

第五次作业——量子计算化学程序说明

Tutorial of Quantum Chemistry Programing - ver.01

</p>

注：本程序说明最初使用Jupyter完成(ipynb格式)，保留并转化成诸多格式如markdown, tex, pdf, html。但转化中会出现一些问题比如latex公式显示的问题，以及排版的问题。这里统一建议使用html或ipynb格式阅读本说明。

0.成员与分工

小组成员：贺麒麟，资治，刘昊明，腾达，乔卓然，贺鑫

程序部分：由 资治，腾达、乔卓然、贺鑫负责完成

文档部分：由 贺麒麟、刘昊明负责完成

分工不是绝对固定，有相互穿插。

本程序是小组合作的在github私人项目 <https://github.com/Utenaq/2018QC-Project-Ab-initio-wavefunction-program> (<https://github.com/Utenaq/2018QC-Project-Ab-initio-wavefunction-program>)，你也可以找到一个公共版本 <https://github.com/XShinHe/Heisenberg> (<https://github.com/XShinHe/Heisenberg>)，两者初期是一致的，前者由乔卓然负责，后者由贺鑫负责。

1.编译和使用方法

编译方法

在主目录或./src子目录下，使用make命令进行编译，需要g++支持。

使用方法

参数	内容	补充
main -h	获得使用帮助	
main -d	进行默认的测试计算 (计算HeH+分子)	测试文件位于../test目录
main -f [file.gjf]	进行其他测试文件的测试	测试文件位于../test目录, 必须使用.gjf文件格式

可选的的测试文件位于../test下，包括H.gjf, He.gjf, H2.gjf, HeH.gjf, H4.gjf, CH4.gjf.

使用的示例如main -f ../test/H2.gjf.

2.建模的类的定义的说明

从输入文件到构建计算的基组矩阵空间，使用了如下的建模结构：

以下类来自于basic_Global.h文件

Point

属性

名称	数据类型	含义
x	int32	位置参数
y	int32r	位置参数
z	int32	位置参数
name	string	代表一个点的名字，抽象属性

方法

名称	接受参数	返回类型	含义
norm	x, y, z	double	求Point坐标平方和的均方根
norm2	x, y, z	double	求Point坐标的平方和
ref_Point	PointA	Point	求Point与参考点A的坐标差值所得到的新Point
ref_norm	PointA	double	求ref_Point的norm值
ref_norm2	PointA	double	求ref_Point的norm2值
operator+	PointA	Point	定义两个Point的加法运算，所得新Point的坐标为它们的x,y,z坐标分别相加
operator-	PointA	Point	定义两个Point的减法运算，所得新Point的坐标为它们的x,y,z坐标分别相减
operator*	double	Point	定义Point和一个数值的乘法运算，所得新Point的坐标为原x,y,z坐标分别乘上接受的常数
operator/	double	Point	定义Point和一个数值的除法运算，所得新Point的坐标为原x,y,z坐标分别除以接受的常数
operator*	PointA	double	求两个Point的内积值
ostream & operator<<	ostream &output, Point &P	ostream	输出点具备有的性质
dist_AB	Point A, Point B	double	求A，B两点距离
dist_AB2	Point A, Point B	double	求A，B两点距离的平方

Orbital (继承自Point)

额外属性

名称	数据类型	含义
L	int	高斯函数x方向的角动量参数
M	int	高斯函数y方向的角动量参数
N	int	高斯函数z方向的角动量参数
alpha	double	高斯函数的alpha

额外方法

名称	接受参数	返回类型	含义
set_Alpha	double	int	设置轨道的alpha值
set_LMN	int ll,mm,nn	int	设置轨道的l,m,n的值
set_XYZ	double xx,yy,zz	int	设置轨道的x,y,z的值
set_XYZ	Point P	int	将某一点的坐标作为轨道的三维空间坐标
get_Point	/	Point	获取Orbital在Point类上的性质，并将其储存在一个新的点中
get_LMN	int ll,mm,nn	int	获取某一个高斯函数的l,m,n
get_XYZ	double xx,yy,zz	int	获取Orbital的三维空间坐标
conv_Aunit	/	int	将轨道的空间坐标转化为原子单位
normGTO	/	double	计算高斯函数的归一化系数
normGTP	double a	double	计算高斯函数归一到a时的归一化系数

Orbital_cgto (继承自Orbital)

额外属性

名称	数据类型	含义
cn	int	cgto中的高斯函数个数
coeffs	double	高斯函数前的收缩系数
alpha	double	高斯函数的alpha值
gtos	Orbital	cgto中的高斯函数

额外方法

名称	接受参数	返回类型	含义
set_Cgto	int num	int	设置Cgto中高斯函数的数量，并建立列表以存储高斯函数
get_CA	idx,cc,aa	int	读取Cgto中编号为第idx个高斯函数的收缩系数和alpha值
set_LMN	ll,mm,nn	int	设置Cgto中每个高斯函数的L,M,N值
set_XYZ	xx,yy,zz	int	设置Cgto中每个高斯函数的x,y,z值
set_XYZ	Point P	int	利用点P的坐标设置Cgto中每个高斯函数的x,y,z值
conv_AUnit	/	int	将坐标转化为原子单位
get_Bohr	/	int	返回高斯函数角动量的和
normGTO	int k	double	返回cgto的第k个高斯函数的归一化系数

Atom (继承自Point)

额外属性

名称	数据类型	含义
znum	int	原子序数
mass	double	原子质量
ncgto	int	描述该原子的cgto的数量
cgto	Orbital_cgto	contracted GTO
perd	int	原子所处的周期
fmly	int	原子所处的族
indx	int	在几何构型中的编号
frag	int	所属片段的编号

额外方法

名称	接受参数	返回类型	含义
get_Point	/	Point	返回原子具备的点的性质
get_Point	Point &P	int	返回原子具备的点的性质
read_Atom	string line	int	从一个字符串中读取原子的名称和位置
link_Info	string my_name	int	根据名称找到对应原子的核电荷数、质量、周期和族
set_Basis	string my_bname,int mysplit, int mynumcs[], int myidxcs[]	int	从基组文件中找到所选原子的高斯基函数的总操作安排，会调用到之后定义的函数
set_Basisspace	int split, int numcs[], int idxcs[]	int	根据所给基组的种类首先生成储存空间以存放基函数
read_Basis	fstream& input	int	读取基组文件中收缩系数和alpha信息
conv_AUnit	/	int	将Atom的x,y,z值转化为原子单位
operator<<	stream &output, Atom &At	ostream	输出原子的信息

额外属性

名称	数据类型	含义
Natom	int	分子所含原子个数
iatom	int	原子的序号
atoms	Atom	分子所含的所有原子

System (继承自Molecule)

额外属性

名称	数据类型	含义
Nmol	int	系统中分子的个数
imol	int	系统中分子的序号
moles	Molecule[]	系统中的所有分子
Bname	string	系统所选用基组类型的名称
Nbasis	int	系统中cgto的数量
split	int	系统中原子价层轨道的劈裂数
numcs	int[]	用于存放系统中基组收缩系数列表
idxcs	int[]	系统中基组收缩系数的逐个加和的列表
idmap	int[]	系统中原子的第一个cgto在列表中的index

额外方法

名称	接受参数	返回类型	含义
set_Natom	int num	int	设定系统所继承的分子属性，包括原子个数、原子编号和idmap
read_Atom	string line	int	读取输入文件中的原子坐标和原子序号
set_Basis	string my_bname	int	根据基组名称，得到numcs,idxcs和split的信息
set_Map	/	int	在原子列表和基函数列表之间建立映射关系
OrbC	int &idx	Orbital_cgto	返回序号为idx的cgto
count_Znum	/	int	返回系统的总核电荷数
conv_AUnit	/	int	转化为原子单位
solve_Top	string line	int	将系统分解为不同的分子或组成部分
operator<<	stream &output, System &S	ostream	输出系统的信息
operator[]	int &idx	Orbital_cgto	返回序号为idx的cgto

以下类来自于basic_Parser.h文件

Task

属性

名称	数据类型	含义
Hiffile	string	输入文件名称
Logfile	string	输出文件名称
Maxmen	int	
job	string	工作任务
Method	string	方法
Basis	string	基组标题
Title	string	标题
Charge	int	电荷
Smuti	int	自旋多重度
Natom	int	原子个数
Nelec	int	电子个数
Nbasis	int	cgto个数

方法

名称	接受参数	返回类型	含义
set_l0	string Hiffile, string Logfile	int	选取输入文件与输出文件
set_job	string job, string method, string basis	int	设置工作任务
read_Predo	/	int	从输入文件中读取原子的信息
read_Task	/	int	从输入文件中读取计算内容的信息
Taskparser	string term	int	工作类型、方法和基组的分析程序
operator<<	stream &output, System &S	ostream	输出task的信息

以下类来自于basic_SCF.h文件

属性

名称	数据类型	含义
report	ofstream	以输出方式打开文件
tasklink	Task	scf调用的task类
SYSlink	System	scf调用的system类
scf_name	string	scf的名称
nsz	int	系统中cgto的数量
nocc	int	分子轨道占据数
threshold	double	判断scf是否收敛的阈值
do_loop	bool	指示自洽场是否继续或跳出
eigen_S	MatrixXd	重叠积分矩阵S
eigen_H	MatrixXd	H积分矩阵
eigen_G	MatrixXd	G积分矩阵
eigen_J	MatrixXd	J积分矩阵
eigen_K	MatrixXd	K积分矩阵
eigen_ERI	Tensor4D	ERI矩阵
eigen_F	MatrixXd	Fock矩阵
eigen_C	MatrixXd	系数矩阵
eigen_P	MatrixXd	密度矩阵
eigen_X	MatrixXd	变换矩阵
eigen_Y	MatrixXd	逆变换矩阵
eigen_Fp	MatrixXd	正则Fock矩阵F'
eigen_Cp	MatrixXd	正则系数矩阵C'
eigen_E	VectorXd	Fock'矩阵的本征值
E	double	体系总能量

名称	数据类型	含义
E_old	double	上一次迭代的体系总能量
eigen_P_old	MatrixXd	上一次迭代的体系密度矩阵
list	int*	指示占据分子轨道的索引表

MatrixXd和Tensor4D类型的变量是一系列的矩阵，进行scf过程时会计算得到它们。下面很多函数不直接接受参数，他们都使用类的**this**指针。

方法

名称	接受参数	返回类型	含义
set_Threshold	double myeps	int	设置scf阈值
set_Space	Task& T	int	建立scf所需要的存储数据的空间，包括一系列的矩阵

3.分子积分的处理

最初使用的是栾助教所给的积分函数。后来换成我们自己写的通用的积分，这里详细说明其计算方法。

PGTO的表达和积分的计算

1. 归一化系数的计算 对于一个一般的PGTO，我们使用记号 $|aAlmn\rangle = x^l y^m z^n \exp(-a(\vec{r} - \vec{A})^2)$ 来表达。
则首先，考虑一个一般的PGTO的归一化系数N:

$$\frac{1}{N^2} = \int x^{2l} y^{2m} z^{2n} \exp(-2a(\vec{r} - \vec{A})^2)$$

利用公式：

$$\int_{-\infty}^{\infty} x^{2n} e^{-ax^2} = \sqrt{\frac{\pi}{a}} \frac{(2n-1)!!}{(2a)^n}$$

所以：

$$\frac{1}{N^2} = \left(\frac{\pi}{2a}\right)^{3/2} \frac{(2l-1)!!(2m-1)!!(2n-1)!!}{(4a)^{l+m+n}}$$

2. GTO乘积定理 对于s-s型的乘积：

$$\exp(-ar_A^2)\exp(-br_B^2) = K\exp(-(a+b)r_P^2)$$

其中 $K = \exp(-\frac{ab}{a+b}AB^2)$ ，代表高斯相乘后的最大峰高。而点 r_P 就是 r_A 和 r_B 两点对于他们指数的加权平均的结果。 广义的乘积定理则可以表述为：

$$(x^l y^m z^n aA)(x^{l'} y^{m'} z^{n'} bB) = [x + PA_x]^l [x + PB_x]^{l'} [y + PA_y]^m [y + PB_y]^{m'} [z + PA_z]^n [z + PB_z]^{n'} K \exp(-\frac{ab}{a+b}r_P^2)$$

3. 重叠积分的计算 对于一般的重叠积分可以根据GTO乘积定理和高斯积分公式进行计算，得到如下结果：

$$\langle aAlmn | bBl'm'n' \rangle = KI_x I_y I_z$$

其中以x方向为例：

$$I_x = \sqrt{\frac{\pi}{a+b}} \sum_{i=0}^{(l+l')} \left[\frac{(l+l')}{2} \right]_{2i} [l' (PA)_x (PB)_x] \frac{(2i-1)!!}{[2(a+b)]^i}$$

其中函数是两个二项式展开的乘积：

$$f_i[l'l'(PA)_x(PB)_x] = \sum_{\lambda}^l \sum_{\mu}^{l'} \delta_{\lambda+\mu,i} (l\lambda) (l'\mu) (PA)_x^{l-\lambda} (PB)_x^{l'-\mu}$$

4. 动能积分的计算 动能积分涉及到高斯函数的求导，高斯函数的求导还是高斯类型的函数：

$$\frac{\partial}{\partial x_A} \mid aAlmn\rangle = (lx_A^{-1} - 2ax_A) \mid aAlmn\rangle$$

$$\frac{\partial^2}{\partial x_A^2} \mid aAlmn\rangle = (l(l-1)x_A^{-2} - 2a(2l+1) + 4a^2x_A^2) \mid aAlmn\rangle$$

所以不难转换为重叠积分的计算：

$$I = \langle aAl_1m_1n_1 \mid -\frac{1}{2}\nabla^2 \mid bBl_2m_2n_2\rangle = I_x + I_y + I_z$$

其中以x方向为例：

$$\begin{aligned} I_x &= \langle aAl_1m_1n_1 \mid -\frac{1}{2}\frac{\partial^2}{\partial x^2} \mid bBl_2m_2n_2\rangle \\ &= -\frac{1}{2}l_2(l_2-1)\langle aAl_1m_1n_1 \mid bB(l_2-2)m_2n_2\rangle + b(2l_2+1)\langle aAl_1m_1n_1 \mid bBl_2m_2n_2\rangle - 2b^2\langle aAl_1m_1n_1 \mid bB(l_2+2)m_2n_2\rangle \end{aligned}$$

5. 电子-核吸引能积分

使用不完全Γ函数 $F_m(\omega) = \int_0^1 \exp(-\omega t^2)t^{2m}dt$

F_m 的Taylor展开:

$$F_m(\omega) = e^{-\omega} \sum_{i=0}^{infy} \frac{(2m-1)!!(2\omega)^i}{(2m+2i+1)!!}$$

也可以使用递推公式：

$$F_{m+1}(\omega) = \frac{1}{\omega} [(2m+1)F_m(\omega) - \exp(-\omega)]$$

以及求导性质：

$$F'_m(\omega) = -F_{m+1}(\omega)$$

首先看 s-s 型的V积分，计算器结果就需要用到laplace变换：

$$\begin{aligned}\langle ar_A \mid \frac{1}{r_C} \mid br_B \rangle &= K \int \exp(-(a+b)r_P^2) \frac{1}{r_C} d\tau \\ &\stackrel{laplace}{=} K \int \exp(-(a+b)r_P^2) \left[\pi^{-1/2} \int_0^\infty \exp(-sr_C^2) s^{-1/2} ds \right] d\tau \\ &= K \pi^{-1/2} \iint_0^\infty \exp(-(a+b)r_P^2) \exp(-sr_C^2) s^{-1/2} ds d\tau \\ &= K_{AB} \pi \int_0^\infty \exp\left[-\frac{(a+b)s}{a+b+s} PC^2\right] s^{-1/2} (a+b+s)^{-3/2} ds\end{aligned}$$

变量代换 $t^2 = s/(a+b+s)$ ，可以讲对s的积分化成不完全Γ函数 F_0 :

$$\langle ar_A \mid \frac{1}{r_C} \mid br_B \rangle = K \frac{2\pi}{a+b} F_0[(a+b)PC^2]$$

对于广义的电子和核吸引势能积分，可以讲高阶的Gauss函数通过对核位置的偏导数降阶，如：

$$\begin{aligned}G(a, A, l+1, m, n) &= \frac{1}{2a} \left[\frac{\partial}{\partial A_x} G(a, A, l, m, n) + l G(a, A, l-1, m, n) \right] \\ \hat{1} &= \frac{1}{2a} \left(\frac{\partial}{\partial A_x} \hat{a}_l + \hat{a}_l \tilde{l} \hat{a}_l \right) \\ \hat{1}_l &= \frac{1}{(2a)^l} \left(\frac{\partial}{\partial A_x} \hat{a}_l + \tilde{l} \hat{a}_l^2 \right)^l = \frac{1}{(2a)^l} \sum_{i=0}^l C_l^i \left(\frac{\partial}{\partial A_x} \right)^i \hat{a}_l^i \tilde{l}^{l-i} \hat{a}_l^{2l-2i} = \frac{1}{(2a)^l} \left(\frac{\partial}{\partial A_x} \right)^l\end{aligned}$$

其中 \hat{a}_l 为x方向角动量的湮灭算符，并却对与l的指标： $if k > l, a_l^k = 0$ ，其中 $\tilde{l}^n \mid l \rangle \stackrel{def}{=} \frac{\Gamma(l)}{\Gamma(l-n)} \mid l \rangle$

那么对于任意的一个V积分：

$$\begin{aligned}
& \langle ar_A^{lmn} | \frac{1}{r_C} | br_B^{l'm'n'} \rangle \\
&= \frac{1}{(2a)^{l+m+n}} \frac{1}{(2b)^{l'+m'+n'}} (\frac{\partial}{\partial A_x})^l (\frac{\partial}{\partial A_y})^m (\frac{\partial}{\partial A_z})^n (\frac{\partial}{\partial B_x})^{l'} (\frac{\partial}{\partial B_y})^{m'} (\frac{\partial}{\partial B_z})^{n'} \langle ar_A^{lmn} | \frac{1}{r_C} | br_B^{l'm'n'} \rangle \\
&= \frac{1}{(2a)^{l+m+n}} \frac{1}{(2b)^{l'+m'+n'}} (\frac{\partial}{\partial A_x})^l (\frac{\partial}{\partial A_y})^m (\frac{\partial}{\partial A_z})^n (\frac{\partial}{\partial B_x})^{l'} (\frac{\partial}{\partial B_y})^{m'} (\frac{\partial}{\partial B_z})^{n'} \left[K \frac{2\pi}{a+b} F_0[(a+b)PC] \right]
\end{aligned}$$

可以展开为6个二项式展开的乘积，剩下的就是我们需要知道K关于核位置的任意阶导数，以及 F_0 关于核位置的任意阶导数。

$$\begin{aligned}
& (\frac{\partial}{\partial A_x})^l (\frac{\partial}{\partial A_y})^m (\frac{\partial}{\partial A_z})^n (\frac{\partial}{\partial B_x})^{l'} (\frac{\partial}{\partial B_y})^{m'} (\frac{\partial}{\partial B_z})^{n'} K \\
&= (-1)^{l'+m'+n'} (\frac{\partial}{\partial A_x})^{l+l'} (\frac{\partial}{\partial A_y})^{m+m'} (\frac{\partial}{\partial A_z})^{n+n'} \exp[-\frac{ab}{a+b} AB^2]
\end{aligned}$$

对比Hermite多项式的定义：

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2} = n! \sum_{m=0}^{[n/2]} \frac{(-1)^m}{m!(n-2m)! 2^m} x^{n-2m}$$

$$Above = (-1)^{l+m+n} H_{l+l'} \left(\frac{ab}{a+b} \right)^{(l+m+n+l'+m'+n')/2} \left(\sqrt{\frac{ab}{a+b}} AB_x \right)^{H_{m+m'}} \left(\sqrt{\frac{ab}{a+b}} AB_y \right)^{H_{n+n'}} \left(\sqrt{\frac{ab}{a+b}} AB_z \right)^{F_0} \exp[-\frac{ab}{a+b} AB^2]$$

另外一方面，可以发现， $F'_m(\omega) = -F_{m+1}(\omega)$ ，忽略m指标的变化，求导规则与 e^{-x} 非常相似，而 $\omega = \alpha_{eff} PC$ ，这样也同样的道理，可以通过Herimite多项式处理：

$$\begin{aligned}
& (\frac{\partial}{\partial A_x})^l (\frac{\partial}{\partial A_y})^m (\frac{\partial}{\partial A_z})^n (\frac{\partial}{\partial B_x})^{l'} (\frac{\partial}{\partial B_y})^{m'} (\frac{\partial}{\partial B_z})^{n'} F_0[(a+b)PC] \\
&= \left(\frac{b}{a+b} \right)^{l+m+n} \left(\frac{a}{a+b} \right)^{l'+m'+n'} (\frac{\partial}{\partial P_x})^{l+l'} (\frac{\partial}{\partial P_y})^{m+m'} (\frac{\partial}{\partial P_z})^{n+n'} F_0[(a+b)PC] \\
&= (-1)^{l+m+n+l'+m'+n'} \left(\frac{b}{a+b} \right)^{l+m+n} \left(\frac{a}{a+b} \right)^{l'+m'+n'} (a+b)^{(l+m+n+l'+m'+n')/2} \\
&\quad \times H_{l+l'}(\sqrt{a+b} PC_x) H_{m+m'}(\sqrt{a+b} PC_y) H_{n+n'}(\sqrt{a+b} PC_z) F_{l+m+n+l'+m'+n'}[(a+b)PC]
\end{aligned}$$

6. 电子排斥能积分的计算

os型积分类似的需要通过laplace变换，这里不假解释了，结果为：

$$I = \frac{2\pi^{5/2}}{(a+b)(c+d)\sqrt{a+b+c+d}} K_{AB} K_{CD} F_0 \big[\frac{(a+b)(c+d)}{(a+b+c+d)} P Q \big]$$

$$K_{AB} = \exp(-\frac{ab}{a+b} AB)$$

$$K_{CD} = \exp(-\frac{cd}{c+d} CD)$$

积分的函数说明 (basic_Integral.h)

函数名称	接受参数	返回类型	计算含义
integral_S_sstype	Orbital_cgto& cgto1, Orbital_cgto& cgto2	double	s-s类型cgto的S积分 (不完整版, 弃用)
integral_ERI_sstype	Orbital_cgto& cgto1, Orbital_cgto& cgto2, Orbital_cgto& cgto3, Orbital_cgto& cgto4	double	s-s类型cgto的ERI积分 (不完整版, 弃用)
integral_T_sstype	Orbital_cgto& cgto1, Orbital_cgto& cgto2	double	s-s类型cgto的T积分 (不完整版, 弃用)
integral_V_sstype	Orbital_cgto& cgto1, Orbital_cgto& cgto2, Atom& P	double	s-s类型cgto的V积分 (不完整版, 弃用)
binomial	int a, int b	int	返回二项式系数 C_a^b
factorial	int a	int	返回阶乘 $a!$
doublefactorial	int a	int	返回双阶乘 $a!!$
Hermite_normial	double x, int n	double	返回Hermite多项式 $H_n(x)$
Kcoeff	Orbital& orb1, Orbital& orb2	double	返回 $\exp[-(ab/(a+b))AB]^{-2}$
general_Kcoeff	Orbital& orb1, Orbital& orb2, int d1, int d2, int d3, int d4, int d5, int d6	double	返回 K_{AB} 在6个核自由度上的任意高阶导数
general_Kcoeff_ERI	Orbital& orb1, Orbital& orb2, int d1, int d2, int d3, int d4, int d5, int d6, int d7, int d8, int d9, int d10, int d11, int d12	double	返回两个K函数乘积 $K_{AB}K_{CD}$ 在12个核自由度上的任意高阶导数
Fn	double w, int m	double	返回m阶的不完全Gamma函数 $F_m(w)$
Fn_pade	double w, int m	double	不完全Gamma函数 $F_m(w)$ 的一种pade近似方案

函数名称	接受参数	返回类型	计算含义
general_Fn_V	Orbital& orb1, Orbital& orb2, Atom& PN, int d1, int d2, int d3, int d4, int d5, int d6	double	返回Fn函数对于6个核自由度上的高阶导数（用于V积分）
general_Fn_ERI	Orbital& orb1, Orbital& orb2, Orbital& orb3, Orbital& orb4, int d1, int d2, int d3, int d4, int d5, int d6, int d7, int d8, int d9, int d10, int d11, int d12	double	返回Fn函数对于12个核自由度上的高阶导数（用于ERI积分）
IntecGTO_S	Orbital_cgto& cgto1, Orbital_cgto& cgto2	double	广义类型的cgto的S积分
InteGTO_S	Orbital& orb1, Orbital& orb2	double	广义类型的gto的S积分
IScoeff	Orbital& orb1, Orbital& orb2, char flag	double	广义类型的gto的S积分一维分量
IntecGTO_T	Orbital_cgto& cgto1, Orbital_cgto& cgto2	double	广义类型的cgto的T积分
InteGTO_T	Orbital& orb1, Orbital& orb2	double	广义类型的gto的T积分
IntecGTO_V	Orbital_cgto& cgto1, Orbital_cgto& cgto2, Atom& PN	double	广义类型cgto的V积分
general_InteGTO_V	Orbital& orb1, Orbital& orb2, Atom& PN, int f1, int f2, int f3, int f4, int f5, int f6, int d1 int d2 int d3 int d4 int d5 int d6	double	广义类型gto的V积分(递推法)

4.自洽场计算的流程

函数说明

参见2中的SCFer类。

流程

1. **main** 函数从命令行接收参数，传给解释器 **basic_Parser.h** (也即构建了一个**Task**对象，并初始化Task).
2. 解释器读入输入文件，构造体系、原子、收缩GTO、GTO的信息，并且通过设定的基组，从../basis文件目录读入需要的CGTO参数.
3. 构造一个**SCFer**对象，由解释器Parser将Task对象传给SCFer. SCFer根据体系的参数确定创建矩阵的大小.
4. SCFer进入loop_SCF程序，首先是调用一系列积分函数(./basic_Integral.h)，计算S、H、ERI矩阵等. 并对重叠矩阵对称正交化得到变换矩阵X和其逆矩阵Y.
5. 通过讲H假定成Fock矩阵，预先做一轮SCF，可以得到初始的猜测的密度矩阵P.
6. 通过密度矩阵P、H矩阵、G矩阵计算Fock矩阵F，并使用变换矩阵X变换到F'.
7. 解F'矩阵的本征值问题，得到解本征值e和本征矢空间C'，并通过变化矩阵X变换到分子轨道系数矩阵C.
8. 计算确定分子轨道占据，进一步确定总能量E、密度矩阵P.
9. 比较得到的总能量E和密度矩阵P是否与上一步的一致，如果一致则跳出自洽场迭代，否则重复自洽场迭代.

5. 对称性的应用(暂无)

6.测试与结果

进行测试如下(测试命令参看编译与使用)

测试体系	结果 [au]	Gauss09对比[au]	结论
H	-0.42244193	-0.4982329	不能解决open-shell问题
He	-2.85516043	-2.8551604	一 致
H2	-1.11003090	-1.1100309	一致
HeH+	-2.89478689	-2.8947869	一致
H4	-1.80246920	-1.7251712	不一致
CH4	震荡问题	-39.9119255	不一致

