

Minichem 0.1

Руководство пользователя

А. В. Олейниченко

22 октября 2016 года

Глава 1

Общие сведения

Данный документ является подробным руководством пользователя к квантовохимической программе Minichem версии 0.1. Программа является учебной, поэтому может содержать ошибки и работать продолжительное время даже на небольших задачах.

1.1 Возможности программы

Доступен расчет энергии в точке методами RHF и UHF.

Глава 2

Установка и запуск

2.1 Установка

Программа Minichem устанавливается из исходных кодов, доступных на GitHub по адресу <https://github.com/aoleynichenko/minichem>. Для сборки требуются установленные библиотеки LAPACK, BLAS, OpenMP. Детали их установки зависят от дистрибутива и здесь не приводятся. На текущий момент мейкфайлы написаны с учетом специфики ОС Linux. Команды для сборки Minichem:

```
cd \${MINICHEM_HOME}/src
make
make install
```

Исполняемый файл Minichem появится в `\${MINICHEM_HOME}/bin`.

2.2 Запуск

Minichem – консольное приложение. Формат команды запуска:

```
minichem <options> <input-files>
```

Удобно добавить директорию `\${MINICHEM_HOME}/bin` в `$PATH`.

По умолчанию вывод результатов осуществляется в стандартный поток вывода `stdout`.

2.3 Диагностика ошибок. Лог

Кроме основного вывода, Minichem также ведет лог (по умолчанию – `minichem.log`), в который записываются сообщения о ходе исполнения программы. Лог содержит развернутые сообщения о произошедших ошибках

и предупреждениях и может оказаться полезным при отладке как самого Minichem, так и input-скриптов.

Глава 3

Описание формата входных файлов

3.1 Язык Minichem

Для входных файлов Minichem разрабатывается специальный скриптовый язык программирования, ориентированный на применение в вычислительной химии и наследующий черты языков Psithon, NWChem и JavaScript.

Язык Minichem чувствителен к регистру (кроме названий базисных наборов и символов химических элементов). Символы конца строки и лишние пробельные символы игнорируются, что в принципе позволяет записать весь входной файл в одну строчку. Это возможно благодаря тому, что каждый оператор языка «знает», сколько аргументов должен принять.

Комментарии начинаются с символа `#` и продолжаются до конца строки (аналогично Python и Bash).

3.2 Типы данных

Molecule. Объекты типа Molecule содержат информацию о заряде молекулы, мультиплетности основного состояния и координатах атомов. Для создания объекта Molecule используется ключевое слово `mol`:

```
mol LiH {  
  mult 1  
  charge 0  
  units atomic  
  H      0.0000  0.0000  0.0000  
  Li     1.4632  0.0000  0.0000  
}
```

Обязательными в определении молекулы являются только ключевое слово `mol` и координаты атомов в формате `x,y,z`. Параметры молекулы, указываемые ключевыми словами `mult`, `charge`, `units` и т.д. необязательны. После ключевого слова `mol` опционально указывается имя молекулы; если оно не указано, объявленная молекула становится текущей (доступной для квантовохимических подпрограмм). Пример объявления безымянной молекулы:

```
# methane molecule
mol {
  C    0.0000    0.0000    0.0000
  H    0.6281    0.6281   -0.6281
  H   -0.6281   -0.6281   -0.6281
  H    0.6281   -0.6281    0.6281
  H   -0.6281    0.6281    0.6281
}
```

Вместо символов элементов можно указывать заряды ядер.

Basis Set. Используемый в расчетах базис атомных орбиталей представляет собой набор гауссовых функций, центрированных на атомах; технически это динамический массив `std::vector`, состоящий из объектов типа `libint2::Shell`. Набор АО для каждого атома генерируется по «шаблону», который не совсем точно может быть назван *базисным набором* (basis set). Задать базисный набор можно с помощью ключевого слова `basis`:

```
basis cc-pVTZ
```

Данная директива «заставляет» Minichem генерировать АО из базисного набора `cc-pVTZ` на *всех* атомах заданной молекулы. Тот же смысл имеет конструкция

```
basis {
  * library cc-pVTZ
}
```

Можно задавать базисные наборы для *каждого* типа атомов:

```
basis mixed_bs {
  O library sto-3g
  H library 6-31g**
}
```

Названия библиотечных базисных наборов, указываемые после ключевого слова `library`, регистронезависимы и могут содержать любые символы (знаки дефиса, звездочки и т.д.). Наконец, можно задать базисный набор вручную (экспоненты записываются в левой колонке, каждая следующая колонка отвечает коэффициентам сжатой базисной функции):

```
basis custom-basis-HeH {
  spherical
  H library cc-pvtz
  # actually, it is hand-written cc-pVTZ basis set for He
  He    S
```



```

234.0000000    0.0025870
35.1600000     0.0195330
7.9890000     0.0909980
2.2120000     0.2720500
He      S
0.6669000     1.0000000
He      S
0.2089000     1.0000000
He      P
3.0440000     1.0000000
He      P
0.7580000     1.0000000
He      D
1.9650000     1.0000000
}

```

Как очевидно из последнего примера, задание базисных наборов вручную может соседствовать с директивами **basis**. Это может оказаться удобным для расширения «стандартного» базисного набора диффузными функциями или использования базиса, отсутствующего в библиотеке Minichem.

По умолчанию Minichem генерирует базис сферических гауссовых функций, «включить» декартов базис можно указанием ключевого слова **cartesian**:

```

basis cart-3-21g {
  cartesian
  N      S
    242.7660000    0.0598657
    36.4851000    0.3529550
    7.8144900    0.7065130
  N      SP
    5.4252200    -0.4133010    0.2379720
    1.1491500    1.2244200    0.8589530
  N      SP
    0.2832050    1.0000000    1.0000000
}

```

Ссылки на оригинальные публикации приведены в начале каждого библиотечного файла с базисным набором. Стиль оформления базисных наборов позаимствован у программного пакета NWChem.

Литература

- [1] LIBINT: A library for the evaluation of molecular integrals of many-body operators over Gaussian functions, Version 2.1.0 (beta) Edward F. Valeev, <http://libint.valeev.net/> .