

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Неелов И.М.

**Введение в молекулярное моделирование
биополимеров**

Учебное пособие



**Санкт-Петербург
2014**

Неелов И.М. Введение в молекулярное моделирование биополимеров СПб: НИУ ИТМО, 2014. – 101 с.

Пособие адресовано для студентов, обучающихся по направлениям 240700 «Биотехнология» 223200 «Техническая физика», 241000 «Энерго- и ресурсосберегающие процессы в химической технологии, нефтехимии и биотехнологии» и содержит сведения по молекулярному моделированию, а также описание методов, команд, функций и форматов файлов, используемых пакетом молекулярного моделирования GROMACS.

Рекомендовано к печати Ученым советом инженерно-физического факультета, протокол № 10 от 14.10.2014.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2014

© Неелов И.М

Введение в молекулярное моделирование биополимеров

Аннотация

Целью пособия является введение в молекулярное моделирование для начинающих. В пособие собраны краткие теоретические сведения по молекулярному моделированию, а также описание методов, команд, функций и форматов файлов, используемых пакетом молекулярного моделирования GROMACS. Описание основано на версии пакета GROMACS 4.5, справочных материалах к этой и более поздним версиям GROMACS (www.gromacs.org) и руководства пользователя (Gromacs User Manual 4.5.4), опубликованного на английском языке. При подготовке пособия использованы также некоторые сведения по пакету GROMACS, приведенные на сайте aleks37.blogspot.com/2010/11/gromacs.html.

Введение

Молекулярное моделирование полимерных и биополимерных систем является в настоящее время мощным теоретическим инструментом исследования молекулярных систем, позволяющим вносить существенный вклад в развитие нанотехнологий и нанобиотехнологий [1-2]. Наряду с квантовомеханическими методами, которые позволяют получать результаты только для систем небольшого размера (обычно от единиц до десятков атомов), метод молекулярной динамики является основным методом моделирования таких систем наnanoуровне. Метод молекулярной динамики заключается в решении классических уравнений движения Ньютона. Впервые метод молекулярной динамики был применен в середине пятидесятых годов прошлого века Олдером [3] для изучения систем, состоящих из одинаковых двумерных дисков на плоскости. В 60-х годах метод применялся для моделирования одноатомных газов (в основном аргона) [4]. В начале 70-х годов этот метод был применен для изучения структуры воды [5]. В это же время метод МД был впервые в мире применен Балабаевым, Гривцовым и Шнолем [6,8] из Института прикладной математики (ИПМ, Москва) и Научно-исследовательского вычислительного центра в Пущино для моделирования простейшей модели линейной полимерной цепочки, состоящей из атомов, соединенных абсолютно жесткими связями («полиаргона»). Для учета жестких связей в этой работе использовались уравнения Лагранжа первого рода. В 1973-1974гг. в дипломной работе Неелова из ИВС РАН, Ленинград [7] была рассмотрена динамика линейной полимерной цепочки с упругими и абсолютно жесткими связями. В дальнейшем (начиная с 1976 года [9]) эти две группы работали совместно над развитием метода молекулярной динамики и его применений для исследования полимерных и биополимерных систем. В 1977 появилась работа Рикарта и Беллеманса [10], которая обычно считается первой работой по молекулярной динамике полимеров, поскольку более ранние российские работы были опубликованы только на русском языке и поэтому были практически

неизвестны. В 1978 в журнале ВМС были опубликованы первые совместные работы по МД полимеров двух российских групп [11], а обзорная статья по материалам совместного моделирования этих групп в 70-х годах была напечатана в 1980 [12] в специальном выпуске *Macromolecules*, посвященном юбилею нобелевского лауреата П.Флори.

Более поздние работы по МД в 80х и 90х годах были посвящены симуляции модельных полимерных цепей в растворе и в расплаве, а также в жидкокристаллическом состоянии и во внешних полях. В дальнейшем метод МД был применен для исследования все более реалистических моделей полимеров и биополимеров различной химической структуры и создания таких стандартных пакетов молекулярного моделирования как PUMA, GROMACS, CHARMM, AMBER и другие [16].

В данном пособии использованы материалы описания системы GROMACS на сайте www.gromacs.org и руководства пользователя (*Gromacs User Manual 4.5.4*), опубликованного на английском языке. При подготовке пособия использованы также сведения по пакету GROMACS, приведенные на сайте aleks37.blogspot.com/2010/11/gromacs.html.

Пособие организовано следующим образом. В *разделе 1* кратко излагаются теоретические сведения о классических уравнениях движения и их дискретизации, а также о потенциалах, используемых в молекулярном моделировании одноатомных газов. В *разделе 2* описываются потенциалы, используемые для моделирования полимерных молекул. В *разделах 3, 4 и 5* и *приложениях* описывается наиболее эффективный пакет программ GROMACS, предназначенный для молекулярного моделирования биополимерных систем. В приложениях также содержится краткая информация, необходимая для его использования в операционной системе LINUX и построения графиков в графическом пакете GRACE.

Раздел 1. Молекулярное моделирование и метод молекулярной динамики для одноатомных газов

Молекулярное моделирование это теоретический метод изучения сложных молекулярных систем на основе атомистических моделей. Оно часто используется в разработке новых материалов, когда требуется точное предсказание физических свойств реальных систем.

Все макроскопические физические свойства могут быть разделены на два типа:

- a) *статические (равновесные) свойства*, такие как средняя потенциальная энергия системы и функции радиального распределения между различными парами атомов в жидкостях и функции распределения по радиусам инерции или по расстояниям между концами для полимерных цепей.
- б) *динамические (неравновесные) свойства*, такие как вязкость жидкости, коэффициенты диффузии, динамика фазовых превращений, кинетика реакций и динамика дефектов в кристаллах.

Выбор конкретного метода зависит от требуемого результата и от способности метода дать надежные результаты на текущем уровне его развития. Релятивистское, зависящее от времени уравнение Шредингера, очень точно описывает свойства молекулярных систем, однако, рассчитать что-либо сложнее системы из нескольких атомов на этом уровне (*ab initio*) невозможно. Поэтому необходимы приближения, причем, чем сложнее система и больше интересующий промежуток времени, тем существеннее эти приближения. На некотором этапе подход *ab initio* должен быть дополнен или заменен эмпирической параметризацией используемой модели. Моделирование, “из первых принципов” не подходит для сложных систем, поэтому для них обычно используется молекулярное моделирование, основанное на механических моделях молекул и эмпирических потенциалах взаимодействия.

Макроскопические свойства всегда представляют собой средние по ансамблю в представительном статистическом ансамбле молекулярных систем. Это имеет два важных следствия:

- а) Для расчета макроскопических свойств системы недостаточно знания свойств одной структуры, даже если ее энергия является минимально возможной. Для вычисления макроскопических свойств необходимо сгенерировать представительный ансамбль при данной температуре. Но и этого недостаточно для расчета параметров термодинамического равновесия, определяемых свободной энергией (фазовые равновесия, константы связи, растворимости, относительные стабильности молекулярных конформаций и т.д.). Расчеты свободных энергий и термодинамических потенциалов требуют специальных дополнений к методам молекулярного моделирования.
- б) Детальная информация о структуре и движениях на атомном уровне, предоставляемая молекулярным моделированием, часто является излишней для определения интересующих макроскопических свойств. Это дает возможность упростить описание взаимодействий и усреднить несущественные детали. Статистическая механика предоставляет теоретическую базу для таких упрощений. Есть ряд методов, например, рассмотрение группы атомов как

одного блока, описание движения в уменьшенном количестве коллективных координат, усреднение молекул растворителя с помощью потенциалов средних сил в совокупности со стохастической динамикой.

Для генерации представительного молекулярного ансамбля есть два метода: а) *моделирование методом Монте-Карло* и б) *молекулярно-динамическое моделирование*. Для генерации неравновесных ансамблей и анализа динамики пригоден только второй метод. Хотя моделирование Монте-Карло проще, чем МД (оно не требует расчета сил), оно не дает значительно лучших результатов за тот же промежуток компьютерного времени. Поэтому МД — это более универсальный метод. Если начальная конфигурация системы очень далека от равновесной (например, если атомы в начальной конформации модельной системы сильно перекрываются), могут возникнуть слишком большие силы, которые приведут к сбою моделирования. В таких случаях необходимо начальное уравновешивание системы, которая часто производится с помощью *минимизации энергии*. Другая причина для ее проведения — это необходимость удаление кинетической энергии системы как целого для повышения точности расчетов.

1.1 Молекулярная динамика. Уравнения движения Ньютона и их дискретизация

Для молекулярного моделирования больших систем, состоящих из многих тысяч или миллионов атомов, обычно используются классические уравнения движения Ньютона

$$m_i a_i = \sum_j F_{i,j},$$

где m_i масса i -го атома, a_i его ускорение и $\sum_j F_{i,j} = F_{i,1} + F_{i,2} + \dots + F_{i,N}$ сумма сил действующих на i -й атом со стороны всех остальных атомов $j=1, \dots, N$ (j обозначает исключение из суммы i -го атома) и N полное число атомов в системе.

Как известно из школьного курса физики ускорение $a_i(t)$ в момент времени t $a_i(t) = dv_i(t)/dt = \Delta v_i/\Delta t \approx (v_i(t) - v_i(t-\Delta t))/\Delta t$ равно производной от скорости $v_i(t)$ по времени t в момент времени t или приближенно равно отношению изменения скорости Δv_i за короткий интервал времени Δt к длине этого интервала. В свою очередь скорость

$$v_i(t) = dx_i(t)/dt \approx \Delta x_i/\Delta t = (x_i(t+\Delta t) - x_i(t))/\Delta t \text{ и, соответственно}$$

$$v_i(t-\Delta t) = dx_i(t-\Delta t)/dt \approx \Delta x_i/\Delta t = (x_i(t) - x_i(t-\Delta t))/\Delta t.$$

Подставляя последние два выражения для $v_i(t)$ и $v_i(t-\Delta t)$ в формулу для $a_i(t)$ получаем, что

$$a_i(t) = ((x_i(t+\Delta t) - x_i(t))/\Delta t - (x_i(t) - x_i(t-\Delta t))/\Delta t)/\Delta t$$

$$\text{или } a_i(t) = (x_i(t+\Delta t) - 2x_i(t) + x_i(t-\Delta t))/\Delta t^2$$

После подстановки полученного значения $a_i(t)$ в исходное уравнение движения Ньютона получаем

$$m_i(x_i(t+\Delta t) - 2x_i(t) + x_i(t-\Delta t))/\Delta t^2 = \sum_j F_{i,j}$$

или

$$x_i(t+\Delta t) = 2x_i(t) - x_i(t-\Delta t)) + (\Delta t^2/m)(\sum_j F_{i,j})$$

Таким образом зная массу атома i и его координаты $x_i(t)$ и $x_i(t-\Delta t)$ в два последовательных момента времени $t-\Delta t$ и t , а также силы $\sum_j F_{i,j}$ действующие на атом i в момент времени t , можно получить значение координаты этого атома $x_i(t+\Delta t)$ в следующий момент времени $t+\Delta t$. Применяя эту формулу для каждого i -го атома много раз можно получить координаты всех атомов i в исследуемой системе $x_i(t+\Delta t), x_i(t+2\Delta t), \dots, x_i(t+k\Delta t)$ в последовательные моменты времени $t+\Delta t, t+2\Delta t, \dots, t+k\Delta t$ (с шагом Δt). Совокупность этих координат в течение времени расчета $k\Delta t$ (или, как говорят, за k шагов расчета) называется *траекторией* системы за этот интервал времени.

1.2 Стохастическая динамика

В стохастической или ланжевеновской динамике (СД или ЛД) уравнения движения записываются следующим образом:

$$m_i a_i - m_i \xi_i v_i = \sum_j F_{i,j} + F_i^{\text{random}}$$

Это означает, что в уравнении движения Ньютона добавляются два дополнительных члена: первый связанный с трением $m_i \xi_i v_i$, и второй, описывающий случайную силу F_i^{random} , действующую на все атомы в системе. Второй член необходим, так как если ввести только трение, то со временем движение всех атомов будет затухать. Коэффициент при втором члене (случайной силе) выбирается таким образом,

$$\langle F_i^{\text{random}}(t) F_j^{\text{random}}(t') \rangle = 2k_B T m_i \xi_i \delta(t') \delta_{ij}$$

чтобы он полностью компенсировал потери на трение. Здесь k_B - постоянная Больцмана, T – температура и ξ_i — коэффициент трения [пс^{-1}],

а $\delta(t')$ и δ_{ij} дельта-функция по времени и символ Кронекера, означающие, что случайная сила, действующая как в различные моменты времени, так и на разные атомы является некоррелированной (функция с такими свойствами обычно называется белым шумом).

Преимуществом стохастической динамики по сравнению с молекулярной динамикой является то, что при моделировании молекул в разбавленном растворе можно обойтись без явного учета молекул растворителя, заменив их действие трением среды и стохастическими толчками со стороны этой среды. Поскольку на долю растворителя в разбавленном растворе приходится основное число степеней свободы, это позволяет существенно ускорить расчеты в СД по сравнению с МД моделированием. Когда $1/\xi$ велико по сравнению с временными масштабами в системе, стохастическую динамику можно представить как молекулярную динамику со стохастическим термостатированием. При моделировании системы в вакууме дополнительным преимуществом является то, что не происходит накопление ошибок в поступательном и вращательном движении системы как целого. Когда $1/\xi$ мало (т.е. трение очень велико) по сравнению с временными масштабами в системе, динамика будет полностью отлична от МД, но выборка все еще будет правильной.

Этот случай обычно рассматривают отдельно и называют соответствующий ему метод броуновской динамики.

1.3 Броуновская динамика

В пределе высокого трения, когда инерционным членом $m_i a_i$ в стохастической динамике можно пренебречь, СД превращается в броуновскую динамику (БД). Уравнение движения записывается в этом случае в виде

$$-\xi_i v_i = \sum_j F_{i,j} + F_i^{\text{random}}$$

Также как в методе СД коэффициент при втором члене (случайной силе) выбирается таким образом,

$$\langle F_i^{\text{random}}(t) F_j^{\text{random}}(t+t') \rangle = 2k_B T \xi_i \delta(t') \delta_{ij}$$

чтобы он полностью компенсировал потери на трение (обозначения аналогичны обозначениям, использованным при описании метода СД).

Броуновская динамика наряду с СД широко использовалась в молекулярном моделировании поведения молекул в разбавленных растворах, поскольку, как уже отмечалось, моделирование без явного учета молекул растворителя существенно уменьшает трудоемкость расчетов.

1.4 Потенциалы взаимодействия для одноатомных молекул

Как уже отмечалось ранее, первыми системами, рассмотренными методом МД, были системы жестких двумерных дисков на плоской поверхности.

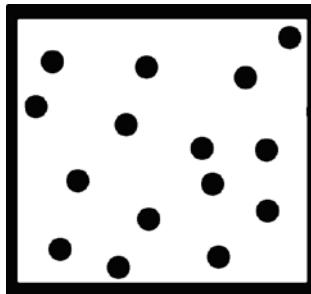


Рис.1.1 Система двумерных дисков на плоскости (или проекция на плоскость трехмерной системы из одинаковых шариков (атомов)) в ящике с непроницаемыми границами.

Следующей по сложности системой является трехмерный аналог этой системы, состоящий из большого числа одинаковых шариков (атомов). Эта система является упрощенной моделью одноатомных благородных газов (например, аргона).

В молекулярном моделировании межмолекулярные (в случае одноатомных газов межатомные) взаимодействия обычно считаются парными и аддитивными. То есть считается, что все межмолекулярные взаимодействия в системе можно описать суммой парных взаимодействий. Взаимодействие между двумя атомами i и j , находящимися на расстоянии r_{ij} может быть описано потенциалом жесткой стенки

$$U(r_{ij}) = \infty \text{ при } r \leq r_0$$

$$U(r_{ij}) = 0 \text{ при } r > r_0,$$

где r_0 – диаметр атома. Это означает, что большую часть времени атомы не взаимодействуют (на расстояниях больших диаметра r_0) и взаимодействуют только в момент столкновения (при $r = r_0$).

Более реалистическим потенциалом является непрерывный потенциал Леннарда-Джонса (ЛД), описывающий Ван-дер-ваальсовские взаимодействия между

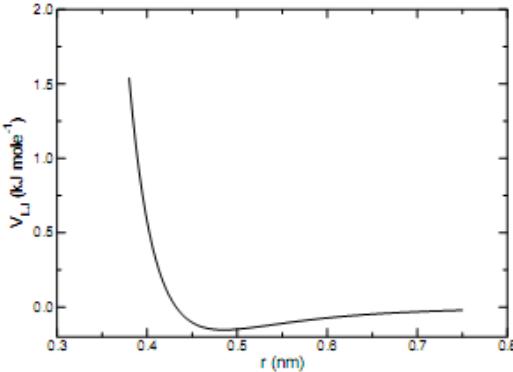


Рис. 1.2. Потенциал Леннарда-Джонса

отдельными атомами, которые являются отталкивательными на малых расстояниях ($r_{ij} \leq \sigma_{ij}$) и притягивающими при $r_{ij} > \sigma_{ij}$ с минимумом при r_{ij}^{\min} примерно равном $1.12\sigma_{ij}$ и убыванием этого притяжения при $r_{ij} > r_{ij}^{\min}$:

$$U_{LJ}(r_{ij}) = 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right) = \frac{C_{ij}^{(1)}}{r_{ij}^{12}} - \frac{C_{ij}^{(2)}}{r_{ij}^6} \text{ при } r_{ij} \leq r_{cut}$$

$$U_{LJ}(r_{ij}) = 0 \quad \text{при } r_{ij} > r_{cut},$$

Здесь ϵ_{ij} глубина потенциальной ямы, σ_{ij} расстояние, на котором потенциал обращается в ноль, а $i, j = 1, \dots, N$, где N – число атомов в системе. Для устранения небольшого скачка потенциала $U_{LJ}(r_{ij})$ в точке обрезания $r_{ij} = r_{cut}$ значение $U_{LJ}(r_{cut})$ обычно вычитается из $U_{LJ}(r_{ij})$:

$$U_{LJ}(r_{ij}) = U_{LJ}(r_{ij}) - U_{LJ}(r_{cut})$$

В отличие от потенциала жесткой стенки потенциал ЛД действует на всех расстояниях r_{ij} от 0 до бесконечности, хотя при больших расстояниях величина $U_{LJ}(r_{ij})$ очень мала. Количество взаимодействующих пар атомов пропорционально N^2 , поэтому число операций необходимых для вычисления сил для систем состоящих из нескольких сотен (а тем более тысяч или миллионов) атомов становится очень большим. Поэтому для того чтобы уменьшить объем этих вычислений потенциал $U_{LJ}(r_{ij})$ обычно обрезают на расстоянии r_{cut} порядка $2-3\sigma_{ij}$ (10-12 ангстрем), на котором этот потенциал уже близок к нулю. В том случае, если атомы заряжены, между этими атомами (точнее ионами) вводится также потенциал электростатического взаимодействия в виде закона Кулона (Coulomb) для точечных зарядов.

$$U_C(r_{ij}) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{r_{ij}}$$

Число операций для вычисления $U_c(r_{ij})$ также пропорционально N^2 , однако поскольку кулоновское взаимодействие уменьшается с расстоянием r_{ij} гораздо медленнее, чем ЛД потенциал, то для уменьшения вычислительной работы кроме радиуса обрезания обычно используются дополнительные поправки и более сложные методы, которые будут обсуждены позже.

1.5 Периодические граничные условия

Как известно газ занимает весь предоставленный ему объем, поэтому, если в начале расчета задать координаты атомов газа в некоторой конечной области, через некоторое время молекулы разлетятся на очень большие расстояния. Для того чтобы моделировать поведение газа или жидкости при заданной плотности или давлении необходимо поместить эти атомы в непроницаемый ящик (box). В реальном МД моделировании обычно рассматривается не очень большое число атомов (от нескольких сотен и тысяч на заре МД моделирования до нескольких десятков и сотен тысяч, а иногда миллионов в настоящее время). Поэтому на границе с ящиком будет находиться большая доля молекул, что нежелательно, если целью исследования является установление свойств макроскопических систем. Чтобы избежать артефактов связанных с граничными условиями, можно использовать периодические граничные условия.

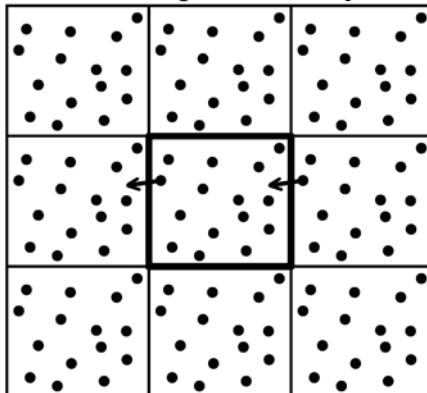


Рис. 1.3 Периодические граничные условия для системы дисков на плоскости (или для проекции 3х мерной системы шариков в ящике с проницаемыми границами)

В этом случае границы периодического ящика являются проницаемыми. В то же время, если атом выходит за одну из границ системы, он входит в нее через противоположную границу. При вычислении взаимодействий между атомами основной ячейки и атомами в ее периодических образах обычно используется *соглашение ближайшего образа*, которое требует, чтобы только один образ каждой частицы при периодических граничных условиях участвовал в парных взаимодействиях.

1.6 Статистические ансамбли и термостатирование

Как уже отмечалось для расчета макроскопических свойств системы недостаточно знания одной конформации системы, даже с минимальной энергией, а необходимо сгенерировать представительный ансамбль конформаций, содержащий достаточно большое число различных состояний

системы. Для этой цели необходимо проводить моделирование в течение как можно большего времени (и/или использовать несколько различных начальных конформаций системы и усреднять результаты как по времени, так и по начальным конформациям).

Первые расчеты методом молекулярной динамики проводились в *NVE ансамбле*, в котором постоянными являются количество частиц, объем, энергия системы. Однако для практических приложений более интересны расчеты в *NVT ансамбле*, где постоянны количество частиц, объем и температура или в *NPT ансамбле*, в котором постоянны количество частиц, давление и температура системы. Для того чтобы поддерживать постоянную температуру в *NVT ансамбле* обычно используются методы терmostатирования Беренденса (Berendsen), Нозе-Гувера (Nosé-Hoover) или пересчета скоростей. Однако, не все методы делают это корректно с точки зрения термодинамики. В больших системах погрешности средних по ансамблю и структурных свойств, кажутся незначительными. Однако систематическое исследование этих проблем почти не проводились, поэтому при интерпретации результатов следует соблюдать осторожность.

Термостат Берендсена

Алгоритм Берендсена корректирует производную температуры по времени согласно уравнению

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau_T}$$

т.е. отклонения температуры $T-T_0$ экспоненциально убывают с временной постоянной равной τ_T .

Поскольку температура определяется кинетической энергией, которая пропорциональна квадрату скорости частиц, то корректируется не сама температура, а скорость частиц. Для этого после определения скоростей частиц их значения умножаются на фактор λ :

$$\lambda = \sqrt{1 + \frac{\Delta t}{\tau_T} \left(\frac{T_0}{T} - 1 \right)}$$

Этот метод имеет то преимущество, что интенсивность теплообмена может изменяться и адаптироваться под конкретные цели: для начального уравновешивания системы величина τ_T может быть довольно малой (например, 0,01 пс), но для надежного моделирования в равновесии она должна быть достаточно большой (например, 0,5 пс). Термостат Берендсена подавляет флуктуации кинетической энергии. Это значит, что он не обеспечивается правильный канонический ансамбль, поэтому, очевидно, выборка будет неточной. Погрешность уменьшается как $1/N$, так что для очень больших систем

большинство средних по ансамблю не будут существенно затронуты, за исключением собственно распределения кинетической энергии. Однако свойства, которые определяют величину флуктуаций (например, теплоемкость) будут затронуты. Похожий термостат, который дает правильные ансамбли — это термостат с масштабированием скоростей, рассмотрен ниже.

Термостат масштабирования скоростей

Термостат масштабирования скоростей это, по сути, термостат Беренсена (см. выше) с дополнительным стохастическим членом, который гарантирует правильное распределение кинетической энергии:

$$dK = (K_0 - K) \frac{dt}{\tau_T} + 2 \sqrt{\frac{KK_0}{N_f}} \frac{dW}{\sqrt{\tau_T}}$$
 где K и K_0 — кинетическая энергия и ее заданное

значение, соответствующее температуре T_0 , N_f — число степеней свободы и dW — винеровский случайный процесс. В этом методе нет дополнительных параметров, за исключением начального числа для инициализации генератора случайных чисел. Этот термостат обеспечивает корректный канонический ансамбль, и в то же время имеет те же преимущества, что и термостат Беренсена: отклонения от заданной температуры затухают пропорционально их величине.

Термостат Нозе-Гувера

Алгоритм Беренсена является популярным эффективным методом для приведения системы к нужной температуре в начале вычислений или после изменения условий расчета (например, температуры). Однако для расчетов равновесных систем необходимо использовать более точные методы. Для этого часто используется метод, впервые предложенный Нозе и позже модифицированный Гувером. В этом случае к системе добавляется тепловой резервуар и потери на трение. Сила трения пропорциональна произведению скорости частицы на коэффициент трения ζ , а производная по времени от ζ вычисляется как разность между текущей кинетической энергией и энергией, соответствующей исходной температуре.

Баростатирование

Баростат Беренсена

Алгоритм Беренсена поддерживает постоянное давление P_0 в системе с помощью уравнения аналогичного уравнению для термостатирования по Беренсену:

$$\frac{dP}{dt} = \frac{P_0 - P}{\tau_P}$$

где P_0 — заданное давление, τ_P — временная постоянная баростата. Давление P_0 является в общем случае тензорной величиной, которая определяется выражением $\mathbf{P} = \frac{2}{V}(\mathbf{K} - \Sigma)$, где V — объем системы, \mathbf{K} — тензор кинетической энергии и Σ — вириал сил. Как и в случае с температурой на самом деле масштабируется не само давление P , а связанные с ним декартовы координаты (X, Y, Z) и соответствующие им вектора периодической ячейки. При этом может быть выбрано как изотропное (одинаковое по всем осям), так и анизотропное масштабирование. Важно отметить, что алгоритм Беренсена обеспечивает правильное среднее давление, но не дает точный NPT ансамбль, поэтому он, также как и термостат Беренсена, чаще используется на стадии начального установления равновесия в системе, чем для проведения расчетов в равновесных системах.

Баростат Парринелло-Рамана (Parrinello-Rahman)

Неточность NPT ансамбля при использовании алгоритма Беренсена может быть проблемой в тех случаях, когда целью расчетов являются флюктуации давления или объема (например, при расчете термодинамических свойств). Эта неточность может быть особенно большой для малых систем. В этом случае можно использовать баростат Парринелл-Рамана, который похож на термостат Нозе-Гувера и позволяет проводить моделирование в точном NPT ансамбле. В этом баростате векторы периодической ячейки моделирования представлены в виде матрицы \mathbf{b} , подчиняющейся уравнениям движения

$$\frac{d\mathbf{b}^2}{dt^2} = VW^{-1}\mathbf{b}^{-1}(P - P_0)$$

где V — объем ячейки, а $W_{ij}^{-1} = 4\pi\beta_{ij}/3\tau_p L$ матрица, определяющая скорость баростатирования и возможную деформацию расчетной ячейки, а матрицы \mathbf{P} и \mathbf{P}_0 — это текущее и заданное значения давления, соответственно. Здесь β_{ij} — изотермическая сжимаемость (которая в общем случае может быть анизотропной), τ_p — временная постоянная баростата и L наибольший размер ячейки моделирования. Баростат Парринелло-Рамана может быть использован совместно с термостатом Нозе-Гувера для проведения МД расчетов в точном NPT ансамбле.

1.7 Задание начальных координат молекул

Начальные координаты молекул в простейшей системе, состоящей из одноатомных молекул, могут быть легко заданы либо регулярным образом, например, на квадратной (в случае дисков на плоскости) или на кубической (в трехмерном случае) решетке, либо случайнym образом. В последнем случае можно, например, для каждого атома генерировать два (три в трехмерном случае) случайных числа, отвечающих проекциям координат каждой частицы в ячейке моделирования. Итого необходимо сгенерировать $2N$ чисел, отвечающих проекциям координат всех N дисков на плоскости или $3N$ чисел — в случае N атомов газа в трехмерном пространстве. Начальные скорости $v_i(t)$ обычно задают либо одинаковыми по величине (со средними значениями равными средней скорости при заданной температуре и случайнym направлением) или случайнym образом, но так чтобы они удовлетворяли распределению Максвелла при заданной температуре.

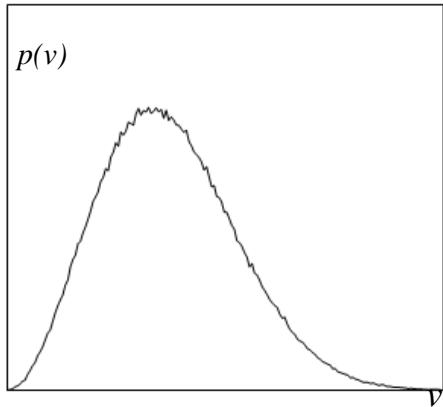


Рис. 1.4. Сгенерированное распределение Максвелла-Больцмана $p(v)$ скоростей v (Чтобы получить это распределение были сгенерированы нормально распределенные случайные величины путем сложения двенадцати случайных величин Rk в пределах $0 \leq Rk < 1$ и последующего вычитания 6,0 из суммы).

В тех алгоритмах, в которых для начала расчета нужна не скорость $v_i(t)$, а значение координат $x_i(t-\Delta t)$ и $x_i(t)$ на двух последовательных шагах по времени $t-\Delta t$ и t значение $x_i(t-\Delta t)$ может быть получено с помощью формулы
 $x_i(t-\Delta t) = x_i(t) - v_i(t)\Delta t$.

1.8 Общий алгоритм МД расчета для одноатомных газов

1. Задать начальные условия (координаты и скорости)

1.1 Задать положения $r_i(t=0)$ всех ($i=1, N$) атомов системы

1.2 Задать скорости $v_i(t=0)$ всех ($i=1, N$) атомов системы (или их координаты x на предыдущем шаге $r_i(t=0-\Delta t)$)

повторить 2, 3, 4 нужное число раз

2. Вычислить силы

2.1 Сила, приложенная к каждому атому $F_i(t)$ вычисляется путем суммирования сил между парами несвязанных атомов:

$$F_i(t) = \sum_j F_{ij}(t)$$

2.2 Вычисляются потенциальные и кинетические энергии, значения температуры T , давления, P и т.д.

3. Обновить конфигурацию, то есть получить координаты (и, если нужно, скорости) на следующем шаге

$$x_i(t+\Delta t) = 2x_i(t) - x_i(t-\Delta t)) + (\Delta t^2/m)(\sum_j F_{ij})$$

4. Вывести результат (если нужно)

Записать в файл/файлы новые координаты, скорости, энергии, T , P и т.д.

5. Построить графики функций, характеризующих статистическое и динамическое поведение системы, используя стандартные функции GROMACS и графические пакеты (например, GRACE).

1.9 Общая схема моделирования

1. Построение начальной конформации
2. Минимизация (если необходимо) потенциальной энергии системы

3. Уравновешивание системы методом МД при заданных условиях (например, T, P, плотность)
4. Проведение “продуктивного” расчета методом МД с записью траектории системы (координат, а также скоростей и сил) в файл/файлы.
5. Анализ полученной траектории (файлов) и вычисление различных статистических и динамических функций (средних, функций распределения, корреляционных функций). Запись этих функций в файлы и построение графиков функций с помощью графических пакетов, например GRACE и ORIGIN (в Windows).

Раздел 2. Молекулярное моделирование полимерных молекул

2.1 Механические модели линейных полимерных цепей

Для моделирования более сложных молекул, содержащих два и более атома, обычно используются механические модели, состоящие из шариков (атомов) и соединяющих их пружин (валентных связей). Мы рассмотрим использующиеся при этом упругие потенциалы на примере модели линейной полимерной цепи. Для разветвленных молекул используются такие же потенциалы.

Для описания валентной связи (bond) между соседними (i и $i+1$) атомами в молекуле обычно используется гармонический упругий потенциал

$$U_b(r_{i,i+1}) = \frac{1}{2} k_{i,i+1}^{(b)} (r_{i,i+1} - b_{i,i+1}^0)^2, \quad i=1, M$$

где $r_{i,i+1}$ – расстояние между атомами, $b_{i,i+1}^0$ – равновесное значение (длина валентной связи) этого расстояния, $k_{i,i+1}^{(b)}$ – коэффициент упругости связи и M – число валентных связей в молекуле.

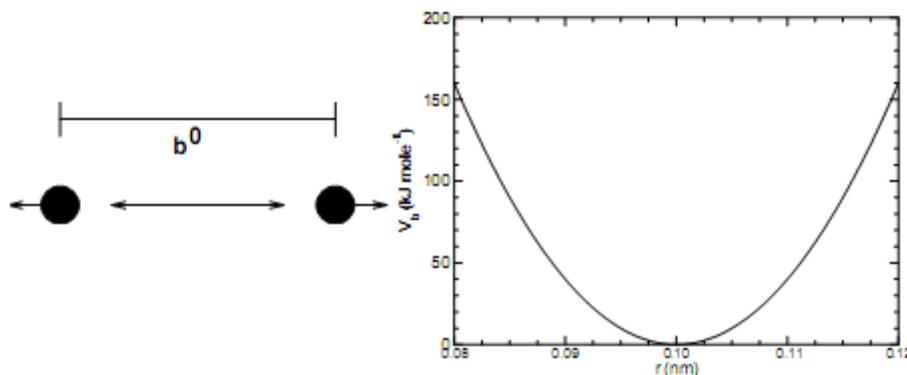


Рис. 2.1 Валентная связь и гармонический потенциал для нее.

Часто при моделировании вместо пружин используются нерастяжимые абсолютно жесткие связи (rods), уравнения для которых могут быть записано в виде

$$r_{i,i+1}^2 - b_{i,i+1}^2 = 0, \quad i=1, M$$

В первом случае мы имеем дело с шаро-пружинными (bead-spring) моделями, а во втором с шаро-стержневыми (bead-rod) моделями.

Если углы между соседними связями могут свободно изменяться, то последняя модель называется свободно-сочлененной.

В реальных молекулах состоящих из трех и более атомов валентные углы $\theta_{i,i+2}$ между соседними валентными связями обычно близки к фиксированным значениям $\theta_{i,i+2}^0$. Поэтому их описывают дополнительным гармоническим упругим потенциалом сходным с потенциалом для валентных связей:

$$U_\theta(\theta_{i,i+2}) = \frac{1}{2} k_{i,i+2}^{(\theta)} (\theta_{i,i+2} - \theta_{i,i+2}^0)^2, i=1, L$$

где L – число валентных углов в молекуле.

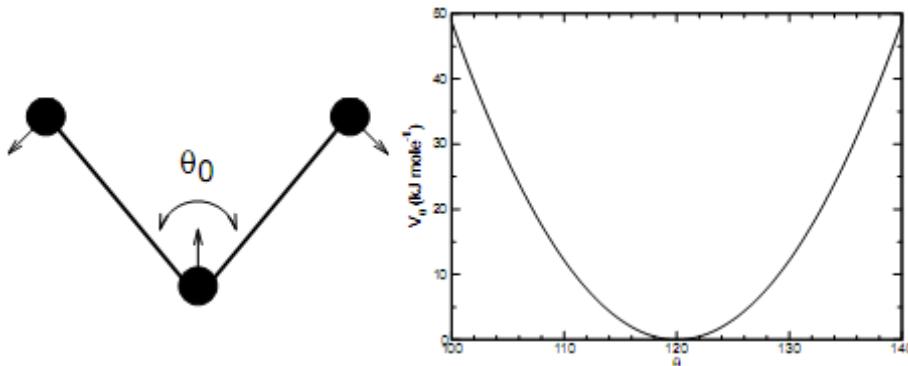


Рис. 2.2 Валентный угол и гармонический потенциал для него.

В том случае, если вращение третьей валентной связи относительно двух предыдущих является свободным, механическая модель называется моделью со свободным внутренним вращением. В противном случае модель называется моделью с заторможенным внутренним вращением и для описания заторможенности вращения вводится дополнительный потенциал:

$$U_\phi(\varphi_{i,i+1,i+2,i+3}) = \frac{1}{2} k_{i,i+1,i+2,i+3}^{(\phi)} (1 - \cos n\varphi_{i,i+1,i+2,i+3}), i=1, K$$

с константами упругости $k_{i,i+1,i+2,i+3}^{(\phi)}$, где $\varphi_{i,i+1,i+2,i+3}$ это двугранный угол между плоскостями, образованными атомами $i, i+1, i+2$ и $i+1, i+2, i+3$, и n число минимумов потенциала (обычно равное трем (в случае пространственной sp^3 гибридизации)

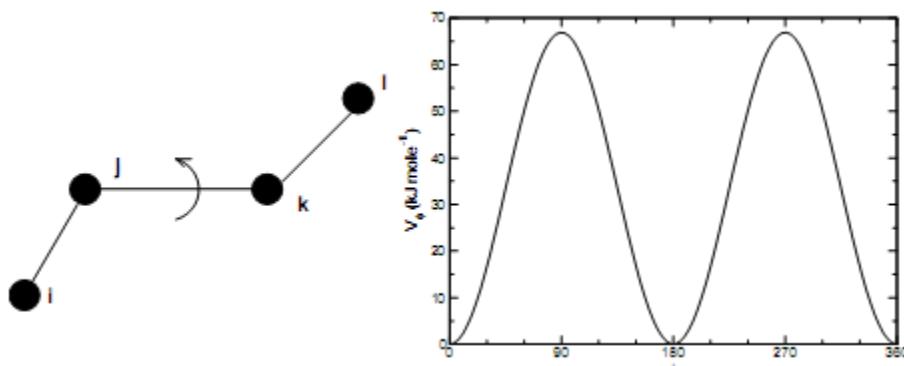


Рис. 2.3. Двугранный угол и потенциал для него.

или двум (для плоской, sp^2 гибридизации)), K - число таких углов в молекуле. Этот потенциал обычно называется в молекулярном моделировании двугранным (dihedral) или торсионным (torsion) (а в физике полимеров потенциалом внутреннего вращения или потенциалом на закручивание полимерной цепи). Для фрагментов углеводородных полимерных цепей, имеющих химическую структуру $(\text{CH}_2)_4$ (например, для боковой цепи аминокислоты лизин), используется специальный вид двугранного потенциала:

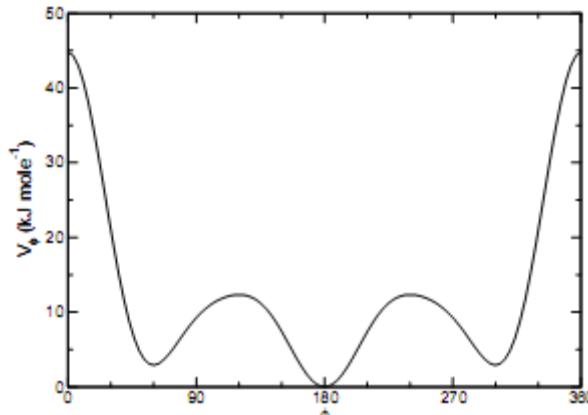


Рис. 2.4. Двугранный потенциал Райкарта-Беллеманса для алканов.

потенциал Рикаэрта-Беллемана (Ryckaert-Bellemans) в виде полинома пятой степени от $\cos \phi$

$$U_\phi(\varphi_{i,i+1,i+2,i+3}) = \sum_{n=1}^5 k_n^{(\varphi)} (\cos \varphi_{i,i+1,i+2,i+3})^n$$

Для поддержания атомов в плоскости циклических структур (например, бензольных колец) и в некоторых других случаях вводятся так называемые «неправильные» двугранные углы и соответствующие им потенциалы, которые

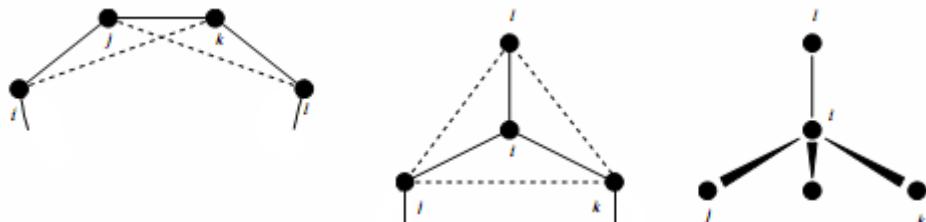


Рис. 2.5. «Неправильные» двугранные углы для бензольных колец (слева) заместителей в кольцах и инверсионные колебания. Неправильный двугранный угол ξ — это угол между плоскостями (i,j,k) и (j,k,l) во всех случаях.

предотвращают отклонение одного из 4 атомов группы от плоскости, образованной другими тремя атомами этой же группы.

2.2 Задание начальных конформаций сложных молекул

Для того чтобы начать расчет системы, подчиняющейся уравнениям движения Ньютона необходимо задать начальные координаты. Это может быть сделано различными способами. В случае небольшого числа атомов и/или простой

структуры молекулы координаты могут быть заданы вручную. В случае более сложных молекул может оказаться необходимым написать специальную компьютерную программу, задающую эти координаты, либо использовать уже готовые программы. Такие готовые программы содержатся в стандартных **молекулярных редакторах** (например, в AVOGADRO, PYMOL, VMD и других). Молекулярные редакторы позволяют не только построить начальные конформации молекул, но и визуализировать их, а также следить за изменением этих конформаций в процессе моделирования. Молекулярные редакторы также часто встроены в пакеты программ молекулярного моделирования (например, в Hyperchem, Material Studio и другие).

2.3 Минимизация энергии системы

В начальных конформациях систем, предназначенных для молекулярного моделирования, молекулы могут иметь перекрывающиеся атомы или группы атомов. Вследствие этого начальные энергии системы могут быть очень велики, что может приводить к большим ошибкам на начальной стадии работы МД алгоритмов. Чтобы избежать этого до начала МД расчетов обычно проводится минимизация энергии системы. Потенциальная энергия молекулярной системы представляет собой сложную функцию (гиперповерхность) в многомерном пространстве. Она имеет одну самую глубокую точку — глобальный минимум и много локальных минимумов, в которых все производные функции потенциальной энергии по координатам равны нулю и все вторые производные неотрицательны. Матрица вторых производных (гессиан) имеет неотрицательные собственные значения; только коллективные координаты, соответствующие перемещению и вращению изолированной молекулы, имеют нулевые собственные значения. В промежутках между локальными минимумами есть седловые точки, в которых гессиан имеет только одно отрицательное собственное значение. Эти седловые точки или точки перевала на поверхности потенциальной энергии позволяют системе переходить от одного локального минимума к другому.

Знание всех локальных минимумов и всех седловых точек системы могло бы позволить приближенно описать энергетическую поверхность этой системы как дискретный набор энергий этих минимумов, а динамику системы как динамику перехода между этими дискретными точками через соседние седловые точки. К сожалению, размерность конфигурационного пространства и количество локальных минимумов обычно настолько велики, что практически невозможно получить полную картину для большинства реальных систем. Поэтому не существует ни одного метода минимизации, который бы мог гарантированно найти глобальный минимум за разумное время.

Методы минимизации энергии можно разделить на три класса:

- 1) Методы, требующие вычисления только самой потенциальной энергии (например, метод симплексов и его вариации). Каждый шаг делается на основе предыдущих вычислений потенциальной энергии в нескольких точках. Эти

методы требуют меньшего числа операций, но сходятся медленнее, чем методы использующие информацию о производных.

2) Методы, использующие первые производные потенциальной энергии (например, метод Ньютона). Поскольку первые производные потенциальной энергии по координатам (силы) вычисляются на каждом шаге МД моделирования, этот класс методов очень удобен для использования в молекулярной динамике.

3) Методы, использующие вторые производные, которые очень хорошо сходятся в окрестности минимумов потенциальной энергии (функция квадратичного потенциала минимизируется за один шаг). Но самое главное, это то, что они требуют вычисления, хранения и обращения матриц размером $3N \times 3N$ (для системы из N частиц), что находится за пределами возможностей большинства рассматриваемых систем. Поэтому в молекулярном моделировании этот класс методов используется очень редко.

Одним из основных методов минимизации энергии используемых в молекулярном моделировании является **метод наискорейшего спуска**. В этом методе шаги делаются в направлении градиента потенциальной энергии (т.е. в направлении действия силы) без учета предыдущих шагов. Размер шага меняется в зависимости от величины градиента, что делает спуск достаточно быстрым вдали от минимумов. Однако его сходимость в окрестности локального минимума (где градиент становится небольшим) может быть довольно медленной. Более быстро сходящийся **метод сопряженных градиентов** использует информацию, полученную на предыдущих шагах. Обычно, метод наискорейшего спуска позволяет привести систему в окрестность ближайшего локального минимума, после чего включается метод сопряженных градиентов, который уточняет положение этого минимума. В молекулярном моделировании также часто используется алгоритм **L-BFGS**, который близок к методу сопряженных градиентов, но в некоторых случаях сходится быстрее.

2.4 Алгоритм МД расчетов для сложных молекул

После приготовления начальной конформации и минимизации энергии системы можно начинать моделирование. Моделирование обычно состоит из двух стадий: дополнительного уравновешивания системы методом МД (БД, СД) и продуктивного расчета для равновесной системы. Дополнительное уравновешивание необходимо не только потому, что система после минимизации может находиться в не очень хорошей конформации (например, ее атомы или группы атомов могут по-прежнему перекрываться), но и потому, что минимизация производится при нулевой температуре, в то время как моделирование обычно производится при температуре порядка 300К.

В обоих случаях общий алгоритм моделирования может быть описан следующим образом:

1. Задать начальные условия (координаты и скорости)

1.1 Задать положения r всех атомов системы

1.2 Задать скорости v всех атомов системы

повторить 2, 3, 4 нужное число раз

2. Вычислить силы

2.1 Невалентные силы, приложенная к каждому атому F_i

вычисляется путем суммирования сил между парами несвязанных атомов:

$$F_i = \sum_j F_{ij}$$

2.2 К силе F_i добавляются силы валентных взаимодействий (зависят от 1..4 соседних по цепи атомов), а также закрепляющие и внешние силы (если они есть).

2.3 Вычисляются потенциальные и кинетические энергии, Т, Р и т.д.

3. Обновить конфигурацию, то есть получить координаты

(и, если нужно, скорости) на следующем шаге

$$x_i(t+\Delta t) = 2x_i(t) - x_i(t-\Delta t)) + (\Delta t^2/m)(\sum_j F_{ij})$$

4. Вывести результат (если нужно)

Записать в файл/файлы новые координаты, скорости, энергии, Т, Р и т.д.

5. Построить графики функций, характеризующих статистическое и динамическое поведение системы, используя стандартные функции GROMACS и графические пакеты (например, GRACE).

2.5 Обработка результатов

Для обработки результатов необходимо вычислить различные функции координат (например, потенциальную энергию), скоростей (например, кинетическую энергию) и сил (например, вириал), которые описывают статистические и динамические свойства исследуемых систем в каждый момент времени. Для описания рассматриваемых систем обычно используются усредненные по времени значения этих функций и в частности:

1. Среднее значение величины $Y(t)$ по времени t $0 < t < t_{max}$

$$\langle Y \rangle = \frac{1}{t_{max}} \sum_t Y(t)$$

где t_{max} это полное время МД расчета.

В молекулярном моделировании часто вычисляются средние значения таких функций как потенциальная и кинетическая энергии, а также вклады различных потенциальных функций в общую потенциальную энергию.

2. Среднеквадратичное значение величины $Y(t)$ по времени t при $0 < t < t_{max}$

$$\langle Y \rangle = \frac{1}{t_{max}} \sqrt{\sum_t Y^2(t)}$$

В молекулярном моделировании часто вычисляются среднеквадратичные значения различных величин, например, расстояний между атомами или группами атомов, в частности, между концами линейных полимерных молекул, среднеквадратичные значения радиусов инерции молекул и их фрагментов, а также комплексов, состоящих из двух и более молекул.

3. Функции распределения. Функцией распределения (вероятностей) случайной величины $Y(t)$ называют функцию $F(Y(t))$, значение которой в точке $y(t)$ равно

вероятности события $Y \leq y$ (то есть события, состоящего только из тех элементарных исходов, для которых $Y \leq y$).

$$\langle F(Y) \rangle = P(Y \leq y)$$

Плотностью вероятности функции распределения (или дифференциальной функцией распределения) $F(Y)$ называют функцию

$$f(Y) = \frac{d}{dy} F_Y(y)$$

которая определяет вероятность попадания значения Y в интервал $y \leq Y \leq y+dy$.

Функции распределения являются более полными характеристиками, чем средние и среднеквадратичные значения величин, которые являются соответственно первым и вторым моментами функций распределения.

В молекулярном моделировании часто вычисляются функции распределения и дифференциальные функции распределения, например,

- по длинам валентных связей
- по расстояниям между концами линейных полимерных или биополимерных цепей (или по расстояниям между любыми другими парами атомов или группами атомов)
- по радиусам инерции
- радиальные функции распределения между одинаковыми (или различными) типами атомов или между группами атомов.

4. Временные корреляционные функции

$$C_Y(t) = \frac{1}{t_{\max}/2} \langle \sum_t Y(t)Y(t+t') \rangle$$

- для длин/ориентаций валентных связей
- для радиусов инерции
- для длин/векторов между концами линейных полимерных цепей (или для расстояний между любыми другими парами атомов).

Раздел 3. GROMACS. Силовые поля и ограничения движения

GROMACS это компьютерная программа для молекулярно-динамического моделирования и минимизации энергии, а также для расчета различных макроскопических свойств физических систем и, в первую очередь, полимеров и биополимеров в разбавленном растворе. Макроскопические физические свойства могут быть разделены на

- a) *статические (равновесные) свойства*, такие как средняя потенциальная энергия системы, температура, давление, плотность, функция радиального распределения для жидкости и другие, и
- б) *динамические свойства*, такие как вязкость жидкости, диффузионные процессы в жидкой фазе, динамика фазовых превращений, кинетика реакций, динамика дефектов в кристаллах.

GROMACS использует классические уравнения движения и механические модели молекул (шарики связанные пружинками или жесткими связями, см. предыдущий раздел).

3.1 Условные обозначения и единицы измерения в GROMACS

В описании GROMACS используется соглашение для обозначения различных величин:

Величина	Обозначение	Пример
Вектор	Жирный курсив	\mathbf{r}_i
Скаляр	Курсив	r_i

Таблица 3.1. Условные обозначения

Подстрочные индексы i, j, k, l используются для обозначения номеров частиц. Так \mathbf{r}_i это радиус-вектор частицы i . В этих обозначениях:

$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ - разность векторов \mathbf{r}_i и \mathbf{r}_j

$r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ - абсолютная величина этой разности, т.е. расстояние между точками с координатами \mathbf{r}_i и \mathbf{r}_j

\mathbf{F} , это сила, приложенная к i -той частице, обозначается как, а \mathbf{F}_{ij} это сила, приложенная к i -той частице со стороны j -той частицы.

Единицы измерения, используемые в МД

В пакете GROMACS используется постоянный набор единиц измерения, который называется МД единицами. Основные молекулярные величины в них имеют значения, близкие к единице. Базовыми единицами являются длина, время, масса, температура, заряд. В МД системе это нанометр (нм), пикосекунда (пс), атомная единица массы (а.е.м.), абсолютная температура (К) и заряд электрона (е) (см. табл. 3.1). Производные единицы МД системы приведены в следующей таблице.

Величина	Обозначение	Единица измерения
длина	R	$\text{нм} = 10^{-9} \text{ м}$
масса	m	$\text{а.е.м.} = 1,6605402 \cdot 10^{-27} \text{ кг}$ (1/12 массы атома ^{12}C)
время	t	$\text{пс} = 10^{-12} \text{ с}$
заряд	q	$\text{е} = \text{заряд электрона} = 1,60217733 \cdot 10^{-19} \text{ Кл}$
температура	T	К

Основные единицы измерения в GROMACS

Величина	Обозначение	Единица измерения
энергия	E, V	$\text{кДж} \cdot \text{моль}^{-1}$
сила	F	$\text{кДж} \cdot \text{моль}^{-1} \cdot \text{нм}^{-1}$
давление	p	$\text{кДж} \cdot \text{моль}^{-1} \cdot \text{нм}^{-3} = 1030/N_A \text{ Па}$ $1,66054 \cdot 10^6 \text{ Па} = 16,6054 \text{ бар}$
скорость	v	$\text{нм} \cdot \text{пс}^{-1} = 1000 \text{ м} \cdot \text{с}^{-1}$
дипольный момент	μ	$\text{е} \cdot \text{нм}$
электрический потенциал	Φ	$\text{кДж} \cdot \text{моль}^{-1} \cdot \text{е}^{-1} = .010364272 \text{ В}$
напряженность		

электрического поля	E	$\text{кДж}\cdot\text{моль}^{-1}\cdot\text{нм}^{-1}\cdot\text{e}^{-1} = 1,0364272 \cdot 10^7 \text{ В}\cdot\text{м}^{-1}$
---------------------	-----	---

Таблица 3.2. Производные единицы измерения в GROMACS

В системе МД единиц, принятой в GROMACS, физические постоянные принимают значения, отличающиеся от их значений в системе СИ (табл. 3.3). Все величины в системе МД указаны в расчете на моль, а не на молекулу. В этом случае, в частности, нет разницы между постоянной Больцмана k_B и универсальной газовой постоянной R : они обе равны $0,00831451 \text{ кДж}\cdot\text{моль}^{-1}\text{К}^{-1}$.

Обозначение	Название	Величина
N_A	число Авогадро	$6,0221367 \cdot 10^{23} \text{ моль}^{-1}$
R	универсальная газовая постоянная	$8,314510 \cdot 10^{-3} \text{ кДж}\cdot\text{моль}^{-1}\cdot\text{К}^{-1}$
k_B	постоянная Больцмана	аналогично
h	постоянная Планка	$0,39903132 \text{ кДж}\cdot\text{моль}^{-1}\cdot\text{пс}$
\hbar	постоянная Дирака	$0,063507807 \text{ кДж}\cdot\text{моль}^{-1}\cdot\text{пс}$
c	скорость света в вакууме	$299792,458 \text{ нм}\cdot\text{пс}^{-1}$

Таблица 3.3. Некоторые физические константы

Приведенные переменные

Для систем с Леннард-Джонсовским взаимодействием удобно использовать приведенные переменные (т.е. $\epsilon_{ii} = \sigma_{ii} = m_i = k_B = 1$ для одного типа атомов), указав исходные данные в безразмерных единицах. Результат так же будет выражен в этих единицах. Исключением является температура, которая будет выражена в $0,00831451$ безразмерных единиц - это является следствием использования постоянной Больцмана при вычислении температуры. В силу этого приведенной температурой будет не T , а $k_B T$. Температура $T = 1$ в GROMACS подразумевает приведенную температуру в $0,008\dots$ единиц, а если необходима приведенная температура равная единице, то нужно указывать значение $120,2717$.

величина	обозначение	связь с СИ
длина	r^*	$r \cdot \sigma^{-1}$
масса	m^*	$m \cdot M^{-1}$
время	t^*	$t \cdot \sigma^{-1} \cdot \sqrt{\epsilon/M}$
температура	T^*	$k_B \cdot T \cdot \epsilon^{-1}$
энергия	E^*	$E \cdot \epsilon^{-1}$
сила	F^*	$F \cdot \sigma \cdot \epsilon^{-1}$
давление	P^*	$P \cdot \sigma^3 \cdot \epsilon^{-1}$
скорость	v^*	$V \cdot \sqrt{\epsilon/M}$
плотность	ρ^*	$N \cdot \sigma^3 V^{-1}$

Табл. 3.4. Приведенные переменные для систем с Леннард-Джонсовским взаимодействием

3.2 Силовые поля и методы их учета в GROMACS

Силовые поля используются при вычислении сил, действующих между атомами. Они, вообще говоря, не являются частью метода моделирования, и их параметры пользователь может при необходимости изменять. Однако предметом ограничений является сама реализация силового поля, используемая конкретной программой. Функции потенциалов могут быть разделены на три типа:

- а) **Невалентные взаимодействия**: леннард-ジョンсовские и кулоновские (или модифицированные кулоновские). Невалентные взаимодействия вычисляются на основе списка соседей (списка атомов, находящихся в пространстве в некотором радиусе от данного атома и не связанных с ними химически), из которого уже убраны исключенные элементы.
- б) **Валентные взаимодействия**: взаимодействия за счет валентных связей и валентных углов, двугранных углов (*dihedrals*) и "неправильных" двугранных углов (*improper dihedrals*). Они вычисляются на основании фиксированных списков.
- в) **Ограничения**: ограничения позиций, расстояний и ориентаций, а также угловые и двугранные ограничения. Они основаны на фиксированных списках, которые должны быть созданы для каждой системы до начала расчетов.

Силовые поля, используемые в GROMACS (до последних версий) были в основном парными и аддитивными и не учитывали поляризуемости молекул. Несмотря на эти ограничения, силовое поле GROMACS является достаточно надежным и удобным и подходит для моделирования биологических макромолекул в водных растворах.

Парность и аддитивность силового поля означает, что все *невалентные* силы являются суммой невалентных парных взаимодействий, а поляризуемости атомов, учтены в среднем через *эффективные парные потенциалы*. Парные взаимодействия могут не работать в ситуациях, существенно отличных от ситуаций в тестовых системах, для которых эти модели были параметризованы. На практике эффективные парные потенциалы работают неплохо, но не всегда дают реалистические значения диэлектрической проницаемости, а также преувеличивают дальнодействующие кулоновские взаимодействия.

В GROMACS обычно используется радиус обрезания для взаимодействий Леннард-Джонса, и, в некоторых случаях, для кулоновских взаимодействий. Это означает, что, начиная с некоторого расстояния между атомами, равного радиусу обрезания, потенциал считается равным нулю. Соглашение ближайшего образа, используемое в GROMACS, требует, чтобы только один образ каждой частицы при периодических граничных условиях участвовал в парных взаимодействиях, из-за чего *радиус обрезания не должен превышать половину размера периодической ячейки*. Этого достаточно для больших систем, и трудности могут возникнуть только в системах, содержащих заряженные частицы. Но в них могут происходить весьма нежелательные явления, например, накопления зарядов на границе обрезания. Поэтому для заряженных систем рекомендуется использование специальных алгоритмов

вычисления дальнодействующих взаимодействий, таких как алгоритм Эвальда и его модификации (PME, P3M), которые будут описаны позже.

В целом валентные взаимодействия в GROMACS описываются совокупностью потенциальных энергий валентных связей (U_b), валентных углов (U_θ), двугранных (U_ϕ) углов и “неправильных” (U_{φ}) двугранных углов (см. предыдущий раздел):

$$U_{val} = U_b + U_\theta + U_\phi + U_{\varphi}.$$

Невалентные взаимодействия описываются двумя членами: Ван-дер Ваальсовского взаимодействия (обычно в виде Леннард-Джонсовского потенциала U_{LD}) и кулоновского взаимодействия (в виде обрезанного кулоновского взаимодействия для точечных зарядов (с поправками или без них) или в форме алгоритма Эвальда).

$$U_{nval} = U_{LD} + U_C$$

Таким образом, общая потенциальная энергия в системе записывается в виде суммы валентных и невалентных взаимодействий:

$$U = U_{val} + U_{nval}.$$

Исключения и 1-4 взаимодействия

Атомы в молекуле, которые ковалентно связаны с данным атомом непосредственно (1-2), а также через один (1-3) или через два атома (1-4), называются *первым, вторым и третьим соседями*, соответственно. Взаимодействия атома i с этими соседними атомами $i+1$ и $i+2$ (например, в линейной полимерной цепи) определяются преимущественно валентными взаимодействиями и не могут быть смоделированы потенциалом Леннард-Джонса. Однако они хорошо моделируются гармоническим валентным потенциалом или ограничением длины связи ($i, i+1$) и гармоническим потенциалом для валентного угла ($i, i+2$). Поэтому первый и второй соседи (атомы $i+1$ и $i+2$ или, как они иначе называются, взаимодействия 1-2 и 1-3) обычно *исключаются* из списка ЛД взаимодействий атома i и называются его *исключениями (exclusions)*.

Для третьего соседа обычное ЛД отталкивание (член пропорциональный r^{-12}) иногда является слишком сильным, и, если его применить, то молекула может деформироваться или даже разорваться. Поэтому для таких взаимодействий в силовом поле GROMOS коэффициенты при ЛД отталкивании уменьшены. Это реализовано через использование отдельных списков между атомами 1-4, 2-5, ... $i, i+3$ по цепи (которые для краткости называются 1-4 взаимодействиями) и обычных ЛД параметров. В другом силовом поле, например, OPLS, обычные параметры Леннард-Джонса уменьшаются в два раза, но при этом также масштабируются притяжение (r^{-6}) и кулоновское взаимодействие. GROMACS может использовать оба этих метода для расчета 1-4 взаимодействий.

Зарядовые группы

При использовании парных потенциалов трудоемкость вычисления сил в МД пропорциональна числу взаимодействующих пар частиц и имеет, поэтому порядок $O(N^2)$, где N - число частиц в системе. Чтобы уменьшить количество вычислений применяется *обрезание* парных потенциалов взаимодействия при

вычислениях невалентных сил. То есть, считается, что частицы взаимодействуют только до определенного расстояния между ними. Это уменьшает сложность до $O(N)$ (обычно с достаточно большим коэффициентом $100N - 200N$). Но это также вносит погрешность, которая в большинстве случаев приемлема, за исключением случаев, когда применение обрезания потенциала приводит к возникновению зарядов. В этих случаях следует рассмотреть возможность использования методов решеточных (сеточных) сумм, реализованных в GROMACS с помощью метода Эвальда (и связанных с ним методов PME и PPPM).

Рассмотрим молекулу воды, взаимодействующую с атомом. Если мы применим обрезание на уровне атом-атом, то мы можем «захватить» взаимодействие атом-кислород (с зарядом $-0,82$) без компенсирующего взаимодействия с атомами водорода. Это приведет к появлению в системе большого дипольного момента. Поэтому нужно иметь группы атомов с нулевым суммарным зарядом, называемые *зарядовыми группами*.

Настройка обрезаний потенциалов

В GROMACS можно установить разные параметры обрезания для разных потенциалов. Они указываются во входном файле `.mdp`. Есть два вида параметров, влияющих на обрезание взаимодействий. Можно выбрать тип взаимодействий, используемый в каждом случае, и то, какие обрезания должны использоваться при поиске соседей.

Для кулоновского и ван-дер-ваальсового взаимодействий есть селекторы типов взаимодействий (обозначенные `vdwtype` и `coulombtype`) и соответствующие им параметры, итого шесть параметров невалентных взаимодействий. Полное их описание приведено в разделе 7.3 руководства GROMACS 4.5. *Поиск соседей (ПС)* может быть осуществлен с использованием однодиапазонного (*single-range*) или двухдиапазонного (*twin-range*) подходов. Поскольку первый ПС является частным случаем второго ПС, мы рассмотрим более общее двухдиапазонное обрезание. В этом случае ПС характеризуется двумя радиусами: r_{list} и $\max(r_{coulomb}, r_{vdw})$. Обычно список соседей составляется каждые 10 шагов или (при шаге интегрирования 2 фс) каждые 20 фс (параметр `nstlist`). В списке соседей хранятся все пары взаимодействий, попадающие внутрь r_{list} . Далее, взаимодействия между парами, не попадающими внутрь r_{list} , но попадающими внутрь $\max(r_{coulomb}, r_{vdw})$, вычисляются во время процедуры ПС. Их силы и энергии сохраняются отдельно и добавляются к короткодействующим силам на каждом шаге между успешными ПС. Если $r_{list} = \max(r_{coulomb}, r_{vdw})$, то силы не вычисляются во время генерации списка соседей.

За исключением простого обрезания, все функции взаимодействия из табл. 3.4 требуют, чтобы ПС был произведен с большим радиусом, чем радиус, указанный в функциональной форме r_c , по причине использования зарядовых групп. Увеличенный радиус составляет порядка 0,25 нм (и соответствует наибольшему расстоянию между двумя атомами в зарядовой группе плюс

расстояние, на которое зарядовая группа может сдвинуться за счет диффузии до следующего обновления списка соседей (т.е. за 20фс)).

	Форма	Параметры
Кулоновское	Простое обрезание	$rc, \epsilon r$
	Реактивное поле	$rc, \epsilon r f$
	Сдвинутая сила	$r1, rc, \epsilon r$
	Switch функция	$r1, rc, \epsilon r$
Ван-дер- ваальсово	Простое обрезание	rc
	Сдвинутая сила	$r1, rc$
	Функция выключения	$r1, rc$

Табл. 3.4 Параметры разных функциональных форм невалентных взаимодействий.

Дальнодействующая электростатика. Суммирование по Эвальду

Электростатическая энергия U_{el} взаимодействия N частиц с их периодическими образами равна:

$$U_{el} = \frac{f}{2} \sum_{n'_x} \sum_{n'_y} \sum_{n'_z} \sum_i \sum_j \frac{q_i q_j}{r_{ij,n}}$$

$(n_x', n_y', n_z') = \mathbf{n}'$ — индексный вектор ячейки. Штрихи показывают, что члены с $i = j$ должны быть пропущены при $(n_x, n_y, n_z) = (0, 0, 0)$. Расстояние r_{ij} - это реальное расстояние между зарядами в основной ячейке и всех ее периодических образах, а не только между ближайшими. Сумма условно сходится, но очень медленно.

Суммирование Эвальда было впервые предложено как метод вычисления дальнодействующих взаимодействий между периодическими образами в кристаллах. Идея заключается в представлении одной медленно сходящейся суммы в виде суммы двух быстро сходящихся слагаемых (в обычном и фурье пространстве) и константы:

$$U_{el} = U_{dir} + U_{rec} + U_0,$$

где первый член имеет вид

$$U_{dir} = \frac{f}{2} \sum_{i,j} \sum_{n'_x} \sum_{n'_z} q_i q_j \frac{\operatorname{erfc}(\beta r_{ij,n})}{r_{ij,n}}$$

а второй член равен

$$U_{rec} = \frac{f}{2\pi U_0} \sum_{i,j} q_i q_j \sum_{n'_x} \sum_{n'_y} \sum_{n'_z} \frac{\exp(-\pi m / \beta)^2 + 2i\pi m(r_i - r_j)}{r_{ij,n}} \text{ и коэффициент}$$

$$U_0 = -\frac{f\beta}{\sqrt{\pi}} \sum_i q_i^2$$

где β — параметр, определяющий относительный вес прямой и обратной сумм, и $\mathbf{m} = (m_x, m_y, m_z)$. Таким образом получается, что можно использовать обрезание потенциала на коротком расстоянии в прямой сумме U_{dir} (порядка 1 нм) и обратной сумме U_{rec} (например, 10 волновых векторов в каждом направлении). К сожалению, вычислительные затраты на обратную часть

суммы растут как N^2 (или $N^{3/2}$ при несколько лучшем алгоритме), поэтому его не следует использовать в больших системах. Следует также иметь в виду, что поскольку суммирование по Эвальду справедливо для периодических систем (таких как кристаллы), то его использование для неупорядоченных систем (таких как жидкости) может приводить к дополнительной упорядоченности в системе.

Использование метода Эвальда в GROMACS

Суммирование методом Эвальда, обычно используют только для небольших систем, поскольку для большинства остальных случаев метод *PME* будет работать гораздо лучше. Если вы все же хотите использовать классическое суммирование Эвальда, то укажите в файле *.mdp* следующее (если ребра ячейки порядка 3 нм):

```
coulombtype = Ewald  
rvdw = 0.9  
rlist = 0.9  
rcoulomb = 0.9  
fourierspacing = 0.6  
ewald_rtol = 1e-5
```

Параметр *fourierspacing* вместе с размерами ячейки определяет наибольшую длину волновых векторов m_x , m_y , m_z для использования в каждом направлении. Параметр *ewald_rtol* — это относительная интенсивность электростатического взаимодействия на границе обрезания. Его уменьшение даст более точную прямую сумму, но менее точную обратную.

Метод *PME*

Суммирование методом частица-сетка Эвальда (*Particle Mesh Ewald*, сокращенно *PME*) было предложено в работах [11, 12] (*Tom Darden*) для улучшения скорости вычисления обратной суммы. В этом методе вместо прямого суммирования волновых векторов заряды располагаются на решетке (сетке) с помощью интерполяции В-сплайнами. Эта сетка затем подвергается быстрому трехмерному преобразованию Фурье (3D-FFT алгоритм), а обратное слагаемое энергии получается единственным суммированием по сетке в обратном пространстве.

Потенциал в узлах сетки вычисляется с помощью обратного преобразования, а затем, используя коэффициенты интерполяции, вычисляются силы, действующие на каждый атом.

Алгоритм *PME* имеет сложность $N \log N$ и на средних и больших системах работает существенно быстрее, чем обычное суммирование по Эвальду. В то же время на малых системах может быть предпочтительнее использовать обычное суммирование по Эвальду, чтобы избежать затрат на подготовку к работе сетки и преобразования Фурье.

Использование метода *PME*

Для того, чтобы использовать *PME*, нужно задать в файле *.mdp* следующие параметры:

```
coulombtype = PME
```

$rwdw = 0.9$
 $rlist = 0.9$
 $rcoulomb = 0.9$
 $fourierspacing = 0.12$
 $pme_order = 4$
 $ewald_rtol = 1e-5$

Параметр *fourierspacing* определяет здесь максимальный размер клетки FFT сетки, а *pme_order* задает порядок интерполяции. Использование интерполяции четвертого порядка (кубической) и заданного размера клетки должно обеспечить точность электростатических энергий до $5 \cdot 10^{-3}$. Т.к. значения энергий ЛД взаимодействий не настолько точны, в данном примере можно даже немного увеличить размер клетки.

Баростатирование может работать с методом *PME*, но надо иметь в виду, что анизотропное масштабирование может приводить к искусственному упорядочиванию в некоторых системах.

Метод *PPPM*

Методы частица-частица частица-сетка (в оригинале *Particle-Particle Particle-Mesh*, сокращенно *PPPM* или *P3M*) Hockney и Eastwood также могут использоваться в GROMACS для вычисления дальнодействующих электростатических взаимодействий. В этом алгоритме заряды всех частиц распределены по сетке размерами (n_x , n_y , n_z) с использованием треугольной весовой функции:

$$W(r) = W(x)W(y)W(z)$$

$$W(\xi) = \begin{cases} \frac{3}{4} - (\xi/h)^2 & \text{при } |\xi| \leq h/2 \\ \frac{1}{2}(3/2 - |\xi|/h)^2 & \text{при } h/2 < |\xi| \leq 3h/2 \\ 0 & \text{при } 3h/2 \leq |\xi| \end{cases}$$

где ξ - расстояние до узла сети в направлении x , y или z . Для каждого заряда нужно учитывать только 27 ближайших узлов сети.

Это распределение зарядов подвергается быстрому преобразованию Фурье по 3D *FFT* алгоритму. В Фурье-пространстве производится свертка с функцией $\hat{G}(k)$:

$$\hat{G}(k) = -\frac{\hat{g}(k)}{\epsilon_0 k^2}$$

где \hat{g} преобразование Фурье от функции распределения зарядов. Она дает дальнодействующий потенциал, который может быть преобразован с помощью прямого *FFT* преобразования в реальный пространственный потенциал. В конце потенциалы и силы получаются с помощью интерполяции. Вычислить полный дальнодействующий тензор вириала с *PPPM* трудно, но возможно получить его след. Поэтому сумма диагональных компонентов тензора давления (и, соответственно, изотропное давление) правильны, но отдельные его компоненты могут быть неверны!)

Использование метода *PPPM*

Для использования *PPPM* укажите в файле *.mdp* следующее:

```
coulombtype = PPPM  
rlist = 1.0  
rcoulomb = 0.85  
rcoulomb_switch = 0.0  
rvdw = 1.0  
fourierspacing = 0.075
```

Для подробностей о switch параметрах см. раздел о модифицированных дальнодействующих взаимодействиях. При использовании *PPPM* мы рекомендуем брать максимум 0,075 нм на узел сетки (например, 20 узлов в 1,5 нм). *PPPM* не обеспечивает такую же точность, как и PME, но может быть чуть быстрее в некоторых случаях. Из-за проблем с вириалом его нельзя использовать вместе с баростатом. *PPPM* отключено в настоящее время в *GROMACS*, но планируется его вернуть.

Конкретные силовые поля используемые в GROMACS

Как уже отмечалось силовые поля не являются частью пакета GROMACS, но GROMACS может использовать как свои силовые поля, так и поля из других пакетов молекулярного моделирования.

Силовые поля состоят из двух компонентов:

- а) Набор уравнений (называемых *потенциальными функциями*), используемых для получения потенциальных энергий и сил. (Они описаны в разделе 2).
- б) Параметры, используемые в этом наборе уравнений. Они не описаны ни здесь, ни в руководстве пользователя GROMACS, но есть в соответствующих файлах в дистрибутиве GROMACS.

Для одного набора уравнений можно использовать разные наборы параметров, но следует при этом учитывать, что комбинация уравнений и параметров образует согласованный комплект. В общем случае опасно самостоятельно изменять параметры, т.к. каждое такое изменение перед использованием должно быть проверено сравнением с экспериментальными данными, опубликовано в рецензируемом издании и документировано.

Версия GROMACS 4.5.4 содержит несколько силовых полей, также дополнительные силовые поля доступны на сайте. Если вы не знаете, какое из них выбрать, мы рекомендуем GROMOS-96 с атомами водорода, включенными в тяжелые атомы, и полноатомные OPLS-AA/L. Опишем эти поля подробнее.

Силовое поле GROMOS-87

Комплект программ и соответствующее силовое поле GROMOS-87 составили фундамент для разработки GROMACS в начале 1990-х. Исходное силовое поле GROMOS-87 недоступно в GROMACS. В предыдущих версиях (до 3.3.2) использовалось т.н. «силовое поле GROMACS», основанное на GROMOS-87 с небольшими изменениями, касающимися взаимодействий атомов кислорода воды с атомами углерода, и новыми 10 атомными типами. Как отмечает pdb2gmx, это силовое поле не рекомендуется к использованию в настоящее время, т.к. имеются более новые и, возможно, более достоверные версии этого

набора параметров. В целях обратной совместимости оно поддерживается в текущей версии. Все необходимые файлы находятся в папке ***gmx.ff*** библиотеки GROMACS.

Полноатомное силовое поле основанное на GROMOS-87

Использованное в GROMACS силовое поле, основанное на GROMOS-87, и включающее все атомы водорода, является практически идентичным полю GROMOS-87, т.к. дополнительные атомы водорода не участвуют в ЛД взаимодействии и имеют нулевой заряд. Отличия имеются только в угловом и несобственном двугранном потенциалах. Это силовое поле полезно, только когда нужны точные позиции атомов водорода, например, при ограничениях расстояний, основанных на ЯМР измерениях. При использовании прочтите, пожалуйста, предыдущий пункт.

Силовое поле GROMOS-96

GROMACS поддерживает силовые поля GROMOS-96. Все параметры и все стандартные блоки силовых полей 43a1, 43a2, 45a3, 53a5 и 53a6 включены в GROMACS и топологии могут быть автоматически получены с помощью команды ***pdb2gmx***. Силовое поле GROMOS-96 — это дальнейшее развитие силового поля GROMOS-87, в частности, это поле имеет улучшения для белков и малых молекул. Однако разработчики GROMACS не рекомендуют использовать его с длинными алканами и жирами. Силовое поле GROMOS-96 отличается от поля GROMOS-87 несколькими моментами:

- а) параметрами силового поля
- б) параметрами валентных взаимодействий, которые здесь не привязаны к атомным типам
- в) потенциалом валентных связей четвертой степени (раздел 4.2.1 руководства GROMACS)
- г) угловой потенциал в виде \cos валентного угла (раздел 4.2.5 руководства GROMACS)

Реализация GROMOS96 в GROMACS отличается от исходной реализации силового поля GROMOS96 тем, что:

- а) поиск соседей для растворителя осуществляется не по первому атому в молекуле растворителя, а по центру зарядовой группы. Впрочем, разница очень мала.
- б) вириал в GROMOS-96 основан на молекулах, а GROMACS использует атомные вириалы.

Силовое поле GROMOS-96 было параметризовано с леннард-ジョンсовским радиусом обрезания 1,4 нм, поэтому не используйте меньшие значения (параметр *rvdw*). Большой радиус обрезания допустим, т.к. ЛД потенциал и силы далее 1,4 нм практически нулевые.

Файлы силового поля GROMOS-96

GROMACS может читать и записывать файлы координат и траекторий в формате GROMOS-96. Они должны иметь расширение ***.g96***. Такой файл может быть входным/выходным файлом GROMOS-96, координатным файлом траектории или их комбинацией. Формат файлов фиксирован: все дробные

числа записаны не в степенном, а в фиксированном формате 15,9, поэтому файлы могут быть огромными. GROMACS поддерживает следующие блоки данных в указанном порядке:

TITLE (обязателен, заголовок)

TIMESTEP (опционально, шаг интегрирования)

POSITION/POSITIONRED (обязательно, координаты x,y,z)

VELOCITY/VELOCITYRED (опционально, скорости v_x,v_y,v_z)

BOX (опционально, размеры периодического ящика)

См. руководство к GROMOS-96 [71] для полного описания блоков. **Обратите внимание**, что все программы GROMACS могут читать сжатые (**.Z** и **.gz**) файлы.

Силовое поле OPLS/AA

(*Описание этого поля не содержится в руководстве по GROMACS*)

Силовое поле CHARMM

В версии 4.5 GROMACS поддерживает силовое поле CHARMM27 для белков, липидов и нуклеиновых кислот. Параметры белков (и некоторые расширения для липидов и нуклеиновых кислот) были тщательно протестированы с помощью сравнение потенциальных энергий конвертированного в GROMACS и стандартного наборов параметров моделирования CHARMM и б) анализ поведения силового поля белков при использовании особых возможностей GROMACS, например, виртуальных позиций (в т.ч. и с длинными шагами по времени) и недавно реализованного неявного растворителя. Подробности и результаты представлены в работе Bjelkmar и др. Параметры нуклеиновых кислот и гема были конвертированы и протестированы Michel Cuendet. При использовании силового поля CHARMM в **pdb2gmx** по умолчанию используются **CMAP** (двугранные перекрестные члены для главных цепей белков), чтобы избежать этого, укажите флаг **-postar**.

Силовые поля AMBER

GROMACS поддерживает несколько вариантов силового поля AMBER, включая AMBER94, AMBER96, AMBER99, AMBER99SB, AMBER99SB-ildn, AMBER03, AMBERGS, которые широко используется в одноименном (AMBER) пакете молекулярного моделирования.

Силовое поле MARTINI

Силовое поле MARTINI — это набор параметров для моделирования с помощью , обобщенных крупнозернистых моделей, который позволяет моделировать разные системы, в том числе белки и мембранны.

3.3 Ограничения движения

В GROMACS используются специальные потенциалы для наложения ограничений на движение частиц в системе, чтобы избежать больших смещений, которые могут привести к сбою программы, или чтобы поддерживать расстояния известные из экспериментальных данных по ЯМР. Ограничения не являются частью силового поля.

Ограничения положений (координат) используются для удержания частиц около определенных положений в пространстве – например около точки с координатами R_i . Они могут использоваться, например, во время уравновешивания, чтобы избежать больших перестроек важных частей молекулы (например, для ограничения движения белка, который подвергается большим силам со стороны не уравновешенного растворителя).

Другим возможными ограничениями в GROMACS являются **ограничения углов**.

Они используются для ограничения угла между двумя парами частиц (i, j и k, l) или между парой частиц (i и j) и осью z .

Ограничения двугранных углов используются для ограничения двугранного угла θ , заданного четырьмя частицами.

Ограничения расстояний используются для наложения ограничений на движения в системе, для которой из эксперимента (например, из ЯМР) известны некоторые расстояния. МД может использоваться в этом случае для улучшения и уточнения структур с использованием данных ЯМР. В GROMACS есть три типа ограничений на пары атомов:

- а) Простые гармонические: связь типа 6 из номенклатуры связей GROMACS.
- б) Кусочно-линейные гармонические: связь типа 10 из номенклатуры связей GROMACS..
- в) Комплексные ЯМР ограничения расстояний, с возможностью усреднения по паре частиц, времени и/или ансамблю.

Раздел 4 Топологии в GROMACS

4.1 Введение

Для того, чтобы проводить МД расчеты, необходимо знать между какими атомами и/или группами атомов должны действовать потенциальные функции, описанные в предыдущих разделах. Также необходимо знать параметры этих функций. Все эти данные в GROMACS записаны в файле *топологии* ***.top**, который содержит *постоянные атрибуты* каждого атома. Атомных типов намного больше, чем элементов, т.к некоторые свойства атомов зависят от их окружения в молекуле, но в силовом поле параметризованы только те элементы, которые присутствуют в биологических системах, плюс некоторые металлы, ионы и кремний. Валентные и специальные взаимодействия определяются фиксированными списками, которые включаются в файлы топологий. Некоторые невалентные взаимодействия должны быть исключены (первые и вторые соседи), т.к. они уже учтены в валентных взаимодействиях. У атомов, кроме постоянных атрибутов, есть еще и динамические — их позиции, скорости и силы. Они не входят в молекулярную топологию и хранятся в файле координат ***.gro** (позиции и скорости) или файле траектории ***.trr** (позиции, скорости и силы).

В этой главе описаны структуры файлов топологии (***.top**) и файлов баз данных (в них хранятся параметры и то, каким образом их следует при необходимости

изменять). Кроме того в этом разделе описываются структуры файлов разных силовых полей.

4.2 Типы частиц

В GROMACS есть три типа частиц, см. табл. 4.1. В основном используются обычные атомы. Иногда (например, в моделях воды TIP4P и TIP5P) используются также виртуальные позиции взаимодействий, а оболочки нужны для поляризуемых моделей, которые мы в данном пособии не рассматриваем.

Частицы	Символ
атомы	A
оболочки	S
виртуальные позиции	$V(D)$

Табл. 4.1. Типы частиц в GROMACS

Типы атомов

Каждое силовое поле определяет набор типов атомов, которые имеют характерные имена или номера и массы (в а.е.м). Эти списки находятся в файле *atomtypes.atp* (*.atp* = *atom type parameters*). В него можно добавлять новые типы или модифицировать имеющиеся. Пример из силового поля *gmx.ff*:

O 15.99940 ; carbonyl oxygen($C=O$)
OM 15.99940 ; carboxyl oxygen(CO^-)
OA 15.99940 ; hydroxyl oxygen(OH)
OW 15.99940 ; water oxygen
N 14.00670 ; peptide nitrogen(N or NH)
NT 14.00670 ; terminal nitrogen(NH_2)
NL 14.00670 ; terminal nitrogen(NH_3)
NR5 14.00670 ; aromatic N (5-ring, 2 bonds)
*NR5** 14.00670 ; aromatic N (5-ring, 3 bonds)
NP 14.00670 ; porphyrin nitrogen
C 12.01100 ; bare carbon (peptide, $C=O$, $C-N$)
CH1 13.01900 ; aliphatic CH-group
CH2 14.02700 ; aliphatic CH₂-group
CH3 15.03500 ; aliphatic CH₃-group

Обратите внимание: GROMACS пользуется атомными типами как именами, а не как номерами (как, например, GROMOS).

Виртуальные позиции

Некоторые силовые поля используют виртуальные позиции (ВП) взаимодействий, сконструированные из позиций других частиц, которые участвуют в некоторых взаимодействиях. Например, использование ВП в бензольных кольцах позволяет воспроизвести правильный квадрупольный момент. Чтобы добавить виртуальные позиции взаимодействий в систему, нужно добавить секцию [virtual_sites?]

в файл топологии, где «?» — это число исходных частиц для виртуальной позиции («2» для типа 2, «3» для типов 3, 3fd, 3fad и 3out, «4» для 4fdn и «N» для N). Последний тип заменяет собой устаревший тип 4fd (тип 1), который иногда бывает нестабильным. 4fd все еще внутренне поддерживается, но использовать его не следует. (Для совместимости с предыдущими версиями GROMACS разрешает использование старого названия [*dummies?*] вместо [*virtual_sites?*] тоже допускается).

Задание виртуальных позиций описывается следующим образом

для типа 2:

```
[ virtual_sites2 ]
; Site from funct a
5 1 2      1  0.7439756
```

для типа 3:

```
[ virtual_sites3 ]
; Site from funct a      b
5 1 2 3     1  0.7439756 0.128012
```

для типа 3fd:

```
[ virtual_sites3 ]
; Site from funct a      d
5 1 2 3     2  0.5       -0.105
```

для типа 3fad:

```
[ virtual_sites3 ]
; Site from funct theta   d
5 1 2 3     3  120       0.5
```

для типа 3out:

```
[ virtual_sites3 ]
; Site from funct a      b      c
5 1 2 3     4  -0.4      -0.4   6.9281
```

для типа 4fdn:

```
[ virtual_sites4 ]
; Site from funct a      b      c
5 1 2 3 4   2  1.0       0.9   0.105
```

Таким образом будут созданы виртуальные позиции номер 5 (номер 5 записан в колонка Site), основанные на позициях исходных атомов, указанных в колонке from, способом (функцией), заданным в колонке funct, с числовыми параметрами в последних колонках. Номера атомов и виртуальных позиций зависят от конкретной молекулы. (Советуем изучить топологии молекул воды TIP4P или TIP5P из поставки GROMACS).

4.3 Файлы параметров

Атомы

Статические свойства (см. табл. 4.2), приписанные данному типу атома, определяются на основе данных из различных источников. Имя и масса атомного типа записаны в файле *atomtypes.atp*, а заряд — в файле **.rtp* (где .rtp означает residue topology parameter file). Это значит, что заряды задаются только в стандартных блоках аминокислот в файле **rtp*

Свойство	Символ	Размерность
тип	—	—
масса	<i>m</i>	а.е.м
заряд	<i>q</i>	е
эпсилон	<i>ε</i>	кДж/моль
сигма	ζ	нм

Табл. 4.2. Статические свойства атомных типов в GROMACS

Параметры невалентных взаимодействий

К невалентным параметрам принадлежат ван-дер-ваальсовы параметры *V* (*C6* или ζ , в зависимости от правила комбинирования) и *W* (*C12* или ϵ), которые перечислены в файле *ffnonbonded.itp*, где *ptype* — это тип частицы (см. табл. 5.1). Как и с валентными параметрами, записи в секциях [*types] применяются к их копиям в файле топологии. Пропущенные параметры вызывают предупреждения, за исключением указанных в разделе 5.3.4.

[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
```

.....

Обратите внимание: большинство поддерживаемых силовых полей также включают столбец *at.num.*, но та же информация получается и из колонки *bond_type* OPLS-AA. Интерпретация параметров *V* и *W* зависит от правила комбинирования, указанного в секции [defaults] в файле топологии.

Некоторые комбинации разных атомных типов могут быть заданы в секции [*nonbond_params*] вместе с параметрами *V* и *W*, как показано выше. Любая комбинация, которая не задана, будет вычислена из параметров соответствующих атомных типов в соответствии с правилом комбинирования.

Если нужно указать σ и ϵ (способы 2 и 3 в руководстве Gromacs), может показаться невозможным иметь ненулевой параметр *C12* вместе с нулевым *C6*, однако установка отрицательного σ поможет это сделать

Параметры валентных взаимодействий

Параметры валентных взаимодействий между разными атомными типами (например, связей, валентных углов, двугранников) перечислены в файле ***ffbonded.itp***. Записи в этой базе данных содержат, соответственно, атомные типы, участвующие во взаимодействии, а также тип и параметры этого взаимодействия. Эта информация затем считывается grompp при обработке топологии и соответствующим образом применяется. Например, параметры из [*bondtypes*] применяются к связями, перечисленным в секции [*bonds*] и т.д. При отсутствии любого валентного параметра в соответствующей секции [**types*] выдается критическая ошибка.

Пример отрывка из такого файла:

```
[ 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 1 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
```

В файл ***ffbonded.itp*** можно добавлять новые параметры. Если вы хотите добавить валентные параметры для новых атомных типов, то удостоверьтесь, что вы определили эти типы в файл ***atomtypes.atp***.

Внутримолекулярные парные взаимодействия

Дополнительные леннард-джонсовы и электростатические взаимодействия между парами атомов в молекуле могут быть добавлены в секцию [*pairs*] определения молекулы. Параметры этих взаимодействий могут быть установлены независимо от параметров невалентных взаимодействий. В

силовых полях GROMOS пары используются только для модификации 1-4 взаимодействий (взаимодействий атомов, разделенных тремя связями). В этих силовых полях 1-4 взаимодействия исключаются из невалентных взаимодействий (см. раздел 5.4).

[pairtypes]

```
;i j func cs6      cs12 ; THESE ARE 1-4 INTERACTIONS
O O 1 0.22617E-02 0.74158E-06
O OM 1 0.22617E-02 0.74158E-06
....
```

Параметры валентных взаимодействий атомных типов в файле *ffnonbonded.itp* перечислены в секции [pairtypes]. В силовом поле GROMOS все эти параметры перечислены явно, но эта секция может быть и пуста для силовых полей вроде OPLS, которые вычисляют 1-4 взаимодействия одинаковым масштабированием параметров. Парные параметры, не присутствующие в секции [pairtypes], вычисляются, только если параметр *gen-pairs* в секции [defaults] файла *forcefield.itp* установлена в «yes». Если же *gen-pairs* равен «no», то команда *grompp* выдаст предупреждение для каждого типа пар, для которых не указаны параметры.

Обычные парные взаимодействия, предназначенные для 1-4 взаимодействий, имеют функциональный тип 1. Функциональные типы 2 и 3 нужны для моделирований свободных энергий. При определении свободных энергий гидратации растворенное вещество нужно разъединить с растворителем. Это может быть сделано добавлением топологии состояния Б, в которой все невалентные параметры растворенного вещества, например, заряды и ЛД параметры, установлены в нуль. Однако разница свободных энергий между этими состояниями не будет равна полной свободной энергии гидратации, т.к. нужно еще добавить свободную энергию ЛД и кулоновских взаимодействий в самом растворенном веществе, которые были обнулены. Все это можно провести в одну стадию, если оставить нетронутыми кулоновское и ЛД взаимодействия в молекулах (ионах) растворенного вещества. Для этого есть функциональный тип 2, идентичный типу 1, за исключением того, что параметры состояния Б всегда идентичны параметрам состояния А. При вычислении параметров секции [pairtypes] не делается различия между функциями типов 1 и 2. Функциональный тип 3 предназначен для замены невалентных взаимодействий. Он использует немасштабированные заряды и ЛД параметры. Также он использует только параметры состояния А.

Обратите внимание, что нужно добавлять исключения для всех пар атомов, участвующих в парных взаимодействиях типа 3, иначе такие пары также попадут в обычные списки соседей.

Все три типа парных взаимодействий всегда используют простые кулоновские взаимодействия, даже когда для невалентных взаимодействий выбраны реактивное поле, PME, суммирование Эвальда или сдвинутые кулоновские взаимодействия. Энергии функций типов 1 и 2 записываются в файлы энергии и журнала в отдельных «LJ-14» и «Coulomb-14» записях для каждой пары групп

энергии. Энергии функций типа 3 добавляются к членам «LJ-(SR)» и «Coulomb-(SR)».

Параметры неявного растворителя

Неявный растворитель поддерживается, начиная с GROMACS 4.5. Для его параметров была введена секция в топологии:

```
[ 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

В этом примере вначале перечислены атомные типы, затем пять чисел и комментарий (после точки с запятой).

Значения в колонках 1–3 сейчас не используются. Они были нужны для более точных алгоритмов вычисления площади поверхности, в частности Still и др. [63]. Колонка 4 содержит ван-дер-ваальсовы атомные радиусы, которые используются при вычислении борновских радиусов. Диэлектрический сдвиг указывается в файле *.mdp* и добавляется к исходному радиусу различными методами. Колонка 5 — это фактор масштабирования для моделей НСТ и ОВС. Эти значения берутся из оригинальной работы по НСТ [64] и из оригинальной работы по Tinker.

Исключения

Исключения для невалентных взаимодействий генерируются командой *grompp* для соседних атомов, разделенных числом связей, не большим числа, которое задано в секции [*moleculetype*] файла топологии (два атома не должны участвовать одновременно в валентных и невалентных взаимодействиях друг с другом — *Прим. перев.*). Частицы рассматриваются как связанные, если они связаны химическими связями ([*bonds*] типов 1..5, 7 и 8) или ограничениями ([*constraints*] типа 1). [*bonds*] типа 5 могут использоваться для соединения двух атомов без приложения сил взаимодействия. Имеется гармоническое взаимодействие ([*bonds*] типа 6), которое не соединяет атомы химической связью. Также имеется второй тип ограничений ([*constraints*] типа 2), который фиксирует расстояние, но не соединяет атомы химической связью. Полный список этих взаимодействий см. в табл. 5.5.

Дополнительные исключения в молекуле могут быть добавлены вручную в секции [*exclusions*]. Каждая строка должна начинаться с одного индекса атома, за которым идут один или несколько других индексов атомов. Все невалентные взаимодействия между первым атомом и всеми остальными будут исключены. Когда все невалентные взаимодействия внутри или между группами атомов должны быть исключены, более удобно и намного эффективнее использовать исключения групп мониторинга энергии (см. раздел 3.3).

Ограничения

Ограничения задаются в секции [*constraints*] в следующем формате: два номера атомов, затем тип функции (1 и 2) и длина ограничения. Единственная

разница между типами 1 и 2 заключается в том, что ограничения типа 1 учитываются при генерировании исключений, а типа 2 — нет (см. выше). Расстояния ограничиваются с помощью алгоритмов LINCS или SHAKE, которые могут быть выбраны в файле **.mdp**. Оба типа ограничений могут изменяться в вычислениях свободных энергий, если задать второе расстояние. Некоторые типы связей и углов могут быть автоматически преобразованы командой ***grompp*** в ограничения, для этого есть несколько опций в файле **.mdf**. В GROMACS также реализован алгоритм SETTLE [42], который является аналитическим решением SHAKE конкретно для воды. SETTLE может быть выбран в файле топологии. Рассмотрим определение SPC молекулы воды:

```
[ moleculetype ]
; 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
```

Секция [settles] определяет первый атом молекулы воды. funct всегда равна 1, длины связей О-Н и Н-Н следует задать. **Обратите внимание**, что этот алгоритм также должен использоваться для TIP3P и TIP4P [96]. TIP3P имеет другую геометрию. TIP4P содержит виртуальные позиции, но т.к. их позиции вычисляются, связи с ними не нужно ограничивать.

4.4 Входные файлы для команды **pdb2gmx**

Команда GROMACS ***pdb2gmx*** генерирует топологию из входного файла координат. Для файла координат поддерживаются несколько форматов, но чаще всего используется файлы типа ***.pdb** (отсюда и название программы). Команда **pdb2gmx** ищет силовые поля в папке GROMACS **share/top** и вашей рабочей папке. Силовые поля определяются из файла **forcefield.itp** в папке с расширением **.ff**. В ней может существовать файл **forcefield.doc**, и если так, то его первую строку **pdb2gmx** покажет пользователю как справку в выборе силового поля. В противном случае пользователь может указать нужное силовое поле через ключ **-ff xxx**. ***pdb2gmx*** будет искать сначала в текущей папке, затем в папке GROMACS **share/top** и, наконец, в первой попавшейся папке **xxx.ff**.

pdb2gmx читает два общих файла: файл атомных типов (**.atp**) из папки с силовым полем и файл **residuetypes.dat** из рабочей папки или папки GROMACS share/top. **residuetypes.dat** определяет, какие имена остатков соответствуют белкам, ДНК, РНК, воде и ионам.

pdb2gmx может читать одну или несколько баз данных с топологической информацией для разных типов молекул. Файлы, принадлежащие к одной базе данных, должны иметь одинаковые базовые имена, желательно указывающие на тип молекул (например, *aminoacids*, *rna*, *dna*). Допустимые файлы:

- **<basename>.rtp** (*обязательный*)
- **<basename>.r2b**
- **<basename>.arn**
- **<basename>.hdb**
- **<basename>.n.tdb**
- **<basename>.c.tdb**

Обязателен только файл **.rtp**, который содержит топологии стандартных блоков. Информация из остальных файлов будет использована только для стандартных блоков, взятых из файла **.rtp** с тем же самым базовым именем. Пользователь может добавлять базовые блоки к силовому полю путем размещения дополнительных файлов с тем же самым базовым именем в его рабочей папке. По умолчанию разрешено только определять дополнительные базовые блоки, но вызов ***pdb2gmx*** с опцией **-rtpo** разрешит замещение в силовом поле стандартных блоков по умолчанию таковыми из локальных файлов.

База данных остатков .rtp

Файлы баз данных остатков (**residue**) имеют расширение **.rtp**. Изначально этот файл содержал стандартные блоки (аминокислоты) для белков, а теперь является GROMACS интерпретацией файла **rt37c4.dat** GROMOS. Таким образом, файлы баз данных остатков содержат информацию (связи, заряды, зарядовые группы и несобственные двугранники) для часто используемых стандартных блоков. Желательно *не изменять* этот файл, т.к. он по умолчанию является входным файлом ***pdb2gmx***, но если все же нужно внести изменения, то внесите их в **.top** файл или в **.rtp** файл в рабочей папке. Задать топологии небольших молекул обычно проще путем прямого написания подключаемого файла ***.itp**. При добавлении нового белкового остатка в базу данных не забывайте добавлять имя остатка в файл **residuetypes.dat**, чтобы команды **grompp**, **make_ndx** и инструменты анализа могли распознать его как белковый остаток.

Файлы **.rtp** используют только команда ***pdb2gmx***. Как упомянуто ранее, нужная ей дополнительная информация — это только связи, заряды, зарядовые группы и несобственные двугранники, т.к. все остальное считывается из входного файла координат. Некоторые белки содержат нестандартные, но описанные в координатном файле, остатки. Вам нужно сделать стандартные блоки для этих нестандартных остатков, иначе не получится ***.top** файл. Это также касается таких молекул в координатном файле, как лиганды, многоатомные ионы,

кристаллизационные со-растворители и т.д. Остаток в базе данных записывается следующим образом:

```
[ bondedtypes ] ; mandatory
;bonds angles dihedrals impropers
    1      1      1      2      ; mandatory
[ GLY ] ; mandatory
[ atoms ] ; mandatory
;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
;atom1 atom2 b0 kb
N      H
N      CA
C      AC
C      O
-C      N
[ exclusions ] ; optional
;atom1 atom2
[ angles ] ; optional
;atom1 atom2 atom3 th0 cth
[ dihedrals ] ; optional
;atom1 atom2 atom3 atom4 phi0 cp mult
[ impropers ] ; optional
;atom1 atom2 atom3 atom4 q0 cq
N      -C      CA      H
-C      -C      AN      -O
[ ZN ]
[ atoms ]
ZN  ZN  2.000  0
```

Формат файла свободный, единственное ограничение — это то, что в строке может быть только одна запись. Первая секция в этом файле — это [*bondedtypes*], за которой идут четыре числа, указывающие типы функций взаимодействия для связей, углов и обоих типов двугранников. Этот файл содержит записи остатков, которые состоят из атомов и (опционально) связей, углов и обоих типов двугранников. Коды зарядовых групп означают их номера. Атомы одной зарядовой группы всегда должны быть расположены непрерывным блоком. При использовании базы данных водородов с [pdb2gmx](#) для добавления отсутствующих атомов водорода имена атомов, определенные в записи файла *.rtp*, должны точно соответствовать именному соглашению, используемому в базе данных водородов. Имена атомов в валентном

взаимодействии могут быть предварены минусом или плюсом, что означает, что атом находится в предшествующем или следующем остатке соответственно. Явно заданные параметры связей, углов и обоих типов двугранников перекрывают стандартные параметры в файлах *.itp*, их следует использовать только в особых случаях. Вместо значений параметров взаимодействия может быть указана строковая константа, это используется в файлах *.rtp* GROMOS-96. Эти константы копируются в файл топологии и там заменяются на параметры силового поля С-препроцессором в команде ***grompp*** с помощью выражений *#define*.

Команда ***pdb2gmx*** автоматически генерирует все углы. Это значит, что в силовом поле *gmx.ff* секция [angles] нужна только для перекрытия параметров из файла *.itp*. В силовом поле GROMOS-96 для всех углов должен быть указан тип функции.

Команда *pdb2gmx* автоматически генерирует один собственный двугранник для каждой способной вращаться связи, особенно с тяжелыми атомами. Если используется секция [dihedrals], то новые двугранники для связей, соответствующих указанным в ней двугранникам, генерироваться не будут. Возможно использовать нескольких двуграных функций для каждой способной вращаться связи.

Команда ***pdb2gmx*** устанавливает число исключения в 3, что значит, что невалентные взаимодействия между атомами, разделенными не более, чем тремя связями, будут исключаться. Невалентные парные взаимодействия будут генерироваться для всех пар атомов, разделенных тремя связями (за исключением пар атомов водорода). Когда нужно исключить больше взаимодействий или когда некоторые парные взаимодействия не должны быть сгенерированы, можно использовать секцию [exclusions], в которой указываются пары имен атомов на отдельных строках. Все невалентные и парные взаимодействия между этими атомами будут исключены.

База данных стандартных блоков остатков *.r2b*

Каждое силовое поле имеет свое собственное именное соглашение для остатков. Большинство остатков имеют постоянные имена, но некоторые, особенно с несколькими протонированными формами, могут иметь разные имена. Файлы *.r2b* используются для преобразования стандартных имен остатков в имена стандартных блоков силового поля. Если таких файлов нет в папке силового поля или остаток в них не упомянут, имя стандартного блока будет принято таким же, как имя остатка. Файл *.r2b* может содержать 2 или 5 колонок. В двухколоночном формате указаны имя остатка в первой колонке и имя стандартного блока во второй. Пятиколоночный формат имеет 3 дополнительные колонки для указания стандартного блока остатка, расположенного на N-конце, C-конце и на обоих концах одновременно. Это полезно, например, для силовых полей AMBER. Если не требуется наличия стандартного блока в каком-то из этих положений, то в соответствующей колонке должен стоять прочерк.

В GROMACS принято определенное соглашение для названий остатков. Его можно понять только с помощью файлов *.rb2* и *specbond.dat* (и еще кода программы *pdb2gmx*). Это соглашение важно только при добавлении типов остатков в файл *.rtp*. Для специальных связей с, например, группой гема, именное соглашение GROMACS вводится через файл *specbond.dat*, который потом при необходимости может быть преобразован файлом *.r2b*.

<i>ARG</i>	протонированный аргинин	
<i>ARGN</i>	нейтральный аргинин	
<i>ASP</i>	отрицательно заряженная аспарагиновая кислота	
<i>ASPH</i>	нейтральная аспарагиновая кислота	
<i>CYS</i>	нейтральный цистеин	
<i>CYS2</i>	цистеин с серной связью к другому цистеину или гему	
<i>GLU</i>	отрицательно заряженная глутаминовая кислота	
<i>GLUH</i>	нейтральная глутаминовая кислота	
<i>HISD</i>	нейтральный гистидин с протонированным N_{δ}	c
<i>HISE</i>	нейтральный гистидин с протонированным N_{ϵ}	c
<i>HISH</i>	положительно заряженный гистидин с обоими протонированными атомами азота	
<i>HIS1</i>	гистидин, связанный с гемом	
<i>LYSN</i>	нейтральный лизин	
<i>LYS</i>	протонированный лизин	
<i>HEME</i>	гем	

Табл. 4.3 Внутреннее именное соглашение в GROMACS.

База данных переименования атомов *xlateat.dat*

Силовые поля часто используют имена атомов, не соответствующие соглашениям IUPAC или PDB. Базы данных (файлы) *.arn* используются для перевода имен атомов в координатном файле в имена атомов в силовом поле. Атомы, не перечисленные в базе данных, сохраняют свои старые имена. Файл имеет три колонки: имя стандартного блока, старое и новое имена атомов соответственно. В имени остатка поддерживаются маски с вопросительным знаком (он соответствует одному любому символу).

Дополнительный файл общего переименования атомов *xlateat.dat* находится в папке share/top и переводит часто встречающиеся нестандартные имена атомов в координатном файле в соглашение IUPAC/PDB. При написании файлов силовых полей вы можете придерживаться стандартных имен атомов, тогда не потребуется последующий перевод, за исключением перевода из стандартных имен атомов в имена силового поля.

3.20. Базы данных водородов .hdb

Базы данных водородов хранятся в файлах **.hdb**. Они содержат информацию для команды **pdb2gmx** о том, как присоединить атомы водорода к уже существующим атомам. В версиях баз данных до GROMACS 3.3 атомы водорода назывались согласно атомам, к которым они были присоединены: первая буква имени атома была заменена на «Н». В последующих версиях атомы Н были перечислены явно, т.к. предыдущий способ был специфичен для белков и не мог быть использован для других молекул. Если к одному атому было присоединено более одного атома водорода, то в конец имени атома водорода добавляется номер. Например, при добавлении двух атомов водорода к ND2 в аспарагине атомы водорода будут именоваться HD21 и HD22. Это важно, т.к. именование атомов в файле **.rtp** должно быть тем же самым. Формат базы данных водородов:

```
; res # additions
#H add type H i j k
ALA 1
1 1 H N -C CA
ARG 4
1 2 H N CA C
1 1 HE NE CD CZ
2 3 HH1 NH1 CZ NE
2 3 HH2 NH2 CZ NE
```

В первой строке указано имя остатка (*ALA* или *ARG*) и количество атомных типов водорода, которые могут быть добавлены к этому остатку из базы данных водородов. После этого идут строки для каждого такого атома, в которых указаны:

- количество добавленных атомов Н

- метод добавления атома:

1. один атом Н в плоскости (например, кольца или пептидной связи)

Генерируется атом водорода (*n*) в плоскости атомов (*i,j,k*) на биссектрисе угла (*j-i-k*) на расстоянии 0,1 нм от атома *i* так, что углы (*n-i-j*) и (*n-i-k*) > 90°.

2. один атом Н (например, OH)

Генерируется один атом водорода (*n*) на расстоянии 0,1 нм от атома *i* так, что угол (*n-i-j*) = 109,5° и двугранный угол (*n-i-j-k*) = транс.

3. два атома Н в плоскости (-NH2)

Генерируются два атома водорода (*n1, n2*) на расстоянии 0,1 нм от атома *i* так, что углы (*n1-i-j*) = (*n2-i-j*) = 120°, а двугранные углы (*n1-i-j-k*) = цис и (*n2-i-j-k*) = транс, так что их имена соответствуют стандартам IUPAC [111].

4. два или три тетраэдрических атома Н (-CH3)

Генерируются два или три атома водорода на расстоянии 0,1 нм от атома i так, что углы $(n1-i-j) = (n2-i-j) = (n3-i-j) = 109,47^\circ$, а двугранные углы $(n1-i-j-k)$ = транс, $(n2-i-j-k)$ = транс + 120° и $(n3-i-j-k)$ = транс + 240° .

5. один тетраэдрический атом H ($C3CH$)

Генерируется один атом водорода (n') на расстоянии 0,1 нм от атома i в тетраэдрической конфигурации, углы $(n'-i-j) = (n'-i-k) = (n'-i-l) = 109,47^\circ$.

6. два тетраэдрических атома ($C-CH_2-C$)

Генерируются два атома Н ($n1, n2$) на расстоянии 0,1 нм от атома i в тетраэдрической конфигурации в плоскости, делящей пополам угол $(j-i-k)$, с углами $(n1-i-n2) = (n1-i-j) = (n1-i-k) = 109,47^\circ$.

7. два атома H, как в воде

Два атома водорода генерируются вокруг атома i в соответствии с SPC геометрией воды [74]. Ось симметрии будет проходить вдоль одной из осей координат (*will alternate between three coordinate axes in both directions*).

(8. и 9. будут даны позже)

10. три атома, как в воде

То же самое, что и в 7, но дополнительно генерируется частица на позиции кислорода с первой буквой имени, замененной на «М». Это нужно для использования с четырехатомными моделями воды, вроде TIP4P [96].

11. четыре атома, как в воде

То же самое, что и в 7, но генерируются еще две частицы на позиции кислорода с именами «LP1» и «LP2». Это нужно для использования с пятиатомными моделями воды, вроде TIP5P [112].

- Имя нового атома Н (или его префикс, например, HD2 для аспарагина; пример приведен выше).

- Три или четыре опорных атома (i,j,k,l), где первым всегда является атом, к которому присоединены атомы водорода. Другие два или три зависят от выбранного метода. Для воды (методов 7–9) есть только один опорный атом.

База данных окончаний (*aminoacids.n.tdb* и *aminoacids.c.tdb*)

Базы данных окончаний хранятся в *aminoacids.n.tdb* и *aminoacids.c.tdb* для N- и C-концов соответственно. Они содержат информацию для команды *pdb2gmx* о том, как следует присоединять новые атомы к существующим, какие атомы нужно удалить или изменить, и как следует добавлять новые валентные взаимодействия. Образец (из файла *gmx.ff/aminoacids.c.tdb*):

```
[ COO- ]
[ replace ]
C      C      C      12.01     1      0.27
[ add ]
2      8      O      C      CA      N
OM      15.9994   -0.635
```

```
[ delete ]
O
[ impropers ]
C      O1      O2      CA
[ None ]
```

Файл состоит из блоков, которые соответствуют разным типам окончаний, которые могут быть добавлены в молекулу. В этом примере [COO-] — это первый блок, соответствующий замене терминального атома углерода на непротонированную карбоксильную группу. [None] — это тип второго конца, который означает, что тот отделяется от молекулы. Именами блоков не могут быть *replace*, *add*, *delete*, *bonds*, *angles*, *dihedrals*, *impropers*. В противном случае они будут похожи на заголовки секций параметров блока, что может запутать читающих этот файл.

Для каждого блока есть следующие опции:

[*replace*]

Замена существующего атома на атом с другим атомным типом, названием, зарядом и/или массой. Эта запись может использоваться для замены атома, существующего одновременно во входном файле координат и в базе данных в файле *rtp*, или для только изменения названия атома во входном файле координат, чтобы оно соответствовало названию в силовом поле. В последнем случае еще должна быть секция [*add*], чтобы добавить такой же точно атом с новым именем (нужные для этого позиция в цепи и связи с другими атомами известны). Такой атом может присутствовать во входных координатах и сохраняться, или может не присутствовать и быть добавленным командой *pdb2gmx*. Для каждого заменяемого атома в строке должны быть указаны следующие поля:

- имя заменяемого атома
- новое имя (опционально)
- новый атомный тип
- новая масса
- новый заряд

[*add*]

Добавляет новые атомы. Для каждого добавленного атома(ов) (или группы атомов) требуется запись в две строки. Первая строка содержит те же поля, что и запись в базе данных водородов (количество атомов, способ добавления, имя нового атома, опорные атомы). Кроме описанных выше способов добавления реализованы еще два, специально для добавок на С-конце:

8. два карбонильных кислорода, $-COO-$

Два атома кислорода (n_1, n_2) генерируются в соответствии с правилом 3 на расстоянии 0,136 нм от атома i с углами $(n_1-i-j) = (n_1-i-j) = 117^\circ$.

9. карбонильные кислороды и водород, $-COOH$

Два атома кислорода ($n1$, $n2$) генерируются в соответствии с правилом 3 на расстоянии 0,123 нм ($n1$) и 0,125 нм ($n2$) от атома i , углы $(n1-i-j) = 121^\circ$ и $(n1-i-j) = 115^\circ$. Один атом водорода (n') генерируется возле $n2$ в соответствии с правилом 2, где $n-i-j$ и $n-i-j-k$ следует читать как $n'-n2-i$ и $n'-n2-i-j$ соответственно.

Во второй строке указываются параметры добавленного атома(ов) тем же образом, что и при замене атомов, т.е.

- атомный тип
- масса
- заряд
- зарядовая группа (опционально)

Как и в базе данных водородов, если к существующему атому присоединяются несколько атомов, то в конец имени каждого из них будет добавляться номер.

Обратите внимание, что как и в базе данных водородов, имя атома теперь находится на той же строке, что и опорные атомы, тогда как до GROMACS 3.3 оно было в начале второй строки. Если зарядовая группа не указана, то добавленный атом будет иметь ту же зарядовую группу, что и атом, с которым он связан.

[*delete*]

Удаляет существующие атомы. Одна строка — один атом.

[*bonds*], [*angles*], [*dihedrals*], [*impropers*]

Добавляют дополнительные параметры валентных взаимодействий. Формат идентичен формату файлов *rtp*.

База данных виртуальных позиций (.vsd)

Т.к. мы не можем полагаться на позиции атомов водорода во входных файлах, нам нужен специальный входной файл, чтобы определить способы и параметры, с которыми следует добавлять виртуальные позиции, соответствующие атомам водорода. Для более сложных способов создания ВП (например, когда целые ароматические боковые цепи делаются жесткими) нам также нужны данные о равновесных длинах связей и углах для всех атомов боковой цепи. Эта информация указывается в файле *.vsd* для каждого силового поля. Как и для окончаний, есть по одному такому файлу для каждого класса остатков в файле *.rtp*. База данных ВП не является простым списком информации. Первая группа секций указывает, какие центры масс (обычно именуемые MCH3/MNH3) следует использовать с группами CH3, NH3 и NH2. В зависимости от равновесных длин связей и углов между атомами водорода и тяжелыми атомами нужно применять слегка различные ограничения расстояний между этими центрами масс. **Обратите внимание**, что нам *не нужно* указывать реальные параметры (они находятся автоматически), а только тип используемого центра масс. Чтобы сделать это, есть три секции: [CH3], [NH3] и [NH2]. В каждой из них по три колонки. Первая колонка — это атомный тип двух (трех) атомов H, вторая — атомный тип следующего тяжелого атома, с которым связан этот, а третья — тип используемого центра

масс. В секции [NH2] также можно указать planar во второй колонке, при этом будет другой способ создания, без центра масс. В разных силовых полях реализован разный взгляд на то, должна ли группа NH2 быть плоской или нет, но мы стремились строго придерживаться стандартных равновесных параметров из силового поля.

Вторая часть базы данных виртуальных позиций содержит явные равновесные длины связей и углы пар/триплетов атомов в ароматических боковых цепочках. Эти записичитываются специальными процедурами в коде генерации ВП, так что если вы захотите ее расширить, например, нуклеиновыми кислотами, вам также понадобится записать туда новый код. Эти секции именуются как сокращенные названия аминокислот ([PHE], [TYR], [TRP], [HID], [HIE], [HIP]) и просто содержат 2 или 3 колонки с именами атомов и длинами связей (нм) или углами (градусы). **Обратите внимание**, что эти величины взяты из равновесной геометрии целой молекулы, поэтому они могут не совпадать с равновесным величинами обособленных связей (углов), если на молекулу действуют внешние силы.

Специальные связи (*specbond.dat*)

Подход, используемый командой *pdb2gmx* для генерации связей между остатками, основан на соединении типа «голова к хвосту» атомов главной цепи белка разных остатков при построении макромолекул. В некоторых случаях, например, дисульфидах, геме, разветвленных полимерах и т.д., необходимо создать связи между остатками, не принадлежащие к белковому скелету. Для этого предназначен файл *specbond.dat*.

добавлении новых записей не забудьте увеличивать это число. Остальные строки файла определяют создание связей. Формат строк следующий:

resA atomA nbondsA resB atomB nbondsB length newresA newresB

В колонках указываются:

resA — имя остатка А, участвующего в связи.

atomA — имя атома в остатке А, образующего связь.

nbondsA — общее число связей, которое может образовать atomA.

resB — имя остатка В, участвующего в связи.

atomB — имя атома в остатке В, образующего связь.

nbondsB — общее число связей, которое может образовать atomB.

length — опорная длина связи. Если расстояние между atomA и atomB во входном файле координат pdb2gmx не равно $length \pm 10\%$, то связь не будет сформирована.

newresA — новое имя остатка А (если нужно). Некоторые силовые поля используют, например, CYS2 для цистеина, связанного дисульфидной связью или связанного с гемом.

newresB — аналогично, новое имя остатка В.

4.5. Форматы файлов GROMACS

4.5.1 Файл топологии (*.top*)

Файлы топологии составляются согласно спецификации GROMACS для молекулярных топологий и могут быть сгенерированы командой *pdb2gmx*. Также указаны размерности всех параметров, какие взаимодействия могут быть модифицированы в вычислениях свободных энергий, какие валентные взаимодействия учитываются командой *grompp* при генерировании исключений и какие валентные взаимодействия могут быть преобразованы *grompp* в ограничения.

Описание разметки файла *.top*:

- Точки с запятой начинают комментарии. Они заканчиваются с переводом строки.
- В строках, оканчивающихся на \, перевод строки игнорируется.
- Названия секций заключаются в [].
- Иерархия топологий состоит из трех уровней:
 - уровень параметров, определяющий некоторые спецификации силового поля (см. табл. 5.4);
 - уровень молекул, который должен содержать определения одной или нескольких молекул;
 - уровень системы, содержащий только информацию о системе ([*system*] и [*molecules*]).
 - Значения должны быть разделены пробелами или табуляторами, но не запятыми.
 - Атомы и молекулы должны нумероваться последовательно, начиная с 1.
 - Файл считывается только за один проход, поэтому недопустимы никакие ссылки вперед: значения параметров должны быть объявлены перед их использованием.
 - Исключения могут генерироваться на основании связей и доопределяться вручную.
 - Типы функций валентных взаимодействий могут быть взяты из стандартных параметров для этих атомных типов или быть указаны явно в их определениях.
 - Возможно применять несколько валентных взаимодействий одного типа к одинаковым атомам.
 - Весьма желательны комментарии и пустые строки.
 - Начиная с версии GROMACS 3.1.3 все секции уровня параметров могут использоваться несколько раз, и нет ограничений на порядок. Единственное, атомный тип должен быть определен перед его использованием в определениях других параметров.
 - Если параметры некоторого взаимодействия определены несколько раз для одной и той же комбинации атомных типов, то используется последнее определение; начиная с GROMACS 3.1.3 команда *grompp* выдает предупреждение о переопределениях параметров.

- Указание секций `[atoms]`, `[bonds]`, `[pairs]`, `[angles]` и т.п. без указания перед этим `[moleculetype]` бессмысленно, в таких случаях выдается предупреждение.
- Аналогично, бессмысленно указывать секцию `[molecules]`, если перед ней нет `[system]`, при этом будет выдано предупреждение.
- После `[system]` допустима только секция `[molecules]`.
- Указание неизвестной строки в `[]` приведет к игнорированию всех данных после нее и выдаче сообщения об ошибке.

Общие параметры

	Секция	№ а.т.	Тип ф.	Параметры	И. п.
обязательная секция	<code>defaults</code>			тип невалентной функции; правило комбинир-ния (<i>cr</i>); генерировать ли пары (no/yes); fudge LJ; fudge QQ (числа)	
обязательная секция	<code>atomtypes</code>			атомный тип; <i>m</i> (а.е.м); <i>q</i> (е); тип частицы; <i>V_(cr)</i> ; <i>W_(cr)</i>	
	<code>bondtypes</code> <code>pairtypes</code> <code>angletypes</code> <code>dihedraltypes (*)</code> <code>constrainttypes</code>			(см. соответствующие секции табл. 5.5 без types в конце)	
ЛД взаимод-я бакингемовы взаимод-я	<code>nonbond_params</code> <code>nonbond_params</code>	2 2	1 2	<i>V_(cr)</i> ; <i>W_(cr)</i> <i>a</i> (кДж/моль); <i>b</i> (нм ⁻¹); <i>C₆</i> (кДж·моль ⁻¹ ·нм ⁶)	

Определение молекул(ы)

обязательно	<code>moleculetype</code>			имя молекулы; <i>nex(nrexcl)</i>	
обязательно	<code>Atoms</code>	1		атомный тип; номер остатка; имя остатка; имя атома; номер зарядовой группы; <i>q</i> (е); <i>m</i> (а.е.м)	тип, <i>q</i> , <i>m</i>
внутримолекулярные взаимодействия и определения геометрий					

Система

обязательно	<code>system</code>	имя системы
обязательно	<code>molecules</code>	имя молекулы; количество таких молекул

Файл топологии (.top)

где

«№ а.т.» — количество атомных типов,

«Тип ф.» — тип функции,

«И. п.» — указывает, какие параметры могут быть интерполированы во время моделирований свободной энергии,

(*cr*) — правило комбинирования, определяющее тип ЛД параметров

(*) — для dihedraltypes можно указать все четыре атома или только два внутренних (для неправильных двугранных углов — внешних) атома

(*nrexcl*) — атомы, на расстоянии *nex* и ближе связей от данного, будут исключены для невалентных взаимодействий с ним.

Таким образом в начале файла, сначала задается тип функции (способ описания) невалентных взаимодействий (секция [defaults]). Затем задаются атомные типы. После них перечисляются стандартные параметры невалентных дальнодействующих ([nonbond_params]) и навалентных парных 1-4 ([pairtypes]) взаимодействий между парами разных типов атомов (количество параметров зависит от выбранного типа функции невалентных взаимодействий — ЛД или бакингемовой). После этого перечисляются стандартные параметры для валентных взаимодействий (связей, угловых, двугранных) и ограничений между разными атомными типами (секции [*types]).

Следующий уровень — уровень молекул. Сначала (секция [molecletype]) указывается тип молекулы, который имеет только два параметра — имя типа и число связей, пары атомов на расстоянии равном или меньшем которого будут исключены из списка взаимодействующих невалентно. После этого в секции [atoms] описываются атомы, входящие в состав этих молекул. Затем указывается, какие из перечисленных атомов связаны между собой (т.е. образуют связи, углы, двугранные углы), участвуют в 1-4 взаимодействиях или находятся в ограничениях. При этом конкретные параметры этих связей, углов, двугранных углов, 1-4 взаимодействий и ограничений будут определяться, исходя из типов участвующих в них атомов по записям в [*types], если они (параметры) не указаны явно.

Наконец, на уровне системы указываются имя системы ([system]) и находящиеся в ней молекулы (ионы) ([molecules]).

При этом не обязательно, чтобы в одном файле топологии были все перечисленные секции. Так, атомные типы и стандартные параметры взаимодействий обычно берутся из силового поля и с помощью механизма включений включаются в файл топологии, содержащий описание молекул и (если это не .itp файл) системы.

Пример файла топологии *urea.top*:

```
;  
; Example topology file  
;  
; The force field files to be included #include "gmx.ff/forcefield.itp"  
[ molecletype ]  
;name nrexcl  
Urea 3  
[ atoms ]  
;nr type resnr residu atom cgnr charge  
1 C 1 UREA C1 1 0.683  
2 O 1 UREA O2 1 -0.683  
3 NT 1 UREA N3 2 -0.622  
4 H 1 UREA H4 2 0.346  
5 H 1 UREA H5 2 0.276
```

6 NT 1 UREA N6 3 -0.622
 7 H 1 UREA H7 3 0.346
 8 H 1 UREA H8 3 0.276
 [bonds]
 ;ai aj funct b0 kb
 3 4 1 1.000000e-01 3.744680e+05
 3 5 1 1.000000e-01 3.744680e+05
 6 7 1 1.000000e-01 3.744680e+05
 6 8 1 1.000000e-01 3.744680e+05
 1 2 1 1.230000e-01 5.020800e+05
 1 3 1 1.330000e-01 3.765600e+05
 1 6 1 1.330000e-01 3.765600e+05
 [pairs]
 ;ai aj funct c6 c12
 2 4 1 0.000000e+00 0.000000e+00
 2 5 1 0.000000e+00 0.000000e+00
 2 7 1 0.000000e+00 0.000000e+00
 2 8 1 0.000000e+00 0.000000e+00
 3 7 1 0.000000e+00 0.000000e+00
 3 8 1 0.000000e+00 0.000000e+00
 4 6 1 0.000000e+00 0.000000e+00
 5 6 1 0.000000e+00 0.000000e+00
 [angles]
 ;ai aj ak funct th0 cth
 1 3 4 1 1.200000e+02 2.928800e+02
 1 3 5 1 1.200000e+02 2.928800e+02
 4 3 5 1 1.200000e+02 3.347200e+02
 1 6 7 1 1.200000e+02 2.928800e+02
 1 6 8 1 1.200000e+02 2.928800e+02
 7 6 8 1 1.200000e+02 3.347200e+02
 2 1 3 1 1.215000e+02 5.020800e+02
 2 1 6 1 1.215000e+02 5.020800e+02
 3 1 6 1 1.170000e+02 5.020800e+02
 [dihedrals]
 ;ai aj ak al funct phi cp mult
 2 1 3 4 1 1.800000e+02 3.347200e+01 2.000000e+00
 6 1 3 4 1 1.800000e+02 3.347200e+01 2.000000e+00
 2 1 3 5 1 1.800000e+02 3.347200e+01 2.000000e+00
 6 1 3 5 1 1.800000e+02 3.347200e+01 2.000000e+00
 2 1 6 7 1 1.800000e+02 3.347200e+01 2.000000e+00
 3 1 6 7 1 1.800000e+02 3.347200e+01 2.000000e+00
 2 1 6 8 1 1.800000e+02 3.347200e+01 2.000000e+00
 3 1 6 8 1 1.800000e+02 3.347200e+01 2.000000e+00
 [dihedrals]

```

;ai aj ak al funct q0 cq
3 4 5 1 2 0.000000e+00 1.673600e+02
6 7 8 1 2 0.000000e+00 1.673600e+02
1 3 6 2 2 0.000000e+00 1.673600e+02
[ 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)
; Include SPC water topology
#include "spc.itp"
[ system ]
Urea in Water
[ molecules ]
;molecule name nr.
Urea 1
SOL 1000

```

Пояснения:

[defaults]:

- *nbfinc* — тип функции невалентного взаимодействия: 1 — ЛД, 2 — Бакингема
- *comb-rule* — правило комбинирования:

1. Для потенциала Леннард-Джонса: указаны $C^{(6)}$ и $C^{(N)}$, и $C_{ij}^M \text{sqrt}(C_i^M C_j^M)$ ($M = 6, N$). Значение по умолчанию для $N=12$, но оно может быть переопределено (например, $N=9$) путем использования последнего параметра в этой строке.

2. указаны σ и ϵ , $\sigma_{ij} = 1/2(\sigma_i + \sigma_j)$ и $\epsilon_{ij} = \text{sqrt}(\epsilon_i \epsilon_j)$

3. указаны σ и ϵ , и $\sigma_{ij} = \text{sqrt}(\sigma_i \sigma_j)$ и $\epsilon_{ij} = \text{sqrt}(\epsilon_i \epsilon_j)$

gen-pairs — вычислять ли параметры парных взаимодействий. По умолчанию no, т.е. параметры 1-4 взаимодействий берутся из списка [pairtypes]. Если их там нет, то выдается критическая ошибка. При значении yes отсутствующие в нем 1-4 параметры вычисляются из обычных ЛД параметров через fudgeLJ.

- *fudgeLJ* — значение, на которое умножаются параметры ЛД взаимодействий для получения параметров парных ЛД 1-4 взаимодействий. По умолчанию 1.

- *fudgeQQ* — аналогично для кулоновского взаимодействия, по умолчанию 1.

- *N* — степень члена отталкивания в 6-*N* потенциале (только для невалентного (не 1-4) ЛД потенциала). Начиная с GROMACS 4.5, команда ***mdrun*** также читает и применяет для значений *N*, не равных 12, табулированные функции взаимодействия (в более старых версиях нужно было использовать пользовательские табулированные взаимодействия).

Обратите внимание, что *gen-pairs*, *fudgeLJ*, *fudgeQQ* и *N* опциональны. *fudgeLJ* используется только при включенном вычислении параметров парных взаимодействий, а *fudgeQQ* используется всегда. Однако если вы хотите указать *N*, то вам нужно будет указать и остальные параметры.

#include "gmx.ff/forcefield.itp": включает валентные и невалентные параметры силового поля. gmx.ff — это имя используемого поля.

[*moleculetype*]: определяет имя молекулы в этом файле топологии и устанавливает число исключения *nrexcl* = 3: невалентные взаимодействия между атомами, разделенными не больше чем тремя связями, будут исключаться.

[*atoms*]: перечисляет атомы молекулы. *nr* и *type* фиксированы, остальные определяются пользователем. Так, можно как угодно назвать *atom*, изменить *cgnr* (по возможности, зарядовая группа должна быть электронейтральна) и заряды.

[*bonds*]: пары атомов, образующих связи, и параметры связей (опциональны)

[*pairs*]: внутримолекулярные 1-4 ЛД и кулоновские взаимодействия

[*angles*]: тройки атомов, образующих валентные углы, и параметры (опциональны)

[*dihedrals*]: в этом случае есть 9 двугранных углов (*funct* = 1), 3 неправильных двугранных угла (*funct* = 2) и ни одного двугранных угла Рикаэрта-Беллеманса. Если нужно добавить двугранники Рикаэрта-Беллеманса, сделайте следующее (на примере декана):

[*dihedrals*]

;ai aj ak al *funct* c0 c1 c2

1 2 3 4 3

2 3 4 5 3

и не забудьте убрать 1-4 взаимодействие в [*pairs*]!

[*position_restraints*]: гармонические ограничения выбранных частиц к опорным позициям. Опорные позиции считаются grompp из отдельного координатного файла.

#include "spc.itp": включает готовый файл топологии воды

[*system*]: название системы, задается пользователем

[*molecules*]: определение общего числа (под)молекул в системе, определенных в этом файле. В этом примере в системе находится 1 молекула мочевины, растворенная в 1000 молекул воды. Тип молекул *SOL* определен в файле *spc.itp*. Каждое указанное здесь имя должно быть упомянуто в [*moleculetype*] ранее в топологии. Порядок определения блоков типов молекул и количеств этих молекул должен соответствовать координатному файлу, подаваемому вместе с файлом топологии *grompp*. Блоки молекул не обязательно должны идти непрерывно, но некоторые команды (например, *genion*) могут работать только с первым или последним отдельным блоком молекул нужного типа. Эти блоки не имеют ничего общего с группами.

4.5.2 Файл топологии для отдельной молекулы (.itp)

Если вы создаете файл топологии, который будете часто использовать (например, для молекулы воды), разумно создать файл **.itp** (как, например, уже сделанный **spc.itp**). В нем будет информация только об одной отдельной молекуле, и его можно будет повторно использовать в многих системах без повторного вызова pdb2gmx или ручного копирования и вставки. Пример такого файла:

```
[ moleculetype ]
;name nrexcl
Urea 3
[ atoms ]
;nr type resnr residu atom cgnr charge
1 C 1 UREA C1 1 0.683
.....
.....
8 H 1 UREA H8 3 0.276
[ bonds ]
;ai aj funct c0 c1
3 4 1 1.000000e-01 3.744680e+05
.....
.....
1 6 1 1.330000e-01 3.765600e+05
[ pairs ]
;ai aj funct c0 c1
2 4 1 0.000000e+00 0.000000e+00
5 6 1 0.000000e+00 0.000000e+00
[ angles ]
;ai aj ak funct c0 c1
1 3 4 1 1.200000e+02 2.928800e+02
3 1 6 1 1.170000e+02 5.020800e+02
[ dihedrals ]
;ai aj ak al funct c0 c1 c2
2 1 3 4 1 1.800000e+02 3.347200e+01 2.000000e+00
3 1 6 8 1 1.800000e+02 3.347200e+01 2.000000e+00

[ dihedrals ]
;ai aj ak al funct c0 c1
3 4 5 1 2 0.000000e+00 1.673600e+02
6 7 8 1 2 0.000000e+00 1.673600e+02
1 3 6 2 2 0.000000e+00 1.673600e+02
С использованием *.itp получится очень короткий файл *.top раствора мочевины:
; The force field files to be included
#include "gmx.ff/forcefield.itp"
```

```

; Include urea topology
#include "urea.itp"
; Include SPC water topology
#include "spc.itp"
[ system ]
Urea in Water
[ molecules ]
;molecule name number
Urea 1
SOL 1000

```

Конструкции *ifdef*

Конструкции *#ifdef* — это очень мощное средство. Благодаря нему можно указать разные параметры для одной молекулы в одном файле **.top*. Приведем пример TFE, в котором есть возможность использовать разные заряды на атомах: заряды, полученные De Loof и др. [113] или Van Buuren и Berendsen [97]. Фактически, возможно использовать большую часть функций C-препроцессора *cpr*, т.к. команда ***grompp*** содержит схожие функции препроцессинга для обработки файлов. Для использования зарядов De Loof нужно:

- указать опцию *define = -DDellof* в файле **.mdp* (содержащем входные параметры ***grompp***) или строку *#define DDellof* в файле **.top* или **.itp* и
- поместить конструкции *#ifdef* в файл **.top* (после *#define*, если оно есть):

```

...
[ 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
...

```

Этот механизм используется командой ***pdb2gmx*** для реализации опциональных ограничений позиций с помощью **#include** в *.itp* файле, содержание которого будет осмысленным только при определенном, корректно указанном **#define**.

4.5.3 Файл координат (*.gro*)

Файлы с расширением *.gro* содержат молекулярную структуру в формате GROMOS-87. Образец части такого файла:

MD of 2 waters, reformat step, PA aug-91

```

6
1 WATER OW1 1 0.126 1.624 1.679 0.1227 -0.0580 0.0434
1 WATER HW2 2 0.190 1.661 1.747 0.8085 0.3191 -0.7791
1 WATER HW3 3 0.177 1.568 1.613 -0.9045 -2.6469 1.3180
2 WATER OW1 4 1.275 0.053 0.622 0.2519 0.3140 -0.1734
2 WATER HW2 5 1.337 0.002 0.680 -1.0641 -1.1349 0.0257
2 WATER HW3 6 1.326 0.120 0.568 1.9427 -0.8216 -0.0244
1.82060 1.82060 1.82060

```

Формат фиксирован, т.е. каждый столбец имеет фиксированное положение. Если вы хотите прочитать такой файл в собственной программе без использования библиотек GROMACS, то вы можете использовать следующие форматы:

Формат С: «%5i%5s%5s%5i%8.3f%8.3f%8.3f%8.4f%8.4f%8.4f»

Или более точно, с заголовком и т.д.:

```

"%s\n", Title
"%5d\n", natoms
for (i=0; (i<natoms); i++) {
    "%5d%5s%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]
Формат в фортране: (i5,2a5,i5,3f8.3,3f8.4)

```

Таким образом, *confin.gro* — это координатный файл GROMACS. Он практически совпадает с таковым GROMOS-87 (для пользователей GROMOS: когда используется с *ntx* = 7). Единственная разница заключается в том, что GROMACS для ячейки использует тензор, а не вектор.

4.5.4 Файлы силовых полей

В GROMACS 4.5, по умолчанию доступны 14 силовых полей. Силовые поля определяются по наличию папок *<name>.ff* в подпапке GROMACS */share/top* и/или рабочей папке. Информация относительно размещения файлов силового поля выводится командой *pdb2gmx*, так что вы легко можете отслеживать то, какая версия силового поля используется, если вы его модифицировали. Силовые поля, включенные в GROMACS:

AMBER03 (Duan и др., J. Comp. Chem. 24, 1999-2012, 2003)
AMBER94 (Cornell и др., JACS 117, 5179-5197, 1995)
AMBER96 (Kollman и др., Acc. Chem. Res. 29, 461-469, 1996)
AMBER99 (Wang и др., J. Comp. Chem. 21, 1049-1074, 2000)
AMBER99SB (Hornak и др., Proteins 65, 712-725, 2006)
AMBER99SB-ILDN (Lindorff-Larsen и др., Proteins 78, 1950-58, 2010)
AMBERGS (Garcia & Sanbonmatsu, PNAS 99, 2782-2787, 2002)
CHARMM27 all-atom (with CMAP)
GROMOS96 43a1
GROMOS96 43a2 (улучшенные двугранные углы для алканов)
GROMOS96 45a3 (Schuler JCC 2001 22 1205)
GROMOS96 53a5 (JCC 2004 vol 25 pag 1656)
GROMOS96 53a6 (JCC 2004 vol 25 pag 1656)
OPLS-AA/L all-atom (2001 aminoacid dihedrals)

Есть также некоторые дополнительные, нерекомендуемые к использованию силовые поля, которые команда *pdb2gmx* также показывает в списке для выбора силового поля, но мы не советуем использовать их в новых моделированиях.

Силовое поле включается в начале файла топологии директивой *#include*, после которой идет *<name>.ff/forcefield.itp*. Эта директива включает файл силового поля, который в свою очередь может включать другие файлы силовых полей. Все силовые поля имеют сходную структуру. Приведем, например, *gmx.ff/forcefield.itp*:

```
#define _FF_GROMACS
#define _FF_GROMACS1
[ defaults ]
;nbfunc comb-rule gen-pairs fudgeLJ fudgeQQ
1 1 no 1.0 1.0
#include "ffnonbonded.itp"
```

```
#include "ffbonded.itp"
```

Первый `#define` может использоваться в топологиях для разбора данных, специфических для всех силовых полей GROMACS, второй — специфических для конкретно этого силового поля. Включенный файл `ffnonbonded.itp` содержит все атомные типы и невалентные параметры, `ffbonded.itp` — все валентные параметры.

Для каждого силового поля есть несколько файлов, используемых только командой `pdb2gmx`. Это: базы данных остатков (`.rtp`), базы данных водородов (`.hdb`), две БД для концевых остатков (`.n.tdb` и `.c.tdb`) и БД атомных типов (`.atp`).

Раздел 5 Программы и параметры моделирования

Вся эта глава есть в формате HTML в папке GROMACS.

Путь по умолчанию: `/usr/local/gromacs/share/html/online.html`

или, если вы устанавливали из пакетов: `/usr/local/share/gromacs/html/online.html`

Также она есть на сайте GROMACS: <http://manual.gromacs.org/current/>

5.1 Типы файлов GROMACS

В табл. 7.1. перечислены и вкратце описаны типы файлов, используемые GROMACS. Более подробное описание есть в HTML справке или на сайте.

Файлы, записанные в формате `xdr`, могут быть прочитаны на любой архитектуре с GROMACS 1.6 или новее, если конфигурационный скрипт найдет в системе библиотеки XDR. Они обычно всегда есть в UNIX, т.к. нужны для поддержки NFS.

Имя и расширение по умолчанию	Тип	Параметр по умолчанию	Описание
<code>atomtp.atp</code>	Asc		Атомные типы, используемые pdb2gmx
<code>eiwit.brk</code>	Asc		Файл банка данных Brookhaven
<code>state.cpt</code>	xdr	<code>-f</code>	Контрольная точка
<code>nnnice.dat</code>	Asc	<code>-f</code>	Общий файл с данными
<code>user.dlg</code>	Asc	<code>-c</code>	Диалоговое окно ngmx
<code>sam.edi</code>	Asc	<code>-c</code>	ED sampling input
<code>sam.edo</code>	Asc	<code>-c</code>	ED sampling output
<code>ener.edr</code>		<code>-c</code>	Общий файл энергий: edr ene
<code>ener.edr</code>	xdr	<code>-o</code>	Файл энергий в портируемом формате xdr
<code>ener.ene</code>	Bin	<code>-l</code>	Файл энергий
<code>eiwit.ent</code>	Asc	<code>-f</code>	Запись в банке данных белков
<code>plot.eps</code>	Asc	<code>-m</code>	Файл Encapsulated Postscript™
<code>conf.esp</code>	Asc	<code>-n</code>	Файл координат в формате ESPResSo
<code>gtraj.g87</code>	Asc	<code>-o</code>	ASCII формат траекторий GROMOS-87
<code>conf.g96</code>	Asc	<code>-f</code>	Файл координат в формате GROMOS-96
<code>conf.gro</code>	Asc	<code>-o</code>	Файл координат в формате GROMOS-87
<code>conf.gro</code>		<code>-p</code>	Структура: gro g96 pdb esp tpr tpb tpa
<code>out.gro</code>		<code>-s</code>	Структура: gro g96 pdb esp
<code>polar.hdb</code>	Asc	<code>-s</code>	База данных водородов
<code>topinc.itp</code>	Asc	<code>-s</code>	Включаемый файл топологии

<i>run.log</i>	<i>Asc</i>	<i>-s</i>	Файл протокола
<i>ps.m2p</i>	<i>Asc</i>	<i>-f</i>	Входной файл для mat2ps
<i>ss.map</i>	<i>Asc</i>	<i>-f</i>	Сопоставление цветам данных матрицы
<i>ss.mat</i>	<i>Asc</i>	<i>-o</i>	Матричный файл данных
<i>gromppmdp</i>	<i>Asc</i>		Входной файл grompp с параметрами МД
<i>hessian mtx</i>	<i>Bin</i>		Матрица Гессе
<i>index.ndx</i>	<i>Asc</i>		Индексный файл
<i>hello.out</i>	<i>Asc</i>		Общий выходной файл
<i>eiwit.pdb</i>	<i>Asc</i>		Файл банка данных белков
<i>residue.rtp</i>	<i>Asc</i>		Типы остатков, используемые pdb2gmx
<i>doc.tex</i>	<i>Asc</i>		Файл LaTeX
<i>topol.top</i>	<i>Asc</i>		Файл топологии
<i>topol.tpb</i>	<i>Bin</i>		Двоичный входной файл для моделирования
<i>topol.tpr</i>			Общий — «—: tpr tpb tpa
<i>topol.tpr</i>	<i>xdr</i>		Портируемый xdr — «—
<i>topol.tpr</i>			Структура и массы (БД): tpr tpb tpa gro g96 pdb
<i>traj.trj</i>	<i>Bin</i>		Файл траектории (зависящий от архитектуры)
<i>traj.trr</i>			Траектория полной точности: trr trj cpt
<i>traj.trr</i>	<i>xdr</i>		Траектория в портируемом формате xdr
<i>root.xpm</i>	<i>Asc</i>		X PixMap-совместимый файл матрицы
<i>traj.xtc</i>			Траектория, входной: xtc trr trj cpt gro g96 pdb
<i>traj.xtc</i>			Траектория, выходной: xtc trr trj gro g96 pdb
<i>traj.xtc</i>	<i>xdr</i>		Сжатая траектория, портируемый формат xdr
<i>graph.xvg</i>	<i>Asc</i>		Файл xvgr/xmgr

Таблица 5.1 Типы файлов GROMACS.

5.2 Параметры и файл управления запуском (*.mdp*)

Общие сведения

Параметры по умолчанию указаны в скобках. Первая опция в списке всегда является опцией по умолчанию. Единицы измерения указаны в квадратных скобках. Разницы между подчеркиванием и дефисом нет. Доступен образец *.mdp* файла, который должен подойти для запуска обычного моделирования. В случае необходимости отредактируйте его нужным образом.

Препроцессор

include:

Папки, включаемые в вашу топологию. Формат:

-I/home/john/mylib -I./otherlib

define:

Определения для препроцессора. По умолчанию их нет. Можно указывать любые определения для задания параметров в топологиях. Изначально доступные варианты:

-DFLEXIBLE

Указывает команде *grompp* включить в топологию гибкую модель воды вместо жесткой. Это может пригодиться при анализе нормальных мод.

-DPOSRES

Указывает команде *grompp* включить файл *posre.itp* в топологию. Используется с ограничениями позиций.

Управление запуском

integrator:

(Несмотря на название, в списке также перечислены алгоритмы, не являющиеся алгоритмами интегрирования, например, алгоритмы минимизации энергии).

md:

Алгоритм *leap-frog* для интегрирования ньютоновых уравнений движения.

md-vv:

Скоростная форма алгоритма Верле для интегрирования ньютоновых уравнений движения. Для *NVE* моделей, начатых с соответствующих точек одной и той же траектории, развития траекторий получается аналитически (но не двоично) идентичными с таковыми *md leap-frog* интегратора. Кинетическая энергия определяется из скоростей на полном шаге, и поэтому она слегка больше. Преимуществами этого интегратора является поддержка более точных обратимых, основанных на разложении Троттера, привязок Нозе-Гувера и Парринелло-Рамана, и вывод полношаговых скоростей. Все это получается за счет дополнительных вычислений, особенно при использовании ограничений, и дополнительных передач данных при распараллеливании. Впрочем, точность интегратора *md* достаточна для большинства моделей.

md-vv-avek:

Скоростная форма интегратора Верле, идентичная *md-vv* за тем исключением, что кинетическая энергия определяется как среднее двух полу шаговых кинетических энергий, как в интеграторе *md*, а, значит, и более точно. При использовании алгоритмов Нозе-Гувера и/или Парринелло-Рамана это немного увеличивает вычислительные затраты.

sd:

Точный *leap-frog* стохастический динамический интегратор. На каждом шаге интегрирования для каждой степени свободы требуются четыре гауссовых случайных числа. При использовании ограничений значения координаты ограничиваются дважды за один шаг. В зависимости от вычислительных затрат на вычисление сил это может занять существенную часть времени моделирования. Температура одной или нескольких групп атомов (*tc_grps*) устанавливается через *ref_t [K]*, а обратная постоянная трения каждой группы — через *tau_t [пс]*. Параметр *tcoupl* игнорируется. Генератор случайных чисел инициализируется числом *ld_seed*. При использовании термостатирования подходящим значением *tau_t* будет 2 пс, т.к. это приведет к трению меньшему, чем внутреннее трение в воде, но достаточному для удаления избыточной теплоты (если не используются *cut-off* или *reaction-field* электростатика). **Обратите внимание:** отклонения температуры спадают в два раза быстрее, чем в термостате Беренсена с такой же *tau_t*.

sdl:

Эффективный *leap-frog* стохастический динамический интегратор. Этот интегратор эквивалентен *sd* за тем исключением, что он требует только одно гауссово случайное число и ограничивает координаты один раз за шаг. Однако он менее точен. Для воды относительная погрешность с ним равна 0,5

$\text{delta_t}/\text{tau_t}$, но для других систем может быть большей. Используйте осторожно и вместе с термостатированием.

bd:

Интегратор Эйлера для броуновской или позиционной ланжевеновой динамики. Скорость в нем равняется силе, деленной на коэффициент трения (bd_fric [а.е.м./пс]), плюс случайный температурный шум (ref_t). При $\text{bd_fric}=0$ коэффициент трения для каждой частицы вычисляется как $\text{massa}/\text{tau_t}$, как и в интеграторе sd. Генератор случайных чисел инициализируется числом ld_seed .

steep:

Метод скорейшего спуска для минимизации энергии. Максимальный размер шага устанавливается через emstep [нм], допуск — через emtol [$\text{кДж}\cdot\text{моль}^{-1}\cdot\text{нм}^{-1}$].

cg:

Метод сопряженных градиентов для минимизации энергии. Допуск равняется emtol [$\text{кДж}\cdot\text{моль}^{-1}\cdot\text{нм}^{-1}$]. Он становится эффективнее, если периодически производится один шаг скорейшего спуска, период задается nscgsteep . Для минимизации энергии перед анализом нормальных мод, который требует очень большой точности, GROMACS должен быть собран с двойной точностью.

l-bfgs:

Квазиньютоновский метод минимизации энергии с подходом уменьшения потребления памяти Broyden-Fletcher-Goldfarb-Shanno. На практике он сходится быстрее метода сопряженных градиентов, но из-за необходимости корректировочных шагов он пока еще не распараллелен.

nm:

Осуществляется анализ нормальных мод. GROMACS должен быть собран с двойной точностью.

tpi:

Тест на введение частиц. Последняя молекула в топологии принимается как вводимая. Нужно предоставить траекторию с помощью опции `mdrun -rerun`, она не должна содержать вводимую молекулу. Введение производится `nsteps` раз на каждом шаге в случайных положениях и ориентациях. Если `nstlist` больше 1, то вокруг одной и той же точки в радиусе `rtpi` будет производиться `nstlist` вставок с использованием одного списка соседей (и одной энергии дальнодействующих взаимодействий при `rvdw` или `rcoulomb > rlist`, что допустимо только для одноатомных молекул). Создание списка соседей затратно, поэтому логично при одном и том же списке практически без затрат осуществлять еще несколько дополнительных вставок. Начальное число генератора случайных чисел задается через `ld_seed`. Температура для вычисления больцмановых весов задается через `ref_t`, она должна соответствовать температуре моделирования исходной траектории. Для `tpi` корректно реализована поправка притяжения (см. раздел 4.8). Все значимые величины записываются в файл, указанный опцией `mdrun -tpi`, а распределение энергий вставок — опцией `-tpid`. Файлы траектории и энергий не создаются. Распараллеленный `tpi` дает те же результаты, что и обычный. Для заряженных молекул более эффективным и точным является

использование PME с густой сетью, т.к. потенциал в системе нужно вычислить только однажды за шаг.

tric:

Тест на введение частиц в заранее указанную полость. Процедура аналогична предыдущей, за тем исключением, что из траектории дополнительно считывается координата места вставки. Желательно, чтобы вводимая молекула была центрирована в (0, 0, 0). GROMACS не делает это автоматически, т.к. в разных случаях может быть оптимальной разная центровка. Влияние параметра rtpi такое же, как и при trp. Поиск соседей проводится один раз за шаг, nstlist не используется. Распараллеленный tric дает те же результаты, что и обычный tinit: (0) [пс]

Начальное время в моделировании (имеет значение для интеграторов md, sd и bd).

dt: (0,001) [пс]

Шаг времени интегрирования (имеет значение для интеграторов md, sd и bd).

nsteps: (0)

Предельное число шагов интегрирования или минимизации энергии, -1 — неограниченно.

init_step: (0)

Начальный шаг. Время на шаге моделирования i вычисляется как

$$t = t_{\text{init}} + dt \cdot (\text{init_step} + i)$$

Параметр λ при моделировании свободных энергий вычисляется как

$$\lambda = \text{init_lambda} + \text{delta_lambda} \cdot (\text{init_step} + i)$$

Также неравновесные параметры моделирования могут зависеть от значения текущего шага. Поэтому для точного перезапуска или повторного проведения части моделирования может понадобиться установить init_step в номер шага, с которого делается перезапуск. tpbconv делает это автоматически.

comm_mode:

linear:

Удалять поступательное движение центра масс.

angular:

Удалять поступательное движение центра масс и вращение вокруг него.

no:

Не делать ни то, ни другое.

nstcomm: (10) [шагов]

Частота удаления движения центра масс.

comm_grps:

Группы, у которых удаляется движение центра масс (по умолчанию — у всей системы).

Ланжевенова динамика

bd_fric: (0) [а.е.м./пс]

Коэффициент трения для броуновской динамики. При bd_fric=0 он будет определен как масса/tau_t для каждой частицы.

ld_seed: (1993) (целое)

Используется для инициализации генератора случайных чисел для температурного шума и броуновской динамики. Если указан -1, то определяется из Process ID. При моделировании БД или СД на нескольких процессорах каждый процессор использует начальное значение, равное `ld_seed` плюс свой номер.

Минимизация энергии

`emtol`: (10,0) [кДж·моль⁻¹·нм⁻¹]

Минимизация прекращается, когда наибольшая из сил становится меньше этого значения.

`emstep`: (0,01) [нм]

Максимальный размер шага.

`nstcgsteep`: (1000) [шаги]

Частота осуществления шага скорейшего спуска при использовании метода сопряженных градиентов.

`nbfgscorr`: (10)

Число корректировочных шагов при использовании метода L-BFGS. Чем больше, тем (по крайней мере, теоретически) точнее, но и медленнее.

Управление выводом

`nstxout`: (100) [шаги]

Частота записи координат в файл траектории, последние координаты записываются всегда.

`nstvout`: (100) [шаги]

Частота записи скоростей в файл траектории, последние скорости записываются всегда.

`nstfout`: (0) [шаги]

Частота записи сил в файл траектории.

`nstlog`: (100) [шаги]

Частота записи энергий в файл протокола, последние энергии записываются всегда.

`nstcalcenergy`: (-1) [шаги]

Частота вычисления энергий, 0 — никогда. Эта опция имеет смысл только при динамике. С двойным радиусом обрезания она должна равняться или быть кратной `nstlist`. Эта опция влияет на производительность при распараллеленном моделировании, т.к. вычисление энергий требует глобальной передачи данных между всеми процессами, что может стать ограничивающим фактором при развитом распараллеливании. Значение по умолчанию -1 делает `nstcalcenergy` равным `nstlist` (если `nstlist`>0, иначе принимается значение 10). Если `nstenergy` оказывается меньше автоматически установленного значения, то используется наименьший общий знаменатель `nstenergy` и `nstlist`.

`nstenergy`: (100) [шаги]

Частота записи энергий в файл энергий, последние энергии записываются всегда. Должна быть кратна `nstcalcenergy`. Обратите внимание, что точные суммы и флюктуации за все МД шаги, деленные на `nstcalcenergy`, сохраняются в

файле энергии, поэтому команда ***g energy*** может вернуть точные средние энергии и флуктуации и при *nstenergy*>1.

nstxtcout: (0) [шаги]

Частота записи координат в файл ***xtc***.

xtc_precision: (1000) [вещественное]

Точность записи координат в файл ***xtc***.

xtc_grps

Группы, записываемые в файл ***xtc***. По умолчанию записывается вся система (если *nstxtcout* больше 0).

energygrps

Группы, записываемые в файл энергий.

Поиск соседей

nstlist: (10) [шаги]

>0

Значение задает частоту обновления списка соседей (и дальнодействующих сил при использовании двух радиусов обрезания). При минимизации энергии список соседей обновляется при каждом вычислении энергии.

0

Список соседей создается однажды и затем не обновляется. Это часто используется для моделирований в вакууме, когда все частицы взаимодействуют между собой.

-1

Автоматическое обновление. Может быть использовано только при сдвинутых, *switched* или пользовательских потенциалах, когда радиус их обрезания может быть меньше *rlist*. При этом получается буфер размером *rlist* минус наибольший радиус обрезания. Список соседей обновляется, только если одна или несколько частиц отдалились на более чем половину длины буфера от геометрического центра их зарядовой группы, определенного при предыдущем поиске соседей. Учитывается масштабирование координат при баростатировании или при опции *deform*. Это значение гарантирует отсутствие артефактов обрезания. Но при больших системах это может требовать значительных вычислительных ресурсов, т.к. обновление списка соседей будет постоянно вызываться одной-двумя частицами, движущимися вблизи границы на половине длины буфера.

ns_type:

grid

Система делится на ячейки, и при создании нового списка соседей на каждом *nstlist* шаге проверяются только атомы, находящиеся в соседних ячейках. В больших системах сеточный поиск выполняется намного быстрее простого поиска.

simple

При создании нового списка соседей проверяется каждый атом в системе.

pbc:

xyz

Использование периодических граничных условий (ПГУ) во всех направлениях.

no

Не использовать ПГУ. Для моделирования без обрезаний установите все радиусы обрезания и nstlist в 0. При этом для достижения лучшей производительности также установите ns_type=simple и используйте разбиение частиц, а не доменное разбиение.

xy

Использование ПГУ только в направлениях *x* и *y*. Это работает только при ns_type=grid и может использоваться совместно с walls. Без стенок или только с одной стенкой размер системы будет бесконечным в направлении *z*, поэтому не могут использоваться баростатирование и суммирование Эвальда. Эти проблемы не возникают при использовании двух стенок.

periodic_molecules:

no

Молекулы конечны, могут использоваться быстрые молекулярные ПГУ.

yes

Для систем с молекулами, которые привязаны к самим себе через ПГУ. Это требует более медленный алгоритм ПГУ и молекулы, которые не собраны как одно целое.

rlist: (1) [нм]

Радиус обрезания короткодействующих взаимодействий.

rlistlong: (1) [нм]

Радиус обрезания дальнодействующих взаимодействий. Этот параметр нужен только при использовании двух радиусов обрезания при switched потенциалах, когда необходим буфер для учета размеров зарядовых групп. Во всех других случаях он автоматически приравнивается самому длинному радиусу обрезания.

Электростатика

coulombtype:

Cut-off

Двухграничное обрезание с радиусом обрезания списка соседей rlist и кулоновским радиусом обрезания rcoulomb, где $rcoulomb \geq rlist$.

Ewald

Классическое суммирование Эвальда. Радиус обрезания rcoulomb должен равняться rlist. Наибольшая длина волновых векторов в обратном пространстве определяется параметром fourierspacing. Относительные погрешности в прямом и обратном пространствах определяются ewald_rtol.

Обратите внимание, что сложность алгоритма составляет $O(N^{3/2})$, поэтому в больших системах он работает очень медленно. Он оставлен главным образом для совместимости — в большинстве случаев PME работает намного лучше.

PME

Быстрый алгоритм сети частиц Эвальда (Particle-Mesh Ewald). Прямое пространство такое же, как в суммировании Эвальда, а обратное

обрабатывается с помощью быстрого преобразования Фурье (FFT). Размеры ячейки сети задаются *fourierspacing*, а порядок интерполяции — *pme_order*. С размером ячейки сети 0,1 нм и кубической интерполяцией точность вычисления кулоновских сил будет составлять $2\text{-}3 \cdot 10^{-4}$. Т.к. это меньше, чем погрешность от ван-дер-ваальсового обрезания, то можно попробовать 0,15 нм. При распараллеленном моделировании интерполяция распараллеливается лучше, чем FFT, поэтому при нем стоит попробовать уменьшить размеры ячейки и увеличить порядок интерполяции.

PPPM

Алгоритм Particle-Particle Particle-Mesh для дальнодействующей электростатики. Используйте, например, *rlist*=0.9, *rcoulomb*=0.9. Шаг сетки определяется параметром *fourierspacing*, разумным значением для PPPM будет 0,05–0,1 нм.

Обратите внимание: PPPM в текущей версии (4.5.4) не функционирует, но разработчики GROMACS собираются реализовать его в будущем с помощью небольшой модификации PME.

Reaction-Field

Реактивное поле с кулоновским радиусом обрезания *rcoulomb*, причем $rcoulomb \geq rlist$. Диэлектрическая проницаемость за пределами обрезания равняется *epsilon_rf*, ее значение 0 соответствует бесконечности.

Generalized-Reaction-Field

Обобщенное реактивное поле с кулоновским радиусом обрезания *rcoulomb*, причем $rcoulomb \geq rlist$. Диэлектрическая проницаемость за пределами обрезания равняется *epsilon_rf*. Ионная сила вычисляется из числа заряженных (т.е. с ненулевым зарядом) зарядовых групп. Температура для этого потенциала устанавливается через *ref_t [K]*.

Reaction-Field-zero

В GROMACS обычное реактивное поле приводит к плохому сохранению энергии. При Reaction-Field-zero эта проблема решена с помощью установки потенциала за пределами обрезания в нуль. Его можно использовать только с бесконечной диэлектрической проницаемостью (*epsilon_rf*=0), т.к. только тогда выполняется это условие. *rlist* должен быть на 0,1–0,3 нм больше, чем *rcoulomb*, чтобы учесть размеры зарядовых групп и диффузию в промежутке между обновлениями списка соседей.

Reaction-Field-nec

Подобно Reaction-Field, реализованному как в GROMACS до версии 3.3. Не применяется поправка к исключенным парам атомов и *self* парам. 1-4 взаимодействия вычисляются с помощью реактивного поля. Отсутствующая поправка на исключенные пары, которые не участвуют в 1-4 взаимодействии, составляет несколько процентов от полной электростатической энергии и дает незначительную разницу сил и давления.

Shift

Аналогично Shift для *vdwtype*. Желательно использовать *Reaction-Field-zero*, которое дает близкие значения потенциалов, но имеет физический смысл и

обеспечивает более точные значения энергий благодаря поправкам на исключения.

Encad-Shift

Кулоновский потенциал уменьшен, используется определение из пакета *Encad-Switch*.

Аналогично *Switch* для *vdwtype*. *Switch* кулоновского потенциала может привести к серьезным погрешностям, лучше использовать *Reaction-Field-zero*.

User

Значения потенциалов задаются пользователем. mdrun будет искать файл *table.xvg* со значениями потенциалов отталкивания, притяжения и кулоновского. Если присутствуют парные 1-4 взаимодействия, mdrun также будет искать файл *tablep.xvg* с их значениями потенциалов. Если для невалентных и парных взаимодействий нужно использовать одинаковые взаимодействия, то пользователь может указать для них один и тот же табличный файл. Эти файлы должны содержать семь колонок: x , $f(x)$, $-f'(x)$, $g(x)$, $-g'(x)$, $h(x)$, $-h'(x)$, где $f(x)$ — кулоновский потенциал, $g(x)$ и $h(x)$ — соответственно потенциалы притяжения и отталкивания. $g(x)$, $-g'(x)$, $h(x)$ и $-h'(x)$ игнорируются, когда *vdwtype* не установлен в *User*. Для невалентных взаимодействий x должен равномерно изменяться от нуля до наибольшего радиуса обрезания плюс *table-extension*. Дальность парных взаимодействий будет ограничена длиной таблицы. Оптимальный шаг x равняется 0,002 нм при одинарной точности и 0,0005 нм при двойной. Значение в нуле не важно. См. главу 6.7.2.

PME-Switch

Сочетание *PME* и *switch* функции в прямом пространстве (см. выше). *rcoulomb* может быть меньше *rlist*. Это обычно используется при *NVE* моделировании. При моделировании с постоянной температурой преимущество улучшенного сохранения энергии обычно перевешивается небольшой потерей точности электростатических взаимодействий.

PME-User

Сочетание *PME* и пользовательских таблиц (см. выше). *rcoulomb* может быть меньше *rlist*. mdrun вычитает вклад *PME* из табличных значений. Поэтому таблицы должны содержать порядка 10 знаков после запятой.

PME-User

Сочетание *PME-User* и *switching* функции (см. выше). *Switching* функция применяется к окончательным межчастичным взаимодействиям, т.е. и к заданной пользователем функции, и к поправке *PME*.

rcoulomb_switch: (0) [нм]

На каком расстоянии начинается модификация кулоновского потенциала.

rcoulomb: (1) [нм]

Кулоновский радиус обрезания.

epsilon_r: (1)

Диэлектрическая проницаемость. 0 обозначает бесконечную величину.

epsilon_rf: (1)

Диэлектрическая проницаемость для реактивного поля. Используется только с ним. 0 обозначает бесконечную величину.

Ван-дер-ваальсовы взаимодействия

vdwtype:

Cut-off

Двухграничное обрезание с радиусом обрезания списка соседей *rlist* и ван-дер-ваальсовым радиусом обрезания *rvdw*, где $rvdw \geq rlist$.

Shift

ЛД (но не бакингемов) потенциал уменьшен, и между *rvdw_switch* и *rvdw* силы плавно спадают до нуля. *rlist* должен быть на 0,1–0,3 нм больше, чем *rvdw*, чтобы учесть размеры зарядовых групп и диффузию в промежутке между обновлениями списка соседей.

Switch

ЛД (но не бакингемов) потенциал не модифицирован до *rvdw_switch*, после чего подгоняется для достижения нуля в *rvdw*. И потенциал, и сила — непрерывные и гладкие, но имейте ввиду, что все *switch* функции приводят к возрастанию силы (т.к. имеется перегиб потенциала и ускоряется его спадание). Радиус обрезания списка соседей *rlist* должен быть на 0,1–0,3 нм больше, чем *rvdw*, чтобы учесть размеры зарядовых групп и диффузию в промежутке между обновлениями списка соседей.

Encad-Shift

ЛД (но не бакингемов) потенциал уменьшен, используется определение из пакета *Encad*.

User

См. это значение для *coulombtype*. Значения функций в нуле не важны. Если вы хотите использовать ЛД поправку, то удостоверьтесь, что *rvdw* соответствует радиусу обрезания пользовательской функции [т.е. длине таблицы?]. Если *coulombtype* не равняется *User*, то значения $f(x)$ and $-f'(x)$ не учитываются.

rvdw_switch: (0) [нм]

На каком расстоянии начинается модификация ЛД потенциала

rvdw: (1) [нм]

Радиус обрезания ЛД или бакингемовых взаимодействий.

DispCorr:

no

Не применять никаких поправок.

EnerPres

Применить поправку на дальнодействующее притяжение для энергии и давления.

Ener

Применить поправку на дальнодействующее притяжение только для энергии.

Таблицы

table-extension: (1) [нм]

На сколько продолжаются таблицы неналентных потенциалов после наибольшего из их радиусов обрезания. Значение должно быть достаточно велико для учета размеров зарядовых групп и диффузии в промежутках между

обновлениями списка соседей. Если таблицы невалентных взаимодействий не используются, то такую длину должны иметь таблицы 1-4 взаимодействий (т.к. 1-4 взаимодействия всегда читаются из таблиц).

energygrp_table:

При использовании пользовательских кулоновских и/или ван-дер-ваальсовых потенциалов возможно указывать пары групп энергии, для которых будут использованы отдельные таблицы. Названия двух групп энергии будут через подчеркивание дописаны к имени табличного файла в порядке их перечисления в *energygrps*. Например, если *energygrps* = *Na Cl Sol* и *energygrp_table* = *Na Na Na Cl*, то *mdrun* будет дополнительно читать файлы *table_Na_Na.xvg* и *table_Na_Cl.xvg*. Со всеми остальными парами групп энергии будет использован файл *table.xvg*.

Суммирование по Эвальду

fourierspacing: (0,12) [нм]

Максимальный шаг FFT сети при использовании PPPM или PME. Для обычного суммирования Эвальда шаг сетки и размеры ячейки определяют наибольшую длину волнового вектора в каждом направлении (см. раздел 4.9.1). Во всех случаях каждое направление может быть переопределено указанием ненулевого значения соответствующему *fourier_n**. Для оптимизации относительной вычислительной нагрузки от межчастичных взаимодействий и сеточной части PME полезно знать, что точность вычисления электростатики остается примерно постоянной, когда кулоновский радиус обрезания и шаг сети изменяются в одно и то же число раз.

fourier_nx (0) ; *fourier_ny* (0) ; *fourier_nz*: (0)

Наибольшие размеры волновых векторов в обратном пространстве (при использовании суммирования Эвальда) или размеры сетки (при использовании PPPM или PME). Эти значения переопределяют *fourierspacing* в соответствующем направлении. Оптимальные значения — степени 2, 3, 5 и 7. Избегайте больших простых чисел.

pme_order (4)

Порядок интерполяции для PME. 4 соответствует кубической интерполяции. При распараллеленном моделировании может иметь смысл попробовать 6, 8 или 10 (и соответственно уменьшить размеры сети).

ewald_rtol (10–5)

Относительная интенсивность прямого сдвинутого эвальдовского потенциала в точке *rcoulomb*. Уменьшение этого значения даст более точную прямую сумму, но при этом понадобится больше волновых векторов для обратной суммы.

ewald_geometry: (3d)

3d

Суммирование Эвальда осуществляется во всех трех направлениях.

3dc

Обратная сумма вычисляется в 3D, но в направлении *z* применяются поправки к силе и потенциалу для осуществления псевдо-2D суммирования. Если система имеет форму слоя в плоскости *xy*, то можно попробовать увеличить размер

системы в направлении z (обычно достаточно, когда высота системы в 3 раза больше толщины слоя) и использовать эту опцию.

epsilon_surface: (0)

Управляет дипольной поправкой к суммированию Эвальда в 3D. По умолчанию она не применяется. Чтобы ее применить, установите ее равной диэлектрической проницаемости мнимой поверхности вокруг вашей бесконечной системы. Будьте внимательны — ее не следует использовать, если в системе есть подвижные заряды. Этот параметр не влияет на дальнодействующие поправки, осуществляемые при $ewald_geometry = 3dc$.

optimize_fft:

no

Не вычислять оптимальный FFT образ сети при запуске.

yes

Вычислять оптимальный FFT образ сети при запуске. Это немного экономит время при длительных моделированиях, но занимает несколько минут вначале.

Термостатирование

tcoupl:

no

Термостатирование не производится.

berendsen

Слабая привязка Беренсена к тепловому резервуару с температурой ref_t [К], с временной постоянной τ_{ref} [пс]. Несколько групп могут быть привязаны по-отдельности, они указываются через пробел в поле tc_grps .

nose-hoover

Термостатирование с помощью расширенного ансамбля Нозе-Гувера. Опорная температура и группы термостатирования указываются так же, как в предыдущем случае, но в этом случае τ_{ref} [пс] определяет период флюктуаций температуры при равновесии, он слегка отличается от времени релаксации в термостате Беренсена. В NVT моделях сохраненная величина энергии записывается в файлы энергии и протокола.

v-rescale

Термостатирование с помощью масштабирования скоростей со стохастическим членом (JCP 126, 014101). Этот термостат похож на термостат Беренсена с тем же τ_{ref} , но стохастический член обеспечивает генерацию правильного канонического ансамбля. Генератор случайных чисел инициализируется значением ld_seed . Термостат работает корректно и при $\tau_{ref}=0$. Для NVT моделей сохраненная величина энергии записывается в файлы энергии и протокола.

nsttcouple: (-1)

Частота осуществления термостатирования. Значение по умолчанию -1 делает *nsttcouple* равным *nstlist* (если *nstlist* ≤ 0 , иначе устанавливается 10). Для скоростной формы интегратора Верле *nsttcouple* устанавливается в 1.

nh-chain-length (10)

Количество сочлененных термостатов Нозе-Гувера для скоростной формы интегратора Верле. Интегратор leap-frog поддерживает только один термостат. Значения параметров этих термостатов не записываются в файл .edr (если не установлена переменная окружения GMX_NOSEHOOVER_CHAINS).

tc_grps:

Группы, которые будут термостатироваться по-отдельности.

tau_t: [пс]

Временные постоянные термостатов (по одной для каждой группы из *tc_grps*), -1 обозначает отсутствие термостатирования.

ref_t: [К]

Опорные температуры термостатов (по одной для каждой группы из *tc_grps*).

Баростатирование

pcoupl:

no

Баростатирование не производится, объем ячейки остается постоянным.

berendsen

Экспоненциальное затухание колебаний давления с временной постоянной *tau_p* [пс]. Система масштабируется на каждом шаге времени. Доказано, что полученный таким образом ансамбль не является термодинамически корректным, но это — самый эффективный способ масштабирования ячейки в начале моделирования.

Parrinello-Rahman

Баростатирование с использованием расширенного ансамбля. Векторы ячейки являются переменными в уравнениях движения атомов. Мгновенное масштабирование не производится. Как и для термостата Нозе-Гувера, временная постоянная *tau_p* [пс] является периодом флюктуаций давления при равновесии. Видимо, это — более подходящий метод, если нужно обеспечивать постоянство давления во время накопления данных, но возможны очень большие колебания при старте от другого давления. Это может быть неприемлемо в моделях, в которых важны точные колебания в NPT ансамбле или когда длительность баростатирования очень коротка, т.к. давление на предыдущем шаге времени в GROMACS используется для получения давления на текущем шаге.

MTK

Баростат Martyna-Tuckerman-Tobias-Klein. Он работает только с md-vv или md-vv-avek, очень близок к баростату Парринелло-Рамана. Как и для термостата Нозе-Гувера, временная постоянная *tau_p* [пс] является периодом флюктуаций давления при равновесии. Тоже возможны очень большие колебания при старте от другого давления. Сейчас реализовано только изотропное масштабирование.

pcoupltype:

isotropic

Изотропное баростатирование с временной постоянной *tau_p* [пс]. Сжимаемость и опорное давление задаются через *compressibility* [бар $^{-1}$] и *ref_p* [бар].

semiisotropic

Баростатирование, изотропное в направлениях x и y , но отличное в направлении z . Это может быть полезно для моделирования мембран. Требуется по два значения compressibility и ref_p: для направлений x/y и z соответственно.

anisotropic

Аналогично, но требуется шесть значений: направления xx , yy , zz , xy/yx , xz/zx и yz/zy соответственно. Если недиагональные сжимаемости равны нулю, то углы прямоугольной ячейки будут оставаться прямыми. Учтите, что анизотропное баростатирование может привести к значительной деформации ячейки.

surface-tension

Контроль поверхностного натяжения для поверхностей, параллельных плоскости xy . В направлении z используется обычное баростатирование. Первое значение ref_p — это опорное поверхностное натяжение, умноженное на количество поверхностей [бар·нм], второе значение — опорное давление в направлении z [бар]. Аналогично, первое значение в compressibility [бар-1] — сжимаемость в направлениях x и y , второе — z . Значение сжимаемости в направлении z должно быть достаточно точным, т.к. оно влияет на сходимость поверхностного натяжения. Если установить его в нуль, то высота системы будет оставаться постоянной.

nstpcouple: (-1)

Частота осуществления баростатирования. Значение по умолчанию -1 делает nstpcouple равным nstlist (если $nstlist \leq 0$, иначе устанавливается 10). Для скоростной формы интегратора Верле nstpcouple устанавливается в 1.

tau_p: (1) [пс]

Временная постоянная баростата.

compressibility: [бар-1]

Сжимаемость (обратите внимание: сейчас она действительно в бар-1). Для воды при 1 атм и 300 К сжимаемость равна $4,5 \cdot 10^{-5}$ бар-1.

ref_p: [бар]

Опорное давление.

refcoord_scaling:

no

Опорные координаты ограничений позиций (см. раздел 7.4) не будут изменяться. Учтите, что с этой опцией вириал и давление будут зависеть от абсолютных значений опорных координат.

all

Все опорные координаты будут масштабироваться согласно матрице баростатирования.

com

Центр масс опорных координат будет масштабироваться согласно матрице баростатирования. Векторы от каждой опорной координаты к их центру масс не будут изменяться. Используется только один центр масс, даже если есть несколько молекул с привязками позиций. При вычислении ЦМ опорных

координат в исходной конфигурации периодические граничные условия не учитывются.

Имитация отжига

В GROMACS имитация отжига может быть задана отдельно для каждой группы термостатирования. Опорная температура является кусочно-линейной функцией, возможно использовать любое число точек и выбирать режим (один проход или цикл) для каждой группы. Имитация отжига осуществляется динамическим изменением опорной температуры в используемом алгоритме термостатирования, поэтому температура системы обычно не сразу принимает опорное значения.

annealing:

Режим имитации отжига для каждой группы термостатирования.

no

Нет имитации отжига, делается лишь привязка к опорной температуре.

single

Последовательность температур проходит один раз. Если моделирование продолжается после последней точки, то температура останется привязанной к последнему значению.

periodic

После прохождения последней точки имитация отжига будет начата сначала.

annealing_points:

Список количеств опорных температур для каждой группы термостатирования.

Нуль в списке обозначает, что для данной группы имитация отжига не будет проводиться.

annealing_time: [пс]

Список моментов изменения опорной температуры для каждой группы термостатирования. Количество значений должно равняться сумме чисел в *annealing_points*.

annealing_temp:

Список проходимых температур для каждой группы термостатирования. Число значений должно равняться сумме чисел в *annealing_points*.

Пример: пусть есть две группы термостатирования с разными режимами отжига:

annealing = single periodic.

Одна группа будет проходить три температуры, другая — четыре:

annealing_points = 3 4

Моменты времени, в которые будет происходить смена опорной температуры:

annealing_time = 0 3 6 0 2 4 6

Наконец, сами опорные температуры:

annealing_temp = 298 280 270 298 320 320 298

Первая группа изначально (в 0 пс) будет находиться при 298 K. Опорная температура будет линейно падать до достижения 280 K в момент времени 3 пс, затем линейно падать до 270 K к 6 пс. После этого система будет оставаться при

270 К. Вторая группа изначально находится при 298 К, затем температура линейно растет до 320 К при 2 пс, где остается постоянной до момента 4 пс. Между 4 пс и 6 пс она уменьшается до 298 К, после чего цикл повторяется: температура возрастает до 320 К при 8 пс (=6+2), не меняется до 10 пс и т.д. grompp отображает эту информацию.

Инициализация скоростей

gen_vel:

no

Не инициализировать скорости. Они все будут установлены в нуль, если не указаны в исходном файле.

yes

grompp инициализирует скорости согласно распределению Максвелла-Больцмана при температуре *gen_temp* [К]. Генератор случайных чисел инициализируется значением *gen_seed* (имеет смысл только при интеграторе md).

gen_temp: (300) [К]

Температура, в соответствии с которой будут инициализированы скорости.

gen_seed: (173529) [целое]

Число для инициализации генератора случайных чисел для случайных скоростей. Если равно -1, то инициализация производится с помощью Process ID.

Связи и ограничения

constraints:

none

Ограничения на связи не будут накладываться (кроме тех, которые явно указаны в топологии), т.е. связи и углы представляются гармоническим или другим потенциалом.

hbonds

Ограничения накладываются только на связи с атомами водорода.

all-bonds

Ограничения накладываются на все связи.

h-angles

Ограничения накладываются на все связи и углы с участием атомов водорода.

all-angles

Ограничения накладываются на все связи и углы.

constraint_algorithm:

LINCS

(*LInear Constraint Solver*). Точность регулируется параметром *lincs_order*, который задает порядок разложения матрицы ограничений. После поправки с использованием обратной матрицы алгоритм итеративно исправляет удлинение, вызванное поворотом связи. Количество таких итераций задается параметром *lincs_iter*. Относительное среднеквадратичное отклонение при ограничении записывается в файл протокола каждые *nstlog* шагов. Если связь за один шаг поворачивается более чем на *lincs_warnangle* градусов, то будет

выводиться предупреждение на stderr (обычно это экран) и в файл протокола. *LINCS* не следует использовать со сдвоенными угловыми ограничениями.

SHAKE

SHAKE несколько медленнее и нестабильнее, чем *LINCS*, но работает с угловыми ограничениями. Относительная погрешность задается через параметр *shake_tol*, для «обычного» моделирования достаточно 0,0001. *SHAKE* не поддерживает ограничения между атомами в разных доменах, поэтому он не может быть использован с доменным разбиением, если есть ограничения, лежащие между зарядовыми группами. Также *SHAKE* нельзя применять для минимизации энергии.

continuation:

Эта опция ранее называлась *unconstrained_start*.

no

Применить ограничения к начальной конфигурации и сбросить оболочки поляризуемых моделей.

yes

Не применять ограничения к начальной конфигурации и не сбрасывать оболочки. Нужно для точного продолжения моделирования и повторных запусков.

shake_tol: (0,0001)

Относительная погрешность *SHAKE*.

lincs_order: (4)

Наибольший порядок разложения матрицы связанных ограничений. Когда ограничения формируют треугольники, для них сверх обычного разложения делается дополнительное разложение того же порядка. Для «обычных» моделерований как правило достаточно 4, 6 нужно для больших шагов времени с виртуальными позициями или броуновской динамикой. Для точной минимизации энергии может понадобиться 8 и больше. С доменным разбиением размер ячейки ограничен длиной, охваченной *lincs_order*+1 ограничениями с углами между ними 120°. Если нужно масштабировать ячейку сверх этого предела, то можно уменьшить *lincs_order* и увеличить *lincs_iter*, т.к. точность не ухудшается при постоянном значении $(lincs_iter+1) \cdot lincs_order$.

lincs_iter: (1)

Число итераций для исправления вращательного удлинения в *LINCS*. Для обычных моделерований достаточно одной, но для *NVE* моделерований, где нужно точно сохранять энергию, или для точной минимизации энергии следует делать две итерации.

lincs_warnangle: (30) [градусы]

Максимальный угол поворота связи, при котором не выдается предупреждение.

morse:

no

Связи представляются указанными в топологии потенциалами.

yes

Все связи представляются потенциалом Морзе.

Исключения групп энергии

energygrp_excl:

Пары групп энергии, для которых все невалентные взаимодействия будут исключены. Например, если есть две группы энергии *Protein* и *SOL*, то запись *energygrp_excl = Protein Protein SOL SOL*

приведет к тому, что будут вычисляться невалентные взаимодействия только между белком и растворителем. Это особенно полезно для ускорения расчетов энергии с *mdrun -rerun* и для исключения взаимодействий внутри замороженных групп.

Стенки

nwall: 0

Если параметр равен 1, то будет одна стенка на высоте $z = 0$; если равен 2, то также будет стенка на высоте $z = \text{высота ячейки}$. Стенки можно использовать только с *pbc=xy*. При значении 2 можно использовать баростатирование и суммирование Эвальда (обычно лучше использовать полуизотропное баростатирование с сжимаемостями по x и y равными 0, т.к. иначе площадь поверхности будет меняться). Взаимодействие стенок с остальной системой задается через *wall_atomtype*. Автоматически добавляются группы энергии *wall0* и *wall1* (для *nwall=2*) для отслеживания взаимодействий групп энергии со стенкой (стенками). Движение центра масс в направлении z не будет удаляться.

wall_atomtype:

Названия атомных типов в силовом поле, параметры которых будут определять взаимодействие стенок с системой. Задание в файле топологии специальных атомных типов для стенок со своими правилами комбинирования позволяет независимо настраивать взаимодействие других атомных типов со стенками.

wall_type:

- а) 9-3 Потенциал Леннард-Джонса, интегрированный по объему стенки (потенциал 9-3).
- б) 10-4 ЛД потенциал, интегрированный по поверхности стенки (потенциал 10-4).
- в) 12-6 Обычный ЛД потенциал.

table

Зависимости потенциалов от расстояния от стенки заданы пользователем. Таблицычитываются из файлов таким же образом, как в опции *energygrp_table*, при этом первое имя файла — для «обычной» группы энергии, а второе — *wall0* или *wall1*. Используются только колонки с притяжением и отталкиванием.

wall_r_linpot: -1 (нм)

Сверх этого расстояния от стенки потенциал становится линейным, а сила — постоянной. Если некоторые атомы находятся далеко от стенки, то при проведении уравновешивания его следует сделать положительным, т.к. если он ≤ 0 (или < 0 для *wall_type=table*), то будет выдана критическая ошибка.

wall_density: [нм⁻³ или нм⁻²]

Количество атомов на единицу объема (для стенки типа 9-3) или поверхности (типа 10-4) для каждой стенки.

wall_ewald_zfac: 3

Множитель для третьего вектора ячейки (только для суммирования Эвальда), наименьшее значение 2. Суммирование Эвальда может использоваться только с *nwall*=2, при этом необходимо, чтобы *ewald_geometry*=3dc. Создающийся таким образом пустой слой в ячейке служит для уменьшения нефизических кулоновских взаимодействий между периодическими образами.

Натяжение центра масс

pull:

no

Не использовать натяжение центра масс (НЦМ). Все связанные с ним опции будут пропущены, и при их наличии в файле *.mdp* будет выдаваться предупреждение.

umbrella

НЦМ с помощью зонтичного потенциала между опорной группой и одной или несколькими другими группами.

constraint

Идентично *umbrella* за тем исключением, что будет использоваться ограничение вместо гармонического потенциала.

constant_force

НЦМ с помощью линейного потенциала и, соответственно, постоянной по величине силы. При таком способе нет опорной позиции, поэтому параметры *pull_init* и *pull_rate* не используются.

pull_geometry:

distance

Натяжение вдоль вектора между двумя группами. Используемые компоненты этого вектора можно выбрать через *pull_dim*.

direction

Натяжение в направлении *pull_vec*.

direction_periodic

Аналогично *direction*, но позволяет расстоянию превышать половину размера ячейки. При этом геометрия ячейки должна быть статична в направлении натяжения (например, нельзя использовать баростатирование в этом направлении), и сила натяжения не добавляется к вириалу.

cylinder

Разработано для натяжения по отношению к слою, когда опорный центр масс является локальным цилиндрическим участком опорной группы. Натяжение происходит в направлении *pull_vec*. Из опорной группы выделяется цилиндрический участок вдоль оси, проходящей через группу натяжения в направлении *pull_vec*, с двумя радиусами. Внутри внутреннего радиуса *pull_r1* все относительные веса равны 1, между *pull_r1* и *pull_r0* веса линейно уменьшаются до нуля. Также учитываются массы атомов. Оба радиуса должны быть меньше, чем половина размера ячейки. Для повернутых цилиндров они

должны быть существенно меньше половины размера ячейки, т.к. расстояние от атома в опорной группе до центра масс опорной группы имеет одновременно радиальную и аксиальную компоненты.

position

Натяжение к позиции опорной группы плюс *pull_init* плюс время·*pull_rate*·*pull_vec*.

pull_dim: (*Y Y Y*)

Компоненты вектора расстояния, которые будут использоваться (если *geometry*=*distance* или *position*). Также эти компоненты будут выведены в файл протокола. *Y* — компонент будет использоваться, *N* — нет.

pull_r1: (1) [нм]

Внутренний радиус цилиндра при *geometry*=*cylinder*.

pull_r0: (1) [нм]

Внешний радиус цилиндра при *geometry*=*cylinder*.

pull_constr_tol: (10–6)

Относительная погрешность ограничений для натяжения с их использованием.

pull_start:

no

Не изменять *pull_init*.

yes

Прибавлять расстояние до центра масс в начальной конфигурации к *pull_init*.

pull_nstxout: (10)

Частота записи центров масс групп натяжения.

pull_nstfout: (1)

Частота записи сил, действующих на группы натяжения.

pull_ngroups: (1)

Количество групп натяжения без учета опорной группы. Если здесь указана только одна группа, то опорная группа и группа натяжения обрабатываются одинаково (если геометрия не цилиндрическая). Ниже приведены параметры натяжения для опорной группы (окончание 0) и первой группы натяжения (окончание 1). Параметры остальных групп натяжения задаются аналогично.

pull_group0:

Имя опорной группы. Если не указано, то используется начало координат (0, 0, 0). При такой абсолютной опорной позиции система становится неинвариантной по отношению к перемещениям, и возникает вопрос о том, что делать с движением центра масс.

pull_weights0:

См. *pull_weights1*.

pull_pbcatom0: (0)

См. *pull_pbcatom1*.

pull_group1:

Имя группы натяжения.

pull_weights1:

Опциональные относительные веса атомов. На них умножаются массы атомов при вычислении центра масс. Если указан нуль, то все веса равны 1. Иначе нужно указать вес для каждого атома этой группы натяжения.

pull_pbcatom1: (0)

Опорный атом для обработки периодических граничных условий внутри группы (не влияет на ПГУ между группами). Этот параметр важен, только когда диаметр группы натяжения больше, чем половина самого короткого вектора ячейки. При определении центра масс будут использоваться те образы атомов группы, которые расположены ближе всего к *pull_pbcatom1*. Если он равен 0, то используется средний по порядку в топологии атом. Этот параметр не используется при цилиндрической геометрии. Значение -1 включит косинусное взвешивание, которое полезно для групп молекул в периодической системе, например, в слое воды (см. Enginet al. J. Chem. Phys. B 2010).

pull_vec1: (0,0 0,0 0,0)

Направление натяжения. grompp нормализует вектор.

pull_init1: (0,0) или (0,0 0,0 0,0) [нм]

Опорная позиция в момент $t = 0$. Скаляр, но при *geometry=position* — вектор.

pull_rate1: (0) [нм/пс]

Скорость изменения опорной позиции.

pull_k1: (0) [кДж·моль⁻¹·нм⁻² или кДж·моль⁻¹·нм⁻¹]

Силовая постоянная. Для зонтичного натяжения — это гармоническая силовая постоянная в $\text{кДж}\cdot\text{моль}^{-1}\cdot\text{нм}^{-2}$. Для натяжения постоянной силой — это силовая постоянная

линейного потенциала, которая равна отрицательной (!) постоянной силе в $\text{кДж}\cdot\text{моль}^{-1}\cdot\text{нм}^{-1}$.

pull_kb1: (pull_k1) [кДж·моль⁻¹·нм⁻² или кДж·моль⁻¹·нм⁻¹]

То же, что и *pull_k1*, но для состояния Б. Используется только при вычислениях свободной энергии (когда включено *free_energy*). Силовая постоянная в таком случае вычисляется как $(1 - \lambda) \cdot \text{pull_k1} + \lambda \cdot \text{pull_kb1}$.

Ограничения

disre:

no

Нет ограничений расстояний (указанные в топологии игнорируются).

simple

Простые ограничения расстояний по молекулам, усреднение по ансамблю может быть осуществлено с помощью *mdrun -multi*, где переменная окружения *GMX_DISRE_ENSEMBLE_SIZE* устанавливает количество систем в каждом ансамбле (обычно равно значению *mdrun -multi*).

ensemble

Ограничения расстояний по ансамблю молекул в одной моделируемой ячейке. Следует использовать только в особых случаях, вроде димеров (эта функция не работает в текущей версии GROMACS).

disre_weighting:

conservative

Закрепляющие силы вычисляются как производные от закрепляющего потенциала.

equal

Ко всем парам атомов в ограничении прилагается одинаковая закрепляющая сила.

disre_mixed:

no

При вычислении закрепляющей силы используется усредненное по времени значение расстояния (см. раздел 4.3.4).

yes

При вычислении закрепляющей силы используется среднее геометрическое усредненного по времени значения расстояния и мгновенного его значения.

disre_fc: (1000) [кДж·моль⁻¹·нм⁻²]

Силовая постоянная ограничений расстояний. Она умножается на отдельный коэффициент для каждого ограничения.

disre_tau: (0) [пс]

Время затухания для скользящего среднего ограничений позиций.

nstdisreout: (100) [шаги]

Частота записи в файл энергий скользящего среднего и мгновенного расстояния для закрепленных пар атомов. Файл энергии может стать очень большим!

orire:

no

Не использовать ограничения ориентаций (данные в файле топологии игнорируются).

yes

Использовать ограничения ориентаций, усреднение по ансамблю возможно осуществить через mdrun –multi.

orire_fc: (0) [кДж·моль]

Силовая постоянная ограничений ориентаций. Она умножается на отдельный коэффициент для каждого ограничения. Если равна нулю, то ограничения не действуют.

orire_tau: (0) [пс]

Время затухания для скользящего среднего ограничений ориентаций.

orire_fitgrp:

Опорная группа для ограничений ориентаций. Для белка хороший выбор — это белковый скелет, *backbone*.

nstorireout: (100) [шаги]

Частота записи скользящего среднего и мгновенных ориентаций всех ограничений и order тензора молекулы в файл энергии (он может стать очень большим!).

Неявный растворитель

implicit_solvent:

no

Не использовать неявный растворитель.

GBSA

Провести моделирование с неявным растворителем с использованием обобщенного формализма Борна. Доступны три метода вычисления борновских радиусов: *Still*, *HCT* и *OBC*, они указываются в поле *gb_algorithm*. Способ неполярной сольватации задается параметром *sa_algorithm*.

gb_algorithm:

Still

Использовать метод *Still* для вычисления борновских радиусов.

HCT

Использовать для этого метод Hawkins-Cramer-Truhlar.

OBC

Использовать для этого метод Onufriev-Bashford-Case.

nstgbradii: (1) [шаги]

Частота вычисления борновских радиусов. В большинстве случаев значения больше 1 приводят к нарушению сохранения энергии и нестабильным траекториям.

rgbradii: (1,0) [нм]

Радиус обрезания при вычислении борновских радиусов. На данный момент должен равняться rlist

gb_epsilon_solvent: (80)

Диэлектрическая проницаемость неявного растворителя

gb_saltconc: (0) [моль/л]

Концентрация солей для модели неявного растворителя, сейчас не используется.

gb_abc_alpha (1); *gb_abc_beta* (0,8); *gb_abc_gamma* (4,85);

Коэффициенты масштабирования для метода ОВС. Значения по умолчанию соответствуют *OBC(II)*, значения для *OBC(I)* равны 0,8; 0 и 2,91 соответственно.

gb_dielectric_offset: (0,009) [нм]

Величина диэлектрического сдвига при вычислении борновских радиусов. Это — расстояние между центром каждого атома и центром его поляризационной энергии.

sa_algorithm

Ace-approximation

Использовать *Ace-type* приближение (по умолчанию).

None

Не вычислять неполярную сольватацию. Для *GBSA* будет вычисляться только полярная часть.

sa_surface_tension: [кДж·моль⁻¹·нм⁻²]

Значение по умолчанию поверхностного натяжения для SA алгоритмов. По умолчанию равно -1; если оставить так, то команда *grompp* установит значение, специфичное для выбранного алгоритма вычисления радиусов ($0,0049 \text{ ккал}\cdot\text{моль}^{-1}\cdot\text{A}^{-2}$ для *Still*, $0,0054 \text{ ккал}\cdot\text{моль}^{-1}\cdot\text{A}^{-2}$ для *HCT* и *OBC*).

Значение 0 при *sa_algorithm*, отличном от None, приведет к тому, что неполярные вычисления проводиться не будут.

5.3 Группы частиц для анализа (команды *make_ndx*, *mk_angndx*)

В этой части описаны способы выборки групп атомов для анализа полученной траектории. Подробная информация о входных и выходных параметрах программ есть в онлайн-руководстве на www.gromacs.org. Выходные файлы часто получаются в виде файлов *.xvg* (для графических пакетов Grace/Xmgr). Большинство команд анализа траектории требует указания групп, для которых эта обработка должна быть проведена. В некоторых из них требуется несколько стандартных групп, но всегда можно использовать индексный файл (*.ndx*).

Группы могут состоять из ряда номеров атомов, но в некоторых случаях также и из номеров молекул. Также возможно указывать серии углов тройками атомных номеров, двугранных углов — четверками, связей или векторов (в молекуле) — парами номеров атомов. Для упрощения создания индексных файлов (имя файла по умолчанию *index.ndx*) есть несколько программ, которые их генерируют с помощью топологии или конфигурации. Чтобы сгенерировать индексный файл, состоящий из ряда номеров атомов используйте команды *make_ndx* или *g_select*. Чтобы сгенерировать индексный файл с углами или двугранными углами, используйте команду *mk_angndx*. Также вы можете создать их вручную. Общий формат индексных файлов:

```
[ Oxygen ]  
1 4 7  
[ Hydrogen ]  
2 3 5 6  
8 9
```

Сначала указывается имя группы в квадратных скобках. Последующие номера атомов могут быть размещены на любом количестве строк. Нумерация атомов начинается с 1.

Стандартные группы

Если командам анализа или команде *grompp* не был указан индексный файл, то предлагаются несколько стандартных групп:

System — все атомы системы

Protein — все атомы белков

Protein-H — все атомы белков, за исключением атомов водорода

C-alpha — атомы C_{α}

Backbone — атомы белкового скелета: N , C_{α} и C

MainChain — атомы главной цепи белка: N , C_{ω} , C и O , а также атомы O C -конца

MainChain+Cb — атомы главной цепи белка плюс C_{β}

MainChain+H — атомы главной цепи белка плюс амидные атомы H и атомы H N -конца

SideChain — все атомы белка, за исключением N , C_{α} , C и O , амидных атомов H , атомов O C -конца и атомов H N -конца

SideChain-H — то же, но без атомов водорода

Prot-Masses — атомы белков без dummy масс (используемых при построении виртуальных позиций в NH_3 группе и боковой цепи триптофана); эта группа создается, только если она отличается от группы *Protein*.

Non-Protein — все небелковые атомы

DNA — все атомы ДНК

RNA — все атомы РНК

Water — молекулы воды (молекулы с именами вроде SOL, WAT, HOH и т.п.).

См. файл **residuetypes.dat** для полного списка их названий.

non-Water — все атомы, за исключением Water

Ion — все, отмеченное как Ion в residuetypes.dat

Water_and_Ions — объединение групп Water и Ions

molecule_name — для всех остатков/молекул, которые не распознаны как белки, ДНК или РНК. Генерируется по одной группе на остаток/молекулу.

Other — все атомы, не принадлежащие к белкам, ДНК или РНК.

Пустые группы не будут предлагаться. Большинство групп содержит только атомы белков. Атом рассматривается как принадлежащий белку, если имя остатка с ним отмечено в файле **residuetypes.dat** как *Protein*. ДНК, РНК и т.д. определяются так же. Если вам нужно модифицировать эту классификацию, то вы можете скопировать файл **residuetypes.dat** в свою рабочую папку и редактировать эту локальную копию.

Выборка (команда *g_select*)

Команда *g_select*

GROMACS также содержит инструмент *g_select*, который может выбирать атомы, основываясь на более гибких критериях, чем *make_ndx*, например, по их координатам. На данный момент это экспериментальный инструмент, который поддерживает только некоторые основные операции, но в будущем эту функциональность предполагается ввести и в другие программы анализа. Описание возможных способов выделения атомов можно прочитать, запустив *g_select* и набрав *help* в появившемся приглашении. Также есть возможность написать свои собственные инструменты анализа, чтобы воспользоваться преимуществом гибкости этих выборок: как образец см. файл *template.c* в папке *share/gromacs/template*.

Приложение 1. Основные команды Linux.

Команда	Описание	Употребление	Основные опции
ls	Посмотреть список файлов	ls [option] [file] примеры: ls (или ls *) ls *.dat	-a : показывать .file служебные файлы (с .) -l : показывать полное описание файлов
passwd	Изменить пароль	passwd	
exit	Выйти из Linux	exit	

Команды связанные с директориями

Команда	Описание	Употребление	Основные опции
mkdir	Создать директорий	mkdir [option] dir пример: mkdir dir1	-p : не давать ошибку, если директорий есть
cd	Перейти в другой директорий	cd [dir] (пример: cd ../dir1)	
rmdir	Удалить пустой директорий	rm dir пример: rm dir1	

Команды связанные с файлами

Команда	Описание	Употребление	Основные опции
cp	Копировать файлы или директории	cp [option] file1 file2 пример: cp file1 file2 cp file1 ../file2	-r : копировать вложенные директории -i : подтверждать
mv	Перемещать файлы или директории	mv file1 file2 пример: mv file1 ../file2	
cat	Подцеплять файл к концу другого файла	cat [option] file1 file2 file3 пример: cat file1 file2 file3	
rm	Удалять файлы или директории	rm [option] file пример: rm file1 rm *.dat rm -ri * rm -rf * (очень опасная команда)	-f : без подтверждения -i : подтверждать -r : удалить вложенные директории
less	Посмотреть содержание файла	less file	
file	Определить тип файла	file file	
chmod	Изменить статус файла (например, его исполняемость)	chmod [option] mode file пример: chmod u+x file	-R : change files and directories recursively

Приложение 2. Основные команды GROMACS (версия 4.5)

Генерирование топологий и координат

Команда **pdb2gmx** преобразует файлы *pdb* в файлы топологии (*.top*) и координат (*.gro*)

Команда **g_x2top** создает топологию, исходя из координат

Команда **editconf** редактирует ячейку и записывает подгруппы

Команда **genbox** заполняет растворителем ячейку моделирования

Команда **genion** добавляет одноатомные ионы или соль в энергетически выгодные позиции

Команда **genconf** увеличивает размеры системы

Команда **genrestr** создает ограничения (restraints) позиций или расстояний

Команда **g_protonate** протонирует молекулы

Моделирование

Команда **grompp** собирает входной файл моделирования *.tpr* из файлов *.gro .top* и *.mdp*

Команда **tpbconv** создает входной файл для перезапуска после завершившегося ошибкой моделирования

Команда **mdrun** осуществляет минимизацию энергии, моделирование и анализ нормальных мод

Просмотр траекторий

Команда **ngmx** визуализирует траекторию, позволяя “прокрутить” на мониторе видеозапись, состоящую из дискретного набора кадров (фреймов)

Команда **g_nmtraj** генерирует виртуальную траекторию из собственного вектора

Обработка энергий

Команда **g_energy** записывает энергию в файлы *.xvg* и вычисляет их средние значения

Команда **g_enemats** выделяет матрицу энергий из файла энергии

Команда **mdrun** с ключом -rerun заново вычисляет энергию во фреймах траектории

Конвертация файлов

Команда **editconf** создает и редактирует файлы структуры (координат)

Команда **trjconv** создает и редактирует файлы траекторий

Команда **trjcat** объединяет файлы траекторий

Команда **eneconv** преобразует файлы энергий

Команда **xpm2ps** преобразует рисунки XPM (XPixMap) в Encapsulated Postscript

Команда **g_sigeps** преобразует ЛД пары параметров *C6, C12* в σ и ε и наоборот

Инструменты

Команда **make_ndx** создает индексные файлы

Команда **mk_angndx** создает индексные файлы для *g_angle*

Команда **gmxcheck** проверяет и сравнивает файлы

Команда **gmxdump** переводит двоичные файлы в текстовый формат

Команда **g traj** рисует графики r , v и F выбранных атомов или групп из траектории

Команда **g analyze** анализирует наборы данных

Команда **trjorder** сортирует молекулы в соответствии с величиной их расстояния до группы атомов

Команда **g filter** применяет частотные фильтры к траекториям (для достижения более плавных смещений атомов при переходе от кадра к кадру при демонстрации “видеозаписи” командой **ngmx**)

Команда **g lie** оценка свободной энергии из линейных комбинаций

Команда **g dyndom** интерполяция и экстраполяция структурных вращений

Команда **g morph** линейная интерполяция конформаций

Команда **g wham** анализ взвешенной гистограммы после зонтичной выборки

Команда **xpm2ps** преобразует рисунки *XPM* (*XPixelMap*) в *Encapsulated Postscript*

Команда **g sham** читает и записывает наборы данных для графических файлов *xmgr* и *xvgr*

Команда **g spatial** вычисляет пространственные функции распределения

Команда **g select** выделение групп атомов на основе гибких критериев

Команда **g pme error** оценка погрешности от использования PME в данном входном файле

Команда **g tune pme** оценивает длительность моделирования, исходя из параметров PME. Полезно для их оптимизации

Расстояния между структурами

Команда **g rms** вычисляет среднеквадратичные отклонения от опорной структуры и матрицы среднеквадратичных отклонений

Команда **g confrms** сопоставляет две структуры и вычисляет среднеквадратичное отклонение

Команда **g cluster** кластерные структуры

Команда **g rmsf** вычисляет колебания атомов

Расстояния между структурами в динамике

Команда **g mindist** вычисляет минимальное расстояние между двумя группами

Команда **g dist** вычисляет расстояния между центрами масс двух групп

Команда **g bond** вычисляет расстояния между атомами

Команда **g mdmat** составляет карты контактов остатков

Команда **g polystat** вычисляет статические свойства полимеров

Команда **g rmsdist** вычисляет средние значения расстояний между парами атомов в -2, -3 или -6 степенях

Распределение масс в динамике

Команда **g traj** рисует графики r , v , F , температуры и вращательной энергии

Команда **g gyrate** вычисляет радиус инерции

Команда **g msd** вычисляет среднеквадратичные отклонения

Команда **g polystat** вычисляет статические свойства полимеров

Команда **g rotacf** вычисляет вращательные функции корреляции молекул

Команда **g rdf** строит функции радиального распределения

Команда **g rotmat** вычисляет матрицу поворота для подгонки к опорной структуре

Команда **g vanhove** строит функции корреляции ван Хофа

Анализ валентных взаимодействий

Команда **g bond** вычисляет распределение длин связей

Команда **mk angndx** создает индексные файлы для ***g_angle***

Команда **g angle** вычисляет распределение и корреляцию углов и двугранных углов

Команда **g dih** анализирует изменения двугранных углов

Структурные свойства

Команда **g hbond** анализирует и рассчитывает параметры водородных связей

Команда **g saltbr** рассчитывает солевые мости

Команда **g sas** вычисляет площадь поверхности, доступную для растворителя

Команда **g order** вычисляет параметры упорядочения атомов в углеродных хвостах

Команда **g principal** вычисляет главные оси инерции групп атомов

Команда **g rdf** строит функции радиального распределения

Команда **g sgangle** вычисляет углы и расстояния между двумя группами

Команда **g sorient** анализирует расположение растворителя вокруг растворенных веществ

Команда **g spol** анализирует ориентацию и поляризацию диполей растворителя вокруг растворенных веществ

Команда **g bundle** анализирует пучки осей, например, трансмембранные оси

Команда **g disre** анализирует ограничения расстояний

Команда **g clustsize** вычисляет распределение размеров кластеров атомов

Команда **g anadock** кластерные структуры из результатов моделирований в Autodock

Кинетические свойства

Команда **g traj** рисует графики ***r***, ***v*** и ***F*** температуры и вращательной энергии

Команда **g velacc** вычисляет автокорреляционные функции скорости

Команда **g tcaf** вычисляет вязкость жидкости

Команда **g bar** вычисляет разницу свободных с помощью Bennett's acceptance ratio

Команда **g current** вычисляет автокорреляционные функции тока

Команда **g vanhove** вычисляет функции корреляции ван Хофа

Команда **g principal** вычисляет главные оси инерции групп атомов

Электростатические свойства

Команда **genion** добавляет одноатомные ионы в энергетически выгодные позиции

Команда **g potential** вычисляет электростатический потенциал поперек ячейки

Команда **g dipoles** вычисляет общий дипольный момент и его флуктуации

Команда **g dielectric** вычисляет диэлектрическую проницаемость как функцию частоты

Команда **g_current** вычисляет диэлектрические проницаемости заряженных систем

Команда **g_spol** анализирует ориентацию и поляризацию диполей растворителя вокруг растворенных веществ

Анализ белков

Команда **do_dssp** определяет вторичную структуру и вычисляет площадь поверхности, доступную для растворителя

Команда **g_chi** вычисляет угол χ и другие двугранные углы

Команда **g_helix** вычисляет основные свойства альфа-спиралей

Команда **g_helixorient** вычисляет локальные углы, изгибы, повороты и ориентации в спиралях

Команда **g_rama** строит диаграммы Рамачандрана

Команда **g_xrama** отображает анимированные диаграммы Рамачандрана

Команда **g_wheel** строит helical wheels

Поверхности раздела

Команда **g_potential** вычисляет электростатический потенциал поперек ячейки

Команда **g_density** вычисляет плотность системы

Команда **g_densmap** вычисляет 2D плоские или аксиально-радиальные матрицы плотности

Команда **g_order** вычисляет параметры упорядочения атомов в углеродных хвостах

Команда **g_h2order** определяет ориентации молекул воды

Команда **g_bundle** анализирует пучки осей, например, трансмембранные оси

Команда **g_membed** внедряет белок в жировой двойной слой

Ковариационный анализ

Команда **g_covar** вычисляет и приводит к диагональному виду матрицу ковариации

Команда **g_anaeig** анализирует собственные вектора

Команда **make_edi** создает входные файлы для essential dynamics sampling

Нормальные моды колебаний

Команда **grompp** создает входной файл моделирования

Команда **mdrun** находит минимум потенциальной энергии

Команда **g_nmeig** приводит гессиан к диагональному виду

Команда **g_nmtraj** создает колебательную траекторию для нормальных мод

Команда **g_anaeig** анализирует нормальные моды

Команда **g_nmens** создает ансамбль структур из нормальных мод.

Приложение 3. Графическая программа Grace

GRACE это графическая программа для построения двумерных графиков научных данных. Он работает под различными операционными системами (Unix, VMS, OS / 2, и Windows (95/98 / NT / XP/7). Его возможности близки к возможностям таких программ с графическим интерфейсом, как SigmaPlot, Microcal Origin, Gnuplot и Genplot. Ее сила заключается в том, что он сочетает в себе удобство графического интерфейса пользователя с мощностью скриптового языка, который позволяет делать сложные вычисления или выполнять автоматизированные задачи.

Программа GRACE разработана на основе программы XMGR, изначально написанная Полом Тернером. От номера версии 4.00, развитие перешло к новой команде разработчиков. Получить свежую информацию о программе GRACE и скачать ее последнюю версию можно найти на сайте <http://plasma-gate.weizmann.ac.il/Grace/>

Для быстрого начала работы с GRACE, можно использовать демонстрации "Help / Examples" (Помощь/Примеры) в меню GRACE. Это обычные проекты Grace, так что пользователь может играть с ними и изменять их после установки пакета GRACE.

Для работы с GRACE надо после установки пакета:

1. Запустить версию XMGRACE.
2. Загрузить данные с помощью опции меню *Data/Import/ASCII*: Загрузить как "SingleData" (т.е как набор данных для одного графика) две колонки ASCII данных, или как "Block Data" (т.е. как набор данных для нескольких графиков) несколько столбцов ASCII данных.
3. Запустить проект "Explorer" с помощью "Edit / Explorer" (Редактировать/Проект) в меню.
4. Нажать на верхний уровень дерева проекта и выбрать "Page size" ("Размер страницы").
5. Отрегулировать масштабы, метки и деления на осях координат X и Y в "Axis" ("Оси").
6. Отрегулировать линии, символы и подписи на осях координат в "SET" ("Установки").
7. Продолжить исследовать более тонкие настройки меню.
8. В GRACE можно манипулировать данными в "Data" / *Transformations* ("Данные/ Преобразования"). Например, чтобы сдвинуть график на 20 единиц масштаба вправо, в "Evaluate Expression" ("Вычислить выражение") можно выбрать тот же набор слева и справа, и сказать, "Formula" (Формула): $y = y - 20$ Как вы, вероятно, заметили, GRACE может сделать гораздо больше, чем эти элементарные преобразования.
9. Когда график будет готов, можно выбрать "File/Save" ("Файл/ Сохранить").
10. Выбрать "File/Print Setup" ("Файл /Настройка Печати") настроить параметры, а затем выбрать в меню "Файл/ Печать", чтобы получить бумажную копию графика.

Приложение 4.

Определение функций для описания внутримолекулярных взаимодействий.

interaction type	directive	# at.	f. tp	parameters	F. E.
bond	bonds(excl,con)	2	1	b_0 (nm); k_b ($\text{kJ mol}^{-1}\text{nm}^{-2}$)	all
G96 bond	bonds(excl,con)	2	2	b_0 (nm); k_b ($\text{kJ mol}^{-1}\text{nm}^{-4}$)	all
morse	bonds(excl,con)	2	3	b_0 (nm); D (kJ mol^{-1}); β (nm^{-1})	
cubic bond	bonds(excl,con)	2	4	b_0 (nm); $C_{i=2,3}$ ($\text{kJ mol}^{-1}\text{nm}^{-4}$);	
connection	bonds(excl)	2	5		
harmonic pot.	bonds	2	6	b_0 (nm); k_b ($\text{kJ mol}^{-1}\text{nm}^{-2}$)	all
FENE bond	bonds(excl)	2	7	b_m (nm); k_b ($\text{kJ mol}^{-1}\text{nm}^{-2}$)	
tab. bond	bonds(excl)	2	8	table number (≥ 0); k (kJ mol^{-1})	k
tab. bond n.c.	bonds	2	9	table number (≥ 0); k (kJ mol^{-1})	k
restraint pot.	bonds	2	10	low, up ₁ , up ₂ (nm); k_{dp} ($\text{kJ mol}^{-1}\text{nm}^{-2}$)	all
LJ/Coul. 1-4	pairs	2	1	$V^{(\sigma)}$; $W^{(\sigma)}$	all
LJ/Coul. 1-4	pairs	2	2	fudge QQ(); q_i , q_j (e), $V^{(\sigma)}$; $W^{(\sigma)}$	
LJ/C. pair NB	pairs_nb	2	1	q_i , q_j (e); $V^{(\sigma)}$; $W^{(\sigma)}$	
angle	angles(con)	3	1	θ_0 (deg); k_θ ($\text{kJ mol}^{-1}\text{rad}^{-2}$)	all
G96 angle	angles(con)	3	2	θ_0 (deg); k_θ (kJ mol^{-1})	all
cross bond-bond	angles	3	3	r_{1e} , r_{2e} (nm); k_{pp} ($\text{kJ mol}^{-1}\text{nm}^{-2}$)	
cross bond-angle	angles	3	4	r_{1e} , r_{2e} r_{3e} (nm); k_{pa} ($\text{kJ mol}^{-1}\text{nm}^{-2}$)	
Urey-Bradley	angles(con)	3	5	θ_0 (deg); k_θ (kJ mol^{-1}); r_{13} (nm); k_{UB} (kJ mol^{-1})	
quartic angle	angles(con)	3	6	θ_0 (deg); $C_{i=0,1,2,3,4}$ ($\text{kJ mol}^{-1}\text{rad}^{-4}$)	
tab. angle	angles	3	8	table number (≥ 0); k (kJ mol^{-1})	k
proper dih.	dihedrals	4	1	ϕ_s (deg); k_ϕ (kJ mol^{-1}); multiplicity	ϕ, k
improper dih.	dihedrals	4	2	ξ_0 (deg); k_ξ ($\text{kJ mol}^{-1}\text{rad}^{-2}$)	all
RB dihedral	dihedrals	4	3	$C_0, C_1, C_2, C_3, C_4, C_5$ (kJ mol^{-1})	all
periodic improper dih.	dihedrals	4	4	ϕ_s (deg); k_ϕ (kJ mol^{-1}); multiplicity	ϕ, k
Fourier dih.	dihedrals	4	5	C_1, C_2, C_3, C_4 (kJ mol^{-1})	all
tab. dihedral	dihedrals	4	8	table number (≥ 0); k (kJ mol^{-1})	k
proper dih. multi	dihedrals	4	9	ϕ_s (deg); k_ϕ (kJ mol^{-1}); multiplicity	ϕ, k
exclusions	exclusions	1		one or more atom indices	

'# at' is the number of atom indices

'f. tp' is function type

'F. E.' indicates which parameters can be interpolated during free energy calculations

(σ) the combination rule determines the type of LJ parameters, see 5.3.2

(excl) used by grompp for generating exclusions

(con) can be converted to constraints by grompp

For free energy calculations, all or no parameters for topology 'B' ($\lambda = 1$) should be added on the same line, after the normal parameters, in the same order as the normal parameters.

Приложение 5.

Внутримолекулярная геометрия и ограничения (constraints and restraints).

interaction type	directive	# at.	f. tp	parameters	F. E.
constraint	constraints ^(excl)	2	1	b_0 (nm)	all
constr. n.c.	constraints	2	2	b_0 (nm)	all
settle	settles	1	1	d_{OH} , d_{HH} (nm)	
vsite2	virtual_sites2	3	1	$a()$	
vsite3	virtual_sites3	4	1	$a(), b()$	
vsite3fd	virtual_sites3	4	2	$a()$; d (nm)	
vsite3fad	virtual_sites3	4	3	θ (deg); d (nm)	
vsite3fout	virtual_sites3	4	4	$a(), b()$; c (nm ⁻¹)	
vsite4fdn	virtual_sites4	5	2	$a(), b()$; c (nm);	
vsite COG	virtual_sitesn	1	1	one or more construc. atom ind.	
vsite COM	virtual_sitesn	1	2	one or more construc. atom ind.	
vsite COW	virtual_sitesn	1	3	one or more pairs consisting of a construc. atom ind. and weight	
position res.	position_restraints	1	1	k_x, k_y, k_z (kJ mol ⁻¹ nm ⁻²)	all
restr. pot.	bonds	2	10	low, up ₁ , up ₂ (nm); k_{dr} (kJ mol ⁻¹ nm ⁻²)	all
distance res.	distance_restraints	2	1	type; label; low, up ₁ , up ₂ (nm); weight ()	
dihedral res.	dihedral_restraints	4	1	type; label; ϕ_0 (deg); $\Delta\phi$ (deg); weight ()	
orient. res.	orientation_restraints	2	1	exp.; label; α ; c (U nm ²); obs. (U); weight (U ⁻¹)	
angle res.	angle_restraints	4	1	θ_0 (deg); k_e (kJ mol ⁻¹); multiplicity	θ, k
angle res. z	angle_restraints_z	2	1	θ_0 (deg); k_e (kJ mol ⁻¹); multiplicity	θ, k

'# at' is the number of atom indices

'f. tp' is function type

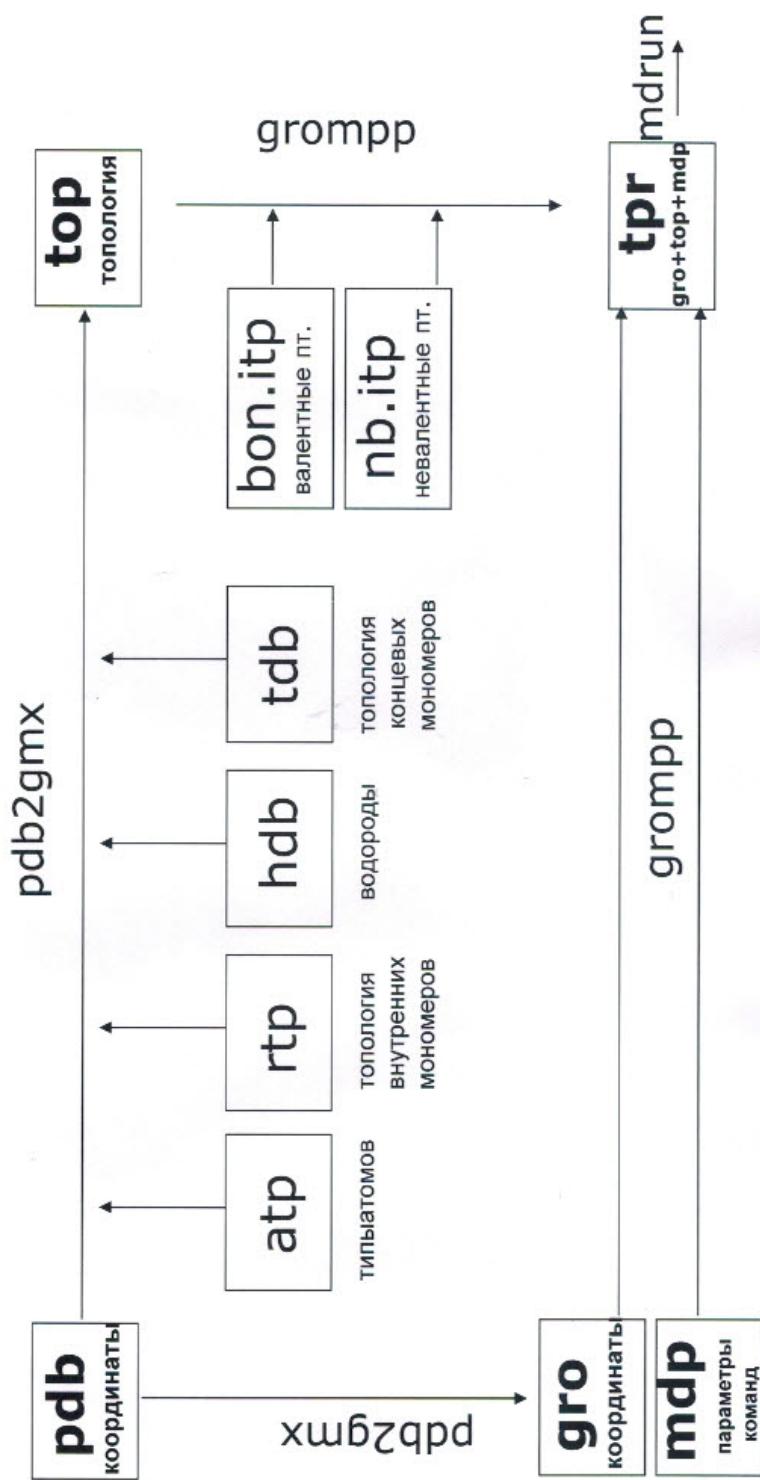
'F. E.' indicates which parameters can be interpolated during free energy calculations

^(excl) used by grompp for generating exclusions

For free energy calculations, all or no parameters for topology 'B' ($\lambda = 1$) should be added on the same line, after the normal parameters, in the same order as the normal parameters.

Приложение 6. Получение TPR файла из PDB файла

Получение “tpr” файла для MD расчета



Литература

1. M.P.Allen, Computer simulation of liquids, Clarendon Press, Oxford, 1987
2. Френкель Д., Смит В. Принципы компьютерного моделирования молекулярных систем: от алгоритмов к приложениям / Пер. с англ. и научн. ред. Иванов В.А., Стукан М.Р. – М.:Научный мир, 2013 – 578с.:105ил. (Фундаментальные основы нанотехнологий: лучшие зарубежные учебники).
3. B. J. Alder and T. E. Wainwright, in Molecular Dynamics by Electronic Computers, In I.Prigogine, Editor, Proc. International Symposium on Transport Processes in Statistical Mechanics(Brussel, 1956), Wiley Interscience, New York, 1957, p. 97-131.
4. L.Verlet Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules // Phys.Rev., 159, 98-103, 1967.
5. Rahman A., Stillinger F.H. Mollecular dynamics study of temperature effects on water structure and kinetics// J. Chem. Phys. **57**, 1281-1292, 1972.
6. Балабаев Н.К., Гриццов А.Г., Шноль Э.Э. // Численное моделирование движения молекул, Препринт ИПМ, Москва, 1972, 38с.
7. Неелов И.М. Уравнения движения и времена релаксации цепной макромолекулы, Дипломная работа, ЛГУ, Ленинград, 1974,
8. Балабаев Н.К., Гриццов А.Г., Шноль Э.Э. Численное моделирование линейной полимерной цепочки. //Докл. АН СССР. 1975. Т. 229. с. 1096-1098
9. Балабаев Н.К., Готлиб Ю.Я., Даринский А.А. Неелов И.М. Моделирование движения мономера в жидкости. Препринт Научного центра биологических исследований, Научно-вычислительный центр АН СССР, Пущино, 1976
- 10.Ryckaert J.P., Ciccotti G, Berendsen H.J.C Numerical integration of Cartesian equations of motion of a systems with constraints-molecular dynamics of n-alkanes, J.Comput.Phys., 23, p.327-341, 1977
- 11.Балабаев Н.К., Готлиб Ю.Я., Даринский А.А. Неелов И.М. // Высокомолек. соед., А20, с. 2194, 1978
- 12.Gotlieb Yu.Ya., Balabaev N.K., Darinskii A.A, Neelov I.M. Investigation of local motions in polymers by the method of molecular dynamics // Macromolecules, 1980, N 3, p.602-608
- 13.B.Hess, D.van der Spoel, E.Lindahl et al GROMACS (Groningen Machine for Chemical Simulations), Version 4.5.4, User manual, 1991-2000, 2001-2010.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ТОПЛИВНО-ЭНЕРГЕТИЧЕСКОГО КОМПЛЕКСА

Кафедра химии входила в состав первых 14 кафедр ЛИТМО, сформированных в 1930 году. В 1930–1960 годах кафедра работала в рамках факультета Точной механики; в период деятельности Инженерно-физического факультета (ИФФ) с 1946 года по 1954 год кафедра входила в состав ИФФ. С 1933 года – кафедру возглавлял известный специалист в области оптического стекла профессор В.Г. Воано, позже – известный русский ученый-химик профессор С.А. Щукарев. С 1954 по 1972 год кафедру возглавлял доцент Г.С. Кошурников.

С момента второго рождения инженерно-физического факультета в 1976 г. кафедра химии вошла в его состав. В это время на кафедре стали развиваться, в основном, три научно-технологических направления: создание новых композиционных оптических материалов; разработка химических сенсоров; технология оптического волокна.

В последующие годы сотрудники кафедры, прежде всего, профессора Новиков А.Ф. и Успенская М.В., существенно переработали методику преподавания курса химии, адаптировав ее к активно внедрявшейся тогда в Университете системе дистанционного обучения. В результате, преподавание курса химии в Университете ИТМО вышло на новый более высокий уровень.

В дальнейшем на кафедре под руководством профессора М.В. Успенской активно развивалось научно-техническое направление в области химии и физики сорбирующих полимерных материалов и нанокомпозитов. В частности, на основе акриловых супервлагоабсорбентов разработан ряд новых материалов многофункционального назначения: сенсоры, жидкие линзы, раневые повязки, искусственные почвы для сельского хозяйства, огнестойкие конструкционные элементы и др.

В связи с этим в 2011 году данная кафедра (исторически – кафедра химии) позиционировала себя как отдельное структурное подразделение Национального исследовательского университета ИТМО в качестве кафедры “Информационных технологий топливно-энергетического комплекса”.

С переходом отечественных предприятий на международные стандарты продукции, повышением требований к охране окружающей среды и внедрением сложных аналитических автоматизированных систем контроля качества и мониторинга, с 2008 года в рамках направления «Техническая физика» кафедра проводит подготовку магистров и бакалавров по профилю «Физико-технические аспекты аналитического приборостроения».

Подготовка включает в себя следующие разделы:

- Компьютерные комплексы для автоматизированного контроля физических, химических, механических, термических, реологических и некоторых других свойств нефтяного сырья и продуктов нефтепереработки;
- Встроенные микропроцессорные комплексы для управления технологическими процессами и измерением широкого круга параметров энергетических установок и систем энергоснабжения;
- Физико-математическое моделирование технологических процессов нефтепереработки и топливно-энергетического комплекса;
- Информационно-аналитические системы и комплексы различного профиля, адаптированные под специфические условия работы на предприятиях ТЭК.

Уникальная программа обучения сочетает фундаментальную подготовку в области информационных систем, физической оптики, молекулярной спектроскопии, аналитической и физической химии, компьютерной метрологии, общехимической технологии и автоматики.

В рамках специальных дисциплин изучаются приборы и методы контроля качества продукции и принципы построения автоматизированных анализаторных систем для предприятий ТЭК, нефтяной и химической промышленности.

Такие системы как основа информационных технологий контроля качества и мониторинга безопасности могут успешно применяться практически на всех предприятиях и лабораториях химического и нефтехимического профиля, а также в металлургической, пищевой и фармацевтической промышленности.

Выпускники кафедры имеют широкие перспективы трудоустройства в современных крупных компаниях ТЭК, таких как Роснефть, ПТК, Газпром, Киришинефтегорснинтез, Лукойл, ТНК-ВР, а также на предприятиях и лабораториях пищевой, фармацевтической и других отраслях промышленности.

Практика эксплуатации предприятий ТЭК подтверждает необходимость создания и применения эффективных систем контроля за безопасностью и систем экологического мониторинга.

В связи с этим с 2011 года были разработаны и открыты бакалаврская и магистерская программы по направлению подготовки 241000 "Энерго- и ресурсосберегающие процессы в химической технологии, нефтехимии и биотехнологии". Основной целью образовательной магистерской программы "Информационные ресурсосберегающие технологии и экологические аспекты на предприятиях ТЭК" является подготовка высококвалифицированных специалистов, соответствующих современным требованиям к выпускникам вуза, с учетом потребностей рынка труда Санкт-Петербурга и регионов России. Будущие магистры будут способны использовать информационные технологии и математическое моделирование для описания различных физических и физико-химических процессов, для контроля качества продукции нефтепереработки, работать на современном оборудовании в научных, научно-производственных и производственных лабораториях по исследованию выпускаемой продукции и т.д.

Основными направлениями научной деятельности в рамках магистерской программы являются:

- Создание приборов и датчиков физических величин и физико-химических параметров углеводородного сырья и продуктов (в том числе на основе нанотехнологий);
- Разработка приборов для измерения параметров качества нефтепродуктов и пищевых продуктов на основе компьютерных технологий;
- Создание эффективных информационных систем контроля качества продукции и коммерческого учета на предприятиях ТЭК на основе приборов и устройств различного назначения;
- Создание эффективных информационных систем мониторинга безопасности эксплуатации объектов ТЭК.

Подготовка магистров ведется с участием ряда промышленных предприятий, научно-производственных объединений, научно-исследовательских институтов и вузов Санкт-Петербурга, что дает возможность получить отличные знания и неоценимый опыт в различных сферах деятельности: производственной, научно-исследовательской, административной и т.д.

Биотехнология и биоинженерия являются приоритетными направлениями современной науки и промышленного производства. Продукты биотехнологии и биоинженерии востребованы в медицине, фармации, биологии, и других высокотехнологичных отраслях народного хозяйства. Разработка новых источников энергии, создание биосовместимых материалов и синтез биологически активных веществ – главные составляющие этих двух наук и отраслей производства. В частности, интенсивно развиваются производство и применение ферментов в переработке различных видов сырья и в получении биопрепаратов. Ферментные технологии имеют преимущества с экономической, технологической и экологической точек зрения, поэтому годовой оборот ферментных препаратов составляет десятки миллионов

долларов США и он непрерывно растёт. По объёму производства ферментные препараты занимают третье место после аминокислот и антибиотиков. Ферментативные процессы, применяемые в технологиях, аналогичны природным, но они более безопасны и для здоровья человека и для окружающей среды.

Развитие этих отраслей сдерживается недостатком специалистов высшего уровня, подготовленных в области информационного обеспечения и средств измерения живых систем и биологических структур.

Для решения проблемы подготовки магистров на стыке информационных технологий, биологии и инженерии объединены усилия двух кафедр: Кафедра химии и молекулярной биологии ИХиБТ и кафедра ИТТЭК, имеющих опыт подготовки специалистов бакалавров и магистров в информационных технологиях и биотехнологии.

В учебный план предлагаемой программы включены, наряду с общеобразовательными, дисциплины по информационной, биологической, химической, технологической подготовке и ряду других отраслей знаний, необходимых в подготовке специалистов заявленного уровня.

В настоящее время на каф. ИТТЭК под руководством проф. Успенской М.В., ведутся работы по направлениям, связанных с созданием материалов для фармакологии и регенеративной медицины, предметов санитарно-гигиенического назначения, а также биосовместимых и биодеградируемых материалов.

Также на кафедре под руководством проф. Неелова И.М. активно развивается моделирование полимеров и биополимеров, начиная от структуры веществ и физико-химических процессов, протекающих в живых организмах до физико-механических и эксплуатационных характеристик материалов и биосистем.

Профессорско-преподавательский состав на кафедре насчитывает 18 человек, из них 6 профессоров и докторов наук.

В настоящее время на базе кафедр НИУ ИТМО создан Международный научно-исследовательский институт биоинженерии, возглавляемый проф. М.В. Успенской, что значительно расширяет экспериментальную базу и научный потенциал кафедр и способствует повышению уровня подготовки кадров высшей категории.

В настоящее время на кафедре трудятся 18 преподавателей, шестеро из них являются докторами наук, профессорами, признанными на международном уровне, членами ученых советов в России и за рубежом.

Оглавление

Аннотация.....	3
Введение	3
Раздел 1. Молекулярное моделирование и метод молекулярной динамики для одноатомных газов.....	5
1.1 Молекулярная динамика. Уравнения движения Ньютона и их дискретизация	6
1.2 Стохастическая динамика	7
1.3 Броуновская динамика.....	8
1.4 Потенциалы взаимодействия для одноатомных молекул.....	8
1.5 Периодические граничные условия	10
1.6 Статистические ансамбли и термостатирование	10
1.7 Задание начальных координат молекул.....	13
1.8 Общий алгоритм МД расчета для одноатомных газов.....	14
1.9 Общая схема моделирования	14
Раздел 2. Молекулярное моделирование полимерных молекул.....	15
2.1 Механические модели линейных полимерных цепей	15
2.2 Задание начальных конформаций сложных молекул.....	17
2.3 Минимизация энергии системы.....	18
2.4 Алгоритм МД расчетов для сложных молекул	19
2.5 Обработка результатов	20
Раздел 3. GROMACS. Силовые поля и ограничения движения.....	21
3.1 Условные обозначения и единицы измерения в GROMACS	22
3.2 Силовые поля и методы их учета в GROMACS	24
3.3 Ограничения движения	32
Раздел 4 Топологии в GROMACS.....	33
4.1 Введение	33
4.2 Типы частиц.....	34
4.3 Файлы параметров	36
4.4 Входные файлы для команды pdb2gmx	40
4.5.1 Файл топологии (.top)	50
4.5.2 Файл топологии для отдельной молекулы (.itp)	56
4.5.3 Файл координат (.gro).....	58
Раздел 5 Программы и параметры моделирования.....	60
5.1 Типы файлов GROMACS	60
5.2 Параметры и файл управления запуском (.mdp).....	61
5.3 Группы частиц для анализа (команды make_ndx, mk_angndx)	84
Приложение 1. Основные команды Linux.....	86
Приложение 2. Основные команды GROMACS (версия 4.5).....	87
Приложение 3. Графическая программа Grace	91
Приложение 4	92
Приложение 5	93
Приложение 6. Получение TPR файла из PDB файла.....	94
Литература	95
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ТОПЛИВНО-ЭНЕРГЕТИЧЕСКОГО КОМПЛЕКСА	96

Неелов И.М.

**Введение в молекулярное моделирование
биополимеров**
Учебное пособие

авторской редакции

Редакционно-издательский отдел НИУ ИТМО

Зав. РИО

Н.Ф. Гусарова

Лицензия ИД № 00408 от 05.11.99

Подписано к печати

Заказ № 3179

Тираж 100

Отпечатано на ризографе

Редакционно-издательский отдел
Санкт-Петербургского национального
исследовательского университета
информационных технологий, механики и оптики



197101, Санкт-Петербург, Кронверкский пр., 49