

# **FLOSIC Documentation and Tutorials**

*Release 2020*

**FLOSIC developers group**

**Nov 02, 2022**

## CONTENTS

<b>1</b>	<b>Introduction and Overview</b>	<b>1</b>
<b>2</b>	<b>Accessing and compiling the code</b>	<b>3</b>
<b>3</b>	<b>Quick start guide to running a FLOSIC calculation</b>	<b>5</b>
<b>4</b>	<b>Exchange correlation functionals</b>	<b>7</b>
<b>5</b>	<b>Controlling a calculation: the NRLMOL_INPUT.DAT file</b>	<b>8</b>
<b>6</b>	<b>FLOSIC Mesh</b>	<b>10</b>
<b>7</b>	<b>Input and Output Files</b>	<b>12</b>
<b>8</b>	<b>Basis sets: the ISYMGEN file</b>	<b>14</b>
<b>9</b>	<b>Tutorials</b>	<b>17</b>
9.1	Tutorial 1: Molecular geometry optimization at the DFT level . . . . .	17
9.2	Tutorial 2: a FLOSIC calculation for H2 . . . . .	18
9.3	Tutorial 3: FOD optimization of a water molecule . . . . .	19
9.4	Plotting density of states using NRLMOL . . . . .	23
<b>10</b>	<b>Getting more information about FLOSIC</b>	<b>34</b>

## INTRODUCTION AND OVERVIEW

The FLOSIC code is based on the UTEP version of NRLMOL, the Naval Research Laboratory Molecular Orbital Library.[1-4] NRLMOL is a massively parallel code for electronic structure calculations on molecules and clusters. It is based on the Kohn-Sham (KS) formulation of density functional theory (DFT) and solves KS equations by expressing the KS orbitals as a linear combination of Gaussian orbitals. NRLMOL was developed by Mark Pederson and collaborators.

NRLMOL contains an implementation of the Fermi-Löwdin orbital self-interaction correction (FLO-SIC) method[5-8] that corrects the self-interaction error for common exchange-correlation functionals. In FLO-SIC, the Kohn-Sham canonical orbitals are transformed into Fermi orbitals, which are orthogonalized to become Fermi-Löwdin orbitals (FLOs). These FLOs are used to evaluate the orbital-dependent self-interaction corrected total energy. The transformation requires a set of parameters that are points in three dimensional space, the so-called Fermi-orbital descriptors (FODs). These FODs form what can be thought of as the electronic geometry. The FOD positions are optimized to obtain the FLOs that minimize the total energy. Thus, there are two geometries to consider in a FLOSIC calculation: The molecular geometry (given by the atoms) and the electronic geometry (given by the FODs).

The optimization of the FODs is a crucial part of any FLO-SIC calculation. FOD optimization is analogous to a molecular geometry optimization. From a given FOD starting point, FOD forces (energy gradients with respect to FOD positions) are calculated and fed into a gradient-based optimization scheme, along with the FOD positions and the total energy, to update FOD positions. The optimization continues until the total energy and the FOD forces are converged.

The following is a list of some SIC-related properties that are calculated using the FLOSIC code:

- Total energy
- SIC contribution to the energy
- Orbital energies
- Orbital contributions to self-interaction correction; orbital moments, self-Coulomb, self-exchange, and self-correlation energy.
- Analytical FOD forces (for FOD optimization using conjugate gradient or LBFGS optimizers)
- Orbitals in .cube format

Physically interesting properties that have been evaluated recently with the FLOSIC code include:[9-14]

- Atomization energies
- Ionization potentials from the highest occupied orbitals
- SIC optimized molecular geometries
- Polarizabilities
- Dipole moments

- Magnetic exchange couplings

\$alpha\$

The following are references to the NRLMOL code:

- [1] M.R. Pederson and K.A. Jackson, *Phys. Rev. B*, **41**, 7453, 1990.
- [2] K.A. Jackson and M.R. Pederson, *Phys. Rev. B*, **42**, 3276, 1990.
- [3] D. Porezag and M.R. Pederson, *Phys. Rev. A*, **60**, 2840, 1999.
- [4] M.R. Pederson et al., *phys. stat. sol. b*, **217**, 197, 2000.

Further information about the FLO-SIC method can be found in the following references:

- [5] M.R. Pederson, A. Ruzsinszky, J.P. Perdew., *J. Chem. Phys.*, **140**, 121103, 2014.
- [6] M.R. Pederson, *J. Chem. Phys.*, **142**, 064112, 2015.
- [7] Z.-h. Yang, M.R. Pederson, J.P. Perdew, *Phys. Rev. A*, **95**, 052505, 2017.
- [8] M.R. Pederson, T. Baruah, *Advances In Atomic, Molecular, and Optical Physics*, Chapter 8, 2015.

Applications of FLO-SIC are described in these references:

- [9] K. Sharkas et al., *J. Phys. Chem. A*, **122**, 9307-9315, 2018.
- [10] R.P. Joshi et al., *J. Chem. Phys.*, **\*\*149\*\***, 164101, 2018.
- [11] D.-y. Kao, K. Withanage, T. Hahn, J. Batool, J. Kortus, K. Jackson, *J. Chem. Phys.*, **147**, 164107, 2017.
- [12] S. Schwalbe, T. Hahn, S. Liebing, K. Trepte, J. Kortus, *J. Comput. Chem.*, **39**, 2463-2471, 2018.
- [13] K. Withanage, K. Trepte, J.E. Peralta, T. Baruah, R. Zope, K.A. Jackson, *J. Chem. Theory Comput.*, **14**, 4122-4128, 2018.
- [14] K. Trepte, S. Schwalbe, T. Hahn, J. Kortus, D.-y. Kao, et al., *J. Comput. Chem.*, **40**, 820-825, 2019.

## ACCESSING AND COMPILING THE CODE

The code is accessible from a git repository hosted at the University of Texas at El Paso (UTEP). To download the latest version, use a git command as follows:

```
git clone sicuser@quantum.utep.edu:pr.git
```

where the password for this repository is available by communicating with the repository administrator. (Currently, you can contact Prof. Rajendra R. Zope, rzope(at)utep.edu.)

Once you have obtained the code from the repository, go to the directory containing the source files. You will need to edit three blocks of lines in the makefile to suit your needs.

### First block: parallel or serial

```
# set Y or N
# Parallel compilation
MPI=Y
# Group calculation
GROUP=N
```

In this block, the user specifies whether the compilation will be for a parallel (MPI=Y) or serial (MPI=N) version of the code. When using a parallel version (MPI=Y), the user can choose to use a multi-level parallel scheme by setting GROUP=Y. This parallelizes over the orbitals in calculating SIC potentials, and also over the grid points. This option can deliver a greater speed up than the single-level parallel scheme, but it is not recommended for new users.

Note:

When GROUP=Y, the user should edit the “igroup” file in the run directory. This file should contain an integer that is an even divisor of the number of orbitals. For example, for 100 orbitals, igroup might be 10 or 20, but not 15.

### Second block: compiler choices

```
# COMPILERS
CC = gcc
FC = mpif90
FF = mpif90
# COMPILER FLAGS
CFLAGS = -O3 -mmodel=medium
FFLAGS = -O3 -mmodel=medium
LFLAGS = -O3 -mmodel=medium
```

The second block is to specify the compilers and their flags used during the compilation. A number of routines require static allocation of arrays. The code needs to be compiled for appropriate array sizes for the system under study and these static parameters are listed in the file called PARAMA2 (*vide infra*)

An example of the compilers used for NERSC is below.

CC = cc

FC = ftn

FFF = ftn

### Third block: linking options

# Linking options

\$(FFF) \$(LFLAGS) \$(OBJ) -o \$(BIN) -llapack -lblas \$(LIBS)

This block specifies the libraries used for linking, where it is recommended to use optimized BLAS and LAPACK libraries if they are available on your platform to achieve the best performance.

Edit PARAMA2 for static parameters :

parameter	brief explanation
<i>MAX_PTS</i>	maximum size of integration grid
<i>MX_SPH</i>	needed to generate the integration mesh
<i>MAXUNSYM</i>	maximum number of orbitals for an atomic basis
<i>NDH</i>	maximum total basis set size
<i>NDH_TOT</i>	maximum number of Hamiltonian matrix elements
<i>MAX_OCC</i>	maximum number of occupied states
<i>MX_GRP</i>	maximum size of symmetry group

An example for serial compilation on a laptop (in this case a mac) is shown below.

**First block:** Compile the serial version

MPI=N

GROUP=N

**Second block:** use gcc and gfortran compilers

#COMPILERS

CC = gcc

FC = gfortran

FFF = gfortran

#COMPILER FLAGS

CFLAGS = -O3

FFLAGS = -O3

LFLAGS = -O3

**Third block:** Use the linking options under Fedora (Quantum/Luis local)

\$(FFF) \$(LFLAGS) \$(OBJ) -o \$(BIN) \$(PCM\_LIBS) \$(EFP\_LIB) -llapack -lblas \$(LIBS)

Use the *make* command on the command line to compile FLOSIC. If the compilation was successful, an executable file titled **nrlmol\_exe** will be created.

## QUICK START GUIDE TO RUNNING A FLOSIC CALCULATION

Basic input files for FLOSIC are called “CLUSTER” and “FRMORB”. The files require minimal input from the user and most parameters required for the calculations are set with default values. You can specify some parameters that control the calculations in the file called “NRLMOL\_INPUT.DAT”. A version of this file with default values is created in the run directory if it otherwise does not exist. We will discuss the details of the parameters in the **NRLMOL\_INPUT.DAT** in a later section.

For now let’s work on setting up of the **CLUSTER** file. It is the minimal input needed to run the FLOSIC code.

An example CLUSTER file is shown below for the LiH molecule. The format is described in more detail in tutorial 1.

```
GGA-PBE*GGA-PBE          (DF TYPE EXCHANGE*CORRELATION)
NONE                      (TD, OH, IH, X, Y, XY, ... OR GRP)
2                          (NUMBER OF INEQUIV. ATOMS IN CH4)
0.0000  0.0000  0.0000 3 ALL
0.0000  0.0000  3.0139 1 ALL
0.0  0.0                  (NET CHARGE AND NET SPIN)
-----OR-----
@XMOL.DAT
IF YOU WISH TO START FROM AN XYZ XMOL FILE
```

To run a FLOSIC calculation, a FRMORB file is required, which contains positions for FODs. An example FRMORB file is shown below for the LiH molecule defined in the CLUSTER file above.

```
2 2
0.000 0.000 0.000
0.000 0.000 2.500
0.000 0.000 0.000
0.000 0.000 2.500
```

The first line states how many up spin FODs, and how many down spin FODs are used in the calculation. There are 2 up FODs, so the next 2 lines (lines 2-3) contain x, y, & z coordinates for up spin FODs. There are also 2 down FODs, so the last 2 coordinates are for down spin FODs (lines 4-5).

To create starting FOD positions, you can simply place them by hand or use an FOD generator such as the fodMC code to create them. You can obtain a copy of the fodMC and documentation [here](#).

Before starting a FLOSIC calculation, make sure that the CALCTYPEV option in the NRLMOL\_INPUT.DAT file is set to ‘SCF-ONLY’ instead of ‘LBFGS’. This ensures that atomic geometries will not be updated after each self-consistent energy evaluation. This is recommended practice when FOD positions are being optimized.

Try running the code using the example CLUSTER and FRMORB files given above. The final energy (EDFT+SIC) in the SUMMARY file after a completed self-consistent calculation should be **-8.052832898** Hartree.

You'll notice the FOD positions in FRMORB have been moved, as they are being optimized, similar to an atomic geometry optimization at the DFT level. Previous FOD geometries are contained in the records file, along with FOD forces.



## **EXCHANGE CORRELATION FUNCTIONALS**

The exchange-correlation functional is specified in the CLUSTER and/or SYMBOL files. The user can currently choose to use LDA (PW92 version) or PBE functionals. The functional string needed to run them is as follows:

LDA-PW91\*LDA-PW91

GGA-PBE\*GGA-PBE

If the user wants to run GGA-PBE exchange-only, for example, the string will be :

GGA-PBE\*GGA-NONE

## CONTROLLING A CALCULATION: THE NRLMOL\_INPUT.DAT FILE

The **NRLMOL\_INPUT.DAT** is an auxiliary input file that allows setting of control parameters for calculation. If it does not exist, then it will be created by the FLOSIC executable using default values.

The NRLMOL\_INPUT.DAT with default values of parameters as of today (August 15, 2019) is given below.

```
# Put Y,N or number next to the equal sign to determine execution
# Don't forget the quotation marks for the letters
# All variables in this list end with V

&input_data
ATOMSPHV      = 'N'
BASISV        = 'DEFAULT' ! Specify basis for calculation(basis.txt)
CALCTYPEV     = 'LBFGS'
DFTD3V        = 'N' ! Set to Y to do include Grimmes DFT-D3 dispersion
DIAG1V        = 1  ! diagonalization to use on regular arrays (diagge.f90)
DIAG2V        = 1  ! diagonalization to use on packed arrays (diag_dspgv.f90)
DIAG3V        = 0  ! diagonalization to use on parallel (sdiagge_n.f90)
DMATV         = 'N' ! Create/use/mix density matrix
DOSOCCUV      = 'N' ! Controls wether to calculate density of states (only in DFA)
FIXMV         = 'N' ! Fix spin moment
JNTDOSV       = 'N' ! This calculates jonit density of states (only in DFA)
MAXSCFV       = 100 ! Maximum SCF iterations
MIXINGV       = 'P' ! (H)amiltonian (P)otential (D)ensity matrix mixing
NONSCFV       = 'N' ! Set to Y to do a non SCF calculation
NONSCFFORCESV = 'N' ! Set to Y to calculate forces in a non SCF calculation
NWFOUTV       = 10 ! Write WFOUT file for every N-th iteration
POPULATIONV   = 'N' ! Population analysis
RHOGRI DV     = 'N' ! Set to Y to execute RHOGRI DV
SCALEDLBFGSV  = 'Y' ! Set to Y to scaled LBFGS (only in SIC)
SCFTOLV       = 1.0D-6 ! SCF tolerance
SPNPOLV       = 'N' ! Run spin polarized calculation from CLUSTER
VERYFINEMESHV = 'N' ! Set to Y to use very fine mesh
SYMMETRYV     = 'N' ! Set to Y to detect symmetry
WFGRIDV       = 'N' ! set to Y to write orbitals in cube format (only in DFA)
WFFRMV        = 'N' ! set to Y to write Fermi orbitals in cube format (only in SIC)
&end
```

A large number of entries related to calculation of certain properties are given as yes or no. Here **N** means *NO*. If **Y** then the calculation of a given property is requested. Note that certain calculations such as, for example, the calculation of joint density of states is often useful only at the final converged geometry.

Most of the variables are already explained briefly in the NRLMOL\_INPUT.DAT file. Below we add some comments on a few of them.

1. **ATOMSPHV**: Set to **Y** to calculate charge and spin charge in each inequivalent atom integrated over a sphere of specified radius.
2. **BASISV**: Default value is **DEFAULT** which refers to the NRLMOL basis optimized for the PBE functional. This variable specifies which basis set is to be used. Users interested in using standard sets from the literature should look up basis.txt in the basis subdirectory of the directory containing the source files. This file lists all the basis sets available. The user needs to specify only the first string before the “.” from the entries in basis.txt. For example, if the user wants to use 6-31G basis then he should replace **DEFAULT** with 6-31G.
3. **CALCTYPEV**: Default is **LBFGS**. Available choices: **SCF-ONLY**, **CONJUGATE-GRADIENT**.
  - **SCF-ONLY** This choice is used when the user is not interested in an atomic geometry optimization. Atomic forces are not computed with this setting.
  - **CONJUGATE-GRADIENT** : Atomic geometry optimization is done using the conjugate-gradient algorithm.
4. **DIAG1V** = Default is **1** This variable allows different diagonalization algorithms to use for diagonalization of the Hamiltonian.

**The options for this variable are:**

- 0) **DSGVX** (LAPACKL: computes selected eigenvalues, and optionally eigenvectors).
  - 1) **DSGVD** (Default if matrix size is below 100). Uses a divide and conquer algorithm to compute eigenvectors.
  - 2) **DSGV** (This is the slowest of the available) option.
5. **DIAG2V** = Default is **1** This variable allows diagonalization of the Hamiltonian using packed storage format for memory savings. This is useful for large system sizes.

Available options are:

- 0) **DSPGVX** (LAPACK: computes selected eigenvalues, and optionally eigenvectors).
- 1) **DSPGVD** (Default if matrix size is below 100). Uses a divide and conquer algorithm to compute eigenvectors.
- 2) **DSPGV** (QR- factorization. This is the slowest of the available) option.

The option 1 (DSPGVD) is the fastest if all eigenvectors are required otherwise use the default 0. Note that FLOSIC doesn't use **SCALAPACK**

6. **POPULATIONV** = Default is **N** When it is set to **Y**, Mulliken and Lowdin Population analysis calculations are run. Note that this is available only for spin polarized calculations.
7. **RHOGRIDV** = Default is **N** When set to **Y**, it will generate a CUBE file for visualization of total and spin density (spin up - spin down).
8. **SCALEDLBFGSV** = Default is **Y** This indicates the FOD optimization algorithm. Here scaled LBFGS is set as the default FOD optimization routine. Conjugate Gradient is used when this option is set to 'N'.

## FLOSIC MESH

In FLOSIC, the mesh is generated by specifying the accuracy of various integrals. The mesh is refined until the specified accuracy of the integrals is accomplished. In this sense, the mesh is variational. For further details about how the mesh is generated please refer to the paper by Pederson and Jackson(Phys. Rev. B 41, 7453). There are two types of meshes used in the calculation. One is the radial mesh used for calculations within the atomic spheres and the other is the interstitial mesh (mesh between atomic spheres). The defaults for these are 1.E-06. The mesh parameters are stored in the MESHDAT file. The default MESHDAT file for a DFT level calculation will look like the one below. It is possible to change the default mesh. The parameters used in creating the mesh are listed in the MESHDAT file. The default MESHDAT file for a DFA calculation is as follows:

F											line 1
0.10000E-06	1.2000										line 2
6											line 3
0.20000	0.40000	0.60000	1.0000	1.6000							line 4
4											line 5
2.1000	10.100	18.100									line 6
2	1	3	5	5	7	9	11	19	21		line 7
4	1	3	5	5	7	9	11	19	21		line 8
4	1	3	6	5	7	9	11	19	21		line 9
6	1	3	6	7	7	9	11	19	21		line 10
0.10000E-06	1.2000										line 11
2.0000	8										line 12
2.0000											line 13

The default MESHDAT file for a FLOSIC calculations is shown below:

As noted above are two types of meshes used in the calculation. One is the radial mesh use for calculations within atomic spheres and the other is the interstitial mesh.

Line 1:

Line 2: The numbers in this line pertain to the radial mesh. The radial mesh is constructed to give integrations involving various Gaussians from short to long range. The small number is the error allowed in integration by the mesh. The second number, 1.2, separates the exponentials of the Gaussians which are tested i.e. the n+1st exponential is 1.2 times nth exponent. Decreasing the error tolerance will result in increase in number of mesh points and better integrals.

Line 3: The atomic sphere is divided into different concentric regions with different numbers of mesh points. The number of such radial zones is written in line 3.

Line 4: The outer radii of the radial zones.

Line 5: The mesh may be different for different atoms. For example, heavier atoms generally need denser radial meshes near the nucleus. Hence the periodic table (upto Z=56) is divided in 4 types of meshes.

Line 6: This line shows the 4 types of meshes for atoms : first one for  $Z < 2.1$ , second one for  $Z < 10.1$  and third one for  $Z < 18.1$  and the last one for  $Z > 18$ .

Line 7,8,9,10: These four lines contain the parameters for each type of atomic mesh. The numbers are

**NPATS, NPIST, NTHET, NPHI, LMAX** FOR EACH RADIAL ZONE

**NPATS :**

**NPIST, NTHET, NPHI :** Before creating the mesh, the space is divided into boxes such that each atom is contained in one box. Then a sphere around the atom is assumed and the radial mesh is created. The space at the box corners between the atomic sphere and the rectangular box is divided in a different mesh. The parameters NPIST, NTHET and NPHI correspond to the mesh at the box corners.

**LMAX :** LMAX for the each radial zone. This mesh will integrate a function of the type  $r \exp(-r^2)$  within the given error limit in this region. Increasing LMAX results in a more refined mesh.

Line 11 : This line contains the same parameters as line 2 but for the interstitial mesh. Decreasing the error limit results in more accurate integrals in the interstitial region.

Line 12: Cutfac, mx1d

**CUTFAC:** If a box transforms into itself due to symmetry, it will be split if it is larger than cutfac times the distance to the closest atom.

**MX1D:** max. number of points in a one-dimensional interstitial partition

Line 13: Splrat

**SPLRAT:** largest allowed ratio for: the size of an atomic box divided by the distance of another atom to the box boundary

The user can set these parameters to exercise control over the quality of calculation. For example the default mesh in FLOSIC code has been tested for FLO-SIC-LSDA calculations and is capable of providing accurate results, while the mesh required in a DFT-LDA calculation does not have to have many mesh points as in a FLO-SIC-LDA calculation. Above values are default. *Warning* When taking energy differences and comparing the energies it is advisable to ensure that same quality of mesh was used in all pertaining calculations.

## INPUT AND OUTPUT FILES

As mentioned above, the minimal set of input files required for a FLOSIC calculation includes CLUSTER and FRMORB. Once the code is run, a number of additional files are generated. Some of these, like NRLMOL\_INPUT.DAT (described earlier) can be edited to control how subsequent calculations run. The following is a list of important input/output files and a brief description of their contents:

- LIST of FILES
- 1. ATOMSPHNN: Charge and spin charge in each inequivalent atom integrated over a sphere of specified radius.
- 2. CLUSTER : Main input file.
- 3. DIPOLE : Contains x, y, z components of dipole moments in the atomic units.
- 4. EVALUES : The canonical eigenvalues for the current step in an SCF calculation. If symmetry is used in the calculation, the eigenvalues are sorted by symmetry, and also by spin, if the calculation is spin-polarized. A list of eigenvalues and the occupation of the corresponding orbital is given at the end of the file.
- 5. EVALNNN : The eigenvalues, their spin, symmetry representation, degeneracy and occupancy, Fermi energy for each iteration number NNN.
- 6. FRCOUT : Total energy, forces on each atom, dipole moment, applied electric field.
- 7. FRMORB : number of up spin FODs and down spin FODs, followed by the up spin FOD positions, then down spin FOD positions.
- 8. fande.out : Iteration, total energy, square root of the sum of the squares of FOD forces, max FOD force.
- 9. GEOCNVRG : Convergence criteria, Total energy , largest atomic force, information about atomic geometry optimization. Is written after every complete SCF cycle.
- 10. HISTORY: Contains history of the geometry optimization. To see the energy as function of optimization step do 'grep -i TR HISTORY' in the directory.
- 11. ISYMGEN: Information about the basis set. (For an in depth description, see chapter 8.)
- 12. records : copy of the FRMORB file used during each iteration, followed by the FOD forces on those FODs.
- 13. RHOTOT : Total density on a specified grid. Is written in Gaussian cubic format.
- 14. RHOSPN : Spin density on a specified grid. Is written in Gaussian cubic format.
- 15. RUNS : Control restart of calculation. Calculations can be restarted from Hamiltonian (**HAMOLD**), wavefunctions (**WFOUT**), or potential (**COUPOT**).

16. SUMMARY : Total energy, electronic charge, kinetic energy and trace of hamiltonian for each iteration.
17. SYMBOL : The exchange-correlation functional is specified on the first line. Atomic coordinates and bias potentials for each atom are stored for each iteration of an atomic geometry optimization. The last line allows the use of a more extensive default basis set (to use, set EXTRABASIS=1). SYMBOL includes essentially the same information as the CLUSTER file. If the SYMBOL file exists, CLUSTER is not read.
18. XMOL.DAT : Geometry in the xyz format. The first column contains atomic numbers. The atomic coordinates are in Angstrom.
19. XMOL.xyz : Geometry in the xyz format. The first column contains atomic labels. The atomic coordinates are in Angstrom. Use **JMOL** to visualize this.

Disclaimer: Only the primary input and output files are explained here. For more information about additional files, please contact a FLOSIC team member or post a question to the FLOSIC listserv (See Ch. 13).

## BASIS SETS: THE ISYMGEN FILE

The basis set information is in the ISYMGEN file which is created when you run FLOSIC. This file contains the basis set for each atom type appearing in the calculation. For each orbital of a given atom, same set of primitive Gaussians is used. For example, the default basis set for carbon is written as below :

The following is the Pederson-Porezag[3] (NRLMOL default) basis in the **ISYMGEN** . The FLOSIC code allows use of other basis sets such as : 6-31G\*, 6-311G\*\*, STO-3G, TZVP, DGDZVP etc. For a full list of available basis sets, see the file basis.txt (PATH\_TO\_FLOSIC/basis/basis.txt). The Pederson-Porezag (NRLMOL default) is specially optimized for the PBE functional, and roughly corresponds to quadruple zeta quality.

```

1      TOTAL NUMBER OF ATOM TYPES
6  6    ELECTRONIC AND NUCLEAR CHARGE
ALL     ALL-ELECTRON ATOM TYPE
1      NUMBER OF ATOMS OF TYPE CAR
ALL-CAR001
EXTRABASIS  CONTROLS USAGE OF SUPPLEMENTARY BASIS FUNCTIONS
12      NUMBER OF BARE GAUSSIANS
5  4  3  NUMBER OF S,P,D FUNCTIONS
0  0  1  SUPPLEMENTARY S,P,D FUNCTIONS
0.2221336090D+05    0.3331736980D+04    0.7579013460D+03
0.2145437190D+03    0.6992488940D+02    0.2508613490D+02
0.9591041750D+01    0.3802455710D+01    0.1489185420D+01
0.5748765300D+00    0.2149473200D+00    0.7720965000D-01

0.1979224900D+00    0.3699897740D+00    0.6364461530D+00
0.1012493110D+01    0.1448078710D+01    0.1717368870D+01
0.1493193200D+01    0.6898716060D+00    0.8607224650D-01
-0.1656669520D-02    0.3776603340D-03    -0.4710534310D-04

-0.4500525960D-01    -0.8462105180D-01    -0.1449656410D+00
-0.2353560110D+00    -0.3421536850D+00    -0.4459512370D+00
-0.4526397090D+00    -0.3221641360D+00    -0.1298841970D-01
0.2013547140D+00    0.1276991250D+00    0.1413546650D-01

0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
0.1000000000D+01    0.0000000000D+00    0.0000000000D+00

0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
0.0000000000D+00    0.0000000000D+00    0.0000000000D+00

```

(continues on next page)



ELECTRONS  
WFOUT

calculation)

The third line specifies the type of calculation :

ALL for all-electron, BHS for BHS pseudopotential, TAB for tabulated user-supplied pseudopotential.

The third and fourth lines specify the number of total such atoms in the geometry and their symbols in the SYMBOL file. EXTRABASIS =1 in SYMBOL file will signal the program to use the supplementary basis functions. Then comes the number of primitive Gaussians followed by the number of the s, p and d -type contracted Gaussians. The number of supplementary functions of s, p and d type are written next. These informations are followed by some blocks of numbers. The first block lists the exponents of the primitive Gaussians. This is then followed by  $N_s$  blocks where  $N_s$  is the number of contracted s-type Gaussians. The first block is the coefficients multiplying the primitive Gaussians for the 1s contracted Gaussian, the second block is for 2s Gaussian and so on. After the  $N_s$  number of such blocks, comes the  $N_p$  blocks corresponding to the p-type contracted Gaussians followed by similar  $N_d$  number of d-type Gaussians. In the example for carbon atom above, the 1s and 2s contracted Gaussian is a linear combination of the all the primitive Gaussians whereas the higher unoccupied s orbitals are taken as single long-range Gaussians. Similarly for p orbitals where only the 2p orbital is occupied. These are then followed by similar blocks corresponding to the supplementary functions. By setting EXTRABASIS = 1 in the SYMBOL file, the additional basis functions described above are added to the calculation. These are sometimes added when calculating dipole moments and polarizabilities.

## TUTORIALS

## 9.1 Tutorial 1: Molecular geometry optimization at the DFT level

Very simple tutorial to get acquainted with electronic structure calculations using FLOSIC. This tutorial will explain how to run FLOSIC for molecular geometry optimizations using DFT.

The **CLUSTER** file is the main input file of FLOSIC. It contains the minimal information to set up a calculation. For this tutorial, we will use a CH<sub>4</sub> molecule, which uses a **CLUSTER** file like the one shown below:

```
GGA-PBE*GGA-PBE          (DF TYPE EXCHANGE*CORRELATION)
NONE                      (TD, OH, IH, X, Y, XY, ... OR GRP)
5                          (NUMBER OF INEQUIV. ATOMS IN CH4)
 0.0000  0.0000  0.0000 6 ALL
 1.1860  1.1860  1.1860 1 ALL
 1.1860 -1.1860 -1.1860 1 ALL
-1.1860  1.1860 -1.1860 1 ALL
-1.1860 -1.1860  1.1860 1 ALL
0.0 0.0                  (NET CHARGE AND NET SPIN)
-----OR-----
@XMOL.DAT
IF YOU WISH TO START FROM AN XYZ XMOL FILE
```

We will now describe the input structure of this file.

The first line is **GGA-PBE\*GGA-PBE**. It means that the exchange-correlation interactions in the systems are modeled within the generalized gradient approximation (GGA) using the Perdew-Burke-Ernzerhof (PBE) parametrization. This is the default functional used in NRLMOL. A few other functionals are also available.

The second line is **NONE**. It refers to point group symmetry of the molecule. For the purposes of the tutorial, we will not enforce symmetry. If you would like to use symmetry, a symmetry (TD,OH, etc.) can be selected in place of **NONE**. In these cases, the code will create a GRPMAT file containing the appropriate symmetry operations (each represented by a 3x3 matrix). If you would like to use symmetry operations directly from an existing GRPMAT file, replace **NONE** with **GRP**.

The third line contains **5**. It specifies the number of inequivalent atoms in the calculation. We're running a CH<sub>4</sub> calculation. So the number of atoms is 5 (1 C and 4 H).

The line following third line contains the cartesian position in atomic units of the Carbon atom. Following the xyz coordinates is the atomic charge. The example listed is for Carbon whose atomic number is 6. The string **ALL** means include all (that is 6 in this case) electrons into the calculation. The next 4 lines are the hydrogen atoms, which follow the same format.

The ninth line in the example file has two fields, charge and moment. The first field is **0.0** which means to perform the calculation for the neutral molecule. If it is **1** then the calculations will be performed for a cation of CH<sub>4</sub>. The next

field which is also **0.0** in this example corresponds to the number of unpaired electrons in the system. CH<sub>4</sub> is a closed shell system, so it has no unpaired electrons. Lines after the Charge and Moment line are ignored.

Now, create an empty directory and execute the code inside of it. Multiple files should be created, including a CLUSTER file. Copy the input from this example into the file called CLUSTER, replacing the default text. Change the CALCTYPE option in the NRLMOL\_INPUT.DAT file to "LBFGS" and run the calculation for CH<sub>4</sub>.

```
$ PATH_TO_FLOSIC/nrlmol_exe
```

Open the **GEOCNVRG** file. If you have done everything correctly then it should be **-40.466969** Hartree.

A new atomic geometry will be appended to the SYMBOL file. SYMBOL is created from the data in CLUSTER. The new geometry was created by a gradient optimization routine (either LBFGS or CONJUGATE-GRADIENT). The file **FRROUT.G0** contains the atomic forces for the previous atomic geometry. Running the code again will carry out a calculation at the updated molecular geometry and a new total energy and new atomic forces will be computed. A new update of the atomic coordinates will also be written into SYMBOL. Repeating this process several times will result in a local minimum energy geometry to be reached. This happens when the maximum force falls below the criterion set in GEOCNVRG.

## 9.2 Tutorial 2: a FLOSIC calculation for H<sub>2</sub>

In this tutorial we will learn to do a simple FLOSIC calculation. We choose the case of the H<sub>2</sub> molecule. Again, as a reminder the **CLUSTER** file is the main input file of FLOSIC. It contains the minimal information to set up the calculation. See below the CLUSTER file for the hydrogen molecule.

```
GGA-PBE*GGA-PBE      # Exchange-correlation parametrization
NONE                  # Point group of system
2                     # No. of atoms
0.0 0.0 0.5          1 ALL  # x,y,z coordinates, Atomic number, ALL means all electron
0.0 0.0 -0.5          1 ALL  # x,y,z coordinates, Atomic number, ALL means all electron
0.0 0.000             # Charge and Moment
```

We will now describe the input structure of this file. The first line is **GGA-PBE\*GGA-PBE**. It specifies that the electron-electron exchange and correlation interactions in the systems are modeled within the generalized gradient approximation (GGA) using the Perdew-Burke-Ernzerhof (PBE) parametrization. This is the default functional used in FLOSIC. A few other functionals are also available.

The second line is **NONE**. It refers to point group symmetry. NONE means that there is no symmetry. To use symmetry point groups from a GRPPMAT file, replace this with **GRP**. A few selected symmetry groups such as Ih, Td, Th, C<sub>3v</sub> etc. can also be provided here. In this case the GRPMAT corresponding to specified point group will be automatically generated by NRLMOL. Even though the H<sub>2</sub> molecule possesses symmetry we will perform the calculation without symmetry, so keep the **NONE** line as it is. A GRPMAT file will be created automatically and will contain the identity matrix only.

The third line specifies the number of inequivalent atoms in calculations. Since we are not making use of point group symmetry, the number of inequivalent atoms is the same as total number of atoms, which is 2 in this case.

The lines following the third line contain information for each of the inequivalent atoms. The first three numbers are the cartesian positions in atomic units, its atomic number, and string ALL. In this example, we have placed one hydrogen atom at (0,0,Z) and the second one at (0,0,-Z). You can put the atoms anywhere. Atomic positions should be given in atomic units, Bohr. Following the xyz coordinates is the atomic number. The example listed is for hydrogen whose atomic number is 1. The string ALL signifies that the calculations are to be performed at the all-electron level. It is also possible to use pseudopotentials. Only the BHS pseudopotentials are hardwired into the code. It is also possible to use user-supplied (numerical) pseudopotentials also, but requires more work and is not recommended for beginners.

The last line in the example has two fields. The first field is 0.0 which means perform the calculation for the neutral molecule. If it is 1 then the calculations will be performed for singly charged cation of the system. The next field, which is 0 in this example, corresponds to the number of unpaired electrons in the system. There are no unpaired electrons in this example. Lines after Charge and Moment line are ignored.

A file called **FRMORB** is also required, which contains the FOD positions. An example FRMORB file for H2 is shown below.

```
1 1
0.00 0.00 0.00
0.00 0.00 0.00
```

The first line contains two fields. The first is the number of spin up FODs (N) and the second is the number of spin down FODs (M). For H2, we only have one electron in each spin channel. The up FOD coordinates (x,y,z) are the following N lines. After that is the coordinates (x,y,z) of the M spin down FODs

To avoid optimizing atomic geometries, change the CALCTYPE field in NRLMOL\_INPUT.DAT from LBFSGS to SCF-ONLY. This is recommended for FLOSIC calculations when the FODs are being optimized.

Now, copy the example input files called CLUSTER and FRMORB into an empty directory. Run the calculation for the H2 molecule using the following command at the prompt.

```
$ PATH_TO_FLOSIC/nrlmol_exe > print
```

Now, browse through the file SUMMARY and look at the energies printed at each iteration of the SCF cycle. You should see that the minimum total energy is reached at self-consistency. Also, look at the EVALUES file in which Kohn-Sham eigenvalues and occupation numbers are printed.

The FOD forces are displayed in the records file, which has the same format as FRMORB, followed by the FOD forces in the same order.

The fande.out file contains the iteration, total DFT+SIC energy, square root of the sum of the squares of the FOD forces, and the max FOD force. When optimizing FODs, this is a good file to check for the convergence of FOD forces.

After completing a self-consistent calculation, the FOD forces are used in a gradient optimization scheme to update the FOD positions. These are written into the FRMORB file. Re-running the code will cause another self-consistent calculation to be performed, using the updated FOD positions. A new total energy and new FOD forces will be calculated, and the FOD positions will again be updated. Repeating this process will result in the optimization of the FOD positions. Convergence can be gauged by the size by the largest FOD force. When this drops below a chosen convergence criterion, the FODs are optimized.

To further optimize FODs, repeatedly run the code until the calculation is optimized to your criterion. A simple iterative loop can help with this process.

For the example of H2, there is only one FOD of each spin. Placing the FODs at any position in space for such a case will give the same energy and the force on the FOD will therefore be zero.

### 9.3 Tutorial 3: FOD optimization of a water molecule

This tutorial explains how to do a FODs optimization taking the example of the water molecule. It is recommended to run a DFT calculation first to get the DFT optimized density. To do this, we start from the usual CLUSTER file:

```
GGA-PBE*GGA-PBE      (DF TYPE EXCHANGE*CORRELATION)
NONE                  (TD, OH, IH, X, Y, XY, ... OR GRP)
3                      (NUMBER OF ATOMS)
0.000  0.000  0.0  8  ALL  (R, Z, ALL-ELECTRON)
1.443  1.128  0.0  1  ALL  (R, Z, ALL-ELECTRON)
```

(continues on next page)

(continued from previous page)

```
-1.443  1.128  0.0  1  ALL  (R, Z, ALL-ELECTRON)
0.000  0.000                (NET CHARGE AND NET SPIN)
```

In the NRLMOL\_INPUT.DAT file we change CALCTYPEV from 'LBFGS' to 'SCF-ONLY' to switch off the molecular optimization. We also turn on the unrestricted calculation setting SPNPOLV = 'Y'. Then we run NRLMOL in a new directory with only CLUSTER and NRLMOL\_INPUT.DAT files:

```
$ PATH_TO_FLOSIC/nrlmol_exe > print.DFT
```

All the normal NRLMOL outputs are generated. If the calculation finished correctly, we should have the following RUNS file

```
0      1      ITBEG, NCALC
4      4      START: 0=SCR.NUC, 1=HAM, 2=POT, 3=LSF, 4=WFUNC, 5=WFUNC_FRAG
0      START HAMILTONIAN IS INTERPOLATED: 0=NO, 1=YES
```

The second number in the first line may be different if a molecular optimization was carried out. It is important to check that we have 4 4 in the second line. This tells FLOSIC to use the wave function stored in WFOUT to start the next calculation.

Now, we have to include the FOD positions. Here, we can use the Monte Carlo FOD generator **fodMC**. For that, we copy XMOL.xyz into a file named **system** and edit the second and last lines as shown:

```
3
angstrom  fixls
O      0.000000  0.000000  0.000000
H      0.76360  0.59691  0.000000
H     -0.76360  0.59691  0.000000
con_mat
```

It might have an empty line at the end. The fodMC code and documentation can be found [here](#). When fodMC is executed 3 files are generated: **CLUSTER**, **FRMORB** and **Nuc\_FOD.xyz**. Thus, it will overwrite our **CLUSTER** file if executed in the same directory. In our case, it does not matter because whenever the **SYMBOL** file is present **CLUSTER** is ignored. **FRMORB** contains the FODs in Bohrs for the unrestricted calculation (first the up, then the down). You should get something like this:

```
5      5
0.000000000000000000  0.000000000000000000  0.000000000000000000
1.2265460953713494    0.95879731507086485    0.000000000000000000
-1.2265460953713494    0.95879731507086485    0.000000000000000000
9.8690853618931581E-005 -0.84709659045951247    -1.0165159441533127
-1.5276582823942571E-005 -0.84709659045951247    1.0165159488293354
0.000000000000000000  0.000000000000000000  0.000000000000000000
9.8690853618931581E-005 -0.84709659045951247    -1.0165159441533127
-1.5276582823942571E-005 -0.84709659045951247    1.0165159488293354
1.2265461383759642    0.95879734868779065    0.000000000000000000
-1.2265461383759642    0.95879734868779065    0.000000000000000000
```

If you do not have fodMC, you can copy-paste the previous **FRMORB**. It is important to always check how the FODs are distributed. For that, you can visualize **Nuc\_FOD.xyz** with the software of your choice. The spin up FODs have the label X and the down FODs have the symbol He by default, so be careful if the He atom is included in your system.

Running the FLOSIC code again in this directory will now cause a FLO-SIC-PBE calculation to be run. (The existence of FRMORB is the flag for running a FLO-SIC calculation).

\$ PATH\_TO\_FLOSIC/nrlmol\_exe > print.001

If everything goes well, we get now all the FLOSIC output files. Check the **SUMMARY** file. It should look like this (note that some columns of this file do not fit on the page):

IT	TRACE	ETOT	EKIN+ENONLOC	CHARGE
	EDFT+SIC	LOWEST		
1	-42.416153386	-76.326968147	76.469793374	9.999999301
	-76.326968147	0.000000000		
2	-40.529422883	-76.287479323	74.155137275	9.999999484
	-76.287479323	-76.326968147		
3	-41.037801666	-76.360861763	75.120435771	9.999999577
	-76.360861763	-76.287479323		
4	-41.547677518	-76.386605548	76.071092878	9.999999652
	-76.386605548	-76.360861763		
5	-41.579433885	-76.387012272	76.156784862	9.999999660
	-76.387012272	-76.386605548		
6	-41.537373241	-76.387091483	76.105263133	9.999999659
	-76.387091483	-76.387012272		
7	-41.535132409	-76.387091342	76.101118384	9.999999659
	-76.387091342	-76.387091483		
8	-41.541270575	-76.387093972	76.111112778	9.999999661
	-76.387093972	-76.387091342		
IT	TRACE	ETOT	EKIN+ENONLOC	CHARGE
	EDFT+SIC	LOWEST		
1	0.000000000	-76.387093972	76.111112778	9.999999661
	-76.325884480	0.000000000		
2	-47.383872787	-76.377766430	75.943917202	9.999999707
	-76.336806819	-76.325884480		
3	-47.434397301	-76.376079958	76.027004027	9.999999708
	-76.337098863	-76.336806819		
4	-47.441559320	-76.375691097	76.055957777	9.999999706
	-76.337109578	-76.337098863		
5	-47.324136604	-76.375845538	75.998150088	9.999999702
	-76.337049496	-76.337109578		
6	-47.312036177	-76.376132863	76.000962346	9.999999700
	-76.337038707	-76.337049496		
7	-47.315234163	-76.376185290	76.002987681	9.999999700
	-76.337041096	-76.337038707		
8	-47.332770384	-76.376285169	76.024925697	9.999999700
	-76.337087551	-76.337041096		
9	-47.303951079	-76.376194186	75.999998287	9.999999700
	-76.337026990	-76.337087551		
10	-47.374868090	-76.376350868	76.064567671	9.999999702
	-76.337132117	-76.337026990		
11	-47.381393225	-76.376294921	76.083303549	9.999999703
	-76.337128855	-76.337132117		
12	-47.365402178	-76.376255647	76.070377576	9.999999703
	-76.337133358	-76.337128855		
13	-47.358580387	-76.376246217	76.060102340	9.999999702
	-76.337132539	-76.337133358		
14	-47.369783782	-76.376261648	76.069168121	9.999999702
	-76.337133952	-76.337132539		

The first block is for the normal DFT calculation with the 3rd and 6th column exactly the same (no SIC). The second

block is for the FLOSIC run. In the standard output file – renamed print.001 in this example (see the execution line above) – search for the word ITERATION.

```

ITERATION    1
=====
READING OLD WAVEFUNCTIONS FROM FILE WFOUT
MREP =          1
N_OCC, NBASF =          5          73          73
N_OCC, NBASF =          5          73          73

```

There it states that the wave function is read from WFOUT and gives the occupied and total number of orbitals. After giving the occupancies, it calculates the Lowdin overlap eigenvalues.

```

LOWDEN OVERLAP EIGENVALUES:
  0.751388    0.791811    1.06742    1.07216    1.31722
BACK FROM LOWSIC
CALLING DIAGGE:    1000          5
LOWDEN OVERLAP EIGENVALUES:
  0.751388    0.791811    1.06742    1.07216    1.31722

```

These values correspond to the eigenvalues of the overlap matrix formed from the Fermi orbitals. This is diagonalized in the Lowdin process. When one or more of these eigenvalues are smaller than 1E-08, it means that two or more of the Fermi orbitals are identical and the calculation stops with a message of bad FOD positions. At the end of the SCF cycle we get the following summary of energy contributions. Some of them are in the **SUMMARY** file too.

```

SUMMARY OF ENERGY CONTRIBUTIONS:
=====
TOTAL ENERGY:          -76.337134
NUCLEAR REPULSION:       9.082196
LOCAL POTENTIAL:        -198.922706
MEAN-FIELD COULOMB:      46.647189
NONLOCAL POTENTIAL:      0.000000
KINETIC:                 76.069168
LOCAL EXCHANGE:          -8.108217
LOCAL CORRELATION:       -0.660695
NONLOCAL EXCHANGE:       -0.816267
NONLOCAL CORRELATION:     0.333071
EXTERNAL ELECTRIC FIELD:  0.000000

```

Once self-consistency is reached, the FOD forces are calculated and FOD positions are updated in **FRMIDT** (and/or in **FRMORB**) using the chosen optimization method. The default is scaled LBFGS, but it sometimes gets stuck or fails to obtain the next step. In those cases, we may switch off this variable in **NRLMOL\_INPUT.DAT** SCALEDLBFGSV = 'N' to use the conjugated gradient for the FOD optimization. As the FODs are already updated and we also have the wave function and the **RUNS** file, we just run FLOSIC in the same directory to do the SCF cycle with the new FODs.

**\$ PATH\_TO\_FLOSIC/nrlmol\_exe > print.002**

We can successively do this or use a script similar to that in the previous tutorial. After doing 5 steps of FOD optimization we can see in **fande.out** how the energy is decreasing and converging and the forces diminishing (not at every step).

```

 1  -76.337133952182  0.533401361320E-02  0.376016287388E-02
 2  -76.336140893884  0.316012354712E-01  0.222645230322E-01
 3  -76.337242375732  0.165238505484E-02  0.109072276621E-02
 4  -76.337256822409  0.184877969533E-02  0.129662662282E-02
 5  -76.337254671865  0.203173772471E-03  0.997893243061E-04

```



All the used FODs, obtained forces, and corrected total energies can be seen in the **records** file.

## 9.4 Plotting density of states using NRLMOL

### 9.4.1 Introduction

Once you start using NRLMOL, eventually you will need to plot the density of states for the system that you are running. The process can get a little messy so this guide should solve out some doubts.

First of, what are densities of states?

Basically, it's the number of electronic states at a given energy, and is one of the most important quantities to describe the electronic state of a structure [1].

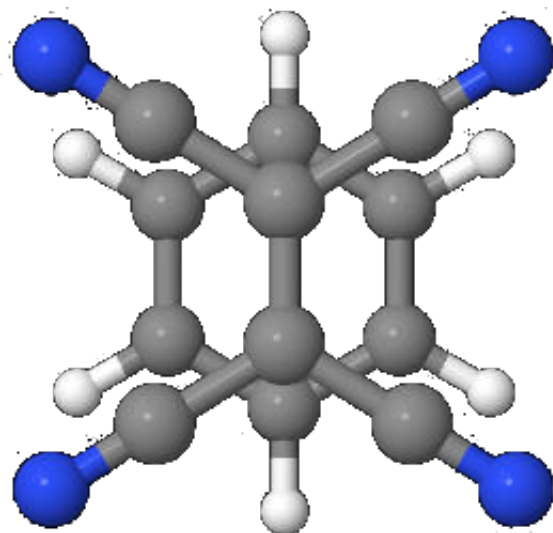
They are calculated via this:  $\rho(E) = \frac{1}{N_k} \sum_{i,k} \delta(\epsilon_{i,k} - E)$

where  $\rho(E)$  is the number of independent-particles states per unit energy,  $\epsilon_{i,k}$  denotes the energy of an electron [2].

The plots themselves are more qualitative than quantitative, since they tell us the localization of states within an energy interval, but the exact number of states for a given energy is rather unimportant. For our purposes (photovoltaics), we are interested mainly in checking that the last occupied state (that is, the HOMO) is localized in the donor, and that the first unoccupied state (LUMO) resides in the acceptor.

### 9.4.2 Setup

So let's do this by example. Let us do the calculation for the density of states for a system we have used previously, Tetracyanoethylene Benzene, or TCNE-Benzene. This is compound molecule made up of two components: the TCNE and Benzene, here the TCNE is in front and the benzene is in the background (Fig. 1).



The **CLUSTER** file to run the calculation of this structure is:

```
GGA-PBE*GGA-PBE
GRP
      22  Number of inequivalent atoms
      0.0000      4.7079      -3.9776      1 ALL UPO
```

(continues on next page)

(continued from previous page)

0.0000	-4.7079	-3.9776	1	ALL	UPO
4.0794	2.3559	-3.9732	1	ALL	UPO
-4.0794	2.3559	-3.9732	1	ALL	UPO
4.0794	-2.3559	-3.9732	1	ALL	UPO
-4.0794	-2.3559	-3.9732	1	ALL	UPO
-4.1665	3.8824	2.9754	7	ALL	UPO
4.1665	3.8824	2.9754	7	ALL	UPO
-4.1665	-3.8824	2.9754	7	ALL	UPO
4.1665	-3.8824	2.9754	7	ALL	UPO
0.0000	2.6433	-3.9697	6	ALL	UPO
0.0000	-2.6433	-3.9697	6	ALL	UPO
2.2925	1.3220	-3.9702	6	ALL	UPO
-2.2925	1.3220	-3.9702	6	ALL	UPO
2.2925	-1.3220	-3.9702	6	ALL	UPO
-2.2925	-1.3220	-3.9702	6	ALL	UPO
0.0000	1.2969	2.9150	6	ALL	UPO
0.0000	-1.2969	2.9150	6	ALL	UPO
-2.3072	2.7100	2.9357	6	ALL	UPO
2.3072	2.7100	2.9357	6	ALL	UPO
-2.3072	-2.7100	2.9357	6	ALL	UPO
2.3072	-2.7100	2.9357	6	ALL	UPO
0.0000	0.0000			Net Charge	and Moment

However, density of states is not a calculation that is normally done when you run NRLMOL, it has to be requested as a post-convergence calculation. That is, once the SCF cycle has converged, this calculation is now executed.

In order to request it, you have to set the option *DOSOCCUV* to 'Y' in the **NRLMOL\_INPUT.DAT** file. For older versions of NRLMOL, you have to create an empty file called **DOSOCCU** to request the calculation.

So, to get the density of states calculated, you need the following:

- The binary file (the program itself)
- The **CLUSTER** file
- Either the **NRLMOL\_INPUT.DAT** with the option set or an empty file called **DOSOCCU**

### 9.4.3 Results

Once the program finishes, it should have generated a list of files called **DOSO001**, **DOSO002**, and so forth (one for every atom in the structure you ran, there should be 22 files for our case of TCNE-Benzene).

The files may also have other starting names (**AF1O**, **AF2O**, **MINO**, **MAJO**) depending on the type of calculation you set up.

So basically, we have a density of states file for every atom in our system that we must plot, but how do we know which file corresponds to what atom? Do we follow the order from the **CLUSTER** file? The answer is no, because NRLMOL orders the atoms once it processes the **CLUSTER** file.

However, remember that NRLMOL prints a file called **XMOL.DAT** which is basically an xyz file to visualize the molecule with Jmol[3] or any other molecular viewer, it contains a listing for the atoms calculated. The order of the atoms in this file is the same order followed by the DOS files.

Here is the contents of **XMOL.DAT** for our TCNE-Benzene:

```

22
TCNE-Benzene
  6   0.00000   1.39877  -2.10067
  6   0.00000  -1.39877  -2.10067
  6   1.21314   0.69957  -2.10094
  6  -1.21314   0.69957  -2.10094
  6   1.21314  -0.69957  -2.10094
  6  -1.21314  -0.69957  -2.10094
  6   0.00000   0.68629   1.54255
  6   0.00000  -0.68629   1.54255
  6  -1.22092   1.43407   1.55350
  6   1.22092   1.43407   1.55350
  6  -1.22092  -1.43407   1.55350
  6   1.22092  -1.43407   1.55350
  1   0.00000   2.49131  -2.10485
  1   0.00000  -2.49131  -2.10485
  1   2.15872   1.24669  -2.10253
  1  -2.15872   1.24669  -2.10253
  1   2.15872  -1.24669  -2.10253
  1  -2.15872  -1.24669  -2.10253
  7  -2.20482   2.05448   1.57451
  7   2.20482   2.05448   1.57451
  7  -2.20482  -2.05448   1.57451
  7   2.20482  -2.05448   1.57451

```

As you can see from this listing, the atoms do not have the same order as in the **CLUSTER** file. Usually, NRLMOL orders atoms by listing carbons first, then hydrogens and the the rest of the atoms (it's the standard ordering for molecules in Chemistry).

From the listing, we now know that the first 12 DOS files correspond to the carbons, files 13-18 correspond to the hydrogens and files 19-22 correspond to the nitrogens. So now we have to plot the DOS files for each of the components and for the full complex, but in order to do so we are going to need membership files.

#### 9.4.4 Membership files

So far, we know that the **DOSO** files generated are for each atom in the system, and they follow the ordering from the **XMOL.DAT** file, so let's put them side by side for clarity:

```

22
TCNE-Benzene
  6   0.00000   1.39877  -2.10067 -> DOS0001
  6   0.00000  -1.39877  -2.10067 -> DOS0002
  6   1.21314   0.69957  -2.10094 -> DOS0003
  6  -1.21314   0.69957  -2.10094 -> DOS0004
  6   1.21314  -0.69957  -2.10094 -> DOS0005
  6  -1.21314  -0.69957  -2.10094 -> DOS0006
  6   0.00000   0.68629   1.54255 -> DOS0007
  6   0.00000  -0.68629   1.54255 -> DOS0008
  6  -1.22092   1.43407   1.55350 -> DOS0009
  6   1.22092   1.43407   1.55350 -> DOS0010
  6  -1.22092  -1.43407   1.55350 -> DOS0011

```

(continues on next page)

(continued from previous page)

6	1.22092	-1.43407	1.55350	-> DOS0012
1	0.00000	2.49131	-2.10485	-> DOS0013
1	0.00000	-2.49131	-2.10485	-> DOS0014
1	2.15872	1.24669	-2.10253	-> DOS0015
1	-2.15872	1.24669	-2.10253	-> DOS0016
1	2.15872	-1.24669	-2.10253	-> DOS0017
1	-2.15872	-1.24669	-2.10253	-> DOS0018
7	-2.20482	2.05448	1.57451	-> DOS0019
7	2.20482	2.05448	1.57451	-> DOS0020
7	-2.20482	-2.05448	1.57451	-> DOS0021
7	2.20482	-2.05448	1.57451	-> DOS0022

We can see from here that file **DOSO0007** corresponds to the seventh carbon atom, file **DOSO0013** corresponds to the first hydrogen atom and so forth. As we previously stated, we need to generate three DOS plots: One for the full complex, one for the Benzene and one for the TCNE.

We clearly can see that the hydrogens belong to the Benzene and the nitrogens to the TCNE, but which carbons belong to the Benzene and which belong to the TCNE? we have no way of knowing this, so we need to open the **XMOL.DAT** file with Jmol to visualize the structure. Once you open it with Jmol, by positioning the cursor over a single atom it will tell you what number it is from the list, jot this number down and make a list for each atom of TCNE and benzene separately.

From this information you can create a membership list of the atom indexes for each of the substructures. For benzene, with the help of Jmol we know that carbon atoms 1-6 belong to the benzene and carbon atoms 7-12 belong to the TCNE. Therefore, the complete membership for benzene (fig. 2) is:

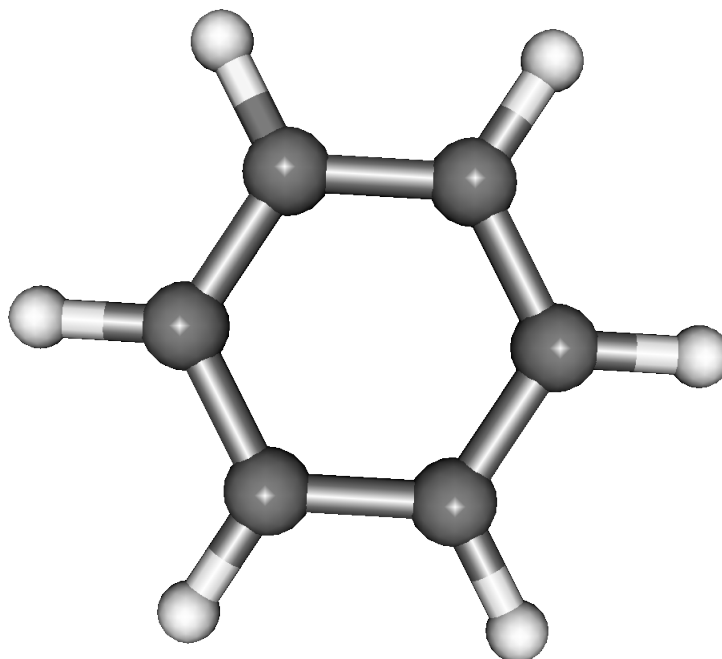
```
1
2
3
4
5
6
13
14
15
16
17
18
```

which are the indexes for the six carbon and the six hydrogen atoms.

For TCNE (fig. 3) the membership is:

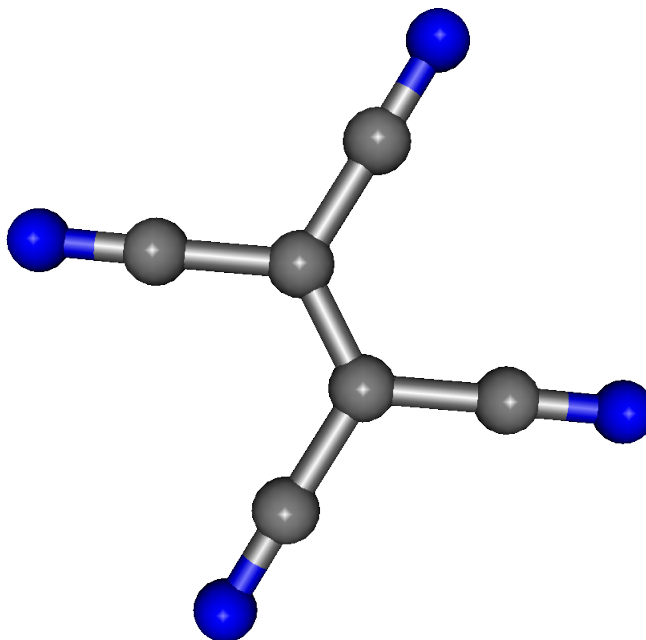
```
7
8
9
10
11
12
19
20
21
22
```

(continues on next page)



(continued from previous page)

wich are the six carbon atoms and the four nitrogen atoms.



We must now use this listings to create the membership files for plotting the DOS for each substructure, we only need to put in the first line of the membership file the total atoms for that membership. The membership file for benzene (let's call it **benzene.mem**) is:

12

(continues on next page)

(continued from previous page)

```
1
2
3
4
5
6
13
14
15
16
17
18
```

In general, the ordering of the atom indexes is not necessary, it's just shown here for clarity.

Similarly the membership file for TCNE (let's call it **tcne.mem**) is:

```
10
7
8
9
10
11
12
19
20
21
22
```

Additionally, we must create the membership file for the full complex, this is trivial, as it is just a listing of the indexes of all the atoms in the structure (let's call it **total.mem**), you can even generate it automatically with the computational tool of your choice.

```
22
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

(continues on next page)

(continued from previous page)

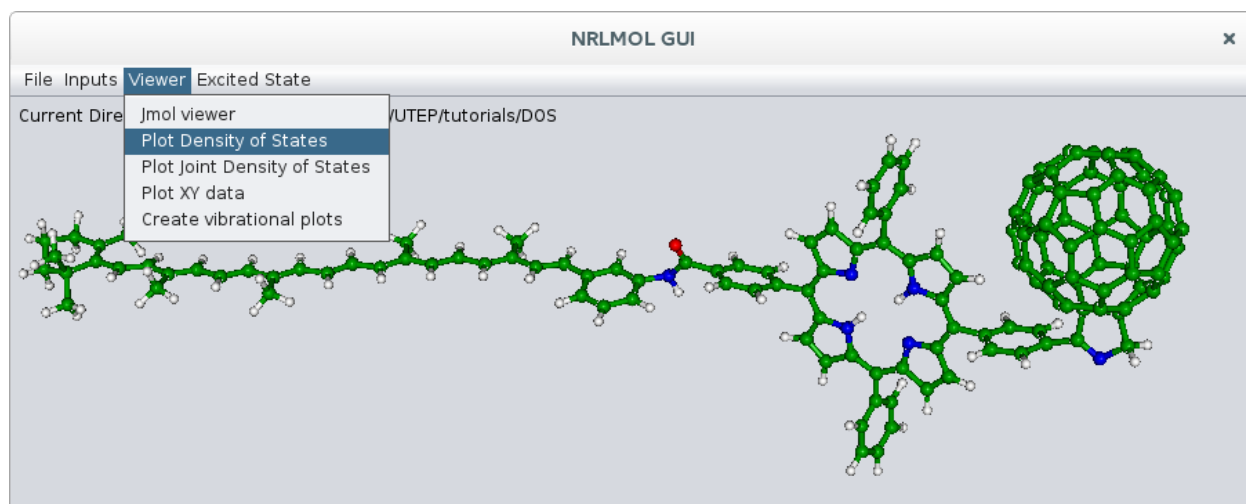
20  
21  
22

Again the ordering is not important as long as each atom is listed.

We can now use these three membership files to plot their corresponding density of states.

### 9.4.5 Plotting

To plot the membership files, we use the NRLMOL\_GUI program to do so. Set the project directory by selecting *Set Project Directory* from the *File* menu. This is the directory where you ran the calculation and that contains the DOS and other output files. After this, select *Plot Density of States* from *Viewer* in the main menu (fig. 4).

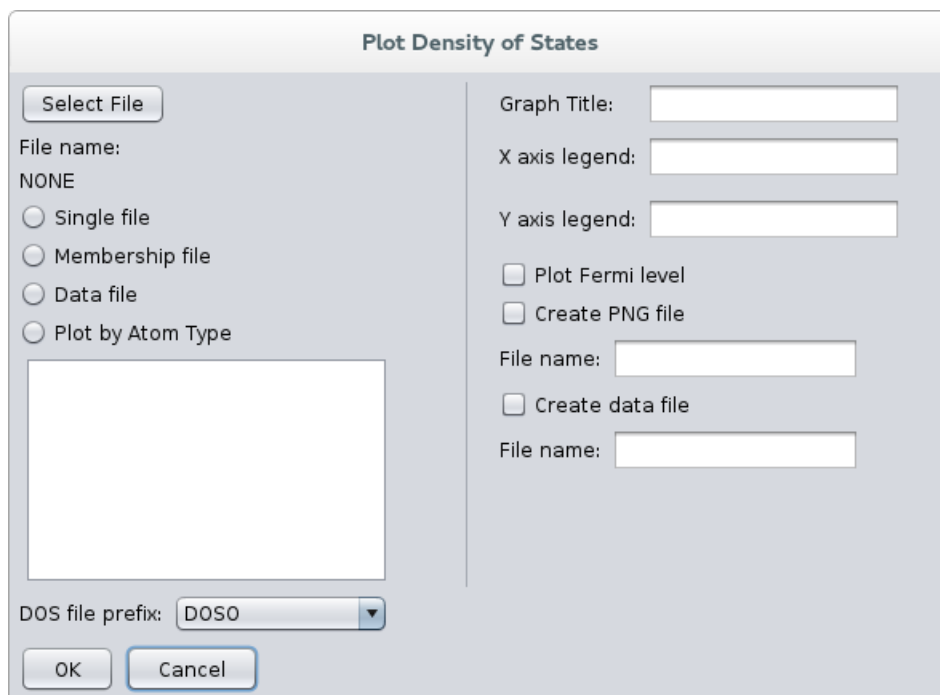


This will open the dialog box for plotting density of states (fig. 5), from here:

- Press the “Select File” button to select the membership file you want to plot.
- Select the “Membership file” radio button to process the file as a membership file.
- Select the “DOS file prefix” to select the prefix of your DOS files (in case they are not named (DOSO))
- Type the titles and axis legends for the graph.
- Click on the checkbox if you want the Fermi level to be plotted (the file **EVALUES** must be present in the project directory).
- You can select to write a graphics file output, although you can do that by doing a right-click on the plot window, this second option is preferred since you can do some adjustments to the graph before saving.
- The last check mark *Create data file* should be checked, and a file name given (only when plotting from a membership file), no extension is required as it will automatically add .dat to the file name.

Why should a data file be created when plotting from a membership file? Well, the process goes as this: The program will take each of the DOS files for each atom, adds them together and puts it in a file for plotting, however; if you want to plot the data yourself with another program, you can use this data file for that purpose. After you are finished, you should now have three final data files:

- benzene.dat a data file containing the DOS information for the benzene atoms generated from the benzene.mem file.



The dialog box is titled "Plot Density of States". It has two main sections. The left section contains a "Select File" button, a "File name:" label with the text "NONE", four radio buttons labeled "Single file", "Membership file", "Data file", and "Plot by Atom Type", a large empty rectangular box, and a "DOS file prefix:" dropdown menu currently set to "DOS0". The right section contains three text input fields for "Graph Title:", "X axis legend:", and "Y axis legend:". Below these are two checkboxes labeled "Plot Fermi level" and "Create PNG file". At the bottom of the right section are two more text input fields, both labeled "File name:". At the very bottom of the dialog are "OK" and "Cancel" buttons.

- tcne.dat a data file containing the DOS information for the tcne atoms generated from the tcne.mem file.
- total.dat a data file containing the DOS information for all the atoms generated from the total.mem file.

Let's now use gnuplot[4] to generate the three plots in a single image from the data files.

### 9.4.6 Gnuplot usage

To now join together the three data files, we need to setup a gnuplot script to process the data in each file and plot them. However, we also need one extra data file for the Fermi level.

We get the Fermi level from the **EVALUES** file: open the file and look for the line containing *FERMI LEVEL* and the number next to it, the number is however given in Hartrees, so you must convert it to electron-volt by multiplying it by 27.2116. You could have also got it from the GUI, if you requested to plot the Fermi level, this goes and reads it from the file and puts in a pop up window so you can jot it down.

For the TCNE-Benzene this value is: -0.206529774179560 so its corresponding value in eV is -5.620005603

So make a file called **fermi.dat** and put this two lines in it:

```
-5.620005603 0
-5.620005603 70
```

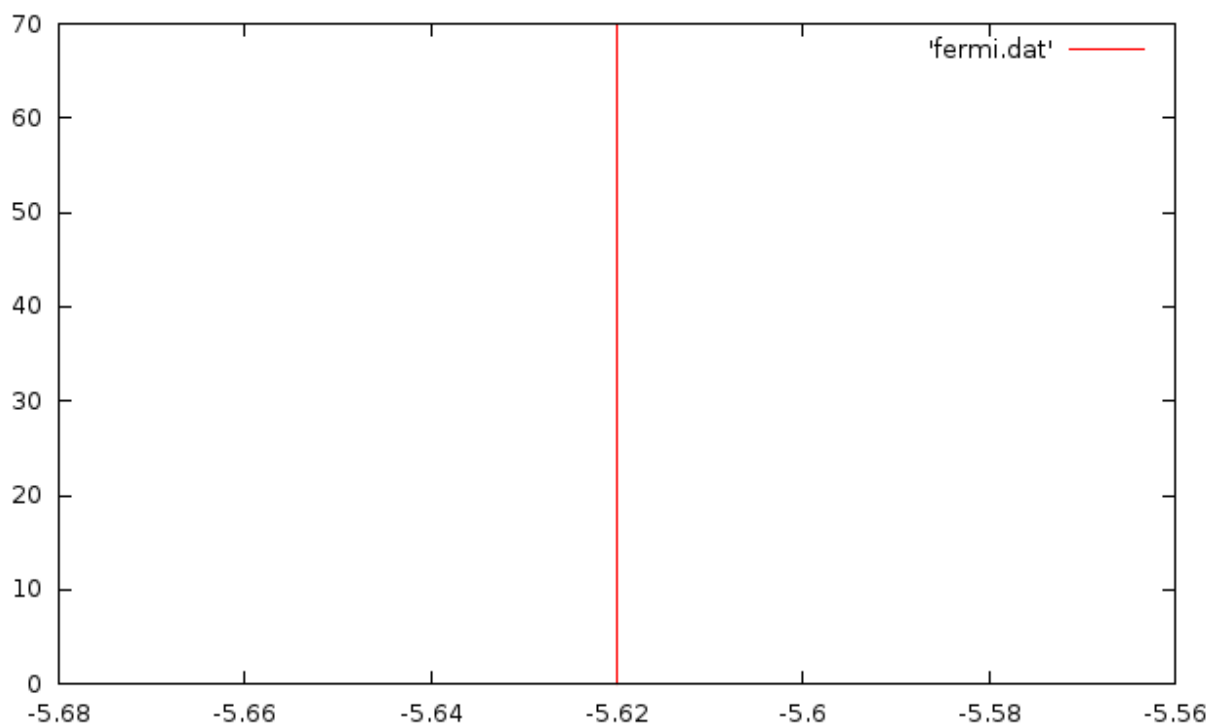
These are just two x,y pairs to be plotted. If you think about it, if the points are connected on a graph, they will generate a vertical line at the Fermi level (fig. 6). We give the second point a big y value (70 in this case, it can be anything) to make shure we cover all the plot surface.

Now we are ready to configure the plotting script for gnuplot. This is just a text file (let's call it **plot\_sample.p**) that conatins the following lines:

```
# This file will create a stacked plot for DOS
set terminal png size 1200,900 linewidth 3
```

(continues on next page)





(continued from previous page)

```

set output 'DOSplot_sample.png'
set linestyle 1 linecolor rgb 'blue'
set linestyle 2 linecolor rgb 'red'
set multiplot layout 3,1
# first plot top
set bmargin at screen 0.65
set tmargin at screen 0.95
set xrange[-12:0]
set yrange[0:10]
set xtics format ' '
set ytics 0,2,8
plot 'tcne.dat' with lines linestyle 1 notitle, 'fermi.dat' with lines linestyle 2 notitle
# Second plot middle
set bmargin at screen 0.35
set tmargin at screen 0.65
plot 'benzene.dat' with lines linestyle 1 notitle, 'fermi.dat' with lines linestyle 2,
↵notitle
# Third plot
set bmargin at screen 0.05
set tmargin at screen 0.35
set xtics format '%g'
plot 'all.dat' with lines linestyle 1 notitle, 'fermi.dat' with lines linestyle 2 notitle
unset multiplot

```

Lines that begin with # are comments and are thus ignored. The configurable options are:

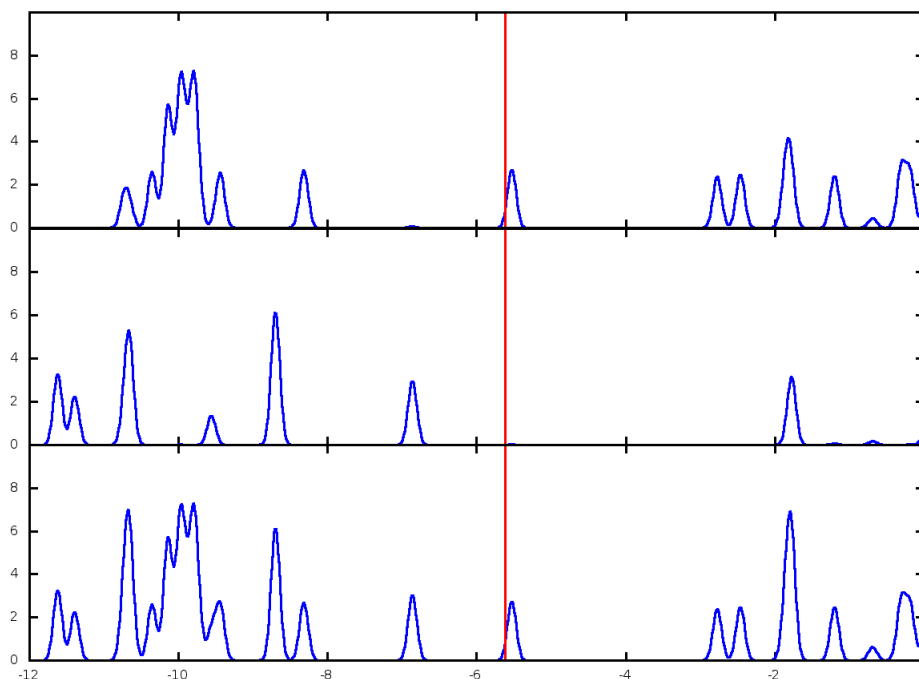
- We set a terminal output of png, this means we will generate a *.png* graphic file
- The requested linewidth is 3 is needed for publication quality graphs.
- The *set output* line specifies the file name of the generated file.
- Two linestyles are defined, which specify the line colors for the DOS data and Fermi level respectively.
- We remove unnecessary data by adjusting the x range (usually on the unoccupied states) to the left of the Fermi level (set *xrange*[-12:0]).
- We adjust the y range so all plots have the same height (set *yrange*[0:10]).
- We remove any number on the x axis until the last plot with: *set xtics format ‘ ‘*
- The *set ytics 0,2,8* means that the y range will be printed from 0 to 8 in increments of 2.
- You can change the order of the plots by specifying the name of the *.dat* file in each section (just don't forget to include the *fermi.dat* file).

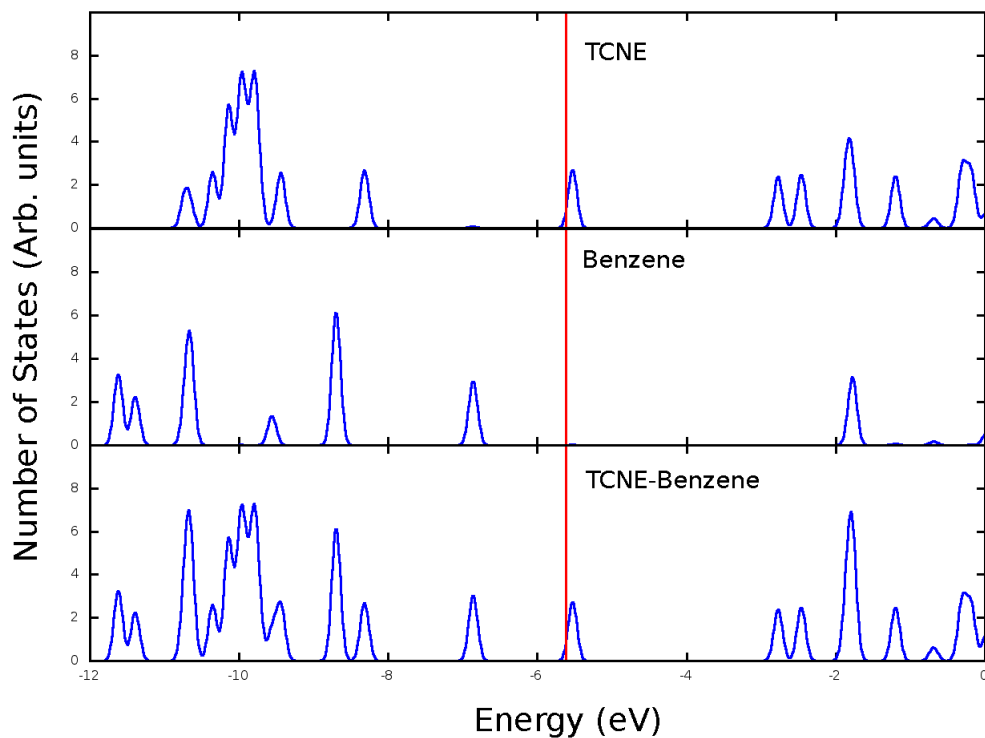
Once you set this up, run the script through gnuplot with the following command (in Linux, other platforms are untested):

*gnuplot plot\_sample.p*

You may need to run this command several times, while adjusting the values in the gnuplot script.

Once you get plot you're comfortable with (fig. 7), you still need to include the Axis labels as well as the descriptions of the individual plots (what part of the molecule each section represents), this can be done with any graphical editor (in the lab I usually use the GIMP[5] package, but it can be done with a presentation editor) to edit the png file (fig. 8).





### 9.4.7 Bibliography

1. David S. Sholl & Janice A. Steckel, *Density Functional Theory: A practical Introduction*, Wiley (2009).
2. Richard M. Martin, *Electronic Structure, Basic Theory and Practical Methods*, Cambridge University Press (2013).
3. Jmol: An open-source Java viewer for chemical structures in 3D. <http://www.jmol.org>
4. gnuplot: A portable command-line driven graphing utility. <http://www.gnuplot.info>
5. GIMP: The GNU Image Manipulation Program. <http://www.gimp.org>

## GETTING MORE INFORMATION ABOUT FLOSIC

Questions about the FLOSIC code or its use can be directed to

- Tunna Baruah (UTEP), [tbaruah@utep.edu](mailto:tbaruah@utep.edu)
- Rajendra R. Zope (UTEP), [rzope@utep.edu](mailto:rzope@utep.edu)
- Koblar Alan Jackson (CMU), [jacks1ka@cmich.edu](mailto:jacks1ka@cmich.edu)

or write to the FLOSIC mail server [electronic\\_structure\\_lab@listserv.utep.edu](mailto:electronic_structure_lab@listserv.utep.edu).

Some FAQs are listed below.

1. **What convergence criteria should be considered for the FOD optimization?** In general, you want to make sure that your total energy is converged within your given SCF tolerance (like in a standard DFT calculation) and that the largest FOD force is at least smaller than  $5E-4$  Ha/Bohr.
2. **How can I start a spin-polarized/ spin-unrestricted calculation?** For a spin-unrestricted calculation, you need to make sure that your underlying DFT density is spin-unrestricted as well. If your system has a permanent spin polarization (i.e. the number of up and down electrons is different), the calculation will always be spin-unrestricted. In the case of a total spin of  $S = 0$  (i.e. same number of up and down electrons), the code automatically carries out a spin-restricted calculation. To switch to a spin-unrestricted setup (i.e. having individual sets of up and down FODs in the FRMORB file), set the SPNPOLV option in NRLMOL\_INPUT.DAT to Y.
3. **Is it possible to force antiferromagnetic spin ordering in some atoms?** Yes, you can specify atoms to have a preference for spin-up density, spin-down density or unpolarized density. The specifications needs to be made in the SYMBOL file. Here is an example :

```
ALL-MAN001 = 2.137000 -0.350000 0.003300 SUP  
ALL-MAN001 = 3.456900 -2.109300 0.093100 SDN  
ALL-CAR001 = 2.720100 -2.490500 0.106000 UPO
```

SUP, SDN and UPO signify spin up, spin down and unpolarized atoms. The default choice is UPO. Note that these are starting values for the atoms. After SCF, the density can be different than what has been initialized.