

Саратовский Государственный Университет имени Н.Г. Чернышевского
Физический факультет
Кафедра радиотехники и электродинамики

О.Е. Глухова, Г.В. Савостьянов

**Вычислительный практикум по моделированию
наноструктур в программном комплексе Kvazar**

Учебно-методическое пособие
для студентов по направлениям подготовки
"Радиофизика" и "Конструирование и технология электронных средств"

2015

УДК 519-688, 51-72, 538.9

ББК В22

Глухова О.Е., Савостьянов Г.В.

Вычислительный практикум по моделированию наноструктур в программном пакете "Kvazar": Учебно-методическое пособие для студентов по направлениям подготовки "Радиофизика" и "Конструирование и технология электронных средств". Саратов, 2015. 44 с. : ил.

В учебно-методическом пособии изложено содержание лабораторных работ по изучению механических и электронных свойств наноструктур, также дается описание методов теоретического исследования, таких как метод сильной связи, метод молекулярной динамики и др. В пособии приводится подробное описание программного пакета молекулярного моделирования "Kvazar" (вычислительные алгоритмы, особенности работы, способы организации расчетов), с помощью которого выполняются лабораторные работы.

Рецензент:

доктор физико-математических наук, профессор,
декан физического факультета СГУ, Аникин В. М.

Содержание

1 Теоретические основы	
программного пакета Kvazar	4
1.1 Метод молекулярной динамики	4
1.2 Метод сильной связи	6
1.3 Силовое поле REBO	8
1.4 Силовое поле AIREBO	11
2 Структура и особенности работы	
программного пакета Kvazar	13
2.1 Установка пакета	13
2.2 Вычислительный модуль	14
2.2.1 Структура вычислительного модуля	14
2.2.2 Классы вычислительного модуля	15
2.2.3 Организация расчетов	21
2.3 Графический модуль	26
2.3.1 Модуль препроцессинга	27
2.3.2 Модуль постпроцессинга	32
3 Лабораторные работы	34
3.1 Лабораторная работа №1	34
3.2 Лабораторная работа №2	36
3.3 Лабораторная работа №3	39
3.4 Лабораторная работа №4	40

Предисловие

Программный пакет Kvazar – многофункциональное программное средство для молекулярного моделирования. В программном пакете реализованы разнообразные методы молекулярного моделирования, такие как метод сильной связи, метод молекулярной динамики, силовые поля REBO, AIREBO. С помощью программного пакета можно изучать механические (в том числе прочностные), эмиссионные и электронно-энергетические свойства наноструктур. Программный пакет реализует возможность проектирования геометрии исследуемых объектов, задания периодических граничных и внешних условий (температура, давление, статическая или динамическая механическая нагрузка и др.). Kvazar предназначен для работы под ОС Linux и может быть использован как на персональных компьютерах, так и на суперкомпьютерах кластерного типа, поддерживающих интерфейс передачи данных MPI.

1 Теоретические основы программного пакета Kvazar

1.1 Метод молекулярной динамики

Метод молекулярной динамики позволяет моделировать движение атомов молекулярной системы. В его основе лежит численный расчет траекторий движения отдельных атомов системы.

В данном методе молекулярная система представляется как механическая система, состоящая из конечного числа материальных точек, обладающих некоторой массой. Положениям материальных точек механической системы соответствуют положения атомов молекулярной системы. II-й закон Ньютона для системы взаимодействующих частиц записывается в следующем виде:

$$m_i \frac{d^2 \vec{r}_i(t)}{dt^2} = \vec{F}_i(\vec{r}), \quad i = \overline{1, n}, \quad (1)$$

где \vec{F}_i - суммарная сила, действующая на i -ый атом со стороны остальных частиц.

Начальные условия задаются в виде

$$\begin{aligned}\vec{r}_i(t_0) &= \vec{r}_i^0, \\ \vec{v}_i(t_0) &= \vec{v}_i^0, \quad \vec{v}_i = \frac{d\vec{r}_i}{dt}, \quad i = \overline{1, n}.\end{aligned}\tag{2}$$

и представляют собой стартовую геометрию молекулярной системы и совокупность начальных скоростей атомов молекулярной системы.

Таким образом, для расчета динамики молекулярной системы необходимо задать функцию потенциальной энергии $U(r)$ и решить краевую задачу (1)-(2).

Существует множество способов численного интегрирования уравнений движения (1). Специфика метода молекулярной динамики состоит в необходимости интегрирования очень большого числа уравнений, что налагает определенные требования на экономию компьютерной памяти. Кроме этого, при расчетах основное время уходит на вычисление силы, действующей на данный атом, т.е. правой части уравнения (1). Связано это с существенной нелинейностью силы взаимодействия и необходимостью суммирования большого числа слагаемых (прежде всего – сил взаимодействия с соседними атомами). Указанное обстоятельство снижает эффективность методов, требующих на каждом шаге многократного вычисления правой части уравнений (1). С этим, в частности, связано редкое применение метода Рунге-Кутта [1] в молекулярной динамике.

Одним из наиболее простых и распространенных методов является алгоритм Верле [2], согласно которому вычисление положения частицы осуществляется по ее предыдущим двум положениям. Расчетная формула метода Верле:

$$\vec{r}(t + \Delta t) = 2\vec{r}(t) - \vec{r}(t - \Delta t) + \vec{a}(t)\Delta t^2 + O(\Delta t^4),\tag{3}$$

где Δt - шаг интегрирования, $\vec{a}(t)$ - ускорение. Данная схема не требует вычисления скоростей.

Метод центральных разностей, который впервые предложил применять в молекулярной динамике Виньярд [3], определяется уравнениями:

$$\vec{v}(t + \frac{1}{2}\Delta t) = \vec{v}(t - \frac{1}{2}\Delta t) + \vec{a}(t)\Delta t + O(\Delta t^4), \quad (4)$$

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t + \frac{1}{2}\Delta t)\Delta t + O(\Delta t^4). \quad (5)$$

Данный метод эквивалентен методу Верле в том смысле, что исключение скоростей из (4),(5) приводит к формуле (3). В отличие от алгоритма Верле, в методе Виньярда на каждом шаге определяются и координаты и скорости частиц. При этом скорости частиц определяются для промежуточных моментов времени. Существует ряд модификаций метода Верле и центральных разностей, однако большинство из них при формально различной схеме интегрирования дает в точности одинаковые траектории частиц.

В программном пакете Kvazar реализованы методы численного интегрирования Верле и Виньярда.

1.2 Метод сильной связи

Основой метода сильной связи является решение стационарного уравнения Шредингера [6]. В данном методе волновая функция электрона, находящегося на молекулярной орбитали описывается в виде линейной комбинации волновых функций [7] орбиталей изолированных атомов:

$$\psi(r) = \sum_{i\alpha} c_{i\alpha} \phi_{i\alpha}(r), \quad (6)$$

где индекс $i\alpha$ соответствует α орбитали i -го атома системы.

В уравнении (6) коэффициенты $c_{i\alpha}$ являются неизвестными. Поскольку совокупность волновых функций орбиталей изолированных атомов не образует полного базиса волновых функций, таким путем нельзя получить точное решение уравнения Шредингера. Однако, путем соответствующего подбора базисных волновых функций, можно получить результаты, очень хорошо согласующиеся с экспериментальными данными [7]. Как известно [7], коэффициенты $c_{i\alpha}$, соответствующие наилучшему решению в данном классе волновых функций, могут быть получены

путем решения так называемого секулярного уравнения:

$$\sum_{j\beta} (H_{i\alpha j\beta} - \epsilon S_{i\alpha j\beta}) c_{j\beta} = 0, \quad (7)$$

где

$$H_{i\alpha j\beta} = \langle \phi_{i\alpha} | H | \phi_{j\beta} \rangle = \int \phi_{i\alpha}^*(r) H \phi_{j\beta}(r) dr, \quad (8)$$

$$S_{i\alpha j\beta} = \langle \phi_{i\alpha} | \phi_{j\beta} \rangle = \int \phi_{i\alpha}^*(r) \phi_{j\beta}(r) dr. \quad (9)$$

Следует отметить, что на практике интегралы обычно заменяются приближенными выражениями.

В результате решения секулярного уравнения определяются коэффициенты $c_{i\alpha}$ и значения энергии молекулярных орбиталей. Энергия занятых электронных состояний соответствует энергии связи и определяется по формуле

$$E_{bond} = \sum_{\gamma} n_{\gamma} \epsilon_{\gamma}. \quad (10)$$

Это выражение является суммой энергий занятых электронами молекулярных орбиталей, n - занятые электронами орбитали, и ϵ_n — собственная энергия электрона, находящегося на соответствующей молекулярной орбитали.

Поскольку полученная таким образом энергия связи не учитывает энергию отталкивания, возникающую при сближении электронных облаков, в рамках метода сильной связи полная энергия системы представляется суммой двух слагаемых

$$E_{tot} = E_{bond} + E_{rep} \quad (11)$$

где E_{rep} – энергия отталкивания, описываемая специальным подобранным эмпирическим потенциалом.

В Kvazar реализован метод сильной связи, основанный на параметризации Гудвина [8]. Элементы матрицы гамильтониана вычисляются следующим образом:

$$H_{i\alpha j\beta}(r) = V_{i\alpha j\beta}(r_0) \left(\frac{r_0}{r} \right)^{n_a} \exp \left(-n_b \left(\frac{r}{r_c} \right)^{n_c} + n_b \left(\frac{r_0}{r_c} \right)^{n_c} \right), \quad (12)$$

Здесь $V_{i\alpha j\beta}$ - значение энергии взаимодействия α орбитали i -го атома с β орби-

талью j -го атома находящихся в равновесном положении. α и β могут соответствовать s, p_x, p_y и p_z орбиталям. Величина $V_{i\alpha j\beta}$ вычисляется с помощью преобразований Слейтера-Костера [9], на основании значений направляющих косинусов вектора, соединяющего атомы i и j , а также энергий взаимодействия орбиталей α и β в σ и π ориентациях по отношению друг к другу. В данной параметризации отсутствует явная форма волновых функций орбиталей изолированных атомов, и базисный набор считается ортонормированным, то есть

$$S_{i\alpha j\beta} = \delta_{i\alpha j\beta}, \quad (13)$$

поэтому решение секулярного уравнения сводится к процессу диагонализации матрицы гамильтониана.

Энергия отталкивания между парой атомов, записывается в виде

$$E_{core}(r) = E_{core}(r_0) \left(\frac{r_0}{r} \right)^{m_a} \exp \left(-m_b \left(\frac{r}{r_c} \right)^{m_c} + m_b \left(\frac{r_0}{r_c} \right)^{m_c} \right), \quad (14)$$

соответственно, полная энергия отталкивания определяется в результате суммирования по всем парам атомов:

$$E_{rep} = \sum_{i < j} E_{core}(r_{ij}). \quad (15)$$

1.3 Силовое поле REBO

Силовое поле REBO относится к классу методов порядка связи и основывается на потенциале Терсоффа [10]. Первый вариант потенциала REBO был предложен Д. Бреннером [11], позднее был значительно усовершенствован и получил название REBO II поколения [12]. Этот потенциал успешно используется для моделирования графеновых структур, углеродных нанотрубок, алмаза и углеводородов.

Энергия химических связей E_b определяется как сумма:

$$E_b = \sum_i \sum_{j > i} [V^R(r_{ij}) - b_{ij} V^A(r_{ij})], \quad (16)$$

где $V^R(r)$ представляет собой энергию отталкивания между парой атомов и запи-

сывается в виде

$$V^R(r) = f^c(r)(1 + Q/r)Ae^{-\alpha r}, \quad (17)$$

$V^A(r)$ представляет собой энергию притяжения между парой атомов

$$V^A(r) = f^c(r) \sum_{n=1,3} B_n e^{-\beta_n r}. \quad (18)$$

Выражение (16) используется для вычисления потенциальной энергии всей системы. Функция b_{ij} называется порядком связи и определяет характер связи:

$$\bar{b}_{ij} = \frac{1}{2}[b_{ij}^{\sigma-\pi} + b_{ji}^{\sigma-\pi}] + b_{ij}^\pi \quad (19)$$

Функции $b_{ij}^{\sigma-\pi}$ и $b_{ji}^{\sigma-\pi}$ зависят от ближайшего окружения и валентных углов, в которых участвуют атомы i и j . Функция b_{ij}^π определяется как сумма двух выражений:

$$b_{ij}^\pi = \Pi_{ij}^{RC} + b_{ij}^{DH} \quad (20)$$

Эмпирическая формула для (19) подобрана Д. Бреннером таким образом, чтобы воспроизводились известные значения порядка связи для дискретных точек. Первое слагаемое в (19) определяется следующим образом:

$$b_{ij}^{\sigma-\pi} = \left[1 + \sum_{k(\neq i,j)} f_{ik}^c(r_{ik})G(\cos(\theta_{ijk}))e^{\lambda_{ijk}} + P_{ij}(N_i^C, N_i^H) \right]^{-\frac{1}{2}} \quad (21)$$

где функция $f^c(r)$ обеспечивает наличие взаимодействия только между ближайшими соседями. Функция P представляет собой бикубический сплайн, который является функцией величин N_i^C и N_i^H , которые, в свою очередь, представляют собой число ближайших к i -му атому атомов углерода и водорода и определяются выражениями:

$$N_i^C = \sum_{k(\neq i,j)}^{carbon atoms} f_{ik}^c(r_{ik}), \quad (22)$$

$$N_i^H = \sum_{l(\neq i,j)}^{hydrogen atoms} f_{il}^c(r_{il}). \quad (23)$$

Выражение π_{ij}^{RC} является трикубическим сплайном F :

$$\pi_{ij}^{RC} = F_{ij} \left(N_i^t, N_j^t, N_{ij}^{conj} \right). \quad (24)$$

N_{ij}^{conj} определяется выражением:

$$N_{ij}^{conj} = 1 + \left[\sum_{k(\neq i,j)}^{carbon} f_{ik}^c(r_{ik}) F(X_{ik}) \right]^2 + \left[\sum_{l(\neq i,j)}^{carbon} f_{il}^c(r_{il}) F(X_{il}) \right]^2, \quad (25)$$

где

$$F(\chi_{ik}) = \begin{cases} 1 & \chi_{ik} < 2 \\ [1 + \cos(2\pi(\chi_{ik} - 2))]/2 & 2 < \chi_{ik} < 3 \\ 0 & \chi_{ik} > 3 \end{cases} \quad (26)$$

и

$$\chi_{ik} = N_k^t - f_{ik}^c(r_{ik}) \quad (27)$$

Слагаемое b_{ij}^{DH} в выражении (20) определяется как:

$$b_{ij}^{DH} = T_{ij}(N_i^t, N_j^t, N_{ij}^{conj}) \left[\sum_{k(\neq i,j)} \sum_{l(\neq i,j)} (1 - \cos^2(\theta_{ijkl})) f_{ik}^c(r_{ik}) f_{jl}^c(r_{il}) \right], \quad (28)$$

где

$$\cos(\theta_{ijkl}) = e_{ijk} e_{ijl}. \quad (29)$$

Функция $T_{ij}(N_i^t, N_j^t, N_{ij}^{conj})$ также является трикубическим сплайном, e_{ijk} и e_{ijl} представляют собой единичные векторы в направлении векторного произведения $R_{ji} \times R_{ik}$ и $R_{ij} \times R_{jl}$, где R - вектор, соединяющий соответствующие атомы.

Функция-переключатель $f^c(r)$ определяется следующим выражением:

$$f_{ij}^c(r) = \begin{cases} 1 & r < D_{ij}^{min} \\ [1 + \cos(\pi((r - D_{ij}^{min})/(D_{ij}^{max} - D_{ij}^{min})))]/2 & D_{ij}^{min} < r < D_{ij}^{max} \\ 0 & r > D_{ij}^{max} \end{cases} \quad (30)$$

1.4 Силовое поле AIREBO

Силовое поле AIREBO представляет собой расширение силового поля REBO путем добавления слагаемых, учитывающих энергию двугранных углов и энергию ван-дер-ваальсового взаимодействия [13]. Полная энергия записывается следующим образом:

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} \left(E_{ij}^{REBO} + E_{ij}^{LJ} + \sum_{k \neq i,j} \sum_{l \neq i,j,k} E_{kijl}^{tors} \right) \quad (31)$$

Ван-дер-ваальсово взаимодействие описывается с использованием модифицированного потенциала Леннарда-Джонса:

$$E_{ij}^{LJ} = S(t_r(r_{ij}))S(t_b(b_{ij}^*))C_{ij}V_{ij}^{LJ}(r_{ij}) + [1 - S(t_r(r_{ij}))]C_{ij}V_{ij}^{LJ}(r_{ij}). \quad (32)$$

V_{ij}^{LJ} – традиционная форма потенциала Леннарда-Джонса

$$V_{ij}^{LJ} = 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right). \quad (33)$$

Потенциал Леннарда-Джонса в силовом поле AIREBO значительно модифицируется путем использования специальных масштабирующих функций. Функция $S(t)$ может быть записана в следующей форме:

$$S(t) = \Theta(-t) + \Theta(t)\Theta(1-t)[1 - t^2(3 - 2t)] \quad (34)$$

Величина потенциала Леннарда-Джонса зависит от ближайшего окружения атома, для которого он вычисляется. Постепенное исключение взаимодействия при уменьшении расстояния r_{ij} осуществляется с помощью масштабирующей функции t_b :

$$t_b(b_{ij}) = \frac{b_{ij} - b_{ij}^{min}}{b_{ij}^{max} - b_{ij}^{min}}, \quad (35)$$

где b_{ij} представляет собой величину порядка связи классического REBO потенциала для атомов i и j . Использование этого параметра позволяет исключить ван-дер-ваальсово отталкивание между парами атомов, которые могут образовать

химическую связь. Также следует отметить, что используемое в формуле (32) значение величины порядка связи b_{ij}^* вычисляется таким образом, будто расстояние между атомами i и j равно максимальному расстоянию, на котором образуется химическая связь для данной пары атомов. Это необходимо, поскольку, в противном случае, величина порядка связи равнялась бы нулю, и наличие ван-дер-ваальсового отталкивания не позволило бы атомам приблизиться на достаточное расстояние для образования химической связи.

Если атомы i и j соединены химической связью, или являются концами цепочки нескольких соединенных химически атомов, то ван-дер-ваальсово взаимодействие между ними корректируется следующей переключающей функцией:

$$C_{ij} = 1 - \max\{w_{ij}(r_{ij}), w_{ik}(r_{ik})w_{kj}(r_{kj}), \forall k, w_{ik}(r_{ik})w_{kl}(r_{kl})w_{lj}(r_{lj}), \forall k, l\} \quad (36)$$

где

$$w_{ij}(r_{ij}) = S'(t_c(r_{ij})), \quad (37)$$

$$S'(t_c(r_{ij})) = \Theta(-t) + \Theta(t)\Theta(1-t)\frac{1}{2}[1 + \cos(\pi t)] \quad (38)$$

Составляющая выражения (31), отвечающая за двугранные углы, образуемые атомами i, j, k, l , имеет следующий вид:

$$E_{kijl}^{tors} = w_{ki}(r_{ki})w_{ij}(r_{ij})w_{jl}(r_{jl})V^{tors}(\omega_{kijl}) \quad (39)$$

где $V^{tors}(\omega_{kijl})$ имеет вид:

$$V^{tors}(\omega_{kijl}) = \frac{256}{405}\epsilon_{kijl}\cos^{10}(\omega_{kijl}/2) - \frac{1}{10}\epsilon_{kijl}. \quad (40)$$

Наличие дополнительного потенциала взаимодействия для двугранных углов позволяет более точно воспроизводить механические свойства графена и других структур, в которых присутствуют планарные углеродные образования.

2 Структура и особенности работы программного пакета Kvazar

Программный пакет состоит из двух основных модулей. Первый модуль предназначен для произведения численных расчетов. Он является основой программного пакета. Второй модуль представляет собой пользовательский графический интерфейс, содержащий модули для пре- и постпроцессинга. Первый модуль может быть использован независимо от второго, и является в этом смысле первичным. Второй (графический) модуль для своей работы требует установленную версию вычислительного модуля, кроме того, данный модуль предназначен исключительно для подготовки входных данных для вычислительного модуля и обработки выходных данных.

2.1 Установка пакета

Открытый программный пакет Kvazar реализуется с использованием системы контроля версий Mercurial [16]. Основной репозитарий доступен в сети интернет по адресу hg.nanokvazar.ru/Kvazar. Для получения актуальной версии программного пакета можно воспользоваться командой Mercurial:

```
hg clone http://hg.nanokvazar.ru/Kvazar
```

Установка программного пакета Kvazar осуществляется при помощи Python модуля easy_install. Программный пакет Kvazar имеет следующий набор зависимостей:

- Numpy - библиотека на Python, реализующая эффективные алгоритмы работы с многомерными массивами, а также разнообразные численные алгоритмы линейной алгебры [17];
- Scipy - библиотека на Python, реализующая большое количество численных алгоритмов для научных вычислений [18];
- PyYaml - представляет собой библиотеку на Python, реализующую необходимый инструментарий для работы с текстовыми файлами в формате YAML;
- mpi4py - библиотека на Python, представляющая собой реализацию OpenMPI [19](реализация MPI с открытым исходным кодом) для Python [20].

- Boost-Python - библиотека на C++, позволяющая реализовывать быстродействующие расширения для Python [21];

Перед сборкой или установкой пакета необходимо установить Boost-Python, прочие зависимости будут разрешены в процессе установки пакета автоматически. Для тестовой сборки программного пакета можно воспользоваться следующей командой:

```
python setup.py build
```

Юнит-тестирование модулей программного пакета осуществляется командой:

```
python setup.py test
```

Для установки пакета можно воспользоваться командой¹:

```
python setup.py install
```

После установки программы и подготовки специального конфигурационного файла в формате YAML² запуск программы на расчет осуществляется командой:

```
kvazar path/to/config.yml
```

2.2 Вычислительный модуль

2.2.1 Структура вычислительного модуля

Вычислительный модуль представляет собой программный пакет, реализованный на Python. Он состоит из набора вычислительных подмодулей, имеющих трехуровневую иерархию. На первом (нижнем) уровне иерархии находятся модули, реализованные на C++ и выполненные в виде расширений Boost-Python. Данные расширения реализуют интерфейсные вычислительные функции, написанные в процедурном стиле программирования. Интерфейсные функции становятся методами классов на следующем (втором) уровне иерархии, который выполнен в объектно-ориентированном стиле программирования на языке Python. Также

¹ Данная команда может потребовать права суперпользователя(root) для успешного осуществления.

² В специальном конфигурационном файле необходимо задать параметры расчета, подробнее о создании конфигурационного файла будет изложено ниже

на втором уровне иерархии находятся связующие модули, которые реализует инфраструктуру программного пакета путем агрегирования модулей первого уровня иерархии. Иерархия модулей исключает циклические зависимости, что упрощает их дальнейшую разработку. Каждый модуль на данном уровне иерархии документирован (используется генератор документации Sphinx). Тестирование всех модулей осуществляется при помощи стандартного инструмента unittest. После установки пакета, модули данного уровня можно использовать для написания пользовательских вычислительных Python-скриптов. На третьем уровне иерархии располагаются модули, реализующие пользовательский доступ к возможностям программного пакета. Данные модули управляют процессом консольного запуска.

2.2.2 Классы вычислительного модуля

На рис. 1 представлена файловая система и UML-диаграмма основных классов вычислительного модуля.

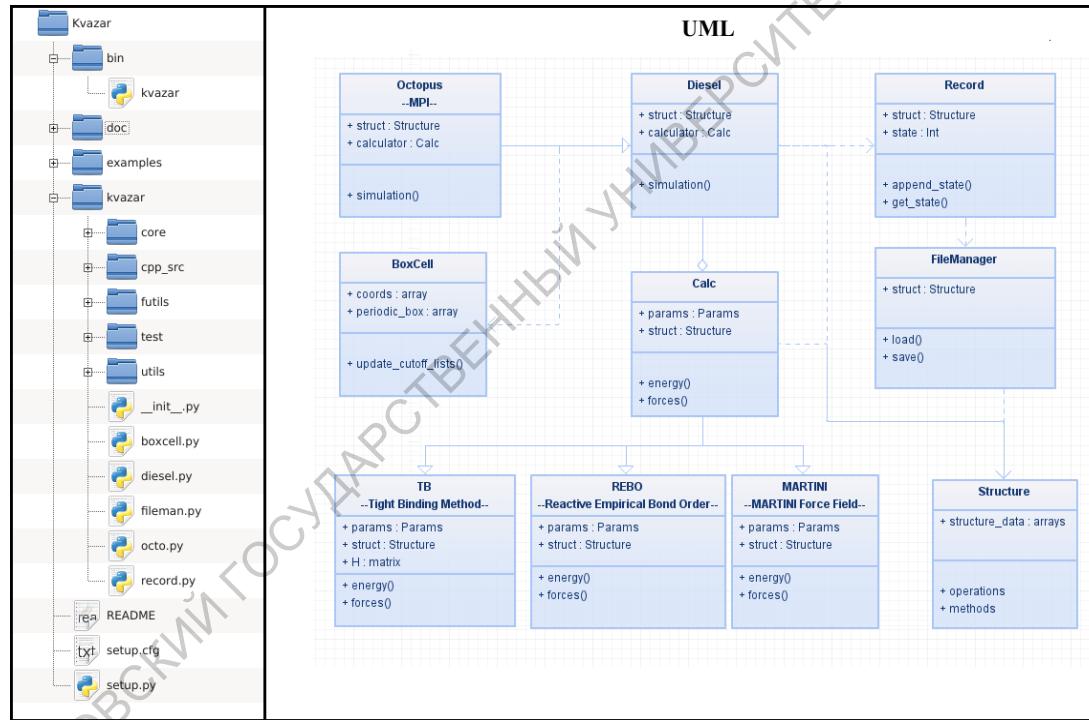


Рис. 1: Файловая система вычислительного модуля(слева) и UML-диаграмма его основных классов(справа)

Класс **Structure** предназначен для оперирования с данными о моделях наноструктур. Модель каждой наноструктуры представляет собой экземпляр класса **Structure**, который содержит информацию о геометрии, топологии и элементар-

ном составеnanoструктуры. Объекты типа **Structure** реализуют операции копирования, выделения подструктур из основной структуры, удаления подструктур, объединение структур и др.

Класс **Calc** реализует единый интерфейс к силовым полям. Каждое силовое поле в этой схеме представляет собой класс-наследник от базового класса **Calc**. Класс **Calc** организует процесс агрегирования интерфейсных функций из Boost-Python расширений. Предполагается, что экземпляры данного класса опирируют над экземплярами класса **Structure**, характеризуя таким образом их свойства.

Класс **FileManager** предназначен для загрузки и сохранения данных о молекулярной структуре в текстовый файл в соответствующем формате (.acd/.pdb).

Класс **Record** предназначен для записи данных о процессе расчета молекулярной динамики в выходной файл, представляющий собой zip-архив. Экземпляр данного классса ассоциируется с соответствующим zip-архивом, предоставляя интерфейс к его данным. Данный класс может быть использовать для записи или чтения соответствующих данных.

Класс **BoxCell** реализует алгоритм связанных ячеек (Link Cells)][9], который предназначен для вычисления списков Верле (Verlet Lists) [10]. Списки Верле, представляют собой списки ближайших соседей для каждого атома. Они используются для быстрого вычисления энергии ван-дер-ваальсового взаимодействия между несвязанными атомами путем отсечения радиуса действия сил ван-дер-ваальса. Этот подход основан на том, что энергия званиямодействия ван-дер-ваальса быстро убывает при увеличении расстояния между атомами (см. рис. 2 (а)), что позволяет пренебречь взаимодействием между далеко расположенными друг от друга парами атомов. При использовании схемы отсечения энергии взаимодействия ван-дер-ваальса на дальних расстояниях функцию вычисления энергии домножают на сглаживающую функцию специального вида, которая обеспечивает плавное уменьшение энергии взаимодействия до нуля (см. рис. 2 (б)).

Поскольку в процессе расчета молекулярной динамики атомы меняют своё положение, с течением времени списки Верле необходимо пересчитывать. Прямой алгоритм, при котором перебираются все пары атомов, имеет квадратичную ассимптотическую сложность, что очень затратно с точки зрения ресурсов. На практике списки Верле обновляются не на каждом временном шаге, а с некоторой периодичностью. Такой подход, очевидно, может привести к существенным

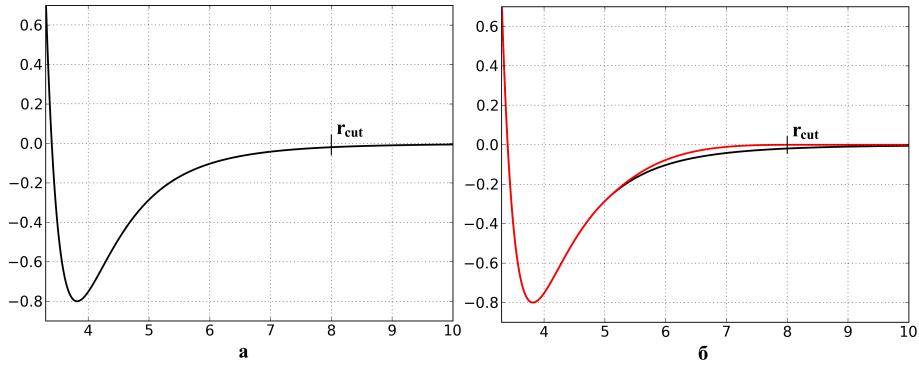


Рис. 2: Схематичный график зависимости энергии ван-дер-ваальса от расстояния между взаимодействующими частицами (а), r_{in} - радиус отсечения. Для предотвращения разрыва производной энергии взаимодействия между двумя частицами используется специальная сглаживающая(б). Красным цветом обозначена сглаженная функция энергии взаимодействия ван-дер-ваальса между двумя частицами.

вычислительным ошибкам. В более удачном подходе списки списки Верле строятся не для радиуса отсечения r_{cut} , а для большего радиуса r_{out} , при этом, если какой-либо атом сместился на расстояние $r_{out} - r_{cut}$ от своего положения в момент предыдущего пересчета списков(см. рис. 3).

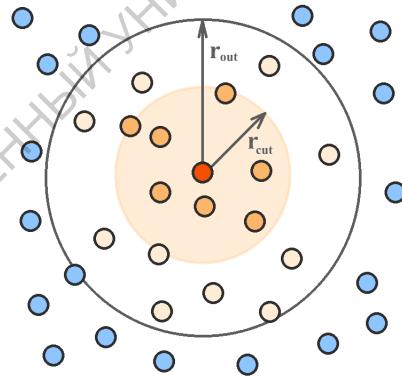


Рис. 3: Принцип построения списков Верле в условиях постоянно меняющегося расположения атомов.

Квадратичная асимптотика построения списков Верле может существенно замедлить расчет при большом числе атомов. Использование алгоритма связанных ячеек (Link Cells), позволяет решить эту проблему. В данном алгоритме пространство разбивается на ячейки, для которых строятся соответствующие списки частиц, а для каждой частицы вычисляется индекс ячейки, к которой она относится. Для построения списка Верле для каждой частицы вместо полного перебора всех

атомов, перебираются только списки частиц соседних ячеек. При равномерной плотности частиц и большом объеме пространства, этот алгоритм имеет линейную асимптотику. На рис. 4 представлена двумерная иллюстрация алгоритма связанных ячеек. Таким образом использование алгоритма связанных ячеек и списков Верле позволяет добиться линейной асимптотики расчета одного шага молекулярной динамики.

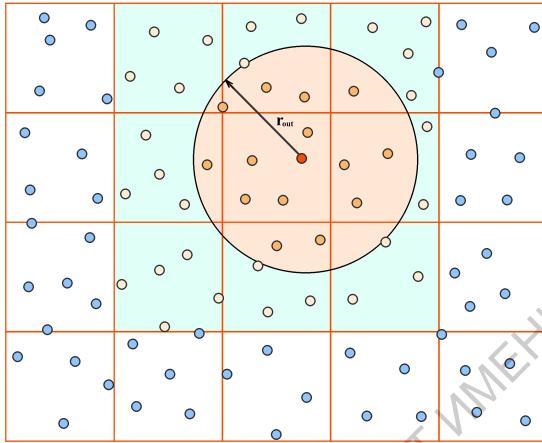


Рис. 4: Иллюстрация применения алгоритма связанных ячеек в двумерном случае. Для получения списка ближайших соседей перебираются атомы в смежных ячейках, при этом отбрасываются атомы, не попавшие в сферу радиуса r_{out}

В методе молекулярной динамики часто бывает удобно использовать периодические граничные условия, при которых счетная область представляет собой параллелепипед, противоположные грани которого замыкаются друг на друга. (см. рис. 5). В терминах периодических граничных условий, каждая частица помимо своего номинального местоположения одновременно существует и в точках, соответствующих образам данной частицы в смежных параллелепипедах. Расстояние между частицами вычисляется как минимальное расстояние между всеми парами соответствующих им образов. Когда в процессе молекулярной динамики частица преодолевает границу, разделяющую ячейки, её координаты сдвигаются таким образом, чтобы она вернулась в ту же самую ячейку с противоположной грани.

При использовании алгоритма связанных ячеек с периодическими граничными условиями, ячейки, расположенные на противоположных гранях параллелепипеда, считаются соседними. Если вычислять расстояния между частицами в терминах периодических граничных условий, списки Верле будут соответствовать спискам Верле для периодических граничных условий, посчитанных полным пере-

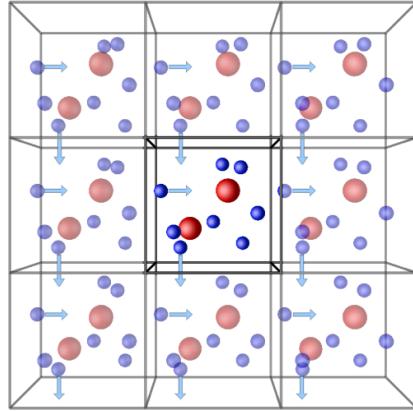


Рис. 5: Иллюстрация применения периодических граничных условий.

бором всех атомов. На рис. 6 представлена двумерная иллюстрация применения алгоритма связанных ячеек с периодическими граничными условиями . Описанные

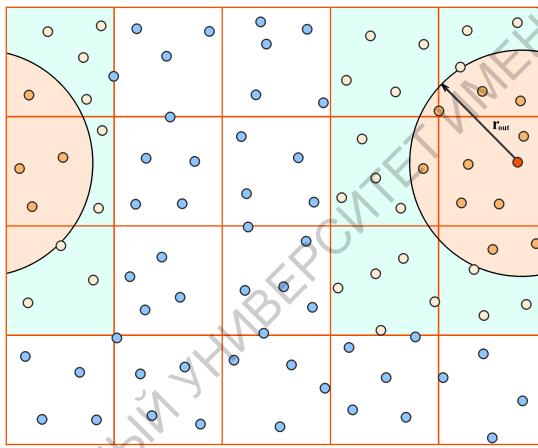


Рис. 6: Иллюстрация применения алгоритма связанных ячеек в двумерном случае при использовании периодических граничных условий.

алгоритмы реализованы в классе **BoxCell**.

Класс **Diesel**, созданный для вычисления молекулярной динамики, использует экземпляр класса **Boxcell** для построения списков ближайших соседей в процессе молекулярно-динамического расчета. Методы класса **Diesel** вычисляют траекторию движения молекулярной системы. Алгоритмы терmostатов позволяют моделировать теплообмен молекулярной системы с внешней средой при заданной температуре.

Объект класса **Diesel** в качестве входных параметров принимает экземпляр класса-наследника **Calc**, реализующего силовое поле, который, в свою очередь, содержит экземпляр класса **Structure**, для которого и будет вычисляться молекулярная динамика. Сохранения данных расчета осуществляется при помощи эк-

земпляра класса **Record**. В классе реализованы методы для вычисления таких динамических параметров молекулярной системы, как температура, полная энергия системы, кинетическая энергия системы, полный импульс системы, полный момент импульса системы и др.

Класс **Octopus** реализует весь набор возможностей класса **Diesel** в параллельном режиме. Для обмена данными в многопроцессорном окружении используется интерфейс обмена данными MPI. Параллельный алгоритм молекулярной динамики использует подход, в котором все вычислительные узлы считаются равноправными и вычисляют динамику некоторого подмножества атомов молекулярной системы. После вычисления соответствующих координат на следующем временном шаге расчета, происходит обмен актуальными координатами структуры, после чего процесс повторяется. На рис. 7 приведена блок-схема алгоритма параллельного вычисления молекулярной динамики, реализованная в классе **Octopus**. Использование параллельных вычислений позволяет значительно увеличить ско-

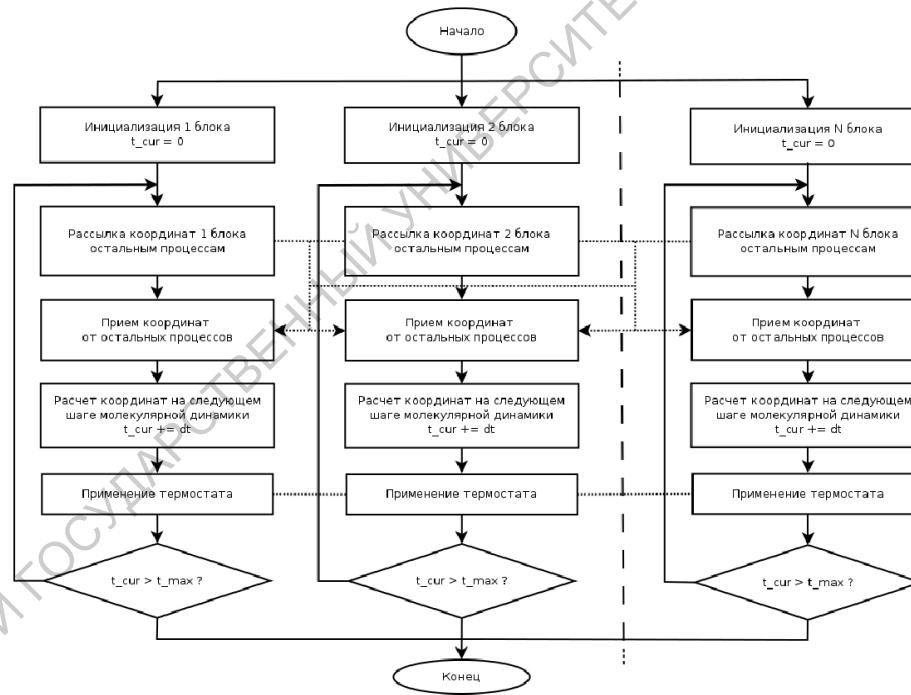


Рис. 7: Блок-схема параллельного алгоритма молекулярной динамики.

рость вычисления молекулярной динамики, но ввиду наличия ресурсозатратных операций по пересылке данных, ускорение имеет не линейную, а логарифмическую зависимость от числа узлов вычислительного кластера. На рис. 8 приведен график зависимости получаемого ускорения в результате использования разного

числа процессов с классом **Octopus** по сравнению с использованием последовательной версии молекулярной динамики, реализованной в классе **Diesel**.

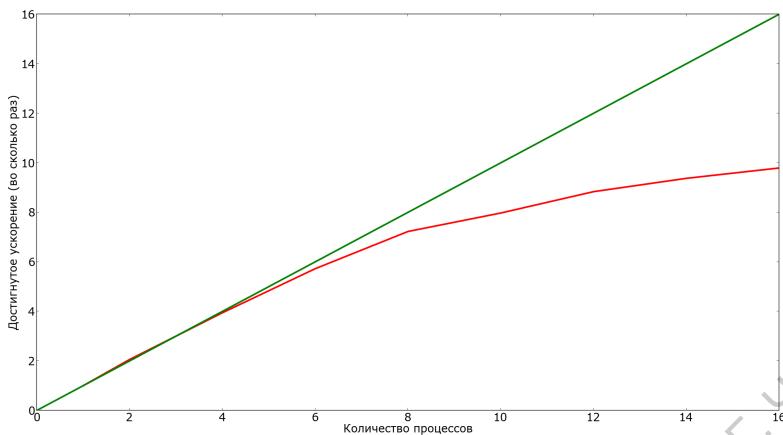


Рис. 8: График зависимости ускорения, полученного в результате использования параллельной версии алгоритма молекулярной динамики, по сравнению с последовательной версией алгоритма, от числа выделенных узлов.

2.2.3 Организация расчетов

Для организации вычислительного процесса необходимо создать специальный конфигурационный файл. Конфигурационный файл представляет собой текстовый файл в YAML формате. В данном формате задаются параметры вычисления.

Параметры могут иметь единичное значение, в этом случае используется формат:

```
parameter : value
```

где *parameter* - имя параметра, *value* - значение данного параметра, это может быть целое число, дробное число, текст без пробелов или логическое значение (yes/no).

Параметры могут иметь список значений, в этом случае используется формат:

```
parameter : [value1, value2, ..., valueN]
```

Также список значений может быть записан в следующем формате:

```
parameter :
```

```
- value1  
- ...  
- valueN
```

Параметры могут быть вложенными, в этом случае используется следующий формат:

```
parameter :  
    inner_parameter1 : value1  
    ...  
    inner_parameterN : valueN
```

Особенностью формата YAML является тот факт, что сами значения *valueI* могут являться списками. Внутренние параметры в качестве своих значений также могут иметь внутренние параметры. Таким образом данный способ задания параметров обладает широкими возможностями и большой гибкостью.

Параметр *calculation_mode* задает вычислительный режим и может принимать значения *SINGLE_POINT*, *MOLECULAR_DYNAMIC* и *OPTIMIZATION*. Вычислительный режим *SINGLE_POINT* позволяет вычислять энергетические и прочие параметры наноструктуры фиксированной геометрии. Вычислительный режим *MOLECULAR_DYNAMIC* используется для вычисления молекулярной динамики системы в выбранном силовом поле. Вычислительный режим *OPTIMIZATION* служит для оптимизации геометрии наноструктуры в рамках заданного силового поля.

Следующие параметры могут быть заданы для каждого из доступных вычислительных режимов:

- *input* - путь к файлу с наноструктурой;
- *output* - путь к выходному файлу;
- *method* - имя силового поля или метода, в рамках которого будет осуществлен расчет. Данный параметр может принимать значения *REBO*, *AIREBO*, *TB*, *MARTINI*;
- *params* - список параметров, который зависит от значения параметра *method*;

- *periodic_box* - задает параметры периодических граничных условий - координаты параллелепипеда, противоположные грани которого замыкаются сами на себя. Необходимо указать минимальные и максимальные значения координат параллелепипеда для каждой из осей декартовой системы координат в ангстремах:

```
periodic_box : [[x_min, y_min, z_min], [x_max, y_max, z_max]]
```

Для вычислительного режима *SINGLE_POINT* могут быть заданы следующие параметры:

- *input*³;
- *method**
- *output*;
- *params*;
- *periodic_box;dt* : 0.1

Для вычислительного режима *MOLECULAR_DYNAMIC* доступно определение следующих параметров:

- *input**
- *output** - выходной файл должен иметь формат *.nmd*;
- *method**
- *max_time** - время расчета в фемтосекундах;
- *save_period** - периодичность сохранения данных - целое число временных шагов;
- *dt* - шаг по времени, по умолчанию - 1 фс;
- *params*;

³Обязательные параметры для данного вычислительного режима помечены символом *

- *periodic_box*;
- *pbc_box_vels* - параметр, позволяющий задать постоянную скорость деформации для параллелепипеда, определяющего периодические граничные условия. Скорость задается в A/fs и представляет собой список из трех чисел - скорость движения граней параллелепипеда по каждой из осей декартовой системы координат;
- *thermostat* - термостат. Текстовый параметр, который может иметь значения *Berendsen*, *Nose-Hoover*;
- *temperature* - температура термостата(обязательный параметр, если задан термостат) - вещественное число, представляющее собой температуру в Кельвинах;
- *monitoring* - мониторинг термодинамических параметров. Задается список параметров, которые нужно вычислять и сохранять в процессе расчета. Список может содержать следующие значения:
 - *temperature* - температура системы в Кельвинах;
 - *U* - потенциальная энергии системы в электронвольтах;
 - *K* - потенциальная энергии системы в электронвольтах;
 - *energy* - полная энергии системы в электронвольтах;
- *constant_velocities* - ограничение на скорости атомов, позволяющее задать постоянную скорость группе атомов, эта возможность может быть использована, например, при моделировании процесса наноиндентирования. Параметр представляет собой список ограничений, каждое ограничение в свою очередь является списком из трех списков, где первый список содержит три числа - скорости в A/fs , второй список состоит из номеров атомов, к которым применяется соответствующее ограничение, третий список (не обязательен), представляет собой список списков пар чисел, где каждая пара задает диапазон атомов к которым, для которых применяется данное ограничение по скорости. Пример использования данного ограничения:

```
constant_velocities :
```

- `[[vx1, vy1, vz1], [[index1, ...]], [[start_1, stop1], ...]]`
- `...`
- `[[vxN, vyN, vzN], [[indexN, ...]], [[startN, stopN], ...]]`

- *external_forces* - ограничение, позволяющее установить добавочную внешнюю силу, действующую на атомы структуры. Формат этого параметра совпадает с форматом параметра *constant_velocities*. Сила задается в Ньютонах.

Ниже приведен пример конфигурационного файла для запуска наноструктуры на вычисление молекулярной динамики в термостате Нозе-Гувера при температуре 300 Кельвин в периодических граничных условиях в рамках потенциала REBO:

```
calculation_mode : MOLECULAR_DYNAMIC
method : REBO
input : path/to/structure.ACD
output : path/to/out.nmd
thermostat : Nose-Hoover
temperature : 300
max_time : 1000
save_period : 10
periodic_box : [[-100.0, -100.0, -100.0], [100.0, 100.0, 100.0]]
monitoring : [energy, temperature, momentum]
```

Для вычислительного режима *OPTIMIZATION* могут быть заданы следующие параметры:

- *input**;
- *method**;
- *output*;
- *params*;
- *periodic_box*;
- *own_grad* - может принимать значения (yes/no). Данный параметр нужен для произведения оптимизации системы в рамках силового поля, для которого не реализована возможность вычисления градиента.

- *periodic_box*;
- *maxiter* - максимальное количество итераций при оптимизации;
- *precision* - точность вычисления координат, соответствующих минимум энегрии системы;
- *step* - стартовое значение "шага" в методе градиентного спуска;
- *optimize_box* - может принимать значения (yes/no). Данный параметр нужен для оптимизации координат периодической коробки, в которую помещается наноструктура;
- *save_period* - целое число. Периодичность сохранения промежуточных данных о геометрии структуры в процессе оптимизации.
- *directions* - список координат, по которым будет происходить оптимизация координат периодической коробки. Координаты указываются в виде меток "x" "y" "z". Например:

```
directions : [x, z]
```

2.3 Графический модуль

Графический модуль представляет собой пользовательский графический интерфейс, состоящий из двух подмодулей для соответственно пре- и постпроцессинга. Подмодуль препроцессинга предназначен для построения, модификации наноструктур. Подмодуль постпроцессинга ориентирован на работу с выходными файлами, создающимися в результате работы вычислительного модуля.

Для работы графического модуля необходимо наличие установленного вычислительного модуля. Модули графического интерфейса имеют двухуровневую иерархию. На нижнем уровне иерархии находятся виджеты для 3D визуализации молекулярных структур, которые используют библиотеку PyOpenGL [22]. Виджеты для визуализации встраиваются в модули, реализующие пользовательский графический интерфейс на следующем уровне иерархии. Как и во всех подобных программах в виджетах графического модуля атомы молекулярной структуры представляются в виде трехмерных сфер, расположенных в соответствующих точках

виртуального пространства, связи между атомами отображаются в виде линий. Размеры и цвета сфер, представляющих атомы, зависят от типов отображаемых атомов.

На втором уровне иерархии располагаются компоненты графического интерфейса, такие как панели инструментов, модули отображения молекулярных структур в 3D, модули проектирования молекулярных структур. Данные модули реализованы с использованием библиотеки PyQt4 [23]. Используемые с предыдущего уровня иерархии базовые виджеты обладают следующей системой навигации при встраивании в интерфейс на Qt:

- при вращении колеса мыши происходит изменение масштаба отображаемой структуры;
- при перемещении мыши с зажатой левой кнопкой мыши, происходит вращение молекулярной структуры относительно курсора структуры;
- курсор структуры выбирается автоматически как центр масс структуры или выделенного участка структуры;
- при перемещении мыши с зажатой правой кнопкой происходит перемещение структуры по экрану.

На рис. 9 изображен пример работы виджета визуализации.

2.3.1 Модуль препроцессинга

Графический модуль препроцессинга реализует пользовательский графический интерфейс, который предназначен для загрузки, редактирования и сохранения молекулярных структур. Данный модуль использует модуль первого уровня иерархии для визуализации процесса конструирования молекулярных структур и процесса их изменения. Он имеет существенно более расширенную систему навигации:

- одновременное нажатие на левую и правую кнопки мыши приводит к выделению единичных атомов, всех атомов заданного выбранного типа или связных участков молекулярной структуры, в зависимости от выбранного режима выделения объектов;

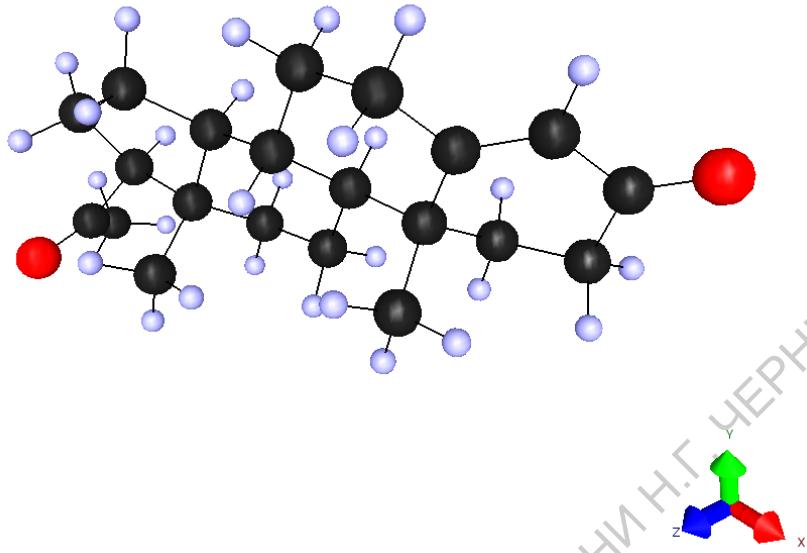


Рис. 9: Визуализация молекулы прогестерона.

- нажатие на клавишу пробел приводит активации режима редактирования молекулярной структуры. В данном режиме перемещение и вращение происходит только с выделенным объектом;
- сочетание клавиш $<\text{Ctrl} + \text{B}>$ создает виртуальную коробку, предназначенную для выделения участковnanoструктур. Виртуальной коробкой можно манипулировать так же, как и молекулярными структурами.

В графическом модуле препроцессинга реализовано программное меню, дублирующее перечисленные функции и добавляющее новые функции (внешний вид программного меню приведено на рис. 10). На рисунке 11 приведен пример работы модуля препроцессинга в режиме редактирования nanoструктуры.

Меню имеет следующую структуру:

- File (см. Рис. 10 а):
 - Load - загрузка:
 - * **Append structure** - добавление к текущей структуре, структуры из файла;
 - * **Load structure** - загрузка структуры из файла;

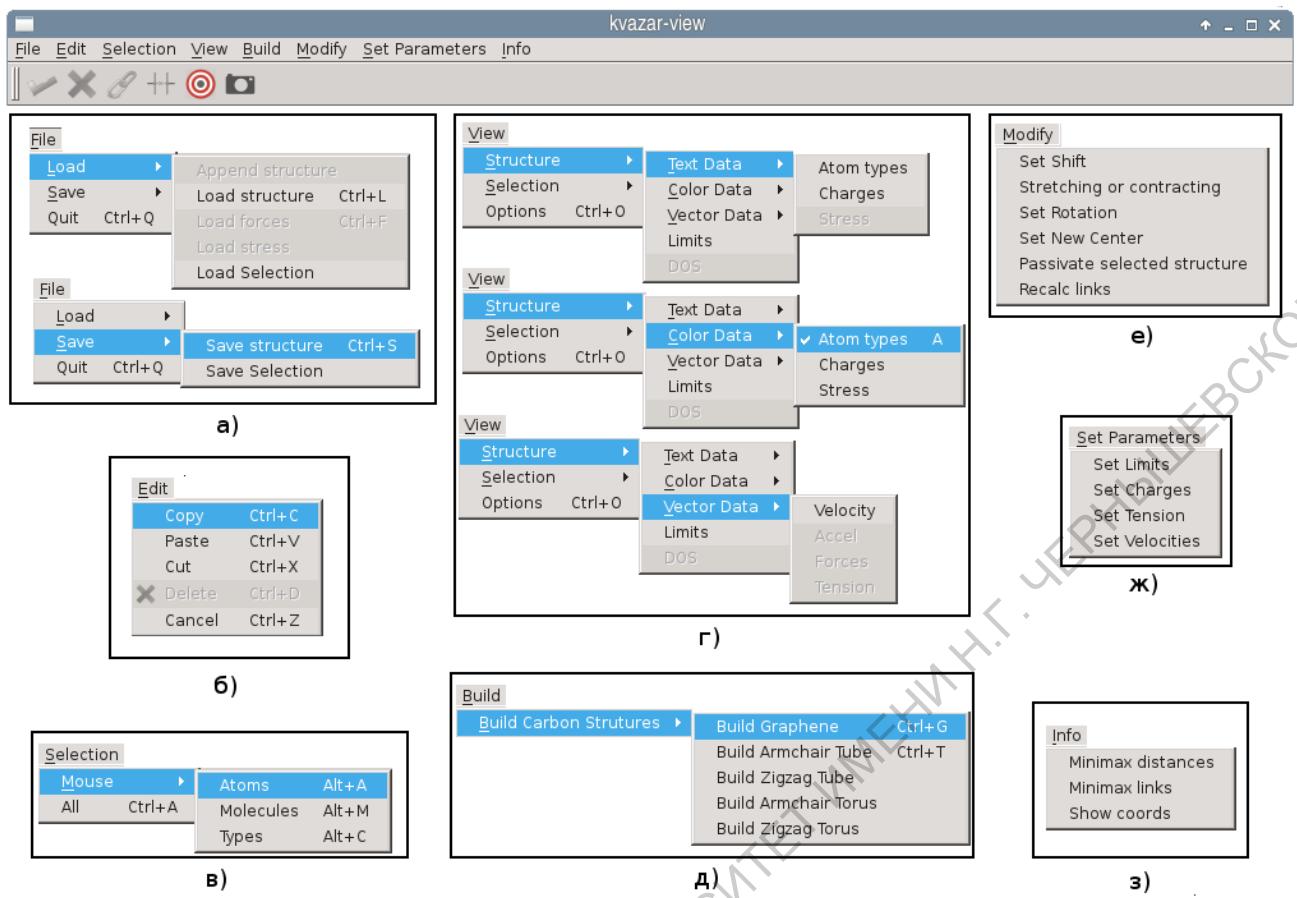


Рис. 10: Программное меню модуля препроцессинга.

- * **Load forces** - загрузка данных о силах, действующих на атомы системы;
- * **Load stress** - загрузка данных о напряжениях на атомах системы;
- * **Load Selection** - загрузка списка выделенных атомов;
- **Save** - сохранение:
 - * **Save structure** - сохранение текущего состояния структуры в файл;
 - * **Save Selection** - сохранение списка выделенных атомов в файл;
- **Edit** (см. Рис. 10 б) - опции редактирования:
 - **Copy** - копировать выделенный участок структуры;
 - **Paste** - вставить скопированный ранее участок структуры;
 - **Cut** - вырезать выделенный участок структуры;

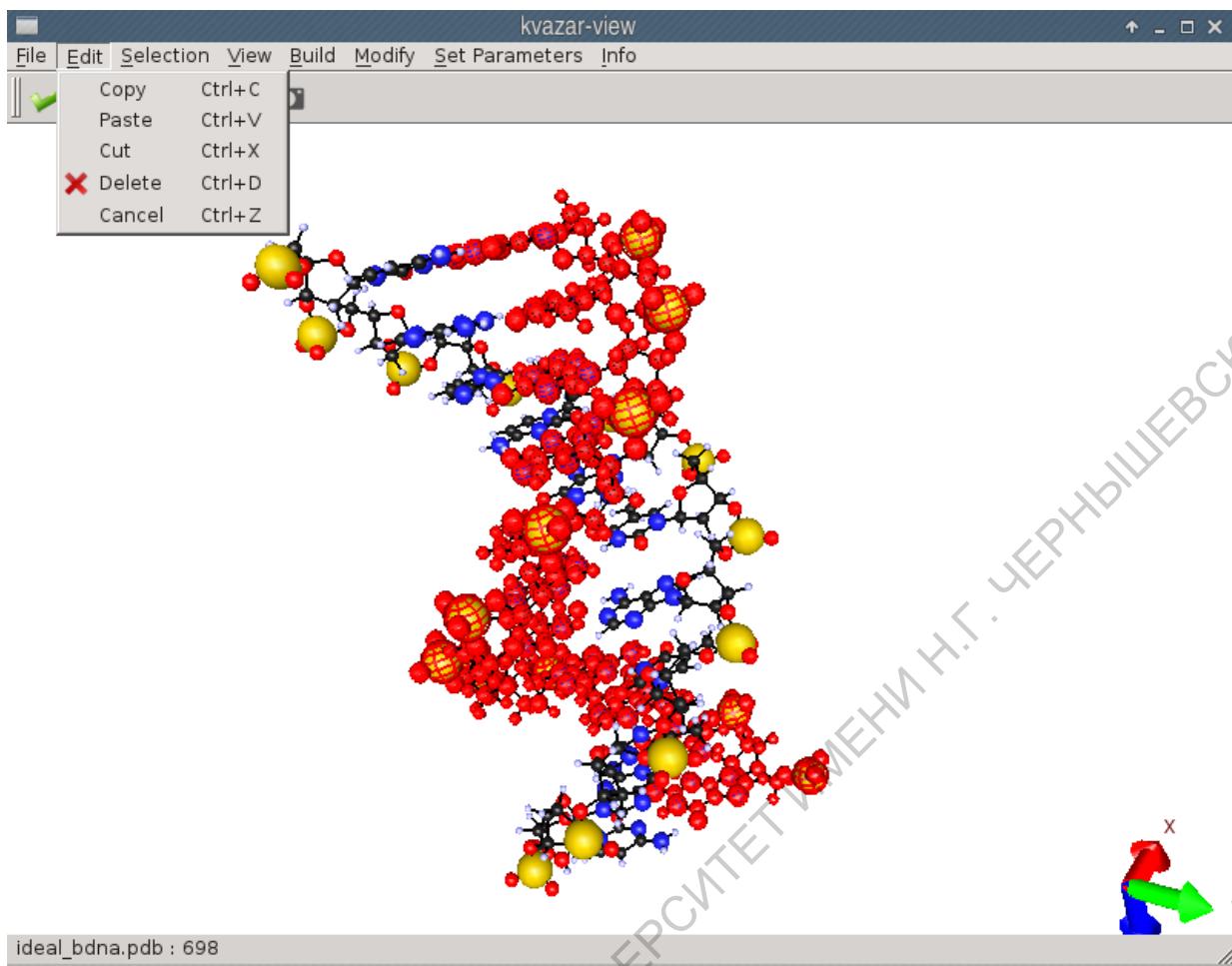


Рис. 11: Пример работы модуля препроцессинга.

- **Delete** - удалить выделенный участок структуры;
- **Cancel** - отменить последнее действие;
- Selection (см. Рис. 10 в) - опции выделения:
 - Mouse - режим выделения мыши:
 - * **Atoms** - выделять по одному атому;
 - * **Molecules** - выделять молекулы;
 - * **Types** - выделять все атомы одинакового типа;
 - **All** - выделить всё;
- View (см. Рис. 10 г) - опции визуализации:
 - Structure - способ отображения структуры:

- * **Text Data** - отображение данных о структуре в виде текста около соответствующих атомов;
 - * **Color Data** - отображение данных о структуре путем изменения цветов атомов;
 - * **Vector Data** - отображение векторных данных, путем отрисовки стрелок, выходящих из центров атомов;
 - * **Limits** - отображение набора ограничений для данной структуры;
 - * **DOS** - отображение графика, содержащего информацию о распределении плотности электронных состояний в зависимости от величины энергии (если эти данные загружены);
 - Selection - способ отображения выделения:
 - * **Atom types** - отображение типов выделенных атомов;
 - * **Numbers** - отображение номеров выделенных атомов;
 - * **Show Box** - отображение коробки, при помощи которой можно выделять атомы структуры;
 - Options - дополнительные опции визуализации;
- Build->Build Carbon Structures (см. Рис. 10 д) - построение наноструктур:
- **Build Graphene** - построение геометрии графена по параметрам;
 - **Build Armchair Tube** - построение геометрии нанотрубки типа Armchair по параметрам;
 - **Build Zigzag Tube** - построение геометрии нанотрубки типа Zigzag по параметрам;
 - **Build Armchair Torus** - построение геометрии нанотора типа Armchair по параметрам;
 - **Build Zigzag Torus** - построение геометрии нанотора типа Zigzag по параметрам;
- Modify (см. Рис. 10 е) - модификация геометрии выделенного участка наноструктуры:
- **Set shift** - линейный сдвиг выделенных атомов на указанный вектор;

- **Stretching or contracting** - линейное растяжение или сдвиг координат выделенных атомов;
 - **Set rotation** - поворот выделенных атомов на нужный угол вокруг заданной оси;
 - **Set new center** - изменение центра координат(не влияет на геометрию структуры);
 - **Passivate selected structure** - присоединение к атомам выделенной структуры атомов водорода;
 - **Recalc links** - пересчет химических связей структуры;
- Set Parameters (см. Рис. 10 ж) - задание значений характеристик выделенных атомов:
 - **Set Limits** - задание набора ограничений для данной структуры (таких как фиксация положения атома и др.);
 - **Set Charges** - задание величины избыточного электрического заряда;
 - **Set Tension** - задание величины напряженности электрического поля;
 - **Set Velocities** - задание величины скоростей;
 - Info (см. Рис. 10 з) - информация о структуре:
 - **Minimax distances** - вычисление минимального и максимального расстояния между всеми парами атомов;
 - **Minimax links** - вычислений минимальной и максимальной длин связей;
 - **Show coords** - отображение таблицы координат и типов атомов. Данную таблицу можно редактировать;

2.3.2 Модуль постпроцессинга

Данный модуль использует виджеты первого уровня иерархии. Он выполнен в стиле классического видео проигрывателя. После открытия файла с записью, становится активным ползунок раскадровки, который позволяет выбрать точку, с

которой начинается проигрывание молекулярной динамики. Как и в любом видеопроигрывателе присутствуют кнопки «старт» и «стоп». Однако, важным отличием от классического видеопроигрывателя является возможность изменить ракурс отображения, что позволяет детально исследовать траекторию, полученную в результате вычисления молекулярной динамики траекторию. Также существует возможность записи фильма с молекулярной динамикой с выбранного ракурса или же с меняющегося ракурса. Помимо просмотра молекулярной динамики модуль постпроцессинга позволяет строить графики изменения энергии и температуры молекулярной системы от времени. На рис. 12 приведен пример работы модуля постпроцессинга. Для запуска модуля необходимо наличие соответствующих скриптов, библиотек (PyOpenGL, PyQt4) и установленного вычислительного модуля.

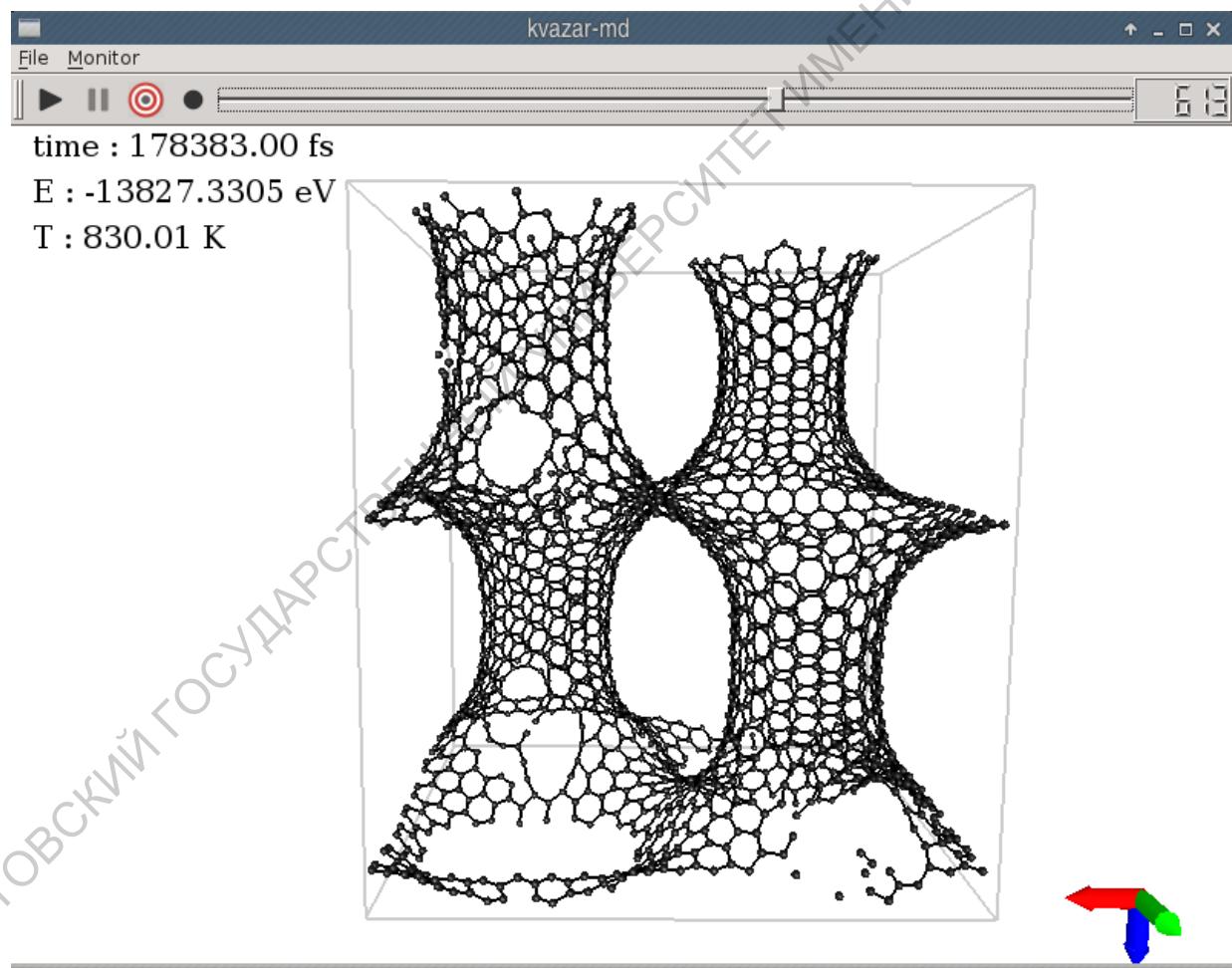


Рис. 12: Пример работы модуля постпроцессинга. Момент разрушения колонного графена в результате его растяжения.

3 Лабораторные работы

3.1 Лабораторная работа №1

"Построение геометрии колонного графена"

Цель работы: овладение навыками конструирования нанообъектов сложной формы в программном пакете Kvazar.

Задачи работы:

- ознакомиться со способами построения нанообъектов в программном пакете Kvazar;
- построить различные конфигурации углеродных нанокластеров (углеродные нанотрубки (УНТ), листы графена, наноторы);
- построить геометрию колонного графена используя стандартные средства графического модуля программного пакета Kvazar;

Теоретическая часть

В настоящее время большой практический интерес вызывают углеродные наноструктуры. Условно их можно разделить на наноструктуры простой и сложной формы. К углеродным наноструктурам простой формы относятся фуллерены, графен, углеродные нанотрубки. Углеродные наноструктуры сложной формы могут представлять собой комбинацию наноструктур простой формы. Примерами углеродных наноструктур сложной формы являются бамбукоподобные нанотрубки (нанотрубки с внутренними перемычками), наностручки (нанотрубки, заполненные фуллеренами), колонный графен (листы графена, соединенные между собой углеродными нанотрубками).

Графен представляет собой двумерную (толщиной в один атом) модификацию углерода, обладающую гексагональной кристаллической решеткой. Нанотрубки получаются путем сворачивания и склейки графенового слоя. Существует множество способов свернуть графитовый слой в цилиндическую поверхность без разрывов, каждый способ сворачивания определяется целыми числами киральности (n, m), которые представляют собой разложение направления сворачивания по векторам трансляции графитовой решётки. Выделяются следующие типы углеродных нанотрубок: акиральные (прямые), киральные (спиральные) нанотрубки.

Ациральные углеродные нанотрубки, в свою очередь, бывают двух типов - zigzag (имеющие вектор киральности $(m, 0)$), armchair (имеющие вектор киральности (n, n)).

Колонный графен представляет собой композит, построенный из ковалентно-связанных дуг с другом углеродных нанотрубок и графеновых листов. Углеродные нанотрубки ориентированы вертикально относительно плоскости графеновых листов и могут быть расположены в шахматном порядке, если структура образована большим числом графеновых листов, расположенных дуг над другом.

В программном пакете Kvazar можно генерировать геометрию прямоугольного листа графена (см. описание программного модуля препроцессинга в разделе 2.3.1) путем задания двух целых чисел N и M . Число N соответствует числу гексагонов вдоль armchair грани прямоугольного листа и условно определяет его ширину, число M соответствует числу атомов вдоль zigzag грани прямоугольного листа и условно определяет его длину. Аналогично генерируются нанотрубки типа zigzag и armchair, параметры N и M для нанотрубок задают геометрические параметры графенового листа, из которого будет свернута соответствующая трубка. Соответственно, для armchair нанотрубки число N определяет диаметр трубки, а число M - её длину, а для zigzag нанотрубки число N определяет длину трубки, а число M - длину. Также в модуле препроцессинга может быть генерирована геометрия наноторов, которые представляют собой свернутые кольцом нанотрубки. Их геометрия также определяется числами N , M , определяющими в данном случае параметры нанотрубки, из которых они будут построены.

Углеродныеnanoструктуры сложной формы могут быть построены при помощи графического модуля. В разделе 2.3.1 также описано, каким образом можно изменять геометрические положение составных компонентов нанообъектов относительно друг друга, удалять участки структур, производить вращение выделенных участков nanoструктур относительно друг друга и т.д. Все эти инструменты дают широкие возможности для построения геометрии углеродных nanoструктур сложной формы.

Практическая часть

1. Построить прямоугольный графеновый лист, шириной от 3.0 до 3.5 нм, и длиной от 5.0 до 5.5 нм.;
2. Построить нанотрубку типа armchair, диаметром от 1.0 до 1.1 нм, и длиной от

- 10.0 до 10.5 нм.;
3. Построить нанотрубку типа zigzag, диаметром от 1.0 до 1.1 нм, и длиной от 10.0 до 10.5 нм.;
 4. Построить наноторп(zigzag или armchair), диаметр нанотрубки от 1.0 до 1.1 нм, диаметр нанотора от 7 до 7.5 нанометров;
 5. Построить соединение графенового листа и вертикально расположенной углеродной нанотрубки, созданных на предыдущем этапе выполнения задания. Необходимо удалить лишние атомы на листе графена, поместить нанотрубку в соответствующее место и добавить ковалентные связи между атомами нанотрубки и графена. Если вдоль соединения образуются четырехугольники или треугольники, необходимо повернуть нанотрубку вдоль вертикальной оси таким образом, чтобы они исчезали;
 6. Используя подготовленную геометрию соединения графенового листа и нанотрубки, построить прямоугольную ячейку колонного графена.

3.2 Лабораторная работа №2

"Оптимизация геометрии колонного графена"

Цель работы: овладения навыками определения оптимальной геометрии наноструктур в рамках программного пакета Kvazar;

Задачи работы:

- ознакомиться с понятием минимума потенциальной энергии наноструктуры;
- ознакомиться с силовым полем REBO;
- ознакомиться с силовым полем AIREBO;
- получить равновесную геометрию наноструктуры колонного графена, путем её оптимизации в рамках потенциала REBO;
- получить равновесную геометрию наноструктуры колонного графена, путем её оптимизации в рамках метода AIREBO;

Теоретическая часть

Краеугольным камнем молекулярного моделирования является понятие энергии молекулярной системы. Поскольку точное определение энергии молекулярной системы методами *ab initio* на текущем этапе развития вычислительной техники не достижимо для молекул, состоящих из достаточно большого количества атомов, существует большое разнообразие приближенных потенциалов, описывающих те или иные энергетические характеристики нанообъектов. Энергия наноструктуры описывается некоторой функцией от координат атомов молекулярной системы. В зависимости от решаемых задач для описания того или иного нанообъекта, может быть выбран тот или иной способ вычисления энергии, каждый из которых имеет свои границы применимости.

Энергетически стабильная геометрия наноструктуры должна соответствовать минимуму её потенциальной энергии. В силу различия способов описания энергии наноструктур, оптимальная геометрия одной и той же наноструктуры при описании разными потенциалами может существенно различаться.

Для определения оптимальной геометрии наноструктуры в рамках того или иного потенциала, описывающего энергию наноструктуры, решается задача поиска минимума потенциальной энергии. С теоретической точки зрения наиболее энергетически выгодная геометрия наноструктуры должна соответствовать глобальному минимуму её потенциальной энергии. Однако решение такой задачи для большинства систем имеет колossalную вычислительную сложность, поэтому на практике обычно используются более простые методы, которые определяют локальный минимум потенциальной энергии.

Потенциал REBO применяется для описания энергии взаимодействия углеводородов. При помощи данного потенциала можно моделировать процессы протекания химических реакций, процессы, происходящие при механических деформациях, исследовать стабильность композитных материалов и др. Ознакомиться с подробным описанием данного потенциала можно в разделе 1.3.

Потенциал AIREBO расширяет возможности потенциала REBO путем добавления слагаемых, отвечающих за ван-дер-ваальсово и торсионное взаимодействия. Эти изменения необходимы для моделирования межмолекулярного взаимодействия, кроме того, добавление слагаемого, описывающего торсионное взаимодействие, обеспечивает более точное моделирование механических свойств объектов,

имеющих гексагональную структуру. Ознакомиться с подробным описанием данного потенциала можно в разделе 1.4.

При моделировании периодических структур полезным может оказаться использование периодических граничных условий. Использование периодических граничных условий означает, что каждый атом молекулярной системы, помимо своего фактического положения, одновременно находится во всех точках пространства, которые получаются при сдвиге его координаты на векторы трансляции. Такой подход позволяет избежать наличия граничных эффектов, связанных с появлением ненасыщенных химических связей на границах моделируемых объектов. Подробнее о периодических граничных условиях можно прочитать в разделе 2.2.2.

Информация о формате конфигурационного файла для производства оптимизации геометрии наноструктур в рамках программного пакета Kvazar изложена в разделе 2.2.3.

Практическая часть

1. Произвести оптимизацию геометрии колонного графена, полученного в ходе выполнения **Лабораторной работы №1** в рамках потенциала REBO;
2. Произвести оптимизацию геометрии колонного графена, полученного в ходе выполнения **Лабораторной работы №1** в рамках потенциала AIREBO;
3. Выделить элементарную ячейку колонного графена и произвести её оптимизацию в методе REBO при использовании периодических граничных условий;
4. Выделить элементарную ячейку колонного графена и произвести её оптимизацию в методе AIREBO при использовании периодических граничных условий;
5. Произвести сравнение стартовой геометрии колонного графена и геометрии, полученной в результате оптимизации методами REBO/AIREBO, в том числе при использовании периодических граничных условий. Сравнить между собой результаты оптимизации геометрии в методах REBO и AIREBO.;

3.3 Лабораторная работа №3

"Моделирование динамики растяжения элементарной ячейки колонного графена"

Цель работы: овладение навыками моделирования процессов методом молекулярной динамики в рамках программного пакета Kvazar;

Задачи работы:

- ознакомиться с принципами метода молекулярной динамики;
- ознакомиться с понятием термостата, применительно к методу молекулярной динамики;
- произвести моделирование динамики колонного графена при заданной температуре;
- произвести моделирование растяжения элементарной ячейки колонного графена до момента разрыва, используя динамические периодические граничные условия;

Теоретическая часть Метод молекулярной динамики позволяет моделировать динамику процессов, происходящих на атомном уровне. Данный метод получил широкое распространение, когда появилась возможность численно решать уравнения движения Ньютона при помощи ЭВМ. В настоящее время границы применимости метода молекулярной динамики постоянно расширяются в силу роста вычислительных возможностей ЭВМ. С теоретической стороной метода молекулярной динамики можно ознакомиться в разделе 1.1.

Существует большое количество алгоритмов, позволяющих расширить границы применимости метода молекулярной динамики, в том числе алгоритмы термостатов, алгоритм списков Верле, алгоритм связанных ячеек, статические и динамические ограничения на подвижность атомов системы и др. Подробнее с некоторыми из перечисленных алгоритмов можно ознакомиться в разделе 2.2.2. При использовании динамических периодических граничных условий при моделировании динамики периодических структур, могут быть смоделированы процессы растяжения или сжатия элементарных ячеек соответствующих структур, что в свою очередь, может быть использовано для определения их механических и даже прочностных характеристик.

Информация о формате конфигурационного файла для расчета молекулярной динамики в рамках программного пакета Kvazar изложена в разделе 2.2.3. Для визуализации молекулярной динамики можно использовать графический модуль постпроцессинга программного пакета Kvazar (см. раздел 2.3.2).

Практическая часть

1. Произвести динамику движения атомов элементарной ячейки колонного графена в периодических граничных условиях в термостате Нозе-Гувера при температуре 300 Кельвин в течение 10 пикосекунд. При задании начальных условий необходимо определить стартовые скорости атомов системы в соответствии с распределением Максвелла. Молекулярная динамика должна производиться в рамках потенциала REBO. В процессе динамики должны сохраняться данные о температуре системы, кинетической и потенциальной энергиях системы.
2. Построить графики зависимости от времени температуры, кинетической и потенциальной энергиях системы;
3. Произвести динамику растяжения элементарной ячейки колонного графена путем задания скорости движения стенок периодической коробки вдоль произвольной оси. Скорость движения стенок коробки должна составлять 50 м/с. Растяжение необходимо производить до момента разрушения элементарной ячейки;
4. На основе данных о моделировании динамики растяжения элементарной ячейки колонного графена определить величину критической деформации элементарной ячейки колонного графена. На основе данных о динамике полной энергии атомов элементарной ячейки в процессе растяжения оценить модуль Юнга и критическое напряжение для элементарной ячейки колонного графена;

3.4 Лабораторная работа №4

"Изучение электронных свойств колонного графена полуэмпирическим методом сильной связи"

Цель работы: овладение навыками вычисления электронных свойств нанокластеров в программном пакете Kvazar.

Задачи работы:

- ознакомиться с теоретическими аспектами метода сильной связи;
- произвести вычисление энергетических характеристик колонного графена методом сильной связи;
- произвести вычисление распределения зарядов на атомах колонного графена;

Теоретическая часть

Метод сильной связи позволяет вычислять электронную структуру нанокластеров. Сущность метода заключается в численном решении стационарного уравнения Шредингера в рамках подхода ЛКАО, согласно которому молекулярная орбиталь описывается в виде линейной комбинации орбиталей изолированных атомов. Информация об особенностях реализации метода сильной связи в программном пакете Kvazar изложена в разделе 1.2.

В результате решения стационарного уравнения Шредингера определяется энергетический спектр нанокластера. Энергетический спектр является важной характеристикой нанокластера, которая может быть использована в молекулярном спектральном анализе, лазерной спектроскопии, квантовой электронике, оптике и т. д. Согласно принципу Паули, каждая электронная орбиталь может быть занята двумя электронами с противоположно направленными спинами. При описании нанокластеров в основном состоянии предполагается, что все валентные электроны системы занимают область энергетического спектра с наименьшей энергией (что соответствует валентной зоне для периодических структур). Последний заполненный энергетический уровень называется – HOMO (highest occupied molecular orbital), первый не заполненный энергетический уровень называется LUMO (lowest unoccupied molecular orbital). Разница между величиной энергии HOMO и LUMO уровней называется энергетической щелью (что соответствует запрещенной зоне для периодических структур). Для наглядной визуализации энергетических характеристик нанокластеров часто используют построение так называемой плотности электронных состояний (DOS -Density of States), т.е. зависимости доли электронных состояний от величины энергии. Плотность электронных состояний для заданной величины энергии определяется как сумма вкладов каждого энергетического уровня, при этом каждый энергетический уровень описывается функцией

Гаусса, для которой медиана равна собственно величине энергетического уровня. В рамках метода сильной связи также может быть рассчитано распределение зарядов на атомах при помощи процедуры, предложенной Маллигеном [24].

Для вычисления энергетических характеристик нанокластеров в рамках метода сильной связи в программном пакете Kvazar необходимо воспользоваться вычислительным режимом SINGLE_POINT (см. раздел 2.2.3). Для отображения DOS или распределения зарядов на атомах нанокластера можно воспользоваться функционалом графического модуля Kvazar (раздел 2.3.1).

Практическая часть

1. Вычислить электронные характеристики колонного графена в рамках метода сильной связи в программном пакете Kvazar;
2. Определить HOMO и LUMO уровни колонного графена, вычислить величину энергетической щели;
3. Построить график зависимости плотности электронных состояний от энергии для колонного графена;
4. Визуализировать распределение зарядов на атомах колонного графена, определить участки атомного каркаса колонного графена, имеющие избыточный электронный заряд. Определить атомы колонного графена, имеющие наибольшую величину положительного и отрицательного электрического заряда. Сделать выводы о влиянии кривизны поверхности и дефектов атомной сетки на перераспределение зарядов.

Список литературы

1. Кобельков Г. М. Бахвалов Н. С., Жидков Н. П. *Численные методы*. М.: Бином, 2001.
2. L. Verlet. Computer ‘experiments’ on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159(98):103, 1967.
3. J. B. Gibson, A. N. Goland, M. Milgram, and G. H. Vineyard. Dynamics of radiation damage. *Phys. Rev.*, 120:1229–1253, Nov 1960.

4. A. Nordsieck. On numerical integration of ordinary differential equations. *Math. Comput.*, 16:22, 1962.
5. M.P. Allen and A.K. Tildesley. *Computer Simulation of Liquids*. Oxford: Clarendon Press., 1987.
6. Лифшиц Е. М. Ландау, Л. Д. *Квантовая механика (нерелятивистская теория)*. — Издание 6-е, исправленное. М.: Физматлит, 2004.
7. У. Харрисон. *Электронная структура и свойства твердых тел*. М.: Мир, 1983.
8. L Goodwin. A new tight binding parameterization for carbon. *Journal of Physics: Condensed Matter*, 3(22):3869, 1991.
9. G. F. Slater, J. C.; Koster. Simplified lcao method for the periodic potential problem. *Physical Review*, 94(6):1498–1524, 1954.
10. J. Tersoff. Modeling solid-state chemistry: Interatomic potentials for multicomponent systems. *Phys. Rev. B*, 39(8):5566–5568, 1989.
11. D.W. Brenner. Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films. *Phys. Rev. B*, 42(15):9458–9471, 1990.
12. Donald W Brenner, Olga A Shenderova, Judith A Harrison, Steven J Stuart, Boris Ni, and Susan B Sinnott. A second-generation reactive empirical bond order (rebo) potential energy expression for hydrocarbons. *Journal of Physics: Condensed Matter*, 14(4):783, 2002.
13. Steven J. Stuart, Alan B. Tutein, and Judith A. Harrison. A reactive potential for hydrocarbons with intermolecular interactions. *Journal of Chemical Physics*, 112(14):6472–6286, 2000.
14. J. Elliott. *Molecular Dynamics Method*. Cambridge, 2002.
15. J. Siewert, H. Marrink, J. Risselada, S. Yefimov, D.P. Tieleman, and A.H. de Vries. The martini force field: Coarse grained model for biomolecular simulations. *Phys. Chem. B.*, 111:7812–7824, 2007.

16. Bryan O’Sullivan. *Mercurial: The Definitive Guide*, volume 284. O'Reilly Media, 2009.
17. Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.
18. Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2015-09-11].
19. Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.
20. L Dalcin. Mpi for python v1.3 documentation, 05 2012.
21. David Abrahams and Ralf W. Grosse-Kunstleve. Building hybrid systems with boost.python. *C/C++ Users Journal*, 2003.
22. Opengl context documentation.
23. J Bodnar. PyQt4 tutorial, 9 2011.
24. R. S. Mulliken. Electronic population analysis on lcao mo molecular wave functions. i. *The Journal of Chemical Physics*, 23(10):1833–1840, 1955.