \mathbf{F}_{ij} - \mathbf{r}_j \otimes \mathbf{F}_{ij} \quad (\text{B.7})$$

$$= -\frac{1}{4} \left(\sum_{i=1}^N \sum_{j=1}^N (\mathbf{r}_i + \delta_i) \otimes \mathbf{F}_{ij} - \sum_{i=1}^N \sum_{j=1}^N \mathbf{r}_j \otimes \mathbf{F}_{ij} \right) \quad (\text{B.8})$$

$$= -\frac{1}{4} \left(\sum_{i=1}^N (\mathbf{r}_i + \delta_i) \otimes \sum_{j=1}^N \mathbf{F}_{ij} - \sum_{j=1}^N \mathbf{r}_j \otimes \sum_{i=1}^N \mathbf{F}_{ij} \right) \quad (\text{B.9})$$

$$= -\frac{1}{4} \left(\sum_{i=1}^N (\mathbf{r}_i + \delta_i) \otimes \mathbf{F}_i + \sum_{j=1}^N \mathbf{r}_j \otimes \mathbf{F}_j \right) \quad (\text{B.10})$$

$$= -\frac{1}{4} \left(2 \sum_{i=1}^N \mathbf{r}_i \otimes \mathbf{F}_i + \sum_{i=1}^N \delta_i \otimes \mathbf{F}_i \right) \quad (\text{B.11})$$

在上面这些公式中, 我们引入了

$$\mathbf{F}_i = \sum_{j=1}^N \mathbf{F}_{ij} \quad (\text{B.12})$$

$$\mathbf{F}_j = \sum_{i=1}^N \mathbf{F}_{ji} \quad (\text{B.13})$$

为 j 对 i 总的作用力. 因为我们使用了牛顿第三定律:

$$\mathbf{F}_{ij} = -\mathbf{F}_{ji} \quad (\text{B.14})$$

在实现时我们必须将含移位 δ_i 的项加倍.

B.1.3 分子内移位(mol-shift)

对键合力和SHAKE, 可以创建一个 mol-shift 列表, 其中存储了周期性. 我们简单地使用一个数组 `mshift`, 其中存储了每个原子在 `shiftvec` 数组中的索引.

生成此列表的算法可以从图论得到, 将分子中的每个粒子视为图中的节点, 将键视为图的边.

1. 用双向图代表键和原子
2. 令所有原子为白色
3. 使白色原子中的一个原子(i)变为黑色, 并把它放在盒子中心
4. 对 i 的每个邻居, 如果它目前为白色, 则将其变为灰色
5. 选择灰色原子中的一个(j), 相对于它的所有黑色邻居, 给它正确的周期性, 并将其变黑

6. 对 j 的每个邻居, 如果它目前为白色, 则将其变为灰色
7. 如果仍然存在任何一个灰色原子, 转到5
8. 如果仍然存在任何一个白色原子, 转到3

使用这种算法, 我们可以

- 优化键合力计算以及SHAKE
- 使用单个和方法从键合力计算维里
- 获得键的双向图表示.

B.1.4 共价键的维里

由于共价键力对维里有贡献, 我们有:

$$b = \|\mathbf{r}_{ij}^n\| \quad (\text{B.15})$$

$$V_b = \frac{1}{2} k_b (b - b_0)^2 \quad (\text{B.16})$$

$$\mathbf{F}_i = -\nabla V_b \quad (\text{B.17})$$

$$= k_b (b - b_0) \frac{\mathbf{r}_{ij}^n}{b} \quad (\text{B.18})$$

$$\mathbf{F}_j = -\mathbf{F}_i \quad (\text{B.19})$$

来源于键的维里为:

$$\Xi_b = -\frac{1}{2} (\mathbf{r}_i^n \otimes \mathbf{F}_i + \mathbf{r}_j \otimes \mathbf{F}_j) \quad (\text{B.20})$$

$$= -\frac{1}{2} (\mathbf{r}_{ij}^n \otimes \mathbf{F}_i) \quad (\text{B.21})$$

B.1.5 SHAKE的维里

SHAKE对维里有着重要贡献. 为满足约束, 力 \mathbf{G} 作用到“摇动”的粒子上. 如果此力不是来自算法(如在标准SHAKE中), 它可以在后面计算(当使用 蛙跳算法 时):

$$\Delta \mathbf{r}_i = \mathbf{r}_i(t + \Delta t) - [\mathbf{r}_i(t) + \mathbf{v}_i(t - \frac{\Delta t}{2}) \Delta t + \frac{\mathbf{F}_i}{m_i} \Delta t^2] \quad (\text{B.22})$$

$$\mathbf{G}_i = \frac{m_i \Delta \mathbf{r}_i}{\Delta t^2} \quad (\text{B.23})$$

在一般情况下, 上式对我们没有帮助. 只有当不需要使用周期性时(如刚性水), 才可以使用上面的公式, 否则我们必须在SHAKE的内部循环中增加维里的计算.

当适用时, 可以使用单个和方式计算维里:

$$\Xi = -\frac{1}{2} \sum_i^{N_c} \mathbf{r}_i \otimes \mathbf{F}_i \quad (\text{B.24})$$

其中 N_c 为约束原子的数目.

B.2 优化

在这里, 我们将描述GROMACS使用的一些算法优化, 不包括并行化. 对其中的一个, $1.0/\sqrt{x}$ 函数的实现, 我们将在B.3节分开处理. 其他最重要优化的论述如下.

B.2.1 水的内部循环

GROMACS使用特殊的内部循环来计算水分子与其它原子的非键相互作用, 使用另一组循环计算水分子之间的相互作用. 这两组循环针对两种类型的水模型进行了高度优化. 对于类似于SPC[81]的三位点模型, 即:

1. 分子中有三个原子.
2. 整个分子属于单个电荷组.
3. 第一个原子具有Lennard-Jones(4.1.1 节)和库仑(4.1.3节)相互作用.
4. 第二和第三个原子只具有库仑相互作用, 且电荷相等.

这些循环也适用于SPC/E[167]和TIP3P[125]水模型. 对类似于TIP4P[125]的四位点水模型:

1. 分子中有四个原子.
2. 整个分子属于单个电荷组.
3. 第一个原子只具有Lennard-Jones(4.1.1 节)相互作用.
4. 第二和第三个原子具有库仑相互作用, 且电荷相等.
5. 第四个原子只具有库仑相互作用.

这些实现方式的好处是, 在单个循环中有更多的浮点运算, 这意味着一些编译器可以更好地调度代码. 然而, 事实证明, 甚至一些最先进的编译器也存在调度问题, 这意味着需要进行手动调整以获得最佳性能. 这可能包括消去相同的子表达, 或移动代码到各处.

B.2.2 Fortran代码

不幸的是, 在一些平台上Fortran 编译器仍好于C编译器. 对于一些机器(例如SGI Power Challenge) 差异可高达3, 对矢量计算机差异可能更大. 因此, 针对英特尔和AMD的x86处理器, 有些占用大量计算时间的子程序被改写为Fortran 甚至汇编代码. 在大多数情况下, 当适用时, Fortran 或汇编循环会通过 `configure` 脚本自动选择, 但你也可以通过设置 `configure` 脚本的选项对此进行调整.

B.3 $1.0/\sqrt{x}$ 函数的计算

B.3.1 简介

GROMACS项目开始于开发一个 $1/\sqrt{x}$ 的处理器, 用以计算:

$$Y(x) = \frac{1}{\sqrt{x}} \quad (\text{B.25})$$

随着项目的继续, 英特尔i860处理器被用于实现GROMACS, 现在几乎已经变成了一个完整的软件项目. $1/\sqrt{x}$ 处理器的实现采用了一步的Newton-Raphson 迭代方案, 为此, 需要查表以提供初始近似值. $1/\sqrt{x}$ 函数使用了两个几乎独立的表格, 分别用于IEEE浮点表示的指数种子和分数种子.

B.3.2 通用

根据[168] $1/\sqrt{x}$ 函数可以使用Newton-Raphson 迭代方案进行计算. 反函数为:

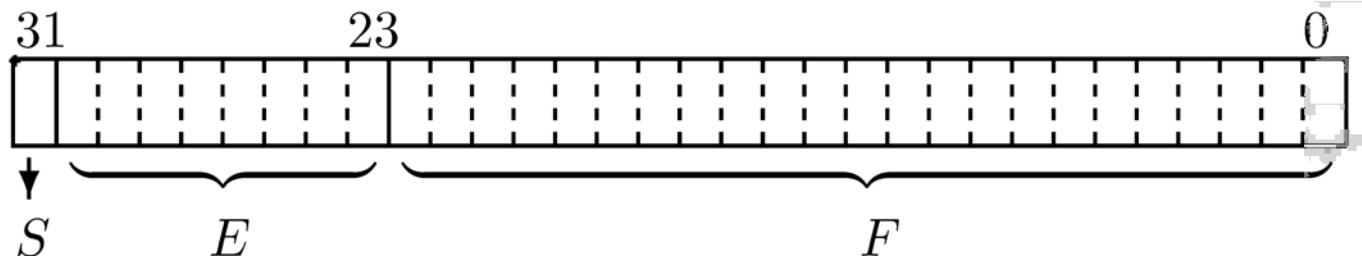
$$X(y) = \frac{1}{y^2} \quad (\text{B.26})$$

因此不直接计算

$$Y(a) = q \quad (\text{B.27})$$

而是使用Newton-Raphson 方法求解方程

$$X(q) - a = 0 \quad (\text{B.28})$$



$$\text{Value} = (-1)^S(2^{E-127})(1.F)$$

图 B.1: IEEE 单精度浮点数格式

通过计算:

$$y_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} \quad (\text{B.29})$$

进行迭代. 在这种近似下, 绝对误差 ε 被定义为:

$$\varepsilon \equiv y_n - q \quad (\text{B.30})$$

利用Taylor 级数展开来估计误差, 根据[168]的方程(3.2)得到:

$$\varepsilon_{n+1} = -\frac{\varepsilon_n^2}{2} \frac{f''(y_n)}{f'(y_n)} \quad (\text{B.31})$$

这是绝对误差的估计.

B.3.3 用于浮点数

IEEE 32位单精度格式的浮点数具有几乎恒定的相对误差 $\Delta x/x = 2^{-24}$. 从前面所述的Taylor 级数展开公式(方程B.31)可以看到, 每步迭代的误差是绝对的, 且一般与 y 有关. 如果将误差表示为相对误差, 有下面的式子

$$\varepsilon_{r_{n+1}} \equiv \frac{\varepsilon_{n+1}}{y} \quad (\text{B.32})$$

并且

$$\varepsilon_{r_{n+1}} = -\left(\frac{\varepsilon_n}{y}\right)^2 y \frac{f''}{2f'} \quad (\text{B.33})$$

对函数 $f(y) = y^{-2}, yf''/2f'$ 项为常数(等于-3/2), 因此相对误差 ε_{r_n} 与 y 无关.

$$\varepsilon_{r_{n+1}} = \frac{3}{2} (\varepsilon_{r_n})^2 \quad (\text{B.34})$$

由此得到的结论是, 函数 $1/\sqrt{x}$ 可以计算到指定的精度.

B.3.4 备查表格的要求

为了使用前面提到的迭代方案计算函数 $1/\sqrt{x}$, 很明显, 解的第一次估计必须足够精确, 这样才能获得精确的结果. 计算的要求是

- IEEE格式的最大可能精度
- 只使用一次迭代以达到最高速度

第一个要求指出, $1/\sqrt{x}$ 的结果可能具有的相对误差 ε_r 等于IEEE 32位单精度浮点数的 ε_r . 由此可以得出初始近似的 $1/\sqrt{x}$, 对后续步骤重写相对误差的定义(方程B.34):

$$\frac{\varepsilon_n}{y} = \sqrt{\varepsilon_{r_{n+1}} \frac{2f'}{yf''}} \quad (\text{B.35})$$

因此对于查表所需的精度为:

$$\frac{\Delta Y}{Y} = \sqrt{\frac{2}{3} 2^{-24}} \quad (\text{B.36})$$

这定义了备查表的宽度必须 ≥ 13 比特.

这样, 备查表的相对误差 ε_{r_n} 是已知的, 由此可以计算参数的最大相对误差. 绝对误差 Δx 定义为:

$$\Delta x \equiv \frac{\Delta Y}{Y'} \quad (\text{B.37})$$

因此:

$$\frac{\Delta x}{Y} = \frac{\Delta Y}{Y} (Y')^{-1} \quad (\text{B.38})$$

因此:

$$\Delta x = \text{constant} \frac{Y}{Y'} \quad (\text{B.39})$$

对 $1/\sqrt{x}$ 函数, 满足 $Y/Y' \propto x$, 所以 $\Delta x/x = \text{constant}$. 前面提到过, 这是所用浮点表示的性质. 备查表参数需要的精度符合:

$$\frac{\Delta x}{x} = -2 \frac{\Delta Y}{Y} \quad (\text{B.40})$$

因此, 使用浮点精度(方程B.36):

$$\frac{\Delta x}{x} = -2 \sqrt{\frac{2}{3}} 2^{-24} \quad (\text{B.41})$$

这定义了备查表的宽度必须 ≥ 12 比特.

B.3.5 指数和分数的独立计算

所使用的IEEE 32位单精度浮点格式规定, 一个数字由一个指数和一个分数表示. 上一节对每个可能的浮点数规定了备查表的长度和宽度. 精度仅由浮点数的分数部分的大小决定. 由此得到的结论是, 备查表的大小为其长度, 前面已规定, 再乘上指数的大小(2^{122^8} , 1Mb). $1/\sqrt{x}$ 函数具有指数部分与分数部分无关的性质. 如果使用浮点表示, 这很明显. 定义:

$$x \equiv (-1)^S (2^{E-127}) (1.F) \quad (\text{B.42})$$

参看图 B.1, 其中 $0 \leq S \leq 1$, $0 \leq E \leq 255$, $1 \leq 1.F < 2$, S, E, F 为整数(规范化条件). 符号位(S)可以省略, 因为 $1/\sqrt{x}$ 只对 $x > 0$ 有定义. $1/\sqrt{x}$ 函数作用于 x 得到:

$$y(x) = \frac{1}{\sqrt{x}} \quad (\text{B.43})$$

或:

$$y(x) = \frac{1}{\sqrt{(2^{E-127})(1.F)}} \quad (\text{B.44})$$

这可以改写为:

$$y(x) = (2^{E-127})^{-1/2} (1.F)^{-1/2} \quad (\text{B.45})$$

定义:

$$(2^{E'-127}) \equiv (2^{E-127})^{-1/2} \quad (\text{B.46})$$

$$1.F' \equiv (1.F)^{-1/2} \quad (\text{B.47})$$

这样 $\frac{1}{\sqrt{2}} < 1.F' \leq 1$ 成立, 因此对规范的实数表示非常重要的条件 $1 \leq 1.F' < 2$, 不再成立. 通过引入一个额外的项, 可以对此进行校正. 将 $1/\sqrt{x}$ 函数应用于浮点数(方程B.45)改写为:

$$y(x) = (2^{\frac{127-E}{2}-1})(2(1.F)^{-1/2}) \quad (\text{B.48})$$

和:

$$(2^{E'-127}) \equiv (2^{\frac{127-E}{2}-1}) \quad (\text{B.49})$$

$$1.F' \equiv 2(1.F)^{-1/2} \quad (\text{B.50})$$

这样 $\sqrt{2} < 1.F \leq 2$ 成立. 这和方程B.42中定义的规范化浮点数的精确有效范围不同. 数值2导致问题. 通过将此值映射到<2的最近表示, 可以解决这个问题. 这种近似引入的小误差在允许范围内.

指数的整数表示是下一个问题. 计算 $(2^{\frac{127-E}{2}-1})$ 会引入分数的结果, 如果 $(127 - E)$ 为奇数. 通过将计算分为奇数部分和偶数部分, 很容易处理这个问题. 对 $(127 - E)$ 为偶数的情况, 方程(方程B.49)中的 E' 作为 E 的函数, 可以使用整数运算精确地计算出来

$$E' = \frac{127 - E}{2} + 126 \quad (\text{B.51})$$

对 $(127 - E)$ 为奇数的情况, 方程(方程B.45)可以改写为:

$$y(x) = (2^{\frac{127-E-1}{2}})(\frac{1.F}{2})^{-1/2} \quad (\text{B.52})$$

因此:

$$E' = \frac{126 - E}{2} + 127 \quad (\text{B.53})$$

这也可以使用整数运算进行精确计算. 注意, 对前面提到的范围分数部分是自动校正的, 因此不需要对指数部分进行额外的校正.

由此得到的结论是:

- 分数和指数的备查表是独立的. 存在两个分数备查表(奇指数和偶指数), 因此必须使用指数的奇/偶信息(1sb比特)选择适当的表格.

- 指数备查表是一个 256×8 比特的表, 对 奇数 和 偶数 进行了初始化.

B.3.6 实现

可以使用一个小的C程序来产生备查表, 它使用了符合IEEE 32位单精度格式的浮点数和操作. 需要注意的是, 因为需要奇/偶 信息, 分数表是前面指定的两倍大(13比特 i.s.o 12 比特).

必须实现方程B.29的函数. 应用到 $1/\sqrt{x}$ 函数, 方程B.28给出:

$$f = a - \frac{1}{y^2} \quad (\text{B.54})$$

因此:

$$f' = \frac{2}{y^3} \quad (\text{B.55})$$

因此:

$$y_{n+1} = y_n - \frac{a - \frac{1}{y_n^2}}{\frac{2}{y_n^3}} \quad (\text{B.56})$$

或:

$$y_{n+1} = \frac{y_n}{2} (3 - ay_n^2) \quad (\text{B.57})$$

其中 y_0 可以在备查表中找到, y_1 给出了具有最大精度的结果. 显然, 对双精度结果只需要一步额外的迭代(以双精度进行).

-
1. $(\mathbf{u} \otimes \mathbf{v})^{\alpha\beta} = \mathbf{u}_\alpha \mathbf{v}_\beta \leftarrow$
 2. Lennard-Jones 和库仑力时计算大约需要50次浮点运算. \leftarrow

附录C 平均值与涨落

C.1 平均值计算公式

注意: 本节来源于参考文献[169].

当分析MD轨迹时, 需要计算物理量 x 的平均值 $\langle x \rangle$ 和涨落

$$\langle (\Delta x)^2 \rangle^{\frac{1}{2}} = \langle [x - \langle x \rangle]^2 \rangle^{\frac{1}{2}} \quad (\text{C.1})$$

N_x 个值 $\{x_i\}$ 的方差 σ_x , 可利用下式进行计算

$$\sigma_x = \sum_{i=1}^{N_x} x_i^2 - \frac{1}{N_x} \left(\sum_{i=1}^{N_x} x_i \right)^2 \quad (\text{C.2})$$

遗憾的是这个公式在数值上不是很精确, 尤其是当 σ_x^2 相比 x 的值为小时. 下面(等价)的表达式数值上更精确

$$\sigma_x = \sum_{i=1}^{N_x} [x_i - \langle x \rangle]^2 \quad (\text{C.3})$$

其中

$$\langle x \rangle = \frac{1}{N_x} \sum_{i=1}^{N_x} x_i \quad (\text{C.4})$$

使用方程C.2和C.4进行计算时, 需要扫描 x_i 值序列两次, 第一次确定 $\langle x \rangle$, 第二次计算 σ_x , 而等式C.1只需要按顺序扫描 $\{x_i\}$ 序列一次. 然而, 可以将方程C.2改写成另一种形式, 其中含有部分和, 这样就可以使用顺序更新算法. 定义部分和

$$X_{n,m} = \sum_{i=n}^m x_i \quad (\text{C.5})$$

和部分方差

$$\sigma_{n,m} = \sum_{i=n}^m \left[x_i - \frac{X_{n,m}}{m-n+1} \right]^2 \quad (\text{C.6})$$

可以证明

$$X_{n,m+k} = X_{n,m} + X_{m+1,m+k} \quad (\text{C.7})$$

和

$$\sigma_{n,m+k} = \sigma_{n,m} + \sigma_{m+1,m+k} + \left[\frac{X_{n,m}}{m-n+1} - \frac{X_{n,m+k}}{m+k-n+1} \right]^2 * \frac{(m-n+1)(m+k-n+1)}{k} \quad (\text{C.8})$$

对 $n = 1$ 可以发现

$$\sigma_{1,m+k} = \sigma_{1,m} + \sigma_{m+1,m+k} + \left[\frac{X_{1,m}}{m} - \frac{X_{1,m+k}}{m+k} \right]^2 \frac{m(m+k)}{k} \quad (\text{C.9})$$

并且对 $n = 1, k = 1$ (方程C.8)变为

$$\sigma_{1,m+1} = \sigma_{1,m} + \left[\frac{X_{1,m}}{m} - \frac{X_{1,m+1}}{m+1} \right]^2 m(m+1) \quad (\text{C.10})$$

$$= \sigma_{1,m} + \frac{[X_{1,m} - mx_{m+1}]^2}{m(m+1)} \quad (\text{C.11})$$

其中我们已经使用了关系

$$X_{1,m+1} = X_{1,m} + x_{m+1} \quad (\text{C.12})$$

利用公式(方程C.11)和(方程C.12), 平均值

$$\langle x \rangle = \frac{X_{1,N_x}}{N_x} \quad (\text{C.13})$$

涨落

$$\langle (\Delta x)^2 \rangle^{\frac{1}{2}} = \left[\frac{\sigma_{1,N_x}}{x} \right]^{\frac{1}{2}} \quad (\text{C.14})$$

通过扫描一次数据就可以获得.

C.2 实现

在GROMACS中, 瞬时能量 $E(m)$ 连同 $\sigma_{1,m}$ 和 $X_{1,m}$ 一起存储在能量文件中. 虽然步数从0开始计数, 但能量和涨落的步数是从1开始计数的, 这意味着, 实现时使用的就是这里给出的方程. 我们在本节给出有些冗长的推导, 是为了以后核查代码和公式时更简单.

C.2.1 部分模拟

这种情况并不少见, 模拟的第一部分, 例如100 ps, 被用于平衡. 然而, 日志文件中给出的平均值和涨落是基于整个模拟过程进行计算的. 平衡时间, 现在作为模拟的一部分, 在这种情况下可能使得平均值和涨落无效, 因为这些数字现在由最初趋于平衡的漂移所主导.

根据方程C.7和C.8, 部分轨迹的平均值和标准偏差可利用下式计算:

$$X_{m+1,m+k} = X_{1,m+k} - X_{1,m} \quad (\text{C.15})$$

$$\sigma_{m+1,m+k} = \sigma_{1,m+k} - \sigma_{1,m} - \left[\frac{X_{1,m}}{m} - \frac{X_{1,m+k}}{m+k} \right]^2 \frac{m(m+k)}{k} \quad (\text{C.16})$$

或者, 更一般地($p \geq 1$ 并且 $q \geq p$):

$$X_{p,q} = X_{1,q} - X_{1,p-1} \quad (\text{C.17})$$

$$\sigma_{p,q} = \sigma_{1,q} - \sigma_{1,p-1} - \left[\frac{X_{1,p-1}}{p-1} - \frac{X_{1,q}}{q} \right]^2 \frac{(p-1)q}{q-p+1} \quad (\text{C.18})$$

注意, 上面算法的实现并不是非常简单的, 因为没有存储模拟中每一时间步的能量. 因此, 我们必须使用方程C.11和C.12, 从 p 时刻的信息来获得 $X_{1,p-1}$ 和 $\sigma_{1,p-1}$:

$$X_{1,p-1} = X_{1,p} - x_p \quad (\text{C.19})$$

$$\sigma_{1,p-1} = \sigma_{1,p} - \frac{[X_{1,p-1} - (p-1)x_p]^2}{(p-1)p} \quad (\text{C.20})$$

C.2.2 组合两次模拟

另一个经常发生的问题是, 必须组合两次模拟的涨落. 考虑下面的例子: 我们有两次模拟, 进行了 n 步的(A)和进行了 m 步的(B), 其中的第二次模拟是第一次的延续. 然而, 第二次模拟从1开始编号, 而不是从 $n+1$ 开始. 对部分和这没有问题, 我们必须加上A的 $x_{1,n}^A$:

$$X_{1,n+m}^{AB} = X_{1,n}^A + X_{1,m}^B \quad (\text{C.21})$$

当我们想根据两部分计算部分方差时, 必须修正 $\Delta\sigma$

$$\sigma_{1,n+m}^{AB} = \sigma_{1,n}^A + \sigma_{1,m}^B + \Delta\sigma \quad (\text{C.22})$$

如果我们定义 x_i^{AB} 为组合并重新编号后的数据点, 有:

$$\sigma_{1,n+m}^{AB} = \sum_{i=1}^{n+m} \left[x_i^{AB} - \frac{X_{1,n+m}^{AB}}{n+m} \right]^2 \quad (\text{C.23})$$

因此

$$\sum_{i=1}^{n+m} \left[x_i^{AB} - \frac{X_{1,n+m}^{AB}}{n+m} \right]^2 = \sum_{i=1}^n \left[x_i^A - \frac{X_{1,n}^A}{n} \right]^2 + \sum_{i=1}^m \left[x_i^B - \frac{X_{1,m}^B}{m} \right]^2 + \Delta\sigma \quad (\text{C.24})$$

或

$$\begin{aligned} & \sum_{i=1}^{n+m} \left[(x_i^{AB})^2 - 2x_i^{AB} \frac{X_{1,n+m}^{AB}}{n+m} + \left(\frac{X_{1,n+m}^{AB}}{n+m} \right)^2 \right] - \\ & \sum_{i=1}^n \left[(x_i^A)^2 - 2x_i^A \frac{X_{1,n}^A}{n} + \left(X_{1,n}^A \right)^2 \right] - \\ & \sum_{i=1}^m \left[(x_i^B)^2 - 2x_i^B \frac{X_{1,n}^B}{n} + \left(X_{1,n}^B \right)^2 \right] = \Delta\sigma \end{aligned} \quad (\text{C.25})$$

所有的 x_i^2 项都分离出来了, 每一项与加和指标 i 无关, 可以简化为:

$$\begin{aligned} & \frac{(X_{1,n+m}^{AB})^2}{n+m} - \frac{(X_{1,n}^A)^2}{n} - \frac{(X_{1,n}^B)^2}{n+m} - \\ & 2 \frac{X_{1,n+m}^{AB}}{n+m} \sum_{i=1}^{n+m} x_i^{AB} + 2 \frac{X_{1,n}^A}{n} \sum_{i=1}^n x_i^A + 2 \frac{X_{1,n}^B}{n} \sum_{i=1}^m x_i^B = \Delta\sigma \end{aligned} \quad (\text{C.26})$$

重新组织第二行的三个部分并使用方程C.21, 得到:

$$\Delta\sigma = \frac{(mX_{1,n}^A - nX_{1,n}^B)^2}{nm(n+m)} \quad (\text{C.27})$$

如果我们通过令 $m = 1$ 来检查上面的公式, 将回归到方程C.11.

C.2.3 能量项加和

g_energy 程序还可以将能量项加和在一起, 例如, 势能+动能=总能. 对部分平均是很容易的, 如果我们 S 个能量组分 s :

$$X_{m,n}^S = \sum_{i=m}^n \sum_{s=1}^S x_i^s = \sum_{s=1}^S \sum_{i=m}^n x_i^s = \sum_{s=1}^S X_{m,n}^s \quad (\text{C.28})$$

对于涨落更复杂些, 例如考虑一下势能和动能的涨落互相抵消的情况. 不过我们可以使用相同的方法, 如前面一样:

$$\sigma_{m,n}^S = \sum_{s=1}^S \sigma_{m,n}^s + \Delta\sigma \quad (\text{C.29})$$

我们代入方程C.6:

$$\sum_{i=m}^n \left[\left(\sum_{s=1}^S x_i^s \right) - \frac{X_{m,n}^S}{m-n+1} \right]^2 = \sum_{s=1}^S \sum_{i=m}^n \left[(x_i^s) - \frac{X_{m,n}^s}{m-n+1} \right]^2 + \Delta\sigma \quad (\text{C.30})$$

展开得到:

$$\begin{aligned} & \sum_{i=m}^n \left[\sum_{s=1}^S (x_i^s)^2 + \left(\frac{X_{m,n}^S}{m-n+1} \right)^2 - 2 \left(\frac{X_{m,n}^S}{m-n+1} \sum_{s=1}^S x_i^s + \sum_{s=1}^S \sum_{s'=s+1}^S x_i^s x_i^{s'} \right) \right] \\ & - \sum_{s=1}^S \sum_{i=m}^n \left[(x_i^S)^2 - 2 \frac{X_{m,n}^s}{m-n+1} x_i^s + \left(\frac{X_{m,n}^s}{m-n+1} \right)^2 \right] = \Delta\sigma \end{aligned} \quad (\text{C.31})$$

含 $(x_i^s)^2$ 的项互相抵消了, 这样我们就可以化简为:

$$\begin{aligned} & \frac{(X_{m,n}^S)^2}{m-n+1} - 2 \frac{X_{m,n}^S}{m-n+1} \sum_{i=m}^n \sum_{s=1}^S x_i^s - 2 \sum_{i=m}^n \sum_{s=1}^S \sum_{s'=s+1}^S x_i^s x_i^{s'} - \\ & \sum_{s=1}^S \sum_{i=m}^n \left[-2 \frac{X_{m,n}^s}{m-n+1} x_i^s + \left(\frac{X_{m,n}^s}{m-n+1} \right)^2 \right] = \Delta\sigma \end{aligned} \quad (\text{C.32})$$

或

$$-\frac{(X_{m,n}^S)^2}{m-n+1} - 2 \sum_{i=m}^n \sum_{s=1}^S \sum_{s'=s+1}^S x_i^s x_i^{s'} + \sum_{s=1}^S \frac{(X_{m,n}^s)^2}{m-n+1} = \Delta\sigma \quad (\text{C.33})$$

如果使用方程C.28展开第一项, 我们得到:

$$-\frac{\left(\sum_{s=1}^S X_{m,n}^s\right)^2}{m-n+1} - 2 \sum_{i=m}^n \sum_{s=1}^S \sum_{s'=s+1}^S x_i^s x_i^{s'} + \sum_{s=1}^S \frac{(X_{m,n}^s)^2}{m-n+1} = \Delta\sigma \quad (\text{C.34})$$

可以重写为:

$$-2 \left[\sum_{s=1}^S \sum_{s'=s+1}^S X_{m,n}^s X_{m,n}^{s'} + \sum_{i=m}^n \sum_{s=1}^S \sum_{s'=s+1}^S x_i^s x_i^{s'} \right] = \Delta\sigma \quad (\text{C.35})$$

或

$$-2 \left[\sum_{s=1}^S X_{m,n}^s \sum_{s'=s+1}^S X_{m,n}^{s'} + \sum_{s=1}^S \sum_{i=m}^n x_i^s \sum_{s'=s+1}^S x_i^{s'} \right] = \Delta\sigma \quad (\text{C.36})$$

这给出

$$-2 \sum_{s=1}^S \left[X_{m,n}^s \sum_{s'=s+1}^S \sum_{i=m}^n x_i^{s'} + \sum_{i=m}^n x_i^s \sum_{s'=s+1}^S x_i^{s'} \right] = \Delta\sigma \quad (\text{C.37})$$

由于我们需要根据所有的数据点 i 进行计算, 一般来说这是不可能的. 我们可以对 $\sigma_{m,n}^S$ 进行估计, 仅使用可用的数据点, 并使用方程C.30的左半部分. 尽管可以使用模拟中的所有时间步对平均值进行计算, 涨落的精度却由能量保存的频率所限制. 由于使用程序如xmgr很容易做到, GROMACS并没有内置此功能.