

minichem

Версия 1.0

Руководство пользователя

Александр Олейниченко

2 сентября 2018 г.

Содержание

1	Общие сведения	2
2	Компиляция и тестирование программы	3
3	Структура входных файлов и запуск задач	3
4	Директивы входных файлов	6
4.1	start	6
4.2	echo	6
4.3	memory	6
4.4	nproc	6
4.5	task	7
4.6	geometry	7
4.7	charge	7
4.8	basis	7
4.9	scf	8
4.10	out	9
5	Примеры входных файлов для типовых задач	10

1 Общие сведения

`minichem` – это небольшая учебная квантовохимическая программа, разработанная с целью поближе познакомиться с техническими аспектами реализации различных методов решения электронного уравнения Шредингера. В ходе работы над программой автор опирался как на учебные пособия, так и на оригинальные работы и (к сожалению, немногочисленные) веб-ресурсы (см. список литературы). Среди использованных материалов два учебных пособия представляются наиболее полно отражающими сложности и особенности алгоритмической реализации методов квантовой химии, а потому особенно важными:

- [1] A. Szabo, N. Ostlund, "Modern Quantum Chemistry";
- [2] T. Helgaker, P. Jorgensen, J. Olsen, "Molecular Electronic-Structure Theory".

Исходный текст программы полностью написан на языке программирования C (стандарт C99). В настоящий момент программа ориентирована на Unix-подобные операционные системы; тем не менее, в будущем программа может быть портирована и на другие платформы.

Ниже перечислены основные возможности программы, предоставляемые пользователю:

- расчет энергии системы (атома или молекулы) при заданной геометрии;
- ограниченный (RHF) и неограниченный (UHF) методы Хартри-Фока-Рутаана;
- техника DIIS улучшения сходимости процедуры решения уравнений ССП;
- распараллеливание на основе технологии OpenMP;
- доступные химические элементы: вся периодическая система (внимание: программа никак не учитывает релятивистские эффекты!);
- базисные наборы: декартовы, с произвольным угловым моментом базисных функций.

Исходный код проекта доступен на Github:

<https://github.com/aoleynichenko/minichem>

Автор будет рад любым замечаниям, вопросам и предложениям:

alexvoleyichenko@gmail.com.

2 Компиляция и тестирование программы

Перед началом сборки необходимо убедиться, что на Вашей машине установлены следующие библиотеки:

- `libc` – стандартная библиотека языка C;
- `MPI` – библиотека для параллельных вычислений для систем с распределенной памятью;
- `OpenMP` – библиотека для параллельных вычислений для систем с общей памятью;
- `BLAS` и `LAPACK` – библиотеки линейной алгебры.

Для непосредственно сборки потребуются инструменты `CMake` и `make`, а также компиляторы языка C (рекомендованы компиляторы GNU или Intel).

Для компиляции зайдите в домашний каталог `minichem` и выполните следующие команды:

```
$ mkdir build && cd build
$ cmake ..
$ make [-jN]
```

В случае успешной компиляции в каталоге `build` появится исполняемый файл `minichem.x`.

Тесты размещены в папке `test`. Система тестирования написана на языке Python2 (должен быть предустановлен). Для тестирования выполните команды:

```
$ cd test
$ python test.py
```

3 Структура входных файлов и запуск задач

Запуск программы `minichem`:

```
$ minichem.x <input-file.inp>
```

Программа направляет результаты работы в стандартный поток вывода, поэтому удобнее сразу перенаправлять его в файл командой `tee`:

```
$ minichem.x <input-file.inp> | tee <output-file.out>
```

Дизайн языка входных файлов позаимствован у программы NWChem [3]. Так же как и NWChem, **minichem** работает как интерпретатор, «исполняя» входной файл как программу, написанную на очень простом скриптовом языке. Язык **minichem** во многом совпадает с языком NWChem; везде, где это было возможно, названия директив и секций были выбраны так, чтобы входной файл мог быть интерпретирован обоими квантовохимическими программами без дополнительных правок.

Входные файлы **minichem** строятся из директив и секций. Директивы – это однострочные команды, секции объединяют наборы директив. Как правило, секции задают параметры для работы какого-либо модуля (например, параметры сходимости процедуры ССП). **minichem** последовательно считывает строки входного файла и соответственно заданным в нем параметрам изменяет свои внутренние переменные. Как только **minichem** доходит до директивы **task**, исполнение входного файла останавливается и начинается расчет. В одном входном файле допустимы несколько директив **task**; между ними пользователь может менять параметры расчетов, организовывая, таким образом, каскад вычислений.

Язык **minichem** также поддерживает однострочные комментарии, начинающиеся со знака `'#'`. Язык нечувствителен к регистру.

В качестве примера входного файла приведем файл, составленный для молекулы бензола в базисе STO-3G. Пояснения ко всем директивам содержатся в комментариях. Подробные описания всех директив и секций даны в разделе 4. Несколько других примеров можно найти в разделе 5.

```
# C6H6 single-point energy
#
# RHF molecular orbitals will be written to the molden-format
# file c6h6.mos

# name for the task
start C6H6

# allowed memory usage
memory 10 mb

# print input file before executing it
echo

# number of OpenMP threads
nproc 8

# geometry (cartesian)
# default: charge = 0
geometry
```

```

C      0.000      1.396      0.000
C      1.209      0.698      0.000
C      1.209     -0.698      0.000
C      0.000     -1.396      0.000
C     -1.209     -0.698      0.000
C     -1.209      0.698      0.000
H      0.000      2.479      0.000
H      2.147      1.240      0.000
H      2.147     -1.240      0.000
H      0.000     -2.479      0.000
H     -2.147     -1.240      0.000
H     -2.147      1.240      0.000
end

# basis set specification
# (keyword SPHERICAL -- only for compatibility with NWChem)
basis "ao basis" SPHERICAL
H      S
      3.42525091          0.15432897
      0.62391373          0.53532814
      0.16885540          0.44463454
C      S
      71.6168370          0.15432897
      13.0450960          0.53532814
      3.5305122           0.44463454
C      S
      2.9412494          -0.09996723
      0.6834831           0.39951283
      0.2222899           0.70011547
C      P
      2.9412494          0.15591627
      0.6834831          0.60768372
      0.2222899          0.39195739
end

# options for the SCF module
# by default: singlet
scf
  print "overlap"      # print AO overlap integrals
  diis 5               # enable DIIS, max subspace dim = 5
  maxiter 20           # max number of SCF iterations
end

# export calculated data (molecular orbitals)
out
  molden      # to the MOLDEN .mos format
end

```

```
# do RHF calculation
task scf
```

4 Директивы входных файлов

4.1 start

Синтаксис:

```
start <string name>
```

Директива **start** задает небольшой не содержащий пробелов идентификатор задачи. Этот идентификатор используется затем в названиях временных и выходных файлов.

4.2 echo

Синтаксис:

```
echo
```

Если директива **echo** встречается во входном файле, **minichem** направляет содержимое входного файла в стандартный вывод. Рекомендуется всегда использовать эту директиву.

4.3 memory

Синтаксис:

```
memory <integer> <string units>
# <units>: one of b (bytes), kb, mb, mw (megawords), gb
```

Максимальное количество оперативной памяти, которое может быть выделено и использовано программой.

4.4 nproc

Синтаксис:

```
nproc <integer>
```

Число OpenMP-нитей (для параллельного исполнения).

4.5 task

Синтаксис:

```
task <string theory>
```

Директива задает квантовохимический метод, который должен быть использован для решения электронной задачи.

Поскольку в настоящее время в программе реализован только метод Хартри-Фока, директива **task** может вызываться только с аргументом **scf**:

```
task scf
```

4.6 geometry

Синтаксис:

```
geometry [units <string units default angstroms>]  
  [charge <integer default 0>]  
  [mult <integer default 1>]  
  <string elem> <real x y z>  
  . . .  
end
```

Составная директива **geometry** задает декартовы координаты входящих в молекулу атомов, единицы измерения расстояний (бор или ангстрем), заряд молекулы и спиновую мультиплетность. По заданным значениям заряда и мультиплетности программа может автоматически выбрать, какой вариант метода Хартри-Фока необходимо использовать.

4.7 charge

Синтаксис:

```
charge <integer>
```

Директива задает общий заряд системы (в атомных единицах). Добавлена для совместимости с NWChem.

4.8 basis

Синтаксис:

```
basis [<string name default "ao basis">] [spherical|cartesian]  
  <string elem> <string shell_L>  
    <real exponent> <real list_of_coefficients>  
  . . .
```

```
. . .  
end
```

Описание базисного набора. В настоящее время реализованы только декартовы базисные наборы (ключевое слово **cartesian**), поэтому ключевое слово **spherical** сейчас необходимо только для совместимости с NWChem.

Коэффициенты сжатых функций расположены в колонках; первая колонка – показатели экспонент гауссовых примитивов.

Угловой момент блока базисных функций (<**string shell_L**>) обозначается, как обычно, буквами S, P, D ... Можно использовать сокращение SP (одна сжатая функция типа *s*, вторая – типа *p*). **minichem** не ставит ограничений на максимальный угловой момент базисных функций.

4.9 scf

Синтаксис:

```
scf  
[rhf | uhf]  
[singlet | doublet | triplet | quartet | quintet]  
[guess (core | eht)]  
[direct | nodirect]  
[maxiter <integer max_no_of_iterations default 50>]  
[diis [<integer subspace_dim default 5>] | nodiis]  
[print <string what>]  
[noprint <string what>]  
end
```

Составная директива **scf** используется для задания параметров процедуры решения уравнений ССП. Может содержать одну или несколько директив из следующего списка:

guess начальное приближение к МО. Возможные значения аргумента:

- **core** – приближение голых ядер: полное пренебрежение двух-электронными интегралами. Хорошо работает только для систем с небольшим числом электронов;
- **eht** – расширенный метод Хюккеля [4] (в варианте Вольфсберга-Гельмгольца [5]).

По умолчанию – расширенный метод Хюккеля.

rhf/uhf вариант метода Хартри-Фока (**rhf** – ограниченный, **uhf** – неограниченный).

`singlet/doublet/triplet/quartet/quintet` спиновая мультиплетность системы.

`maxiter` максимальное число итераций решения уравнений ССП. Значение по умолчанию: 50.

`diis` использовать технику DIIS [6, 7] для ускорения сходимости уравнений ССП. Необязательный аргумент директивы – максимальная размерность пространства итераций, которые используются для экстраполяции (значение по умолчанию: 5). Использование DIIS может быть рекомендовано практически во всех случаях и поэтому включено по умолчанию.

`nodiiis` не использовать технику DIIS.

`direct` использовать «прямой» вариант метода ССП (двухэлектронные интегралы рассчитываются на ходу на каждой итерации, а не считываются из предварительно подготовленного файла). «Прямой» ССП может работать на порядок медленнее «обычного». Иногда может быть полезен в случае нехватки дискового пространства.

`nodirect` использовать «обычный» вариант метода ССП (хранение интегралов на диске) (по умолчанию).

`print` какая дополнительная информация должна быть напечатана (но не печатается по умолчанию). Тип аргумента – строка в двойных кавычках. Возможные аргументы директивы `print` приведены в таблице:

"final vectors analysis"	коэффициенты разложения МО
"overlap"	интегралы перекрывания в АО-базисе
"kinetic"	интегралы оператора кинетической энергии
"potential"	интегралы оператора электрон-ядерной потенциальной энергии
"eri"	двухэлектронные кулоновские интегралы

`noprint` работает прямо противоположно директиве `print` (запрещает печатать дополнительной информации). Аргументы такие же, как у `print` (см. выше).

4.10 out

Синтаксис:

```
out
  [molden]
end
```

Составная директива `out` позволяет экспортировать результаты расчетов для дальнейшей их обработки другими программами. Может содержать следующие директивы:

`molden` экспортировать информацию о базисном наборе и молекулярные орбитали в файл формата MOLDEN (`.mos`) [8, 9, 10]. В настоящий момент экспорт реализован только для метода RHF.

5 Примеры входных файлов для типовых задач

UHF-расчет атома лития в базисе STO-3G:

```
start Li
echo

geometry units atomic
  Li 0 0 0
end

basis "ao basis" SPHERICAL
Li    S
      16.1195750      0.15432897
      2.9362007      0.53532814
      0.7946505      0.44463454
Li    SP
      0.6362897      -0.09996723      0.15591627
      0.1478601      0.39951283      0.60768372
      0.0480887      0.70011547      0.39195739
end

scf
  uhf
  diis
  doublet
end

task scf
```

UHF-расчет дублетной частицы CH_3 :

```

start CH3
echo

geometry
C      0.08745162      -0.08744725      -0.08742186
H      0.52503428       0.78909888      -0.52508604
H     -0.78884450      -0.52530342      -0.52536318
H      0.52530257      -0.52529218       0.78892712
end

# STO-3G
basis "ao basis" SPHERICAL
H      S
      3.42525091          0.15432897
      0.62391373          0.53532814
      0.16885540          0.44463454
C      S
      71.6168370          0.15432897
      13.0450960          0.53532814
      3.5305122           0.44463454
C      S
      2.9412494          -0.09996723
      0.6834831          0.39951283
      0.2222899          0.70011547
C      P
      2.9412494          0.15591627
      0.6834831          0.60768372
      0.2222899          0.39195739
end

scf
  uhf
  doublet
end

task scf

```

Список литературы

- [1] A. Szabo and N.S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Books on Chemistry. Dover Publications, 1996.
- [2] T. Helgaker, P. Jorgensen, and J. Olsen. *Molecular Electronic-Structure Theory*. Wiley, 2008.
- [3] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, and W.A. de Jong. NWChem: a comprehensive and scalable open-source solution for large scale molecular simulations. *Comput. Phys. Commun.*, 181:1477, 2010.
- [4] R. Hoffmann. An Extended Hückel Theory. I. Hydrocarbons. *J. Chem. Phys.*, 39(6):1397 – 1412, 1963.
- [5] M. Wolfsberg and L. Helmholz. The Spectra and Electronic Structure of the Tetrahedral Ions MnO_4^- , CrO_4^{2-} , and ClO_4^- . *J. Chem. Phys.*, 20(5):837, 1952.
- [6] P. Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chem. Phys. Lett.*, 73(2):393, 1980.
- [7] P. Pulay. Improved SCF convergence acceleration. *J. Comput. Chem.*, 3(4):556.
- [8] G. Schaftenaar and J.H. Noordik. Molden: a pre- and post-processing program for molecular and electronic structures. *J. Comput. Aided Mol. Des.*, 14:123, 2000.
- [9] G. Schaftenaar, E. Vlieg, and G. Vriend. Molden 2.0: quantum chemistry meets proteins. *J. Comput. Aided Mol. Des.*, 31:789, 2017.
- [10] The Molden Format. http://www.cmbi.ru.nl/molden/molden_format.html.