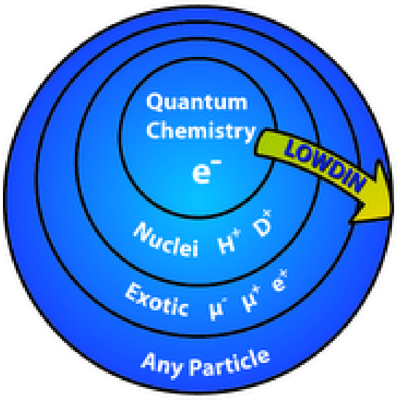
LOWDIN 2

User Manual
Version 2015.1



Quantum and Computational Chemistry Group
Chemistry Department
National University of Colombia

Bogotá, Colombia
December 11, 2018

Edited by:

Jorge Alfonso Charry Martínez
Félix Santiago Moncada Arias
Edwin Fernando Posada Correa
José Mauricio Rodas Rodríguez
Jhonathan Romero Fontalvo

Andrés Reyes Velasco

Contents

1 Program **3**
- 1.1 Installation . . . 3
- 1.2 Running . . . 4
- 1.3 Input . . . 4
- 1.4 Basis and database . . . 5
- 1.5 Outputs files . . . 7

2 Hartree-Fock Calculations **9**
- 2.1 RHF . . . 9
- 2.2 UHF . . . 13
- 2.3 HF Options . . . 13
- 2.4 Integrals options . . . 14
- 2.5 SCF options . . . 15
- 2.6 Reading a .vec file . . . 16
- 2.7 Auxiliary basis . . . 16

3 Moller-Plesset Calculations **19**
- 3.1 Example . . . 20
- 3.2 Options . . . 21

4 Propagator Calculations **23**

5 Configuration Interaction **27**

6 Outputs **31**
- 6.1 Molden and AIM files . . . 31
- 6.2 Gaussian Cubes . . . 31
- 6.3 Gnuplot files . . . 32
- 6.4 Example . . . 33

Introduction

LOWDIN is a computational program that implements the Any Particle Molecular Orbital (APMO) method. LOWDIN input file offers a unique flexibility, allowing users to exploit all the program's capabilities to study systems containing any type and number of quantum species.

The latest version of this manual can be found in this link:
`https://www.overleaf.com/read/yghcdxvgdmvy`

Chapter 1

Program

1.1 Installation

In order to install lowdin2 from the binary files please take the following steps

1. copy the lowdin2 folder to any directory.

   ```
   cp -r lowdin2/ /home/user/
   ```

2. Give ”read”, ”write” and ”execute” permissions to lowdin2 folder

   ```
   chmod 711 -R /home/user/lowdin2/
   ```

3. In the lowdin2/lowdinvars.sh file please add the current path of the lowdin2 folder and the path for the temporary files.

   ```
   LOWDIN_HOME="/home/user/lowdin2
   LOWDIN_SCRATCH="/scr/"
   ```

   make sure the directory for temporary files already exists with enough free disk space.

4. Copy the lowdin2 script to your /home/user/bin/ folder and give it permissions

   ```
   cp /home/user/lowdin2/lowdin2 /home/user/bin/
   chmod 711 /home/user/bin/lowdin2
   ```

   4.1. if you don’t have a bin folder in your home.

   ```
   mkdir /home/user/bin
   ```

   4.2. add the following line to your /home/user/.bashrc

   ```
   export PATH=$PATH:/home/user/bin/
   ```

   4.3. Refresh the .bashrc file

   ```
   source /home/user/.bashrc
   ```

5. Add the lowdin path in the /home/user/bin/lowdin2 script.

   ```
   VARS_PATH="/home/user/lowdin2/lowdinvars.sh"
   ```

6. Done.

1.2 Running

To run lowdin

```
lowdin2 -i inputname.lowdin
```

or

```
lowdin2 -i inputname.lowdin -p
```

if you want to see the output on the console screen. With the flag "-h" you can see the following list of available options

```
lowdin2 -h
Usage:
   $ lowdin -i file.lowdin [-t [all] [list] [file]] [-n number] [-v number] [-p] [-s] [-h]

   -i file.lowdin
      This is the input file name
   -n number
      This will set the number of OMP threads
   -t all
      This will run all the test files located on the test database.
   -t list
      This will list all the test files located on the test database.
   -t file
      This will run a specific test file which is located on test database.
   -v number
      This is the lowdin version that will be used
   -p
      This will print the output file to the standard output on the fly
   -w
      This will save the LOWDIN .wfn file
   -k
      This will keep the temporary files in the scratch directory after running the calculation
   -s
      This activate the singleton mode
   -h
      This will print this same message
```

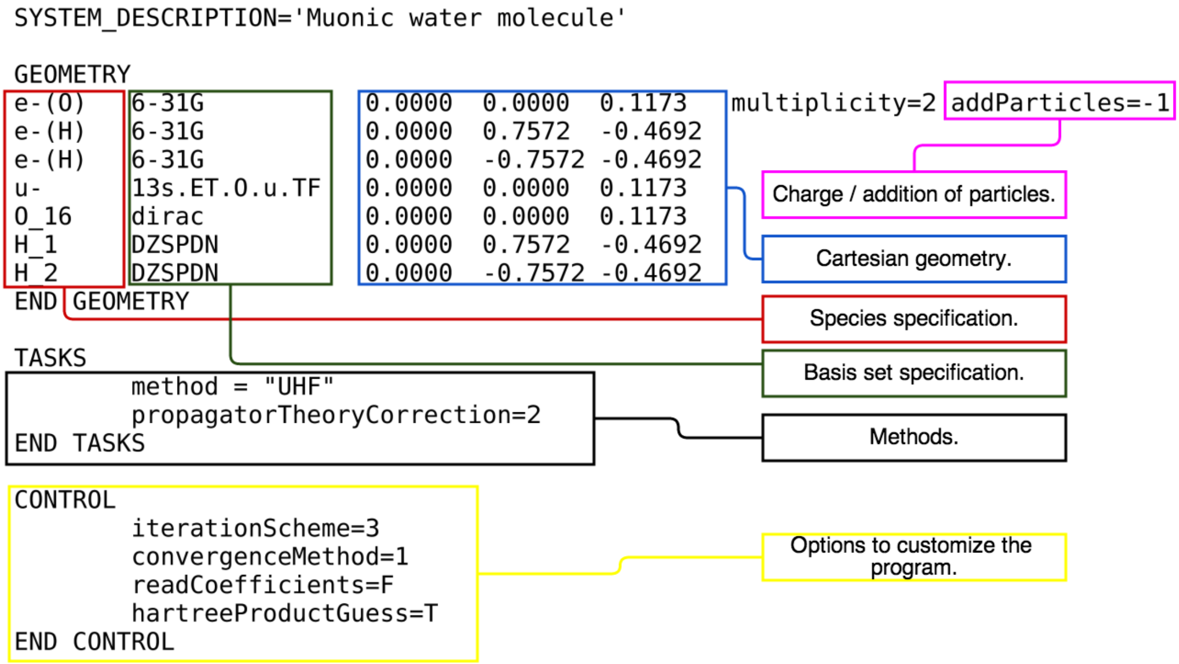
1.3 Input

Figure 1.1 presents an example of an input file. The minimum required blocks to run a calculation are `GEOMETRY`, `TASKS`, and `CONTROL`.

The `GEOMETRY` block provides the information needed to build the molecular system. The first column declares the type of the quantum species. As shown in Figure 1.1, `e-(H)` and `e-(O)` define the electrons of a Hydrogen and a Oxygen atom respectively; `u-` defines a negative muon, `O_16`, `H_1` and `H_2` define ^{16}O, ^{1}H and ^{2}H nuclei respectively.

The second column declares the basis sets. When the **'dirac'** basis is chosen, the particle is treated as a classical point charge. The third, fourth and fifth columns declare the x, y, z coordinates of the particle basis set center.

The sixth column provides additional information via keywords **addParticles** and **multiplicity**. These keywords are used to change the default values. **addParticles** is used to modify the number of particles of a quantum species. As shown in the provided example, one electron is removed from the system. **multiplicity** defines the multiplicity for open shell calculations. In the example, an electronic multiplicity of 2 was chosen. Additionally the charge and mass of quantum



```
SYSTEM_DESCRIPTION='Muonic water molecule'

GEOMETRY
e-(O)   6-31G          0.0000  0.0000   0.1173  multiplicity=2 addParticles=-1
e-(H)   6-31G          0.0000  0.7572  -0.4692
e-(H)   6-31G          0.0000  -0.7572 -0.4692
u-      13s.ET.0.u.TF  0.0000  0.0000   0.1173
O_16    dirac          0.0000  0.0000   0.1173
H_1     DZSPDN         0.0000  0.7572  -0.4692
H_2     DZSPDN         0.0000  -0.7572 -0.4692
END GEOMETRY

TASKS
        method = "UHF"
        propagatorTheoryCorrection=2
END TASKS

CONTROL
        iterationScheme=3
        convergenceMethod=1
        readCoefficients=F
        hartreeProductGuess=T
END CONTROL
```

Figure 1.1: Lowdin input

particles can be changed with the keywords **q=** and **m=** respectively, otherwise the previously defined default values will be used for each quantum species, see section 1.4.

The `TASKS` block defines the type of calculation to be performed by LOWDIN. All capabilities are listed in section capabilities. Finally `CONTROL` block contains all parameters needed to control the behavior of the program, such as thresholds, maximum number of SCF cycles, etc.

1.4 Basis and database

Basis set

LOWDIN contains a large library of electronic basis sets. Some of the nuclear basis sets developed by Nakai [1, 2] and Hammes-Schiffer [3] groups are also available. Home-made basis sets for nuclei, positrons and muons have been generated by employing the even-tempered basis set scheme [4, 5].

Any type of quantum particles or basis sets can be loaded even if they are not present in the main library. Users can upload new basis sets and quantum species through the addition or modification of some plain text files matching the information provided in the input file.

The basis set are located in the folder `lowdin2/lib/basis/`. The `deMon2k` format was selected for the basis set files in Lowdin. The name of these files corresponds should be the same name of the basis set declared in the input.

For example, the positronic basis set named `E+-H-7SPD-AUG-CC-PVTZ` implies that the basis set should be in a file with the same name (in uppercase) `lowdin2/lib/basis/E+-H-3S2P1D-AUG-CC-PVTZ` without extension. The basis file structure is

```
# Comment are marked with "#"
O-ParticleName ParticleSymbol (basisSetName) BASIS TYPE: 1 # 1 for e- 2 for other specie
#
numberOfContractions
iterator angularMoment numberOfPrimitives
exponent coefficient
iterator+1 angularMoment numberOfPrimitives
```

```
exponent coefficient
.
.
.
```

Example:

```
O-POSITRON E+ (E+-H-3S2P1D-AUG-CC-PVTZ) BASIS TYPE: 2
#
6
1 0 1
0.00079879 1.00000000
2 0 1
0.00252600 1.00000000
3 0 1
0.00798791 1.00000000
4 1 1
0.00322552 1.00000000
5 1 1
0.01020000 1.00000000
6 2 1
0.00781083 1.00000000
```

Databases

The folder lowdin2/lib/dataBases/ contains several files related to information of quantum and classical particles.

- atomicElements.lib

  This file has some physical and chemical properties of electrons and nuclei (atoms).

```
&ELEMENT
        NAME = "HYDROGEN"
        SYMBOL = "H"
        ATOMICNUMBER = 1
        MASS = 1.00794
        MELTINGPOINT = 13.81
        BOILINGPOINT = 20.28
        DENSITY = 0.084
        ELECTRONAFFINITY = -73
        IONIZATIONENERGY1 = 1312
        ELECTRONEGATIVITY = 2.1
        COVALENT = 0.30
        ATOMIC = 0.25
        KLAMT = 0.00
        VANDERWAALS = 1.2
        ISOTOPES = 1, 1.0078250321, 99.9885, 0.5,
                2, 2.0141017780, 0.0115, 1,
                3, 3.0160492675, 0.0, 0.5,
                4, 4.02783, , -2.0,
                5, 5.03954, , ,
                6, 6.04494, , ,
                7, 3.1289311806, , 1,
/
```

- elementalParticles.lib

This file defines the basic physical properties of quantum particles: name, symbol, charge, mass and spin.

```
&PARTICLE
        NAME = "ELECTRON"
        SYMBOL = "E-"
        CATEGORY = "LEPTON"
        CHARGE = -1
        MASS = 1.0
        SPIN = 0.5
/
```

- constantsOfCoupling.lib

  Similarly, this file defines some internal constants used in the Lowdin code.

```
&SPECIE
        NAME = "ELECTRON"
        SYMBOL = "E-"
        KAPPA = -1.0
        ETA = 2.0
        LAMBDA = 2.0
        PARTICLESFRACTION = 0.5
/
```

KAPPA	Sign of the exchange operator.
ETA	Number of particles per orbital.
LAMBDA	Number of particles per orbital (used in post-HF methods).
PARTICLESFRACTION	Inverse of the number of particles per orbital.

1.5 Outputs files

For the simplest possible calculation performed with Lowdin, these are the output files.

.out	Contains the basic output information: molecular system details, basis set, geometry and the results for the selected method.
.aux	The Lowdin input written as a Fortran 90 namelist.
lowdin.wfn	A binary file with all information related to matrices and energies for a Hartree-Fock based method.
lowdin.sys	Minimal information about molecular system, basis set and geometry.
lowdin.bas	Minimal information about basis set and geometry.

Table 1.1: Basis Lowdin output files

For additional information about any output files not listed above, please read the other sections according to the calculation requested.

Chapter 2

Hartree-Fock Calculations

2.1 RHF

One of the basic methodologies under the Any Particle Molecular Orbital (APMO) approach, is the Restricted Hartree-Fock method (RHF). An example of the hydrogen fluoride molecule, considering the electrons and the hydrogen nucleus as quantum particles can be constructed by writing `method` = ”`RHF`” in the `CONTROL` group as follows

```
GEOMETRY
    e-(F)   cc-pVTZ   0.00 0.00 0.91
    e-(H)   cc-pVTZ   0.00 0.00 0.00
    F       dirac     0.00 0.00 0.91
    H$\_$1     DZSPDN     0.00 0.00 0.00
END GEOMETRY

TASKS
    method = "RHF"
END TASKS
```

In the output the following lines will be printed

```
-----------------------------------------------------------------------
** PROGRAM: HF (Hartree Fock).      Author: S.A. Gonzalez, E. Posada
-----------------------------------------------------------------------
INFO: RUNNING IN RHF MODE.

-----------------------------------------------------------------------
** PROGRAM INTS                          Author: E. F. Posada, 2013
-----------------------------------------------------------------------
INFO: RUNNING IN ONE$\_$PARTICLE MODE.

 Stored   2500 overlap integrals of specie E- in file lowdin.opints
 Stored    400 overlap integrals of specie H$\_$1 in file lowdin.opints
 Stored   2500 kinetic integrals of specie E- in file lowdin.opints
 Stored    400 kinetic integrals of specie H$\_$1 in file lowdin.opints
 Stored   2500 attraction integrals of specie E- in file lowdin.opints
 Stored    400 attraction integrals of specie H$\_$1 in file lowdin.opints
 Stored   7500 Moment integrals of specie E- in file lowdin.opints
 Stored   1200 Moment integrals of specie H$\_$1 in file lowdin.opints

** TOTAL elapsed Time INTS :      0.151 (s)
```

here are the elapsed time to calculate the one particle integrals for each quantum species.

```
 LIBINT library, Fermann, J. T.; Valeev, F. L. 2010
 LOWDIN-LIBINT Implementation V. 2.1  Posada E. F. ; Reyes A. 2011
 ---------------------------------------------------------------------
**Stored        30102 non-zero repulsion integrals of species: E-
** TOTAL elapsed Time INTS :      0.151 (s)
 Stored       136345 non-zero repulsion integrals of species: E-
** TOTAL elapsed Time INTS :      0.260 (s)
**Stored        23939 non-zero repulsion integrals of species: E-
** TOTAL elapsed Time INTS :      0.377 (s)
 Stored         2493 non-zero repulsion integrals of species: H$\_$1
** TOTAL elapsed Time INTS :      0.037 (s)
**Stored          248 non-zero repulsion integrals of species: H$\_$1
** TOTAL elapsed Time INTS :      0.030 (s)
**Stored          622 non-zero repulsion integrals of species: H$\_$1
** TOTAL elapsed Time INTS :      0.135 (s)
 LIBINT library, Fermann, J. T.; Valeev, F. L. 2010
 LOWDIN-LIBINT Implementation V. 2.1  Posada E. F. ; Reyes A. 2011
 ---------------------------------------------------------------------
 LIBINT parameters:
 Stack size                            30000
 number of primitives                     70
 Memory required (in words)            38176

 Stored        68628 non-zero repulsion integrals between species: E- / H$\_$1

** TOTAL elapsed Time INTS :      0.140 (s)
```

later the two-particles integrals: the intraspecies integrals $(\alpha - \alpha)$ and the interspecies integrals $(\alpha - \beta)$.

```
 Begin Multi-Species SCF calculation:

 ---------------------------------------------------------
 Iteration      Energy             Energy Change
 ---------------------------------------------------------
    1      -90.1548278348       90.1548278348
    2      -96.0496694906        5.8948416558
    .
    .
    .
   31     -100.0183849887        0.0000000033
   32     -100.0183849903        0.0000000016

 ...end Multi-Species SCF calculation
```

Next, the self-consistent field (SCF) procedure will be carried out. First, the convergence of the total energy will be printed and later the convergence for each quantum species. Here it is important to mention that only one SCF procedure was performed.

```
  Begin SCF calculation by: E-
 -------------------------

 -----------------------------------------------------------------
 Iteration      Energy           Density Change         DIIS Error
 -----------------------------------------------------------------
    1     -100.5851578943        0.5452778366         --
    2     -103.4847643686        0.0181727056         --
    .
    .
    .
   30     -105.2116701318        0.0000008938         --
   31     -105.2115117058        0.0000006320         --

 ... end SCF calculation

 Begin SCF calculation by: H$\_$1
 -------------------------

 -----------------------------------------------------------------
 Iteration      Energy           Density Change         DIIS Error
 -----------------------------------------------------------------
    1        0.0992916581        1.9676485511         --
    2       -0.6890196867       17.1532366339         --
    .
    .
    .
   30       -0.8445017555        0.0000223413         --
   31       -0.8445010548        0.0000031232         --

 ... end SCF calculation

** TOTAL elapsed Time SCF :      1.980 (s)

  EIGENVALUES AND EIGENVECTORS:
 ==============================

   Eigenvectors for: E-
 -----------------

             -26.2810       -1.5899       -0.7519       -0.6404       -0.6404

1   F    S      0.970921     -0.005978     -0.002568     -0.000000      0.000000
2   F    S     -0.010700      0.543960     -0.136492     -0.000000      0.000000
3   F    S      0.046167      0.157824     -0.032848      0.000000     -0.000000
4   F    S     -0.004354      0.325246     -0.255324     -0.000000      0.000000
5   F   Px     -0.000000     -0.000000      0.000000     -0.320122     -0.211321
6   F   Py      0.000000     -0.000000      0.000000     -0.211321      0.320122
7   F   Pz     -0.001244     -0.033682     -0.326090     -0.000000      0.000000
    .
    .
    .
```

The next part of the output file are the eigenvalues (at the top of the columns) and the eigenvectors for each quantum species.

```
  ENERGY COMPONENTS:
 ==================

         TOTAL KINETIC ENERGY      =       100.0133180735
         TOTAL POTENTIAL ENERGY    =      -200.0317030638

                      TOTAL ENERGY =    -100.018384990309
```

```
               VIRIAL RATIO (V/T) =         2.0000506624

 COMPONENTS OF KINETIC ENERGY:
-----------------------------

              E- Kinetic energy   =        99.9949305385
             H_1 Kinetic energy   =      0.0183875351

             Total kinetic energy =       100.0133180735

 COMPONENTS OF POTENTIAL ENERGY:
-------------------------------

          Fixed potential energy    =         0.0000000000
          Q/Fixed potential energy  =      -239.3984608764
          Coupling energy           =        -6.0376255794
          Repulsion energy          =        45.4043833919
          ExternalPotential energy  =         0.0000000000

             Total potential energy =      -200.0317030638

 Repulsion energy:
------------------

          E-/E- Repulsion energy  =        45.4043833919
        H_1/H_1 Repulsion energy  =         0.0000000000

 Coupling energy:
----------------

              E- Coupling energy  =        -6.0376261693
             H_1 Coupling energy  =        -6.0376271804

 Quantum/Fixed interaction energy:
-----------------------

        E-/Fixed interact. energy =      -244.5731994669
        H_1/Fixed interact. energy=         5.1747385905

 END ENERGY COMPONENTS
```

The last section is the summary of the energy components. The above terms are explained in the table 2.1

TOTAL KINETIC ENERGY	Sum of the total kinetic energy for all quantum species.
TOTAL POTENTIAL ENERGY	Sum of the total potential energy interactions in the molecular system.
TOTAL ENERGY	TOTAL KINETIC ENERGY + TOTAL POTENTIAL ENERGY.
VIRIAL RATIO (V/T)	-TOTAL POTENTIAL ENERGY/TOTAL KINETIC ENERGY.
α Kinetic energy	Sum of the Kinetic energy for all α particles.
Fixed potential energy	Sum of the potential between point charges.
Q/Fixed potential energy	Sum of the potential between point charges and quantum species.
Coupling energy	Sum of total two-particles coulomb potential between all combinations of different quantum species.
Repulsion energy	Sum of total two-particles potential between particles of the same quantum species for all quantum species.
α/α Repulsion energy	Sum of total two-particles potential between particles of the α species.
α Coupling energy	Sum of total two-particles potential between particles of the α species with all the others quantum species.
α/Fixed interact. energy	Sum of the potential between point charges and the particles of the α species.

Table 2.1: RHF energy components.

2.2 UHF

To perform an Unrestricted Hartree-Fock calculation the keyword `method` = ”`RHF`” should be included. The Unrestricted orbitals will be constructed only for electrons. The multiplicity can be selected by adding the `multiplicity` option in one of electrons rows, by default the multiplicity is 1.

```
GEOMETRY
    e-(F)   cc-pVTZ   0.00 0.00 0.91
    e-(H)   cc-pVTZ   0.00 0.00 0.00 multiplicity=1
    F       dirac     0.00 0.00 0.91
    H$\_$1     DZSPDN     0.00 0.00 0.00
END GEOMETRY
TASKS
    method = "UHF"
END TASKS
```

Similarly to the RHF calcution, the output of UHF is the same but now the `E − ALPHA` and `E − BETA` quantum species will be constructed from the `E−` species.

2.3 HF Options

Here are some basic HF options.

writeCoefficientsInBinary Write the eigenvectors from a previous calculation stored as a binary file, otherwise it will be write as a plain text format. Default `T`.

writeEigenValuesInBinary Write the eigenvalues from a previous calculation stored as a binary file, otherwise it will be write as a plain text format. Default `T`.

readCoefficients Read the eigenvectors from a previous calculation. Default `F`.

readCoefficientsInBinary Read the eigenvectors from a previous calculation stored as a binary file, otherwise it will be read from a plain text format. Default `T`.

readEigenValues Read the eigenValues from a previous calculation. Default `F`.

readEigenValuesInBinary Read the eigenValues from a previous calculation stored as a binary file, otherwise it will be read from a plain text format. Default `T`.

noScf Do not perform a SCF procedure, just calculates the energy. `F`.

finiteMassCorrection Add the finite mass correction of Mohallem[6] extended to any particle. Default `F`.

removeTranslationalContamination Remove the translational contamination to the total kinetic operator [7]. Defautl `F`.

buildTwoParticlesMatrixForOneParticle Build the two particles matrix and include it to the Fock matrix. Defautl `F`.

buildMixedDensityMatrix Build different density matrices for e-(alpha) and e-(beta) electronic densities. Default `F`.

Here are some basic integrals options and SCF options.

2.4 Integrals options

integralStackSize Read or write stacks of two particles integrals of this size. Default `30000`.

integralScheme Selects the method for calculate the two particles integrals. "`LIBINT`": Use the libint library [8]. "`RYS`": Use Rys Quadrature. Default "`LIBINT`".

2.5 SCF options

scfNonElectronicEnergyTolerance — SCF Convergence is reached when the non-electronic energy change between two consecutive SCF iterations is less than this value for non-electronic quauntum species. Default $\texttt{1.0E}-\texttt{5}$

scfElectronicEnergyTolerance — SCF Convergence is reached when the electronic energy change between two consecutive SCF iterations is less than this value for electronic wavefunction. Default $\texttt{1.0E}-\texttt{6}$

nonElectronicDensityMatrixTolerance — SCF Convergence is reached when the non-electronic density change between two consecutive SCF iterations is less than the thos (in absolute value) for non-electronic quauntum species. Default $\texttt{5.0E}-\texttt{4}$

electronicDensityMatrixTolerance — SCF Convergence is reached when the electronic density change between two consecutive SCF iterations is less than the thos (in absolute value) for non-electronic quauntum species. Default $\texttt{1.0E}-\texttt{6}$

totalEnergyTolerance — SCF Convergence is reached when the total energy change between two consecutive SCF iterations is less than this value. Default $\texttt{1.0E}-\texttt{7}$

scfNonElectronicMaxIterations — Maximum number of iterations of SCF for non-electronic species. Default `10000`

scfElectronicMaxIterations — Maximum number of iterations of SCF iterations for electrons. Default `20000`

scfGlobalMaximumIterations — Global maximum number of iterations of SCF iterations for all quantum species. Default `5000`

convergenceMethod — Selects the convergence method: 0 NONE, 1 DAMPING, 2 DIIS, 3 LEVEL SHIFTING or 4 DAMPING/DIIS. Default `1`

iterationScheme — Defines the iteration scheme for SCF calculations with multiple quantum species: 0 NONELECRONIC FULLY, 1 ELECTRONIC FULLY, 2 CONVERGED INDIVIDIALLY or 3 SCHEMESIMULTANEOUS. Default `3`

scfElectronicTypeGuess — Selects the initial electronic density: ”HUCKEL”, ”HCORE” or ”ONES”. Huckel: , Hcore: Diagonalize the one electron Hamiltonian to obtain the initial eigenvectors. Ones: Fill the density matrix with ones for all occupied orbitals. Default `HCORE`

scfNonElectronicTypeGuess — Selects the initial non-electronic density: ”HCORE” or ”ONES”. Default `HCORE`

scfConvergenceCriterium — Selects the convergence criterium for all quantum species. ”ENERGY” or ”DENSITY”. Default `ENERGY`

debugScfs — Print additional information in the output file for debugging. One particle hamiltonian matrix, Fock matrix, two particles matrix and total energy for each quantum species. Default `F`

2.6 Reading a .vec file

One useful option is read the eigenvectors from a previous Lowdin calculation. To do that, follow these steps

1. Run a Lowdin calcuation saving the *wfn* file. For example :

```
GEOMETRY
    e-(F)   cc-pVTZ   0.00 0.00 0.91
    e-(H)   cc-pVTZ   0.00 0.00 0.00
    F       dirac     0.00 0.00 0.91
    H$\_$1     DZSPDN    0.00 0.00 0.00
END GEOMETRY

TASKS
    method = "RHF"
END TASKS

lowdin2 -i inputName.lowdin -w
```

2. Copy the `lowdin.wfn` file to a new file `lowdin.vec`

```
cp lowdin.wfn lowdin.vec
```

3. Run the new Lowdin calculation with the option `readCoefficients` = `T` and with the desired changes, for example increasing the charge of the proton and freezing the SCF procedure for electrons.

```
GEOMETRY
    e-(F)   cc-pVTZ   0.00 0.00 0.91
    e-(H)   cc-pVTZ   0.00 0.00 0.00
    F       dirac     0.00 0.00 0.91
    H$\_$1  DZSPDN    0.00 0.00 0.00 q = 1.05
END GEOMETRY

TASKS
    method = "RHF"
END TASKS
CONTROL
    readCoefficients = T
    scfElectronicMaxIterations = 2  ! The minimum iterations is 2.
END CONTROL
```

2.7 Auxiliary basis

The way to include auxiliary basis centers in Lowdin is by adding and removing particles with the option `addParticles` as follows. First the regular input with only one positronic basis center on the Fluorine atom

```
GEOMETRY
    e-(F)   cc-pVTZ   0.00 0.00 0.91
    e-(H)   cc-pVTZ   0.00 0.00 0.00
    F       dirac     0.00 0.00 0.91
    H       DZSPDN    0.00 0.00 0.00
    e+      e+-F-7SPD-AUG-CC-PVDZ 0.00 0.00 0.91
END GEOMETRY
```

including a new line for `e+` will add another positron to the molecular system

```
GEOMETRY
    e-(F)   cc-pVTZ   0.00 0.00 0.91
    e-(H)   cc-pVTZ   0.00 0.00 0.00
    F       dirac     0.00 0.00 0.91
    H       DZSPDN    0.00 0.00 0.00
    e+      e+-F-7SPD-AUG-CC-PVDZ 0.00 0.00 0.91
    e+      e+-F-7SPD-AUG-CC-PVDZ 0.00 0.00 0.00
END GEOMETRY
```

writting the option `addParticles=-1` it will remove one positron but not the basis set

```
GEOMETRY
    e-(F)   cc-pVTZ   0.00 0.00 0.91
    e-(H)   cc-pVTZ   0.00 0.00 0.00
    F       dirac     0.00 0.00 0.91
    H       DZSPDN    0.00 0.00 0.00
    e+      e+-F-7SPD-AUG-CC-PVDZ 0.00 0.00 0.91
    e+      e+-F-7SPD-AUG-CC-PVDZ 0.00 0.00 0.00 addParticles=-1
END GEOMETRY
```

note that several basis center could be added for any quantum species in any position

```
GEOMETRY
    e-(F)   cc-pVTZ   0.00 0.00 5.00 addParticles=-9
    e-(F)   cc-pVTZ   0.00 0.00 0.91
    e-(H)   cc-pVTZ   0.00 0.00 0.00
    F       dirac     0.00 0.00 0.91
    H       DZSPDN    0.00 0.00 0.00
    e+      e+-F-7SPD-AUG-CC-PVDZ 0.00 0.00 5.00 addParticles=-1
    e+      e+-F-7SPD-AUG-CC-PVDZ 0.00 0.00 0.91
    e+      e+-F-7SPD-AUG-CC-PVDZ 0.00 0.00 0.00 addParticles=-1
END GEOMETRY
```

here the new line `e-F` would add 9 electrons for that reason 9 electrons should be removed with the option `addParticles=-9` in the spirit of auxiliary functions.

Chapter 3

Moller-Plesset Calculations

Lowdin can perform second order many-body perturbation theory corrections to the energy on systems including any kind of quantum particle. These corrections are probably the simplest way to recover the effects of correlation between particles of different species.

For more information on APMO-MP2 calculations see
S. A. González, A. Reyes, "Nuclear Quantum Effects on the He2H+ Complex With the Nuclear Molecular Orbital Approach", Int. J. Quant. Chem. 110 689 (2010)

3.1 Example

An APMO-MP2 on the hydrogen fluoride molecule, where the electrons and hydrogen nucleus are treated as quantum particles should be like this:

```
GEOMETRY
        e-(F)   cc-pvtz   0.00 0.00 0.91
        e-(H)   cc-pvtz   0.00 0.00 0.00
        F       dirac     0.00 0.00 0.91
        H       dzsph     0.00 0.00 0.00
END GEOMETRY
TASKS
        method="RHF"
        mollerPlessetCorrection=2
END TASKS
```

The output will include the summary of the MP2 results

```
 POST HARTREE-FOCK CALCULATION
 MANY-BODY PERTURBATION THEORY:
==============================

        MOLLER-PLESSET FORMALISM
         ORDER OF CORRECTION =     2

         E(0) + E(1) =  -1.000183849903E+02
                E(2) =  -3.004542681882E-01
                       --------------------
               E(MP2)=  -1.00318839258497476E+02

        -----------------------------------
         E(n){ Specie }    E(n) / Hartree
        -----------------------------------

        E(2){  E-    } =  -2.89877340E-01
        E(2){ H_1    } =   0.00000000E+00

        E(2){  E-/H_1} =  -1.05769285E-02
```

Where the E(2) is the second order correction to the energy, and the E(MP2) is the Hartree-Fock energy plus E(2). This summary also includes the intraspecies and interspecies contributions to E(2).

3.2 Options

APMO-MP2 calculations are performed when the option
mollerPlessetCorrection=2
is present in the "TASKS" block.

There are only one option related to MP2 calculations.

mpFrozenCoreBoundary	Omits this number of occupied electronic molecular orbitals in the MP2 calculations (core electrons). Default 0.

Table 3.1

Chapter 4

Propagator Calculations

LOWDIN can calculate ionization potentials for any species employing the propagator formalism, where ionization energy for an specific orbital is calculated as the Koopmans value plus many body perturbation theory (MBPT) corrections. The current implementation includes second order corrections (APMO/EP2) and second order plus transition operator corrections (APMO/TOEP2) [9]. The latter method takes advantage of fractional occupation to include additional relaxation in calculated ionization energies. And lately, the third order corrections [5]. Details of the propagator formalism can be found in references [10, 11].

To perform a propagator calculations using LOWDIN, the order of MBPT (2 or 3) must be specified in the `TASKS` block using the keyword `propagatorTheoryCorrection`. For example:

```
TASKS
    method = "RHF"
    propagatorTheoryCorrection = 2
END TASKS
```

A default calculation obtains ionization energies for the HOMO and LUMO orbitals of all the species present in the input. Currently, the third order corrections is only available from a UHF reference. The following options in the block `OPTIONS`, allows the user to have more control of the propagator calculation.

OPTIONS

`IonizeMO =` This option specifies an orbital for which the propagator calculation will be perform. An integer must be specified. If this option is activated, the propagator calculation is performed only for the specified orbital. If several species are include in the input, the species at which the orbital of interest belongs must be also specified using the option `ionizeSpecie`. (Default=0. All the orbitals are calculated)

`ionizeSpecie =` Array which specifies the species at which the orbital of interest (specified by `ionizeSpecie`) belongs. (Default="e-"). If it appears without any other specification, the calculations are run only for the specified species. A maximum of 20 different species is allowed.

`ptTransitionOperator =` A logical flag to activate the transition operator method, in which the orbital of interest has a fractional occupation. If `ptTransitionOperator=T the options IonizeMO, ionizeSpecie and MOfractionOccupation are mandatory. (Default=F)`

`MOfractionOccupation =` This option specifies the fractional occupation of the orbital of interest for the transition operator method. This value can be any real between 0 and 1. The recommended value is 0.5. (Default=1.0)

EXAMPLES

Default Input: Ionization energies for a water molecule using EP2

```
SYSTEM_DESCRIPTION='Water molecule'

GEOMETRY
e-(O)   6-31G     0.0000 0.0000  0.1173   multiplicity=1
e-(H) 6-31G     0.0000 0.7572 -0.4692
e-(H) 6-31G       0.0000 -0.7572 -0.4692
O_16  Nakai-7-SPD 0.0000 0.0000 0.1173
H-a_1 Nakai-7-SPD 0.0000 0.7572 -0.4692
H-b_1 Nakai-7-SPD 0.0000 -0.7572 -0.4692
END GEOMETRY

TASKS
        method = "RHF"
        propagatorTheoryCorrection=2
END TASKS

CONTROL
ionizeSpecie="H-A_1" "H-B_1"
END CONTROL
```

At the end of the output the information of ionization energies is found as a table:

```
  POST HARTREE-FOCK CALCULATION
  PROPAGATOR THEORY:
 ==============================

         PROPAGATOR FORMALISM FOR SEVERAL FERMIONS SPECIES
          ORDER OF CORRECTION =     2

         SPECIE: H-A_1
         ---------------------------------------------
           Orbital   KT (eV)   EP2 (eV)     P.S
         ---------------------------------------------
              1      -24.5978  -17.0139    0.8775
              2      -23.9426  -57.7316    0.5384
         ---------------------------------------------
         SPECIE: H-B_1
         ---------------------------------------------
           Orbital   KT (eV)   EP2 (eV)     P.S
         ---------------------------------------------
              1      -24.5978  -17.0139    0.8775
              2      -23.9426  -57.7316    0.5384
         ---------------------------------------------
```

If the user are only interested on calculating just one orbital, for example, the valence electronic orbital (5), the options `ptJustOneOrbital`,`ionizeSpecie` and `IonizeMO` must be included as:

```
CONTROL
   ...
   IonizeMO=5
   ionizeSpecie="E-"
   ptJustOneOrbital=T
   ...
END CONTROL
```

Example transition operator method + second order propagator: TOEP2

```
SYSTEM_DESCRIPTION='Water molecule'

GEOMETRY
e-(O)   6-31G     0.0000        0.0000  0.1173   multiplicity=1
e-(H)   6-31G     0.0000        0.7572  -0.4692
e-(H)   6-31G     0.0000        -0.7572 -0.4692
O_16    Nakai-TRF-7SP    0.0000        0.0000  0.1173
H-a_1   Nakai-TRF-7SP    0.0000        0.7572  -0.4692
H-b_1   Nakai-TRF-7SP    0.0000        -0.7572 -0.4692
END GEOMETRY

TASKS
        method = "UHF"
        propagatorTheoryCorrection=2
END TASKS

CONTROL
        ionizeSpecie="E-ALPHA"
        IonizeMO=5
        MOfractionOccupation=0.5
        ptTransitionOperator=T
END CONTROL
```

At the end of the output the information of ionization energies is found as a table:

```
  POST HARTREE-FOCK CALCULATION
  PROPAGATOR THEORY:
 ==============================

        PROPAGATOR FORMALISM FOR SEVERAL FERMIONS SPECIES
         ORDER OF CORRECTION =     2  + TRANSITION OPERATOR

        SPECIE: E-ALPHA
        ---------------------------------------------------------------------------------
          Orbital   KT (eV)  EP2 (eV)      P.S   SCS-EP2(eV)     P.S   SOS-EP2(eV)     P.S
        ---------------------------------------------------------------------------------
             5      -12.1883  -12.7366    0.4827    -12.7243    0.4982    -12.7179    0.5061
        ---------------------------------------------------------------------------------
```

This also prints the SCS-EP2 and SOS-EP2 values, available for any UHF-EP2 calculation [12].

Chapter 5

Configuration Interaction

The LOWDIN determinant CI program is capable of computing the first n-eigenvalues and associated eigenvectors at any truncated CI level up to Full-CI. Here, the wave function is constructed as a linear combination of all the possible excited determinants built from the reference (HF) wave function, which includes simultaneous excitations of particles of different species that account for interspecies correlation. The latest implementation includes CIS, CISD, CISDT, CISDTQ, CISDTQQ (singles doubles triples quadruples quintuples configuration interaction) and Full-CI starting from a UHF reference (open shell or closed shell)

To perform a CI calculation using LOWDIN, the excitation level must be specified in the `TASKS` block using the keyword `ConfigurationInteractionLevel`. For example:

```
TASKS
    method = "UHF"
    ConfigurationInteractionLevel = "CISD"
END TASKS
```

The following options in the block `OPTIONS`, allows the user to have more control of the CI calculation and diagonalization procedure.

OPTIONS

`numberOfCIStates` = First n-eigenpairs to be calculated. Default = 1

`CIdiagonalizationMethod` = Select the diagonalization method for the CI Hamiltonian matrix. Available options: "DSYEVR" and "DSYEVX" LAPACK's eigensolvers of real symmetric matrices, requires the storage of the full matrix, high memory usage. "JADAMILU" a Jacobi-Davidson diagonalizer of large scale matrices, the memory requirements are greatly reduced by direct calculation of the CI matrix to perform a matrix-vector product. Default = "DSYEVR"

`CIsizeOfGuessMatrix` = Size of the initial CI matrix required to obtain the initial guess for JADAMILU. Default = 300

`CIConvergence` = Tolerance for the eigenvector residual. Only for JADAMILU. Default = 0.0001

`CImatvecTolerance` = Tolerance for the matrix-vector product. Only for JADAMILU. Default = 1E-10

`CIMadSpace` = This parameter describes the size of the search space beside the eigenvectors that need to be computed. Only for JADAMILU. Default = 5

`CIPrintThreshold` = Print all CI coefficients with an absolute value larger than this threshold. Default 1E-1

`CIPrintEigenVectorsFormat` = Select the format for printing CI coefficients. "OCCUPIED", for each configuration, print the occupied orbitals for all quantum species in the same order specified in the geometry. "ORBITALS", for each configuration, print the orbital occupation for all quantum species in the same order specified in the geometry block. "0" for unoccupied orbitals, and "1" for occupied. "NONE" don't print CI coefficients. Default "OCCUPIED".

For JADAMILU, the required memory is approximately calculated as $(3 * \texttt{CIMadSpace} + \texttt{numberOfCIStates} + 3) * N$, where N is the total number of configurations.

Additionally, the number of frozen core orbitals and the number of active orbitals (active occupied orbitals + active virtual orbitals) can be specified for each quantum species in the $\texttt{INPUT}_{\texttt{C}}\texttt{I}$ block as follows:

```
INPUT_CI
        species="E-ALPHA" core=0 active=0
        species="E-BETA" core=0 active=0
END INPUT_CI
```

if these values are not supplied or are declared to be zero, then all occupied orbitals will be consider as active occupied orbitals (no frozen core) and all virtual orbitals will be included in the active space by default.

EXAMPLES

FCI calculation for PsH

```
GEOMETRY
        e-(H)   SHARON-E-6S2P   0.00    0.00    0.00 addParticles = 1
        H       dirac           0.00    0.00    0.00
        e+      SHARON-E+6S2P   0.00    0.00    0.00
END GEOMETRY
TASKS
        method = "UHF"
        configurationInteractionLevel ="FCI"
END TASKS
CONTROL
        numberOfCIstates=3
        CIdiagonalizationMethod = "JADAMILU"
        CIsizeOfGuessMatrix = 500
        CIConvergence = 1E-4
        CImatvecTolerance = 1E-10
        CIMadSpace = 10
END CONTROL
OUTPUTS
       moldenFile
END OUTPUTS
INPUT_CI
        species="E-ALPHA" core=0 active=0
        species="E-BETA" core=0 active=0
        species="POSITRON" core=0 active=0
END INPUT_CI
```

Chapter 6

Outputs

Besides the standard output, Lowdin can generate other type of outputs to view the results of an APMO calculation. Lowdin input has an "OUTPUTS" block to request these outputs. Currently, it can generate:

- Molden files for each specie.
- AIM files. These files are generated with the molden2AIM program [13].
- Gnuplot 2D and 3D graphs for density and orbitals

6.1 Molden and AIM files

To generate molden or AIM files simply add in the OUTPUTS block

```
moldenFile
wfnFile
wfxFile
NBO47File
```

Lowdin will generate an .molden, .wfn, .wfx, or .47 file for each quantum specie in the input. For molden, there are three format types that can be selected with the CONTROL option

moldenFileFormat = QUANTUM, Define the coordinates, GTO and MO for each quantum species individually.
= STANDARD, Same that QUANTUM but including the coordinates of classical particles.
= MIXED, Same that STANDARD but including 1s GTO for each classical particles with zero contribution in the MO. (Default)

6.2 Gaussian Cubes

NOTE: NOT IMPLEMENTED YET! To generate a Gaussian cube please use the molden file. Lowdin generates Gaussian density or orbital cubes of a chosen specie, adding in the OUTPUTS block the lines
orbitalCube specie= orbital=
densityCube specie=

Lowdin will generate an .cub file for each cube requested. These cubes can be read by many visualization programs, such as VMD.

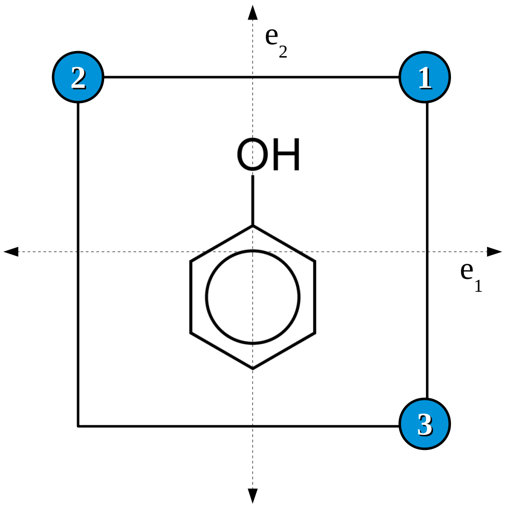


Figure 6.1: density and orbitals plots scheme

To change the number of points in each cube, adjust the points density in the CONTROL block, adding the line
cubePointsDensity= (default=125)
This number indicates the number of points contained in one atomic unit of volume.

6.3 Gnuplot files

Lowdin generates plots of the density or orbitals of a chosen specie using Gnuplot. To do this, add in the OUTPUTS block the lines

```
densityPlot specie= dimensions= point1=X1 Y1 Z1 point2=X2 Y2 Z2 point3=X3 Y3 Z3 \\
orbitalPlot specie= orbital= dimensions= point1=X1 Y1 Z1 point2=X2 Y2 Z2 point3=X3 Y3 Z3
```

To create 2D plots, set dimensions=2 and provide TWO points in atomic units. For example
densityPlot specie=e- dimensions=2 point1=0.0 0.0 -2.0 point2=0.0 0.0 2.0
This will create a plot of the electron density along the Z axis, from -2.0 to 2.0 a.u.

To create 3D plots, set dimensions=3 and provide THREE points in atomic units. Usually these will be 3 corners of a rectangle defined in one plane. For example
densityPlot specie=e- dimensions=3 point1=0.0 2.5 2.5 point2=0.0 -2.5 2.5 point3=0.0 2.5 -2.5
This will create a plot of the electron density in the YZ plane, from -2.5 to 2.5 a.u. in both axis. See Figure 6.1.

Lowdin will generate three files for each Gnuplot: A .dens or a .orb with the raw data of the plot; a .gnp with the gnuplot script to generate the plot; and a .eps with the plot itself.

To change the number of points in the graph, adjust the number of points in each dimension in the CONTROL block, adding the line

```
numberOfPointsPerDimension=     (default=100)
```

6.4 Example

```
GEOMETRY
        e-(H)   6-31G   0.00 0.00 0.91
        e-(F)   6-31G   0.00 0.00 0.00
        H_1     DZSPDN  0.00 0.00 0.91
        F       dirac   0.00 0.00 0.00
END GEOMETRY

TASKS
        method = "RHF"
END TASKS

OUTPUTS
        moldenFile
        wfnFile
        orbitalPlot specie=E- orbital=5 dimensions=3 point1=0.0 2.5 2.5 point2= 0.0 -2.5 2.5 point
        densityPlot specie=H_1  dimensions=2 point1=0.0 0.0 -2.5 point2= 0.0 0.0 2.5
END OUTPUTS
```

Assuming that this input is named 'OutputsExample', this calculation will generate the following files:

- OutputsExample.E-.molden
- OutputsExample.H_1.molden
- OutputsExample.E-.wfn
- OutputsExample.H_1.wfn
- OutputsExample.out4.H_1.2D.dens
- OutputsExample.out4.H_1.2D.dens.gnp
- OutputsExample.out4.H_1.2D.dens.eps
- OutputsExample.out5.E-.3D.orb5
- OutputsExample.out5.E-.3D.orb5.gnp
- OutputsExample.out5.E-.3D.orb5.eps

Bibliography

[1] H. Nakai. Simultaneous determination of nuclear and electronic wave functions without Born–Oppenheimer approximation: Ab initio NO+ MO/HF theory. *Int. J. Quantum Chem.*, 86(6):511–517, 2002.

[2] Hiromi Nakai, Minoru Hoshino, Kaito Miyamoto, and Shiaki Hyodo. Elimination of translational and rotational motions in nuclear orbital plus molecular orbital theory. *J. Chem. Phys*, 122(16):164101–164101, 2005.

[3] S.P. Webb, T. Iordanov, and S. Hammes-Schiffer. Multiconfigurational nuclear-electronic orbital approach: Incorporation of nuclear quantum effects in electronic structure calculations. *J. Chem. Phys.*, 117:4106, 2002.

[4] Félix Moncada, Daniel Cruz, and Andrés Reyes. Muonic alchemy: Transmuting elements with the inclusion of negative muons. *Chem. Phys. Lett.*, 539–540(0):209–213, 2012.

[5] J. Romero, J. A. Charry, R. Flores-Moreno, M. T. do N. Varella, and A. Reyes. Calculation of positron binding energies using the generalized any particle propagator theory. *The Journal of Chemical Physics*, 141(11):114103, 2014.

[6] José R. Mohallem, Leonardo G. Diniz, and Adriano S. Dutra. Separation of motions of atomic cores and valence electrons in molecules. *Chemical Physics Letters*, 501(4–6):575 – 579, 2011.

[7] Takayoshi Ishimoto, Masanori Tachikawa, and Umpei Nagashima. Simultaneous analytical optimization of variational parameters in gaussian-type functions with full configuration interaction of multicomponent molecular orbital method by elimination of translational and rotational motions: Application to isotopomers of the hydrogen molecule. *The Journal of Chemical Physics*, 128(16):–, 2008.

[8] J. T. Fermann and E. F. Valeev. Libint: Machine-generated library for efficient evaluation of molecular integrals over gaussians, 2003.

[9] J. Romero, E. Posada, R. Flores-Moreno, and A. Reyes. A generalized any particle propagator theory: Assessment of nuclear quantum effects on electron propagator calculations. *The Journal of Chemical Physics*, 137(7):074105, 2012.

[10] J. Linderberg and Y. Öhrn. *Propagators in Quantum Chemistry, 2nd ed.* Wiley-Interscience, Hoboken, NJ, 2004.

[11] Roberto Flores-Moreno, V. G. Zakrzewski, and J. V. Ortiz. Assessment of transition operator reference states in electron propagator calculations. *The Journal of Chemical Physics*, 127(13):–, 2007.

[12] J. Romero, J. Charry, H Nakai, and A. Reyes. Improving quasiparticle second order electron propagator calculations with the spin-component-scaled technique. *Chemical Physics Letters*, 591:82 – 87, 2014.

[13] molden2AIM program. `https://github.com/zorkzou/Molden2AIM`.