

Министерство образования и науки Российской Федерации
Южно-Уральский государственный университет
Институт естественных и точных наук
Кафедра компьютерного моделирования и нанотехнологий

539.2(07)
С587

С.А. Созыкин, А.Н. Соболев

ФИЗИКА НАНОРАЗМЕРНЫХ СИСТЕМ

Учебное пособие

Часть 1

Челябинск
Издательский центр ЮУрГУ
2016

УДК 539.219.3(075.8)
С587

*Одобрено
учебно-методической комиссией
института естественных и точных наук*

*Рецензенты:
П.С. Попель, А.Е. Майер*

Созыкин, С.А.

С587 Физика наноразмерных систем: учебное пособие / С.А. Созыкин, А.Н. Соколов. – Челябинск: Издательский центр ЮУрГУ, 2016. Ч. 1. –113 с.

Пособие предназначено для изучения возможностей, предоставляемых квантово-химическим пакетом SIESTA, для моделирования физико-химических свойств материалов. Первая часть пособия посвящена описанию методов, реализованных в пакете SIESTA и параметров входного файла, позволяющих настраивать эти методы для получения более точных результатов. Во второй части рассматриваются возможности программ «Пре- и постпроцессор для моделирования свободных и деформированных углеродных нанотрубок с помощью пакетов Firefly и Siesta» и XCrysDen по сопровождению расчетов с помощью пакета SIESTA. В тексте пособия приведены задания для выполнения самостоятельного численного исследования свойств углеродных нанотрубок. Подобраны разноплановые задания, которые позволяют максимально полно исследовать объект моделирования.

Пособие относится к курсу «Физика наноразмерных систем» в рамках вариативной части основной образовательной программы по направлению 03.04.01 «Прикладные математика и физика».

УДК 539.219.3(075.8)

© Издательский центр ЮУрГУ, 2016

Оглавление

Введение	4
1. Начало работы с пакетом SIESTA	
1.1. Что такое SIESTA?	6
1.2. Установка пакета SIESTA	8
1.3. Входной и выходные файлы	9
1.4. Атомная структура модели	11
1.5. Обменно-корреляционный функционал	16
1.6. Самосогласование	18
1.7. Спин	20
1.8. Приближение псевдопотенциала	21
1.9. Базисный набор	28
1.10. Сетка в координатном пространстве	32
1.11. Оптимизация структуры и молекулярная динамика	39
1.12. Упражнение 1	45
2. Расчет физико-химических свойств	
2.1. Зонная структура	50
2.2. Плотность электронных состояний	55
2.3. Оптические свойства	59
2.4. Теория Ландауэра - Буттикера	64
2.5. TranSIESTA	67
2.6. Вычисление фононных спектров	76
2.7. Упражнение 2	87
3. Вспомогательное программное обеспечение	
3.1. Пре- и постпроцессор для моделирования свободных и деформированных углеродных нанотрубок	88
3.2. Denchar	97
3.3. XCrysden	103
3.4. Упражнение 3	108
Заключение	109
Библиографический список	110

Введение

Учебное пособие разработано в соответствии ФГОС ВО по направлению подготовки 03.04.01 «Прикладная математика и физика» (уровень магистратуры). Целью представленного пособия является формирование у магистрантов ряда общепрофессиональных и профессиональных компетенций, соответствующих научно-исследовательской деятельности:

- способностью применять теорию и методы математики для построения качественных и количественных моделей объектов и процессов в естественнонаучной сфере деятельности (ОПК-2);
- способности использовать на практике углубленные фундаментальные знания, полученные в области естественных и гуманитарных наук, и владением научным мировоззрением (ОПК-3);
- способности применять современные методы анализа, обработки и представления информации в сфере профессиональной деятельности (ОПК-5);
- способности самостоятельно и (или) в составе исследовательской группы разрабатывать, исследовать и применять математические модели для качественного и количественного описания явлений и процессов и (или) разработки новых технических средств (ПК-1);
- способности ставить, формализовать и решать задачи, умением системно анализировать научные проблемы, генерировать новые идеи и создавать новое знание (ПК-2).

Пособие предназначено для изучения возможностей, предоставляемых квантово-химическим пакетом SIESTA, для моделирования физико-химических свойств материалов. Большинство примеров и заданий направлено на изучение углеродных нанотрубок.

Первая часть пособия посвящена описанию методов, реализованных в пакете SIESTA и параметров входного файла, позволяющих настраивать эти методы для получения более точных результатов. Во второй части рассматриваются возможности программ «Пре- и постпроцессор для моделирования свободных и деформированных углеродных нанотрубок с помощью пакетов Firefly и Siesta» и XCrysDen по сопровождению расчетов с помощью пакета SIESTA. В тексте пособия приведены задания для выполнения самостоятельного численного исследования свойств углеродных нанотрубок. Подобраны разноплановые задания, которые позволяют максимально полно исследовать объект моделирования.

Преподавателям рекомендуется кратко ознакомить студентов с теоретической частью и ответить на возникшие вопросы, затем дать студентам несколько заданий для практической работы с постепенным увеличением их сложности. Разный уровень заданий позволяет индивидуально подобрать их в соответствии с уровнем подготовки студента.

Данное пособие относится к курсу «Физика наноразмерных систем» в рамках вариативной части основной образовательной программы по направлению 03.04.01 «Прикладные математика и физика». На курс отводится 3 зачётные единицы.

1. Начало работы с пакетом SIESTA

1.1. Что такое SIESTA?

SIESTA (название расшифровывается как Spanish Initiative for Electronic Simulations with Thousands of Atoms) — название метода и программного пакета, реализующего этот метод, которые позволяют проводить быстрые расчеты электронной структуры и моделирование методом первопринципной молекулярной динамики различных систем в рамках теории функционала электронной плотности. Это означает, что пакет SIESTA способен решать задачи, относящиеся *только* к свойствам системы в основном электронном состоянии, такие как определение уравнения состояния, построение фазовых диаграмм, нахождение фононных спектров. Напротив, для нахождения, например, спектра электронных возбуждений будет лучше использовать программный пакет, основанный на TDDFT — функционале электронной плотности, зависящем от времени.

Как и во всех кодах, основанных на теории функционала плотности, в рамках пакета SIESTA не рассчитывается многоэлектронная волновая функция. Вместо этого метод позволяет получить интегральную электронную плотность. Следовательно, SIESTA не может рассчитать квантовохимические эффекты, связанные с конфигурационным взаимодействием, даже в основном состоянии. Кроме этого, у SIESTA есть проблемы с описанием Ван-дер-Ваальсовых взаимодействий и водородных связей.

SIESTA относится к классу *псевдопотенциальных* методов (см. раздел 1.8). Это означает, что при решении уравнения Кона–Шэма в расчет берутся только валентные электроны, находящиеся на внешних энергетических уровнях. Основные же электроны исключаются из рассмотрения и заменяются псевдопотенциальной функцией. Точности такого приближения обычно достаточно для описания химических связей, равновесной геометрии, фононов и т.д.

Важным отличием пакета SIESTA от других аналогичных программных пакетов является выбор базиса, по которому раскладываются решения уравнения Кона–Шэма (см. раздел 1.9). В самом деле, можно выделить несколько больших групп пакетов: те, которые используют в качестве базиса функции, центрированные на атомах, и использующие наборы независимых от атомов функций (чаще всего — наборы плоских волн).

Кроме этого, существуют гибридные пакеты, соединяющие точность описания электронной плотности вблизи атомов с гибкостью описания межатомного пространства. Выбор используемого базиса — всегда торг между точностью, гибкостью и скоростью расчета. Методы, использующие базис плоских волн, позволяют увеличить точность расчета, изменив единственный параметр (радиус обрезания), но при этом размер базиса и, соответственно, время расчета увеличивается в геометрической прогрессии. Кроме этого, в таких методах время расчета при заданном радиусе обрезания растет с увеличением объема суперячейки (и не зависит от количества атомов), следовательно, расчет пористой структуры или молекулы, помещенной в центр ячейки, может оказаться очень долгим. По другому делу обстоит с методами, использующими атомноцентрированный базис — время расчета в таких методах растет с увеличением количества атомов (и числа базисных функций на этих атомах). Таким образом, если мы в состоянии найти хороший базис, способный точно описать электронную плотность именно там, где это необходимо (например, в области формирования химических связей), то такой расчет получается очень быстрым и точным. Однако, построение такого «хорошего» базиса является в своем роде искусством.

SIESTA относится к пакетам с атомноцентрированным базисом. Это значит, что этот пакет подходит для расчета больших по объему систем, но для систем из большого числа атомов может оказаться необходимым оптимизация базиса для ускорения расчета. Кроме этого, может понадобиться валидация результатов расчета путем их сравнения (на меньшей системе) с результатами методов, использующих плоские волны. Необходимо добавить, что при расчете SIESTA не принимает в расчет симметрию системы: расчет симметричной системы занимает столько же времени, как и неупорядоченной.

Радиальная часть базисных функций в SIESTA может быть практически любой; желательно лишь, чтобы она плавно стремилась к нулю далее некоторого радиуса. Этот радиус обрезания определяется размером базиса; он должен включать вторую либо третью координационные сферы. SIESTA рассчитывает параметры перекрытия различных базисных функций эффективным образом вследствие того, что это перекрытие локализовано в пространстве. Чем меньше перекрытие базисных функций (координационное число каждого атома), тем быстрее проводится расчет системы. Время расчета уменьшается еще и потому, что для системы с низкой симметрией необходимо брать интеграл по меньшему числу k -точек обратного пространства. Напротив, плотноупакованные металлические системы сложны для расчета как за счет сильного перекрытия базисных функций, так и из-за необходимости разбивать зону Бриллюэта

на большое число k -точек. Низкоразмерные системы, описанные в настоящем пособии, являются идеальными для расчета с помощью программного пакета SIESTA, и поэтому мы в дальнейшем будем использовать именно этот пакет.

1.2. Установка пакета SIESTA

Пакет SIESTA распространяется в виде исходных кодов, которые могут быть скомпилированы в Unix подобных системах (или эмуляторе Cygwin в операционных системах семейства Windows). Скачать дистрибутив можно с официального сайта <https://launchpad.net/siesta>.

Приведем последовательность действий по компиляции пакета SIESTA. Командой

```
tar -xzvf siesta-<версия пакета>.tgz
```

можно извлечь исходный код из архива. Далее необходимо перейти в директорию, где будут размещены временные и исполняемый файлы пакета SIESTA:

```
cd siesta-<версия пакета>/Obj,
```

создать структуру директорий, ожидаемую командой *make*

```
sh ../Src/obj_setup.sh
```

и запустить скрипт для подготовки *arch.make* файла, содержащего необходимые для компиляции пакета сведения о расположении компиляторов и библиотек

```
../Src/configure
```

(при компиляции последовательной версии пакета) или

```
../Src/configure --enable-mpi
```

(при компиляции параллельной версии пакета). Если вы увидели сообщения об ошибке, нужно изменить переменные среды или самостоятельно отредактировать файл *arch.make*. Для получения исполняемого файла достаточно выполнить команду *make*. В случае успешной компиляции в текущей директории должен появиться файл *siesta* — единственный исполняемый файл пакета SIESTA.

Начиная с версии 3.0 пакет SIESTA включает возможность вычисления электронной проводимости с использованием метода TranSIESTA. TranSIESTA - это метод получения электронной структуры открытой системы в виде структуры конечной длины, расположенной между двумя полубесконечными электродами. Компиляция пакета TranSIESTA проводится аналогично компиляции SIESTA, но вместо *make* необходимо выполнить


```
make transiesta.
```

Корректность работы пакета SIESTA можно проверить с помощью набора тестов, расположенных в папке Tests. Для запуска определенного теста (например, `sinw`), достаточно перейти в содержащую его директорию (`sinw`) и выполнить `make`. В случае корректной работы SIESTA на экран будет выведен следующий текст:

```
>>>> Running sinw test...
==> Copying pseudopotential file for Si...
==> Copying pseudopotential file for H...
==> Running SIESTA as ../../../../siesta
===> SIESTA finished successfully
```

1.3. Входной и выходные файлы

Входной файл пакета SIESTA, содержащий все физические сведения о моделируемой системе и параметры расчета, имеет формат `fdf` (flexible data format – гибкий формат данных). Этот формат позволяет записывать параметры моделирования в любом порядке, или опускать их описание в пользу значения по умолчанию (если оно определено). Перечислим основные правила записи этого файла.

- В FDF-файле после переменной идет ее значение. Если переменная не указана, она принимается равной значению по умолчанию.
- Имена переменных в FDF-файле нечувствительны к регистру, а символы «-», «_», «.» в именах переменных игнорируются. Таким образом, `LatticeConstant` и `lattice_constant` обозначают одну и ту же переменную.
- Текст, идущий после символа `#`, воспринимается как комментарий.
- Логические значения могут быть записаны как `T`, `true`, `.true.`, `yes`, `F`, `false`, `.false.`, `no`. Отсутствие значения эквивалентно «`true`».
- Строки не нужно заключать в апострофы.
- Действительные значения, соответствующие физическим величинам, должны указываться вместе с идущими за ними единицами измерений. Дробная часть чисел отделяется точкой.
- Составные структуры данных называются блоками (`blocks`) и размещаются между `%block <название блока>` и `%endblock <название блока>`.
- Можно подключать (`%include`) другие FDF-файлы. Если переменная встречается больше одного раза, используется ее первое значение.
- Если имя переменной написано с ошибками, оно не будет интерпретировано (список допустимых переменных отсутствует). После запуска расчета значение всех переменных (включая отсутствующие во входном файле) можно найти в файле `fdf.log`.

Пакет SIESTA записывает лог своей работы в стандартный выходной поток, который обычно перенаправляется (пользователем) в выходной файл.

Сразу после запуска программа выводит версию кода (программы). Затем выводится входной файл FDF (кроме пустых строк). После этого программа считывает псевдопотенциалы и генерирует базисный набор, используемый в вычислениях. Информация об этих этапах отражается в выходном файле. После этого начинается главный (внешний) цикл моделирования. Выводится информация о шаге моделирования молекулярной динамики MD (или оптимизации геометрии по методу сопряженных градиентов CG) и цикле самосогласования SCF в нем.

SIESTA может записывать различные данные в стандартный выходной поток в зависимости от значений параметров, указанных ниже. По умолчанию большинство из них не сохраняется. Параметры выводятся на всех шагах оптимизации геометрии или молекулярной динамики. Для больших систем они могут занимать много места (координаты, собственные значения, силы, и т.д.). В дополнение к стандартному потоку SIESTA выводит информацию в другие файлы.

Задавая для переменной LongOutput значение «true» (значение по умолчанию: «false»), вы изменяете значения по умолчанию для некоторых опций, получая в выходном потоке больше информации. В частности, переопределяются следующие опции:

- WriteKpoints: определяет будут ли записываться в главный выходной файл координаты узлов сетки в k -пространстве;
- WriteKbands: определяет будут ли записываться в главный выходной файл координаты k -векторов, используемых для построения зонной структуры;
- WriteCoorStep: определяет будут ли на каждом шаге моделирования молекулярной динамики или оптимизации структуры выводиться в главный выходной файл координаты атомов;
- WriteForces: определяет будут ли на каждом шаге моделирования выводиться в выходной файл силы, действующие на атомы;
- WriteEigenvalues: если этот параметр равен true, собственные значения гамильтониана выводятся в выходной файл;
- WriteWaveFunctions: определяет будет ли выведен в выходной файл список волновых функций, записанных в файл Systemlabel.selected.WFSX;
- WriteMullikenPop (целочисленная переменная): определяет степень детализированности выводимых в выходной файл результатов анализа заселенности по Милликену:
0: не выводится;

- 1: заселенности атомов и орбиталей;
- 2: 1 + заселенность области перекрытия атомов;
- 3: 2 + заселенность перекрытия орбиталей.

Явное переопределение значения для любого из этих параметров отменяет значение, установленное для него опцией `LongOutput`.

1.4. Атомная структура модели

Информацию о структуре модели можно указать двумя способами: в традиционном формате или в новом формате `Z`-матрицы, используя блок `Zmatrix`. Независимо от способа, которым описана структура, следующие параметры являются обязательными.

- Строковая переменная `SystemName` — название системы (не более 150 символов включая пробелы). Значение по умолчанию: пустая строка.
- Строковая переменная `SystemLabel` — краткое название системы, используемое в названиях выходных файлов (значение по умолчанию: «`siesta`»). Длина строки не должна превышать 20 символов (без пробелов).
- Целочисленный параметр `NumberOfSpecies` задает количество различных типов атомов. Атомы одного химического элемента, но с различными псевдопотенциалами (см. раздел 1.8) или базисными наборами (см. раздел 1.9) считаются разными. Значение по умолчанию отсутствует (его необходимо задать явно).
- Целочисленный параметр `NumberOfAtoms` задает количество атомов в модели. Значение по умолчанию отсутствует (его необходимо задать явно).

Традиционный способ описания в `fdf` файле

При традиционном способе описания размер ячейки должен быть указан с использованием некоторой комбинации вариантов `LatticeConstant`, `LatticeParameters`, `LatticeVectors` и `SuperCell`. Если ничего не указано, SIESTA построит кубическую ячейку, в которой совокупность атомов будет рассматриваться как отдельная молекула. Позиции атомов в ячейке указываются в блоке `AtomicCoordinatesAndAtomicSpecies`.

Обратите внимание на следующие параметры.

Параметр `LatticeConstant` определяет масштаб базисных векторов решетки. Значение по умолчанию: минимальный размер, позволяющий вместить систему (предполагается, что это молекула) без межячеечных

взаимодействий, увеличенный на 10%. Примечание: значение параметра `LatticeConstant` могут быть необходимы для других опций, таких как единицы измерения k -точек, используемых для расчетов зонной структуры.

Блок данных `LatticeParameters` позволяет задать базисные вектора решетки с помощью шести действительных чисел: трех модулей векторов (a , b и c) и трех углов (α (угол между \vec{b} и \vec{c}), β и γ). Модули векторов задаются в единицах `LatticeConstant`, углы – в градусах. Значение по умолчанию: 1.0 1.0 1.0 90. 90. 90.

Блок данных `LatticeVectors` позволяет указать базисные вектора ячейки в единицах `LatticeConstant`. Они считываются как матрица `CELL(ixyz,ivector)`, каждый вектор записывается в строку. Значение по умолчанию:

```
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
```

Если используется значение по умолчанию для параметра `LatticeConstant`, базисные вектора ортогональны, но не обязательно соответствуют кубической решетке.

Блок данных `SuperCell` задает целочисленную матрицу 3×3 в единицах ячейки:

```
%block SuperCell
M(1,1) M(2,1) M(3,1)
M(1,2) M(2,2) M(3,2)
M(1,3) M(2,3) M(3,3)
%endblock SuperCell.
```

Суперячейка определяется как $SuperCell(ix; i) = \sum_j CELL(ix; j)M(j; i)$.

Входная строка соответствует базисному вектору. Блок `SuperCell` игнорируется, если атомная структура считывается из файла `XV` (информация от предыдущих запусков программы). Позиции атомов должны быть заданы только для элементарной ячейки. Они будут автоматически транслированы по всей суперячейке. Переменная `NumberOfAtoms` соответствует числу атомов в элементарной ячейке. В выходном файле все значения приводятся для суперячейки. Все остальные параметры входного файла (за исключением переменной `NumberOfAtoms` и позиций атомов), включая `kgridMonkhorstPack` должны относиться ко всей суперячейке (в старых версиях пакета это было не так). Значение по умолчанию: суперячейка совпадает с элементарной ячейкой.

Строчковая переменная `AtomicCoordinatesFormat` определяет формат записи позиций атомов во входном файле (значение по умолчанию: `NotScaledCartesianBohr`). Допустимы следующие варианты:

- Bohr или NotScaledCartesianBohr (атомные позиции в борах в декартовых координатах);
- Ang или NotScaledCartesianAng (атомные позиции в ангстремах в декартовых координатах);
- ScaledCartesian (атомные позиции в декартовых координатах в единицах постоянной решетки);
- Fractional или ScaledByLatticeVectors (позиции атомов в базисных векторах ячейки).

Строковая переменная `AtomCoorFormatOut` (по умолчанию совпадает с переменной `AtomicCoordinatesFormat`) указывает формат записи позиций атомов в выходном файле. Доступны те же варианты, что и для `AtomicCoordinatesFormat`.

Блок данных `AtomicCoordinatesOrigin` задает вектор, определяющий сдвиг позиций атомов (значение по умолчанию: 0.000 0.000 0.000).

Блок данных `AtomicCoordinatesAndAtomicSpecies` определяет позиции и сорт каждого атома. Информация о каждом атоме приводится в отдельной строке. Считывание проводится в цикле:

```
From ia = 1 to natoms
  read: xa(ix,ia), isa(ia),
```

где $xa(ix, ia)$ — координаты ia -го атома в формате, определенном переменной `AtomicCoordinatesFormat`, $isa(ia)$ — сорт атома ia . Значение по умолчанию отсутствует. Позиции атомов должны быть указаны в этом блоке или в блоке `Zmatrix`.

В качестве примера приведем `fdf` файл для моделирования карбина.

```
1 SystemName      Carbin # название системы
2 SystemLabel    Carbin # короткое название системы
3
4 NumberOfSpecies 1
5 NumberOfAtoms  1
6
7 %block ChemicalSpeciesLabel
8     1 6 C
9 %endblock ChemicalSpeciesLabel
10
11 LatticeConstant 1.27620 Ang
12 %block LatticeParameters
13     5 5 1 90 90 90
14 %endblock LatticeParameters
15
16 AtomicCoordinatesFormat Fractional
17 %block AtomicCoordinatesAndAtomicSpecies
18     0 0 0 1
19 %endblock AtomicCoordinatesAndAtomicSpecies
```

В этом примере модель содержит один атом в элементарной ячейке. Очевидно, что и количество сортов атомов равно единице. Этот атом является углеродом (указан заряд ядра: 6). Соответствующий ему псевдопотенциал должен находиться в файле C.psf. Базисные вектора элементарной ячейки ортогональны. Параметр трансляции вдоль оси z равен 1.27620 Ang, что в пять раз меньше параметров трансляции в двух оставшихся измерениях.

Преимуществом традиционного формата является простота описания системы. Однако, с его помощью сложно работать с системами с ограничениями. Для наложения ограничений лучше подходит формат Z-matrix.

Z-matrix формат

Блок Z-matrix предназначен для ввода информации о атомной структуре системы с использованием формата Z-матрицы, а также контроля параметров оптимизации. Это особенно полезно при работе с молекулярными системами или оптимизации с ограничениями (например, поиск переходных состояний или жестких движений частей системы). Формат также позволяет гибридное использование форматов Z-матрицы и декартовых или дробных блоков (это удобно для изучения молекул на поверхности).

Приведем пример описания молекулы воды в формате Z-матрицы:

```

1 %block Zmatrix
2 molecule fractional
3 1 0 0 0 0.0 0.0 0.0 0 0 0
4 2 1 0 0 HO1 90.0 37.743919 1 0 0
5 2 1 2 0 HO2 HOH 90.0 1 1 0
6 variables
7 HO1 0.956997
8 HO2 0.956997
9 HOH 104.4
10 %endblock Zmatrix.
```

Перечислим разделы, которые могут быть использованы в блоке Z-matrix.

В разделе molecule указываются атомы, которые составляют молекулу и ее геометрия. для уточнения способа задания координат атомов могут быть указаны опции «fractional» или «scaled». При отсутствии этих ключевых слов предполагается, что расстояния приведены в единицах ZM.UnitsLength.

В каждой строке описывается один атом в следующем формате:

Nspecies i j k r α t ifr ifa ift

Значения *Nspecies*, *i*, *j*, *k*, *ifr*, *ifa*, и *ift* — целые числа, а *r*, *α*, и *t* —

действительные числа двойной точности.

Для большинства атомов, $Nspecies$ — номер сорта атома, r — расстояние до атома с номером i , α — угол, образованный данным атомом и атомами с номерами j и i , t — торсионный угол, образованный данным атомом и атомами с номерами k , j , и i . Значения ifr , ifa и ift — целочисленные флаги, показывающие нужно ли оптимизировать r , α , и t , соответственно (0 для фиксированного значения, 1 для оптимизации).

Информация о первых трех атомах в молекуле вводится в особом формате (недостаточно атомов, чтобы указать определенные расстояния и углы). Для первого атома: r , α , и t — декартовы координаты атома. Для второго атома: r , α , и t — координаты второго атома в сферической системе координат, начало отсчета которой совпадает с первым атомом. r — радиус, α — полярный угол (угол между осью z и радиус-вектором). t — азимутальный угол (угол между осью x и проекцией радиус-вектора на плоскость xy). Для третьего атома: параметры имеют стандартное значение, но торсионный угол определен по отношению к воображаемому атому в z -направлении над атомом j .

В разделе *cartesian* определяется блок атомов, координаты которых должны быть указаны в декартовой системе координат. Может быть указано ключевое слово *fractional* или *scaled* (при отсутствии берется значение `ZM.UnitsLength`).

Формат ввода для каждого атома блока следующий:

$Nspecies \ x \ y \ z \ ix \ iy \ iz,$

где $Nspecies$, ix , iy , и iz — целочисленные, x , y , z — действительные переменные. $Nspecies$ — номер типа атома, x , y , и z — декартовы координаты атома. Величины ix , iy и iz — целочисленные флаги, показывающие нужно ли варьировать соответствующие координаты (0 — не варьировать, 1 — варьировать).

Блок может содержать перечисленные ниже разделы, облегчающие его чтение.

При описании геометрии модели можно использовать символьные константы, описанные в разделе *constants*. Формат объявления прост: за символьной переменной следует значение:

НОН 104.4.

Секция *variables* позволяет объявить символьные переменные для описания геометрии:

НО1 0.956997.

Раздел *constraint* позволяет пользователю задать связи между использованными в Z -матрице переменными и константами:

constraint

var1 var2 A B,

где $var1$ и $var2$ – величины в определении Z -матрицы, A и B – действительные числа. Переменные связаны соотношением $var1 = A \cdot var2 + B$.

Параметр `zm.UnitsLength` определяет единицу измерения длин в Z -матрице: Bohr или Angstroms (значение по умолчанию: Bohr).

Параметр `zm.UnitsAngle` определяет единицы измерения углов в Z -матрице: radians или degrees (значение по умолчанию: rad).

При описании атомной структуры очень просто допустить ошибку. Эти ошибки могут не привести к аварийному завершению расчета SIESTA, но полученные в этом случае результаты будут неверными. Для контроля корректности атомной структуры можно визуализировать *.XV файл (например, при помощи программы XCrysden (см. стр. 103)).

1.5. Обменно-корреляционный функционал

Современные расчеты свойств молекул по теории функционала электронной плотности основаны на приближении Кона-Шэма, обоснованного двумя теоремами [1, 2]. Первая из них утверждает, что все свойства молекулы в основном состоянии могут быть определены по ее электронной плотности $\rho_0(x, y, z)$. Иными словами, по данной функции $\rho_0(x, y, z)$ возможно рассчитать любую характеристику этой молекулы. Например, энергию основного состояния E_0 можно представить в виде функционала

$$E_0 = E_0[\rho_0(x, y, z)].$$

Вторая теорема Хоэнберга и Кона утверждает, что всякая пробная электронная плотность соответствует энергии, не меньшей чем истинная энергия основного состояния. В молекулярных расчетах, основанных на теории функционала плотности, электронная энергия – это энергия электронов, движущихся в потенциале неподвижных атомных ядер. Этот ядерный потенциал называют «внешним».

Согласно теореме Хоэнберга и Кона, электронная энергия молекулы может быть записана как сумма

$$E[\rho] = E_t[\rho] + E_v[\rho] + E_j[\rho] + E_{xc}[\rho],$$

где E_t – кинетическая энергия электронов, E_v – потенциальная энергия электрон-ядерного притяжения и отталкивания ядер, E_j – энергия отталкивания электрона от средней электронной плотности, созданной всеми остальными электронами системы, а E_{xc} – обменно-корреляционная поправка. Член E_{xc} состоит из двух вкладов, отвечающих собственно обменной и корреляционной энергиям:

$$E_{xc}[\rho] = E_x[\rho] + E_c[\rho],$$

причем, обычно $|E_x[\rho]|$ много больше, чем $|E_c[\rho]|$. Например, для атома аргона $E_x[\rho] = -30.19$ Хартри, в то время как $E_c[\rho] = -0.72$ Хартри.

Следует подчеркнуть, что упомянутые теоремы лишь доказывают зависимость энергии от электронной плотности, но не предлагают никакого способа поиска вида обменно-корреляционного функционала.

Приближение локальной электронной плотности (LDA)

Наиболее простым приближением для $E_{xc}[\rho(r)]$ является приближение локальной электронной плотности. В основе этого подхода лежит предположение, что электронная плотность $\rho(r)$ очень слабо зависит от пространственных координат. Термин «локальная» используется потому, что в определении функционала входит значение величины электронной плотности $\rho(r)$ в каждой точке пространства (и не входят производные от электронной плотности).

Приближение локальной спиновой плотности (LSDA)

Неограниченный LDA метод называют приближением локальной спиновой плотности LSDA. Он получается, если в модели LDA предположить, что электронам со спинами α и β в однородном электронном газе соответствуют различные орбитали ψ_α^{KS} и ψ_β^{KS} . Это приводит к различным плотностям ρ^α и ρ^β для электронов с противоположными спинами. Преимуществом данного подхода является то, что его можно использовать для систем с одним или несколькими неспаренными электронами. Такая ситуация возникает, например, в радикалах. Для систем, в которых все электроны спарены приближение, LSDA эквивалентно LDA.

Приближение обобщенных градиентов (GGA)

Несмотря на свою простоту, метод LDA часто дает весьма удовлетворительное согласие с экспериментальными результатами. Однако в ряде случаев он оказывается неэффективным. Естественным способом его совершенствования является градиентная коррекция функционала (приближение обобщенного градиента – GGA), где обменно-корреляционный функционал E_{xc} зависит не только от электронной плотности, но и от ее градиента. Такие функционалы называют нелокальными.

Перечислим параметры, позволяющие описать используемый обменно-корреляционный функционал.

Строковая переменная `xc.functional` задает тип обменно-корреляционного функционала. Значениями этой переменной могут

быть LDA или GGA. Значение по умолчанию: LDA.

Строковая переменная `XC.authors` задает параметризацию обменно-корреляционного функционала (значение по умолчанию: PZ). Перечислим допустимые значения.

- CA (Ceperley-Alder) совпадает с PZ (Perdew-Zunger) [3]. LDA.
- PW92 (Perdew-Wang-92) [4]. LDA.
- PBE (Perdew-Burke-Ernzerhof) [5]. GGA.
- revPBE (усовершенствованный Perdew-Burke-Ernzerhof) [6]. GGA.
- RPBE (усовершенствованный Perdew-Burke-Ernzerhof) [7]. GGA.
- WC (Wu-Cohen модификация PBE функционала) [8]. GGA.
- PBEsol (Perdew-Burke-Ernzerhof для твердых тел) [9]. GGA.
- LYP. GGA. Объединяет обменный функционал Becke ([10]) и корреляционный функционал Lee, Yang и Parr ([11]). Модифицирован Miehlich, Savin, Stoll и Preuss [12].

Блок данных `XC.hybrid` позволяет пользователю создавать «смешанные» функционалы путем объединения (в определенных пропорциях) доступных в SIESTA функционалов (`XC.authors`). Стоит отметить, что эти функционалы не содержат обменного вклада Хартри-Фока, который должен быть в «гибридных» функционалах.

Первая строка блока должна содержать число смешиваемых функционалов. Последующие строки содержат значения `XC.functional`, `XC.authors`, «веса» для обменной и корреляционной частей. Если задано только одно число, то оно будет использовано и для обменной, и для корреляционной частей функционалов. Следующий пример показывает, как создать функционал, в котором корреляционная часть получается сложением 75% корреляционной части функционала CA и 25% PBE, а обменная часть получается сложением 50% обменных частей упомянутых функционалов:

```
%block XC.hybrid
2
LDA CA 0.5 0.75
GGA PBE 0.5 0.25
%endblock XC.hybrid
```

Значение по умолчанию: отсутствует.

1.6. Самосогласование

Система уравнений метода функционала плотности, описывающая атомную структуру в приближении Кона-Шэма, должна удовлетворять самосогласованным распределениям плотности электронов. Ввиду существенной нелинейности этой системы единственным общим методом по-

строения самосогласованного решения является итеративный способ, при котором надо поочередно решать уравнение Пуассона для потенциала и систему уравнений Шредингера для одночастичных волновых функций в эффективном потенциале. В вычислительной физике конденсированных сред функционал энергии Харриса является самосогласованным приближением к теории функционала плотности Кона-Шэма [13]. Этот подход определяет энергию комбинированной системы в виде функции электронных плотностей отдельных областей. Энергия при использовании функционала Харриса изменяется при удалении системы от равновесной электронной плотности гораздо меньше, чем при использовании функционала Кона-Шэма. Таким образом, во многих системах точность без процедуры самосогласования может оказаться достаточной.

Перечислим основные параметры процедуры самосогласования.

Логическая переменная `Harris_functional` позволяет выбрать способ расчета энергии и сил: самосогласованный с использованием функционала Кона-Шэма или не самосогласованный с функционалом Харриса.

- `false`: функционал Кона-Шэма.
- `true`: функционал Харриса. Этот вариант сокращает время расчета, но для некоторых систем приводит к ошибкам. Силы вычисляются на первом шаге цикла самосогласования. Этот вариант может использоваться только для обменно-корреляционного функционала в параметризации Perdew-Zunger. Если этот параметр равен `true`, значения `DM.UseSaveDM`, `MaxSCFIterations`, `SCFMustConverge` и `DM.MixSCF1` автоматически меняются на `False`, `1`, `False` и `False`, соответственно.

Значение по умолчанию: `false`.

Целочисленный параметр `MaxSCFIterations` задает максимально число итераций самосогласования на каждом шаге оптимизации геометрии или молекулярной динамики. Значение по умолчанию: `50`.

Логическая переменная `SCFMustConverge` определяет поведение программы в случае, если критерий сходимости не был достигнут за максимальное число итераций процедуры самосогласования. По умолчанию обновляются силы и проводится следующий шаг оптимизации геометрии или молекулярной динамики. Если значение переменной равно `true`, расчет прекращается. Значение по умолчанию: `false`.

Действительный параметр `DM.MixingWeight` задает вес α матрицы плотности, полученной на n -ой итерации самосогласования в матрице плотности для $n + 1$ итерации:

$$\rho_{in}^{n+1} = \alpha \rho_{out}^n + (1 - \alpha) \rho_{in}^n.$$

Значение по умолчанию: `0.25`.

Целочисленный параметр `DM.NumberPulay` определяет значение параметра экстраполяционного метода Pulay, который улучшает сходимость и позволяет достигать критерия сходимости даже в тех случаях, когда подход линейного смешения не сходится. Начальное приближение для $n + 1$ итерации конструируется по известным начальным и результирующим матрицам `DM.NumberPulay` предшествующих итераций самосогласования по следующей формуле:

$$\rho_{in}^{n+1} = \alpha \bar{\rho}_{out}^n + (1 - \alpha \bar{\rho}_{in}^n),$$

где $\bar{\rho}_{out}^n$ и $\bar{\rho}_{in}^n$ получены на основе данных предыдущих $N = \text{DM.NumberPulay}$ итераций:

$$\bar{\rho}_{out}^n = \sum_{i=1}^N \beta_i \rho_{out}^{(n-N+i)};$$

$$\bar{\rho}_{in}^n = \sum_{i=1}^N \beta_i \rho_{in}^{(n-N+i)}.$$

Значение β_i получается минимизацией разности между $\bar{\rho}_{out}^n$ и $\bar{\rho}_{in}^n$. Значение α равно переменной `DM.MixingWeight`. Если `DM.NumberPulay` равно 0 или 1, применяется линейное смешение. Значение по умолчанию: 0.

1.7. Спин

Способ учета спина электрона в процессе расчета позволяют задать следующие пять переменных.

`SpinPolarized` — логическая переменная для выбора неполяризованного (`false`) и поляризованного (`true`) по спину расчета. Значение по умолчанию: `false`.

`NonCollinearSpin` (логическая переменная). Если эта переменная равна `true`, расчет проводится с неколлинеарным спином. Это достигается использованием спинорных волновых функций и (2×2) матриц спиновой плотности в каждой точке сетки [14, 15, 16]. Неколлинеарный расчет невозможен, если было выбрано распараллеливание расчета по точкам обратного пространства (`Diag.ParallelOverK`). Значение по умолчанию: `false`.

`FixSpin` (логическая переменная). Если эта переменная равна `true`, расчет проводится с фиксированным значением полного спина системы, заданного переменной `TotalSpin`. Эта опция может быть использована

только при расчете с коллинеарным спином. Значение по умолчанию: false.

`TotalSpin` (действительное значение). Спин системы (в единицах спина электрона, 1/2). Параметр используется только если `FixSpin = True`. Значение по умолчанию: 0.0.

`SingleExcitation` (логическая переменная). Если эта переменная равна true, SIESTA вычисляет очень грубое приближение к низшему возбужденному состоянию, изменяя заселенности НОМО (наивысшего занятого) и LUMO (наинизшего незанятого) уровней. Значение по умолчанию: false.

1.8. Приближение псевдопотенциала

Для описания взаимодействия электронов и иона SIESTA использует псевдопотенциалы (также, как и большинство программных кодов, использующих плоские волны в качестве базисных функций; это их отличает от так называемых полноэлектронных программ).

Остовные электроны относительно устойчивы к воздействию окружения атома. Можно считать, что их (большой) вклад в энергию не меняется при формировании молекулы или кристалла из изолированных атомов. Изменение энергии при химических превращениях обусловлено валентными электронами. Изменение энергии валентных электронов гораздо проще рассчитать точно, если вычесть из энергии связи энергию электронов остова. Кроме того, сильный кулоновский потенциал ядра и локализованные волновые функции остовных электронов трудно описать численно.

Волновые функции электронов изолированного атома являются собственными функциями атомного гамильтониана, поэтому они взаимно ортогональны. Поскольку остовные состояния локализованы вблизи ядра, волновые функции валентных электронов быстро осциллируют в данной области, чтобы удовлетворить требованию ортогональности с волновыми функциями электронов остова. Это приводит к большой величине кинетической энергии валентных электронов в области ядра, которая почти уравнивает потенциальную энергию в сильном кулоновском потенциале. Таким образом, валентные электроны гораздо слабее связаны с ядром, чем электроны остова.

Расчеты упрощаются, если заменить сильный кулоновский потенциал $-\frac{Z_{eff}}{r}$ и остовные электроны эффективным псевдопотенциалом V_{ps} , а волновые функции валентных электронов ψ — псевдо-волновыми функциями ψ_{ps} , которые плавно изменяются в области остова [17, 18] (см. рис.

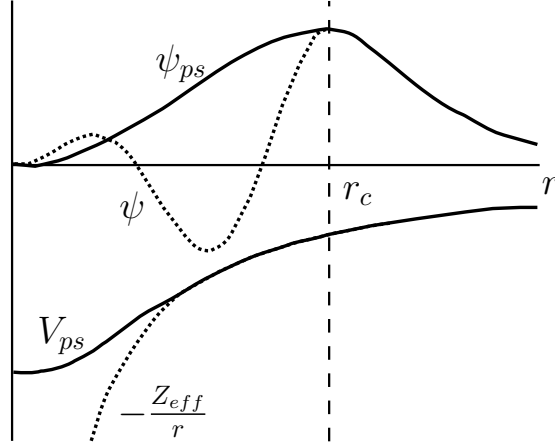


Рис. 1.1.: Связь между истинными и псевдопотенциальными потенциалами и волновой функцией

1.1).

Обратимся к подходу ортогонализированных плоских волн [19]. Рассмотрим атом с гамильтонианом \hat{H} , состояния остовных электронов $\{|\chi_n\rangle\}$, соответствующие им собственные значения энергии $\{E_n\}$ и валентное состояние $|\psi\rangle$ с собственным значением энергии E . Из этих состояний можно построить гладкое псевдо-состояние $|\varphi\rangle$:

$$|\psi\rangle = |\varphi\rangle + \sum_n^{core} a_n |\chi_n\rangle.$$

Валентные состояния должны быть ортогональны всем остовным состояниям (которые, конечно, взаимно ортогональны), так что выражения

$$\langle \chi_m | \psi \rangle = 0 = \langle \chi_m | \varphi \rangle + a_m$$

определяют коэффициенты разложения $\{a_n\}$. Таким образом,

$$|\psi\rangle = |\varphi\rangle - \sum_n^{core} |\chi_n\rangle \langle \chi_n | \varphi \rangle.$$

Подставляя эти выражения в уравнение Шредингера

$$\hat{H}|\psi\rangle = E|\psi\rangle,$$

получим

$$\hat{H}|\varphi\rangle - \sum_n^{core} E_n |\chi_n\rangle \langle \chi_n | \varphi \rangle = E|\varphi\rangle - E \sum_n^{core} |\chi_n\rangle \langle \chi_n | \varphi \rangle.$$

Последнее выражение можно переписать в виде

$$\hat{H}|\varphi\rangle + \sum_n^{\text{core}} (E - E_n) |\chi_n\rangle \langle \chi_n | \varphi\rangle = E|\varphi\rangle.$$

Таким образом, псевдо-состояние удовлетворяет уравнению Шредингера с дополнительным нелокальным потенциалом \hat{V}_{nl} :

$$\left[\hat{H} + \hat{V}_{\text{nl}} \right] |\varphi\rangle = E|\varphi\rangle,$$

$$\hat{V}_{\text{nl}} = \sum_n^{\text{core}} (E - E_n) |\chi_n\rangle \langle \chi_n|.$$

Энергия состояния, описываемого псевдо-волновой функцией, такая же как у исходного валентного состояния. Дополнительный потенциал V_{nl} , который локализован на ядре, является отталкивающим и экранирует часть сильного кулоновского потенциала, так что в результате суммирования получается слабый псевдопотенциал. Как только атом начинает взаимодействовать с окружением, энергии собственных состояний изменяются. Однако, если остовные состояния отстоят от валентных достаточно далеко по энергии (т.е. $\delta E \ll E - E_n$), то разумным приближением является приравнение E в V_{nl} к собственному значению энергии валентного электрона. Псевдопотенциал должен быть подобен реальному потенциалу в первом порядке по E . Этого можно добиться путем построения сохраняющих норму псевдопотенциалов.

Рассмотрим плоскую волну с волновым вектором \vec{k} рассеяния от некоторого сферически-симметричного потенциала локализованного внутри сферы радиуса r_c с центром в начале координат. Плоские волны можно разложить по сферическим волнам

$$\exp(i\vec{k}\vec{r}) = 4\pi \sum_{l=0}^{\infty} \sum_{m=-l}^l i^l j_l(kr) Y_{lm}^*(\vec{e}_{\vec{k}}) Y_{lm}(\vec{e}_{\vec{r}}),$$

где $\vec{e}_{\vec{k}}$ обозначает единичный вектор в направлении вектора \vec{k} . Эти сферические волны затем упруго рассеиваются на потенциале, который вводит сдвиг фаз δ_l , который связан с логарифмической производной точного радиального решения для данного l и энергией $E = \frac{1}{2}k^2$ в пределах остова, вычисленными на границе остова

$$L_l(E) = \left[\frac{d}{dr} \lg [R_l(r, E)] \right]_{r=r_c} = \frac{R_l'(r_c, E)}{R_l(r_c, E)} = k \frac{j_l'(kr_c) - \text{tg}(\delta_l) n_l'(kr_c)}{j_l(kr_c) - \text{tg}(\delta_l) n_l(kr_c)}.$$

j_l и n_l обозначают сферические функции Бесселя и фон Неймана, соответственно.

Радиальная составляющая волновой функции $R_l(r, E)$ связана с решением уравнения Шредингера в области остова с моментом импульса, определенным квантовыми числами l и m , и энергией E :

$$\psi_{lm}(r, E) = R_l(r, E)Y_{lm}(\vec{e}_r).$$

Сферические волны со сдвигом фазы могут рекомбинировать с образованием рассеянной волны. Можно определить сдвиг фаз η_l

$$\delta_l = n_l\pi + \eta_l,$$

который оказывает такое же влияние (амплитуда рассеяния зависит от $\exp(2i\delta_l)$, поэтому множитель π в δ_l ни на что не влияет) и определить n_l требуя, чтобы параметр η_l лежал в интервале $0 \leq \eta_l \leq \pi$. Целочисленный параметр n_l равен числу узлов $R_l(r, E)$, два в случае рисунка 1.1, и равен числу остовных состояний с моментом импульса l .

Псевдопотенциал определяется как потенциал, у которого полный сдвиг фаз равен η_l , поэтому радиальная псевдо-волновая функция не имеет узлов и, следовательно, потенциал не имеет остовных состояний. Рассеяние на псевдопотенциале такое же как и на исходном потенциале.

Псевдопотенциал можно считать хорошим, если он позволяет воспроизвести логарифмическую производной волновой функции для изолированного атома. Заменив полный потенциал псевдопотенциалом, мы можем еще раз решить уравнение Шредингера в области остова, чтобы получить радиальные составляющие псевдо-волновой функции $R_{ps,l}(r, E)$.

Рассмотрим методику Троуллиера и Мартинса [20] создания псевдопотенциалов. Запишем задачи на собственные значения для одного и того же дифференциального оператора, но с различными собственными значениями:

$$\begin{cases} y_1''(x) + p(x)y_1'(x) + q(x)y_1(x) = \lambda_1 y_1(x), \\ y_2''(x) + p(x)y_2'(x) + q(x)y_2(x) = \lambda_2 y_2(x). \end{cases} \quad (1.1)$$

В контексте однородных дифференциальных уравнений, величина, известная как Вронскиан определяется как

$$W(x) = y_1(x)y_2'(x) - y_2(x)y_1'(x) \quad (1.2)$$

и может быть вычислена по формуле

$$W(x) = W_0 \exp \left[- \int^x p(x') dx' \right], \quad (1.3)$$

в которой константа W_0 является произвольной и не имеет значения.

После проведения анализа, который приводит к уравнению 1.3 для переменных, определенных в уравнении 1.2, но для функций, которые являются решением системы 1.1, мы получим

$$W(x) = \left[(\lambda_2 - \lambda_1) \int^x dx' y_1(x') y_2(x') e^{\left[\int^{x'} dx'' p(x'') \right]} + W_0 \right] e^{\left[- \int^x dx' p(x') \right]}. \quad (1.4)$$

Вронскиан может быть переписан в обозначениях логарифмической производной

$$W(x) = y_1(x) y_2(x) \frac{d}{dx} \{ \lg [y_2(x)] - \lg [y_1(x)] \}. \quad (1.5)$$

Используя уравнения 1.4 и 1.5 при решении уравнения Шредингера для радиальной волновой функции $R_l(r, E)$, переопределив

$$x \rightarrow r; \quad p(x) \rightarrow \frac{2}{r}; \quad q(x) \rightarrow -2 \left[V(r) + \frac{l(l+1)}{r^2} \right]; \quad \lambda \rightarrow -2E,$$

и используя пределы интегрирования по r от 0 до r_c , получим

$$\begin{aligned} & \left[r^2 R_{l,1}(r) R_{l,2}(r) \frac{d}{dr} \{ \lg [R_{l,2}(r)] - \lg [R_{l,1}(r)] \} \right]_0^{r_c} = \\ & = -2(E_2 - E_1) \int_0^{r_c} dr r^2 R_{l,1}(r) R_{l,2}(r). \end{aligned}$$

Умножая на -2π и учитывая, что вклад нижнего предела в левой части уравнения равен нулю из-за множителя r^2 , получим:

$$\begin{aligned} & -2\pi r_c^2 \frac{R_{l,1}(r_c) R_{l,2}(r_c)}{E_2 - E_1} \frac{d}{dr} [\lg [R_{l,2}(r)] - \lg [R_{l,1}(r)]]_{r=r_c} = \\ & = 4\pi \int_0^{r_c} dr r^2 R_{l,1}(r) R_{l,2}(r). \end{aligned}$$

Перейдем к пределу $E_2 \rightarrow E_1$. При этом $R_{l,2}(r) \rightarrow R_{l,1}(r)$. Рассматривая левую часть как производную по энергии мы получим желаемый результат:

$$-2\pi r_c^2 R_l^2(r_c) \frac{d}{dE} \left[\frac{d}{dr} \lg (R_l(r)) \right]_{r=r_c} = 4\pi \int_0^{r_c} dr r^2 R_l^2(r),$$

т.е. первая производная по энергии от логарифмической производной вычисленная на границе остова, непосредственно связана с нормой радиальной части волновой функции в области остова. Таким образом, если псевдо-волновая функция сохраняет норму

$$4\pi \int_0^{r_c} dr r^2 R_l^2(r) = 4\pi \int_0^{r_c} dr r^2 R_{ps,l}^2(r),$$

тогда сдвиг фазы псевдопотенциала будет таким же, как у реального потенциала в первом порядке по энергии, и это может быть достигнуто путем приравнивания псевдо-волновой функции к истинной волновой функции электрона вне остова.

Для точного учета совокупного влияния ядра и электронов остова необходимо использовать нелокальный псевдопотенциал, так как состояния с различными моментами импульса рассеиваются по-разному. В общем случае нелокальный псевдопотенциал можно выразить в полулокальной форме

$$\hat{V}_{ps} = \hat{V}_{loc} + \sum_{m=-l}^l |lm\rangle \delta\hat{V}_l \langle lm|,$$

где $|lm\rangle$ – сферическая гармоника Y_{lm} . Выбор локального потенциала \hat{V}_{loc} является произвольным, но обычно сумма по l ограничивается лишь несколькими слагаемыми (например, $l = 2$), так что локальная часть должна описывать тот потенциал, который соответствует отброшенной части ряда.

Недостатком полулокальной формы является большая вычислительная сложность. Эта проблема может быть преодолена аналитически при определенном наборе локализованных базисных функций, но наиболее распространенным решением является использование формы Клеймана-Биландера [21]:

$$\hat{V}_{KB} = \hat{V}_{loc} + \sum_{lm} \frac{|\delta\hat{V}_l \phi_{lm}\rangle \langle \phi_{lm} \delta\hat{V}_l|}{\langle \phi_{lm} | \delta\hat{V}_l | \phi_{lm}\rangle},$$

где $|\phi_{lm}\rangle$ – собственное состояние псевдогамильтониана. Этот оператор действует на некоторое состояние так же как и оригинальный полулокальный оператор \hat{V}_{ps} , но теперь количество необходимых для вычисления проекций линейно (а не квадратично) зависит от количества базисных функций.

Сохраняющие норму псевдопотенциалы могут быть сгенерированы программой Atom, расположенной в папке Pseudo/atom. На официальном

сайте пакета SIESTA можно скачать пригодные для большинства задач псевдопотенциалы. SIESTA может использовать псевдопотенциалы, сгенерированные рядом других кодов (как правило, в формате .psf). Помните, что все псевдопотенциалы должны быть тщательно протестированы перед использованием.

Псевдопотенциалы считываются SIESTA из разных файлов: по одному для каждого типа атомов (перечисленных в блоке `ChemicalSpeciesLabel`). Название файла должно быть: `Chemical_label.vps` (неформатированный) или `Chemical_label.psf` (ASCII), где `Chemical_label` соответствует названию в блоке `ChemicalSpeciesLabel`. SIESTA может также работать с псевдопотенциалами в формате XML, разработанном Junquera, Garcia и Verstraete, который обеспечивает совместимость с кодом ABINIT. Директория `Util/pseudo-xml` содержит программу, позволяющую преобразовать псевдопотенциалы из формата .XML в формат .psf.

Блок данных `ChemicalSpeciesLabel` определяет различные типы атомов в модели, ассоциируя с ними числа для дальнейшей идентификации.

```
%block Chemical_Species_label
1 6 C
2 14 Si
3 14 Si_surface
%endblock Chemical_Species_label.
```

Первое число в строке задает код типа атома. За ним следует атомный номер (в таблице Менделеева) и строковое название. Это название используется впоследствии для поиска соответствующих файлов, а именно, файлов псевдопотенциалов, пользовательских базисных наборов и других. Отрицательные номера атомов используются для ghost атомов (см. `PAO.basis`).

Атомные номера больше 200 используются для обозначения искусственных атомов (созданных, например, как «смесь» двух реально существующих атомов для расчетов «виртуальных кристаллов» (VCA)). В этом случае блок `SyntheticAtoms` должен присутствовать для того, чтобы предоставить SIESTA информацию об основном состоянии этого искусственного атома.

```
%block Chemical_Species_label
1 201 ON-0.50000
%endblock Chemical_Species_label
%block SyntheticAtoms
1 # индекс типа атома
2 2 3 4 # число валентных электронов в состояниях с l=0,1,2,3
2.0 3.5 0.0 0.0 # занятые валентные состояния с l=0,1,2,3
%endblock SyntheticAtoms.
```

Псевдопотенциалы для искусственных атомов могут быть созданы с помощью программ `mixps` и `fractional` из директории `Util/VCA`. Значение по умолчанию отсутствует. Его необходимо задать явно.

Блок данных `AtomicMass` позволяет пользователю задавать массы различных типов атомов. Это полезно, например, при изучении динамики изотопов. Если индекс типа атома не найден в пределах блока, предполагается использование стандартного значения массы для соответствующего атомного числа. Если блок отсутствует, все массы — стандартные. Каждая строка состоит из типа элемента (целое число) и значения (действительное число) массы. Порядок строк не важен. Если нет целых чисел и/или действительных чисел в пределах строки, строка игнорируется.

```
%block AtomicMass
3 21.5
1 3.2
%endblock AtomicMass
```

Значение по умолчанию: (блок отсутствует или пуст). Используются стандартные массы. Для `ghost` атомов значение по умолчанию `1.d30 a.u.`

1.9. Базисный набор

Независимо от того, какой неэмпирический метод используется для моделирования (Хартри-Фок, DFT, ...), точные волновые функции электронов аппроксимируются комбинацией аналитических функций (базисный набор). Выбор в пользу того или иного базисного набора осуществляется из соображения простоты описания реальных физических систем. Так, если проводится моделирование металлов, для которых характерны делокализованные электроны, логичным кажется использование базиса плоских волн. Если же проводится рассмотрение систем с ковалентными связями, то лучше всего подходит базис из функций, которые хорошо описывают форму электронных орбиталей. Можно не задумываться об этом, используя хорошо известный в мире и изученный вами пакет, но за это придется пожертвовать скоростью расчетов. Дело в том, что для детального описания ковалентной связи может потребоваться ряд из колоссального числа плоских волн, а достаточно всего одной орбитали, повторяющей форму этой связи. И наоборот, описать делокализованный в кристалле электрон в базисе плоских волн намного проще.

Рассмотрим подходы к выбору атомноподобных базисных функций. Хотя в качестве базисных можно использовать любые линейно независимые функции, охватывающие область пространства, в которой находятся электроны, оптимальной является ситуация, когда выбранная базисная функция максимально близка к радиальной составляющей точной водо-

родоподобной атомной орбитали. Такие функции известны как орбитали слейтеровского типа (ОСТ), отвечающие потенциалу центрального поля и удовлетворяющие асимптотическому поведению точной волновой функции как вблизи ядра, так и на больших расстояниях от него. Недостатком ОСТ является то, что они не подходят для быстрого вычисления двухцентровых кулоновских и обменных интегралов. Эти задачи значительно упрощаются при использовании в качестве базисных функций орбиталей гауссова типа (ОГТ). Однако индивидуальные ОГТ и вблизи ядра и вдали от него ведут себя иначе, чем точные волновые функции. Этот недостаток можно исправить. Дело в том, что атомные орбитали (АО) хорошо аппроксимируются линейной комбинацией ОГТ с различными экспонентами. Чтобы обеспечить правильное поведение АО на малых расстояниях от ядра, приходится дополнять ОГТ функциями, имеющими радиальную зависимость типа e^{-sr} . Опыт показывает, что использование даже десяти ОГТ для аппроксимации АО позволяет вычислять интегралы существенно быстрее, чем при использовании одной ОСТ.

В пакете SIESTA реализован метод расчета, предполагающий разреженность гамильтоновой матрицы и матриц перекрытия. Разреженность требует либо пренебрежения матричными элементами, которые достаточно малы, либо использования орбиталей, которые равны нулю за пределами определенного радиуса (радиус обрезания) [22]. Разработчики пакета SIESTA приняли последний подход. В пределах этого радиуса атомные орбитали представляют собой произведение радиальной функции и сферических гармоник. Для атома I , расположенного в точке R_I ,

$$\phi_{Iln}(r) = \phi_{In}(r_I) Y_{lm}(\hat{r}_I),$$

где $r_I = r - R_I$, l — орбитальный момент импульса, m — проекция момента импульса. Момент импульса может быть сколь угодно большим и, как правило, будет несколько орбиталей (помечены индексом n) с одинаковой угловой зависимостью, но разными радиальными зависимостями, которые условно называются multiple ζ базисом.

В случае минимального базисного набора (single- ζ (SZ)) удобным и эффективным оказался метод Санки (Sankey) и Никлевски (Niklewski) [22]. В этом методе базисными орбиталями являются собственные функции гамильтониана для случая (псевдо) атомов в сферической области, радиус которой может быть различным для каждой орбитали:

$$\left(-\frac{1}{2r} \frac{d^2}{dr^2} r + \frac{l(l+1)}{2r^2} + V_l(r) \right) \phi_l(r) = (\epsilon_l + \delta\epsilon_l) \phi_l(r).$$

Энергия $\epsilon_l + \delta\epsilon_l$ выбирается таким образом, чтобы первый узел $\phi_l(r)$ наблюдался при выбранном радиусе обрезания r_l^c . Для того чтобы получить

хорошо сбалансированный базис, как правило, лучше использовать одинаковую добавку к энергии $\delta\epsilon_l$, а не общий радиус r^c для всех атомов и моментов импульса. Это означает, что радиусы орбиталей зависят от сорта атомов и момента импульса.

На практике обнаруживается, что подобная процедура для multiple- ζ базисного набора оказывается неэффективной. Вместо этого применяется схема, основанная на методе валентного расщепления. В этом методе первая орбиталь (first- ζ базисная орбиталь) представляется линейной комбинацией орбиталей гауссового типа, подогнанной под атомные собственные функции. Вторая орбиталь (second- ζ орбиталь) строится путем замены в пределах радиуса r_l^s (радиус расщепления) одной из орбиталей гауссового типа, входящей в first- ζ (обычно самая медленно спадающая), полиномом:

$$\phi_l^{2\zeta}(r) = \begin{cases} r^l(a_l - b_l r^2) & r < r_l^s \\ \phi_l^{1\zeta}(r) & r \geq r_l^s \end{cases},$$

где a_l и b_l определяются требованием непрерывности функции и ее производной в точке r_l^s . Орбитали с большими ζ получаются введением большего количества орбиталей гауссового типа.

Для достижения хорошей сходимости, в дополнение к орбиталям валентных электронов атома, как правило, необходимо также включать поляризационные орбитали, для учета деформации, вызванной образованием связи. Использование псевдоатомных орбиталей с большим моментом импульса неэффективно, потому что они, как правило, слишком удалены. Вместо этого, рассматривают валентные псевдоатомные орбитали

$$\phi_{lm}(r) = \phi_l(r)Y_{lm}(r).$$

Для того, чтобы ее поляризовать, прикладывают небольшое электрическое поле \mathcal{E} в направлении оси z . Используя теорию возмущений первого порядка

$$(H - E)\delta\phi = -(\delta H - \delta E)\phi, \quad (1.6)$$

где $\delta H = \mathcal{E}z$ и $\delta E = \langle \phi | \delta H | \phi \rangle = 0$, так как δH нечетная. Правила отбора предполагают, что в возмущенные орбитали будут иметь только компоненты с $l' = l \pm 1$, $m' = m$:

$$\delta H \phi_{lm}(r) = (\mathcal{E}r \cos \theta) (\phi_l(r)Y_{lm}(r)) = \mathcal{E}r \phi_l(r) (c_{l-1}Y_{l-1} + c_{l+1}Y_{l+1}) \quad (1.7)$$

и

$$\delta \phi_{lm}(r) = \varphi_{l-1}(r)Y_{l-1,m}(r) + \varphi_{l+1}(r)Y_{l+1,m}(r). \quad (1.8)$$

Так как в базисе уже есть орбитали с моментом импульса $l - 1$, выделим компонент $l + 1$, подставляя 1.7 и 1.8 в 1.6, умножая на $Y_{l+1,m}^*(r)$ и

интегрируя по угловым переменным. Таким образом, получим уравнение

$$\left[-\frac{1}{2r} \frac{d^2}{dr^2} r + \frac{(l+1)(l+2)}{2r^2} + V_l(r) - E_l \right] \varphi_{l+1}(r) = -r\phi_l(r),$$

где так же исключены множители \mathcal{C} и c_{l+1} , которые влияют только на нормировку функции $\varphi_{l+1}(r)$. В базис включаются поляризационные орбитали: $\phi_{l+1,m}(r) = N\varphi_{l+1}(r)Y_{l+1,m}(r)$, где N – нормировочный множитель.

Важно подчеркнуть, что ни метод SIESTA, ни код программы не связаны с использованием какого-либо особого вида атомных орбиталей. Пользователь может использовать в SIESTA подготовленный им базис. У большинства пользователей, однако, нет своих собственных базисных наборов. Для этих пользователей разработаны схемы создания базисных наборов в рамках программы с минимальным количеством входных параметров. Если ничего не указано во входном файле, SIESTA создает базисный набор по умолчанию, который может быть хорошим приближением. Пользователь может настроить ряд параметров, которые влияют на качество и эффективность базисного набора.

Строка PAO.BasisSize определяет размер базисного набора. Допустимы следующие значения.

- SZ или MINIMAL: минимальный или single- ζ базис.
- DZ: Double- ζ базисный набор.
- SZP: Single- ζ базисный набор с добавлением поляризационных орбиталей.
- DZP или STANDARD: Как и DZ, но с добавлением поляризационных орбиталей.
- TZ: Triple- ζ базисный набор.
- TZP: Triple- ζ базисный набор с добавлением поляризационных орбиталей.
- TZDP: Triple- ζ базисный набор с добавлением Double- ζ базисного набора для поляризационных орбиталей.

Основные состояния атомов, используемые в пакете SIESTA, определены в исходном файле Src/periodic_table.f. Для некоторых элементов (например, Pd), конфигурация может быть не стандартной. Значение по умолчанию: DZP.

Блок данных PAO.BasisSizes позволяет определить различные базисные наборы для каждого типа атомов. Например,

```
%block PAO.BasisSizes
Si DZ
H DZP
```

```
O SZP
%endblock PAO.BasisSizes.
```

Условие обрезания для орбиталей задается действительным значением энергии `PAO.EnergyShift`. Параметр учитывается, если отсутствует блок `PAO.Basis` или когда радиусы, определенные в этом блоке, равны нулю для первой ζ . Значение по умолчанию: 0.02 Ry.

Параметр `PAO.SplitNorm` задает долю нормы, соответствующую нерасщепленной части `second- ζ` орбитали. Если используется `multiple- ζ` , соответствующие радиусы получаются умножением `PAO.SplitNorm` на 1/2, 1/4 Этот параметр оказывает влияние только если отсутствует блок `PAO.Basis` или когда радиусы, указанные в этом блоке равны нулю для $\zeta > 1$. Значение по умолчанию: 0.15 (значения должны быть в диапазоне от 0.05 до 0.5).

1.10. Сетка в координатном пространстве

SIESTA использует трехмерную сетку для расчета некоторых интегралов и представления плотности заряда и потенциала. Ее детализированность определяется плоской волной отсечки, указанной в опции `MeshCutoff` (значение по умолчанию: 100 Ry). Все плоские волны, соответствующие кинетической энергии, меньшей, чем параметр `MeshCutoff`, могут быть представлены на сетке без наложения. Для функций (например, эффективного потенциала), являющихся разложением только по этим плоским волнам, возможно прямое и обратное преобразования Фурье без каких-либо приближений.

Наличие сетки приводит к нарушению трансляционной симметрии: граничному эффекту, возникающему из-за того, что разложения зарядовой плотности и потенциала должны содержать плоские волны с энергией выше `MeshCutoff`. Этот эффект называется эффектом яичной коробки (egg box effect). Он проявляется в том, что полная энергия и силы неестественно зависят от положения ядер относительно точек расчетной сетки. Опишем причину возникновения этого эффекта. На рис. 1.2 (а) изображены функция $V(x)$ и некоторые точки x_i на оси x , представляющие узлы сетки. Аппроксимацией интеграла от функции $V(x)$ будет $\sum_i V(x_i)$. Выбранные значения $V(x_i)$ представлены на рисунке выколотыми кругами. Предположим, что $V(x)$ является концентрацией электронов в атоме, тогда интеграл будет иметь смысл суммарного заряда. Если мы сдвинем атом вправо, то получим ситуацию, представленную на рис. 1.2 (б). Он очень похож на рис. 1.2 (а), но одна из точек x_i лежит в области резкого изменения распределения $V(x)$. Это существенно повлияет на оценку

интеграла суммой $\sum_i V(x_i)$. Таким образом, каждый раз, когда это резкое изменение пересекает точку сетки, оценка интеграла будет изменяться. Это и есть эффект яичной коробки. Он возникает, если функция не является достаточно гладкой (или дискретизация не достаточно подробна).

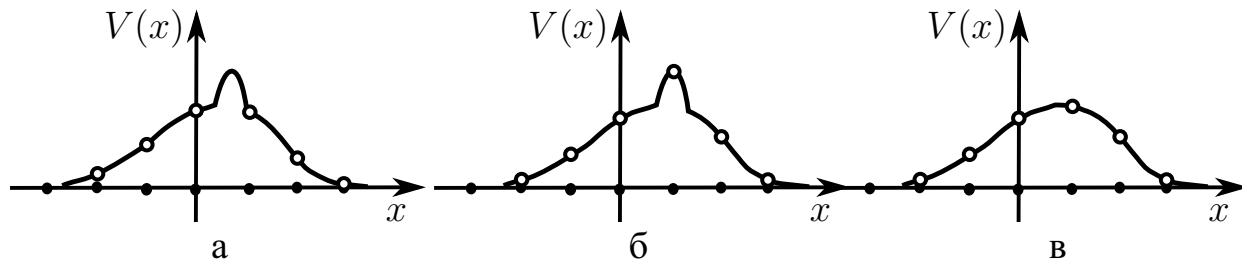


Рис. 1.2.: К пояснению эффекта яичной коробки

Одним из способов избежать появления этого эффекта является улучшение дискретизации сетки. Другим, более изысканным и экономичным решением, может быть сглаживание распределения $V(x)$. Сначала для получения спектра $V(k)$ выполняется Фурье-преобразование функции $V(x)$. Известно, что проблемы вызывают длины волн λ меньше шага сетки a . Можно вычислить критическое волновое число $k_c = 2\pi/a$. Из спектра $V(k)$ убираются коэффициенты $V(k)$ с $k > k_c$. После этого проводится обратное преобразование Фурье $V(k)$, чтобы получить сглаженную $V(x)$ (см. рис. 1.2 (в)).

Для активации сглаживания («фильтрация» — технический термин) в SIESTA необходимо установить параметру `FilterCutoff` значение энергии, соответствующей плоской волне с длиной больше размера сетки.

Перечислим основные параметры, позволяющие настраивать сетку в координатном пространстве.

`FilterCutoff` — энергия обрезания, используемая для фильтрации всех атомных базисных функций и потенциала нейтрального атома. Базисные функции (которые должны быть возведены в квадрат для получения электронной плотности) фильтруются с энергией обрезания, уменьшенной в $0,7^2 \simeq 0,5$ раза. Параметр `FilterCutoff` не должен превышать `MeshCutoff`, чтобы избежать эффекта яичной коробки для сил, действующих на атомы. Не нужно пытаться варьировать `MeshCutoff`, одновременно изменяя `FilterCutoff`, так как последний фактически изменяет используемые базисные функции. Лучше зафиксировать достаточно большим `FilterCutoff` и варьировать только `MeshCutoff`. Если `FilterCutoff` не задан явно, он вычисляется по `FilterTol`.

`FilterTol` — кинетическая энергия, отброшенная в результате фильтрации каждой базисной функции. В то время как `FilterCutoff` уста-

навливает общее условие обрезания в обратном пространстве для всех базисных функций, `FilterTol` устанавливает определенное значение для каждой базисной функции, подобно тому, как `EnergyShift` устанавливает их в координатном пространстве. По этой причине разумно использовать близкие значения для обоих параметров. Максимальная энергия обрезания, необходимая для выполнения `FilterTol` на всех базисных функциях, используется (умноженная на $1/0,7^2 \simeq 2$) для фильтрации потенциала нейтрального атома. Параметр `FilterTol` будет проигнорирован, если `FilterCutoff` присутствует во входном файле. Если отсутствуют и `FilterCutoff`, и `FilterTol`, фильтрация не выполняется.

Если значение `FilterCutoff` очень маленькое (или `FilterTol` очень большое) некоторые отфильтрованные базисные орбитали окажутся некорректными и приведут к неверным результатам или даже остановке программы.

Для разумных значений энергии отсечки эффект яичной коробки для полной энергии, как правило, незначителен. Тем не менее, он может существенно повлиять на процесс релаксации, значительно увеличивая количество шагов, а также может испортить фононный расчет, делая его результаты более сложными, чем у оптимальной структуры.

Для выбранной сетки эффект яичной коробки может быть уменьшен путем восстановления симметрии следующих характеристик:

1. вклад в энергию, зависящий от сетки в координатном пространстве;
2. силы;
3. тензор напряжений;
4. дипольный момент.

Полная симметризация подразумевает усреднение (интегрирование) по расчетной ячейке. Она может проводиться и для заданного набора жестких сдвигов модели в пределах расчетной ячейки. Характеристики системы усредняются по набору таких сдвигов. Процедура самосогласования проводится для исходной системы. После получения равновесной (соседней) матрицы плотности, та же (фиксированная) матрица плотности используется для расчета необходимых свойств системы, смещенной в определенном направлении (предполагается, что эффект яичной коробки несущественно влияет на матрицу плотности). Необходимые для каждого смещения ресурсы процессора сопоставимы с необходимыми для одного дополнительного шага самосогласования.

Рассмотрим, например, кубическую ячейку. Если блок `GridCellSampling` отсутствует или не содержит данных (значение по умолчанию), то будет использоваться только исходная модель (0,0,0). Блок:

```
%block GridCellSampling
0.5 0.5 0.5
%endblock GridCellSampling
\begin{fdrcode}
```

приведет к рассмотрению центра ячейки в качестве второй точки. Блок:

```
\begin{fdrcode}
```

```
%block GridCellSampling
0.5 0.5 0.0
0.5 0.0 0.5
0.0 0.5 0.5
%endblock GridCellSampling
```

задаст ГЦК-подобное разбиение, а

```
%block GridCellSampling
0.5 0.0 0.0
0.0 0.5 0.0
0.0 0.0 0.5
0.0 0.5 0.5
0.5 0.0 0.5
0.5 0.5 0.0
0.5 0.5 0.5
%endblock GridCellSampling
```

приведет к рассмотрению кубической ячейки, сторона которой в два раза меньше, чем у исходной ячейки. Выбор оптимального набора дополнительных точек, позволяющих достичь максимального эффективного параметра обрезания, неочевиден. Он зависит от типа ячейки.

Пример. Эффект яичной коробки в MgO

Рассмотрим примитивную ячейку MgO. Она содержит два атома: Mg и O. Для проверки наличия эффекта яичной коробки можно использовать скрипт `run_edge`, написанный на языке Python.

```
#!/home/ssozykin/bin/anaconda/bin/python
# для запуска программ
from subprocess import Popen, PIPE
import os
import re      # для использования регулярных выражений
import shutil # для копирования файлов

SystemLabel = 'MgO'
runcom = 'siesta < inputrun.fdf > ./input.out'
# команда запуска расчета пакета SIESTA
N = 20      # число шагов
mesh = 16   # число точек сетки в координатном
            # пространстве вдоль оси z
rmlist = 'rm *.DM *.XV' # команда для удаления
                    # временных файлов
```

```

# для получения свойств из выходного файла SIESTA
def FileProp(filename,prop,count = 1,type = 'int'):
    property = 1
    k = 1
    if os.path.exists(filename):
        MdSiestaFile = open(filename)
        str1 = MdSiestaFile.readline()
        while str1!='':
            if (str1 != '') and (str1.find(prop)>=0):
                str1 = str1.replace(prop,' ')
                if (type == 'string'):
                    property = Helpers.spacedel(str1)
                else:
                    prop1 = re.findall(r"[0-9,\.,-]+", str1)[0]
                    if (type == 'int'):
                        property = int(prop1)
                    if (type == 'float'):
                        property = float(prop1)
            if (k == count):
                return property
            k+=1
            str1 = MdSiestaFile.readline()
        MdSiestaFile.close()
    return property
# обнаружен файл, оставшийся от предыдущего расчета
if (os.path.exists(SystemLabel+'.DM')):
    Popen(rmlist,shell=True,stdout=PIPE).communicate()
# главный цикл
for i in xrange(0,N):
    # если расчет для этой итерации еще не запускался
    if (not os.path.exists('./input'+str(i)+'.out')):
        my_file = './inputrun.fdf'
        f = open(my_file, 'w')
        f1 = open("log.txt","a")
        f2 = open('input.fdf')
        for line in f2:
            f.write(line)
        f2.close()
        f.write('%block AtomicCoordinatesOrigin\n')
        f.write('0.0 0.0 '+str(((1.0*i)/mesh)/N)+'\n')
        f.write('%endblock AtomicCoordinatesOrigin\n')
        string = str(i)
        f.close()
        Popen(runcom,shell=True,stdout=PIPE).communicate()
        shutil.copy('input.out', 'input'+str(i)+'.out')
        arg1 = 'input'+str(i)+'.out'
        arg2 = 'siesta: Etot ='
        Etot = FileProp(arg1,arg2,2,'float')

```

```

f1.write(string + ' ' + str(Etot) + "\n")
f1.close()
# удалить временные файлы
if (os.path.exists(SystemLabel+'.DM')):
    Popen(rmlist, shell=True, stdout=PIPE).communicate()

```

Для корректной работы скрипта необходимо верно инициализировать параметр *mesh* (число точек сетки в координатном пространстве вдоль оси *z*). В процессе работы скрипта запускается *N* расчетов, отличающихся сдвигом по оси *z* в блоке `AtomicCoordinatesOrigin`. Файл `log.txt` содержит энергии системы для каждой итерации главного цикла. По изменению энергии от итерации к итерации можно судить о наличии эффекта яичной коробки.

За основу `fdf` файла возьмем содержащийся в директории `Tests` дистрибутива SIESTA.

```

1 # FDF file for bulk MgO
2 SystemName Magnesium Oxide Crystal
3 SystemLabel          MgO
4 NumberOfAtoms        2
5 NumberOfSpecies      2
6
7 %block Chemical_Species_Label
8   1   12   Mg
9   2    8    O
10 %endblock Chemical_Species_Label
11
12 PAO.BasisSize        SZ
13
14 AtomicCoordinatesFormat ScaledCartesian
15 AtomicCoorFormatOut   Ang
16
17 %block AtomicCoordinatesAndAtomicSpecies
18   .000   .000   .000   1
19   .500   .500   .500   2
20 %endblock AtomicCoordinatesAndAtomicSpecies
21
22 LatticeConstant      4.10 Ang
23
24 %block LatticeVectors
25   0.000  0.500  0.500
26   0.500  0.000  0.500
27   0.500  0.500  0.000
28 %endblock LatticeVectors
29
30 kgrid_cutoff         7. Ang
31
32 XC.functional        LDA
33 XC.authors            CA

```

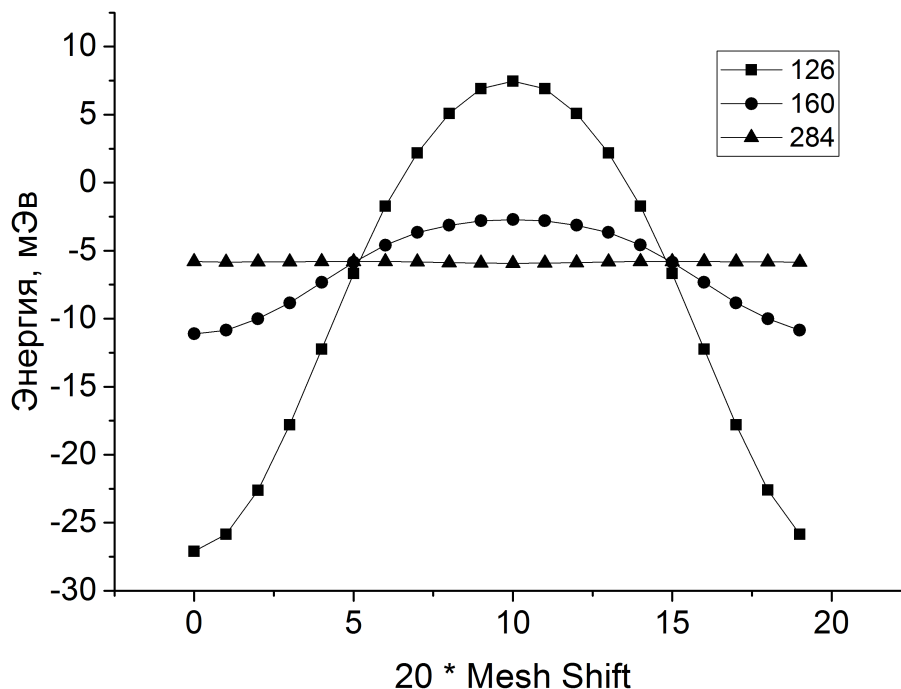


Рис. 1.3.: К оценке зависимости эффекта яичной коробки от MeshCutoff

```

34 SpinPolarized      .false.
35 MeshCutoff        100. Ry
36 MaxSCFIterations  100
37 DM.MixingWeight   0.3
38 DM.Tolerance      1.d-4
39 DM.NumberPulay    3
40 SolutionMethod    diagon
41 ElectronicTemperature 5 K
42 MD.TypeOfRun      cg

```

Результат работы скрипта для трех различных значений MeshCutoff (и трех значений параметра *mesh* в скрипте) приведен на рис. 1.3. Как видно из этого рисунка, эффект яичной коробки становится мало заметен при увеличении MeshCutoff до 284 Ry. При MeshCutoff = 100 Ry ошибка в энергии, обусловленная сдвигом модели в расчетной ячейке может достигать 35 мэВ. Хотя эта величина мала по сравнению с полной энергией системы (458 эВ), она может внести существенную ошибку при расчете свойств, которые зависят от разности энергий. Данный способ устранения эффекта яичной коробки требует заметного увеличения ресурсов процессора и памяти.

В качестве альтернативы рассмотрим еще два подхода: использование блока GridCellSampling и фильтрацию. Первый из них реализовывался для следующего набора дополнительных направлений сдвига:

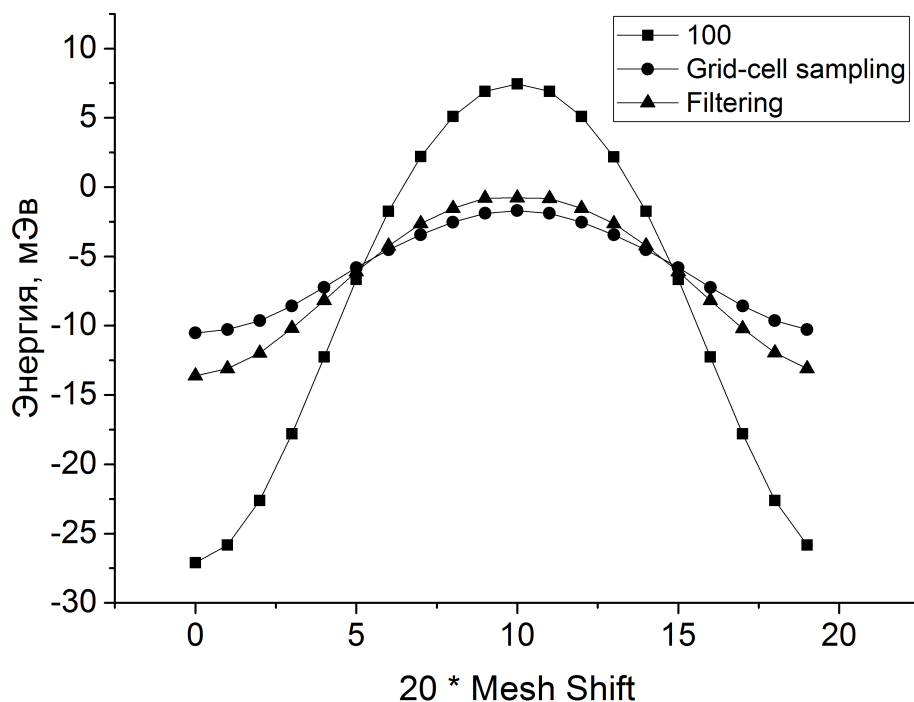


Рис. 1.4.: Альтернативные способы уменьшения эффекта яичной коробки

```
%block GridCellSampling
0.5 0.0 0.0
0.0 0.5 0.0
0.0 0.0 0.5
%endblock GridCellSampling.
```

Для активации фильтрации во входной файл необходимо добавить три строки:

```
PAO.Filter T
VNA.Filter T
FilterCutoff 100 Ry.
```

Результаты расчетов приведены на рис. 1.4. Из этого рисунка видно, что оба подхода обеспечивают примерно одинаковое уменьшение эффекта яичной коробки.

1.11. Оптимизация структуры и молекулярная динамика

Код SIESTA имеет внешний цикл изменения атомной структуры: для заданных атомных позиций вычисляется электронная структура, силы и напряжения, обновляются атомные позиции (и, возможно, вектора транс-

ляции) и происходит переход к следующей итерации.

В пакете SIESTA реализовано несколько алгоритмов молекулярной динамики и оптимизации геометрии. Выбор метода производится инициализацией переменной `MD.TypeOfRun` (строка):

- CG — оптимизация координат по методу сопряженных градиентов.
- Broyden — оптимизация координат по модифицированной схеме Бройдена.
- FIRE — оптимизация координат по быстрому алгоритму [23].
- Verlet — стандартный алгоритм молекулярной динамики Верле.
- Nose — молекулярная динамика с термостатом Нозе.
- ParrinelloRahman — молекулярная динамика по методу Паринелло-Рамана.
- NoseParrinelloRahman — молекулярная динамика с термостатом Нозе и баростатом Паринелло-Рамана.
- Anneal — молекулярная динамика с отжигом при необходимой температуре и (или) давлении (см. переменную `MD.AnnealOption`).
- FC — вычисление матрицы силовых констант для расчета фононного спектра (см. стр. 76).

Значение по умолчанию: Verlet (CG для системы из одного атома).

Опционально (см. переменную `MD.VariableCell`) оптимизация структуры методами CG, Broyden и FIRE может включать оптимизацию векторов трансляции.

Остановимся подробнее на параметрах, относящихся к оптимизации атомной структуры.

Оптимизация структуры

В режиме оптимизации структуры SIESTA смещает атомы (и, возможно, изменяет вектора трансляции) пытаясь минимизировать силы, действующие на атомы. Перечисленные ниже опции являются общими для всех методов оптимизации геометрии. Приставка «MD» вводит в заблуждение. Она сохранена для совместимости с предшествующими версиями кода.

Если логическая переменная `MD.VariableCell` равна `true`, вектора трансляции оптимизируются вместе с оптимизацией координат атомов. Можно задать гидростатическое давление или произвольный тензор напряжения (обратите внимание на параметры `MD.MaxStressTol`, `MD.TargetPressure`, `MD.TargetStress`, `MD.ConstantVolume` и `MD.PreconditionVariableCell`). Переменная `MD.VariableCell` используется только если `MD.TypeOfRun` равен CG, Broyden или FIRE. Значение по умолчанию: `false`.

Если логическая переменная `MD.ConstantVolume` равна `true`, объем элементарной ячейки поддерживается постоянным при оптимизации векторов трансляции: могут изменяться только форма ячейки и координаты атомов. Обратите внимание, что в этом случае не имеет смысла указывать приложенные напряжения или давление (за исключением анизотропных напряжений). Переменная `MD.ConstantVolume` используется только если `MD.TypeOfRun` равно `CG`, `Broyden` или `FIRE` и `MD.VariableCell` равно `true`. Значение по умолчанию: `false`.

Если логическая переменная `MD.RelaxCellOnly` равна `true`, проводится оптимизация только параметров ячейки (по методам `Broyden` или `FIRE`). Дробные координаты не изменяются. Для расчетов с использованием `Zmatrix`, дробные координаты первого атома в каждой молекуле сохраняются фиксированными (не предпринимается попыток изменения расстояний или углов). Этот параметр используется только если `MD.TypeOfRun` равен `FIRE` или `Broyden`, а переменная `MD.VariableCell` равна `true`. Значение по умолчанию: `false`.

`MD.MaxForceTol` (действительное значение силы) – критерий сходимости по силам при оптимизации координат. Вычисления заканчиваются если максимальная сила, действующая на атом меньше `MD.MaxForceTol`. Этот параметр используется только если `MD.TypeOfRun` равен `CG`, `Broyden` или `FIRE`. Значение по умолчанию: `0.04 eV/Ang`.

`MD.MaxStressTol` (действительное значение давления) – критерий сходимости по напряжению при оптимизации методом сопряженных градиентов с включенной опцией `variable-cell`. Вычисления заканчиваются если максимальная сила, действующая на атом меньше чем `MD.MaxForceTol`, а максимальная компонента тензора напряжений меньше `MD.MaxStressTol`. Этот параметр используется только если `MD.TypeOfRun` равен `CG`, `Broyden` или `FIRE`, а параметр `MD.VariableCell` равен `true`. Значение по умолчанию: `1.0 GPa`.

`MD.NumCGsteps` – максимальное число шагов оптимизации по методу сопряженных градиентов (или `Broyden`) (минимизация будет прервана раньше, если будет достигнут критерий сходимости по силам). Используется только если `MD.TypeOfRun` равен `CG` или `Broyden`. Значение по умолчанию: `0`.

`MD.MaxCGDispl` – максимальное смещение атома за один шаг минимизации. Используется только если `TypeOfRun` равен `CG`, `Broyden` или `FIRE`. Для метода `Broyden` дополнительно можно ограничить смещения используя параметр `MD.Broyden.Initial.Inverse.Jacobian`. Для метода `FIRE` этого результата можно добиться выбирая меньший шаг по времени. Учитывайте, что существуют специфичные опции для `Zmatrix`, которые переопределяют данный параметр. Значение по умолчанию: `0.2 Bohr`.

`ZM.ForceTolLength` — критерий сходимости по силам при изменении длин в *Z*-матрице. Значение по умолчанию: 0.00155574 Ry/Bohr.

`ZM.ForceTolAngle` — критерий сходимости по силам при изменении углов в *Z*-матрице. Значение по умолчанию: 0.00356549 Ry/rad.

`ZM.MaxDisplLength` — максимальное изменение длин в *Z*-матрице на одном шаге моделирования. Значение по умолчанию: 0.2 Bohr.

`ZM.MaxDisplAngle` — максимальное изменение углов в *Z*-матрице на одном шаге моделирования. Значение по умолчанию: 0.003 rad.

Оптимизация геометрии по методу сопряженных градиентов

Этот метод используется по умолчанию для оптимизации геометрии. Опция `MD.UseSaveCG` (логическая переменная) реализована именно для этого метода и не работает для остальных. Она указывает, нужно ли считывать информацию о предыдущем запуске оптимизации геометрии по методу сопряженных градиентов, сохраненную в файл `SystemLabel.CG`. Для корректного продолжения прерванного расчета необходимо так же присвоить переменной `MD.UseSaveXV` значение `true`. Если необходимые файлы не существуют, печатается предупреждающее сообщение, но работа программы не прерывается. Значение по умолчанию `false`.

Оптимизация по алгоритму Бройдена

Этот метод использует модифицированный алгоритм Бройдена для построения матрицы Якоби [24]. Специфичными для него являются два параметра.

Целое число `MD.Broyden.History.Steps` задает количество шагов релаксации, по которым модифицированный алгоритм Бройдена строит матрицу Якоби. Используется только если `MD.TypeOfRun` равен `Broyden`. Значение по умолчанию: 5.

Логическая переменная `MD.Broyden.Cycle.On.Maxit` показывает нужно ли вытеснять старые данные новыми при достижении максимального количества данных, которые хранятся для вычисления матрицы Якоби. Альтернативой является повторный запуск алгоритма минимизации Бройдена с первого шага (может быть полезно при проблемах с минимизацией). Используется только если `MD.TypeOfRun` равен `Broyden`. Значение по умолчанию: `true`.

Оптимизация FIRE

Реализация быстрого метода релаксации (Fast Inertial Relaxation Engine — FIRE) [23] в совместимой с `CG` и `Broyden` форме. Стоит обра-

тить внимание на действительное значение времени `MD.FIRE.TimeStep`, которое задает фиктивный шаг по времени для оптимизации FIRE. Значение по умолчанию: `MD.LengthTimeStep`.

Закалка в молекулярной динамике

Этот метод основан на методе молекулярной динамики, но используются для структурной релаксации. Следует отметить, что входной параметр `Zmatrix` не совместим с молекулярной динамикой. Начальная геометрия может быть задана с помощью формата `Zmatrix`, но обобщенные координаты `Zmatrix` не будут обновлены. Отметим также, что критерии сходимости по силе и напряжению не влияют на условия прерывания этого метода. Расчет завершается после указанного количества шагов молекулярной динамики.

Выполнить закалку в молекулярной динамике позволяет логический параметр `MD.Quench`. В этом режиме каждый компонент скорости устанавливается в ноль, если он направлен против соответствующей компоненты силы. Параметр используется только если переменная `MD.TypeOfRun` равна `Verlet` или `ParrinelloRahman`. Он несовместим с термостатом Нозе. Этот режим позволяет проводить оптимизацию только координат атомов (при `MD.TypeOfRun = Verlet`) или координат атомов и формы ячейки (при `MD.TypeOfRun = ParrinelloRahman`). Значение по умолчанию: `false`.

Заданное механическое напряжение

Для оптимизации атомной структуры и расчетов молекулярной динамики при постоянном давлении полезно ознакомиться со следующими параметрами.

`MD.TargetPressure` (действительное значение давления). Заданное давление для метода Парринелло-Рамана, оптимизации расчетной ячейки и отжига. Используется только если параметр `MD.TypeOfRun` равен `ParrinelloRahman`, `NoseParrinelloRahman`, `CG`, `Broyden`, `FIRE (variable cell)` или `Anneal` (если параметр `MD.AnnealOption` равен `Pressure` или `TemperatureandPressure`). Значение по умолчанию: `0.0 GPa`.

`MD.TargetStress` (блок данных). Внешний тензор напряжений для оптимизации расчетной ячейки. Компоненты тензора напряжений приведены в строке в следующем порядке `xx`, `yy`, `zz`, `xy`, `xz`, `yz`. Единицы измерения как и у `MD.TargetPressure`, но с противоположным знаком. Например, одноосное напряжение сжатия 2 ГПа вдоль направления 100 задается командами

```
MD.TargetPressure 2. GPa
%block MD.TargetStress
-1.0 0.0 0.0 0.0 0.0 0.0
%endblock MD.TargetStress
```

Этот параметр используется только если параметр `MD.TypeOfRun` равен CG, Broyden, или FIRE, а `MD.VariableCell` равен true. Значение по умолчанию: гидростатическое давление: -1., -1., -1., 0., 0., 0.

`MD.RemoveIntramolecularPressure` (логическая переменная). Если этот параметр равен true, то вклады в напряжение от внутренних степеней свободы молекул будут вычитаться из тензора напряжений, используемого в оптимизации расчетной ячейки. Это осуществляется приближенно с использованием вириальной формы напряжения, и предполагая, что средняя сила по молекуле представляет собой межмолекулярное напряжение. Значение по умолчанию: false.

Молекулярная динамика

В этом режиме программа перемещает атомы (и, возможно, базисные вектора) в соответствии с силами, используя классические уравнения движения. Следует отметить, что формат *Z*-матрицы не совместима с молекулярной динамикой. Начальная геометрия может быть задана с помощью формата *Z*-матрицы, но обобщенные координаты не обновляются.

Целочисленное значение `MD.FinalTimeStep` задает номер последнего шага моделирования молекулярной динамики. Значение по умолчанию: 1 (совпадает с номером первого шага моделирования).

Действительное значение `MD.LengthTimeStep` определяет шаг по времени для моделирования методом молекулярной динамики. Значение по умолчанию: 1.0 fs.

`MD.InitialTemperature` (действительное значение температуры или энергии). Начальная температура для моделирования методом молекулярной динамики. Атомам приписываются скорости в соответствии с распределением Максвелла для данной температуры. Скорость центра масс равна нулю. Эта переменная используется только если параметр `MD.TypeOfRun` равен Verlet, Nose, ParrinelloRahman, NoseParrinelloRahman или Anneal. Значение по умолчанию: 0.0 K.

`MD.TargetTemperature` (действительное значение температуры или энергии). Конечная температура для термостата Нозе и параметра отжига. Используется только если параметр `MD.TypeOfRun` равен Nose, NoseParrinelloRahman или Anneal (если параметр `MD.AnnealOption` равен Temperature или TemperatureandPressure). Значение по умолчанию: 0.0 K.

Обобщенная масса в подходе Нозе задается параметром `MD.NoseMass`

(действительное значение момента инерции), который определяет масштаб времени в методе Нозе. Используется только если параметр `MD.TypeOfRun` равен `Nose` или `NoseParrinelloRahman`. Значение по умолчанию: $100.0 \text{ Ry} \cdot \text{fs}^2$.

`MD.ParrinelloRahmanMass` (действительное значение момента инерции). Обобщенная масса в методе Паринелло-Рамана. Этот параметр определяет временной масштаб динамики Паринелло-Рамана. Используется только если переменная `MD.TypeOfRun` равна `ParrinelloRahman` или `NoseParrinelloRahman`. Значение по умолчанию: $100.0 \text{ Ry} \cdot \text{fs}^2$.

Строка `MD.AnnealOption` определяет тип отжига.

- `Temperature` (достичь заданную температуру посредством изменения скоростей);
- `Pressure` (достичь заданное давление путем изменения размера и формы расчетной ячейки);
- `TemperatureandPressure` (достичь заданные температуру и давление).

Этот параметр используется только если переменная `MD.TypeOfRun` равна `Anneal`. Значение по умолчанию: `TemperatureAndPressure`.

Действительное значение времени `MD.TauRelax` определяет время релаксации для достижения заданной температуры и (или) давления. Этот параметр является грубой оценкой необходимого времени для достижения заданных температуры и давления. Он используется только если параметр `MD.TypeOfRun` равен `Anneal`. Значение по умолчанию: 100.0 fs .

Действительное значение давления `MD.BulkModulus` задает модуль всестороннего сжатия. Параметр используется только если `MD.TypeOfRun = Anneal` при `MD.AnnealOption = Pressure` или `TemperatureAndPressure`. Значение по умолчанию: 100.0 Ry/Bohr^3 .

1.12. Упражнение 1

Рассмотрим входные файлы для запуска расчета оптимизации геометрии однослойных углеродных нанотрубок (3,3) и (5,0) (можно скачать по адресу <http://physics.susu.ac.ru/courses/nanophys>).

Входной файл УНТ (3,3)

```
1 SystemName          nanotube (3,3)
2 SystemLabel         SWNT
3 NumberOfAtoms      12
4 NumberOfSpecies     1
5
6 %block ChemicalSpeciesLabel
7 1 6 C
8 %endblock ChemicalSpeciesLabel
```

```

9
10 PAO.BasisSize      DZP
11 PAO.EnergyShift   0.05 eV
12 MeshCutoff        150.0 Ry
13
14 %block kgrid_Monkhorst_Pack
15 1 0 0 0
16 0 1 0 0
17 0 0 1 0
18 %endblock kgrid_Monkhorst_Pack
19
20 SpinPolarized True
21 SolutionMethod     diagon
22 AtomicCoordinatesFormat      NotScaledCartesianAng
23
24 XC.functional=LDA
25 XC.authors=CA
26 ZM.UnitsLength = Ang
27
28 %block Zmatrix
29 cartesian
30 1 1.591966 1.345688 0.000000 1 1 1
31 1 -1.042261 1.805249 0.000000 1 1 1
32 1 -1.961383 0.705839 0.000000 1 1 1
33 1 -1.042261 -1.805249 0.000000 1 1 1
34 1 0.369417 -2.051527 0.000000 1 1 1
35 1 2.084522 -0.000000 0.000000 1 1 1
36 1 1.961383 -0.705839 1.241014 1 1 1
37 1 1.042261 1.805249 1.241014 1 1 1
38 1 -0.369417 2.051527 1.241014 1 1 1
39 1 -2.084522 0.000000 1.241014 1 1 1
40 1 -1.591966 -1.345688 1.241014 1 1 1
41 1 1.042261 -1.805249 1.241014 1 1 1
42 %endblock Zmatrix
43
44 LatticeConstant      24.4 Ang
45
46 %block LatticeParameters
47 1.0 1.0 0.1 90. 90. 90.
48 %endblock$\\;\\;\\;\\;$ LatticeParameters
49
50 MD.TypeOfRun         cg
51 MD.NumCGsteps        360
52 MD.MaxCGDispl        0.15 Ang
53 MD.UseSaveXV         yes
54
55 WriteCoorXmol        True
56 WriteForces          True
57 WriteMullikenPop     1

```

Входной файл УНТ (5,0)

```
1 SystemName          nanotube (5,0)
2 SystemLabel         SWNT
3 NumberOfAtoms       15
4 NumberOfSpecies     1
5
6 %block ChemicalSpeciesLabel
7 1 6 C
8 %endblock ChemicalSpeciesLabel
9
10 PAO.BasisSize       DZP
11 PAO.EnergyShift     0.05 eV
12
13 MeshCutoff          150.0 Ry
14
15 %block kgrid_Monkhorst_Pack
16 1 0 0 0
17 0 1 0 0
18 0 0 1 0
19 %endblock kgrid_Monkhorst_Pack
20
21 SpinPolarized True
22 SolutionMethod      diagon
23
24 AtomicCoordinatesFormat NotScaledCartesianAng
25
26 XC.functional=LDA
27 XC.authors=CA
28 ZM.UnitsLength = Ang
29
30 %block Zmatrix
31 cartesian
32 1 1.583736 1.150651 0.000000 1 1 1
33 1 -0.604933 1.861793 0.000000 1 1 1
34 1 -1.957605 0.000000 0.000000 1 1 1
35 1 -0.604933 -1.861793 0.000000 1 1 1
36 1 1.583736 -1.150651 0.000000 1 1 1
37 1 0.604933 1.861793 0.710500 1 1 1
38 1 -1.583736 1.150651 0.710500 1 1 1
39 1 -1.583736 -1.150651 0.710500 1 1 1
40 1 0.604933 -1.861793 0.710500 1 1 1
41 1 1.957605 -0.000000 0.710500 1 1 1
42 1 0.604933 1.861793 2.131500 1 1 1
43 1 -1.583736 1.150651 2.131500 1 1 1
44 1 -1.583736 -1.150651 2.131500 1 1 1
45 1 0.604933 -1.861793 2.131500 1 1 1
46 1 1.957605 -0.000000 2.131500 1 1 1
47 %endblock Zmatrix
48
```

```

49 | LatticeConstant          28.4 Ang
50 |
51 | %block LatticeParameters
52 | 1.0 1.0 0.1 90. 90. 90.
53 | %endblock LatticeParameters
54 |
55 | MD.TypeOfRun             cg
56 | MD.NumCGsteps            360
57 | MD.MaxCGDispl            0.15 Ang
58 | MD.UseSaveXV             yes
59 |
60 | WriteCoorXmol            True
61 | WriteForces              True
62 | WriteMullikenPop         1

```

Значение подавляющего большинства параметров в этих файлах вам уже известны из этого пособия. Об остальных вы можете узнать из официального руководства пользователя пакета SIESTA. Полные ответы на представленные ниже задания должны содержать описание алгоритма выполнения, графики, таблицы и выводы.

Задание 1. Очевидно, что для моделей нанотрубок строго определен лишь параметр трансляции вдоль оси нанотрубки. Размер расчетной ячейки в двух других измерениях определяется конкретной задачей. Предположим, что мы хотим провести моделирование индивидуальной нанотрубки. В этом случае параметр трансляции вдоль осей x и y (ось z сонаправлена с осью нанотрубки) должен быть достаточно большим. Рост размера ячейки увеличивает время расчета. Подберите оптимальное соотношение параметров трансляции: при увеличении размера ячейки изменение энергии системы не должно превышать некоторого выбранного значения.

Задание 2. Варьируя переменную `LatticeConstant`, найдите равновесный параметр трансляции для фрагментов УНТ (3,3) и (5,0). Рассмотреть случай использования обменно-корреляционных функционалов SA и PBE.

Задание 3. Как изменяются результаты, полученные в задании 2, при увеличении или уменьшении переменной `EnergyShift` в 2 раза?

Задание 4. Как изменяются результаты, полученные в задании 2, при увеличении или уменьшении переменной `MeshCutoff`? Определите приемлемое значение параметра `MeshCutoff`.

Задание 5. Как зависят результаты задания 2 от выбора значения `PAO.BasisSize` (SZ, SZP, DZ, DZP, TZ, TZP)?

Задание 6. Варьируя параметры `DM.NumberPulay` и `DM.MixingWeight`, установите оптимальные значения этих переменных с точки зрения ско-

рости сходимости (рассмотрите любую из нанотрубок).

Задания 1-6 являются неполным списком вопросов, на которые исследователь должен ответить перед началом масштабных расчетов свойств определенной модели. Выбор оптимальных параметров может существенно сократить время работы и повысить достоверность результатов.

2. Расчет физико-химических свойств

2.1. Зонная структура

SIESTA может использовать два метода для определения электронной структуры системы. Одним из них является стандартная диагонализация, которая работает для всех систем и имеет кубическую зависимость сложности алгоритма от размера модели. Другой основан на минимизации функционала на наборе локализованных функций. Время расчета во втором случае линейно зависит от размера системы, но подходит он только для систем с четко разделенными занятыми и пустыми состояниями. Выбор метода расчета осуществляется инициализацией строковой переменной `SolutionMethod`. По умолчанию используется диагонализация (`diagon`). Для активации альтернативного метода нужно присвоить этой переменной значение `OrderN`. Расчет зонной структуры выполняется после завершения оптимизации геометрии. Результат записывается в файл `SystemLabel.bands`.

Строка `BandLinesScale` определяет масштаб k -векторов, заданных в `BandLines` и `Band-Points`. Возможные значения:

- `pi/a`: координаты k -векторов задаются в декартовой системе координат, в единицах π/a , где a — постоянная решетки;
- `ReciprocalLatticeVectors`: k -вектора задаются в векторах обратного пространства.

Значение по умолчанию: `pi/a`. Необходимо явно определить параметр `LatticeConstant` в FDF файле.

Блок данных `BandLines` задает линии, вдоль которых рассчитываются энергии (обычно вдоль направлений высокой симметрии). Пример для решетки ГЦК:

```
%block BandLines
1 1.000 1.000 1.000 L \# Начало в точке L
20 0.000 0.000 0.000 \Gamma \# 20 точек от L до $\Gamma$
25 2.000 0.000 0.000 X \# 25 точек от $\Gamma$ до X
30 2.000 2.000 2.000 \Gamma \# 30 точек от X до $\Gamma$
%endblock BandLines,
```

где последняя колонка является необязательным LaTeX символом, используемым при построении зонной структуры. Если необходимы только данные точки (не линии), просто укажите 1 в первом столбце каждой

строки. Первый столбец первой строки всегда должен быть равен 1.

Для расчета зонной структуры углеродных нанотрубок в соответствующем блоке нужно указать следующее.

```
%block BandLines
1 0.000 0.000 0.000 \Gamma \# Начало в точке \Gamma
200 0.000 0.000 kmax Z \# 200 точек от \Gamma до Z
%endblock BandLines,
```

где вместо k_{\max} для кресловидных УНТ нужно подставить π/a , а для зигзагообразных — $\pi/(\sqrt{3}a)$, a — параметр трансляции в направлении оси Z .

Используйте этот блок только если `SolutionMethod = diagon`. k -точки этого блока не связаны с k -сетками, используемыми для расчета полной энергии и плотности заряда. Этот блок может быть переопределен блоком `BandPoints` (если присутствуют оба). Значение по умолчанию: не рассчитывается.

Блок данных `BandPoints` позволяет рассчитывать энергию для списка произвольных k -точек. Единицы измерения k -точек определяются переменной `BandLinesScale`. Генерируется файл `Systemlabel.bands`, который содержит координаты k -точек (в а.е.) и соответствующие энергии (в эВ). Пример:

```
%block BandPoints
0.000 0.000 0.000
1.000 0.000 0.000
0.500 0.500 0.500
%endblock BandPoints
```

Значение по умолчанию: энергии не рассчитываются.

Логическая переменная `writeKbands` определяет выводятся ли в основной выходной файл координаты k -векторов, используемых для построения зонной структуры. Эта переменная используется только если `SolutionMethod = diagon`. Значение по умолчанию: `false`.

Логическая переменная `writeBands` определяет выводятся ли в основной выходной файл собственные значения Гамильтониана, соответствующие k -векторам. Эта переменная используется только если `SolutionMethod = diagon`. Значение по умолчанию: `false`.

Пример. Расчет зонной структуры УНТ (3,3)

Приведем пример `.fdf` файла для расчета зонной структуры углеродной нанотрубки (3,3).

1	SystemName	nanotube (3,3)
2	SystemLabel	SWNT
3	NumberOfAtoms	12

```

4  NumberOfSpecies      1
5
6  %block ChemicalSpeciesLabel
7  1  6  C
8  %endblock ChemicalSpeciesLabel
9
10 PAO.BasisSize      DZP
11 PAO.EnergyShift    0.05 eV
12 MeshCutoff        300.0 Ry
13
14 %block kgrid_Monkhorst_Pack
15  16  0  0  0
16  0  16  0  0
17  0  0  256  0
18 %endblock kgrid_Monkhorst_Pack
19
20 AtomicCoordinatesFormat  NotScaledCartesianAng
21
22 XC.functional=LDA
23 XC.authors=CA
24 ZM.UnitsLength = Ang
25
26 %block Zmatrix
27 cartesian
28  1  -1.047296  -1.824411  0.009275  1  1  1
29  1  -1.977588   0.717694  0.009293  1  1  1
30  1   1.610063   1.354002  0.009392  1  1  1
31  1  -1.056152   1.819447  0.009474  1  1  1
32  1   2.103792   0.005350  0.009479  1  1  1
33  1   0.367179  -2.071354  0.009506  1  1  1
34  1   1.977588  -0.717714  1.231710  1  1  1
35  1   1.056160  -1.819432  1.231760  1  1  1
36  1  -0.367197   2.071367  1.231857  1  1  1
37  1   1.047315   1.824412  1.231857  1  1  1
38  1  -1.610057  -1.354016  1.231890  1  1  1
39  1  -2.103789  -0.005298  1.231893  1  1  1
40 %endblock Zmatrix
41
42 LatticeConstant  2.44492076099  Ang
43
44 %block LatticeParameters
45  10.0 10.0 1.0 90. 90. 90.
46 %endblock LatticeParameters
47
48 MD.TypeOfRun      cg
49 MD.NumCGsteps     360
50 MD.MaxCGDispl     0.15 Ang
51 MD.MaxForceTol    0.04 eV/Ang
52

```

```

53 BandLineStyle pi/a
54
55 %block Bandlines
56 1 0.000 0.000 0.000 \Gamma #start at gamma
57 200 0.000 0.000 8.000 X #200 k points to x
58 %endblock BandLines
59
60 WriteBands true

```

Для обработки файла System-label.bands и построения зонной структуры предназначена входящая в дистрибутив пакета SIESTA утилита Gnubands (находится в папке Util/). Эта утилита может быть скомпилирована командой

```
gfortran -o gnubands gnubands.f
```

Если при расчете пакет SIESTA сгенерировал файл SystemLabel.bands, то подготовить его к построению зонной структуры можно командой

```
gnubands < SystemLabel.bands > SystemLabel.bands.gnuplot.dat,
```

где имя второго файла может быть произвольным. При выполнении последней программы может появиться ошибка «STOP Dimensions in gnubands too small». В этом случае необходимо увеличить параметр maxb в начале файла gnubands.f, перекомпилировать утилиту и попробовать запустить команду еще раз.

График зонной структуры можно построить программой gnuplot (можно скачать по адресу <http://www.gnuplot.info/>). Для этого нужно выполнить следующие команды.

- Переходим в папку с данными
gnuplot> cd 'path';
- выполняем построение
gnuplot> plot 'SWNT33gnu.bands' using 1:2 with lines;
- выбираем диапазоны по осям x
gnuplot> set xrange [0:0.68];
- и y
gnuplot> set yrange [-10:0];
- и перестраиваем график
gnuplot> replot.

Для удобства использования результатов в публикациях зонную структуру обычно выводить в файл формата *.eps или *.png. Приведем короткий скрипт gnuplot, упрощающий эту процедуру.

```

1 set terminal pngcairo size 10cm,16cm enhanced font 'Verdana,10'
2 set output 'bandideal33.png'
3 set title 'Band structure'

```

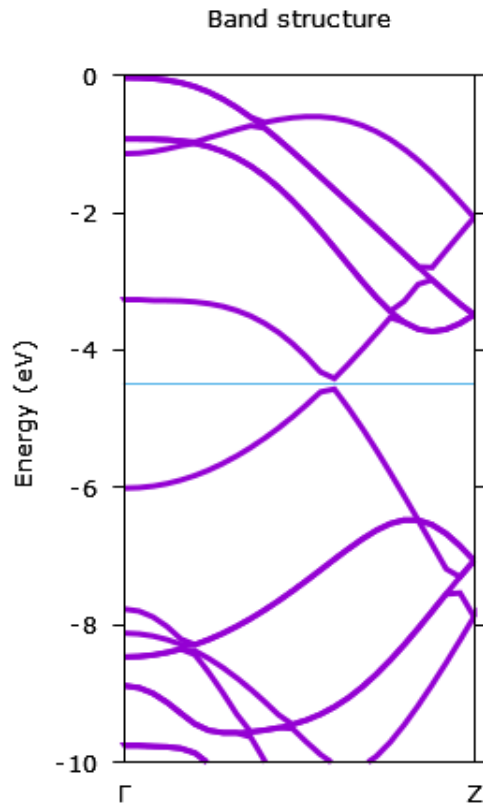


Рис. 2.1.: Зонная структура углеродной нанотрубки (3,3)

```

4 set noxzeroaxis
5 set tics out
6 set noxtics
7 set nokey
8 set yrange[ -10.00: -0.00]
9 set xrange[ 0.0000: 0.68]
10 set ylabel ' Energy (eV) '
11 set xtics ( '{/Symbol G}' 0.00, ' Z ' 0.68)
12 set multiplot
13 plot 'SWNT33gnu.bands' with lines lw 3
14 plot -4.4803 lt 3
15 set nomultiplot
16 unset output

```

На рис. 2.1 приведен результат выполнения этого скрипта.

2.2. Плотность электронных состояний

Полная плотность состояний

Плотностью электронных состояний (DOS) называется число разрешенных состояний электронов в единичном интервале энергий. Единицы измерения DOS: число состояний / эВ. DOS ($g(\varepsilon)$) нормируется следующим образом:

$$\int_{-\infty}^{+\infty} g(\varepsilon) d\varepsilon = \text{количество базисных орбиталей в расчетной ячейке.}$$

Расчет полной плотности состояний можно провести несколькими способами:

- Собственные значения гамильтона для самосопряженного решения могут быть сохранены в файл `SystemLabel.EIG` в формате, аналогичном `SystemLabel.bands`, но без информации о `kmin`, `kmax`, `emin`, `emax`. Для дальнейшей обработки и получения плотности состояний может быть использована утилита `Eig2dos` (размещена в каталоге `Util/`). Если логическая переменная `writeEigenvalues` равна `true`, собственные значения гамильтониана записываются в главный выходной файл. Если она равна `false`, эта информация записывается в файл `Systemlabel.EIG`. Эта опция может быть использована только если переменная `solutionMethod` равна `diagon`. По умолчанию переменная `writeEigenvalues` равна `false`.
- В качестве побочного продукта при расчете парциальной плотности состояний.
- В качестве одного из файлов, генерируемых утилитой `Util/COOP/mpgor` во время анализа электронной структуры. Этот метод позволяет указывать диапазон энергий и разрешение, без повторного запуска SIESTA.

Парциальная плотность остояний

Существует два варианта получения парциальной плотности состояний: плотности состояний, относящейся к определенному атому или группе атомов.

- Используя приведенные ниже опции.
- Используя утилиту `Util/COOP/mpgor` для анализа электронной структуры в режиме PDOS. Этот метод позволяет задавать диапазон энергий, орбитали и разрешение без перезапуска расчета SIESTA.

Блок `ProjectedDensityOfStates` позволяет записать полную плотность состояний (DOS) и парциальную плотности состояний (PDOS) в заданном интервале энергий в файлы `SystemLabel.DOS` и `SystemLabel.PDOS`, соответственно. В блоке должна быть одна строка с энергиями диапазона для PDOS (по отношению к программному нулю), ширина пика (энергия) для уширения собственных значений, число точек, единицы измерения энергии. Пример:

```
%block ProjectedDensityOfStates
-20.00 10.00 0.200 500 eV
%endblock ProjectedDensityOfStates
```

По умолчанию PDOS создается для той же сетки точек в обратном пространстве, которая используется для самосогласованного расчета. Тем не менее, отдельный набор k -точек, может быть получен с помощью одного из вариантов: `PDOS.kgrid_cutoff` или `PDOS.kgrid_Monkhorst_Pack`. Формат этих опций точно такой же, как у `kgrid_Cutoff` и `kgrid_Monkhorst_Pack`, соответственно. Две энергии диапазона должны быть упорядочены по возрастанию. DOS записывается в файл под названием `SystemLabel.DOS`. Формат этого файла: значение энергии, DOS (спин вверх), DOS (спин вниз).

Парциальная плотность состояний для всех орбиталей в элементарной ячейке записывается последовательно в файл с именем `SystemLabel.PDOS`. Этот файл структурирован с использованием пробелов и XML тегов. Также создается машиночитаемый (но не очень удобный для восприятия человеком) XML файл `pdos.xml`. Оба они могут быть обработаны с помощью программы в `Util/pdosxml`. Файл `.PDOS` может быть обработан утилитой, написанной Андреем Постниковым¹. Последняя утилита является более удобной, так как не требует повторной компиляции для вывода парциальной плотности состояний для других атомов или состояний электронов. Ниже приведена последовательность действий при работе этими утилитами.

Утилита `fmpdos`

Утилита скачивается в виде одного файла с исходным кодом на языке `fortran`. Она может быть откомпилирована, например, с помощью `gfortran` командой

```
gfortran fmpdos.f -o fmpdos.
```

При запуске программа запрашивается имя файла с парциальной плотностью состояний, находящегося в папке с исполняемым файлом. Ука-

¹Исходный код программы свободно доступен и находится по адресу <http://www.home.uni-osnabrueck.de/apostnik/download.html>.

зывается выходной файл. После этого указываются атомы, для которых выводится плотность состояний. Можно указать номер атома (или все атомы) или определенный тип химического элемента. После этого указывается главное квантовое число.

```
./fmpdos
Input file name (PDOS):
C168.PDOS
Output file name :
out
Extract data for atom index (enter atom NUMBER, or 0 to select all),
or for all atoms of given species (enter its chemical LABEL):
0
Extract data for n= ... (0 for all n ):
0
```

Локальная плотность состояний

Локальная плотность состояний (LDOS) — это DOS, взвешенная по амплитуде соответствующих волновых функций, в различных точках пространства — функция энергии и положения. SIESTA может выводить LDOS проинтегрированную в некотором диапазоне энергий. Эта информация может быть использована для получения простых изображений STM в приближении Terso-Hamann (см. Util/STM/simple-stm).

Блок `LocalDensityOfStates` позволяет записать LDOS, проинтегрированную между двумя заданными энергиями на сетке, используемой DHSCF, в файл `SystemLabel.LDOS`. Блок должен состоять из одной строки с энергиями диапазона интегрирования LDOS (по отношению к программному нулю) и их единиц измерения. Примером может служить:

```
%block LocalDensityOfStates
-3.50 0.00 eV
%endblock LocalDensityOfStates.
```

Значение по умолчанию: LDOS не рассчитывается и не записывается.

Пример. Локальная плотность электронных состояний молекулы этана

Рассмотрим молекулу этана (C₂H₆). В качестве стартовой конфигурации можно взять следующую.

```
1 NumberOfAtoms      8
2 NumberOfSpecies    2
3
4 %block ChemicalSpeciesLabel
5 1 1 H
6 2 6 C
```

```

7  %endblock ChemicalSpeciesLabel
8
9  ZM.UnitsLength = Ang
10
11 %block Zmatrix
12 cartesian
13 2  -0.588625  -0.483116  0.000000  1  1  1
14 2   0.588625   0.483116  0.000000  1  1  1
15 1  -0.219340  -1.533068  0.000000  1  1  1
16 1  -1.215419  -0.319453  -0.905047  1  1  1
17 1  -1.215419  -0.319453   0.905047  1  1  1
18 1   0.219340   1.533068  0.000000  1  1  1
19 1   1.215419   0.319453  -0.905047  1  1  1
20 1   1.215419   0.319453   0.905047  1  1  1
21 %endblock Zmatrix
22 #Поскольку мы хотим рассмотреть изолированную молекулу,
23 #то параметры трансляции вдоль осей координат должны быть
24 #такими большими, чтобы взаимодействие <<соседних>> молекул
25 #не оказывало влияния на результат.
26 LatticeConstant      5.00 Ang
27 %block LatticeParameters
28 1.0 1.0 1.0 90. 90. 90.
29 %endblock LatticeParameters

```

После окончания расчета запустим утилиту rho2xsf. Укажем Specify SystemLabel (or 'siesta' if none): C112
Now define the grid cell for your XCrysDen plot.
Note that it can be arbitrarily chosen with respect to the Siesta simulation cell, and it needs not to be orthogonal. We'll define it by the origin point and three spanning vectors. They can be given in Bohr or Ang.
Would you use Bohr (B) or Ang (A) ? A
Enter origin point in Ang : -2.5 -2.5 -2.5
Enter 1st spanning vector in Ang : 5 0 0
Enter 2nd spanning vector in Ang : 0 5 0
Enter 3rd spanning vector in Ang : 0 0 5
The box contains 8 atoms.
Now define the grid. If you want it two-dimensional, give 1 as number of grid points along one spanning vector.
Enter number of grid points along three vectors: 100 100 100
Add grid property (LDOS, RHO, ...; or BYE if none): LDOS
Found and opened: C112.LDOS
mesh0 = (40 40 40), nspin= 1
For is=1: max. grid value = 0.00000E+00 at iix,iiy,iiz= 1 1 1
For is=1: min. grid value = 0.00000E+00 at iix,iiy,iiz= 1 1 1
Add grid property (LDOS, RHO, ...; or BYE if none): BYE

A wild guess! There is no file C112.BYE; close XSF and quit.

Результат работы программы можно визуализировать с помощью XCrysden (см. раздел 3.3).

2.3. Оптические свойства

Диэлектрическая проницаемость материала (ε) описывает поляризацию (электрическую поляризуемость) и абсорбционные свойства материала в зависимости от частоты приложенного электрического поля. Диэлектрическая проницаемость состоит из двух частей: ε_1 и ε_2 . Величина ε_1 показывает, насколько сильно поляризуется материал в электрическом поле, а ε_2 характеризует поглощение энергии внешнего поля.

Комплексная диэлектрическая проницаемость вводится следующим образом:

$$\varepsilon(\lambda) = \varepsilon_1(\lambda) + i\varepsilon_2(\lambda).$$

ε_1 и ε_2 связаны друг с другом соотношением Крамерса–Кронига — интегральной связью между действительной и мнимой частями любой комплексной функции, аналитичной в верхней полуплоскости. Согласно ему, для комплексной функции $\chi(\omega) = \chi_1(\omega) + i\chi_2(\omega)$ комплексной переменной ω , аналитичной в верхней полуплоскости ω и стремящейся к нулю при $|\omega| \rightarrow \infty$, соотношения Крамерса – Кронига записываются следующим образом:

$$\chi_1(\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\chi_2(\omega')}{\omega' - \omega} d\omega'$$

и

$$\chi_2(\omega) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\chi_1(\omega')}{\omega' - \omega} d\omega',$$

где интегралы берутся в смысле главного значения (по Коши). Видно, что $\chi_1(\omega)$ и $\chi_2(\omega)$ не являются независимыми, а значит, полная функция может быть восстановлена, если задана только её действительная или мнимая часть.

Если во входном файле SIESTA логической переменной `OpticalCalculation` присвоено значение `true`, мнимая часть диэлектрической функции будет сохранена в файл с названием `Systemlabel.EPSIMG`. Расчет выполняется с помощью простейшего подхода, основанного на дипольных матричных элементах перехода между различными собственными функциями самосогласованного Гамильтониана. Для молекул

расчеты производятся с использованием матричных элементов в координатном представлении, а для твердых тел расчет осуществляется в импульсном пространстве. По умолчанию диэлектрическая функция не рассчитывается (переменная `OpticalCalculation` равна `false`).

Диапазон энергии, в котором рассчитывается ε можно задать двумя действительными значениями энергии: `Optical.EnergyMinimum` и `Optical.EnergyMaximum`. Эти параметры имеют значения по умолчанию `0 Ry` и `10 Ry`, соответственно.

Действительная величина `Optical.Broaden` задает гауссово уширение, применяемое к рассчитанным свойствам. Значение по умолчанию: `0 Ry`.

Для учета переоценки ширины запрещенной зоны в методе DFT, может быть применен сдвиг незаполненных состояний, известный как `scissor operator`. Величина сдвига задается параметром `Optical.Scissor`. Этот сдвиг применяется только к оптической части расчета. Значение по умолчанию: `0 Ry`.

Целочисленное значение `Optical.NumberOfBands` задает количество полос (band), которые включены в расчет оптических свойств. Очевидно, что оно должно быть больше, чем число занятых полос и не должно превышать количество базисных функций (которые определяют количество доступных незаполненных зон). Хотя включение в расчет всех полос позволит получить наиболее точные результаты, это увеличит длительность вычислений. Значение по умолчанию: все полосы.

Блок данных `Optical.Mesh` содержит 3 числа, определяющие размер сетки, используемой для интегрирования по зоне Бриллюэна. Например:

```
1 %block Optical.Mesh
2 5 5 5
3 %endblock Optical.Mesh.
```

Эти три значения представляют собой число точек сетки в направлении каждого вектора обратной решетки. Значение по умолчанию: предполагается наличие только одной k -точки.

Если логическая переменная `Optical.OffsetMesh` равна `true`, то сетка смещена от гамма точки для нечетного числа точек. Значение по умолчанию: `false`.

Строка `Optical.PolarizationType` позволяет выбрать один из трех вариантов поляризации воздействующего поля:

- `polarized`: предполагает использование электрического поля в заданном направлении;
- `unpolarized`: предполагает распространение света в заданном направ-

- ленин;
- polycrystal.

Значение по умолчанию: polycrystal.

Блок данных `Optical.Vector` содержит 3 числа, определяющие направление вектора электрического поля, либо распространения света, для поляризованного (polarized) и неполяризованного (unpolarized) расчетов, соответственно. Пример:

```
%block Optical.Vector  
1.0 0.0 0.5  
%endblock Optical.Vector.
```

Значение по умолчанию: Empty.

Обработка результатов

В директории `utils/optical` содержатся две программы (`input.f` и `optical.f`) для расчета различных оптических свойств. Обе эти программы могут быть скомпилированы, например, командами

```
f77 -o input input.f  
f77 -o optical optical.f
```

Для их работы необходим файл `Systemlabel.EPSIMG`, который генерируется пакетом SIESTA. Этот файл содержит мнимую часть диэлектрической проницаемости (см. стр. 59) в диапазоне энергий, определенном во входном файле SIESTA.

Первая программа создает входной файл для программы `optical`. На вход программе нужно подать файл `Systemlabel.EPSIMG`. Для этого достаточно запустить

```
input < Systemlabel.EPSIMG
```

Программа создаст файл `e2.dat`, если расчет не спинполяризованный, или два файла (`e2.dat.spin1` и `e2.dat.spin2`) для спинполяризованного случая.

Если правило F -суммы (см. стр. 62) не выполняется на величину, большую, чем определенный процент, указанный в программе с помощью параметра `THRESHOLD`, мнимая часть диэлектрической проницаемости будет дополнена хвостом в форме

$$\frac{C}{w^p},$$

где C и p подобраны так, чтобы обеспечить непрерывность и выполнение правила сумм. Это не очень достоверный метод, и в некоторых случаях может приводить к неверным результатам. Эта корректировка может быть деактивирована путем установки `THRESHOLD` равным нулю.

Программа *optical* считывает файл с названием *e2.dat*, содержащий мнимую часть диэлектрической проницаемости, и создает файлы *e1.interband.out* (содержит действительную часть диэлектрической проницаемости), *e2.interband.out* (мнимая часть, вычисленная по данным в файле *e1.interband.out*; служит проверкой качества преобразования Крамерса–Кронига).

Для металлов необходимо также учесть вклад Друде (связан с внутризонными переходами) в виде

$$\frac{\omega_p^2}{\omega(\omega + i\gamma)},$$

где ω_p^2 рассчитывается SIESTA и содержится в начале файла *Systemlabel.EPSIMG*; γ — эмпирический параметр, обратное время релаксации.

Также создаются файлы:

- *epsilon_real.out* — действительная часть диэлектрической проницаемости;
- *epsilon_img.out* — мнимая часть диэлектрической проницаемости;
- *refrac_index.out* — показатель преломления;
- *absorp_index.out* — коэффициент ослабления (extinction coefficient);
- *absorp_coef.out* — коэффициент поглощения в см^{-1} ;
- *reflectance.out* — отражательная способность;
- *conductivity.out* — оптическая проводимость в $(\text{Ом}\cdot\text{м})^{-1}$.

Правило F -сумм

Правилом F -сумм (правилом Thomas-Reiche-Kuhn) для электрона с гамильтонианом

$$\hat{H} = \frac{\hat{p}^2}{2m_e} + V(\vec{r})$$

называется следующее выражение:

$$\sum_k (E_k - E_0) |\langle 0 | \vec{r} | k \rangle|^2 = \frac{3\hbar}{2m_e},$$

где $\langle 0 |$ — основное состояние, $|k\rangle$ — состояние с номером k .

Электропроводность конденсированных сред

На электропроводность в конденсированных средах часто оказывают влияние различные механизмы рассеяния электронов, которые сложно описать в рамках теории функционала электронной плотности. Для

металлов наибольший вклад в электропроводность вносит подвижность почти свободных электронов, находящихся на поверхности Ферми, которая, в свою очередь, зависит от межзонных электронных возбуждений (оптических переходов). В идеальном случае, электропроводность при постоянном токе равна мнимой части диэлектрической проницаемости $\varepsilon_2(\omega)$, взятой в пределе $\omega \rightarrow 0$. Этот формализм называется формализмом Кубо–Гринвуда [25, 26]. Существуют несколько граничных случаев, в которых этот формализм дает весьма точные результаты. Во-первых, это идеально чистые металлы при очень низких температурах — системы, в которых наличие других механизмов рассеяния, кроме чисто электронных возбуждений, является пренебрежимо малым. Во-вторых, это случай жидкого или газообразного металла, в котором структурный беспорядок является основной причиной рассеяния электронов, позволяя нам не принимать во внимание другие механизмы рассеяния, такие, как фононы или примесные атомы. К настоящему времени проведено много исследований, которые показывают, что при повышении температуры повышается точность первопринципного расчета электропроводности системы.

2.4. Теория Ландауэра - Буттикера

Прозрачную квантовомеханическую формулировку электронного транспорта в небольших проводниках была впервые предложена в 1957 году Ландауэром (работа переиздана в 1996 году [27]), который предложил простую формулу, которая установила связь между вероятностью передачи электрона и электронной проводимостью в одномерных структурах. Идея Ландауэра была позднее обобщена Буттикером [28] на случай многоканальных-мультиконтактных (multi-channel-multi-probe) устройств. С тех пор большинство теоретических работ по электронной проводимости в нано-масштабе были основаны на этом формализме (см. [29] и ссылки в ней).

Если приложить разность потенциалов U к металлическому или полупроводниковому объекту, электроны в окрестности уровня Ферми приобретут ускорение. Во время движения электроны рассеиваются на примесях или колебаниях решетки (фононах), и, если рассеяние является неупругим, то они могут потерять часть приобретенного ими импульса. В результате, если материал достаточно длинный, то процесс ускорения и рассеяния повторяется и несет электроны в направлении электрического поля, создавая дрейфовый ток I . Уравнение, которое описывает эту ситуацию, является хорошо известным законом Ома: $U = RI$, где R есть сопротивление, вызванное рассеянием, которое пропорционально длине устройства.

При рассмотрении очень малых устройств можно предположить, что рассеяния на примесях не происходит. Точнее, длина свободного пробега электрона (10-50 нм для металлических макроскопических тел при комнатной температуре [30]) больше либо соизмерима с длиной устройства. Электроны могут проникать через образец без рассеяния, и электрическое сопротивление перестает зависеть от длины материала. Этот тип проводимости называется баллистическим транспортом электронов.

Простейшей моделью квантовой проводимости является модель в терминах одноэлектронных волновых функций рассеянных на пространственно зависящем потенциале. Можно предположить, что потенциал находится между двух резервуаров электронов, каждый из которых испускает частицы с равновесным распределением в область рассеяния через идеально проводящие электроды. Резервуары, в общем случае, имеют разные химические потенциалы, μ_L и μ_R , их разность, $\mu_L - \mu_R$, описывает приложенную разность потенциалов. Электрический ток, проходящий через устройство, соответствует потоку электронов, проходящих между резервуарами.

В простейшем случае, электроны движутся когерентно по всему

устройству, испытывая только упругие столкновения в области рассеяния. Любая потеря когерентности в результате неупругих столкновений требует описания на более сложном уровне. Для дополнительного упрощения будем считать, что резервуары электронов имеют определенные характеристики: (I) резервуары безотражательные, т.е. электроны, входящих в резервуар от электродов всегда принимаются, и (II) резервуары макроскопические в том смысле, что химические потенциалы поддерживаются на уровне μ_L и μ_R независимо от входа и выхода электронов через электроды. Два предположения подразумевают, что резервуары могут быть описаны независимым равновесным распределением Ферми-Дирака, так что вероятность того, что электрон обладает энергией E имеет вид функции Ферми

$$f(E - \mu) = \frac{1}{\exp^{(E-\mu)/k_B T}}, \quad (2.1)$$

где k_B - постоянная Больцмана, T - температура, μ - химический потенциал резервуара.

Предположим теперь, что электрон с энергией $\mu_R < E < \mu_L$ оказывается на левом электроде. Этот электрон, описывается бегущей в правом направлении волной Блоха в одной из мод левого электрода. Обозначим $1D$ волновое число, энергию и групповую скорость этой моды через k , E_k , и $V_k = \frac{1}{\hbar} \frac{dE_k}{dk} > 0$, соответственно. Когда электрон попадает в центральную область модели Ландауэра, он подвергается упругому рассеянию и с некоторой вероятностью T_k передается в правый электрод. Затем он проходит через правый электрод и, наконец, входит в правый резервуар. Если он не передается в правый электрод, он отражается обратно в левый электрод с вероятностью R_k ($T_k + R_k = 1$) и в конечном итоге возвращается в левый резервуар. Ток электронов в соответствии с такой последовательностью событий дается первым слагаемым в формуле

$$I = \frac{e}{L} \sum_k f(E_k - \mu_L) V_k T_k + \frac{e}{L} \sum_{-k'} f(E_{-k'} - \mu_R) V_{-k'} T_{-k'}, \quad (2.2)$$

где $\frac{e}{L}$ вводится для нормировки в $1D$ объеме L . Второе слагаемое соответствует обратной ситуации, когда электроны с волновым числом $-k'$ проходят от правого резервуара по направлению к левому ($V_{-k'} < 0$).

Чтобы прийти к формуле 2.2 мы рассматривали вероятности передачи T_{\pm} электронов с конкретными волновыми векторами $\pm k$. Таким образом, суммарные коэффициенты передачи для определенной энергии электронов E , могут быть выражены как

$$T_{\pm}(E) = \sum_{\pm k} T_{\pm k} \delta(E_k - E),$$

для движущихся влево (-) и вправо (+) соответственно, а δ - дельта-функция Дирака. Как следствие наличия симметрии относительно обращения времени эти коэффициенты должны быть одинаковыми: $T^-(E) = T^+(E) = T(E)$. Тогда уравнение 2.2 можно переписать в виде интеграла по энергии, преобразуя сумму в интеграл, то есть, $\sum_k \rightarrow 2 \cdot \frac{L}{2\pi} \int dk$, где множитель 2 учитывает спиновое вырождение, и после применения $dk = 2\pi dE_k/hV_k$ получим формулу

$$I = \frac{2e}{h} \int_{-\infty}^{\infty} T(E)[f(E_k - \mu) - f(E - \mu_R)]dE, \quad (2.3)$$

которая является ключевой в теории Ландауэра-Буттикера.

Давайте теперь посмотрим, что происходит при нулевой температуре в рамках изложенного выше формализма. Отметим, что в этом пределе функция Ферми в уравнении 2.1 становится ступенчатой функцией. Если мы также предполагаем, что разность $\mu_R - \mu_L$ настолько мала, что $T(E) \rightarrow T(E_F)$ не зависит от E , то уравнение 2.3 может быть заменено

$$I = \frac{2e}{h} T(E_F) \int_{\mu_R}^{\mu_L} dE = \frac{2e}{h} T(E_F)(\mu_L - \mu_R), \quad (2.4)$$

для случая $\mu_R < \mu_L$.

Так как разность потенциалов между двумя электродами составляет $V_b = (\mu_R - \mu_L)/e$, мы приходим к следующему выражению для проводимости

$$G = \frac{I}{V_b} = \frac{2e^2}{h} T(E_F), \quad (2.5)$$

которое согласуется с формулой Ландауэра 1957 года [27].

Формула 2.5 предполагает, что максимальная проводимость одного канала с двумя спиновыми состояниями равна $G_0 = \frac{I}{V_b} = \frac{2e^2}{h} T(E_F)$, в случае баллистической проводимости ($T = 1$). В общем случае проводимость канала отличается от G_0 так как $T \neq 1$. Полная проводимость G является суммой вкладов всех возможных каналов, вклад каждого из которых меньше или равен G_0 .

2.5. TranSIESTA

TranSIESTA – это метод расчета электронной структуры открытых систем, образованных конечной подсистемой, зажатой между двумя бесконечными проводящими контактами (метод подробно описан в [31]). Расчеты по методу TranSIESTA включают вычисление электронной плотности по DFT гамильтониану с использованием методов функций Грина, вместо обычной процедуры диагонализации. Расчет транспортных свойств включает вычисление характеристик электродов и области рассеяния (исследуемый объект + электроды). При расчете электродов SIESTA генерирует файлы `<SystemLabel>.TSHS`, расположение которых необходимо указывать в `fdf` файле при расчете области рассеяния.

Расчет TranSIESTA для области рассеяния начинается с обычной процедуры SIESTA, генерирующей матрицу плотности (DM) с использованием обычной схемы Кона-Шэма для периодических систем. Это решение используется в качестве первоначального для самосогласованного цикла метода функций Грина. SIESTA хранит матрицу плотности в файле с расширением `.DM`. В случае расчета TranSIESTA матрица плотности записывается в файл с именем `<SystemLabel>.TSDE`. При повторном запуске расчета информация из файла `.TSDE` принимается в качестве начального приближения матрицы плотности. В этом случае не выполняется начальный запуск процедуры SIESTA. При увеличении приложенной к области рассеяния разности потенциалов, хорошей практикой является использование файла `.TSDE`, полученного при меньшем напряжении. Важно отметить, что файл `<SystemLabel>.TSDE` имеет тот же формат, что и `<SystemLabel>.DM` (с дополнительной информацией о концах модели). Таким образом, можно, например, использовать DENCHAR (см. стр. 97) для анализа неравновесной плотности заряда.

Результатом расчета TranSIESTA, как и расчета SIESTA, является равновесная матрица плотности. Транспортные свойства вычисляются с помощью программы `tbtrans` (расположена в каталоге `Util/TBTrans`). Эта программа на основании информации из файлов `.TSHS` для области рассеяния и электродов позволяет рассчитать спектр пропускания и ток. Входным файлом для программы `tbtrans`, как правило, служит `fdf` файл, использовавшийся для расчетов TranSIESTA, с дополнительными опциями `tbtrans`.

Код TranSIESTA предполагает, что первые атомы, указанные в файле `fdf`, соответствуют левому электроду, а последние – к правому. Направлением протекания тока считается ось `z`. Атомы левого электрода должны иметь `z` компоненты координаты меньше, чем атомы правого электрода. Электроды не должны находиться в области протекания тока (по `z`

компонентам координат атомов). Также важно учитывать, что атомные позиции электродов в .fdf файле должны быть получены трансляцией атомных структур, использованных для расчета свойств электродов. Кроме того, можно использовать буферные атомы. Эта возможность полезна при моделировании с различными левым и правым электродами. В этом случае TranSIESTA не будет учитывать буферные атомы, которые используются только для начального расчета SIESTA для получения более совершенного контакта электродов. Количество буферных атомов определяется параметрами `TS.BufferAtomsLeft` (значение по умолчанию: 0) и `TS.BufferAtomsRight` (значение по умолчанию: 0). Целочисленный параметр `TS.BufferAtomsLeft` задает число атомов, которые не учитываются при расчете TranSIESTA (начиная с первого). Целочисленный параметр `TS.BufferAtomsRight` задает число атомов, которые не учитываются при расчете TranSIESTA (начиная с последнего).

При расчете TranSIESTA предполагается наличие периодических граничных условий в x и y направлениях. Разбиение k -пространства в двух направлениях должно быть таким же, как при расчете электродов. В противном случае расчет будет прерван с ошибкой. На практике это означает, что первая и вторая строка в блоке `kgrid_Monkhorst` в этих расчетах должны совпадать. При расчете `tbtrans` так же необходимо указывать разбиение обратного пространства с использованием блока `kgrid_Monkhorst` или `TBT_kgrid_Monkhorst_Pack` (предпочтительно). Причем, это разбиение может отличаться от указанного при расчете TranSIESTA. Дело в том, что сходимость функции пропускания относительно разбиения обратного пространства может быть медленнее, чем для матрицы плотности.

Электродами могут служить поверхности, нанопроволоки, нанотрубки или даже атомные цепочки. Размер элементарной ячейки вдоль оси z должен быть достаточно большим, чтобы орбитали в пределах элементарной ячейки взаимодействовали только с одной ближайшей ячейкой (размер элементарной ячейки может быть оценен пространственным размером орбитальных базисных функций). TranSiesta будет предупреждать, если это требование не соблюдается.

Приведенные ниже опции используются только для расчета области рассеяния при помощи пакета TranSIESTA. Строковый параметр `SolutionMethod` должен быть установлен равным «`transiesta`». Логическая переменная `TS.SaveNS` указывает, необходимо ли сохранять Гамильтониан в файл с расширением `.TSHS`. Она должна равняться `true` при расчете Гамильтониана электродов (значение по умолчанию). Эта переменная так же должна равняться `true`, если планируется дальнейший анализ результатов программой `tbtrans`. Обычно в цикле самосогласования матрицы плотности предшествующих итераций используются для расчета

стартовой матрицы плотности для последующей итерации. Если логический параметр `TS.MixN` равен `true` (значение по умолчанию `false`), вместо указанного выше приема, проводится построение Гамильтониана на основе Гамильтонианов предыдущих итераций. Это может улучшить сходимость. Если логическая переменная `TS.UpdateDMCRonly` равна `true` (значение по умолчанию), в ходе цикла самосогласования TranSIESTA (функции Грина) обновляются только элементы матрицы плотности контакта. Функции Грина электродов зависят от атомной структуры и приложенной разности потенциалов. Если логическая переменная `TS.CalcGF` равна `false` (значение по умолчанию `true`), в качестве начального приближения будут использоваться функции Грина, полученные ранее (файлы `.GF`).

Для указания расположения файлов с данными об электродах и числе атомов в них существуют следующие шесть переменных. Строковая переменная `TS.HSfileLeft` (`TS.HSfileRight`) задает имя выходного файла `.TSHS`, полученного при расчете свойств левого (правого) электрода (значение по умолчанию отсутствует). Строковая переменная `TS.GFfileLeft` (`TS.GFfileRight`) задает имя `.GF` файла левого (правого) электрода. Если этот файл не будет найден, он будет создан. Значение по умолчанию `Left.GF` (`Right.GF`). Число атомов, учитываемых в левом (правом) электроде, задается целочисленным параметром `TS.NumUsedAtomsLeft` (`TS.NumUsedAtomsRight`). Оно должно быть меньше или равно (значение по умолчанию) числу атомов в элементарной ячейке левого (правого) электрода. В первом случае, учитываются последние `TS.NumUsedAtomsLeft` (`TS.NumUsedAtomsRight`) атомов, а атомы в электроде должны быть упорядочены по возрастанию z -координаты.

Для расчета матрицы плотности D в методе TranSIESTA необходимо вычислить интеграл

$$D = \frac{1}{2\pi i} \int_{-\infty}^{\infty} G(\varepsilon) d\varepsilon,$$

где G – функция Грина (см. [31]). Из теории функций комплексного переменного известно, что такой интеграл несобственный интеграл можно вычислить, понимая его как часть интеграла по замкнутому контуру, состоящему из отрезка $[-R, R]$ в пределе R , стремящегося к бесконечности, и дуги окружности (см. рис. 2.2). Перечислим параметры входного файла TranSIESTA, характеризующие этот контур.

Параметр `TS.ComplexContour.Emin` задает положение нижней границы контура. Она должна лежать ниже энергии основного состояния модели (значение по умолчанию: -3.0 Ry). Число точек интегрирования на дуге окружности (от `TS.ComplexContour.Emin` до $E_F = 0$) задается параметром `TS.ComplexContour.NumCircle` (значение по умолчанию: 24). Количество

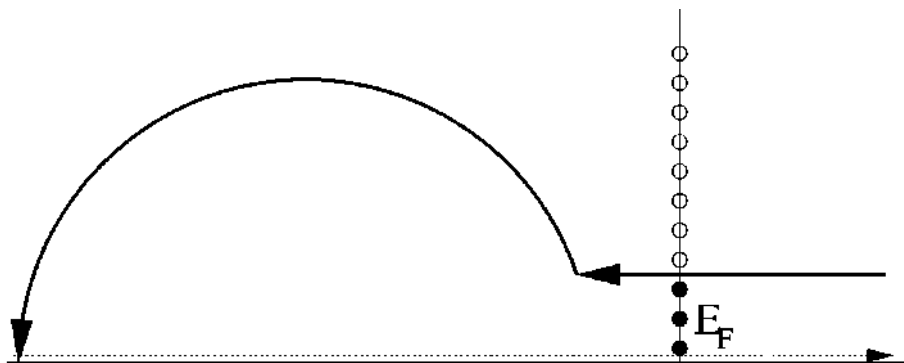


Рис. 2.2.: К пояснению схемы интегрирования

точек на линейной части контура задается целочисленным параметром `TS.ComplexContour.NumLine` (значение по умолчанию: 6). Целочисленный параметр `TS.ComplexContour.NumPoles` определяет количество полюсов (ближайшие к энергии Ферми полюса показаны закрашенными кругами на рис. 2.2), охватываемых контуром (значение по умолчанию: 6).

Утилита `tbtrans`

Код программы `tbtrans` расположен в папке `Util/TBTrans`. Эта программа используется для получения транспортных свойств в процессе обработки выходных файлов расчета `TranSIESTA`. Для запуска программе требуются `.TSHS` файлы электродов (возможно один файл, если электроды одинаковые), и `.TSHS` файл области рассеяния. Расположение этих файлов задается уже упомянутыми переменными `TS.HSfileLeft` и `TS.HSfileRight`, а так же строковой переменной `TS.TBT.HSfile`, которая указывает расположение `.TSHS` файла области рассеяния (значение по умолчанию: `SystemLabel.TSHS`).

Шкала энергии в программе `tbtrans` смещена таким образом, что уровень Ферми системы в отсутствии приложенного напряжения соответствует нулю. При конечном напряжении, энергия Ферми левого электрода принимается равной $V/2$, а правого — $-V/2$.

Разность потенциалов задается переменной `TS.Voltage`. Диапазон энергий и число точек для расчета функции пропускания определяются переменными

`TS.TBT.Emin`: минимальное значение энергии, для которого рассчитывается функция пропускания (значение по умолчанию: -2.0 eV).

`TS.TBT.Emax`: максимальное значение энергии, для которого вычисляется функция пропускания (значение по умолчанию: 2.0 eV).

`TS.TBT.NPoints`: число точек разбиения в интервале от `TS.TBT.Emin` до

TS.TBT.Emax (значение по умолчанию: 100).

Обратите внимание, что важно указывать напряжение, так как эта информация не хранится в файле .TSHS. Ток будет рассчитываться с использованием полученной функции пропускания, поэтому нужно указать окно энергий больше, чем приложенное напряжение.

Разбиение обратного пространства по k -точкам задается блоком kgrid_Monkhorst_Pack_block. Усредненная по k -точкам функция пропускания выводится в файл <SystemLabel>.AVTRANS.

Код программы tbtrans распараллелен только по k -точкам. По этой причине число расчетных узлов (ядер) не должно превышать число k -точек.

Пример. Расчет проводимости УНТ (3,3)

Так как УНТ могут проводить электрический ток в баллистическом режиме (а следовательно их сопротивление не зависит от длины), сопротивление нанотрубки может оказаться существенно меньше, чем сопротивление в области контактов с электродами. Поэтому при использовании УНТ очень важно создать их надежный контакт с электродами.

Согласно теории Ландауэра (см. приложение 2.4) в баллистическом режиме проводимость определяется формулой $G = \frac{2e^2MT}{h}$, где M — количество разрешенных каналов проводимости, T — квантово-механический коэффициент пропускания, который характеризует вероятность перехода электрона от одного электрода к другому. Множитель «2» в этой формуле появляется из-за спинового вырождения. На величину « T » могут оказывать влияние дефекты в УНТ и изгибы УНТ. Две дисперсионные кривые, пересекающиеся на уровне Ферми УНТ с металлическим типом проводимости свидетельствуют о наличии двух каналов проводимости. Таким образом, наименьшее возможное сопротивление УНТ, включая контактное сопротивление, составляет 6,47 кОм.

Для исключения из рассмотрения контактных явлений в качестве электродов использовались УНТ с теми же индексами хиральности, что и у исследуемого образца (в качестве объекта моделирования выбирался фрагмент УНТ (3,3) длиной 0.24 нм (12 атомов углерода)).

Необходимый для расчета свойств электрода fdf файл мало отличается от использованных нами ранее fdf файлов. Стоит лишь обратить внимание на увеличение количества k -точек. Для расчета используется пакет TranSIESTA.

1	SystemName	elecSWNT33
2	SystemLabel	elecSWNT33
3	NumberOfAtoms	12

```

4 | NumberOfSpecies      1
5 | WriteMullikenPop    1
6 |
7 | %block ChemicalSpeciesLabel
8 |   1  6  C
9 | %endblock ChemicalSpeciesLabel
10 |
11 | PAO.BasisSize      DZ
12 | PAO.SplitNorm      0.59
13 | PAO.EnergyShift    0.05 eV
14 | PAO.SplitNorm      0.57896
15 | MeshCutoff         300.0 Ry
16 | SpinPolarized      .true.
17 |
18 | %block kgrid_Monkhorst_Pack
19 |   4  0  0  0
20 |   0  4  0  0
21 |   0  0 120  0
22 | %endblock kgrid_Monkhorst_Pack
23 |
24 | ZM.UnitsLength = Ang
25 | LatticeConstant    2.44492076099 Ang
26 |
27 | %block LatticeParameters
28 |   10.0 10.0 1.0 90. 90. 90.
29 | %endblock LatticeParameters
30 |
31 | # Atomic coordinates
32 | %block Zmatrix
33 | cartesian
34 |   1  1.59042419  1.33862736  0.00972278  1  1  1
35 |   1 -1.04260379  1.7983006  0.00963033  1  1  1
36 |   1 -1.9545281  0.70811209  0.0092095  1  1  1
37 |   1 -1.03541135 -1.80228638  0.00944798  1  1  1
38 |   1  0.36369799 -2.0467446  0.00988278  1  1  1
39 |   1  2.07875101  0.0043931  0.00962579  1  1  1
40 |   1  1.95453152 -0.70814897  1.23149948  1  1  1
41 |   1  1.03544559  1.80227449  1.23185833  1  1  1
42 |   1 -0.36371707  2.04675882  1.23220986  1  1  1
43 |   1 -2.07875205 -0.00431065  1.23190018  1  1  1
44 |   1 -1.59040183 -1.33865926  1.23204063  1  1  1
45 |   1  1.04255686 -1.79832967  1.23176172  1  1  1
46 | %endblock Zmatrix
47 |
48 | XC.functional=LDA
49 | XC.authors=CA
50 |
51 | SolutionMethod Diagon
52 |

```



```

53 DM.MixSCF1      T
54 MaxSCFIterations      300
55 DM.MixingWeight      0.03
56 DM.Tolerance        1.d-4
57 DM.UseSaveDM        true
58 DM.NumberPulay      5
59 Diag.DivideAndConquer no
60
61 MD.TypeOfRun      CG
62 MD.NumCGsteps      0
63 MD.UseSaveXV      .true.
64
65 SaveHS                .false.
66 SaveElectrostaticPotential True
67 SaveTotalPotential    no
68 WriteCoorXmol        .true.
69 WriteMDXmol          .true.
70 WriteMDhistory       .false.
71 WriteEigenvalues     yes

```

Для расчета сопротивления нанотрубки необходимо знать равновесную конфигурацию модели (она нам известна по предыдущим примерам и заданиям). К ней нужно добавить информацию об электродах. Создать модель УНТ с электродами можно с помощью программы «Пре- и пост-процессор для моделирования свободных и деформированных углеродных нанотрубок с помощью пакетов Firefly и Siesta» (см. стр. 92). fdf файл УНТ (3,3) с электродами может быть следующим.

```

1 SystemName          nanotube (3,3)
2 SystemLabel         C36
3 NumberOfAtoms       36
4 NumberOfSpecies     1
5 WriteMullikenPop    1
6
7 %block ChemicalSpeciesLabel
8   1 6 C
9 %endblock ChemicalSpeciesLabel
10
11 PAO.BasisSize      DZ
12 PAO.SplitNorm      0.669
13 PAO.EnergyShift    0.05 eV
14 MeshCutoff         300.0 Ry
15
16 %block kgrid_Monkhorst_Pack
17   4 0 0 0
18   0 4 0 0
19   0 0 128 0
20 %endblock kgrid_Monkhorst_Pack

```

```

21
22 SpinPolarized .true.
23 MaxSCFIterations      150
24
25 SolutionMethod  Transiesta
26 NetCharge      0.00
27
28 DM.MixSCF1      T
29 MaxSCFIterations      300
30 DM.MixingWeight    0.03
31 DM.Tolerance      1.d-4
32 DM.UseSaveDM      true
33 DM.NumberPulay     5
34
35 Diag.DivideAndConquer no
36 NeglNonOverlapInt False
37 AtomicCoordinatesFormat NotScaledCartesianAng
38
39 XC.functional=LDA
40 XC.authors=CA
41 ZM.UnitsLength = Ang
42
43 LatticeConstant      73.3476 Ang
44
45 %block LatticeParameters
46   0.33 0.33 0.1 90. 90. 90.
47 %endblock LatticeParameters
48
49 %block Zmatrix
50 cartesian
51   1  -1.954528   0.708112  -2.433790   1   1   1
52   1  -1.035411  -1.802286  -2.433552   1   1   1
53   1   2.078751   0.004393  -2.433374   1   1   1
54   1  -1.042603   1.798300  -2.433370   1   1   1
55   1   1.590424   1.338627  -2.433277   1   1   1
56   1   0.363697  -2.046744  -2.433117   1   1   1
57   1   1.954531  -0.708148  -1.211500   1   1   1
58   1   1.042556  -1.798329  -1.211238   1   1   1
59   1   1.035445   1.802274  -1.211142   1   1   1
60   1  -2.078752  -0.004310  -1.211100   1   1   1
61   1  -1.590401  -1.338659  -1.210959   1   1   1
62   1  -0.363717   2.046758  -1.210790   1   1   1
63   1  -1.954528   0.708112   0.009209   1   1   1
64   1  -1.035411  -1.802286   0.009447   1   1   1
65   1   2.078751   0.004393   0.009625   1   1   1
66   1  -1.042603   1.798300   0.009630   1   1   1
67   1   1.590424   1.338627   0.009722   1   1   1
68   1   0.363697  -2.046744   0.009882   1   1   1
69   1   1.954531  -0.708148   1.231499   1   1   1

```

```

70 | 1 1.042556 -1.798329 1.231761 1 1 1
71 | 1 1.035445 1.802274 1.231858 1 1 1
72 | 1 -2.078752 -0.004310 1.231900 1 1 1
73 | 1 -1.590401 -1.338659 1.232040 1 1 1
74 | 1 -0.363717 2.046758 1.232209 1 1 1
75 | 1 -1.954528 0.708112 2.452209 1 1 1
76 | 1 -1.035411 -1.802286 2.452448 1 1 1
77 | 1 2.078751 0.004393 2.452626 1 1 1
78 | 1 -1.042603 1.798300 2.452630 1 1 1
79 | 1 1.590424 1.338627 2.452723 1 1 1
80 | 1 0.363697 -2.046744 2.452883 1 1 1
81 | 1 1.954531 -0.708148 3.674499 1 1 1
82 | 1 1.042556 -1.798329 3.674762 1 1 1
83 | 1 1.035445 1.802274 3.674858 1 1 1
84 | 1 -2.078752 -0.004310 3.674900 1 1 1
85 | 1 -1.590401 -1.338659 3.675040 1 1 1
86 | 1 -0.363717 2.046758 3.675210 1 1 1
87 | %endblock Zmatrix
88 |
89 | %block ProjectedDensityOfStates
90 | -24.00 15.00 0.100 1000 eV
91 | %endblock ProjectedDensityOfStates
92 |
93 | # MD variables
94 |
95 | MD.TypeOfRun CG
96 | MD.NumCGsteps 0
97 | MD.UseSaveXV .true.
98 |
99 | WriteCoorStep .true.
100 | WriteEigenvalues yes
101 |
102 | # Transiesta information
103 | # GF OPTIONS
104 | TS.ComplexContour.Emin -30.0 eV
105 | TS.ComplexContour.NPoles 03
106 | TS.ComplexContour.NCircle 30
107 | TS.ComplexContour.NLine 10
108 | # BIAS OPTIONS
109 | TS.biasContour.NumPoints 20
110 |
111 | # TS OPTIONS
112 | TS.Voltage 0.020 eV
113 |
114 | # TBT OPTIONS
115 | TS.TBT.Emin -0.5 eV
116 | TS.TBT.Emax +0.5 eV
117 | TS.TBT.NPoints 100
118 | TS.TBT.NEigen 3

```

```

119 TS.TBT.Eta          0.000001 Ry
120
121 # Write hamiltonian
122 TS.SaveHS          .true.
123
124 # LEFT ELECTRODE
125 TS.HSfileLeft     ./elecSWNT33.TSHS
126 TS.NumUsedAtomsLeft  12
127 TS.BufferAtomsLeft   0
128
129 # RIGHT ELECTRODE
130 TS.HSfileRight    ./elecSWNT33.TSHS
131 TS.NumUsedAtomsRight 12
132 TS.BufferAtomsRight  0

```

Скрипт запуска расчета:

```

transiesta < input.fdf > ./input.out
tbtrans < input.fdf > tbtrans.out

```

В конце файла `tbtrans.out` находится строка вида $Voltage, Current(A) = 2.00000000E - 022.66037359E - 06$. Запустив серию расчетов для различных значений переменной $TS.Voltage$. Можно построить вольт-амперную характеристику (ВАХ) исследуемой модели. На рис. 2.3 приведена ВАХ УНТ (3,3) в диапазоне напряжений от 0 до 0.2 В.

Оценка электрического сопротивления индивидуальной нанотрубки (3,3) дает значение 7.5 кОм, что на 16 % больше указанной выше величины (6,47 кОм). Это, однако, нельзя назвать ошибкой. Дело в том, что, как мы видели ранее (см. стр. 51), в зонной структуре УНТ (3,3) имеется запрещенная зона (небольшая). По этой причине ее сопротивление не может равняться 6,47 кОм, величине, полученной в предположении отсутствия запрещенной зоны.

2.6. Вычисление фононных спектров

Еще одной отличительной особенностью пакета SIESTA является возможность рассчитать с его помощью фононные спектры моделей.

Фононы — квазичастицы, описывающие коллективные колебания в ансамбле атомов. Рассмотрим кристалл из L элементарных ячеек с S базисными атомами в каждой ячейке (общее число атомов в таком кристалле равно $N = LS$), находящийся при низкой температуре. Для каждой ячейки l можно определить положение s -го иона в ней как $\vec{R}_{ls} = \vec{R}_l + \vec{R}_s$. Ионы в таком кристалле претерпевают малые колебания относительно положе-

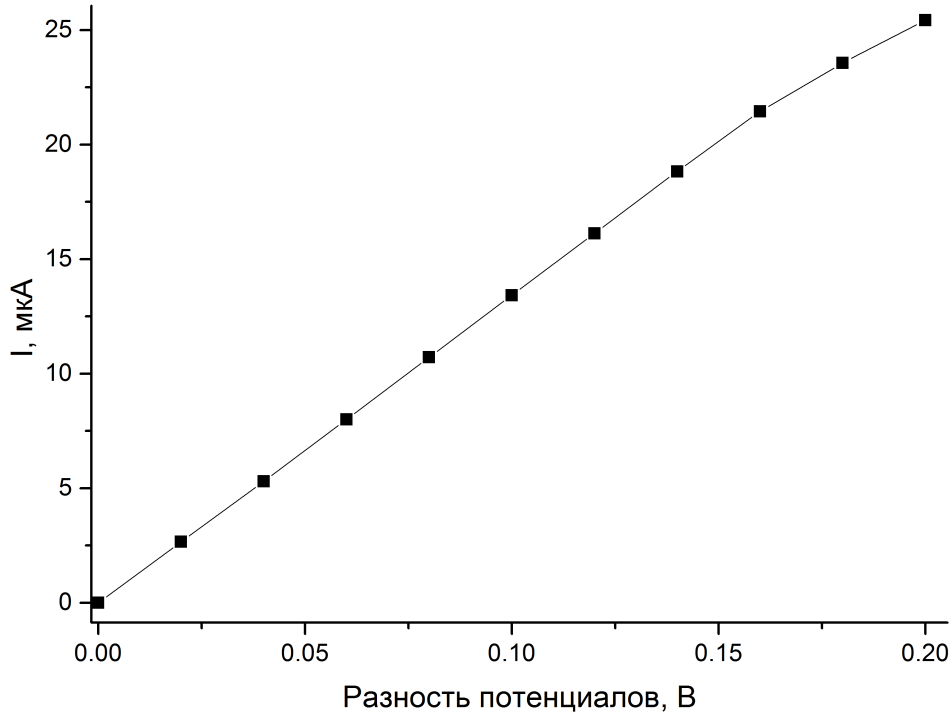


Рис. 2.3.: Вольт-амперная характеристика УНТ (3,3)

ния равновесия, причем смещения атомов будут зависеть от их положения в кристалле в целом: для каждого момента времени мы можем записать положение атома $\vec{R}_{ls}(t)$ как

$$\vec{R}_{ls}(t) = \vec{R}_{ls} + \vec{u}_{ls}(t), \quad (2.6)$$

где \vec{R}_{ls} — радиус-вектор положения равновесия s -го атома в l -й ячейке, и $\vec{u}_{ls}(t)$ — его малое смещение в зависимости от времени. Мы можем рассчитать потенциальную энергию такого кристалла, разложив ее в ряд Тейлора около равновесных позиций атомов:

$$U(\vec{R}_{ls}(t)) = U(\vec{R}_{ls}) + \sum_{ls\alpha} \frac{\partial U}{\partial u_{ls\alpha}} u_{ls\alpha}(t) + \frac{1}{2} \sum_{ls\alpha} \sum_{l't\beta} \frac{\partial^2 U}{\partial u_{ls\alpha} \partial u_{l't\beta}} u_{ls\alpha}(t) u_{l't\beta}(t) + \dots$$

В данном разложении α и β — декартовы компоненты вектора \vec{u} , и все производные взяты в равновесном положении кристалла. Первый член ряда — потенциальная энергия идеального кристалла, в котором все атомы находятся в равновесных позициях E_{eq} . Если атомы в кристалле близки к конфигурации с минимальной энергией, линейный член разложения (так как производные берутся в равновесном положении) будет равен нулю, и первый ненулевой член разложения будет квадратичным по смещениям атомов. Кроме этого, будем считать, что все члены разложения

высших порядков будут пренебрежимо малы (такое приближение называется *гармоническим*):

$$U_{harm} = U_{eq} + \frac{1}{2} \sum_{l\alpha, l't\beta} \Phi_{l\alpha, l't\beta} u_{l\alpha} u_{l't\beta}, \quad (2.7)$$

В последнем уравнении константы $\Phi_{l\alpha, l't\beta}$ называются *силовыми константами* в реальном пространстве и определяются как

$$\Phi_{l\alpha, l't\beta} = \frac{\partial^2 U}{\partial u_{l\alpha} \partial u_{l't\beta}}. \quad (2.8)$$

Эти константы должны удовлетворять ряду условий, следующих из изотропии пространства, симметрии кристалла и инвариантности кристалла относительно его переноса на вектор, кратный вектору трансляции кристалла — вторая производная энергии может зависеть только от разницы между l и l' :

$$\Phi_{l\alpha, l't\beta} = \Phi_{0s\alpha, (l'-l)t\beta}. \quad (2.9)$$

Покажем, что матрица силовых констант определяет зависимость между силой \vec{F}_{ls} , действующей на атом s в ячейке l , и смещением $\vec{u}_{l't}$ атома t в ячейке l' . Для этого запишем классические уравнения движения ионов:

$$\vec{F}_{ls} = M_s \vec{a}_{ls}. \quad (2.10)$$

Вспомним, что $\vec{F} = -\nabla U$, или при разложении по декартовым координатам $F_{ls\alpha} = -\partial U / \partial u_{l\alpha}$, и использовав для U разложение (2.7), получим (проверьте!)

$$F_{ls\alpha} = - \sum_{l't\beta} \frac{\partial^2 U}{\partial u_{l\alpha} \partial u_{l't\beta}} u_{l't\beta} = - \sum_{l't\beta} \Phi_{l\alpha, l't\beta} u_{l't\beta}, \quad (2.11)$$

что и требовалось доказать.

Найдем ускорение частиц. Для этого продифференцируем соотношение (2.6) два раза по времени:

$$\vec{a}_{ls} = \frac{d^2 \vec{R}_{ls}(t)}{dt^2} = \frac{d^2 \vec{u}_{ls}}{dt^2},$$

или при разложении на декартовы компоненты

$$a_{ls\alpha} = \frac{d^2 u_{ls\alpha}}{dt^2}. \quad (2.12)$$

Подставив соотношения (2.11) и (2.12) во второй закон Ньютона (2.10), получим систему из $3N$ связанных уравнений относительно малых смещений атомов:

$$M_s \frac{d^2 u_{l s \alpha}}{dt^2} = - \sum_{l' t \beta} \Phi_{l s \alpha, l' t \beta} u_{l' t \beta}. \quad (2.13)$$

Будем искать решения этой системы уравнений в виде коллективного колебания — *фонона*:

$$u_{l s \alpha}(t) = \eta_{m \vec{q}}(s \alpha) e^{i \vec{q} \cdot \vec{R}_l} e^{-i \omega_{m \vec{q}} t}, \quad (2.14)$$

где зависимость решения от времени является периодической функцией с периодом колебаний $\omega_{m \vec{q}}$, а в пространстве в каждый момент времени решение представляет собой плоскую волну с волновым вектором \vec{q} , определенную в точках, кратных \vec{R}_l . m в решении является его номером («модой» колебаний), а $\eta_{m \vec{q}}(s \alpha)$ — α -я компонента вектора, называемого *вектором поляризации* нормальной моды колебаний. Необходимо заметить, что такой подход очень похож на методику, применяемую для получения волновой функции электронов в рамках теоремы Блоха: колебания ионов отличаются волновым вектором \vec{q} , причем если кристалл бесконечно периодический, используются только разрешенные значения волнового вектора.

Подставим решения (2.14) в систему (2.13). Для этого возьмем вторую производную (2.14) по времени, сократим обе части на $e^{-i \omega_{m \vec{q}} t}$ и домножим обе части на $e^{-i \vec{q} \cdot \vec{R}_l}$:

$$M_s \omega_{m \vec{q}}^2 \eta_{m \vec{q}}(s \alpha) = \sum_{t \beta} \left[\sum_{l'} \Phi_{l s \alpha, l' t \beta} e^{i \vec{q} \cdot (\vec{R}_{l'} - \vec{R}_l)} \right] \eta_{m \vec{q}}(t \beta).$$

Вследствие того, что суммирование по l' производится по всем элементарным ячейкам кристалла, мы можем использовать свойство (2.9) матрицы силовых констант зависеть только от разницы между ячейками l' и l кристалла и перенести начало координат в ячейку с номером l :

$$M_s \omega_{m \vec{q}}^2 \eta_{m \vec{q}}(s \alpha) = \sum_{t \beta} \left[\sum_{l'} \Phi_{0 s \alpha, l' t \beta} e^{i \vec{q} \cdot \vec{R}_{l'}} \right] \eta_{m \vec{q}}(t \beta). \quad (2.15)$$

В скобках стоит дискретное преобразование Фурье матрицы силовых констант в реальном пространстве:

$$\sum_{l'} \Phi_{0 s \alpha, l' t \beta} e^{i \vec{q} \cdot \vec{R}_{l'}} = \tilde{\Phi}_{s \alpha, t \beta}(\vec{q}).$$

Таким образом, движение атомов может быть описано в виде системы следующих динамических уравнений:

$$M_s \omega_{m\vec{q}}^2 \eta_{m\vec{q}}(s\alpha) = \sum_{t\beta} \tilde{\Phi}_{s\alpha,t\beta}(\vec{q}) \eta_{m\vec{q}}(t\beta).$$

Можно записать эту же систему в матричной форме:

$$\tilde{\Phi}(\vec{q}) \eta(\vec{q}) = \omega^2(\vec{q}) M_s \delta_{st} \delta_{\alpha\beta} \eta(\vec{q}), \quad (2.16)$$

где $\tilde{\Phi}(\vec{q})$ — матрица размером $3S \times 3S$, в которой содержится результат преобразования Фурье силовых констант, $\eta(\vec{q})$ — вектор длиной $3S$ собственных смещений фононов с волновым вектором \vec{q} , $\omega(\vec{q})$ — частота фононов, и $M_s \delta_{st} \delta_{\alpha\beta}$ — диагональная матрица размером $3S \times 3S$, содержащая массы атомов для трех декартовых координат смещения каждого атома.

Мы можем заметить, что система (2.16) очень похожа на задачу поиска собственных значений матрицы $\tilde{\Phi}(\vec{q})$ за исключением наличия матрицы масс атомов. Для исключения этой матрицы из уравнения мы можем переопределить смещение (2.14) как

$$u_{l s \alpha}(t) = \frac{1}{\sqrt{M_s}} \gamma_{m\vec{q}}(s\alpha) e^{i\vec{q} \cdot \vec{R}_l} e^{-i\omega_{m\vec{q}} t}, \quad (2.17)$$

Тогда, повторяя предыдущие вычисления, мы можем определить *динамическую матрицу* системы как

$$\tilde{D}_{s\alpha,t\beta}(\vec{q}) = \frac{1}{\sqrt{M_s M_t}} \tilde{\Phi}_{s\alpha,t\beta}(\vec{q})$$

и переписать систему (2.16) в виде задачи на поиск собственных значений:

$$\tilde{D}(\vec{q}) \gamma(\vec{q}) = \omega^2(\vec{q}) \gamma(\vec{q}), \quad (2.18)$$

где $\gamma(\vec{q})$ — собственные вектора фононов. Можно показать, что динамическая матрица системы является эрмитовой.

Для решения уравнения (2.18) вначале необходимо найти динамическую матрицу (либо, что то же самое, матрицу силовых констант). Для этого существует два метода: метод линейного отклика, также называемый теорией возмущений функционала электронной плотности (Density Functional Perturbation Theory, DFPT), и техника малых отклонений. Основная идея метода линейного отклика, предложенного Барони и др. [32] — использование теоремы Хеллмана–Фейнмана для того, чтобы показать,

что линейное изменение электронной плотности при внесении возмущения в кристалл приводит к изменению энергии этого кристалла во втором (на самом деле, третьем [33]) порядке теории возмущений. При использовании стандартной теории возмущений изменение первого порядка зарядовой плотности может быть рассчитано с использованием невозмущенных волновых функций, и, соответственно, требует расчета в рамках ТФП для кристалла в основном состоянии. Если возмущение — фонон с волновым вектором \vec{q} , определение изменения электронной плотности в первом порядке теории возмущений может помочь в определении динамической матрицы системы для волнового вектора \vec{q} . Это может быть проделано для любого волнового вектора \vec{q} без необходимости строить большую суперячейку.

Методика малых отклонений атомов, реализованная в пакете SIESTA, определяет матрицу силовых констант на основе того, как изменяется сила, действующая на атом s , если сдвинуть атом t на малое расстояние в некотором направлении. Действительно, согласно определению силовых констант (2.8),

$$\Phi_{0s\alpha,l't\beta} = \frac{\partial}{\partial u_{0s\alpha}} \left(\frac{\partial U}{\partial u_{l't\beta}} \right) = -\frac{\partial F_{l't\beta}}{\partial u_{0s\alpha}},$$

то есть по силам, действующей на некоторый атом t в ячейке l' при сдвиге атома s в ячейке 0 в некотором направлении α , можно определить столбец матрицы $\Phi_{0s\alpha,l't\beta}$ для всех $(l't\beta)$. При повторении расчета сил для всех $(ls\alpha)$ полностью определяется матрица силовых констант Φ . Ясно, что количество проводимых расчетов при этом может быть втрое больше количества атомов в элементарной ячейке; однако для многих материалов симметрия может существенно уменьшить это количество. Кроме этого, следует иметь в виду одно важное обстоятельство. Из-за того, что расчеты в SIESTA проводятся с периодическими граничными условиями, суперячейка, в которой проводятся расчеты фононных спектров, должна быть достаточно большой, чтобы элементы матрицы Φ были пренебрежимо малы на границе ячейки для любых $s\alpha$ в ячейке 0. Для большинства материалов (особенно металлов) так оно и есть. Однако, в ионных материалах сходимость фононного спектра с увеличением размера суперячейки может быть очень медленной. Это происходит вследствие того, что сдвиг зарядов в таких материалах образует диполи, взаимодействующие с дальнедействующими силами, уменьшающимися только как r^{-3} . Эти диполи также создают макроскопическое электрическое поле, которое ответственно за расщепление мод оптической ветви на компоненты, параллельную и перпендикулярную внешнему электрическому полю (ЛО-ТО расщепление).

Для расчета фононных спектров в SIESTA встроен набор программ Vibra (расположен в папке `Utils/Vibra`). При компиляции Vibra (которая происходит запуском команды `make` в каталоге `Utils/Vibra/src`) происходит сборка двух программ — *fcbuild* и *vibrator*. Расчет состоит из трех этапов:

1. построение суперячейки для расчета колебаний;
2. получение матрицы силовых констант;
3. решение задачи на собственные значения и получение собственных векторов и частот фононов.

Построение суперячейки

Для построения суперячейки мы будем использовать программу *fcbuild*. Эта программа в качестве входных данных принимает специально подготовленный FDF-файл. Разберем его на примере расчета фононного спектра карбина с поликумуленовым строением. Карбин с поликумуленовым строением — одномерная цепочка из атомов углерода, в которой каждый атом связан с двумя соседними двойными связями; таким образом, в элементарную ячейку входит один атом углерода. FDF-файл выглядит следующим образом:

```
1 SystemName      Carbin
2 SystemLabel     Carbin
3
4 NumberOfSpecies 1
5 NumberOfAtoms   1
6 %block ChemicalSpeciesLabel
7 1 6 C
8 %endblock ChemicalSpeciesLabel
9
10 LatticeConstant 1.27620 Ang
11 %block LatticeParameters
12 5 5 1 90 90 90
13 %endblock LatticeParameters
14
15 AtomicCoordinatesFormat Fractional
16 %block AtomicCoordinatesAndAtomicSpecies
17 0 0 0 1 12.011
18 %endblock AtomicCoordinatesAndAtomicSpecies
19
20 SuperCell_1 0
21 SuperCell_2 0
22 SuperCell_3 2
23
24 %block BandLines
25 1 0.000 0.000 0.000 \Gamma
```

```
26 | 200 0.000 0.000 1.000 X
27 | %endblock BandLines
```

Разберем этот файл подробно. Первые две строки содержат полное и краткое название системы. Строки 4 и 5 определяют количество химических элементов и количество атомов в системе; эти параметры описаны в разделе 1.4 данного пособия. Строки 6–8 содержат описание химического строения системы — блок `ChemicalSpeciesLabel`, в котором каждому типу атомов ставится в соответствие его зарядовое число и название (служащее именем файла псевдопотенциала, см. раздел 1.8). Строки 10–13 содержат описание геометрии системы; оно должно содержать характерную длину системы (постоянную решетки, `LatticeConstant`) и вектора трансляции в блоке `LatticeParameters` или `LatticeVectors` (подробнее см. в разделе 1.4). Строки 15–18 определяют положение атомов в системе. Отметим, что для программы *fcbuild* формат *Z-matrix* задания положения атомов неприемлем. Положения атомов необходимо задавать в «традиционном» блоке `AtomicCoordinatesAndAtomicSpecies`, причем в отличие от SIESTA, для которой необходимо задавать только координаты и тип атомов, *Vibra* требует указание еще и массы атомов (для перехода между собственными векторами и собственными смещениями фононов).

Следующие строки специфичны для *Vibra*. Строки 20–22 определяют суперячейку для расчета фононного спектра методом малых отклонений в трех измерениях. Параметры `SuperCell_x` равны количеству слоев, окружающих центральную ячейку, в каждом из трех измерений. Для получения локализованных силовых констант необходимо выбирать достаточно большую суперячейку, однако при этом следует соблюдать осторожность, потому что выбор большого числа в `SuperCell_x` экспоненциально увеличивает размер системы для расчета на втором шаге. Так, например, выбор каждого из параметров `SuperCell_x` равным 2 увеличивает размер системы в 125 раз! Поэтому стоит увеличивать суперячейку только вдоль тех измерений, которые существенны для точности расчета.

Последний блок, строки 24–27, определяет линии между точками высокой симметрии в обратном пространстве, вдоль которых программа рассчитывает $\omega(\vec{q})$ — зонную структуру фононов (см. раздел 2.1).

После подготовки FDF-файла сохраним его как `fcbuild.fdf` и передадим его на вход ранее скомпилированной программе *fcbuild*:

```
# fcbuild < fcbuild.fdf
```

Программа выдаст на экран информацию о записанной структуре и создаст файл `FC.fdf`, использующийся для расчета матрицы силовых констант с помощью пакета SIESTA. В этом файле определены количество атомов в суперячейке, ее геометрия; координаты всех атомов в суперя-

чейке. Кроме этого, в нем указан тип расчета (`MD.TypeOfRun FC`), определены номера атомов из первоначальной элементарной ячейки, которые необходимо смещать на малое расстояние для расчета матрицы силовых констант (в `MD.FCfirst` указан первый, а в `MD.FClast` — последний атом промежутка), и указана величина этого смещения (`MD.FCdispl`, значение по умолчанию — 0.04 Bohr). Для изменения величины смещения добавьте в FDF-файл параметр `AtomicDispl`.

Построение матрицы силовых констант

Для получения матрицы силовых констант с помощью SIESTA необходимо подготовить еще один входной FDF-файл с использованием значений из `FC.fdf`. Давайте на него посмотрим:

```
1 SystemName Carbin
2 SystemLabel Carbin
3
4 NumberOfSpecies 1
5 %block ChemicalSpeciesLabel
6 1 6 C
7 %endblock ChemicalSpeciesLabel
8
9 PAO.BasisSize DZP
10 MeshCutoff 130.0 Ry
11
12 %include FC.fdf
```

Данный файл содержит ту информацию, которой нет в `FC.fdf` — о химическом строении системы и базисе разложения волновых функций электронов. Естественно, в него следует добавить все параметры, необходимые для точного расчета. При запуске SIESTA с этим файлом в качестве входного будут поочередно сдвинуты в отрицательном и положительном направлениях по трем координатным осям все атомы подготовленной суперячейки с номерами от `MD.FCfirst` до `MD.FClast` включительно; при этом создается матрица силовых констант, которая записывается в файл `Carbin.FC`. Этот файл для каждого сдвига каждого атома содержит три декартовы компоненты производной вектора силы (в $\text{эВ}/\text{Å}^2$), действующей на каждый атом в суперячейке; в конце расчета этот файл должен содержать $6SN$ строк.

Расчет силовых констант производится независимо для каждого атома, требующего сдвига. Это означает, что процедура построения матрицы силовых констант является полностью распараллеливаемой — компоненты матрицы, отвечающие разным атомам, могут быть рассчитаны на различных компьютерах, а затем результаты таких расчетов могут быть собраны

вместе. К сожалению, в пакете SIESTA такого функционала нет, однако это нетрудно проделать вручную — создать несколько различных расчетов, изменить в каждом из них параметры `MD.FCfirst` и `MD.FClast`, а после завершения всех расчетов собрать получившиеся FC-файлы в один, содержащий в себе информацию о всех рассчитанных силовых константах в верном порядке.

При аварийном завершении расчета он может быть продолжен; для этого необходимо вручную убрать из FC-файла информацию о силовых константах, уже рассчитанных для сдвигов атома, для которого расчет не был завершён (последние kN строк, где k — количество рассчитанных сдвигов последнего атома). После этого надо вручную изменить номер `MD.FCfirst` в `FC.fdf` и перезапустить расчет.

Получение частот

После получения матрицы силовых констант над ними совершается преобразование Фурье для получения динамической матрицы, и затем решается задача на собственные значения для нахождения частот и мод фононов. Это происходит в программе *vibrator*, которая принимает на вход тот же файл `fcbuild.fdf`, упоминавшийся ранее:

```
# vibrator < fcbuild.fdf
```

При запуске программы создаются два новых файла:

- `Carbin.bands` — частоты фононов (в см^{-1}); структура файла такая же, как и при расчете электронной зонной структуры;
- `Carbin.vectors` — моды фононов (компоненты векторов сдвигов атомов, отвечающих каждому коллективному колебанию).

Сетка k -точек (векторов \vec{q}) для преобразования Фурье задается в блоке `BandLines`. Перед проведением расчета $\omega(\vec{q})$ можно провести расчет частот фононов для Γ -точки; для этого в блоке `BandLines` файла, подготовленного для *fcbuild*, должна остаться одна строка

```
%block BandLines  
1 0.000 0.000 0.000 \Gamma  
%endblock BandLines
```

Расчет фононов для Γ -точки проводится без увеличения суперячейки; при этом можно обнаружить некоторые ошибки во входных данных. Так, согласно теории, в любой системе должно быть три *акустические* моды с $|\omega| < 0.1 \text{ см}^{-1}$, если это не так, следует проверить входные данные. Кроме этого, частоты остальных (*оптических*) мод должны быть положительными — наличие мод с отрицательными (на самом деле, мнимыми) частотами свидетельствует о недостаточной релаксации системы. Это можно исправить следующим образом. Возьмите моду с наибольшей по моду-

лю отрицательной частотой и передвиньте атомы системы вдоль соответствующих им компонент собственного вектора, отвечающего этой моде. Такое изменение системы должно уменьшить полную энергию системы. Взяв новое положение за первоначальное, вновь запустите структурную релаксацию системы и пересчитайте фононные частоты.

Обработка результатов

Результаты расчета фононного спектра можно представить тремя способами: графиком зонной структуры вдоль линий, соединяющих точки высокой симметрии системы (см. раздел 2.1); графиком плотности фононных состояний (строится утилитой *phdos*) и рисунком, показывающим направление и величину смещения атомов для различных мод колебаний. Рассмотрим последние два способа более подробно.

Утилита *phdos*, написанная А. Постниковым¹, строит график локальной плотности фононных состояний для Γ -точки обратного пространства. Этот график имеет смысл только в том случае, если число атомов в первоначальной ячейке достаточно велико. Плотность состояний рассчитывается утилитой как совокупность пиков дельта-функций, соответствующих частотам, рассчитанным программой *vibrator*, с весами, равными квадратам соответствующих компонент собственных векторов. Для правильной работы программы она должна быть откомпилирована компилятором *ifort*; кроме этого, суперячейка для расчета силовых констант должна быть равна элементарной ячейке. В программе также есть возможность получить свертку фононных состояний с волной с заданным q ; это может позволить получить спектральную плотность фононов системы (детально такие вычисления описаны в [34]). На выходе программа выдает два файла:

- *.VST — набор собственных частот и соответствующих им весов;
- *.WST — плотность состояний как непрерывная функция от частоты, полученная размытием пиков частот. Величина размытия пиков, равная 10 см^{-1} , задается и может быть изменена в коде программы.

Для полноты полученной картины можно визуализировать моды колебаний, возбуждаемых в ячейке, с помощью программы XCrySDen. Более полно эта программа будет рассмотрена в разделе 3.3.

¹Исходный код программы свободно доступен и находится по адресу <https://www.mail-archive.com/siesta-l@listserv.uam.es/msg02311.html>.

2.7. Упражнение 2

Задание 1. Убедитесь, что при использовании одной k - точки плотность электронных состояний (DOS) углеродных нанотрубок из упражнения 1 воспроизводится неудовлетворительно. Для этого в исправленные вами входные файлы (с оптимальными параметрами), полученные в результате выполнения заданий упражнения 1, добавьте строки

```
%block ProjectedDensityOfStates
-24.00  15.00  0.100  1000  eV
%endblock ProjectedDensityOfStates.
```

В качестве отчета приведите рисунок плотности электронных состояний в окрестности (± 2 эВ) уровня Ферми и сделайте выводы.

Задание 2. Определите оптимальное количество k - точек (учитывайте, что направления вдоль осей x и y равноправны, а размер ячейки в координатном пространстве вдоль оси z меньше, чем в двух других измерениях). Под оптимальным количеством понимается такое число k - точек, при увеличении которого не изменяется плотность состояний (визуально). Добавьте во входной файл строки

```
%block kgrid_Monkhorst_Pack
1  0  0  0.0
0  1  0  0.0
0  0  1  0.0
%endblock kgrid_Monkhorst_Pack.
```

Задание 3. Для оптимального количества k -точек рассчитайте зонную структуру. Во входной файл добавьте строки вида

```
BandLinesScale      ReciprocalLatticeVectors
%block Bandlines
1      0.000      0.000      0.000      \textbackslash Gamma
99     0.000      0.000      0.500      Z   #99 k точек в направлении Z
%endblock BandLines
WriteBands true
```

Задание 4. Докажите, что за проводимость углеродных нанотрубок отвечают $2p$ электроны. Для этого постройте парциальную плотность состояний.

Задание 5. Постройте спектр оптического поглощения углеродных нанотрубок (3,3) и (5,0).

3. Вспомогательное программное обеспечение

3.1. Пре- и постпроцессор для моделирования свободных и деформированных углеродных нанотрубок

Исследовательская деятельность специалиста по моделированию связана с выполнением множества однообразных операций, автоматизация которых может сохранить много времени. Именно для этой цели Созыкиным С.А. была написана программа «Пре- и постпроцессор для моделирования свободных и деформированных углеродных нанотрубок с помощью пакетов Firefly и Siesta».

Инсталляция программы

Программа распространяется в виде установочного файла для операционных систем семейства Windows на безвозмездной основе (по письменному запросу по электронной почте: sozykinsa@susu.ru). Для запуска программе требуется Framework версии 4.0. В случае успешного завершения процесса установки в списке доступных программ меню «Пуск» появится ярлык для запуска программы.

В директории с установленной программой кроме исполняемого файла вы обнаружите руководство пользователя и директорию «Examples», в которой содержатся следующие файлы:

1. `nanotube(3,3)_simple.xyz` — данные о структуре нанотрубки с индексами хиральности (3,3);
2. `nanotube(3,3)_Tinker.xyz` — данные о структуре нанотрубки с индексами хиральности (3,3) в формате, совместимом с пакетом химических приложений Chem Ultra;
3. `nanotube(3,3).inp` — исходные данные для запуска расчета с помощью пакета Firefly;
4. `nanotube(3,3).fdf` — исходные данные для запуска расчета с помощью пакета Siesta.

Для деинсталляции установленной программы с Вашего компьютера

запустите `uninstall.exe` из директории, в которой расположен исполняемый файл программы.

Особенности графического интерфейса

Доступ к возможностям программы осуществляется посредством меню и боковой панели (рис. 3.1). При одновременной работе с несколькими моделями нанотрубок для каждой из них в области «Открытые вкладки» располагается отдельная вкладка. Пользователю доступны различные режимы управления мышью: поворот модели нанотрубки в активном окне, перемещение активной модели нанотрубки, изменение масштаба прокручиванием колесика мыши, поворот моделей во всех открытых вкладках, перемещение моделей во всех открытых вкладках (перечислены в порядке расположения соответствующих переключателей в окне программы слева направо (рис. 3.1)). При нажатии на соответствующий переключатель вокруг него появляется красная квадратная рамка. При повторном нажатии этот индикатор исчезает. В случае если одновременно открыты несколько вкладок с моделями нанотрубок полезными могут оказаться оставшиеся 3 кнопки панели «Режимы управления мышью»: переход к предыдущей вкладке, отображение с интервалом, указанным в `Properties` `View` `Step`, всех открытых вкладок, переход к следующей вкладке.

Для увеличения рабочего пространства на экране можно скрыть боковую панель. Для этого нужно выбрать `View` `Do not show side panel`. Для отображения этой панели необходимо выполнить `View` `Show side panel`. Пользователю доступны пять режимов отображения моделей нанотрубок (раздел меню `View`):

1. Ball visual model – все атомы отображаются в виде шаров (цвет зависит от сорта атома);
2. Line visual model – положения центров атомов соединяются прямыми линиями;
3. Line + Ball – комбинация двух предшествующих пунктов;
4. Except carbon – отображение всех атомов кроме атомов углерода в виде шаров, все связи отображаются линиями одинаковой толщины.
5. Except carbon 1 – отображение всех атомов кроме атомов углерода в виде шаров, связи C–C отображаются линиями меньшей толщины.

В разделе меню `View` для удобства восприятия информации можно включать или отключать прорисовку номеров атомов, осей декартовой системы координат, которые будут следовать за поворотами и сдвигами модели. При внесении в структуру нанотрубок дефектов, форма их сечения, перпендикулярного оси нанотрубки, перестает быть окружностью. Для оценки степени деформации системы была реализована возможность

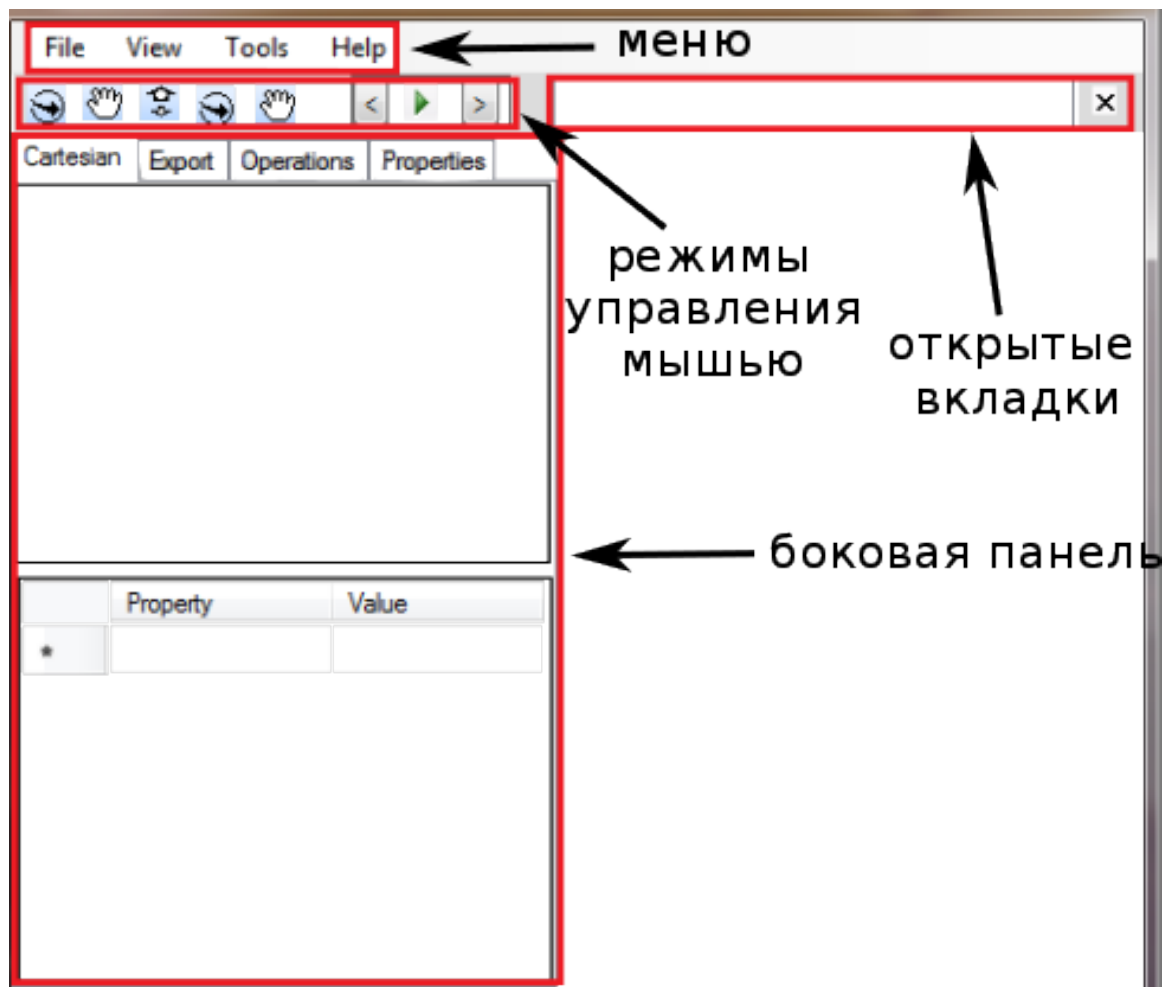


Рис. 3.1.: Доступ к основным возможностям программы

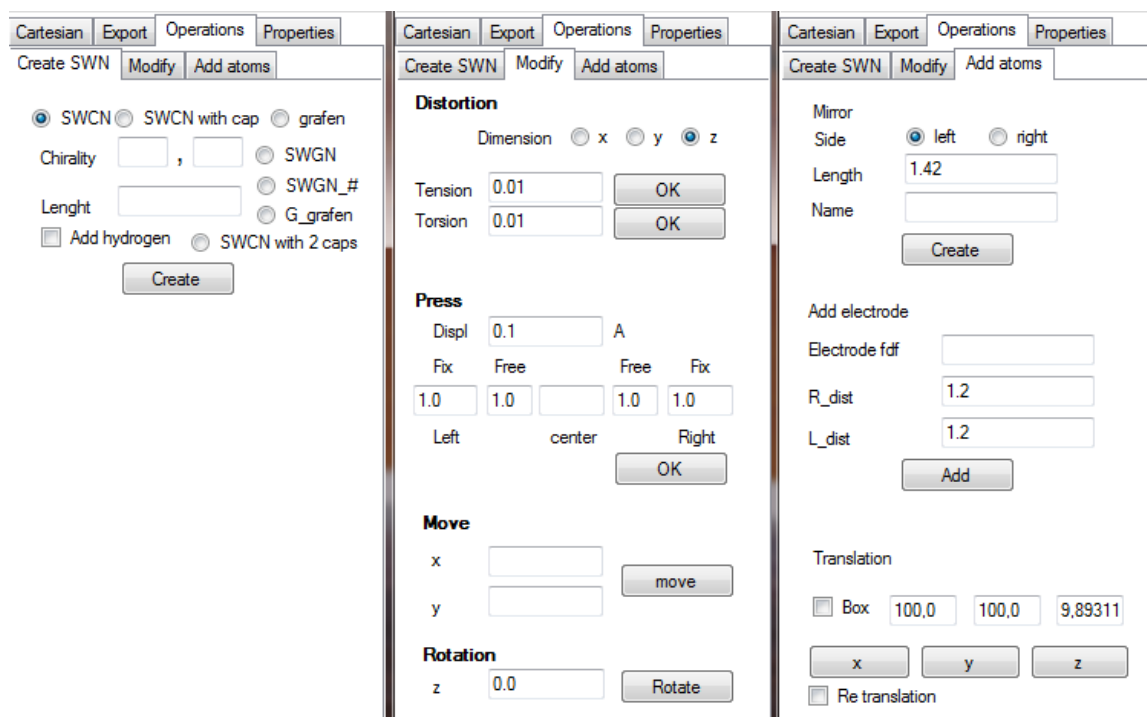


Рис. 3.2.: Создание моделей нанотрубок и графена и их преобразование отображения эллипса в рабочей области программы.

Создание стартовых конфигураций моделей

В программе реализован алгоритм создания моделей фрагментов углеродных и золотых нанотрубок с открытыми концами с произвольными индексами хиральности, закрытых с одного конца углеродных нанотрубок (6,6), (10,10) и (10,0), а также графена (рис. 3.2а). Для создания модели однослойной углеродной нанотрубки нужно выделить переключатель «SWNT». В поля, расположенные справа от слова «Chirality» вводятся индексы хиральности нанотрубки n и m (целые числа). Значение поля *length* (длина модели нанотрубки) ограничивает сверху длину нанотрубки, так как эта величина является дискретной и определяется количеством «колец» каркаса. Опция «Add hydrogen» позволяет добавлять водород к концевым атомам углерода. Для создания модели однослойной углеродной нанотрубки с закрытым концом нужно выделить переключатель «SWNT with cap». При этом появляется раскрывающийся список доступных для выбора нанотрубок. Для создания закрытых с двух концов нанотрубок нужно выделить пункт «SWCN with 2 cap». В литературе отсутствует общепринятый способ выбора базисных векторов нанотрубок из атомов золота. По этой причине в разных работах одной и той же нанотрубке могут приписываться разные индексы хиральности. Для их создания нужно

выделить пункт «SWG N» или «SWG N_#». Для создания модели графена нужно выделить переключатель «grafen». Индексы хиральности в данном случае задают ориентацию графена, т.е. при сворачивании графена получается нанотрубка с такими индексами хиральности.

Добавление электродов и «отражение» модели

Для расчета проводимости УНТ необходимо расположить электроды на нужном расстоянии от границ исследуемого образца. Сделать это можно, используя вкладку **Operations** » **Add atoms** » **Add electrode** (рис. 3.2в). При этом указывается путь к файлу в формате fdf, содержащий описание структуры электрода; и расстояние между атомами нанотрубки и электрода (независимо для обоих концов нанотрубки).

Для создания моделей контактов УНТ-УНТ можно воспользоваться вкладкой **Operations** » **Add atoms** » **Mirror** (рис. 3.2в), с помощью которой можно зеркально отразить модель УНТ (как слева, так и справа), находящуюся в активной вкладке, и поместить ее на заданном расстоянии от исходной модели.

Создание модели напряженной нанотрубки

Для определения механических характеристик нанотрубок при помощи какого-либо пакета программ необходимы модели напряженных структур. Для их получения можно воспользоваться вкладкой **Modify** (рис. 3.2б). Поддерживаются растяжение и кручение модели вдоль осей декартовой системы координат (необходимо выбрать соответствующий переключатель (x, y или z)). В случае растяжения (**Tension**) нужно указать величину растяжения в процентах от длины модели. При кручении (**Torsion**) вводятся значения угла закрутки в радианах на 1 нм длины модели.

Пользователь может создавать структуры для моделирования сжатия УНТ в направлении, перпендикулярном ее оси. При этом предоставляется возможность разделить фрагмент на 3 области (рис. 3.3): с замороженными степенями свободы атомов углерода (обеспечивает контакт с электродами), область «свободных» атомов углерода и деформируемую область. Вторая область вводится для связи фиксированных концевых областей с центральной областью, где фиксированными являются y-компоненты сдвинутых в процессе деформации атомов (z направление оси нанотрубки). Для создания модели нанотрубки, сжатой описанным образом, в окне программы необходимо задать следующие параметры: **Displ** – величина сжатия деформируемой области в ангстремах, **Fix** – расстояние от ле-



Рис. 3.3.: Схема деформирования фрагмента УНТ (5,5), использованная при моделировании поперечного сжатия

вого и правого концов модели (задаются независимо друг от друга), в пределах которого степени свободы атомов нанотрубки предполагаются замороженными, $Free$ — расстояние от границ левой и правой областей, указанных выше, в пределах которого на степени свободы атомов нанотрубки не накладывается никаких ограничений.

Создание входных файлов для пакета SIESTA и сохранение моделей

Программа позволяет создавать входные файлы для пакетов неэмпирического моделирования Siesta (*.fdf) и Firefly (*.inp), а также сохранять результаты работы в файл формата хуз (варианты структуры данного файла описаны ниже). При создании .inp и .fdf файлов, как правило, необходимо указывать большое количество параметров. Программа позволяет задавать только некоторые (наиболее употребительные) параметры. При выполнении конкретных расчетов, возможно, потребуется вручную добавить некоторые данные в файл.

Необходимые для создания входных файлов для пакета SIESTA инструменты размещены на вкладке **Export** **>** **.fdf(Siesta)** (рис. 3.4, 3.5), содержание которой разделено на 3 группы:

1. Common — основные параметры;
2. MD — выбор метода моделирования;
3. DFT — параметры обменно-корреляционного функционала.

Детальное описание параметров содержится в руководстве пользователя пакета SIESTA. В разделе Freeze предоставляется возможность выбора

двух вариантов заморозки степеней свободы атомов нанотрубки. Для выбора первого из них нужно поставить галочку напротив нужных (или всех) осей координат (x, y, z) и ввести желаемые расстояния в соответствующие поля (в ангстремах). После нажатия кнопки Create вычисляются границы параллелепипеда, в который помещается модель и замораживаются все три степени свободы атомов, находящихся на расстоянии меньше указанного от соответствующей границы (при условии наличия галочки напротив соответствующего измерения). Второй вариант заморозки координат реализуется при наличии галочки слева от «from press». В этом случае учитываются действия, выполненные на других вкладках (например, сжатие нанотрубки в направлении, перпендикулярном ее оси).

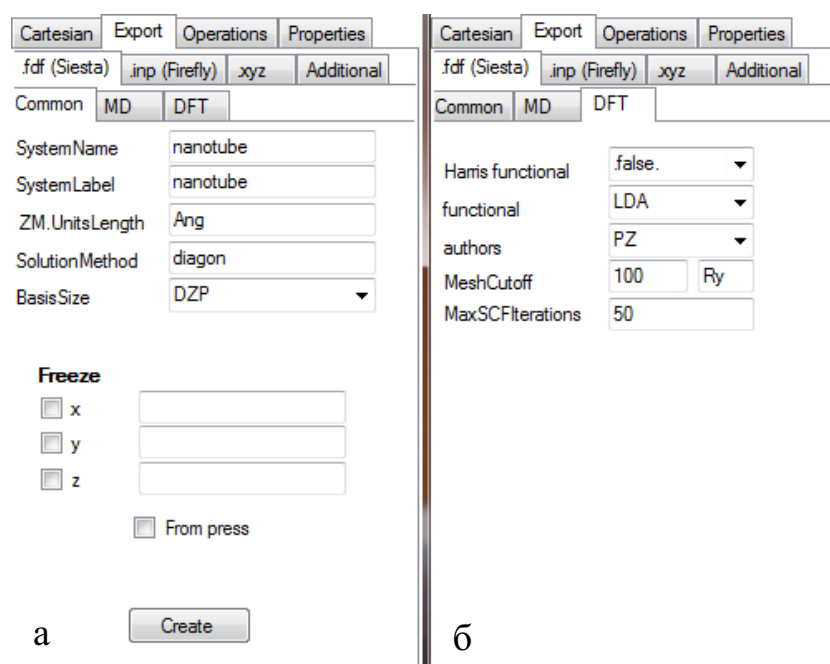


Рис. 3.4.: Вкладки для выбора параметров при создании входных файлов для пакета Siesta

Программа позволяет сохранять координаты атомов моделей нанотрубок в четырех различных форматах .xyz (см. рис. 3.6). Один из них (Simple) содержит набор координат и зарядов атомов. Второй, Tinker, хранит информацию еще и о наборе химических связей, имеющих в системе (именно этот формат используется программой ChemUltra – широко распространенной коммерческой графической оболочкой). Можно так же сохранить структуру в формате, совместимом с программами JMol и XCrySDen. Можно сохранять как исходную ориентацию моделей (независимо от того, как она выглядит в окне программы) или текущую ориентацию в пространстве (с учетом сдвигов и поворотов).

Программа может импортировать данные из файлов форматов .xyz

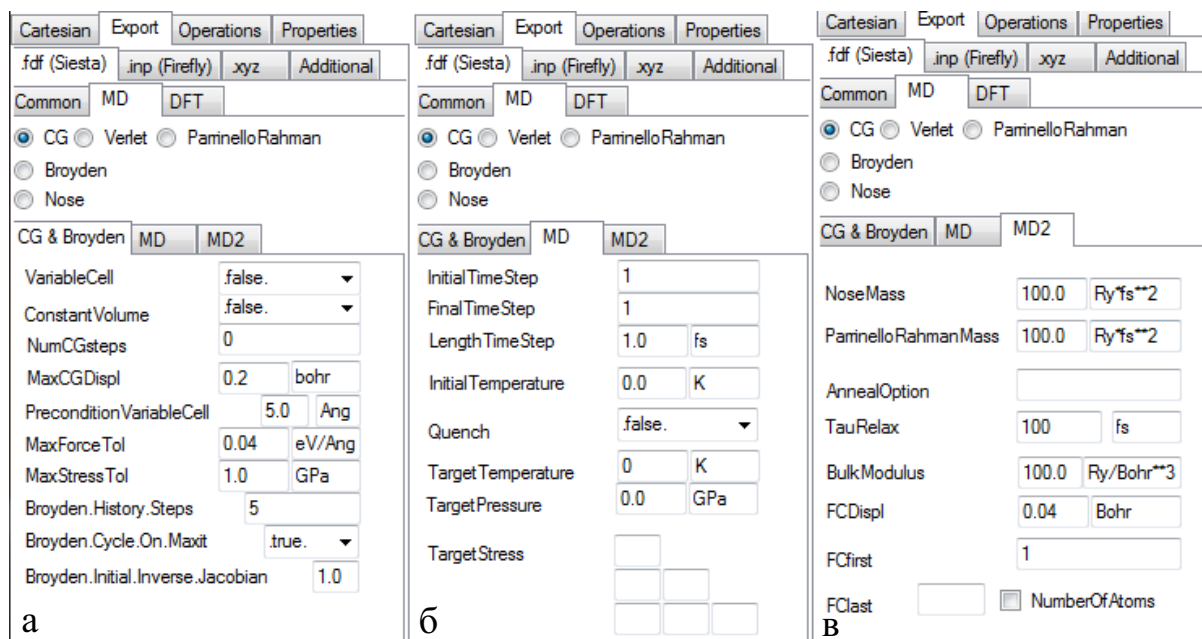


Рис. 3.5.: Вкладки для выбора параметров метода молекулярной динамики при создании входных файлов для пакета Siesta

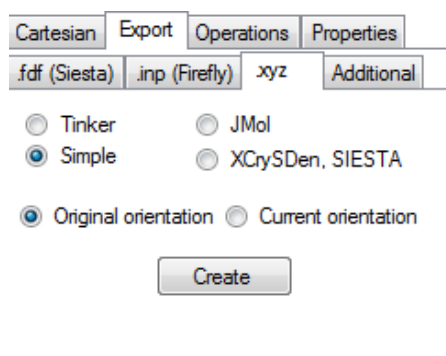


Рис. 3.6.: Экспорт модели в формат XYZ

(простой и совместимый с Chem Ultra), .inp (Firefly), .out (Firefly), .ANI (Siesta), .STRUCT OUT(Siesta), .FDF(Siesta), информацию о трехмерном распределении зарядовой плотности (в формате, совместимом с пакетом Gaussian) (Load CUBE), Movie - набор из координат нескольких моделей (рис. 3.7).

Инструменты

Вспомогательные возможности программы для подготовки входных файлов и работы с результатами расчетов, полученных с помощью пакета SIESTA, располагаются на вкладке Tools (рис. 3.8). При наведении курсора на **Tools** **SIESTA** открывается список, при щелчке мыши на эле-

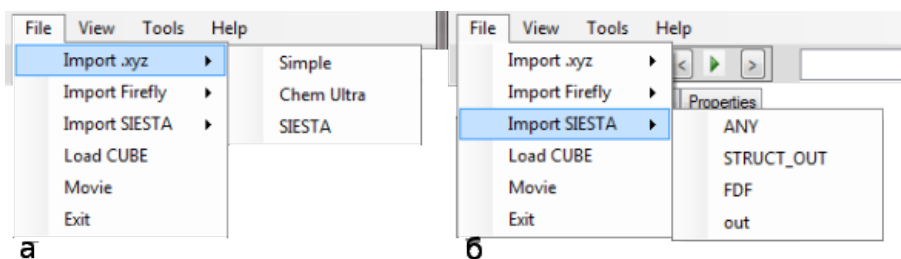


Рис. 3.7.: Импорт данных

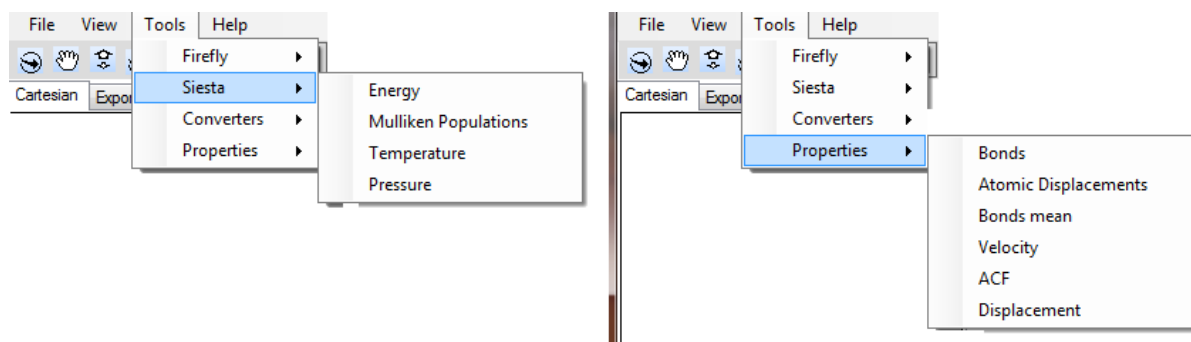


Рис. 3.8.: Вспомогательные возможности программы для подготовки входных файлов и обработки результатов расчетов

ментах которого, можно получить информацию о

1. Energy – информация об изменении полной энергии системы при оптимизации геометрии;
2. Mulliken populations – заселенности атомов, оцененные по методике Маликена;
3. Temperature – изменение температуры системы в процессе расчета молекулярной динамики;
4. Pressure – изменение давление в системе в процессе расчета молекулярной динамики.

На вкладке **Tools** **Properties** (см. рис. 3.8 б) можно получить информацию о

1. Bonds – длины связей в активной модели;
2. Bonds mean – среднее значение длин связей в серии моделей (оптимизация геометрии или молекулярная динамика);
3. ACF – автокорреляционная функция.

На рисунке 3.9 приведены вкладки для доступа к дополнительным возможностям настройки программы и расчета свойств. В разделе «Common» (рис. 3.9 а) можно указать необходимо ли отображать информацию о системе в таблице на вкладке «Cartesian» (для отображения нужно отметить «Show grid information») и трехмерную модель системы

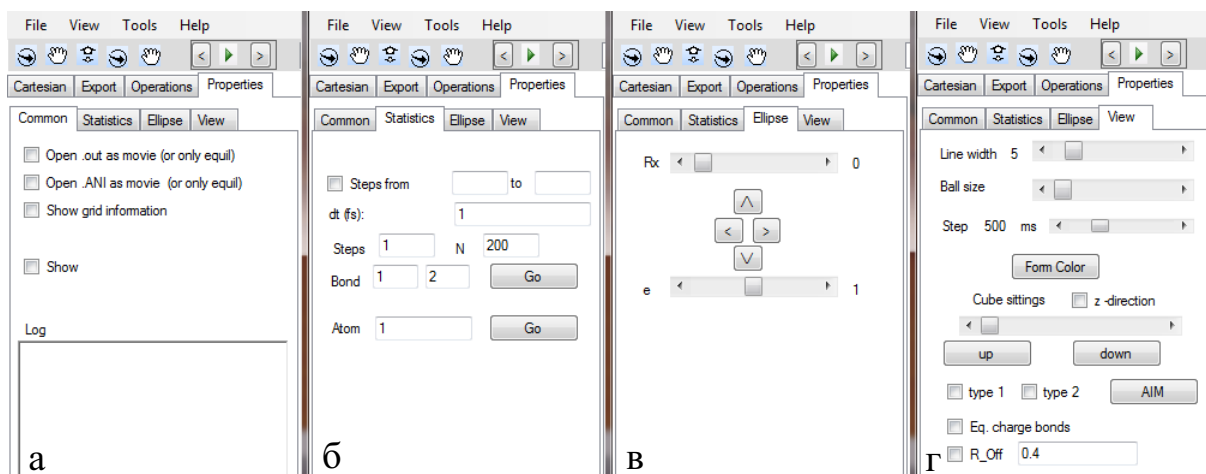


Рис. 3.9.: Вкладка properties

в правой области окна программы (опция «Show»). На этой же вкладке можно просмотреть лог программы.

В разделе «Statistics» (рис. 3.9 б)) можно создать файл с информацией о положении выбранного атома или длины связи между определенными атомами в серии моделей (оптимизация геометрии, расчет молекулярной динамики). Так же на этой вкладке можно задать параметры расчета автокорреляционной функции.

В разделе «Ellipse» (рис. 3.9 в)) настраиваются параметры эллипса, отображаемого в сечении нанотрубки для оценки степени ее деформации.

В разделе «View» (рис. 3.9 г)) регулируются толщина линий, изображающих связи между атомами и размеры атомов.

3.2. Denchar

Программа Denchar позволяет подготовить для визуализации плотность заряда и волновые функции электронов в координатном пространстве. Код программы Denchar расположен в директории Tools/Denchar. Для его компиляции достаточно выполнить команду `make denchar`.

Программа может работать в двух режимах, определяемых строковым параметром Denchar.TypeOfRun: 2D или 3D (значение по умолчанию: 2D). В 2D режиме плотность заряда и электронные волновые функции выводятся на двумерной секте, указанной пользователем. Результат выводится в файл в виде списка $x_i, y_i, f(x_i, y_i)$, где f — плотность заряда или волновая функция, x_i и y_i — координаты в декартовой системе координат. Этот файл может быть использован для построения контурной карты при

помощи графических программ. В 3D режиме плотность заряда и электронные волновые функции выводятся в файл формата Gaussian Cube, который может быть визуализирован, например, программами Moldel или Molekel. Для построения сетки используется ортогональная система координат, которая может не совпадать с используемой при расчете пакетом SIESTA.

При запуске программы Denchar считывается информация из файлов (SystemLabel.PLD и SystemLabel.DIM), сгенерированные пакетом SIESTA при переменной WriteDenchar равной true. Кроме того, необходимы файлы, которые содержат информацию о базисном наборе для каждого типа атомов: ChemicalSpecies.ion. При построении плотности заряда необходимо также файл, содержащий матрицу плотности SystemLabel.DM. При построении волновых функций необходим файл SystemLabel.WFSX.

Программа Denchar считывает параметры расчета из стандартного входного потока. Во входном файле кроме переменных SystemLabel, NumberOfSpecies и ChemicalSpeciesLabel, имеющих те же значения, что и при запуске пакета SIESTA, нужно задать ряд параметров, большинство из которых определяют сетку в координатном пространстве.

Переменные Denchar.PlotCharge и Denchar.PlotWaveFunctions определяют физическую величину, которая будет рассчитана программой Denchar. Логическая переменная Denchar.PlotCharge указывает нужно ли рассчитывать плотность зарядовой плотности (значение по умолчанию: false). Если эта переменная равна true, в каталоге с расчетом должен присутствовать файл SystemName.DM. Логическая переменная Denchar.PlotWaveFunctions указывает нужно ли рассчитывать волновые функции в координатном пространстве (значение по умолчанию: false). Хотя бы один из параметров Denchar.PlotCharge или Denchar.PlotWaveFunctions должен быть равен true.

Пользователю предоставляется возможность изменять единицы измерения координат и плотности заряда. Строковая переменная Denchar.CoorUnits задает единицы измерения координат точек на плоскости (значение по умолчанию: Bohr). Допустимые значения: Ang (ангстрем) или Bohr (боровский радиус). Строковая переменная Denchar.DensityUnits определяет единицы измерения зарядовой плотности в выходном файле (значение по умолчанию: Ele/bohr**3). Допустимые значения: Ele/Bohr**3 (элементарный заряд / радиус бора в кубе), Ele/Ang**3 (элементарный заряд / ангстрем в кубе), Ele/UnitCell (элементарный заряд / параметр решетки).

Число точек секты вдоль осей x, y и z (в режиме 3D) задают переменные Denchar.NumberPointsX, Denchar.NumberPointsY и Denchar.NumberPointsZ, соответственно. Значения по умолчанию: 50.

Действительные длины `Denchar.MinX`, `Denchar.MaxX`, `Denchar.MinY`, `Denchar.MaxY`, `Denchar.MinZ` и `Denchar.MaxZ` задают максимальные (значения по умолчанию: 3.0 bohrs) и минимальные (значения по умолчанию: -3.0 bohrs) компоненты координат сетки, в узлах которой рассчитываются зарядовая плотность или волновые функции.

Строковая переменная `Denchar.PlaneGeneration` определяет выбор способа генерации двумерной плоскости (значение по умолчанию: `NormalVector`). Допустимые значения: `NormalVector` (плоскость определяется нормалью к ней), `TwoLines` (плоскость задается двумя векторами, принадлежащими плоскости), `ThreePoints` (плоскость задается тремя точками), `ThreeAtomicIndices` (плоскость задается положениями трех указанных атомов).

Блок данных `Denchar.CompNormalVector` задает вектор нормали к плоскости (значение по умолчанию: 0.000 0.000 1.000). Нормаль задает семейство плоскостей. Для выбора одной из них необходимо указать две точки: лежащую в плоскости и определяющую направление оси x. Первая из них задается блоком `Denchar.PlaneOrigin` (значение по умолчанию: 0.000 0.000 0.000). Вторую точку нужно указать в блоке данных `Denchar.X-Axis` (значение по умолчанию: 1.000 0.000 0.000, единица измерения: bohrs).

Если параметру `Denchar.PlaneGeneration` присвоено значение `TwoLines`, блок данных `Denchar.Comp2Vectors` позволяет задать компоненты двух векторов на плоскости. Первый из них определяет направление оси x. Необходимо так же задать начало координат (см. `Denchar.PlaneOrigin`). Значение по умолчанию:

```
1.000 0.000 0.000
0.000 1.000 0.000.
```

Если параметр `Denchar.PlaneGeneration` равен `ThreePoints`, блок данных `Denchar.Coor3Points` позволяет задать координаты трех точек, принадлежащих плоскости. Первая из них принимается за начало координат. Вектор, соединяющий первую и вторую точку задает направление оси x. Значение по умолчанию:

```
1.000 0.000 0.000
0.000 1.000 0.000
0.000 0.000 1.000
```

Если переменная `Denchar.PlaneGeneration` равна `ThreeAtomicIndices`, блок данных `Denchar.Indices3Atoms` позволяет указать номера трех атомов модели, принадлежащих плоскости. Предполагается, что начало координат совпадает с положением первого атома. Направление оси x задает вектор, соединяющий первый и второй атомы. Значение по умолчанию:

1 2 3

Выходные файлы

Программа Denchar генерирует разные выходные файлы в зависимости от режима запуска (2D или 3D) и указанной для построения величины (плотность заряда и / или волновые функции).

Выходные файлы, создаваемые в 2D режиме имеют одинаковый формат: три столбца, первые два из которых задают координаты точек на плоскости, а третья задает величину плотности заряда или волной функции в данной точке. Эти файлы могут быть использованы для построения контурных карт или других 2D представлений с помощью графических программ.

В случае выбора зарядовой плотности в качестве физической величины, которую необходимо отобразить на плоскости, генерируются файлы SystemLabel.CON.SCF и SystemLabel.CON.DEL (для спин-поляризованного расчета: SystemLabel.CON.UP, SystemLabel.CON.DOWN, SystemLabel.CON.MAG и SystemLabel.CON.DEL). Файл SystemLabel.CON.SCF (SystemLabel.CON.UP и SystemLabel.CON.DOWN: для двух ориентаций спина в случае спин-поляризованного расчета) содержит сведения о величине зарядовой плотности в заданной плоскости координатного пространства. Файл SystemLabel.CON.DEL хранит данные о разности между зарядовой плотностью системы и суммой зарядовых плотностей изолированных атомов. Файл SystemLabel.CON.MAG содержит намагниченность: разность между плотностями заряда со спином вверх и вниз.

Для построения волновых функций Denchar создает отдельный файл (SystemLabel.CON.WF# : номер волновой функции #) для каждой волновой функции, сведения о которой хранятся в сгенерированном SIESTA файле SystemLabel.WFSX. В случае спин-поляризованного расчета для каждой волновой функции создаются два файла: SystemLabel.CON.WF#.UP и SystemLabel.CON.WF#.DOWN.

В 3D режиме выходные файлы имеют Gaussian Cube формат. Эти файлы могут быть использованы для визуализации трехмерных поверхностей при помощи таких программ как Molden или Molekel. Если программа Denchar запускалась для последующей визуализации зарядовой плотности, генерируются файлы SystemLabel.RHO.cube (SystemLabel.RHO.UP.cube и SystemLabel.RHO.DOWN.cube: для спин-поляризованного расчета) и SystemLabel.DRHO.cube. Файл SystemLabel.RHO.cube содержит информацию о плотности заряда в узлах трехмерной сетки. Файл SystemLabel.DRHO.cube хранит сведения о разности плотностей заряда модели и суммой плотностей заряда

изолированных атомов.

При подготовке к визуализации волновых функций для каждой волновой функции, описанной в файле SystemLabel.WFSX, создается файл SystemLabel.WF#.cube (SystemLabel.WF#.UP.cube и SystemLabel.WF#.DOWN.cube для спин-поляризованного расчета), содержащий данные о волновой функции в узлах трехмерной сетки.

Пример. Карта плотности заряда УНТ (3,3)

Рассмотрим модель УНТ (3,3) из 12 атомов. Для демонстрации возможностей Denchar рассмотрим следующий файл input.fdf. Обратите внимание на его последнюю строку. Именно она в дальнейшем позволит построить карту электронной плотности. Выполним расчет.

```
1 SystemName          nanotube (3,3)
2 SystemLabel         SWNT
3 NumberOfAtoms      12
4 NumberOfSpecies     1
5
6 %block ChemicalSpeciesLabel
7   1  6  C
8 %endblock ChemicalSpeciesLabel
9
10 PAO.BasisSize       DZP
11 PAO.EnergyShift    0.05 eV
12 MeshCutoff         300.0 Ry
13
14 %block kgrid_Monkhorst_Pack
15   1  0  0  0.5
16   0  1  0  0.5
17   0  0 32  0.5
18 %endblock kgrid_Monkhorst_Pack
19
20 AtomicCoordinatesFormat  NotScaledCartesianAng
21
22 XC.functional=LDA
23 XC.authors=CA
24 ZM.UnitsLength = Ang
25
26 %block Zmatrix
27 cartesian
28   1  -1.047296  -1.824411  0.009275  1  1  1
29   1  -1.977588   0.717694  0.009293  1  1  1
30   1   1.610063   1.354002  0.009392  1  1  1
31   1  -1.056152   1.819447  0.009474  1  1  1
32   1   2.103792   0.005350  0.009479  1  1  1
33   1   0.367179  -2.071354  0.009506  1  1  1
34   1   1.977588  -0.717714  1.231710  1  1  1
```

```

35  1  1.056160  -1.819432  1.231760  1  1  1
36  1  -0.367197  2.071367  1.231857  1  1  1
37  1  1.047315  1.824412  1.231857  1  1  1
38  1  -1.610057  -1.354016  1.231890  1  1  1
39  1  -2.103789  -0.005298  1.231893  1  1  1
40  %endblock Zmatrix
41
42  LatticeConstant  2.44492076099  Ang
43
44  %block LatticeParameters
45  10.0 10.0 1.0 90. 90. 90.
46  %endblock LatticeParameters
47
48  MD.TypeOfRun      cg
49  MD.NumCGsteps     360
50  MD.MaxCGDispl     0.15 Ang
51  MD.MaxForceTol    0.04 eV/Ang
52
53  WriteDenchar      True

```

Подготовим следующий входной файл (denchar.fdf) для программы Denchar.

```

1  SystemLabel      SWNT
2  NumberOfSpecies  1
3
4  %block ChemicalSpeciesLabel
5  1  6  C
6  %endblock ChemicalSpeciesLabel
7
8  Denchar.TypeOfRun 2D
9  Denchar.PlotCharge true
10 Denchar.CoorUnits  Ang
11 Denchar.DensityUnits Ele/Ang**3
12 Denchar.PlaneGeneration ThreeAtomicIndices
13 Denchar.Indices3Atoms 1 2 6
14
15 Denchar.MinX      -5.0 Ang
16 Denchar.MaxX      8.0 Ang
17 Denchar.MinY      -5.0 Ang
18 Denchar.MaxY      8.0 Ang
19
20 Denchar.NumberPointsX 300
21 Denchar.NumberPointsY 300

```

Запустим расчет командой

```
# denchar < input.fdf.
```

Результаты вычислений сохраняются в ряде выходных файлов и стандартном выходном потоке.

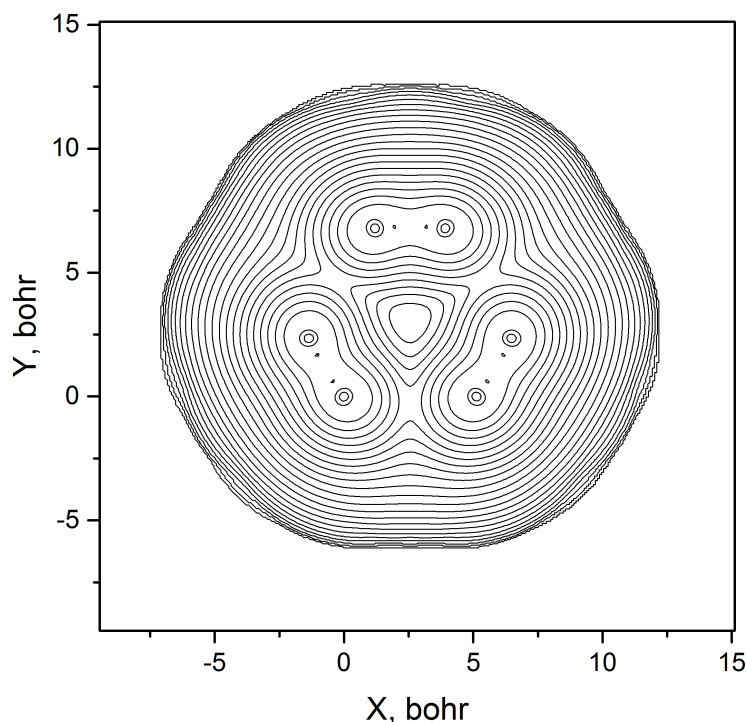


Рис. 3.10.: карта электронной плотности УНТ (3,3)

Используя привычную вам программу для построения графиков, вы сможете визуализировать файл SWNT.CON. Результат должен оказаться похож на приведенный на рис. 3.10.

3.3. XCrySDen

XCrySDen — программа для визуализации структуры кристаллов и молекул, изоповерхностей и контуров. Эта программа может быть запущена на большинстве операционных систем, включая Linux, MacOSX (требуется X11) и Windows (требуется CYGWIN). Название является сокращением от X-Windows Crystalline Structures and Densities. XCrySDen распространяется под лицензией GNU General Public License, эту программу можно скачать уже в собранном виде с веб-сайта <http://www.xcrysden.org>. XCrySDen имеет интуитивно понятный пользовательский интерфейс и позволяет манипулировать кристаллическими структурами и их 2-х и 3-х мерными свойствами, которые отображаются двумерными графиками или трехмерными изоповерхностями.

Возможности программы для визуализации результатов работы пакета SIESTA включают в себя

- визуализацию структуры модели;
- визуализацию шагов молекулярной динамики или оптимизации струк-

- туры;
- построение зарядовой и спиновой плотности;
 - визуализацию колебательных мод;
 - построение поверхности Ферми модели.

Для преобразования выходных файлов пакета SIESTA во входные файлы программы XCrySDen в формате XSF в состав дистрибутива пакета входит написанный А.Постниковым пакет утилит Sies2xsf (размещен в папке /Util/Contrib/APostnikov). Сборка исполняемых файлов утилит производится при запуске команды *make* (сперва требуется проверить и при необходимости изменить переменную `OBJDIR` в Make-файле; она должна указывать на директорию, в которой находится результат компиляции пакета). Каждая из утилит работает в интерактивном режиме, и после ответа на несколько вопросов записывает результат своей работы в XSF-файл. Рассмотрим работу с утилитами и программой XCrySDen более подробно.

Атомная структура

Для построения атомной структуры модели необходим выходной файл XV пакета SIESTA. Заметим, что этот файл не создается по умолчанию: для его генерации нужно изменить значение параметра `WriteCoorStep` на `true`. Для перевода его в формат, понимаемый XCrySDen, воспользуемся утилитой *xv2xsf* из набора Sies2xsf. Для этого скопируем утилиту в рабочую директорию и запустим ее. Утилита задаст вопрос о кратком имени («метке», `SystemLabel`) расчета и создаст в рабочей директории файл с расширением XSF. Для открытия этого файла программой XCrySDen выберем в меню программы пункт **File** **Open Structure** **Open XSF** и выберем в открывшемся диалоге XSF файл. После открытия файла нам доступна трехмерная модель системы и набор инструментов для работы с ней. Инструменты включают получение информации о выделенном атоме (`Atoms info`), определение расстояния между двумя атомами (`Distance`) и углов, составленных тремя атомами (`Angle`), а также двугранного угла, определяемого четырьмя атомами (`Dihedral angle`). Кроме этого, в программе есть большой набор параметров, изменяющих представление модели; мы советуем разобраться с ними самостоятельно.

Сохранение полученного рисунка в файл возможно при выборе в меню команды **File** **Print**. При этом появляется диалоговое окно `Print to file` с выбором из большого количества различных графических форматов. Надо понимать, что в файл сохранится то, что в данный момент находится на экране, поэтому для улучшения качества рисунка перед сохранением стоит развернуть окно программы на весь экран, приблизить модель

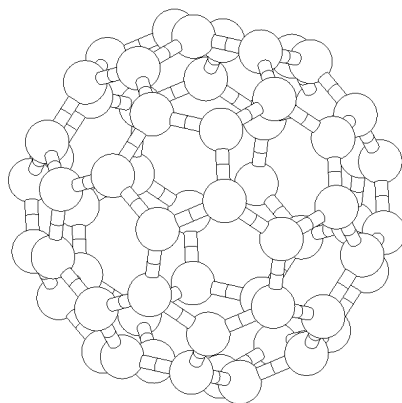


Рис. 3.11.: Фуллерен

до такой степени, чтобы она занимала большую часть окна программы, и увеличить параметр разбиения модели (**Modify** **Tessellation factor**) до 40–50.

Анимация молекулярной динамики

Для отображения анимации шагов молекулярной динамики в XCrySDen существует формат **.AXSF**. Открыть такой файл можно через пункт меню: **File** **Open structure** **Open AXSF** или набрав в командной строке:

```
xcrysdem --axsf < имя файла >
```

Для перевода файлов истории молекулярной динамики, созданных SIESTA, в формат AXSF используем утилиту *md2axsf*. При запуске ее в директории с расчетом она запрашивает (в интерактивном режиме) несколько параметров: краткое имя расчета; файл, из которого необходимо считать данные (доступен выбор из неформатированных файлов **.MD** и **.MD_CAR** или отформатированного файла **.ANI**); шаги анимации; вектора трансляции ячейки.

Запись о каждом шаге молекулярной динамики содержит номер шага. Программа *md2axsf* просит пользователя указать первый и последний шаги и интервал. После чтения всей информации из файла программа запрашивает:

```
Cleanly read in          <NNN> MD steps
You may wish to keep only some of these steps,
MDfirst, MDfirst+MDstep, MDfirst+2*MDstep etc. till (not
exceeding) MDlast from the list above. Specify three numbers
MDfirst, MDlast, MDstep - or 0 for any of them as default :
```

Например, если пользователь укажет: 10 0 3, это будет означать, что необходимо записать в файл шаги с 10 до последнего с шагом 3. В файле AXSF они будут иметь номера 1, 2, 3, 4 и т.д.

После загрузки AXSF-файла в XCrySDen открывается диалог Animation Control Center, в котором можно управлять анимацией, а также изменить скорость проигрывания и ее шаг. Кроме этого, в диалоге доступна кнопка Animated GIF/MPEG/AVI, при нажатии на которую разворачивается диалог сохранения анимации в файл. Следует учесть, что сохранение в файл осуществляется внешними программами — анимированный GIF-файл создается программой *convert* из пакета ImageMagick, а видеофайлы создаются программой *mencoder* в составе mplayer. Если в системе недоступны эти программы, опция сохранения анимации в файл также будет недоступна.

Зарядовая плотность

Основные возможности программы XCrySDen по просмотру и обработке трехмерных данных, сгенерированных пакетом SIESTA.

1. Просмотр свойств, рассчитанных в каждой точке расчетной сетки:
 - RHO (параметр SaveRho);
 - IORHO (параметр SaveDeltaRho);
 - VH (параметр SaveElectrostaticPotential);
 - VT (параметр SaveTotalPotential);
 - IOCH (параметр SaveIonicCharge);
 - TOCH (параметр SaveTotalCharge);
 - LDOS (блок LocalDensityOfStates).

Программа rho2xsf позволяет (и требует) выбрать отображаемую область в качестве произвольного параллелепипеда (может не совпадать с элементарной ячейкой), заданного началом отсчета и тремя векторами. Позиции атомов и свойства сетки будут восстановлены во всей отображаемой области, с учетом периодичности решетки. Скрипт rho2xsf работает в интерактивном режиме и должен быть запущен без передачи каких-либо параметров. Он использует файл .XV (для построения атомов в выбранной области) и, при необходимости, файлы с информацией о свойствах (.RHO, .LDOS и т.д.). Выходной файл имеет расширение XSF. Его можно открыть с помощью программы XCrySDen.

Использовать подготовленные .XSF файлы можно следующим образом:

выберите в меню XCrySDen: File -> Open Structure -> Open XSF или в командной строке наберите: xcrysdn -xsf < XSF file >

При этом будет прочитана кристаллографическая информация и показаны атомы в элементарной ячейке. Если в файле XSF есть скалярное поле (трехмерные свойства), в подменю "Tools" главного ме-

ню XCrySDen становится активным элемент «Data Grid», который открывает карту доступных данных сетки. Выберите блок (если их много - LDOS, RHO и т.д.) для построения. Внутри блока можно активировать-деактивировать субблоки и изменить их содержимое посредством умножения на произвольное число «Multiply factor». rho2xsf создает два блока для спин поляризованного расчета. Обратите внимание, что XCrySDen активирует по умолчанию только первый субблок. При необходимости можно активировать оба (для суммирования по спином) или, например, для второго блока задать «Multiply factor» равным -1.0 (для анализа спиновой плотности).

2. Поверхности Ферми. Построение поверхности Ферми осуществляется при открытии файла *.BXSf: File -> Open Structure -> Open BXSf или в командной строке: xcrysdem -bxsf < BXSf file >

Файл создается программой ene2bxsf, который обрабатывает полученные ранее пакетом SIESTA собственные значения (на достаточно детализированной сетке в обратном пространстве). Так как XCrySDen требует, чтобы k -сетка включала точку Гамма, запрещено использование сдвинутой k -сетки, а также использование kgrid_cutoff, потому что он вводит сдвиг k -сетки по умолчанию. Поэтому нужно задать разбиение по трем векторам обратной решетки в блоке kgrid_Monkhorst_Pack, т.е.

```
%block kgrid_Monkhorst_Pack
24  0  0  0.
0   24 0  0.
0   0  24 0.
%endblock kgrid_Monkhorst_Pack
```

Скрипт должен быть запущен в каталоге, в котором содержатся файлы расчета SIESTA; после запроса SystemLabel, скрипт открывает следующие файлы:

...XV (для считывания векторов трансляции решетки),
 ...KP (для считывания неприводимых k -точек),
 ...EIG (для считывания собственных значения для всех k -точек и зон).

В результате работы скрипта создается файл .BXSf (два для спин-поляризованного расчета), содержащий упорядоченный список собственных значений (зона за зоной) для всех k -точек.

Обобщим назначение скриптов в директории /Util/Contrib/APostnikov/:

xv2xsf: .XV → .XSF (просмотр структуры и ячейки)

rho2xsf: .XV, .RHO или .LDOS → .XSF (атомы в выбранной ячейке и данные на сетке)

md2axsf: .XV, .ANI или .MD → .AXSF (анимация структуры)

eig2bxsf: .XV, .KP или .EIG \rightarrow .VXSf (поверхность Ферми)

3.4. Упражнение 3

Задание 1. Постройте сечение эквипотенциальных поверхностей зарядовой плотности углеродной нанотрубки (5,5). Рассмотреть плоскость, содержащую ось нанотрубки.

Задание 2. Убедитесь, что электроны вблизи энергии Ферми делокализованы. Для доказательства этого постройте локальную плотность состояний в интервале ± 0.5 эВ вблизи уровня Ферми.

Заключение

В настоящее время в распоряжении теоретиков имеется много компьютерных программ, имеющих разные области применения. Из текста пособия читатели должны были понять, что из существующих подходов к неэмпирическому моделированию (полноэлектронный и псевдопотенциальный) последний позволяет получать достоверные результаты за существенно меньшее время (и с использованием меньших вычислительных ресурсов). Кроме того, реализованный в квантово-механическом пакете SIESTA базис ЛКАО позволяет описывать молекулярные системы с использованием меньших ресурсов.

В результате изучения данного пособия студенты должны освоить методику проведения квантово-механических расчетов на примере моделирования физико-химических свойств (модуль Юнга, предел прочности, зонная структура, плотность электронных состояний, спектр оптического поглощения) углеродных нанотрубок.

Приведенные в пособии задания подобраны с учетом возможности их выполнения на персональных компьютерах. Такие задания имеют только учебный интерес. При увеличении размеров моделей и включении в рассмотрение адсорбции (атомарных или молекулярных комплексов) или дефекты структуры все задачи приобретают научный интерес. В качестве отправной точки исследований можно порекомендовать ознакомиться с обзором [35].

Библиографический список

- [1] Hohenberg P., Kohn W. Inhomogeneous Electron Gas // Physical Review. — 1964. — Vol. 136. — P. B864.
- [2] Kohn W., Sham L. J. Self-Consistent Equations INcluding Exchange and Correlation Effects // Physical Review B. — 1965. — Vol. 140. — P. A1133.
- [3] Perdew John P., Zunger Alex. Self-interaction correction to density-functional approximations for many-electron systems // Physical Review B. — 1981. — Vol. 23, no. 10. — P. 5048–5079. — 0706.3359.
- [4] Perdew J.P., Wang Y. Accurate and simple analytic representation of the electron-gas correlation energy // Physical Review B. — 1992. — Vol. 45. — P. 13244.
- [5] Perdew J.P., Burke K., Ernzerhof M. Generalized Gradient Approximation Made Simple. // Physical Review Letters. — 1996. — oct. — Vol. 77, no. 18. — P. 3865–3868. — URL: <http://www.ncbi.nlm.nih.gov/pubmed/10062328>.
- [6] Zhang Yingkai, Yang Weitao. Comment on “Generalized Gradient Approximation Made Simple” // Physical Review Letters. — 1998. — Vol. 80, no. 4. — P. 890–890.
- [7] Hammer B., Hansen L., Norskov J. Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals // Physical Review B. — 1999. — Vol. 59, no. 11. — P. 7413–7421.
- [8] Wu Zhigang, Cohen R. E. More accurate generalized gradient approximation for solids // Physical Review B. — 2006. — Vol. 73, no. 23. — P. 235116. — URL: <http://link.aps.org/doi/10.1103/PhysRevB.73.235116>.
- [9] Restoring the Density-Gradient Expansion for Exchange in Solids and Surfaces / John P. Perdew, Adrienn Ruzsinszky, Gábor I. Csonka et al. // Physical Review Letters. — 2008. — Vol. 100, no. 13. — P. 136406. —

URL: <http://link.aps.org/doi/10.1103/PhysRevLett.100.136406>.

- [10] Becke A. D. Density-functional exchange-energy approximation with correct asymptotic behavior // *Physical Review A*. — 1988. — Vol. 38, no. 6. — P. 3098–3100. — *PhysRevA.38.3098*.
- [11] Lee Chengteh, Yang Weitao, Parr Robert G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density // *Physical Review B*. — 1988. — Vol. 37, no. 2. — P. 785–789. — *PhysRevA.38.3098*.
- [12] Results obtained with the correlation energy density functionals of becke and Lee, Yang and Parr / Burkhard Miehlich, Andreas Savin, Hermann Stoll, Heinzwerner Preuss // *Chemical Physics Letters*. — 1989. — Vol. 157, no. 3. — P. 200–206.
- [13] Harris J. Simplified method for calculating the energy of weakly interacting fragments // *Physical Review B*. — 1985. — Vol. 31, no. 4. — P. 1770–1779.
- [14] Oda Tatsuki, Pasquarello Alfredo, Car Roberto. Fully Unconstrained Approach to Noncollinear Magnetism: Application to Small Fe Clusters // *Physical Review Letters*. — 1998. — Vol. 80, no. 16. — P. 3622–3625.
- [15] First principles simulations of the magnetic and structural properties of iron / V. M. García-Suárez, C. M. Newman, C. J. Lambert et al. // *European Physical Journal B*. — 2004. — Vol. 40. — P. 371–377.
- [16] Optimized basis sets for the collinear and non-collinear phases of iron / V M García-Suárez, C M Newman, C J Lambert et al. // *Journal of Physics: Condensed Matter*. — 2004. — Vol. 16, no. 30. — P. 5453–5459. — URL: <http://stacks.iop.org/0953-8984/16/i=30/a=008?key=crossref.e72b80db7bda2a8c2dba587f832f47dc>.
- [17] Phillips J.C. Energy-Band Interpolation Scheme Based on a Pseudopotential // *Physical Review*. — 1958. — Vol. 112. — P. 685–695.
- [18] Phillips J.C., Kleinman L. New method for calculating wave functions in crystals and molecules // *Physical Review*. — 1959. — Vol. 116. — P. 287–294.

- [19] Herring C. A new method for calculating wave functions in crystals // *Physical Review*. — 1940. — Vol. 57, no. 1934. — P. 1169–1177.
- [20] Troullier N., Martins J.L. Efficient pseudopotentials for plane-wave calculations. II. Operators for fast iterative diagonalization // *Physical Review B*. — 1991. — Vol. 43. — P. 1993–2006.
- [21] Kleinman L., Bylander D.M. Efficacious Form for Model Pseudopotentials // *Physical Review Letters*. — 1982. — Vol. 48. — P. 1425–1428. — URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.48.1425>.
- [22] Sankey O.F., Niklewski D.J. Ab initio multicenter tight-binding model for molecular-dynamics simulations and other applications in covalent systems // *Physical Review B*. — 1989. — Vol. 40. — P. 3979–3995.
- [23] Structural relaxation made simple / Erik Bitzek, Pekka Koskinen, Franz Gähler et al. // *Physical Review Letters*. — 2006. — Vol. 97, no. 17. — P. 170201.
- [24] Johnson D. D. Modified Broydens method for accelerating convergence in self-consistent calculations // *Physical Review B*. — 1988. — Vol. 38, no. 18. — P. 12807–12813.
- [25] Kubo Ryogo. Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems // *Journal of the Physical Society of Japan*. — 1957. — Vol. 12, no. 6. — P. 570–586.
- [26] Greenwood DA. The Boltzmann equation in the theory of electrical conduction in metals // *Proceedings of the Physical Society*. — 1958. — Vol. 71, no. 4. — P. 585.
- [27] Landauer R. Spatial variation of currents and fields due to localized scatterers in metallic conduction (and comment) // *J. Math. Phys.* — 1996. — Vol. 37. — P. 5259.
- [28] Generalized many-channel conductance formula with application to small rings / M. Büttiker, Y. Imry, R. Landauer, S. Pinhas // *Physical review B*. — 1985. — May. — Vol. 31, no. 10. — P. 6207–6215.
- [29] Datta S. *Quantum Transport: Atom to Transistor*. — Cambridge University Press, 2005. — P. 404. — ISBN: 9780521631457.

- [30] First-Principles Calculations in Real-Space Formalism: Electronic Configurations and Transport Properties of Nanostructures / K. Hirose, T. Ono, Y. Fujimoto, S. Tsukamoto. — Imperial College Press, 2005. — P. 253.
- [31] Density-functional method for nonequilibrium electron transport / M. Brandbyge, J.-L. Mozos, P. Ordejón et al. // Physical review B. — 2002. — Mar. — Vol. 65. — P. 165401.
- [32] Baroni Stefano, Giannozzi Paolo, Testa Andrea. Green's-function approach to linear response in solids // Physical Review Letters. — 1987. — Vol. 58, no. 18. — P. 1861.
- [33] Gonze Xavier, Vigneron J-P. Density-functional approach to nonlinear-response coefficients of solids // Physical Review B. — 1989. — Vol. 39, no. 18. — P. 13120.
- [34] Postnikov AV, Pagès Olivier, Hugel Joseph. Lattice dynamics of the mixed semiconductors (Be, Zn) Se from first-principles calculations // Physical Review B. — 2005. — Vol. 71, no. 11. — P. 115206.
- [35] Eletsii, A.V. and Zitserman, V.Yu. and Kobzev G.A. Nanocarbon materials: Physicochemical and exploitation properties, synthesis methods, and enegretic applications // High Temperature. — 2015. — Vol. 53, no. 1. — P. 130–150. — URL: <http://link.springer.com/10.1134/S0018151X15010034>.