概要

NTChem2013 を利用するためには，NTChem の実行ファイルが導入されている計算機センターのユーザーとして利用するか，開発代表者 (nakajima@riken.jp) に連絡して，利用者の計算機環境に NTChem のコンパイル済み実行ファイルを導入して用いるかのいずれかの方法がある．利用者の環境において利用したい場合は，開発者代表に連絡を取り相談するとよい．2015 年 7 月現在，NTChem が導入されている計算機センターは自然科学研究機構 岡崎共通研究施設 計算科学研究センター (RCCS) と公益財団法人 計算科学振興財団 (FOCUS) スパコンシステムだけであるが，順次導入していく計画である．

NTChem の詳細について

NTChem に関連する情報については，Web サイト
https://molsc.riken.jp/index.html
を参照するとよい．利用法の詳細や新しい情報を得るためには開発者と連絡をとるとよい．また，ユーザーは NTChem ユーザーメーリングリスト
ntchem@googlegroups.com
を登録の上，利用することが可能である．登録は開発代表者に連絡すること．

NTChem の文献と研究成果発表時の引用義務

NTChem を用いて得た成果を公表するときは Web サイト
https://molsc.riken.jp/ntchem_e.html
および、レビュー

T. Nakajima, M. Katouda, M. Kamiya, and Y. Nakatsuka, Int. J. Quantum Chem. **115**, 349–359 (2015).

を引用してほしい．また，下記の機能を利用した場合は以下の論文を引用してほしい．

Douglas–Kroll

T. Nakajima and K. Hirao, J. Chem. Phys. **113**, 7786–7789 (2000).

T. Nakajima and K. Hirao, Chem. Rev. **112**, 385–402 (2012).

RESC

T. Nakajima and K. Hirao, Chem. Phys. Lett. **302**, 383–391 (1999).

RI-MP2

M. Katouda and T. Nakajima, J. Chem. Theory Comput. **9**, 5373–5380 (2013).

M. Katouda, A. Naruse, Y. Hirano and T. Nakajima, J. Comput. Chem. **37**, 2623–2633 (2017).

M. Katouda and T. Nakajima, J. Comput. Chem. **38**, 489‐507 (2017).

Quantum Monte Carlo

Y. Nakatsuka, T. Nakajima, M. Nakata, and K. Hirao, J. Chem. Phys. **132**, 054102 (7 pages) (2010).

Y. Nakatsuka, T. Nakajima, and K. Hirao, J. Chem. Phys. **132**, 174108 (8 pages) (2010).

T. Nakajima and Y. Nakatsuka, in Practical Aspects of Computational Chemistry I: An Overview of the Last Two Decades and Current Trends, edited by J. Leszczynski, M. K. Shukla, H. de Rode (Springer), 293–317 (2012).

GFC

Y. Kurashige, T. Nakajima, and K. Hirao, J. Chem. Phys. **126**, 144106 (4 pages) (2007).

M. A. Watson, Y. Kurashige, T. Nakajima, and K. Hirao, J. Chem. Phys. **128**, 054105 (7 pages) (2008).

Y. Kurashige, T. Nakajima, T. Sato, and K. Hirao, J. Chem. Phys. **132**, 244107 (7 pages) (2010).

DL-FIND

J. Kästner, J. M. Carr, T. W. Keal, W. Thiel, A. Wander, and P. Sherwood, J. Phys. Chem. A, **113**, 11856–11865 (2009).

DFT-D3

S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, J. Chem. Phys. **132**, 154104 (19 pages) (2010).

S. Grimme, S. Ehrlich, and L. Goerigk, J. Comput. Chem. **32**, 1456–1465 (2011).

NAMELIST &Control

&Control は計算全体をコントロールするために必要なネームリストです．これには以下のような要素が含まれます.

Name	[CHARACTER]
NCorePerIO	[INTEGER]
Symm	[CHARACTER]

Name [CHARACTER] (default = ’ntchem’)

Name はプログラム中で使用される中間ファイル名のベースとして用いられます. 例えば，Name = ’Foo’と指定したインプットを用いる場合，基底関数の情報は”Foo.Basis”，構造の情報は”Foo.Geom”というファイルから読み込まれます. NTChem では共通の Name を含む複数のインプットを利用することで，プログラム間の連携をとります．例として以下のように RHF 計算を行った後，その分子軌道を用いて MP2 計算を行う場合のインプットを示します.

ファイル：h2o_rhf.inp

```
&Control Name=’h2o_test’ /

&scf SCFType=’RHF’, …, /

(構造・基底の情報など)
```

ファイル：h2o_mp2.inp

```
&Control Name=’h2o_test’ /

(MP2 計算に必要な情報)
```

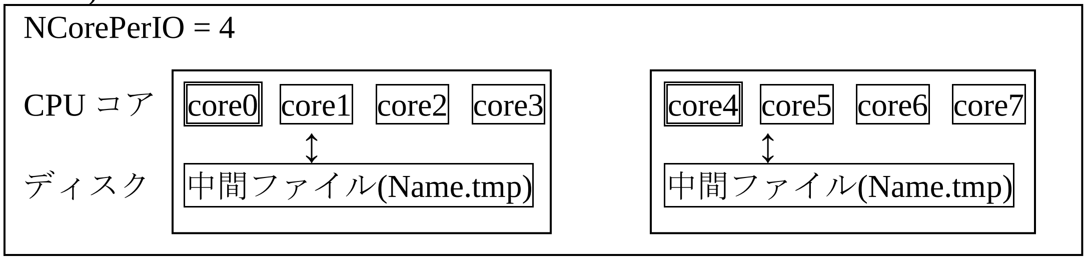
最初に”h2o_rhf.inp”を用いて scf を実行することで，RHF 分子軌道が”h2o_test.MO”というファイルに保存されます．インプットファイルの名前である”h2o_rhf”ではなく，Name に指定された”h2o_test”が利用されることに注意してください．続いて”h2o_mp2.inp”を用いて mp2 を実行することで，“h2o_test.MO”ファイルなどの情報を利用して，MP2 計算が行われます.

NCorePerIO [INTEGER]　(default = 1)

THIS ITEM IS OBSOLESCENT.

NCorePerIOは，並列計算時のファイルIOを何個のCPUコアをまとめたグループで行うかを指定します．例えばNCorePerIO = 4を指定した場合，8コアでの並列計算であれば下図のようにcore0，core4の2つのCPUコアがインプットファイル・中間ファイルにアクセスし，他のCPUコアから情報を集約してファイルに書き込む，または，情報を読み取って他のCPUコアに伝達します．NCorePerIOの値は計算機環境と並列方式によって制限されます．

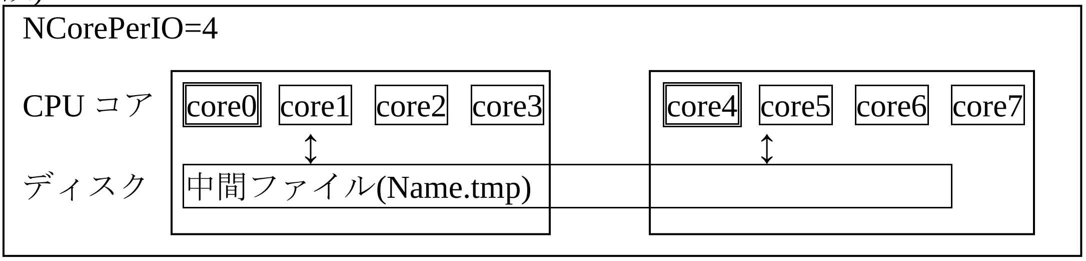
図：中間ファイルにアクセスするCPUコア(二重線のコアのみ中間ファイルにアクセス)



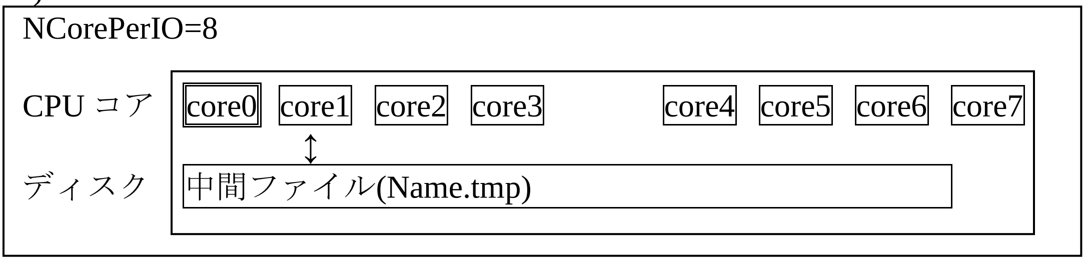
NCorePerIOの設定では，各計算機でのワークディレクトリを共有しているコア数に注意する必要があります．ワークディレクトリを共有するコア数が，(1) 全コア，(2) 2コア以上，(3) 1コア，のそれぞれについて説明します．以下ではMPI並列の場合を解説しますが，MPI/OpenMPハイブリッド並列の場合，$OMP_NUM_THREADS環境変数で指定された数で実際のCPUコア数を割っただけのCPUコアがあると考えて下さい．(例：京で$OMP_NUM_THREADS=8，16ノード，128CPUコアの計算を行う場合，仮想的に1ノードあたり1CPUコア，計16コアとなります)

(1) 全ノードでワークディレクトリが共有されている場合
NCorePerIOがノード数より小さいと，core0とcore4が同名のファイルに書き込みを行うために正常に動作しません．この場合はNCorePerIOを全CPUコア数と同じ値にしてください．

図：全ノードがワークディレクトリを共有している場合
(誤)



(正)



(2) N (> 1)個のCPUコアがワークディレクトリを共有する場合
NCorePerIO < Nでは(1)と同様の問題が起こります．NCorePerIOはNの倍数に設定してください．

(3) 各CPUコアが各自のワークディレクトリを持つ場合
京でランクディレクトリを利用する場合などが，このパターンに該当します．NCorePerIOは任意の値を利用できます．ただしNCorePerIOが小さい場合，IO時間が増加するのに加え，IO担当コア(上記例でのcore0，core4)間での通信が大規模になり，通信のオーバーヘッドが増加します．逆にNCorePerIOが大きい場合，ファイルの読み取り時の通信が増加します．

Symm [CHARACTER] (default = ‘auto’)

Symm は分子の対称性の取り扱い方を指定します．分子系の対称性が記述された SymmLog ファイルを利用する場合にのみ，このオプションは意味を持ちます．ワークディレクトリに SymmLog ファイルが存在しない場合，分子系は C_1 対称性を持つものとして扱われます．対称性を利用する場合，プログラム symmetry_ntqc を利用して SymmLog ファイルを生成し，ワークディレクトリにコピーした上でこのオプションを指定してください．オプションの要素としては，解析された分子系の対称性群 (SymmLog ファイルに記述されています) の部分群のうち，アーベル群を指定することが可能です (C1，Cs，Ci，C2，C2v，C2h，D2，D2h)．初期値’auto’では最も対称性の高い群を利用します．SymmLog ファイルは，Name の値に関わらず常に同一名称 (“SymmLog”) なので，複数の計算でワークディレクトリを共有している場合，他の計算の SymmLog ファイルから間違った情報を読み取ってしまう場合があります．この場合，ワークディレクトリを分けるか，Symm = ’C1’を明示的に指定してください．

計算対象系の指定

計算の対象となる分子の構造，各原子に張られる基底関数及び有効内殻ポテンシャル (ECP) は，タイトルと文字列”End”で囲まれた領域 (カード) で指定します.

Geom カード	: 分子の構造を指定するカード
Basis カード	: 基底関数を指定するカード
Basis_***カード	: 補助基底関数を指定するカード (Basis カードの説明参照)
ECP カード	: 有効内殻ポテンシャル (ECP) を指定するカード

これらのカードは全て原子名の指定を含み，Geom カードで指定された座標にある各原子に対して，対応した原子名の基底関数, ECP が割り振られます．原子名の指定では大文字・小文字は区別されることに注意してください.

例：チタン原子に対する構造・基底・ECP の原子名指定

	Geom カード 中での指定	Basis カード 中での指定	ECP カード 中での指定	
1.	Ti	Ti	Ti	(正)
2.	ti	ti	ti	(正)
3.	Ti	Ti	TI	(誤)
4.	Ti	ti	ti	(誤)

1 の例では構造，基底, ECP の各カード内での原子名指定が”Ti”と揃っているので，正しく情報が割り振られます.
2 の例では，全て小文字で“ti”と揃っているので正しく情報が割り振られます.
3 の例では，ECP カード中でのみ“TI”と全て大文字で指定されているため，構造カード中で”Ti”と指定された原子には基底関数は設定されますが，ECP は設定されません.
4 の例では，構造カード中では原子名を”Ti”と指定しているのに対して，基底・ECP カード内では”ti”と指定しているため基底関数・ECP 共に設定されません.

(注) NTChem のカード内では, 区切り文字としてタブの使用が出来ません. スペースによる区切りを利用してください.

分子構造の指定 (Geom カード)

Geom カードは分子の座標を指定します.

書式

```
Geom
CAtom1          CentrX1         CentrY1         CentrZ1
CAtom2          CentrX2         CentrY2         CentrZ2
…
End
```

CAtom [CHARACTER]　　原子の元素記号 (ダミー原子は‘X’もしくは’x’で指定)

CentrX, CentrY, CentrZ [REAL]　　原子の x，y，z 座標 (数値の単位は&BasInp で指定)

原子基底関数の指定 (Basis カード)

Basis カードは原子基底関数の指定に用います.

書式

```
Basis
CAtom1
        CAngl1        NSgmt1
        Expnt1        CCoef1
        Expnt2        CCoef2
        …
        CAngl2        NSgmt2
        Expnt1        CCoef1
        …
****
CAtom2
        …
****
        …
****
End
```

CAtom [CHARACTER]	原子の元素記号
CAngl [CHARACTER]	縮約関数の軌道角運動量 (S, P, D, F, ...)
NSgmt [INTEGER]	縮約関数を展開するプリミティブ Gauss 関数の数
Expnt [REAL]	プリミティブ Gauss 関数の指数
CCoef [REAL]	プリミティブ Gauss 関数の縮約係数
“****”	当該元素の入力を完了

注意

NTChem は SP シェルには現在対応していません．SP シェルを含む基底関数系 (Pople 基底系等) を用いる際には，S シェルと P シェルに分けて使ってください.

基底関数の並び順について

Cartesian 型: アルファベット順，例えば d 関数では，xx, xy, xz, yy, yz, zz.

Spherical 型: 昇降順，例えば d 関数では，-2, -1, 0, +1, +2.

各種補助基底の指定

“Basis”カードに加えて，計算条件に応じて以下のような補助基底情報を入力することができます.

“Basis_GFC”: GFC 用の補助基底
”Basis_GFCGrad”: GFC gradient 用の補助基底
“Basis_ProjMO”: 射影 MO 用の基底
“Basis_MPCore”: Model potential の内殻の基底
“Basis_ProjMP”: Model potential 用の補助基底
“Basis_ProjQR”: QRel 用の補助基底 (通常は基底関数の短縮を外したもの)
“Basis_RIJ”あるいは“Basis_RISCF”: RIDFT の Coulomb 用の補助基底
“Basis_RIC”: RIMP2 の補助基底
“Basis_PSFit”: PS 法における Fitting Basis 用の補助基底

ECP の指定

ECP カードで ECP の情報を指定できます.

書式

```
ECP
CAtom1
        LMax            ZCore
        Title1
        NSgmt1
        NGauss1                 Expnt1          CCoef1
        NGauss2                 Expnt2          CCoef2
        …
        Title2
        NSgmt2
        NGauss1                 Expnt1          CCoef1
        …
        ****
CAtom2
        …
        ****
End
```

CAtom [CHARACTER]	原子の元素記号
LMax [INTEGER]	最低軌道角運動量
ZCore [INTEGER]	ECP で置き換える電子の数
Title [CHARACTER]	項のタイトル (任意). “****”なら当該元素の入力を完了
NSgmt [INTEGER]	当該項を展開するプリミティブ Gauss 関数の数
NGauss [INTEGER]	プリミティブ Gauss 関数の軌道角運動量
Expnt [REAL]	プリミティブ Gauss 関数の指数
CCoef [REAL]	プリミティブ Gauss 関数の展開係数

NAMELIST &BasInp

&BasInp は分子構造や基底関数の読み取りに関するネームリストです．

Units	[CHARACTER]
GTOType	[CHARACTER]
NormP	[LOGICAL]
NormF	[LOGICAL]
GTOType_GFC	[CHARACTER]
NormP_GFC	[LOGICAL]
NormF_GFC	[LOGICAL]
IPrint	[INTEGER]

Units [CHARACTER] (default = ’AU’)

Units = ‘AU’もしくは‘Bohr’を指定すると，Geom カードで指定された原子座標を原子単位で読み取ります．Units = ‘Ang’を指定すると，オングストローム単位で読み取ります．

GTOType [CHARACTER] (default = ’Spherical’)

GTOType = ‘Spherical’もしくは‘Cartesian’で，純粋な関数もしくはカーテシアン関数を用います．Basis_GFC と Basis_GFCGrad 以外の全ての基底関数系に適用されます．

NormP [LOGICAL] (default = .TRUE.)

NormP = .TRUE.が指定された場合，プリミティブ Gauss 関数を規格化します．Basis_GFC と Basis_GFCGrad 以外の全ての基底関数系に適用されます．

NormF [LOGICAL] (default = ’TRUE’)

NormF = .TRUE.が指定された場合，縮約基底関数を規格化します．Basis_GFC と Basis_GFCGrad 以外の全ての基底関数系に適用されます．

GTOType_GFC [CHARACTER] (default = ’Spherical’)

GTOType と同様ですが，Basis_GFC と Basis_GFCGrad で指定された基底関数系に適用されます．

NormP_GFC [LOGICAL] (default = .TRUE.)

NormP と同様ですが，Basis_GFC と Basis_GFCGrad で指定された基底関数系に適用されます．

NormF_GFC [LOGICAL] (default = ’TRUE’)

NormF と同様ですが，Basis_GFC と Basis_GFCGrad で指定された基底関数系に適用されます．

IPrint [INTEGER]　(default = 0)

プリントオプションです．IPrint = 0 はデフォルト出力．IPrint = 1 で原子座標が，IPrint = 2 以上で原子座標および原子間距離が追加出力されます．

Module BasInp

Function

This module controls the inputs for molecular specification.

Required NAMELIST

&Control
&BasInp

NAMELIST &BasInp

Parameters

Unit select unit for atom coordinates in input file (default = ’AU’)
‘AU’ … use atomic unit
‘Ang’ … use angstrom
‘Bohr’ … use Bohr (= ‘AU’)

GTOType flag of Gaussian-type orbital (default = ’Spherical’)
‘Spherical’… use Spherical Gaussian-type orbitals
‘Cartesian’… use Cartesian Gaussian-type orbitals

NormP flag to normalization for primitive Gaussian (default = ’T’)
T … do normalization
F … do not normalization

NormF flag to normalization for basis function (default = ’T’)
T … do normalization
F … do not normalization

GTOType_GFC flag of Gaussian-type orbital for GFC calculation (default = ’Spherical’)
‘Spherical’… use spherical Gaussian-type orbitals
‘Cartesian’… use Cartesian Gaussian-type orbitals

NormP_GFC flag to normalization for primitive Gaussian in GFC calculation (default = ’T’)
T … do normalization
F … do not normalization

NormF_GFC flag to normalization for basis function in GFC calculation (default = ’T’)
T … do normalization
F … do not normalization

IPrint print option (default = 0)

Module MDInt1

Function

This module controls the one-electron integration calculations such as overlap, kinetic, nuclear attractive interaction, and dipole moment.

Required NAMELIST

&Control
&MDInt1

NAMELIST &MDInt1

Parameters

CalDip flag for dipole moment integral calculation (default = .TRUE.); the origin of dipole moment integrals is assumed to be a coordinate origin
F … do not calculate dipole moment integrals
T … calculate dipole moment integrals

CalChg flag for Coulomb attraction integrals from point charges (default = .FALSE.)
F … do not calculate point charge integrals
T … calculate point charge integrals

Only1c flag for only one-center integrals for nuclear attraction
T … do use only one-center integration
F … do not use only one-center integration (default)

QRel1c flag for only one-center integrals for relativistic nuclear attraction
T … do use only one-center integration
F … do not use only one-center integration (default)

NDDO Flag for the neglect of diatomic differential overlap (NDDO) method
T … do NDDO calculation
F … do not use only one-center integration (default)

ThrInt threshold value of integration (default = 1.0D-15)

ThrPrim threshold value of integration targeting primitive Gaussian (default = 1.0D-20)

ElcFld strength of electrostatic field (default: ElcFld(1:3) = Zero)

QRelHam flag for one-electron relativistic Hamiltonian calculation (default = ‘NREL’)
NREL …. nonrelativistic
DK1 … use first-order Douglas–Kroll (DK) method
DK2 … use second-order Douglas–Kroll (DK) method
DK3 … use third-order Douglas–Kroll (DK) method
RESC … use relativistic elimination of small components
ZORA … use zeroth-order regular approximation
FPRA … use free-particle regular approximation
IORA … use infinite-order regular approximation

ThrQRel threshold for linear dependency of relativistic Hamiltonian calculation (default = 1.0D-9)

CLight speed of light (atomic unit) (default = 137.0359895D0)

Finite flag to finite nuclear effect (default = .FALSE.)
T … do consider finite nuclear effect
F … do not consider finite nuclear effect (default)

IPrint print option (default = 0)

Module ECP

Function

This module controls the calculation for effective core potential integrals.

Required NAMELIST

&Control
&ECP

NAMELIST &ECP

Parameters

IPrint	print option (default = 0)

Module SOInt1

Function

This module controls the one-electron integration calculations with spin–orbit interaction.

Required NAMELIST

&Control
&SOInt1

NAMELIST &SOInt1

Parameters

SNSO flag for screened-nuclear spin–orbit (SNSO) approximation for two-electron spin–orbit contribution (default = .TRUE.)
T … use SNSO approximation
F … do not use SNSO approximation, that is, use bared one-electron SO integrals

Only1c flag for only one-center integrals for SO (default = .FALSE.)
T … do use only one-center integration
F … do not use only one-center integration (default)

Finite flag to finite nuclear effect (default = .FALSE.)
T … do consider finite nuclear effect
F … do not consider finite nuclear effect (default)

CLight speed of light (atomic unit) (default = 137.0359895D0)

QRelHam flag for spin–orbit calculations (default = ‘NREL’)
NREL … = BP
DK1 … use first-order Douglas–Kroll (DK) method
BP … use Breit–Pauli approximation
ZORA … use zeroth-order regular approximation
IORA … use infinite-order regular approximation

ThrQRel threshold for linear dependency of relativistic Hamiltonian calculation (default = 1.0D-9)

ThrInt threshold value of integration (default =1.0D-15)

ThrPrim threshold value of integration targeting primitive Gaussian (default = 1.0D-20)

IPrint print option (default = 0)

Module Huckel

Function

This module controls the extended Hückel calculation. This module mainly intends to generate the initial MO guess for the succeeding SCF calculation. The current implementation is restricted to the non-iterative and non-relativistic Hückel calculation. Even initial guess obtained this kind of calculation may be available for the succeeding relativistic calculation.

Required NAMELIST

&Control
&Huckel

Prepared input data files

Name.Basis
Name.Geom
Name.HCore
Name.Overlap
Name.NucRepl
Name.Charge (optionally)

Created output data files

NAMELIST &Huckel

Parameters

UHF flag to indicate whether the spin unrestricted extended Hückel method is used
F … do not use the unrestricted extended Hückel (default)
T … use the unrestricted extended Hückel

OrthType orbital orthogonalization option (default = ‘Cholesky’)
‘Cholesky’… use Cholesky decomposition of the overlap matrix to obtain the orthogonalization matrix
‘Canonical’… canonical orthogonalization
‘Symmetric’… symmetrical orthogonalization

ThrOvlp threshold for linear dependency of canonical orthonormal orbitals (default = 1.0D-6); ThrOvlp is available only for OrthType = 'Canonical'

NOccA number of electrons for alpha orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

NOccB number of electrons for beta orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

IPrint print flag (default = 0)
0 … normal printing
1 … debug printing

Module ProjMO

Function

Required NAMELIST

&Control
&ProjMO

Prepared input data files

Created output data files

NAMELIST &ProjMO

Parameters

UHF flag to indicate the density matrix is calculated with spin-unrestricted HF (UHF) or KS-DFT (UKS) method
F … density matrix is calculated with spin-restricted HF (RHF) or KS-DFT (RKS) method
T … density matrix is calculated with spin-unrestricted HF (UHF) or KS-DFT (UKS) method

SOrbit flag to use the density matrix which includes spin–orbit interaction (default = .FALSE.)
F … the density matrix includes no spin–orbit interaction
T … the density matrix includes spin–orbit interaction

OrthMO flag to orthogonalize the molecular orbitals (default = .FALSE.)
F … do not orthogonalize the molecular orbitals
T … orthogonalize the molecular orbitals

OrthType method for orthogonalization of molecular orbitals (default = ‘Cholesky’)
‘Cholesky’… use Cholesky decomposition of the overlap matrix to obtain the orthogonalization matrix
‘Canonical’… canonical orthogonalization
‘Symmetric’… symmetrical orthogonalization

NOccA number of electrons for alpha orbitals (default)

NOccB number of electrons for beta orbitals (default)

ThrOvlp threshold for linear dependency of canonical orthonormal orbitals (default = 1.0D-6); ThrOvlp is available only for OrthType = 'Canonical'

ThrInt threshold value of integration (default = 1.0D-15)

ThrPrim threshold value of integration targeting primitive Gaussian (default = 1.0D-20)

IPrint print flag (default = 0)

Module ProjDens

Function

Required NAMELIST

&Control
&ProjDens

Prepared input data files

Created output data files

NAMELIST &ProjDens

Parameters

UHF flag to indicate the density matrix is calculated with spin-unrestricted HF (UHF) or KS-DFT (UKS) method
F … density matrix is calculated with spin-restricted HF (RHF) or KS-DFT (RKS) method
T … density matrix is calculated with spin-unrestricted HF (UHF) or KS-DFT (UKS) method

SOrbit flag to use density matrix which includes spin–orbit interaction (default = .FALSE.)
F … the density matrix includes no spin–orbit interaction
T … the density matrix includes spin–orbit interaction

ThrInt (default = 1.0D-15)

ThrPrim (default = 1.0D-20)

IPrint print flag (default = 0)

Module SCF

Function

This module controls the calculation of Hartree–Fock (HF) and Kohn–Sham (KS) density functional theory (DFT) energies. Closed shell and spin unrestricted HF and KS-DFT energies can be calculated as well as open shell restricted (pseudo-canonical) energies.

Required NAMELIST

&Control
&SCF
&DFT (optionally if DFT = .TRUE.)
&DFTNum (optionally if DFT = .TRUE.)
&Int2 (optionally if CoulType = ’Analy’ and/or ExchType = ’Analy’)
&RIInt2 (optionally if CoulType = ’RI’)
&FEFInp (optionally if CoulType = ’GFC’)
&FMM (optionally if CoulType = ’GFC’)
&ADPTInp (optionally if CoulType = ’GAP’, ’PSGAP’, ’PSGAP+1C’, ’PSGAP+3C’)
&PSSCF (optionally if CoulType = ’PS’, ’PS+1C’, ’PS+3C’, ’PSGAP’, ’PSGAP+1C’, ’PSGAP+3C’ and/or ExchType = ’PS’, ’PS+1C’)
&PSNum (optionally if CoulType = ’PS’, ’PS+1C’, ’PS+3C’, ’PSGAP’, ’PSGAP+1C’, ’PSGAP+3C’ and/or ExchType = ’PS’, ’PS+1C’)
&COSMO (optionally if SCRFType = ’COSMO’)
&ZRF (optionally if SCRFType = ’ZRF’)

Prepared input data files

Name.Basis
Name.Geom
Name.HCore
Name.Overlap
Name.NucRepl
Name.Charge (optionally)
Name.MO
Name.OrbEne (optionally if FON with smearing)

Created output data files

NAMELIST &SCF

Parameters

SCFType type of SCF wavefunction (default = ‘ ’)
‘ ’ … ‘RHF’ for even electrons and ‘UHF’ for odd electrons
‘RHF’ … restricted HF/DFT
‘UHF’ … unrestricted HF/DFT
‘CUHF’… constrained (pseudo-canonical) HF/DFT
‘ROHF’… restricted open HF/DFT

DFT flag to carry out a DFT / UDFT calculation (default = .FALSE.)
F … Hartree–Fock SCF / UHF calculation
T … DFT / UDFT calculation

SCRFType trigger for continuum solvent model (default = ‘ ’)
‘ ’ … gas-phase calculation
‘ZRF’ … Onsager's reaction field model
‘COSMO’… continuum solvent model COSMO
SCRFType = ‘ZRF’ and ‘COSMO’ require NAMELIST &ZRF and &COSMO, respectively

Direct flag to indicate whether the direct or disk-base SCF is used (default = .TRUE.)
F … use disk-base SCF
T … use direct SCF

DiffDen flag to use the density difference technique to accelerate the SCF convergence (default = .TRUE.)
F … do not use the density difference technique
T … use the density difference technique

CoulType computational type for two-electron Coulomb integrals (default = ‘Analy’)
‘Analy’ … analytical integrals
‘RI’ … resolution of the identity (RI) approximation
‘GFC’ … Gaussian–finite elements Coulomb (GFC) approximation (Serial only)
‘GAP’ … Gaussian and plane-wave (GAP) method
‘PS’ … pseudospectral (PS) method
‘PS+1C’… PS method including one-center-atomic analytic correction
‘PS+3C’… PS method including up to three-center-atomic analytic correction
‘PSGAP’… PS and GAP method
‘PSGAP+1C’… PS and GAP method including one-center-atomic analytic correction
‘PSGAP+3C’… PS and GAP method including up to three-center-atomic analytic correction
‘None’ … no Coulomb calculation

ExchType computational type for HF exchange integrals (default = ‘Analy’)
‘Analy’ … analytical integrals
‘RI’ … resolution of the identity (RI) approximation (NYI)
‘PS’ … pseudospectral method
‘PS+1C’… PS method including one-center-atomic analytic correction
‘None’ … no exchange calculation

Skip1e flag to skip the calculation of one-electron kinetic-energy and potential terms (default = .FALSE.)
F … calculate one-electron terms
T … skip calculation of one-electron terms

Skip2e flag to skip the calculation of two-electron terms (default = .FALSE.)
F … calculate two-electron terms
T … skip calculation of two-electron terms

Guess initial orbital guess option (default = ‘ReadMO’)
‘ReadMO’… orbitals read from Name.MO file
‘ReadOnMO’… orthonormalized MOs read from Name.MO file
‘ReadDens’… density matrix read from Name.Dens file
‘HCore’… bare nucleus Hamiltonian orbitals
‘GWH’ … generalized Wolfsberg–Helmholtz
‘Diagonal’… this is available only for PDMSCF = T

OrthType orbital orthogonalization option (default = ‘Cholesky’)
‘Cholesky’… use Cholesky decomposition of the overlap matrix to obtain the orthogonalization matrix
‘Canonical’… canonical orthogonalization
‘Symmetric’… symmetrical orthogonalization

ThrOvlp threshold for linear dependency of canonical orthonormal orbitals (default = 1.0D-6); ThrOvlp is available only for OrthType = 'Canonical'

RstrctMO flag to select the restriction of orbital interchanges (default = .FALSE.)
F … do not restrict orbital interchanges
T … restrict orbital interchanges during the SCF calculation

AlterMOA(1), AlterMOA(2)
interchange alpha MOs between AlterMOA(1) and AlterMOA(2) (default = 0, 0)
0, 0 … no interchange

AlterMOB(1), AlterMOB(2)
interchange beta MOs between AlterMOB(1) and AlterMOB(2) (default = 0, 0)
0, 0 … no interchange

NOccA number of electrons for alpha orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

NOccB number of electrons for beta orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

ThrDen convergence criterion for the density matrix (default = 1.0D-5)

ThrEne convergence criterion for the total SCF energy (default = 1.0D-6)

MaxIter maximum number of iterations (default = 200)

MaxDIIS maximum number of the DIIS error vectors (default = 6)
MaxDIIS = 0 indicates that no DIIS method is used

DIISType DIIS type (default = ‘C1DIIS’)
‘C1DIIS’… original C1-DIIS
‘C2DIIS’… C2-DIIS of Sellers

OnBasDIIS flag to use orthogonalized atomic basis functions in the DIIS method (default = .FALSE.)
F … do not use orthogonalized atomic basis functions
T … use orthogonalized of atomic basis functions

VShift value of orbital energy shift for virtual orbitals (default = 0.1)

DynShift flag to carry out a dynamic virtual shift (default = .FALSE.)
F … do not use the dynamic shift
T … use the dynamic shift

FacDamp damping factor used in the damping method (default = 0.4)

DynDamp flag to carry out a dynamic damping (default = .FALSE.).
F … do not use the dynamic damping
T … use the dynamic damping of Zerner and Hehenberger

MaxDamp maximum number of iterations in the damping step (default = 0)
N … the damping scheme is used in the first *N* times in the SCF calculation

MixDamp flag to combine the damping scheme with the DIIS method (default = .FALSE.)
F … do not combine the damping method with DIIS
T … combine the damping method with DIIS
MixDamp may be available to adopt the approach similar to the dynamical mixing method of Anderson

DEMSCF flag to select the direct energy minimization (default = .FALSE.)
F … do not use the direct energy minimization
T … use the direct energy minimization

FinDiag flag to diagonalize the Fock / KS matrix after the SCF calculation (default = .TRUE.)
F … diagonalize the Fock / KS matrix after the SCF calculation
T … do not diagonalize the Fock / KS matrix after the SCF calculation
FinDiag is available to obtain molecular orbitals and their energies for the diagonalization-free SCF method

FOEne flag to calculate the first-order correction to the SCF energy (default = .FALSE.)
F … do not evaluate the first-order SCF energy
T … evaluate the first-order SCF energy
FOEne is available to perform the dual-level DFT calculation with MaxIter = 0

CoulEne flag to calculate the Coulomb energy individually (default = .FALSE.)
F … do not evaluate the Coulomb energy
T … evaluate the Coulomb energy

FONType broadening type for the fractional occupation number scheme (default = ‘ ’)
‘ ’ … trigger for the conventional fixed occupation number approach
‘Gauss’ … Gaussian broadening
‘Fermi’ … Fermi broadening

WidFON broadening parameter (default = 0.0)

FElecA number of alpha electrons with the fractional number (default = 0.0)
0.0 … FElecA = NOccA
FElecA is prioritized over NOccA if FElecA is explicitly given

FElecB number of beta electrons with the fractional number (default = 0.0)
0.0 … FElecB = NOccB
FElecB is prioritized over NOccB if FElecB is explicitly given

VarFON flag to calculate the energy correction for the non-number conserving change (default = .FALSE.)
F … number conserving change
T … non-number conserving change

IPrint print flag (default = 0)
0 … normal printing
1 … debug printing

Module SOSCF

Function

This module controls the calculation of Hartree–Fock (HF) and Kohn–Sham (KS) density functional theory (DFT) energies. Closed shell and spin unrestricted HF and KS-DFT energies can be calculated as well as open shell restricted (pseudo-canonical) energies.

Required NAMELIST

&Control
&SOSCF
&DFT
&DFTNum
&Int2

NAMELIST &SOSCF

Parameters

Almost the same as &SCF

NAMELIST &Int2

Parameters

IntType specifies the method to evaluate electron repulsion integrals (ERI) (default = ‘Libint’)
‘Libint’ … use Libint library ver. 1 (direct SCF only)
‘Libint2’… use Libint library ver. 2 (direct SCF only)
‘MD4’ … use McMurchie–Davidson method (direct or disk-oriented SCF)
‘MD2’ … use McMurchie–Davidson method (direct SCF)
‘DRK2’ … useDupuis–Rys–King method (direct SCF)

SPType specifies the method to evaluate ERI involving only s and p functions (default = ‘PH’)
‘PH’ … use Pople–Hehre method
‘ACE’ … use ACE (accompanying coordinate expansion) method
‘Smash’ … use Smash library based on Pople–Hehre and McMurchie-Davidson method

PScreen flag to invoke Schwarz integral prescreening in direct SCF (default = T)
F … do not use Schwarz prescreening
T … use Schwarz prescreening

Only1c flag to discard multicenter ERI (default = F)
F … do not discard multicenter ERI
T … compute only one-center ERI

Only2c flag to discard three- and four-center ERI (default = F)
F … do not discard three- and four-center ERI
T … compute only one- and two-center ERI

NDDO flag to invoke NDDO (neglect of diatomic differential overlap) approximation to molecular Hamiltonian (default = F)
F … do not use NDDO
T … use NDDO

ThrPre threshold in Schwarz integral prescreening; this parameter has no effect when PScreen = F (default = 1.0D-12)

ThrInt threshold for ignoring ERI in constructing Fock matrix; the same threshold is applied to preexponent factor (default = 1.0D-15)

ThrPrim the products of primitives with preexponential factor less than ThrPrim are skipped (default = 1.0D-20)

Comments

IntType = ’Libint’ is faster than IntType = ’MD4’. For sp-type integrals, SPType = ’ACE’ is slightly faster than SPType = ’PH’. The latter is always used for range-separated type integrals.

NAMELIST &RIInt2

Parameters

IntType specifies the method to evaluate three-center ERI (default = ”Libint”)
‘Libint’ … use Libint library ver. 1
‘MD4’ … use McMurchie–Davidson method

PScreen flag to invoke Schwarz integral prescreening in direct RI-SCF (default = T)
F … do not use Schwarz prescreening
T … use Schwarz prescreening

ThrPre threshold in Schwarz integral prescreening; this parameter has no effect when PScreen = F (default = 1.0D-12)

ThrInt threshold for ignoring ERI in constructing Fock matrix; the same threshold is applied to preexponent factor (default = 1.0D-15)

ThrPrim products of primitives with preexponential factor less than ThrPrim are skipped (default = 1.0D-20)

NAMELIST &FEFInp

This NAMELIST includes information of finite element functions used in GFC method

Parameters

FEFNthShp order of Shepard interpolation (default = 3)
1 – 5 are available values

FEFICutWF integer to control w.f. cutoff (default = 10)
N … $R_{threshold}$ = SQRT(N * ln10 / exponential)

FEFElmEdg interval of elements in a.u. (default = 1.8D+0)

GFCPreCond method of GFC preconditioning (default = “Chole”)
“Chole” … do Cholesky decomposition preconditioning

GFCMixSolv method of MixSolv (default = “PCR”)

GFCFixSolv (default = ‘PCR’)

GFCMixFix (default = ‘FixGT’)

GFCMixConvIni(default = 1.0D-12)

GFCMixConvFin(default = 1.0D-22)

GFCFixConvIni (default = 1.0D-22)

GFCFixConvFin(default = 1.0D-22)

GFCSOROmgGTF(default = 1.0D+0)

GFCSOROmgFEF(default = 1.0D+0)

GFCSORMaxIt (default = 10)

GFCBCEval method to evaluate boundary condition in GFC method (default = “FMM”)
“FMM” … use fast multipole moment method
“Analy” … use analytical evaluation of electro static potential
“Skip” … skip evaluation

GFCSwitchDen (default = 1.0D-1)

GFCSwitchEne (default = 1.0D-1)

FEFBchEdg (default = 8)

FEFPotCut (default = 1.0D-99)

GFCICutPre (default = 13)

GFCIntThr	(default = 11)
GFCGrdThr	(default = 8)
GFCSORConv	(default = 10)

NAMELIST &FMM

This NAMELIST includes information of fast multipole method used in GFC method

Parameters

LMax		(default = 4)
TLMax		(default = 12)
Algorithm		integer to indicate algorithm (default = 5)
1	…	do_Null
2	…	do_FQ
3	…	do_BQ
4	…	do_NLogN
5	…	do_FMM
Grain		(default = 1.0D0)
Dens_Screen		(default = 1.0D-15)
Extent_Min		(default = 1.0D-3)
FEDim		(default = 10)
LIPN		(default = 2)

NAMELIST &ADPTInp

This NAMELIST includes information of Adaptive density partitioning technique (ADPT) used in GAP method

Parameters

ErrThr_Initial error threshold in ADPT (default = 1.0D-8)

ErrThr_Final error threshold in ADPT, but not used now (default = 1.0D-8)

CubEdg (default = 2.0D-01)

ICut (default = 15)

IEFTyp method to constrict the extended box in GAP method (default = 2)
1 … exact but inefficient
2 … small Error but efficient
3 … manually

NBchEg (default = 5)

ITol (default = 15)

ITail (default = 15)

CubScale (default = 3)

NAMELIST &PSSCF

Parameters

PSFit flag to invoke fitting basis in PS method (default = T)
F … do not use fitting basis
T … use fitting basis

PSFitInCore flag to keep the part of dealiasing matrix on memory space in PS fitting basis (default = F)
F … do not keep the part of dealiasing matrix on memory space, i.e., do file I/O
T … keep the part of dealiasing matrix on memory space

ThrInt threshold value of integration (default = 1.0D-15)

ThrPrim threshold value of integration targeting primitive Gaussian (default = 1.0D-20)

IPrint print option (default = 0)
0 … normal printing
1 … additional information is printed out

NAMELIST &PSNum

Parameters

GridType grid type (default = ‘Prune’)
‘Prune’ … prune scheme based on Ledbedev's grid
‘Lebedev’… Ledbedev's grid
‘Adaptive’… adaptive grid of Krack and Koster
Prune sets of grids are implemented for (NRad, NAng) = (35, 110), (50, 194), (75, 194), (75, 302), (99, 590)

QuadRad quadrature type for radial part (default = ‘EulMac’)
‘EulMac’… Euler–MacLaurin quadrature
‘GauChe’… Gauss–Chebyshev quadrature

CellType atomic partition function (default = ‘SSF’)
‘SSF’ … scheme of Stratmann, Scuseria, and Frisch
‘Becke’ … Becke’s scheme

NRad number of radial integration points in Ledbedev or prune grid (default = 99)

NAng number of angular integration points in Ledbedev or prune grid (default = 590)

GrdTol tolerance for the numerical integration in the adaptive grid scheme (default = 1.0D-5)

BatchType method to store grid points into a batch (default = 0)
0 … store crude grid points into a batch
1 … do sorting of grid points before storing into a batch

GridInCore flag to keep grid batches on memory space (default = F)
F … do not keep grid batches on memory space, i.e., do file I/O
T … keep grid batches on memory space

NAMELIST &DFTNum

Parameters

GridType grid type (default = ‘Prune’)

‘Prune’ … prune scheme based on Ledbedev's grid

‘Lebedev’… Ledbedev's grid

‘Adaptive’… adaptive grid of Krack and Koster

Prune sets of grids are implemented for (NRad, NAng) = (35, 110), (50, 194), (75, 194), (75, 302), (99, 590)

QuadRad quadrature type for radial part (default = ‘EulMac’)

‘EulMac’… Euler–MacLaurin quadrature

‘GauChe’… Gauss–Chebyshev quadrature

CellType atomic partition function (default = ‘SSF’)

‘SSF’ … scheme of Stratmann, Scuseria, and Frisch

‘Becke’ … Becke’s scheme

NRad number of radial integration points in Ledbedev or prune grid (default = 99)

NAng number of angular integration points in Ledbedev or prune grid (default = 590)

GrdTol tolerance for the numerical integration in the adaptive grid scheme (default = 1.0D-5)

BatchType method to store grid points into a batch (default = 0)

0 … store crude grid points into a batch

1 … do sorting of grid points before storing into a batch

NAMELIST &DFT

XCType，XType，CTypeに含まれるparameterはfixされている．HFFac，RSMu，RSFacなどを変えたい場合はXFunとCFunを使う．
TAUHCTH, TAUHCTHHYBが抜けている．神谷君

Parameters

DFTFun		flag to select the DFT routine (default = T)
T	…	use exchange correlation functionals in dftfun_lib
F	…	use exchange correlation functionals in dft_lib

XCType	flag to select standalone DFT exchange correlation functional	
‘HCTH’	[1], (GGA)	
‘HCTH120’	[2], (GGA)	
‘HCTH147’	[2], (GGA)	
‘HCTH407’	[3], (GGA)	
‘B3LYP’	[4], (GGA), (Hybrid)	
‘B97’	[5], (GGA), (Hybrid)	
‘B971’	[1], (GGA), (Hybrid)	
‘B972’	[7], (GGA), (Hybrid)	
‘B97D’	[8], (GGA)	
'M06'	[9], (meta-GGA), (Hybrid)	(DFTFun=.T. only)
'M06_L'	[10], (meta-GGA)	(DFTFun=.T. only)
'M06_HF'	[11], (meta-GGA), (Hybrid)	(DFTFun=.T. only)
'M06_2X'	[9], (meta-GGA), (Hybrid)	(DFTFun=.T. only)
'VS98’	[12], (meta-GGA)	(DFTFun=.T. only)
‘wB97'	[13], (GGA), (LC)	(DFTFun=.T. only)
'wB97X'	[13], (GGA), (LC), (Hybrid)	(DFTFun=.T. only)
'wB97XD'	[14], (GGA), (LC), (Hybrid)	(DFTFun=.T. only)
‘CAMB3LYP’	[15], (GGA), (LC), (Hybrid)	(DFTFun=.T. only)
‘BNL07’	[16], (GGA), (LC)	(DFTFun=.T. only)
‘LCPBE’	[21], (GGA), (LC)	(DFTFun=.T. only)
‘LCwPBE’	[22], (GGA), (LC)	(DFTFun=.T. only)
‘EDF1’		
‘FT97’		
‘LCTPSS’		
‘tauHCTH’		
‘tauHCTHhyb’		
‘TPSS’		

XType	flag to select DFT exchange functionals
‘LDA’	[17], (LDA)
’SLATER’	[17], (LDA)
‘B88’	[18], (GGA)
‘BECKE’	[18], (GGA)
‘PW91’	[19], (GGA)
‘PBE’	[20], (GGA)
‘LCB88’	[21], (GGA), (LC)
‘LCLDA’	[21], (LDA), (LC)

‘LCPBE’	[21], (GGA), (LC)
‘LCwPBE’	[22], (GGA), (LC)
‘B3’	
‘FT97B’	
‘MPW91’	

CType	flag to select DFT correlation functionals
'VWN5'	[23], (LDA)
'VWN5RPA'	[23], (LDA)
'VWN1RPA'	[23], (LDA)
'PZ81'	[24], (LDA)
'PW92'	[25], (LDA)
'P86'	[26], (GGA)
'LYP'	[27], (GGA)
'PW91'	[19], (GGA)
'PBE'	[20], (GGA)
'OP'	[28], (GGA). this functional must be used with ‘B88’ or ‘LCB88’ exchange functional

XFun flag to specify combined exchange functionals

CFun flag to specify combined correlation functionals

XFac mixing factors of exchange functionals

CFac mixing factors of correlation functionals

HFFac scaling factors of Hartree–Fock exchange

RSMu a parameter for the long range correction scheme; this keyword applies only when the exchange functional is range-separated type functional

RSFac scaling factors of Long-range Hartree–Fock exchange; this keyword applies only when the exchange functional is range-separated type functional

Comments

(LDA): LDA type functionals, (GGA): GGA type functionals, (meta-GGA): meta-GGA type functionals, (Hybrid) hybrid-type functional, (LC): long-range corrected functionals

References

[1] F. A. Hamprecht, A. J. Cohen, D. J. Tozer, and N. C. Handy, J. Chem. Phys. 109, 6264 (1998).

[2] A. D. Boese, N. L. Doltsinis, N. C. Handy, and M. Sprik. J. Chem. Phys. 112, 1670 (2000).

[3] A. D. Boese and N. C. Handy, J. Chem. Phys. 114, 5497 (2001).

[4] A. D. Becke, J. Chem. Phys. 98, 5648 (1993). This functional could be specified as (XFun = B88GGA, LDA, CFun = LYP, VWN5, XFac = 0.72, 0.80, CFac = 0.81, 0.19, HFFac = 0.20).

[5] A. D. Becke, J. Chem. Phys. 107, 8554 (1997).

[6] F. A. Hamprecht, A. J. Cohen, D. J. Tozer, and N. C. Handy, J. Chem. Phys. 109, 6264 (1998).

[7] P. J. Wilson, T. J. Bradley, and D. J. Tozer, J. Chem. Phys. 115, 9233 (2001).

[8] S. Grimme, J. Comp. Chem. 27, 1787 (2006).

[9] Y. Zhao and D. G. Truhlar, Theor. Chem. Acc. 120, 215 (2008).

[10] Y. Zhao and D. G. Truhlar, J. Chem. Phys. 125, 194101 (2006).

[11] Y. Zhao and D. G. Truhlar, J. Phys. Chem. A, 110, 5121 (2006); Y. Zhao and D. G. Truhlar, J. Phys. Chem. A, 110, 13126 (2006).
[12] T. van Voorhis and G. E. Scuseria, J. Chem. Phys. 109, 400 (1998).
[13] J.-D. Chai and M. Head-Gordon, J. Chem. Phys. 128, 084106 (2008).
[14] J.-D. Chai and M. Head-Gordon, Phys. Chem. Chem. Phys. 10, 6615 (2008).
[15] T. Yanai, D. Tew, and N. Handy, Chem. Phys. Lett. 393, 51 (2004).
This functional could be specified as (XFun = LCB88, B88, CFun = LYP, VWN5, XFac = 0.46, 0.35, CFac = 0.81, 0.19, HFFac = 0.19, LCMu = 0.33, LCFac = 0.46)
[16] E. Livshits and R. Baer, Phys. Chem. Chem. Phys. 9, 2932 (2007).
[17] J. C. Slater and K. H. Johnson, Phys. Rev. B 5, 844 (1972).
[18] A. D. Becke, Phys. Rev. A 38, 3098 (1988).
[19] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais, Phys. Rev. B 46, 6671 (1992).
[20] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. 77, 3865 (1996); 78, 1396 (1997).
[21] H. Iikura, T. Tsuneda, T. Yanai, and K. Hirao, J. Chem. Phys. 115, 3540 (2001).
[22] O. A. Vydrov and G. E. Scuseria, J. Chem. Phys. 125, 234109 (2006).
[23] S. J. Vosko, L. Wilk, and M. Nusair, Can. J. Phys. 58, 1200 (1980).
[24] J. P. Perdew and A. Zunger, Phys. Rev. B 23, 5048 (1981).
[25] J. P. Perdew and Y. Wang, Phys. Rev. B 45, 13244 (1992).
[26] J. P. Perdew, Phys. Rev. B 33, 8822 (1986).
[27] C. Lee, W. Yang, and R. G. Parr, Phys. Rev. B 37, 785 (1988).
[28] T. Tsuneda, T. Suzumura, and K. Hirao, J. Chem. Phys. 110, 10664 (1999); 111, 5656 (1999).

Module SCFGrad

Function

This module controls the gradient calculation. This module can be usually employed after the self-consistent calculation (SCF) (see also SCF module).

Required NAMELIST

&Control
&SCF
&SCFGrad

NAMELIST &SCFGrad

Parameters

CoulDType flag for Coulomb integration (default = ‘   ’)
‘   ’ … use the same method as the previous SCF calculation
‘Analy’ … use analytical method for Coulomb-type integration
‘RI’ … use resolution-of-identity (RI) approximation

ExchDType flag for exchange integration (default = ‘   ’)
‘   ’ … use the same method as the previous SCF calculation
‘Analy’ … use analytical method for exchange-type integration

Grad flag to gradient calculation (default = .TRUE.)
T … calculate energy gradient
F … do not calculate energy gradient

GradGFC flag to GFC gradient calculation (default = .FALSE.)
T … do GFC gradient calculation
F … do not GFC gradient calculation

DenOnly (default = .FALSE.)
T …
F …

ReadDenEW (default = .FALSE.)
T …
F …

DenNR flag to use nonrelativistic density in the relativistic gradient (default = .FALSE.)
F … do not use nonrelativistic density in the relativistic gradient
T … use nonrelativistic density in the relativistic gradient

IPrint print option (default = 0)

Module SOSCFGrad

Function

This module controls the gradient calculation with spin–orbit interaction. This module can be usually employed after the self-consistent calculations (SCF) with SO interaction.

Required NAMELIST

&Control
&SCF
&SOSCF
&SOSCFGrad

NAMELIST &SOSCFGrad

Parameters

CoulDType flag for Coulomb integration (default = ‘ ’)
‘ ’ … use the same method as the previous SOSCF calculation
‘Analy’ … use analytical method for Coulomb-type integration
‘RI’ … use resolution-of-identity (RI) approximation

ExchDType flag for exchange integration (default = ‘ ’)
‘ ’ … use the same method as the previous SOSCF calculation
‘Analy’ … use analytical method for exchange-type integration

Grad flag to gradient calculation (default = .TRUE.)
T … calculate energy gradient
F … do not calculate energy gradient

GradGFC flag to GFC gradient calculation (default = .FALSE.)
T … do GFC gradient calculation
F … do not GFC gradient calculation

DenOnly (default = .FALSE.)
T …
F …

ReadDenEW (default = .FALSE.)
T …
F …

DenNR flag to use nonrelativistic density in the relativistic gradient (default = .FALSE.)
F … do not use nonrelativistic density in the relativistic gradient
T … use nonrelativistic density in the relativistic gradient

IPrint print option (default = 0)

NAMELIST &Int1D

Parameters

ThrInt threshold value of integration (default = 1.0D-15)

ThrPrim threshold value of integration targeting primitive Gaussian (default = 1.0D-20)

IPrint print option (default = 0)

NAMELIST &Int2D

Parameters

Int2DType specifies the method to evaluate ERI derivatives (default = ’Libint’)
Libint … use Libderiv library ver. 1
Libint2 … use Libderiv library ver. 2

DenScreen flag to invoke ERI derivatives prescreening using two-particle density matrix elements (default = T)
F … do not use prescreening
T … use prescreening

DenCut threshold for ignoring ERI derivatives in integral prescreening; this parameter has no effect when DenScreen = F (default = 1.0D-13)

DTol products of four primitives with preexponential factor multiplied by contraction coefficients less than DTol are skipped; this parameter has no effect when DenScreen = F (default = 1.0D-12)

ThrInt products of four primitives with preexponent factor less than ThrInt are skipped (default = 1.0D-15)

ThrPrim products of primitives with preexponential factor less than ThrPrim are skipped (default = 1.0D-20)

NAMELIST &RIInt2D

Parameters

Int2DType specifies the method to evaluate three-center ERI derivatives (default =”Libint”)
‘Libint’ … use Libderiv library ver. 1

ThrInt products of four primitives with preexponent factor less than ThrInt are skipped (default = 1.0D-15)

ThrPrim products of primitives with preexponential factor less than ThrPrim are skipped (default = 1.0D-20)

Module TDDFT

Function

This module controls the calculation of molecular excitation energies by time-dependent density functional theory computations (or time-dependent Hartree–Fock, also known as the Random Phase Approximation).

Required NAMELIST

&Control
&TDDFT
&DFT (optionally if DFT = .TRUE.)
&DFTNum (optionally if DFT = .TRUE.)
&Int2 (optionally if CoulType = ’Analy’ and/or ExchType = ’Analy’)

Input files

Name.Basis
Name.Geom
Name.SCF_Info
Name.MO
Name.Overlap
Name.OrbEne
Name.OccNum
Name.Dipole
Name.SymInfo

Output files

Name.TDDFT_Info
Name.TDEne
Name.TDVec1
Name.TDVec2

NAMELIST &TDDFT

Parameters

NStates number of excited states to be solved (default = 1)

CIType flag to whether the full TDDFT or the Tamm/Dancoff approximation is used
'CIS' … Tamm/Dancoff approximation
'RPA' … full TDDFT

Triplet flag to calculate triplet excited states; this keyword applies only when the reference is a closed shell (default = F)
T … calculate both singlet and triplet excited states
F … calculate only singlet excited states

NFrzOA number of frozen alpha occupied orbitals (default = 0)

NFrzOB number of frozen beta occupied orbitals (default = 0)

NFrzVA number of frozen alpha virtual orbitals (default = 0)

NFrzVB number of frozen beta virtual orbitals (default = 0)

NActCoreA number of active alpha core orbitals (default = 0)

NActCoreB number of active core beta core orbitals (default = 0)

NBlock number of trial vectors contracted with integrals (default = 1)

MaxIter maximum number of iterations in Davidson diagonalization (default = 200)

MaxSub maximum number of subspace for Davidson diagonalization (default = 600)

ThrConv convergence threshold for residual vectors in Davidson diagonalization (default = 1.0D-05)

IPrint print flag (default = 0)
0 … normal printing

PrintCD Full rotational strength tensor elements, relevant for circular dichroism of oriented molecules, are printed along with normal scalar contributions of the rotational strength (length and velocity forms). They are given in 10^{-40} CGS unit. In addition, velocity form of dipole transition moments, orbital angular moments (magnetic dipole moments), and quadrupole transition moments (length and velocity forms) are printed in atomic unit with associated oscillator strengths. The length quadrupole transition moments are not traceless in this module. In order to activate this option, CalDip=T should be set in the NAMELIST &MDInt1. For definition of transition moments and tensors, see Hansen and Bouman, Adv. Chem. Phys. **44** 545 (1980).
T … print quantities related circular dichroism as described above
F … print only length form of dipole transition moments

Module SOTDDFT

Function

This module controls the calculation of molecular excitation energies by spin-orbit time-dependent density functional theory computations.

Required NAMELIST

&Control
&SOTDDFT
&DFT (optionally if DFT = .TRUE.)
&DFTNum (optionally if DFT = .TRUE.)
&Int2 (optionally if CoulType = ’Analy’ and/or ExchType = ’Analy’)

Input files

Name.Basis
Name.Geom
Name.SOSCF_Info
Name.MO
Name.Overlap
Name.OrbEne
Name.OccNum
Name.Dipole
Name.SymInfo

Output files

Name.SOTDDFT_Info
Name.SOTDEne
Name.SOTDVec

NAMELIST &SOTDDFT

Parameters

NStates number of excited states to be solved (default = 1)

CIType flag to whether the full TDDFT or the Tamm/Dancoff approximation is used
‘CIS’ … Tamm/Dancoff approximation (default)

NonCol flag to whether the non-collinear spin-flip TDDFT calculation is performed (default = F)
T … perform non-collinear spin-flip TDDFT calculation
F … does not perform non-collinear spin-flip TDDFT calculation (not yet implemented)

NFrzOA number of frozen alpha occupied orbitals (default = 0)

NFrzOB number of frozen beta occupied orbitals (default = 0)

NFrzVA number of frozen alpha virtual orbitals (default = 0)

NFrzVB number of frozen beta virtual orbitals (default = 0)

NActCoreA number of active alpha core orbitals (default = 0)

NActCoreB number of active core beta core orbitals (default = 0)

NBlock number of trial vectors contracted with integrals (default = 1)

MaxIter maximum number of iterations in Davidson diagonalization (default = 200)

MaxSub maximum number of subspace for Davidson diagonalization (default = 600)

ThrConv convergence threshold for residual vectors in Davidson diagonalization (default = 1.0D-05)

IPrint print flag (default = 0)
0 … normal printing

PrintCD Full rotational strength tensor elements, relevant for circular dichroism of oriented molecules, are printed along with normal scalar contributions of the rotational strength (length and velocity forms). They are given in 10^{-40} CGS unit. In addition, velocity form of dipole transition moments, orbital angular moments (magnetic dipole moments), and quadrupole transition moments (length and velocity forms) are printed in atomic unit with associated oscillator strengths. The length quadrupole transition moments are not traceless in this module. In order to activate this option, CalDip=T should be set in the NAMELIST &MDInt1. For definition of transition moments and tensors, see Hansen and Bouman, Adv. Chem. Phys. **44** 545 (1980).
T … print quantities related circular dichroism as described above
F … print only length form of dipole transition moments

Module TDGrad

Function

This module controls the analytic gradient calculation for the excitation energies from a time-dependent density functional theory calculation

Required NAMELIST

&Control
&DFT (optionally if DFT = .TRUE.)
&DFTNum (optionally if DFT = .TRUE.)
&Int2
&Int2D (optionally if CoulDType = ’Analy’ and/or ExchDType = ’Analy’)
&TDGrad

Input files

Name.Basis
Name.Geom
Name.SCF_Info
Name.MO
Name.DenEW (optionally if ReadDenEW = .TRUE.)
Name.OrbEne
Name.OccNum
Name.TDDFT_Info
Name.TDEne
Name.TDVec1
Name.TDVec2

Output files

Name.ExGrad.(State)

NAMELIST &TDGrad

Parameters

Root target excited-states used for the geometrical derivative calculation (default = 1)

ThrConv convergence threshold for residual vectors in the solution of the Z-vector equations (default = 1.0D-05)

MaxIter maximum number of iterations in solving the Z-vector equations (default = 200)

ReadDenEW flag to ground-state energy-weighted density matrix (default = .FALSE.)
‘T’ … read from Name.DenEW file
‘F’ … construct energy-weighted density matrix

NBlock number of trial vectors contracted with integrals (default = 1)

TotGrad flag to whether total excited-state gradients or difference energy gradients are calculated (default = .TRUE.)
‘T’ … total excited-state gradients
‘F’ … difference energy gradients

CoulDType flag for Coulomb integration (default = ‘Analy’)
‘Analy’ … use analytical method for Coulomb-type integration

ExchDtyp flag for Exchange integration (default = ‘Analy’)
‘Analy’ … use analytical method for Exchange-type integration

Grad flag to gradient calculation (default = .TRUE.)
T …
F …

IPrint print flag (default = 0)
0 … normal printing

Comments

‘DFTFun’ in NAMELIST &DFT must be .TRUE.

Module DFTD3

Function

This module controls the DFT-D empirical dispersion correction calculation of the energy and its analytic gradient for density functional theory and Hartree–Fock calculations. This module contains interface to the DFTD3 program developed by S. Grimme, which is freely downloadable from webpage (http://toc.uni-muenster.de/DFTD3) under the terms of the GNU General Public License as published by C the Free Software Foundation; either version 1.

Required NAMELIST

&Control
&DFTD3

Input files

Name.Geom
Name.TotEne
Name.Grad
Name.SCF_Info

Output files

Name.TotEne
Name.Grad

NAMELIST &DFTD3

Parameters

Energy flag to perform DFT-D energy calculation (default = .TRUE.)
‘T’ … perform DFT-D energy calculation
‘F’ … skip DFT-D energy calculation

Echo flag to control print out option (default = .TRUE.)
‘T’ … print out detailed information
‘F’ … disable print out

Grad flag to perform DFT-D gradient calculation (default = .TRUE.)
‘T’ … perform gradient calculation
‘F’ … skip gradient calculation

Anal flag to performs a detailed analysis of pair contributions (default = .FALSE.)
‘T’ … perform detailed analysis
‘F’ … skip detailed analysis

Func flag to select DFT exchange correlation functional (default = ‘’)
‘bp86’
‘blyp’
‘b97d’
‘pbe’
‘mpwlyp’
‘b3lyp’
‘b3pw91’
‘bh-lyp’
‘camb3lyp’
‘pbe0’
‘wb97xd’
‘m06’
‘m06l’
‘m062x’
‘m06hf’
‘b2-plyp’
‘b2gp-plyp’
‘mpw2-plyp’
‘LCwPBE’

Version flag to control version (default = 4)
2 … Switch to old DFT-D2 version
3 … Switch to DFT-D3 version 3
4 … Switch to DFT-D3 version 4 (version 3 with Becke–Johnson (BJ) damping)

Comments

Func flag should be specified to match the DFT exchange correlation functional specified in &DFT namelist. Correcting Hartree–Fock results is only recommended with BJ-damping. DFTD3 must be calculated after SCFGrad/SOSCFGrad when energy gradients are required with Grad = T.

Module MP2

Function

This module controls the calculation of the Møller–Plesset energy correction in the second order. It works for spin-restricted closed-shell Hartree–Fock (RHF) wavefunctions (RMP2), spin-unrestricted open-shell HF (UHF) wavefunctions (UMP2), and spin-restricted open-shell HF (ROHF) wavefunctions (ROHF-MP2). The resolution of identity approximation MP2 (RI-MP2) is available for RMP2, UMP2, and ROHF-MP2 energies.

The same MP2 correction may be calculated with the CC program (the input of which is described in Module CC). The MP2 module is preferable to the CC module when dealing with large systems. The present program has been used for MP2 calculations involving up to 250 basis functions and RI-MP2 calculations involving up to 10000 basis functions.

Required NAMELIST

&Control
&MP2
&Int2
&RIInt2

Prepared input data files

Name.Basis
Name.Basis_RIC (only required for RI-MP2 calculations)
Name.Geom
Name.SCF_Info
Name.TotEne
Name.MO
Name.OrbEne

Created output data files

Name.TotEne
Name.TotEne_MP2

NAMELIST &MP2

Parameters

MP2Type keyword for selecting type of MP2 calculation (default = ‘Direct’)
Direct … direct MP2 calculation
RIMP2 … RI-MP2 calculation
RI … RI-MP2 calculation

MP1 flag to carry out a MP1 calculation (default = .FALSE.)
F … not performing MP1 calculation
T … performing MP1 calculation

NFrzOA number of core (occupied) alpha orbitals excluded from the MP2 calculation (default = 0)

NFrzOB number of core (occupied) beta orbitals excluded from the MP2 calculation (default = 0)

NFrzVA number of virtual alpha orbitals excluded from the MP2 calculation (default = 0)

NFrzVB number of virtual beta orbitals excluded from the MP2 calculation (default = 0)

COSFac scaling factors of opposite-spin contributions of MP2 correlation energy (default = 1.0D+00)

CSSFac scaling factors of same-spin contributions of MP2 correlation energy (default = 1.0D+00)

RIOrthType keyword for selecting scheme for inversion of a two-center matrix of auxiliary basis integrals in RI-MP2 calculations (default = ‘Cholesky’)
‘Cholesky’… use Cholesky decomposition based scheme
‘Canonical’… use canonical orthogonalization based scheme

ThrRI threshold for linear dependency of auxiliary basis functions on inversion of a two-center matrix of auxiliary basis integrals in RI-MP2 calculations (default = 1.0D-6); ThrOvlp is available only for RIOrthType = 'Canonical'

MP2BatchLv selecting the size of batch of orbitals in the MP2 calculation (default = 0: use single batch)

InCore flag to carry out a in-core RI-MP2 calculation (default = .FALSE.)
F … not performing in-core RI-MP2 calculation
T … performing in-core RI-MP2 calculation

VPair flag to carry out a RI-MP2 parallel calculation based on the virtual orbital based MPI task distribution (default = .TRUE.)
F … performing MP2 parallel calculation with occupied orbital task distribution
T … performing MP2 calculation with virtual orbital task distribution

IPrint print flag (default = 0)

0	…	normal printing
1	…	additional information is printed out

Comments about memory control

The user should take care of the size of required main memory requirements when MP2 calculations are performed.

In the case of direct MP2 calculations (MP2Type = ‘Direct’), ovn^2/n_o words of main memory are required. (o: number of occupied orbitals, v: number of virtual orbitals, n: number of basis functions, n_o: number of occupied orbital batch). The user can control total memory requirements by setting MP2BatchLv that corresponds to the number of occupied orbital batches $n_o = 2^{MP2BatchLv}$. The default is MP2BatchLv = 0 for the best computational performance using available memory as much as possible and avoiding multiple computation of four-center atomic orbital integrals. However, this default is not suitable for the calculations of large molecules where the required memory sizes exceed the limit of available memory sizes. To reduce the required memory sizes less than available memory sizes, the user should increase MP2BatchLv from 0 to larger numbers.

In the case of semi-direct RI-MP2 calculations (MP2Type = ‘RIMP2’ and InCore=F), $ovn_x/(pn_v)$ words of main memory (n_v: number of virtual orbital batch, n_x: number of auxiliary basis functions, p: number of processor nodes) are required. The user can control total memory requirements by setting MP2BatchLv that corresponds to the number of occupied orbital batches $n_v = 2^{MP2BatchLv}$. The default is MP2BatchLv = 0 for the best computational performance using available memory as much as possible. However, this default is not suitable for the calculations of large molecules where the required memory sizes exceed the limit of available memory sizes. To reduce the required memory sizes less than available memory sizes, the user should increase MP2BatchLv from 0 to larger numbers.

For the efficient RI-MP2 calculations of massively parallel supercomputers such as k computer, the user can use full in-core RI-MP2 scheme that is faster than semi-direct RI-MP2 by setting InCore = T. Note that the required memory size of full in-core RI-MP2 calculations is $2ovn_x/p$ words and should not exceed the available memory sizes.

Module MP2Grad

Function

This module controls the density matrix and analytical energy gradient calculation at MP2 level. This module can be usually employed after the self-consistent field (SCF) calculation. Properties like atomic populations, electric moments, and electric static potentials at the MP2 level can be calculated using the MP2 density matrix after the MP2 gradient calculation. At present, RI-MP2 analytical energy gradient calculation with RHF reference is available. Frozen core and virtual orbitals are not supported.

Required NAMELIST

&Control
&MP2Grad

Prepared input data files

Name.Basis
Name.Basis_RIC (only required for RI-MP2 calculations)
Name.Geom
Name.SCF_Info
Name.TotEne
Name.MO
Name.OrbEne
Name.Dens

Created output data files

Name.Dens
Name.TotEne
Name.Grad

NAMELIST &MP2Grad

Parameters

Grad flag to gradient calculation (default = .TRUE.)
T … calculate energy gradient
F … do not calculate energy gradient and only calculate MP2 density matrix

MP2Type keyword for selecting type of MP2 gradient calculation (default = ‘RIMP2’)
RIMP2 … RI-MP2 calculation
RI … RI-MP2 calculation

RIOrthType keyword for selecting scheme for inversion of a two-center matrix of auxiliary basis integrals in RI-MP2 calculations (default = ‘Cholesky’)
‘Cholesky’… use Cholesky decomposition based scheme
‘Canonical’… use canonical orthogonalization based scheme

ThrRI threshold for linear dependency of auxiliary basis functions on inversion of a two-center matrix of auxiliary basis integrals in RI-MP2 calculations (default = 1.0D-6); ThrOvlp is available only for RIOrthType = 'Canonical'

MP2BatchLv selecting the size of batch of orbitals in the MP2 calculation (default = 0: use single batch)

MaxIterCPHF maximum number of coupled perturbated Hartree-Fock (CPHF) iterations (default = 100)

ThrConvCPHF convergence threshold for CPHF equation (default = 1.0D-05)

IPrint print option (default = 0)
0 … normal printing
1 … additional information is printed out

Module CC

Function

This module controls the calculation of correlation energy with the coupled-cluster (CC) ansatz. The molecular orbitals may be spin-restricted or spin-unrestricted. (Serial only)

Required NAMELIST

&Control
&CC

Prepared input data files

Name.SCF_Info
Name.TotEne
Name.MO
Name.OrbEne
Name.ERI_Info
Name.ERI

Created output data files

Name.CC_Info
Name.TNN (NN is a two-digit integer up to the order of CC wavefunction)

NAMELIST &CC

Parameters

Guess controls the initialization of cluster amplitude vectors (default = ’Init’)
‘Init’ … initialize by MP2 amplitude (doubles) or zero (otherwise)
‘Read’ … initial amplitude vectors are read from files

NoT1 flag to discard single excitations (default = .FALSE.); this option works when NMax = 2
F … do not discard singles
T … discard singles

CC2 flag to use CC2 (second-order approximation to coupled-cluster with singles and doubles) method (default = F); this option works when NMax = 2
F … do not use CC2
T … use CC2

DoTran flag to carry out two- and four-index integral transformation prior to solving the CC equations (default = .TRUE.)
F … do not perform integral transformation; MO integrals are read from files
T … perform integral transformation

NMax order of the CC wavefunction (default = 2)

NFrzOA number of frozen alpha occupied orbitals (default = 0)

NFrzOB number of frozen beta occupied orbitals (default = 0)

NFrzVA number of frozen alpha virtual orbitals (default = 0)

NFrzVB number of frozen beta virtual orbitals (default = 0)

MaxIter maximum number of CC iterations (default = 100)

MaxDIIS maximum number of DIIS error vectors; MaxDIIS <= 1 indicates that no DIIS method is used (default = 6)

ThrGrd convergence criterion for RMS gradient (default = 1.0D-07)

ThrEne convergence criterion for correlation energy change (default = 1.0D-07); convergence is achieved when both correlation energy change and RMS gradient are below criterions

CalcMP flag to calculate the third-order Møller–Plesset (MP3) correlation energy prior to CC iteration (default = .FALSE.)
F … do not calculate MP3 energy
T … calculate MP energies

IPrint print option (default = 0)
0 … default printing
1 … additional information is printed out

Comments

Type of the CC wavefunction is determined by NMax, which is the maximum level of excitation of the cluster operator. The default is NMax = 2, which leads to CCSD (coupled-cluster with singles and doubles) wavefunction. All the excitations with an excitation level equal to or less than NMax are explicitly involved in the CC wavefunction. The following special options are available when NMax = 2: (1) Single excitations can be eliminated by setting NoT1 = T, leading to CCD (coupled-cluster with doubles) wavefunction (2) CC2 = T invokes the CC2 (second-order approximation to CCSD) method.

The CC module requires atomic-orbital ERI files Name.ERI_Info and Name.ERI, which are created by running a disk-oriented SCF calculation. This can be done by specifying Direct = F in &SCF and IntType = ’MD4’ in &Int2.

Module SOCC

Function

This module controls the calculation of correlation energy with the two-component spin–orbit coupled-cluster (SOCC) ansatz. (serial only)

Required NAMELIST

&Control
&SOCC

Prepared input data files

Name.SOSCF_Info
Name.TotEne
Name.MOSO
Namem.OrbEne
Namem.ERI_Info
Name.ERI

Created output data files

Name.SOCC_Info
Name.TNN (NN is a two-digit integer up to the order of SOCC wavefunction)

NAMELIST &SOCC

Parameters

Guess controls the initialization of cluster operators (default = 'Init')
'Init' … initialize by MP2 amplitude (doubles) or zero (otherwise)
'Read' … initial operator amplitudes are read from files

NoT1 flag to discard single excitations (default = .FALSE.); this option works when NMax = 2
F … do not discard singles
T … discard singles

CC2 flag to use SOCC2 (second-order approximation to spin–orbit coupled-cluster with singles and doubles) method (default = .FALSE.); this option works when NMax = 2
F … do not use SOCC2
T … use SOCC2

DoTran flag to carry out two- and four-index integral transformation prior to solving the CC equations (default = .TRUE.)
F … do not perform integral transformation; MO integrals are read from files
T … perform integral transformation

NMax order of the SOCC wavefunction (default = 2)

NFrzOA number of frozen alpha occupied orbitals (default = 0)

NFrzOB number of frozen beta occupied orbitals (default = 0)

NFrzVA number of frozen alpha virtual orbitals (default = 0)

NFrzVB number of frozen beta virtual orbitals (default = 0)

MaxIter maximum number of CC iterations (default = 100)

MaxDIIS maximum number of DIIS error vectors (default = 6)
MaxDIIS <= 1 indicates that no DIIS method is used

ThrGrd convergence criterion for RMS gradient (default = 1.0D-07)

ThrEne convergence criterion for correlation energy change (default = 1.0D-07); convergence is achieved when both correlation energy change and RMS gradient are below criterions.

CalcMP flag to calculate third-order spin–orbit Møller–Plesset (SOMP3) correlation energy prior to CC iteration (default = F)
F … do not calculate SOMP3 energy
T … calculate SOMP3 energy

IPrint print option (default = 0)
0 … default printing
1 … additional information is printed out

Comments

The meanings of almost all of the parameters are same as those in &CC. The SOCC module requires the atomic-orbital ERI files Name.ERI_Info and Name.ERI. One needs to run scf.exe in the disk-oriented mode (Direct = F in &SCF and IntType = ’MD4’ in &Int2) to create those ERI files. Note that soscf.exe runs only in the direct mode.

Module CCLR

Function

This module controls the calculation of molecular electronic properties (ground-state electronic properties, excitation energy, and ionization energy) with the coupled-cluster linear response (CCLR) ansatz. (Serial only)

Required NAMELIST

&Control
&CCLR

Required input data files

Name.TNN (NN is a two-digit integer up to the order of CC wavefunction)
Name.CC_Info
Name.SCF_Info
Name.OrbEne
Name.INTNN (MO integral files created in cc.exe. NN is a two-digit integer from 01 to 13)

Created output data files

Name.RNN (Optional. NN is a two-digit integer up to the order of CC wavefunction)
Name.CCLR_ExcEne (Optional)
Name.CCLR_IP (Optional)

NAMELIST &CCLR

Parameters

GuessR controls the initialization of excitation vector (default = ’None’)
‘None’ … initialize using orbital energy differences (single excitations) or zero (otherwise)
‘Read’ … initial vectors are read from files

ExcdEne flag to calculate excitation energies (default = .FALSE.)
F … calculate excitation energies
T … do not calculate excitation energies

IonzPot flag to calculate ionization potentials (default = .FALSE.)
F … do not calculate ionization potentials
T … calculate ionization potentials

NMaxR order of excitation operator (default = 2)

NState number of excited states to be solved; this parameter has no effect when ExcdEne = .FALSE. (default = 1)

NStateIP number of ionization states to be solved; this parameter has no effect when IonzPot = .FALSE. (default = 1)

NumVec initial number of expansion vectors in Davidson diagonalization for excitation energy (default = 1); the value should be equal to or larger than NState

NumVecIP initial number of expansion vectors in Davidson diagonalization for ionization potential (default = 1); the value should be equal to or larger than NStateIP

NumVecMax maximum number of expansion vectors in Davidson diagonalization (default = 30); the value should be equal to or larger than 2*MAX(NState, NStateIP)

MaxItDiag maximum number of iterations in Davidson diagonalization (default = 100)

ARTol convergence threshold for residual vectors in Davidson diagonalization (default = 1.0D-06)

ThreDiagE convergence threshold for energy changes in Davidson diagonalization (default = 1.0D-09)

RPrintTol tolerance for printing out excitation operator amplitudes (default = 1.0D-01)

ThreResidue convergence threshold for residual vectors in solving linear equations for Lambda vector, Jacobian left eigenvectors, and Zeta vectors (default = 1.0D-06)

IPrint print option (default = 0)
0 … default printing
1 … additional information is printed out

Module SOCCLR

Function

This module controls the calculation of molecular electronic properties (ground-state electronic properties, excitation energy, and ionization energy) with the spin–orbit coupled-cluster linear response (SOCCLR) ansatz. (Serial only)

Required NAMELIST

&Control
&SOCCLR

Required input data files

Name.TNN (NN is a two-digit integer up to the order of CC wavefunction)
Name.SOCC_Info
Name.SOSCF_Info
Name.OrbEneSO
Name.INTNN (MO integral files created in socc.exe. NN is a two-digit integer from 01 to 13)

Created output data files

Name.RNN (Optional. NN is a two-digit integer up to the order of CC wavefunction)
Name.SOCCLR_ExcEne (Optional)
Name.SOCCLR_IP (Optional)

NAMELIST &SOCCLR

Parameters

GuessR controls the initialization of excitation vector (default = ’None’)
‘None’ … initialize using orbital energy differences (single excitations) or zero (otherwise)
‘Read’ … initial vectors are read from files

ExcdEne flag to calculate excitation energies (default = .FALSE.)
F … calculate excitation energies
T … do not calculate excitation energies

IonzPot flag to calculate ionization potentials (default = .FALSE.)
F … do not calculate ionization potentials
T … calculate ionization potentials

NMaxR order of excitation operator (default = 2)

NState number of excited states to be solved; this parameter has no effect when ExcdEne = .FALSE. (default = 1)

NStateIP number of ionization states to be solved; this parameter has no effect when IonzPot = .FALSE. (default = 1)

NumVec initial number of expansion vectors in Davidson diagonalization for excitation energy (default = 1); the value should be equal to or larger than NState

NumVecIP initial number of expansion vectors in Davidson diagonalization for ionization potential (default = 1); the value should be equal to or larger than NStateIP

NumVecMax maximum number of expansion vectors in Davidson diagonalization (default = 30); the value should be equal to or larger than 2*MAX(NState, NStateIP)

MaxItDiag maximum number of iterations in Davidson diagonalization (default = 100)

ARTol convergence threshold for residual vectors in Davidson diagonalization (default = 1.0D-06)

ThreDiagE convergence threshold for energy changes in Davidson diagonalization (default = 1.0D-09)

RPrintTol tolerance for printing out excitation operator amplitudes (default = 1.0D-01)

ThreResidue convergence threshold for residual vectors in solving linear equations for Lambda vector, Jacobian left eigenvectors, and Zeta vectors (default = 1.0D-06)

IPrint print option (default = 0)
0 … default printing
1 … additional information is printed out

Comments

The meanings of almost all of the parameters are same as those in &CCLR.

Module R4QMC

Function

This module controls the quantum Monte Carlo (QMC) calculation. Nonrelativistic VMC and DMC, relativistic (ZORA) VMC and DMC calculations and corresponding optimization of correlation factor parameters are available.

Required NAMELIST

&Control
&QMCCtrl
&QMCMPI (MPI only)

Required Cards

QMCRun
QMCCusp
QMCOpt
QMCJastrow

Input files

Name.Basis
Name.Geom
Name.MO
Name.SCFInfo
Name.QMCXin
Name.QMCSXin
Name.Pot_MPCore
Name.Basis_MPCore
Name.CSftQMC

Output files

Name.QMCOut
Name.QMCOrp
Name.QMCDat
Name.QMCXyz
Name.QMCHst

Q&A

Q1. How to use nonrelativistic cusp correction?
A1. Set QMCCusp card, and set RelCusp = F, RelBound = F in &QMCCtrl

Q2. How to use relativistic cusp correction?
A2. Set QMCCusp card, and set RelCusp = T, RelBound = T in &QMCCtrl

Q3. How to calculate force?
A3-1. Set SForce in DataSet section
It computes bare estimator of Hellmann–Feynman force
A3-2. Set Force in DataSet section
It computes modified estimator of Hellmann–Feynman force
Modified estimator depends on the wave function
A3-3. Set NucDeriv & Force in DataSet section and set CalcPulay = T in &QMCCtrl

It computes modified Hellmann–Feynman force + Pulay term
Make sure NucDeriv is set above Force in DataSet section

Q4. How to optimize Jastrow factor parameters?
A4. Set QMCJastrow card and set RunName = ‘OPT’ in &QMCCtrl
In optimization, NBlock in QMCRun card has different meaning

NBlock ${NBlk1} ${NBlk2}

${NBlk1} … # of equilibrium block in generating 1 fixed-sample-points
${NBlk2} … Max # of fixed-sample-points generation

Set NOptFix in &QMCCtrl if required

Q5. How to restart calculation?
A5. After one VMC or DMC calculation finished, the final electron configurations are saved in the formatted file ${Name}.QMCXyz${core_number}, where ${core_number} is in null for core0 and in I10 format for others. Rename these files as ${Name}.QMCXin${core_number} and set “Init 2” in QMCRun card.

NAMELIST &QMCMPI

Parameters

TotNWlk_MPI total number of walkers distributed among all processors (default = 0)

0 … NStdWlk, NMaxWlk, NMinWlk keywords in NAMELIST &QMCCtrl is used for all nodes

Other … input number of walkers are distributed among all processors

thrNSample total number of required samples in double precision (NYI)

NAMELIST &QMCCtrl

Parameters

NNucSet integer to specify how many nuclei-set are given in input file (default = 1)

NPrimSet integer to specify how many primitive-set are given in input file (default = 1)

NElec number of total electrons (default = 0)
0 … automatically determine the number of electrons by atomic charges and Charge keyword

RandSeed random number seed in integer (default = 0)
0 … use clock time as the random number seed

NStdWlk Standard number of walkers (default = 100)

NMaxWlk maximum number of walkers (default = 200)

NMinWlk minimum number of walkers (default = 50)

InitR initialization radius in double precision (default = 1.0D0)

InitR0 initialization threshold of inter-particle distance (default = 0.0D0)

QMCBasis flag to use QMCBasis cards as input basis set (default = .FALSE.)
F … read basis set from Name.Basis file
T … read basis set from QMCBasis card in input file

QMCMO flag to use prepared Name.QMCMO file
F … generate Name.QMCMO file from Name.MO file
T … use prepared Name.QMCMO file

Charge total charge of the system (default = 0)
0 … neutral system

NTChem flag to specify R4QMC is used in connection with NTChem (default = .TRUE.)
F … R4QMC is running standalone, no Name.Basis file, Name.MO file, and so on
T … NTChem HF/KS run is pre-performed and files are prepared

Unit length unit in R4QMC, overwritten by Unit keyword in &Control (default = “Bohr”)
Bohr … Bohr unit is used
Ang … Angstrhom unit is used

useFile flag to use Name.QMCOut file as output (default = .FALSE.)
F … use standard output stream
T … print output in Name.QMCOut file

DLevel integer to specify debug print level (default = 0)
0 … no debug print
1 – 5 … do debug print

ADWlk flag to control branching step in DMC with one-by-one electron move (default = .FALSE.)
F … do branching after all electrons are moved
T … do branching after each electron is moved

ET1stOnly flag to control trial energy evaluation in DMC (default = .FALSE.)
F … update trial energy in every block
T … determine trial energy in the starting of DMC

RunName calculation to be performed (default = “VMC”)
VMC … perform variational Monte Carlo calculation
DMC … perform nonrelativistic diffusion Monte Carlo calculation
ZMC … perform relativistic (ZORA) diffusion Monte Carlo calculation
OPT … optimize Jastrow factor parameter (for scalar w.f.)
FPC … evaluate local quantities at given fixed point
SCAN … evaluate local quantities on the given 1 dimensional line
MOS … evaluate MO values (NYI)
SOVMC… perform 2 component spin–orbit variational Monte Carlo calculation
SOOPT … optimize Jastrow factor parameter (for 2 component w.f.)

NGrid number of angular grid points used in non-local pseudo potential evaluation (default = 18)
6, 18, 26, 50 are available numbers

fineGet flag to kill ill-conditioned samples in accordance with local energy value (default = .FALSE.)
F …
T …

NOptFix number to specify how many times fixed samples are used in optimization (default = 10)
1 … use free sample optimize driver
> 1 … use fixed sample optimize driver

NInterval interval to sample electron positions in fixed-sample generation (default = 20)

QMCCoreBasis flag to use QMCCoreBasis cards as input core basis set (default = .FALSE.)
F … read basis set from Name.Basis_MPCore file
T … read basis set from QMCCoreBasis card in input file

NExtraSP number of extra sample points in optimization

NDivHist number of division of histogram range (default =)

MaxRHist maximum range of histogram in a.u.

TypeFN type of finite nuclei (default = “HCS”)
“HCS” … use homogeneously charged sphere
“GCD” … use Gaussian charge distribution

fineOpt flag to control sample point selection in optimization (default = .TRUE.)
F … use all sample points
T … select sample points by local energy values

LightFix flag to control fixed sample file in optimization (default = .FALSE.)
F … store w.f. values in fixed sample file
T … store only electron positions in fixed sample file and compute w.f. each optimization step

RelCusp flag to use relativistic cusp correction (default = .FALSE.)
F … use nonrelativistic cusp correction if QMCCusp card exists
T … use relativistic (ZORA) cusp correction if QMCCusp card exists

RelBound flag to use relativistic boundary in cusp correction initialization (default = .FALSE.)
### please set the same value as RelCusp

NMinRelCusp lower bound of reference point in relativistic cusp correction initialization (default = 30)

DispCusp flag to display cusp correction initialization (DEBUG: default = .FALSE.)
F … do not display cusp correction initialization
T … display cusp correction initialization information

NonNodeCross flag to kill node-crossing move in VMC run (DEBUG: default = .FALSE.)
F … do not reject node-crossing move in VMC
T … reject node-crossing move in VMC

DistortV flag to control velocity distortion in nonrelativistic DMC (default = .TRUE.)
F … do classical DMC
T … do stabilized DMC by distorting velocity

BlockOut flag to control block-wise output (default = .FALSE.)
F … do not display block-wise output
T … display brief summary of block run

thrRen threshold of electron–nucleus distance (default = 0.0D0)
r … reject move if electron–nucleus distance is under r

CalcPulay flag to calculate Pulay term in force evaluation (default = .FALSE.)
F … only evaluate Hellmann–Feynman term
T … evaluate Pulay term

ERef reference energy (default = 0.0d0)
0.0d0 … reference energy is evaluated in the beginning of DMC run
Given … reference energy is fixed at the given value

CalcSO flag to enable complex arithmetics for spin–orbit calculation (default = .TRUE. in SOVMC & SOOPT, = .FALSE. otherwise)
F …

T …

NoSpinMove flag to prohibit spin variable move (DEBUG: default = .FALSE.)
F … move spin in electron move
T … fix spin variable in electron move

DoSigma flag to compute spin derivative of wave function
F …
T … compute spin derivative (required for spin–orbit term evaluation)

UseLogJF flag to handle Jastrow factor value in the log format (default = .TRUE.)
F …
T …

DoubleFix flag to use sample points of previous step in the optimization
F … in the *i*-th optimization step, fixed samples are generated thoroughly with the *i*-th parameters
T … samples generated with the (*i* – 1)-th parameter is also used in the *i*-th optimization step

CARD QMCRun

card to control QMC run

```
QMCRun
        {Run condition section}
$DataSet
        {DataSet section}
End
```

{Run condition section} :
give any of following members (if skipped, default set is used)
each member should be written in 1 line

MoveElec {arg}

control how many electrons to move in 1 step, one-by-one (OBO) or all-at-once (AAO)
{arg} is composed of 1, 2, or 4 elements, each of which is either OBO or AAO
if 3 elements are given, 3rd element is neglected and recognized as 2 elements argument

Number of elements and their recognition (MoveElec, MoveType, NBlock, NStep, TimeStep)

# of elements	VMC/EQU	VMC/RUN	DMC/EQU	DMC/RUN
1	element-1	element-1	element-1	element-1
2	element-1	element-2	element-1	element-2
4	element-1	element-2	element-3	element-4

MoveType {arg}

control the method to propose next electron position
{arg} is composed of 1, 2, or 4 elements, each of which is RandBox, RandGauss, or RandFP

RandBox	…	move to any position in cubic box with uniform probability
RandGauss	…	move to any position with Gaussian distribution probability
RandFP	…	use Fokker–Planck equation to move electrons

NBlock {arg}

number of blocks in each calculation stage, {arg} is composed of 1, 2, or 4 positive integers

NStep {arg}

number of steps in each block, {arg} is composed of 1, 2, or 4 positive integers

TimeStep {arg}

time step in each calculation stage, {arg} is composed of 1, 2, or 4 positive real numbers

IMainWF {arg}

control which wave function is used as guide function, {arg} is 1 integer

UseFN (0)

flag to use finite nuclei, if 0 is given, explicitly turn off the flag

UsePP (0)

flag to use pseudo potential, if 0 is given, explicitly turn off the flag

UseMP (0)

flag to use model potential, if 0 is given, explicitly turn off the flag

UseProj (0)
flag to use projection operator in model potential, if 0 is given, explicitly turn off the flag

UseJF (0)
flag to use Jastrow factor, if 0 is given, explicitly turn off the flag

BSum (0)
flag to use balanced sum in data evaluation, if 0 is given, explicitly turn off the flag

Init {arg}
flag to control walker initialization method, {arg} is any of following integers
0 … distribute electrons randomly
1 … distribute electrons to keep ion-neutrality
2 … read electron positions from *.QMCXin (or *.QMCSXin) file

VScan {arg}
control scan vector, {arg} is three real numbers

{DataSet section} :

WFValue
E(NR) … nonrelativistic energy
E(ZORAN) … ZORA energy (only nuclear potential used in kinetic term)
E(ZORANSO)
Weight
RR
R(e_N)
MinR(e_N)
MaxR(e_N)
R(e_e)
MinR(e_N)
MaxR(e_N)
SForce
Force
NucDeriv
ONE
Add
Mult
Sz

CARD QMCCusp

Card to control cusp correction information

```
QMCCusp
  Range ${Range}
  MO  ${iMO}  Nuc ${iNuc1} (${iNuc2} …)
End
```

Range [REAL(8)]	cusp correction radius
iMO [INTEGER]	MO to enforce cusp correction
iNuc [INTEGER]	Nuclei index

CARD QMCOpt

Card to control parameter optimization

```
QMCOpt
  Same ${iNuc1} ${iNuc2}
  eeCusp       (0)
  fixScale     (0)
  Obj          ${Obj}
  VRatio
  SpinDep ${iSD}
End
```

CARD QMCJastrow

Card to control Jastrow correlation factor

```
QMCJastrow
  ${#elec_A} ${#elec_B} ${iNucSet}
  AA   ${Scale_AA}
  AB   ${Scale_AB}
  BB   ${Scale_BB}
  ${L1} ${M1} ${N1}   ${Coef_AA}  (${Coef_AB} (${Coef_BB}))
  ${L2} ${M2} ${N2}   ${Coef_AA}  (${Coef_AB} (${Coef_BB}))
  EN   ${iNuc}   ${Scale_eN}
  ${L3} ${M3} ${N3}   ${Coef_AN}  (${Coef_BN})                    … e-N term
  ${L4} ${M4} ${N4}   ${Coef_AAN}  (${Coef_ABN} (${Coef_BBN}))… e-e-N term
End
```

Jastrow factor used in R4QMC has the following form,

$$J = \exp\left(S^{\text{e-e}} + S^{\text{e-N}} + S^{\text{e-e-N}}\right).$$

Here, $S^{\text{e-e}}$ is the electron-electron (e-e) term,

$$S^{\text{e-e}} = \sum_{I=1}^{L^{\text{e-e}}} s_I^{\text{e-e}} \,,$$

$$s_I^{\text{e-e}} = \sum_{i<j}^{N_\alpha} a_{I,\alpha\alpha}^{\text{e-e}} \bar{r}_{ij}^{n_I} + \sum_{i}^{N_\alpha}\sum_{j}^{N_\beta} a_{I,\alpha\beta}^{\text{e-e}} \bar{r}_{ij}^{n_I} + \sum_{i<j}^{N_\beta} a_{I,\beta\beta}^{\text{e-e}} \bar{r}_{ij}^{n_I} \,, \qquad \ldots \text{(A1)}$$

$S^{\text{e-N}}$ is the electron-nucleus (e-N) term,

$$S^{\text{e-N}} = \sum_{A}^{M}\sum_{J=1}^{L_A^{\text{e-N}}} s_J^{\text{e-N},A} \,,$$

$$s_J^{\text{e-N},A} = \sum_{i}^{N_\alpha} a_{J,\alpha}^{\text{e-N},A} \bar{r}_{iA}^{l_J} + \sum_{i}^{N_\beta} a_{J,\beta}^{\text{e-N},A} \bar{r}_{iA}^{l_J} \,, \qquad \ldots \text{(A2)}$$

and $S^{\text{e-e-N}}$ is the electron-electron-nucleus (e-e-N) term,

$$S^{\text{e-e-N}} = \sum_{A}^{M}\sum_{K=1}^{L_A^{\text{e-e-N}}} s_K^{\text{e-e-N},A} \,,$$

$$\begin{aligned} s_K^{\text{e-e-N},A} &= \sum_{i<j}^{N_\alpha}\left(1-\delta_{l_K,m_K}/2\right) a_{K,\alpha\alpha}^{\text{e-e-N},A}\left(\bar{r}_{iA}^{l_K}\bar{r}_{jA}^{m_K} + \bar{r}_{iA}^{m_K}\bar{r}_{jA}^{l_K}\right)\bar{r}_{ij}^{n_K} \\ &\quad + \sum_{i<j}^{N_\beta}\left(1-\delta_{l_K,m_K}/2\right) a_{K,\beta\beta}^{\text{e-e-N},A}\left(\bar{r}_{iA}^{l_K}\bar{r}_{jA}^{m_K} + \bar{r}_{iA}^{m_K}\bar{r}_{jA}^{l_K}\right)\bar{r}_{ij}^{n_K} \\ &\quad + \sum_{i}^{N_\alpha}\sum_{j}^{N_\beta}\left(1-\delta_{l_K,m_K}/2\right) a_{K,\alpha\beta}^{\text{e-e-N},A}\left(\bar{r}_{iA}^{l_K}\bar{r}_{jA}^{m_K} + \bar{r}_{iA}^{m_K}\bar{r}_{jA}^{l_K}\right)\bar{r}_{ij}^{n_K} \,, \qquad \ldots \text{(A3)} \end{aligned}$$

where the scaled distance $\bar{r} = r/(1+br)$ and b is the scale parameter required in the QMCJastrow card. All $s_I^{\text{e-e}}$, $s_J^{\text{e-N},A}$, and $s_K^{\text{e-e-N},A}$ terms can be generalized with modified coefficients a as following,

$$s=\sum_{i<j}^{N_\alpha}\left(1-\delta_{l,m}/2\right)a_{\alpha\alpha}\left(\overline{r}_{iA}^{l}\overline{r}_{jA}^{m}+\overline{r}_{iA}^{m}\overline{r}_{jA}^{l}\right)\overline{r}_{ij}^{n}+\sum_{i<j}^{N_\beta}\left(1-\delta_{l,m}/2\right)a_{\beta\beta}\left(\overline{r}_{iA}^{l}\overline{r}_{jA}^{m}+\overline{r}_{iA}^{m}\overline{r}_{jA}^{l}\right)\overline{r}_{ij}^{n}$$
$$+\sum_{i}^{N_\alpha}\sum_{j}^{N_\beta}\left(1-\delta_{l,m}/2\right)a_{\alpha\beta}\left(\overline{r}_{iA}^{l}\overline{r}_{jA}^{m}+\overline{r}_{iA}^{m}\overline{r}_{jA}^{l}\right)\overline{r}_{ij}^{n}\ . \qquad \ldots \text{(A4)}$$

The generalized term s is characterized by nucleus index A and three integers $\left(l,m,n\right)$.

The e-e term is the $\left(l=0,m=0,n\neq 0\right)$ case, where

$$s=\sum_{i<j}^{N_\alpha}a_{\alpha\alpha}\overline{r}_{ij}^{n}+\sum_{i<j}^{N_\beta}a_{\beta\beta}\overline{r}_{ij}^{n}+\sum_{i}^{N_\alpha}\sum_{j}^{N_\beta}a_{\alpha\beta}\overline{r}_{ij}^{n}\ ,$$

and the coefficient a in eq.(A4) is the same as the $a^{\text{e-e}}$ in eq.(A1).

The e-N term is the $\left(l\neq 0,m=0,n=0\right)$ case, where

$$s=\sum_{i<j}^{N_\alpha}a_{\alpha\alpha}\left(\overline{r}_{iA}^{l}+\overline{r}_{jA}^{l}\right)+\sum_{i<j}^{N_\beta}a_{\beta\beta}\left(\overline{r}_{iA}^{l}+\overline{r}_{jA}^{l}\right)+\sum_{i}^{N_\alpha}\sum_{j}^{N_\beta}a_{\alpha\beta}\left(\overline{r}_{iA}^{l}+\overline{r}_{jA}^{l}\right)$$
$$=\left(N_\alpha-1\right)\sum_{i}^{N_\alpha}a_{\alpha\alpha}\overline{r}_{iA}^{l}+\left(N_\beta-1\right)\sum_{i}^{N_\beta}a_{\beta\beta}\overline{r}_{iA}^{l}+N_\beta\sum_{i}^{N_\alpha}a_{\alpha\beta}\overline{r}_{iA}^{l}+N_\alpha\sum_{i}^{N_\beta}a_{\alpha\beta}\overline{r}_{iA}^{l}\ .$$

If the coefficients $a_{\alpha\alpha}$, $a_{\beta\beta}$, and $a_{\alpha\beta}$ have the same values, the equation becomes,

$$s=\left(N-1\right)\sum_{i<j}^{N}a\overline{r}_{iA}^{l}\ .$$

$\left(l,m,n\right)$ in the QMCJastrow card is this three integers, and the coefficients are $a_{\sigma\sigma'}^{\text{e-e}}$, $a_{\sigma}^{\text{e-N}}$, and $a_{\sigma\sigma'}^{\text{e-e-N}}$ in eqs. (A1-A3).

Module ONIOMPrep

Function

This module creates Geom file of each calculation level.

Required NAMELIST

&Control
&ONIOM

Prepared input data files

Name.Geom
Name.ONIOM

Created output data files

NameLR.Geom
NameHM.Geom
NameLM.Geom

Module ONIOM

Function

This module calculates ONIOM energy and its gradient.

Required NAMELIST

&Control
&ONIOM

Prepared input data files

Name.ONIOM
NameLR.TotEne
NameLR.Grad
NameHM.TotEne
NameHM.Grad
NameLM.TotEne
NameLM.Grad

Created output data files

Name.TotEne
Name.Grad

NAMELIST &ONIOM

Parameters

NameLR prefix name which is used for Low-Real level calculation (default = ‘NameLR’)

NameHM prefix name which is used for High-Model level calculation (default =’NameHM’)

NameLM prefix name which is used for Low-Model level calculation (default =’NameLM’)

Module QMMMPrep

Function

This module creates Geom file of each calculation level according to main Geom file.

Required NAMELIST

&Control
&QMMM

Prepared input data files

Name.ONIOM
Name.Geom

Created output data files

NameQM.Geom
NameMM.Geom
NameQM.Charge

Module QMMM

Function

This module calculates the QM/MM energy and its gradient.

Required NAMELIST

&Control
&QMMM

Prepared input data files

Name.ONIOM (Optional)
Name.Geom
NameQM.Geom
NameQM.TotEne
NameQM.Grad
NameQM.ChGrad
NameQM.MMInfo
NameMM.Geom
NameMM.TotEne
NameMM.Grad
NameMM.MMInfo

Created output data files

Name.TotEne
Name.Grad

NAMELIST &QMMM

Parameters

Param name of molecular mechanics parameter file (default = ’ ‘)

NameQM prefix name which is used for quantum mechanical calculation (default = ’NameQM’)

NameQM prefix name which is used for molecular mechanics calculation (default = ’NameMM’)

Module MFEP

Function

This module calculates approximate Gibbs free energy using QM/MM MFEP scheme. Two types of run, reference generation and sample evaluation, are available.

Required NAMELIST

&Control
&MFEP

[In reference generation mode (Reference=T)]

Prepared input data files

Name.Geom
[Tinker input file]
[Tinker parameter file]
[Tinker archive file]
Name.Ref.MFEP_Info … Information file of QM/MM attribution and MD condition
Name.Ref.Charge … Averaged solvent charge
Name.Ref.QMCHG … Fitted charge of QM atoms

Created output data files

Name.Ref.MFEPDelta
Name.Ref.MFEPE0MM

[In sample evaluation]

Prepared input data files

Name.Geom
Name.Sub.inp
Name.Sub.Geom
Name.Sub.Basis
[Tinker input file]
[Tinker parameter file]
[Tinker archive file]
Name.Ref.MFEP_Info
Name.Ref.Charge
Name.Ref.QMCHG
Name.Ref.MFEPDelta
Name.Ref.MFEPE0MM
Name.TotEne (if DoGlobal = F)
Name.Grad (if DoGlobal = F)

Created output data files

Name.TotEne
Name.Grad
Name.TSEne
Name.TSGrad

NAMELIST &MFEP

Parameters

NameSub base name of sub (QM only) input files (default = Name.Sub)

NameRef base name of reference input files (default = Name.Ref)

Reference flag to perform reference generation run (default = .FALSE.)
F … Run as a sample evaluation mode
T … Run as a reference generation mode

MDInput name of Tinker input file

MDParam name of Tinker parameter file

DoGlobal flag to perform whole system enthalpy calculation (default = .TRUE.)
F … do whole system enthalpy calculation in other module
T … do whole system enthalpy calculation in MFEP module

IPrint print option (default = 0)
0 … default printing
1 … additional information is printed out

Module DLFind

Function

Required NAMELIST

&Control
&DLFind

Prepared input data files

Name.Geom

Created output data files

Name.Geom
Name.Hessian
Name.NormMod

NAMELIST &DLFind

Parameters

Tolerance main convergence criterion (max grad component) (default = 0.00045)

Tolerance_E convergence criterion on energy change (default = Tolerance / 450)

Printl how verbosely to write info to stdout (default = 4)
0 … no printout
2 … print something
4 … be verbose
6 … debug

Printf how verbosely files should be written (default = 2)

MaxCycle maximum number of cycles (default = 100)

MaxEne maximum number of energy and gradient evaluations (default = 100000)

ICoord type of internal coordinates (default = 3)
Unit place 0… Cartesians
Unit place 1… HDLC - internals
Unit place 2… HDLC - TC
Unit place 3… DLC - internals
Unit place 4… DLC - TC
10X … NEB with endpoints free
11X … NEB with endpoints moving only perpendicular to their tangent direction
12X … NEB with frozen endpoints
13X … NEB with endpoints free; only initialization in coordinates X, optimization in Cartesians
14X … NEB with endpoints moving only perpendicular to their tangent direction; only initialization in coordinates X, optimization in Cartesians
15X … NEB with frozen endpoints; only initialization in coordinates X, optimization in Cartesians
20X … dimer method; translation and rotation of the dimer are covered by the optimizer specified trough IOpt; requires two energy evaluation per iteration
21X … dimer method; rotation of the dimer is done by a line search within the dimer module, two energy calculations are used per rotation; requires at least two energy evaluation per iteration
22X … dimer method; rotation of the dimer is done by a line search within the dimer module, one energy calculation is done per iteration, the other one is interpolated; requires at least two energy evaluation per iteration

MassWeight use mass-weighted coordinates (default = .FALSE.)

IOpt type of optimization algorithm
0 … steepest descent
2 … conjugate gradient following Polak–Ribie`re with CG restart every 10 steps (hardcoded at the moment)
3 … LBFGS
10 … P-RFO

11 … just calculate the Hessian and do a thermal analysis
20 … Newton–Raphson/quasi-Newton
30 … damped dynamics using the variables TimeStep, Fric0, FricFac, and Fricp; the frictions are defined that 0 corresponds to free (undamped) dynamics, and 1 corresponds to steepest descent

LBFGS_Mem number of steps in LBFGS memory (default = 3*number of atoms, 5 <= LBFGS_Mem <= 50)

ILine type of line search or trust radius (default = 1)
0 … simple scaling of the proposed step, taking MaxStep into account
1 … trust radius based on energy as acceptance criterion (recommended for L-BFGS optimization)
2 …
3 … trust radius based on gradient as acceptance criterion (recommended for CG optimization)

MaxStep maximum length of the step in internals (default = 0.5)

ScaleStep constant factor with which to scale the step (default = 1.0)

NImage number of images in NEB calculation (default = 10)

NEBk force constant for NEB calculation (default = 0.01)

NEB_Climb_Test threshold scale factor for spawning climbing image (default = 3.0)

NEB_Freeze_Test threshold scale factor for freezing NEB images (default = 1.0)

Delta delta –x in finite-difference Hessian (default=0.01)

Update Hessian update scheme
0 … no update; always recalculate the Hessian (default = 2)
1 … Powell update
2 … Bofill update
3 … BFGS update

MaxUpd maximum number of Hessian updates (default = 50)

InitHessian option for method of calculating the initial Hessian (default = 2)
1 … build by one point finite difference of the gradient
2 … build by two point finite difference of the gradient
3 … build a diagonal Hessian with a single one point finite difference
4 … set the Hessian to be an identity matrix

MinStep Hessian is not updated if step < MinStep (default = 0.00001)

Fric0 start friction for dumped dynamics (default = 0.3)

FricFac factor to reduce friction whenever the energy is decreasing (default = 0.95)

FricP friction to use whenever energy increasing (default = 0.3)

Dump after how many energy and gradient calculations to dump a checkpoint file? (default = 0)

Restart restart mode (default = 0)

Temperature temperature for thermal analysis (default = 300.0)

Soft when the absolute value of eigenvalue of Hessian is less than Soft, ignore this mode in P-RFO (default = 0.005)

OtherGeom prefix name for last geometry of NEB calculation (default = “ “)

CoordType NTChem original mnemonic for coordinate type (default = “ “)
“INTERNAL”… use DLC – internals; set unit place of ICoord to 3
“XYZ” … use Cartesian coordinate; set unit place of ICoord to 0

Opt NTChem original mnemonic for optimizer
“SD” … use Steepest descent; set IOpt = 0
“CG” … use Conjugate gradient; set IOpt = 2
“LBFGS”… use L-BFGS; set IOpt = 3
“TS” … use P-RFO; set IOpt = 10
“NR” … use Newton–Raphson/quasi-Newton; set IOpt = 20
“DUMP”… use Damped dynamics; set IOpt = 30
“FREQ”… carry out frequency analysis; set IOpt = 11

NEB flag to indicate whether NEB is performed
F … do not carry out NEB calculation
T … carry out NEB calculation; add 150 to ICoord

Dimer flag to indicate whether dimer method is performed
F … do not use dimer method
T … use dimer method; add 200 to ICoord

NEBPath file name which include NEB initial images (default = “ “)

NInitImg number of images in NEBPath

ParaNEB flag to indicate whether energy evaluation is performed at the same time in NEB
F … evaluate a energy of individual images one by one
T … evaluate energy of individual images at the same time on different processors

Module NEB

Function

This module controls the nudged elastic band (NEB) calculation.

Required NAMELIST

&Control
&NEB

Prepared input data files

Name.Basis
Name.Geom
Name.HCore
Name.Overlap
Name.NucRepl
Name.Charge (optionally)

Created output data files

NAMELIST &NEB

Parameters

NImage maximum number of images (default = 5)

MaxIter maximum number of NEB optimizations (default = 1000)

Restart flag to restart the NEB calculation (default = .FALSE.)
F … do not restart NEB
T … restart NEB
NImage Name.Geom.*n* ($1 \leq n \leq$ NImage) are required if Restart = .TRUE.

String flag to use the string method (default = .FALSE.)
F … do not use the string scheme
T … use the string scheme

ConSpr spring constant in atomic units (default = 0.02); enforce ConSpr = 0.0 in case of String = T

Climb flag to use a climbing image formulation (default = .FALSE.)
F … no climbing image
T … climbing image

Refine_Edge flag to refine endpoints of the NEB path (default = .FALSE.)
F … frozen endpoints
T … endpoints free

OneByOne

Update type of optimization algorithm (default = 'DampMD')
'DampMD'… damped molecular dynamics
'Verlet' … Verlet algorithm
'CG' … conjugate gradient
'LBFGS'… L-BFGS

ThrGMax convergence criterion for the maximum absolute gradient in atomic units (default = 0.00045)
This controls the following two criteria: (1) maximum absolute gradient, ThrGMax, (2) root-mean-square of gradients, (2/3)*ThrGMax; criteria specified by ThrXMax should also be fulfilled

ThrXMax convergence criterion for the maximum absolute component of the step vector (default = 0.0018)
This controls the following two criteria: (1) maximum absolute component of the step vector, ThrXMax, (2) root-mean-square of the step vector, (2/3)*ThrXMax; criteria specified by ThrGMax should also be fulfilled

FacDMD damping factor in damped MD (default = 0.3)

DynDMD flag to carry out a dynamic damping in damped MD (default = .TRUE.)

F … do not use the dynamic damping so that a damping facter is constant in NEB
T … use the dynamic damping

NResetCG the conjugate gradient scheme is reset every NResetCG times (default = 20)

IO_NEB unit number of the NEB output file (default = 6)

Module AIMD

Function

This module calculates the classical trajectory based on any ab initio calculation level. Some sampling schemes for initial condition (coordinate and velocity) are also implemented.

Required NAMELIST

&Control
&AIMD

Prepared input data files

Name.Geom
Name.NormMod
Name.TotEne
Name.Grad
Name.NACP (NYI)
Name.StateNum (NYI)

Created output data files

Name.AIMD

NAMELIST &AIMD

Parameters

TimeStep time step for molecular dynamics in atomic unit (default = 8.26)

NStep number of molecular dynamics step (default = 100)

Restart flag to indicate whether restart molecular dynamics or not (default = .FALSE.)
F … do not restart molecular dynamics
T … read AIMD file and restart molecular dynamics

VelUnit unit of input velocity in Velo card (default = ‘AU’)
AU … atomic unit
AngFs … angstrom per femto-second

TUnit unit of input time step (default = ‘AU’)
AU … atomic unit
FS … femto second

OutUnit unit of output (default = ‘AU’)
AU … atomic unit
Else … use angstrom for length, femto-second for time, kcal/mol for energy

Kelvin … simulation temperature which is used in NVT ensemble and initial condition sampling (default = 300)

Ensemble ensemble for molecular dynamics (default = ’NVE’)
‘NVE’ … microcanonical ensemble
‘NVT’ … canonical ensemble (use Nosé–Hoover chain thermostat)

NTempSample number of steps which is used to evaluate the averaged temperature (default = 10)

SurfHop flag to use the surface hopping (Does not work)
F … do not use surface hopping
T … use the surface hopping

StateNum electronic state (default = 1)

NState number of electronic state in calculation

Name1 prefix name of NormMod file of molecule1 (default = ’’)

Name2 prefix name of NormMod file of molecule2 (default = ’’)

NAtom1 Number of atoms of molecule1 (default = NAtom)

NAtom1 Number of atoms of molecule2 (default = 0)

VibSample computational type for vibrational sampling (default = ‘None’)
‘None’ … do not sample vibrational motion

‘Classical’… classical sampling
‘Quantum’… Wigner sampling

RotSapmle flag to sample rotational motion
F … do no sample rotational motion
T … sample rotational motion

OriSample flag to sample orientation
F … do no sample orientation
T … sample orientation

PosSample flag to sample relative position of molecule1 and molecule2
F … do no sample relative position
T … sample orientation relative position

TransSample flag to sample relative translational motion of molecule1 and molecule2
F … do no sample translational energy
T … sample translational energy

ThermSample flag to use random thermal sampling
F … do not use random thermal sampling
T … use random thermal sampling

Distance distance between molecule 1 and molecule2 in Bohr (default: two times the sum of the diameters of the molecule1 and molecule2)

ImpParam impact parameter (default: randomly determined)

ETrans relative translational energy of molecule1 and molecule2 (default: randomly determined)

RATTLE flag to use RATTLE algorithm to integrate the Newton's equation of motion (default = .FALSE.)
F ... do not use RATTLE algorithm (usual velocity Verlet algorithm)
T ... use RATTLE algorithm

ThrRAT Convergence threshold for RATTLE (default = 1.0D-7)

MaxRAT Maximum number of iteration of RATTLE (default = 1000)

RandSeed seed of random number (default = 0: use system clock)

IPrint print flag
1 … simple print
2 … normal print
3 … debug print

Module Pop

Function

This module carries out the population analysis.

Required NAMELIST

&Control
&Pop

Prepared input data files

Name.Overlap
Name.LMO
Name.MO
Name.Basis
Name.OrbEne
Name.OrbEneSO
Name.MOSO
Name.Dens
Name.DensSO
Name.SCF_Info
Name.SOSCF_Info

Created output data files

NAMELIST &Pop

Parameters

SOrbit flag to use the density matrix which include spin–orbit interaction (default = .FALSE.)
F … use the density matrix which does not include spin–orbit interaction
T … use the density matrix which include spin–orbit interaction

PrintMO flag to indicate whether print the molecular orbital coefficients (default = .FALSE.)
F … do not print molecular orbital coefficient
T … print molecular orbital coefficient

Mulliken flag to carry out a Mulliken population analysis (default = .TRUE.)
F … do not carry out Mulliken population analysis
T … carry out Mulliken population analysis

Lowdin flag to carry out a Löwdin population analysis (default = .TRUE.)
F … do not carry out Löwdin population analysis
T … carry out Löwdin population analysis

FragPop (default = .FALSE.)

ThrPrFrag (default = 0.02D+00)

IPrint print flag (default = 1)

Module Prop

Function

This module calculates the dipole moment as well as the transition dipole moment. The origin – not the center of mass – is referred in the calculation of moments.

Required NAMELIST

&Control
&Prop

Prepared input data files

Created output data files

NAMELIST &Prop

Parameters

SOrbit flag to use the density matrix which include spin–orbit interaction (default = .FALSE.)
F … use the density matrix which does not include spin–orbit interaction
T … use the density matrix which include spin–orbit interaction

Dipole flag to calculate dipole moment (default = .TRUE.)
F … do not calculate dipole moment
T … calculate dipole moment

Quadrupole (default = .FALSE.)

TranDip flag to calculate transition dipole moment (default = .FALSE.)
F … do not calculate transition dipole moment
T … calculate transition dipole moment

ThrOsc (default = 1.D-3)

IPrint print flag (default = 1)

Module MagProp

Function

This module controls the calculation of Hartree–Fock (HF) and Kohn–Sham (KS) density functional theory (DFT) energies. Closed shell and spin unrestricted HF and KS-DFT energies can be calculated as well as open shell restricted (pseudo-canonical) energies.

Required NAMELIST

&Control
&MagProp

NAMELIST &MagProp

Parameters

Module ESP

Function

This module evaluates electrostatic potential (ESP) and performs Merz–Kollman analysis. Evaluated ESP is printed out in {$NAME}_ESP.cube file.

Required NAMELIST

&Control
&ESP

Input files

Name.Basis	
Name.Geom	
Name.SCF_Info	
Name.Dens	(if SOrbit = F)
Name.DensSO	(if SOrbit = T)

Output files

{Name}.ESP.cube

NAMELIST &ESP

Parameters

DoCube flag to punch out ESP cube file (default = .TRUE.)
F … only do MK analysis
T … punch out ESP cube file and do MK analysis

SOrbit flag to use density calculated with two-component calculations (default = .FALSE.)
F … use 1-component density file (Name.Dens)
T … use 2-component density file (Name.DensSO)

UHF flag to indicate whether UHF/UKS density is used (default = .FALSE.)
F … density file has been generated in RHF or RKS calculation
T … density file has been generated in UHF or UKS calculation

NOccA number of electrons for alpha orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

NOccB number of electrons for beta orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

GridInterval interval of grid nodes in atomic unit (default = 2.0D0)

ESPICutPre integer to control exponent cutoff in one-electron integral evaluation (default = 13)

ESPICutWF integer to control grid range (default = 9)

ESPPotCut real to control potential cutoff in one-electron integral evaluation (default = 1.0D-99)

Module CubeGen

Function

This module generates cube files of density (Total/Alpha/Beta/Spin) and MOs (or spinor density if SOrbit = T).

Required NAMELIST

&Control
&CubeGen

Input files

Name.Basis
Name.Geom
Name.SCF_Info
Name.Dens (if SOrbit = F)
Name.DensSO (if SOrbit = T)

Output files

Name.{spin}Dens.cube	(Name_ADens.cube / Name_BDens.cube)	
Name.TotDens.cube		
Name.SDens.cube		
Name.{spin}{#MO}.cube	(ex. Name_B3.cube)	
Name.{spin}{#MO}{xyz}.cube	(ex. Name_A12x.cube)	… Hidden option
Name.MOSO{#spinor}.cube	(ex. Name_MOSO1.cube)	

NAMELIST &CubeGen

Parameters

DoDen flag to generate density cube file (default = .TRUE.)
F … do not generate density cubes
T … generate density cubes

DoD1MO flag to generate MO 1st derivative cube file (default = .FALSE. / Hidden option)
F … do not generate MO 1st derivative cubes
T … generate MO 1st derivative cubes

SOrbit flag to use density calculated with 2-component calculations (default = .FALSE.)
F … use 1-component density file (Name.Dens)
T … use 2-component density file (Name.DensSO)

UHF flag to indicate whether UHF/UKS density is used (default = .FALSE.)
F … density file has been generated in RHF or RKS calculation
T … density file has been generated in UHF or UKS calculation

DoCut flag to skip small primitive evaluation (default = .TRUE.)
F … evaluate all primitives at all grid point
T … skip small primitive evaluation

NOccA number of electrons for alpha orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

NOccB number of electrons for beta orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

GridType method to generate grid (default = “Auto”)
“Auto” … automatically determine the grid
“MinMax”… use input lower and upper bound of grid

Grid keyword to indicate quality of cube grid
“Coarse” … use 1/3 bohr grid
“Fine” … use 1/6 bohr grid
“Ultrafine” … use 1/12 bohr grid

GridInterval interval of grid nodes in atomic unit (default = 1/3) ignored if Grid keyword is specified

CubeICutPrim integer to control exponent cutoff in one-electron integral evaluation (default = 10)

CubeICutWF integer to control grid range (default = 4)

[Following entries are required to generate alpha MO cube]
ALMO lowest alpha MO for MO-cube generation (default = 0)
AUMO highest alpha MO for MO-cube generation (default = 0)

[Following entries are required to generate beta MO cube]
BLMO lowest beta MO for MO-cube generation (default = 0)

BUMO	highest beta MO for MO-cube generation (default = 0)

[Following entries are required to generate spinor-density cube]

LMOSO	lowest spinor for spinor-density-cube generation (default = 0)
UMOSO	highest spinor for spinor-density-cube generation (default = 0)

[Following entries are required if GridType = “MinMax”]

XMinGrid	lower bound of grid point in x direction in a.u.
XMaxGrid	upper bound of grid point in x direction in a.u.
YMinGrid	lower bound of grid point in y direction in a.u.
YMaxGrid	upper bound of grid point in y direction in a.u.
ZMinGrid	lower bound of grid point in z direction in a.u.
ZMaxGrid	upper bound of grid point in z direction in a.u.

Comments

Q1. How to obtain density difference?

A1. In order to compute density difference between system A and system B

1. Calculate density cube of system A and check grid range from output
2. Calculate density cube of system B with GridType = ”MinMax”, and set grid range and GridInterval same as which used in the calculation of system A
3. Subtract density cube of B from density cube of A (manually)

Q2. How to generate cubes for several MOs? E.g. cubes for MO 1, 4, 6, 11.

A2. Use utility program “sortmo” to rearrange MOs and set ALMO = 1, AUMO = 4 (# of required MOs)

Q3. How to speed up cube generation?

A3. Try following options

1. Use larger GridInterval
2. Use smaller CubeICutPrim
3. Use smaller CubeICutWF

Module NT2FChk

Function

This module creates a Gaussian formatted checkpoint file.

Required NAMELIST

&Control

Prepared input data files

Name.Geom
Name.Basis
Name.MO
Name.Dens
Name.Hessian
Name.SCF_Info

Created output data files

Name.FChk

Module GenGFC

Function

This module generates auxiliary basis functions for the GFC method.

Required NAMELIST

&Control
&GenGFC

Prepared input data files

Name.Basis
Name.Dens
Name.EneCoul (Intermediate file generated by SCF module)

Created output data files

None (GFC basis is printed in the standard output)

NAMELIST &GenGFC

Parameters

UHF flag to indicate whether the spin unrestricted HF (UHF) or KS-DFT (UKS) method is used (default = .FALSE.)

NOccA number of electrons for alpha orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

NOccB number of electrons for beta orbitals (default = 0)
0 … automatically determine the number of electrons for the neutral molecule

NAnglAux angular quantum number of auxiliary basis function (default = 0)

TgtAngl target angular momentum (default = 0)

t0 (default = 1.0D-5)

h0 (default = 1.0D-1)

SVDThr (default = -1.0D+99)

ExpThr (default = -1.0D+99)

CnvThr (default = -1.0D+99)

prin (default = 1)

Alph (default = 1.0D-13)

Beta (default = 3.0D+0)

NPrimAux (default = (/0, 0, 0, 0, 0/))
(integer array 0:5)

MinPrimAux (default = 0)

MaxPrimAux (default = 0)

Module BasisRIJ

Function

This module generates auxiliary basis functions for the RIDFT method.

Required NAMELIST

&Control
&GenGFC

Prepared input data files

Name.Geom
Name.Basis

Created output data files

Module CopyGeom

Function

This module copies a NTChem Geom file from Master node to all I/O node in the parallel calculations.

Required NAMELIST

&Control

Prepared input data files

Name.Geom (master node)

Created output data files

Name.Geom (all I/O node other than master node)

Module Geom2xyz

Function

This module creates a XMol xyz file from a NTChem Geom file.

Usage

geom2xyz.exe {input}.Geom {output}.xyz

Required NAMELIST

None

Prepared input data files

{input}.Geom

Created output data files

{output}.xyz

Module RotGeom

Function

This module calculates a rotated Cartesian coordinate.

Usage

The users should prepare the input file before perform calculations by following file format.

```
IC1, IC2, IC3          # the centers 1 to 3
X1, Y1, Z1             # reference coordinate 1
X2, Y2, Z2             # reference coordinate 2
X3, Y3, Z3             # reference coordinate 3
NAtom                   # number of atoms
CO1X, CO1Y, CO1Z    # coordinate of atom 1
CO2X, CO2Y, CO2Z    # coordinate of atom 2
```

To calculate the rotated Cartesian coordinate, type

```
rotgeom.exe < {input_file}
```

where the input file is specified via the standard input The result is prints out onto the standard output and the users can store the result by redirecting the standard output to a text file..

Required NAMELIST

None

Prepared input data files

{input_file}

Created output data files

None

Module Cube

Function

This module calculates the sum or difference of cube file.

Usage

In all cases the grid should be consistent in all cube files. Intermediate file named, "work.cube1" or "work.cube2" will be generated. To calculate the difference of two cube files A.cube and B.cube, type

```
 ruby cube.rb A.cube - B.cube > result.cube
```

To calculate the sum of two cube files A.cube and B.cube, type

```
ruby cube.rb A.cube + B.cube > result.cube
```

More than two cube files also can be treated as follows

```
ruby cube.rb A.cube - B.cube - C.cube > result.cube
```

The result is prints out onto the standard output in cube format. The users can store the result by redirecting the standard output to a text file.

Required NAMELIST

None

Prepared input data files

{input}.cube

Created output data files

work.cube1
work.cube2

Module MO2Bin

Function

This program generates the binary MO file from the formatted MO file

Required NAMELIST

&Control
&MO2Bin

Prepared input data files

Name.MO_Txt (if SOrbit = .FALSE.)
Name.MOSO_Txt (if SOrbit = .TRUE.)

Created output data files

Name.MO (if SOrbit = .FALSE.)
Name.MOSO (if SOrbit = .TRUE.)

NAMELIST &MO2Bin

Parameters

NBF number of basis functions

NMO number of MOs

UHF flag to use the UHF MO_Txt file
F … use the RHF MO_Txt file
T … use the UHF MO_Txt file

SOrbit flag to use the MOSO_Txt file including spin-orbit interaction
F … use the MO_Txt file
T … use the MOSO_Txt file

Module MO2FChk

Function

This program converts the binary MO file into the formatted check point (.fchk) file using the template .fchk file

Usage [This program does not use NTChem style input file]

```
./mo2fchk.exe    ${Inp.fchk}       ${NTC.mo}       ${Out.fchk}
```

Required

${Inp.fchk}	… template .fchk file (Gaussian format)
${NTC.mo}	… MO file generated with NTChem
${Out.fchk}	… output .fchk file

Module MO2Txt

Function

This program generates the formatted MO file from the binary MO file

Required NAMELIST

&Control
&MO2Txt

Prepared input data files

Name.MO (if SOrbit = .FALSE.)
Name.MOSO (if SOrbit = .TRUE.)

Created output data files

Name.MO_Txt (if SOrbit = .FALSE.)
Name.MOSO_Txt (if SOrbit = .TRUE.)

NAMELIST &MO2Txt

Parameters

NBF number of basis functions

NMO number of MOs

UHF flag to use the UHF MO_Txt file
F … use the RHF MO_Txt file
T … use the UHF MO_Txt file

SOrbit flag to use the MOSO_Txt file including spin-orbit interaction
F … use the MO_Txt file
T … use the MOSO_Txt file

Module SortMO

Function

This program sorts MO ordering
If, SortMOA(1:3) = (/5, 9, 12/) is given in the namelist “&SortMO”, first three alpha MOs in the output MO file are the 5-th, 9-th, and 12-th alpha MOs in the original input MO file, and the other MOs are sorted in ascending order in the original ordering

Required NAMELIST

&Control
&SortMO

Prepared input data files

Name.MO
Name.MOSO

Created output data files (overwritten)

Name.MO
Name.MOSO

Namelist &SortMO

Parameters

NBF	number of basis functions
NMO	number of MOs
UHF	flag to use the UHF MOs
SOrbit	flag to use the MOSOs including spin-orbit interaction
SrtMOA	integer array to indicate alpha MO numbers sorted
SrtMOB	integer array to indicate beta MO numbers sorted
SrtMOSO	integer array to indicate MOSO numbers sorted

Module FChk2MO

Function

This module creates NTChem MO file and either OrbEne or OccNum files from a Gaussian fchk file.

Required NAMELIST

&Control
&FChk2MO

Prepared input data files

{input}.fchk

Created output data files

{input}.MO
{input}.OrbEne or {input}.OccNum

NAMELIST &FChk2MO

Parameters

UHF flag to indicate the molecular orbitals are spin-unrestricted (default = .FALSE.)
F … molecular orbitals are spin-unrestricted
T … molecular orbitals are spin-restricted

NatOrb flag to indicate that the orbital eigenvalues should be saved as occupation numbers, rather than orbital energies (default = .FALSE.)
F … eigenvalues are saved in .OrbEne file
T … eigenvalues are saved in .OccNum file

IPrint Print option (default = 0)
0 … default printing
>= 1 … print molecular orbital coefficients

Module FChk2Dens

Function

This module creates an NTChem Dens file from a Gaussian fchk file.

Usage

```
fchk2dens.exe {input}.fchk {output}.Dens
```

Required NAMELIST

None

Prepared input data files

{input}.fchk file

Created output data files

{output}.Dens

Module Dens2FChk

Function

This module creates

Usage

```
fchk2dens.exe {input1}.fchk {input2}.Dens {output}.fchk
```

Required NAMELIST

None

Prepared input data files

{input1}.fchk
{input2}.Dens

Created output data files

{output}.fchk