Инструкция по использованию менеджера расчётов “СИМАН”

Аксенов Дмитрий

Москва, 2016 г.

Оглавление

Глава 1

Введение

Предлагаемый менеджер является оболочкой между пользователем и различными программами, предназначенными для выполнения компьютерного моделирования и теоретических расчётов. Использование менеджера позволяет существенно упростить запуск большого числа расчётов и их последующий анализ. Встроенные функции автоматически ведут протоколирование запущенных задач и сохраняют в базе данных основные результаты расчёта. За счёт представления результатов расчётов в стандартизированном виде, их последующий анализ существенно упрощается. В частности, достаточно дописать простые функции, которые будут строить рисунки и таблицы для вашей статьи на основе полученных результатов. В файле history сохраняются действия пользователя и их время. В файле log детально протоколируется работа менеджера. В файле calc.s в бинарном виде сохраняются все результаты расчётов и исходные структуры в компактном виде. Есть возможность быстрого продолжения расчетов.

Глава 2

Создание нового проекта для расчётов в VASP

Для создания нового проекта нужно создать папку с именем проекта и поместить туда файлы `start.py` (можно переименовать в название проекта, например `VTi.py`), `header.py` и `sets.py`.

Также необходимо скопировать папку с потенциалами (по умолчанию `potpaw_P` и создать папку `/geo`. В файле `header.py` необходимо указать основные пути и имя кластера.

Глава 3

Добавление нового расчёта

Придумать имя для группы, к которой будет относиться расчет. Например `hcp` по типу кристаллических структур, для которых будут проводиться расчеты. Придумать имя для расчёта. Как правило, это имя структуры плюс дополнительные обозначения, которые могут уточнять тип расчета. Например `hs222` (расшифровка сокращения: hcp cell; sizes: 222) В файле header.py в конце функции `update_des()` прописать параметры расчета в виде:

```
struct_des['hs221']      = des("hcp", "hex cell: 8 atoms; ");
```

Первый параметр `hcp` - это имя группы расчётов. Второй параметр - произвольное описание расчёта, чтобы было понятно, что из себя представляет структура и расчет. Подготовить входной файл с начальной кристаллической структурой для данного расчета в формате `Abinit` ?описать формат?. Файл должен иметь расширение .geo; внутри файла должна быть строка `version 1`. Номер версии может быть произвольным; его нужно также указать в имени файла. Например файл будет называться hs221.1.geo. Файл необходимо поместить в папку geo/hcp/hs221/. Теперь для данного расчёта нужно подготовить набор параметров (`set`), которые будет использовать `VASP`. Для этого необходимо открыть файл sets.py и изменить необходимые параметры в имеющемся по умолчанию наборе под именем `8`. Далее, если необходимы другие наборы параметров, их можно получить из существующих, используя команду

```
s = inherit_iset('8new', '8', varset);
```

По традиции в начале названия набора параметров необходимо указать цифру. `8` - полный расчет энергии без релаксации; `9` - оптимизация положений атомов; `2` - полная релаксация геометрии, объема и положений атомов. В файле VTi.py в третьем разделе добавить строку

```
add_loop('hs222',      '8', 1, up = 'up1')
```

и запустить скрипт. Менеджер автоматически создаст все файлы и скопирует их на кластер. Расчет станет доступным внутри скрипта как `calc[('hs222','8', 1)]`. Объект `calc[('hs222','8', 1)].init` содержит информацию об исходной атомной структуре. Перейти на кластер и запустить расчёты, набрав в консоли `./run`. Когда расчёт будет закончен переименовать имеющуюся команду `add_loop` в `res_loop` и запустить скрипт. Менеджер скопирует файл OUTCAR с кластера на рабочий компьютер и проанализирует его. На экране появятся основные данные в следующем порядке:

```
| Имя файла | Полн. энергия | a | c | Размеры ячейки | Объем ячейки
| Плотн. к точек | Тензор напр. (MPa) | Давление | N атомов | Время расч. ч()
| N шаг. рел., среднее N электрон. шаг на 1 шаг рел., кол. электрон. шагов
| Кол. предупреждений | total drift | Кол. операций симметрий |
```

Полученную строку легко использовать для создания LaTeX 2_εтаблиц рабочего дневника. Объект `calc[('hs222','8', 1)].end` содержит информацию о структуре после релаксации

Глава 4

Возможности системы

4.1. DFT+U расчеты

Для выполнения DFT+U расчёта создаем специальный сет параметров `8U` на основе `8`:

```
s = inherit_iset('8U', '8', varset, override = 1) #
s.set_vaspp('ISTART', 1) #use from previous step
s.set_vaspp('ICHARG', 1) #
s.set_vaspp('LDAU', '.TRUE.')
s.set_vaspp('LDAUTYPE', 2) #Dudarev
s.set_vaspp('LDAUPRINT', 1) #
s.set_vaspp('LASPH', '.TRUE.') #
s.set_vaspp('LDAUL', {'Ti':2  }   ) # 2=d-orbitals if one number is
   provided than it is used for all species
s.set_vaspp('LDAUU', {'Ti':4.2}   ) #if one number is provided than it
    is used for all species
s.set_vaspp('LDAUJ', {'Ti':0.5}   )
```

Обратите внимание, что параметры которые в INCAR должны указываться в виде списка для каждого сорта атомов, задаются с помощью функции `s.set_vaspp()` в виде словаря для каждого атомного элемента: `LDAUL`, `LDAUU`, `LDAUJ`. В случае, когда элемент не указан, для них будут использованы нулевые значения (`-1` для `LDAUL`).

В менеджере "Симан"реализован метод `U ramping` [Meredig2010]. Чтобы им воспользоваться необходимо запустить расчёт в виде:

```
add_loop('Li4Ti8O16.r', '8U', 1, 'up1', calc_method = 'u_ramping',
   u_ramping_region = (0, 4, 0.2), savefile = 'o')
```

Параметр `u_ramping_region` задаёт диапазон изменения параметра `U` от 0 до 3.8 эВ с шагом 0.2 эВ. Изменение производится только для элементов, у которых ненулевые значения. После окончания расчета на сервере остается файлы 1.U2_2.OUTCAR для каждого `U`, а также для последнего расчёта сохраняется 1.U3_8.DOSCAR и 1.U3_8.CHGCAR. Также создается файл ENERGIES, который содержит полные энергии для каждого `U`. Как обычно, для считывания результатов достаточно выполнить команду

```
res_loop('Li4Ti8O16.r', '8U', 1, 'up1', calc_method = 'u_ramping')
```

При этом будет считан файл 1.U3_8.OUTCAR для последнего `U`, а значения `U` и соответствующие им энергии сохраняться в полях `self.u_ramping_u_values` и `self.u_ramping_energies`.

Глава 5

Описание основных классов и структур

В программе использует два класса (смотри что такое класс в описании языков объектно-ориентированного программирования): `class Calculation()` - основной класс. С его помощью создаются экземпляры (объекты), которые содержат в себе всю информацию об одном расчёте. Для примера предположим, что мы создали объект `hcpTi = Calculation()`. Это можно сделать вручную, но в программе такие объекты создаётся в процессе работы функции `add_loop()` и автоматически вносятся в словарь `calc`. Коротко опишем, что делает эта функция: После создания объекта, необходимо считать входную геометрию. Для этого используется метод `read_geometry()`. В результате данный объект содержит всю основную исходную информацию о расчёте. Вот список доступных полей:

`class Set()` - специальный класс, который используется для описания наборов контролирующих расчёт параметров.

Глава 6

Список изменений в программе СИМАН2

Для изменения параметров сета можно и нужно использовать только эти методы и атрибуты:

```
s.set_potential()
s.set_relaxation_type()
s.set_add_nbands(float)
```

- устанавливает внутренний атрибут `add_nbands`, указывающий что минимально необходимое число зон будет увеличено в `add_nbands` раз. Рекомендуемое значение - `1.25`. Но для маленьких ячеек может потребоваться `2-4`.

`s.kpoints_file` - `1` или `0` - определяет будет ли создан файл с к-точками

`s.set_vaspp(''VASP_KEY'', ''VALUE'')` - для избежания путаницы с `Abinit` теперь для всех остальных параметров, кроме типа релаксации используется последний метод. Параметры сохраняются во внутреннем словаре `s.vasp_params`.

В случае если параметр `KSPACING` не удовлетворяет вашим потребностям, можно явно задать необходимый набор k-точек. `s.set_ngkpt( (int, int, int) )` - внутренний аттрибут `s.ngkpt` с более высоким приоритетом чем `s.vasp_params['K` !!! Настоятельно рекомендуется использовать предложенные методы и не работать напрямую с атрибутами.