

GPView 1.2

Software Manual



<http://www.life-tp.com/gpview>

Tian Shi

Department of Chemistry

Ping Wang

Department of Computer Science,

Wayne State University

gpview@life-tp.com

Copyright (C) <2015-2016> Tian Shi & Ping Wang.

All right Reserved.

DEDICATION

This work is dedicated to our family.

Table of Contents

DEDICATION	2
Table of Contents.....	3
Overview	5
Installations.....	7
Functions and Examples.....	8
1. Display Molecular Structures	8
1) Overview	8
2) Molecular structures from Gaussian 09 log files	11
3) Molecular structures from PDB files	12
2. Animate Molecular Dynamics Trajectories	12
1) Molecular Structures.....	12
2) Non-Adiabatic Excited-State Molecular Dynamics (NA-ESMD)	13
3. Display Electronic Structures (Cube Files)	16
1) Visualize single cube.....	16
2) Visualize a pair of Cubes.....	17
4. 2D Contour Plots and Matrix Color Plots	17
5. Density of States and Oscillator Strength	18
6. Extract Data from Gaussian 09 RWF dumped file.....	19
7. Analysis of Transition Density Matrices.....	21
1) Calculate Charge Transfer Number Matrices (CTNM)	21
2) Calculate Contracted Transition Density Matrices	23
8. Wave functions based analysis.....	23
1) Overview of fchk, wfx and cube files	23
2) Molecular Orbitals.....	27
3) Total SCF Density (Closed Shell Systems).....	28
4) Natural Transition Orbitals.....	29
5) Hole, Particle and Transition Densities	31
6) Natural Orbitals (Use Total CI Density)	33
7) Natural Difference Orbitals (Use Total CI Density)	34
8) Natural Orbitals (Use Total Rho CI Density)	36
9) Natural Difference Orbitals (Use Total Rho CI Density)	36
10) Detachment and Attachment Densities	38
11) Generate Grids Efficiently.....	39
Examples	41
Acknowledgement	42

Overview

The GPView is a C++ software for wave function analysis and visualizations. It is developed and maintained by Tian Shi and Ping Wang. Please cite the following paper, if you are using GPView:

Tian Shi, Ping Wang (2016), "GPView: a program for wave function analysis and visualization", [arXiv:1602.07302](https://arxiv.org/abs/1602.07302) [physics.chem-ph].

Main Visualization Functions include:

- a) Interface with GPV-ESMD program, which is a package for Non-Adiabatic Excited-State Molecular Dynamics (NA-ESMD), and animate the results of a single molecular dynamics trajectory, including molecular structures, potential energy surfaces, non-adiabatic coupling terms, hopping probability and bond lengths. Plot the average state populations as well.
- b) Visualize molecular structures. GPView can interface with G09 gif (or com), G09 fchk (or fch), G09 wfx, G09 log files, XYZ file, PDB file, and show molecular structures. It can also read molecular structures from input files of GAMESS, MOPAC, Q-chem, Mopro and NWchem.
- c) Visualize electronic structures. GPView can read grid points in CUB (or CUBE) files and display iso-surfaces for molecular orbitals and electron density. It can also display two cubes together, which is very convenient for visualizing and comparing HOMO-LUMO and Hole-Particle densities, simultaneously.
- d) Contour plots and color maps of matrices, e.g. Plot contracted transition density matrices and charge transfer number matrices.
- e) Plot Density of States and Spectrum (Oscillator Strength).

Main Analysis Functions include:

- a) Interface with G09 rwf dumped files. GPView can extract matrices, such as overlap matrix and density matrix, from G09 rwf dumped files.
- b) Interface with G09 fchk files. The parameters of wave functions are stored in fchk files. GPView can read these parameters, perform calculation and convert them to parameters for molecular orbitals, natural orbitals, natural transition orbitals and natural difference orbitals. They will be stored in wfx files.
- c) Interface with G09 wfx files. GPView can read parameters from wfx files, calculate orbitals or density, and output grid points to cube files.
- d) Transition density matrix based analysis. GPView can calculate charge transfer number matrices and contracted transition density matrices.
- e) Wave function based analysis. GPView can calculate and generate cubes for following orbitals and density.
 - Molecular orbitals (MO)
 - Natural transition orbitals (NTO)
 - Natural orbitals (NO)

- Natural difference orbitals (NDO)
 - SCF density
 - Total density obtained from NO
 - Hole-particle and transition density
 - Detachment-Attachment density
 - Gradient norm of electron density
 - Laplacian of electron density
- f) Simple calculations based on cube files, such as sum, difference and scale.
- **Note: The analysis functions so far are for restricted closed shell system and unrestricted system. We are still testing the restricted open shell system. If you have questions or find any bug, please let us know. We appreciate your help.**

Save Figures and Videos:

- a) Save the screenshots to PNG files.
- b) Save the MD animations to AVI files.

We hope this software can be helpful in your work. If you experience any bug, or if you have any suggestion, please let us know by emailing gpview@life-tp.com. Your comments are important for GPView.

Installations

- **Windows**

Installation and Uninstallation are both very easy. GPView is installed in C:\ by default. You can't change the directory actually. You just need to follow the instruction of installer. GPView will not affect anything out of C:\GPView directory. To uninstall it, click uninstall exe in the GPView folder. Alternately, you can go to feature and application to uninstall it.

We highly recommend you to download GPView from <http://life-tp.com/gpview>. This is the only legal link where you can get a clean software.

- **Mac**

Download dmg file from the website and install the GPView to Applications. Then, open the terminal and copy the libraries in the /Applications/GPView/Libraries/* directory to /usr/local/lib.

You can run:

```
cd /Applications/GPView/Libraries/
```

```
sudo cp *vtk* /usr/local/lib
```

```
sudo cp *arma* /usr/local/lib
```

Now GPView is ready to use. To uninstall GPView, delete the GPView folder in Applications and also remove the shared libraries.

You can run:

```
sudo rm /usr/local/lib/*vtk*
```

```
sudo rm /usr/local/lib/*arma*
```

Note: Since GPView is developed under Windows system, we cannot guarantee that it runs properly in Mac or Linux system. Actually, we have already experienced bugs on visualization functions. We are trying to fix these problems.

Functions and Examples

The functions in GPView are introduced in this section. Most examples shown in this manual will be available in the GPView website. If not, you can request them from authors. We would like to welcome you to GPView.



Figure 1 Welcome Page.

1. Display Molecular Structures

1) Overview

GPView can read molecular structures from different files and display them on the screen. When GPView is open, three windows will pop up: The console, GUI (FLTK), and the display window (VTK). The console is used to print out useful information, such as the number of atoms. The display window can show 3D object, like molecular and electronic structures. GPView provides full graphic user interface (GUI), which makes it easier to use.

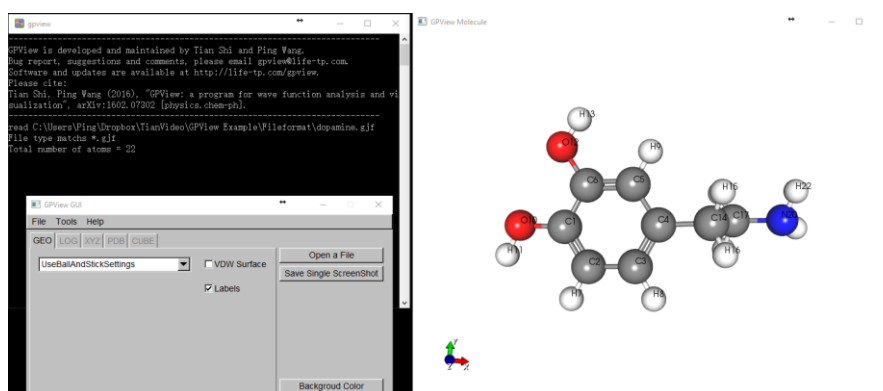


Figure 2 Visualize Molecular Structures

Let's take a look at the main panel of GUI.

- Open a file – Select and load a file.
- Save Single Screenshot – Save the current screenshot.

- Background Color – Change color of background of VTK window.
- Manual Bar – Most functions are under the 'Tools' manual bar.

The panel (Tabs) on the left is initially deactivated. When you successfully load a file it will be activated automatically. The program can read molecular geometries from different files according to their extensions, e.g. gjf for Gaussian input. Therefore, make sure you are using correct extensions. The molecular structures can be shown by three different representations:

Balls and Sticks Setting

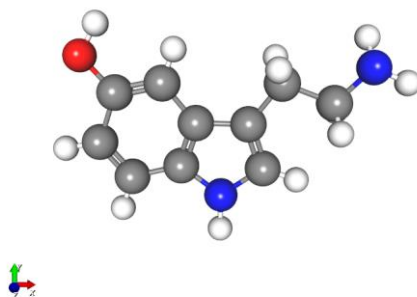


Figure 3 Display a molecule (SEROTONIN) use balls and sticks setting.

Liquorice Stick Setting

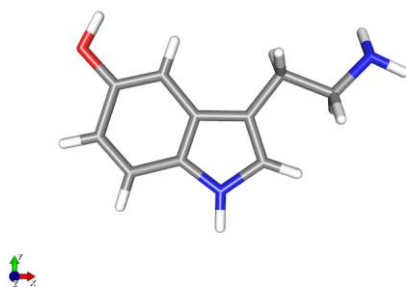


Figure 4 Display a molecule using liquorice stick setting.

VDW Spheres Setting

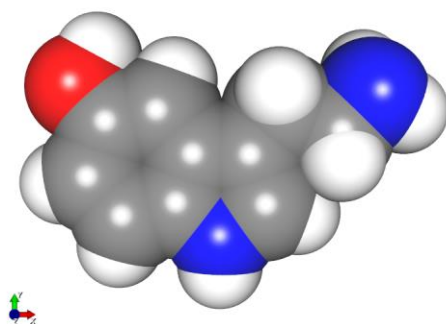


Figure 5 Display a molecule using VDW Sphere Setting.

The 'labels' check button is used to display the label of all the atoms. The number after the atom symbol corresponds to the order that atom in the file.

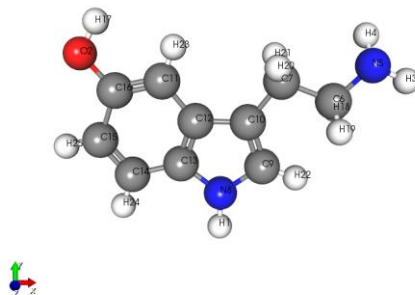


Figure 6 Display a molecule with labels of atoms.

GPView is compatible with the following list of file formats:

File.gjf (Gaussian 09 input)

File.com (Gaussian 09 input)

File.log (Gaussian 09 output)

File.fchk (Gaussian 09 format check point file)

File.cub (Gaussian 09 cube file)

File.cube (Gaussian 09 cube file)

File.wfx (Store information of wave functions)

File.xyz (Molecule structures, MD)

File.pdb (Protein Data Bank)

File.gamin (GAMESS US input)

File.gukin (GAMESS UK input)

File.inp (GAMESS input)

File.mop (MOPAC input)

File.mopcrt (MOPAC input)

File.mp (Mopro input)

File.qcin (Q-chem input)

File.nw (NWchem input)

2) Molecular structures from Gaussian 09 log files

The analysis functions in GPView are based on outcome of Gaussian 09 so far, so by interfacing G09 log or out files, GPView can read more information than structures.

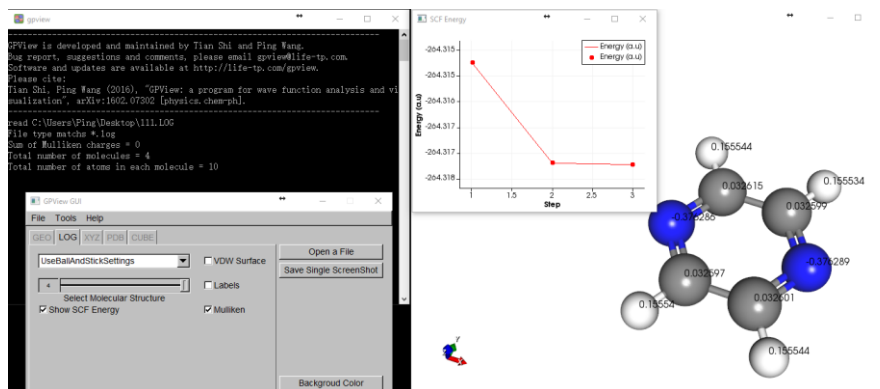


Figure 7 Interface with Gaussian 09 log/out file.

GPView can read more than one structures from G09 log files. This function is designed for visualizing structures after geometry optimization. You can view structure of each step by moving the slider. The SCF energy can also be displayed.

GPView can also read the Mulliken Charge and label them on atoms.

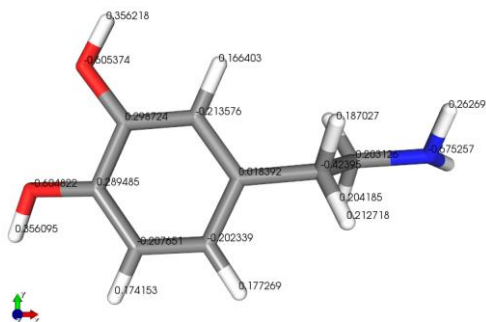


Figure 8 Display a molecule (dopamine, Gaussian log file) with Mulliken Charge.

3) Molecular structures from PDB files

The functions for viewing molecules in PDB file are not complete so far. It will be very slow to use above three representation settings, therefore, the default setting for viewing these molecules is 'protein ribbon setting'. Here is an example.



Figure 9 Visualize PDB files.

Structures can be downloaded from <http://www.rcsb.org/pdb/home/home.do>

In the future, we will write more modules on molecular structure visualizations. Therefore, if you have any suggestion, please let us know by gpview@life-tp.com.

2. Animate Molecular Dynamics Trajectories

1) Molecular Structures

In molecular dynamics of small molecules, the structures of a single trajectory are usually stored in an XYZ file. GPView can read these structures and create animations.

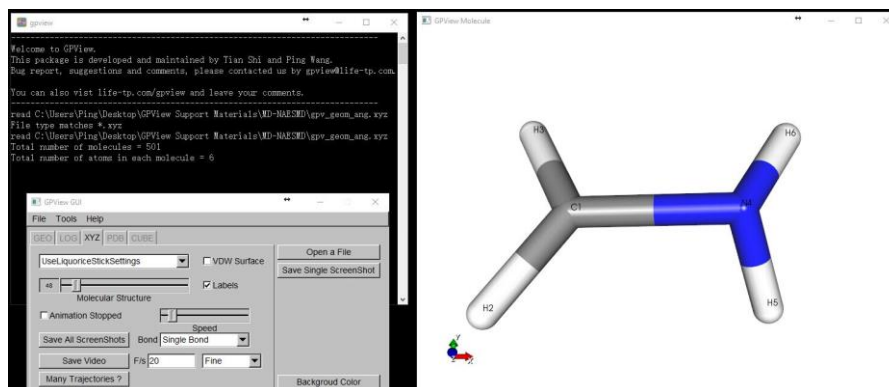


Figure 10 Visualize MD single trajectories

In the above figure, the console shows that there are 501 molecular structures in the XYZ file and the methaniminium molecule has 6 atoms.

In the GUI, you can also choose the style of molecules. There are many other functions:

- Molecular Structure – Choose the structure you want to view.
- Animation – Start/Stop an animation.
- Speed – Speed of an animation.
- VDW Surface – Show/hide VDW surface.
- Labels – Show/hide labels for atoms.
- Bond – Option for bonds. It will be discussed in details.
- Save All Screenshots – Save each structure as a PNG file.
- Save Video – Save animation as AVI videos.
- Many Trajectories? – Used to load multiple files and save multiple animations at one time.
- F/s – Frames per second.
- Quality – Coarse, Medium and Fine.

Animation will start and repeat automatically at the end of each loop if you check 'Animation' button. It is also possible to adjust the speed of animations.

In GPView, the bonds are built according to the distance of atoms. In the 'Bond' choice box, you can choose to display single bond, or multi bond, or no bond.

- Single bond – There will be either no bond when atoms are far from each other or single bond when atoms come close.
- Multi bond – No bond if atoms are far from each other. Single bond when atoms come close. Double or triple bonds when atoms are close enough.
- No bond – Only atoms will be displayed.

You can save the animation if you find a typical trajectory to an AVI file. Before saving the animation, it is possible to reset the frame rates and video qualities. There are three options for video qualities: Coarse, Medium and Fine. In this function, you don't need to install additional libraries.

What about if you want to save more than one trajectories? You can try 'Many Trajectories?' button. Multiple files will be load and animations will be converted to AVI videos, regardless whether the molecules in these files are the same or not. The filenames of videos will be the same as those for XYZ files. The only problem is that GPView will not check the orientation of molecules.

2) Non-Adiabatic Excited-State Molecular Dynamics (NA-ESMD)

If you are interested in this topic, we are assuming you have been familiar with NA-ESMD. In this function, the animations are based on the results produced by the GPV-ESMD program, which is developed and maintained by Tian Shi (The author of GPView program). The GPV-ESMD program has not been released yet, however, since GPView is a visualization tool, you don't need to worry about the reliability of the animation results in science.

The GPView program can create animations for the following variables:

- Molecular Structures (MD) – stored in gpv_geom_ang.xyz
- Bond Lengths – calculated from structures.
- Potential Energy Surfaces – stored in gpv_en.txt
- State Populations – stored in gpv_popu.txt
- Non-Adiabatic Coupling Terms – stored in gpv_nact_plot.txt
- Hopping Probabilities – stored in gpv_prob.txt

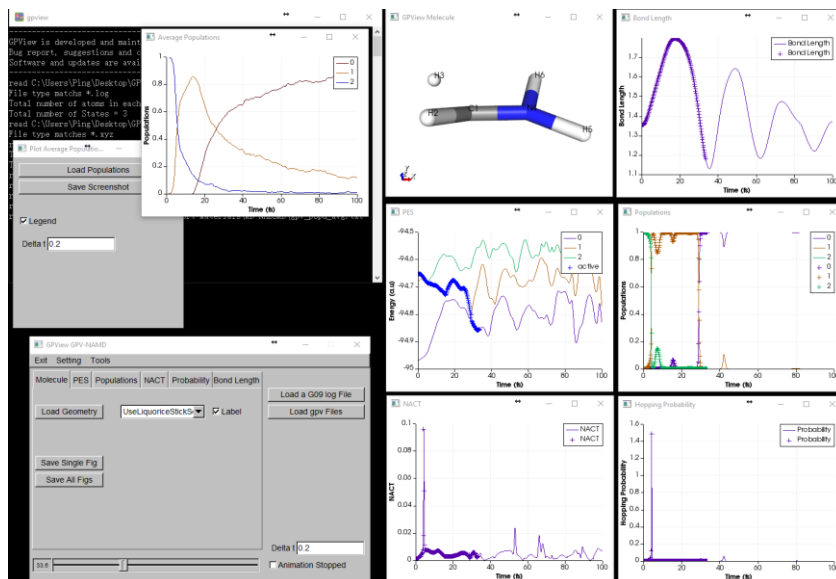


Figure 11 NA-ESMD GUI

Since GPV-ESMD interfaces with Gaussian 09, GPView will first read necessary parameters, such as the number of excited states, from Gaussian log file. Then, the left panel will be activated. Now, we have to load geometry, since GPView can count total number of time steps in MD from XYZ files, which will be used as parameters for loading other variables. Finally, you can load the rest variables, as you want. This order is required, otherwise, GPView will close with error.

- The slider – choose time steps.
- Delta t – convert X-axis from time steps to fs.
- Load gpv Files – One click to load all files except for the G09 log file.

In summary, you have to follow two ways to load files.

- Load G09 log files -> Load Geometry -> Load other files.
- Load G09 log files -> Load gpv Files.

During initial condition sampling and average ensemble calculations, GPView can do 2D line plots for:

- Absorption spectrum
- Density of States
- Average Populations

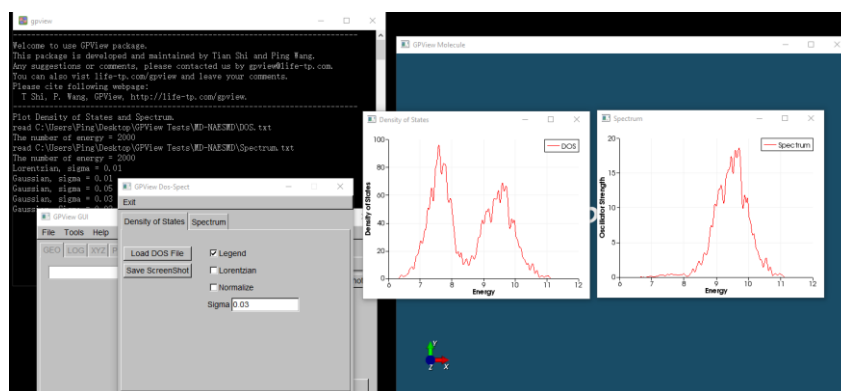


Figure 12 DOS and Spectrum from the sampling of 1000 geometry configurations with Wigner Distribution

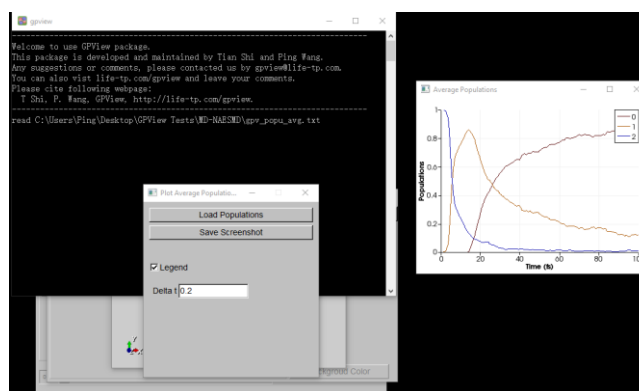


Figure 13 Average Populations of 200 trajectories. Ground State and first two excited states

Finally, there are many other NA-ESMD software available online. We will list some open-source software:

Newton-X -- <http://www.newtonx.org/>

JADE -- <http://jade-package.github.io/JADE/>

PYXAID -- <http://www.acsu.buffalo.edu/~alexeyak/pyxaid/overview.html>

SHARC -- https://sharc-md.org/?page_id=2

In this section, if you have special needs, please contact us by gpview@life-tp.com. We will extend this function in the future.

3. Display Electronic Structures (Cube Files)

1) Visualize single cube

In addition to visualize molecular structures, GPView can also visualize the electronic structures, including molecular orbitals and electron densities. The 3D grid information is stored in Gaussian Cub or Cube files. GPView can read these cube files and convert them to figures.

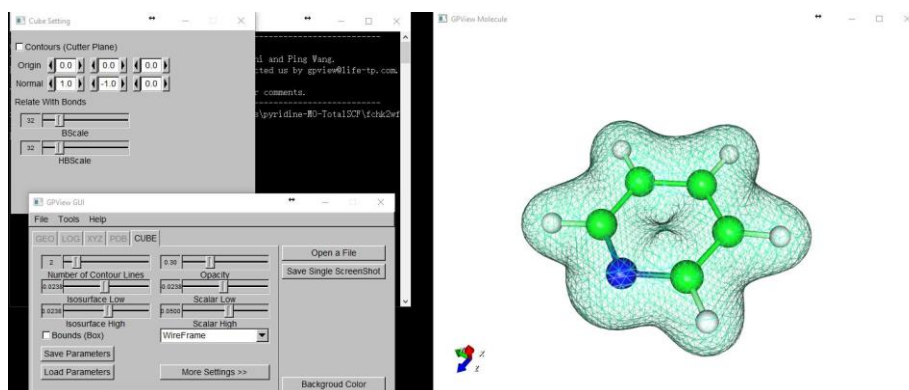


Figure 14 Display Electronic Structures

In the GUI panel, there many functions relate with parameter settings:

- Number of Contour Lines – Change Number of contours (surfaces 1-10).
- Opacity – Change opacity of surfaces (0-1).
- Isosurface Low – Change the lower bound.
- Isosurface High – Change the upper bound.
- Scalar Low – Relate with the color of surface. Color for lower bound.
- Scalar High – Relate with the color of surface. Color for upper bound.
- Bounds (Box) – Show/Hide the bounding box.
- Representation – Surface and Wireframe.
- Save Parameters – Save your preferred parameters for future use. You can change them in the text file.
- Load Parameters – Load your preferred parameters.
- More Settings – Show the extension.
- Contours (Cutter Plane) – Cutter plane with contours.
- Origin – Origin of Cutter plane.
- Normal – Direction of Cutter plane.
- BScale – Relate with bonds between heavy atoms.
- HBScale – Relate with bonds between H and heavy atoms

2) Visualize a pair of Cubes

Sometimes, we need to observe two orbitals at the same time, such as HOMO-LUMO and Hole-Particle densities. Therefore, we design a function in GPView to visualize and interact with two cubes, simultaneously.

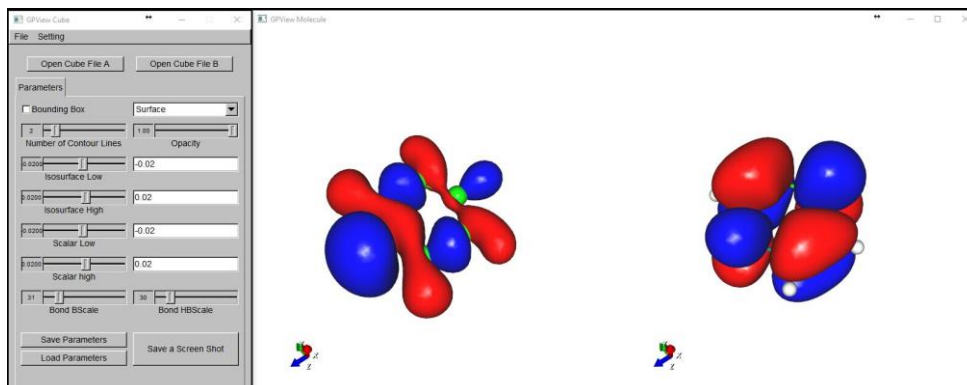


Figure 15 A pair of cubes

The functions include:

- Open Cube File A – Load Cube file 1.
- Open Cube File B – Load Cube file 2.
- Bounds (Box) – Show/Hide the bounding box.
- Representation – Surface and Wireframe.
- Number of Contour Lines – Change Number of Contours (Surfaces 1-10).
- Opacity – Change opacity of surfaces (0-1).
- Isosurface Low – Change the lower bound.
- Isosurface High – Change the upper bound.
- Scalar Low – Relate with the color of surface. Color for lower bound.
- Scalar High – Relate with the color of surface. Color for upper bound.
- BScale – Relate with bonds between heavy atoms
- HBScale – Relate with bonds between H and heavy atoms
- Save Parameters – Save your preferred parameters for future use.
- Load Parameters – Load your preferred parameters.
- Save a Screen Shot – Save current screenshot.

4. 2D Contour Plots and Matrix Color Plots

GPView can plot matrices as well, i.e. 2D contour plots and color maps of matrices. Sometimes, we may need to visualize a density or transition density matrix by contours or color maps. This function may be useful if you already have a matrix or a lot of matrices.

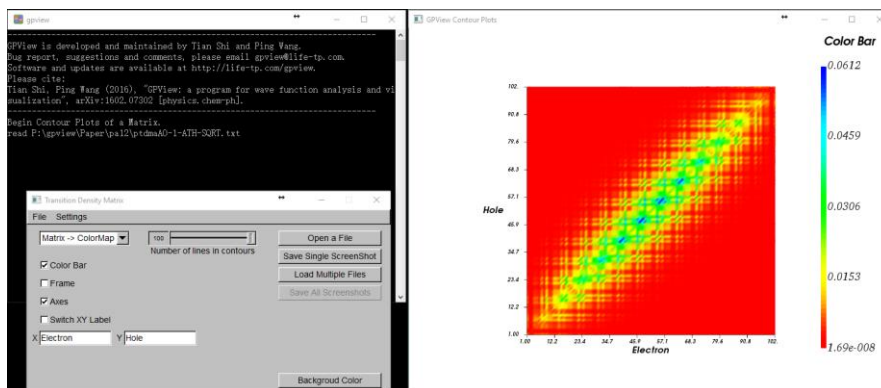


Figure 16 Color Maps of Matrices

In this function, we can load a file with a matrix and show it as either contours or color maps. Some functions in the GUI include

- Number of lines in contours – Adjust the number of Contours
- Color Bar – Show/hide color bar
- Frame – Show/hide frame
- Axes – Show/hide axes
- Switch XY Label – Switch the labels for X and Y axes
- X – Change the labels for X
- Y – Change the labels for Y
- Background Color – Change the color of background

We may have many (e.g. 100) matrices to plot. Don't worry. We can use 'Load Multiple Files' to load files as many as we want, set up parameters and then 'Save All ScreenShots' to save all of them as figures.

5. Density of States and Oscillator Strength

We've already introduced this function in NA-ESMD part. In this function, if you load Gaussian 09 log files from excited-state calculations, you will get Density of States and Absorption Spectrum (Oscillator Strength) for excited states.

In this function, we use both Gaussian and Lorentzian broaden functions. You can choose any of them.

Gaussian:

$$DOS(\omega) = \sum_i e^{-\frac{(\omega-\omega_i)^2}{2\sigma^2}}$$

$$I(\omega) = \sum_i f_i \cdot e^{-\frac{(\omega-\omega_i)^2}{2\sigma^2}}$$

Lorentzian:

$$DOS(\omega) = \sum_i \frac{\sigma^2}{(\omega - \omega_i)^2 + \sigma^2}$$

$$I(\omega) = \sum_i \frac{f_i \cdot \sigma^2}{(\omega - \omega_i)^2 + \sigma^2}$$

You can always reset σ in the panel.

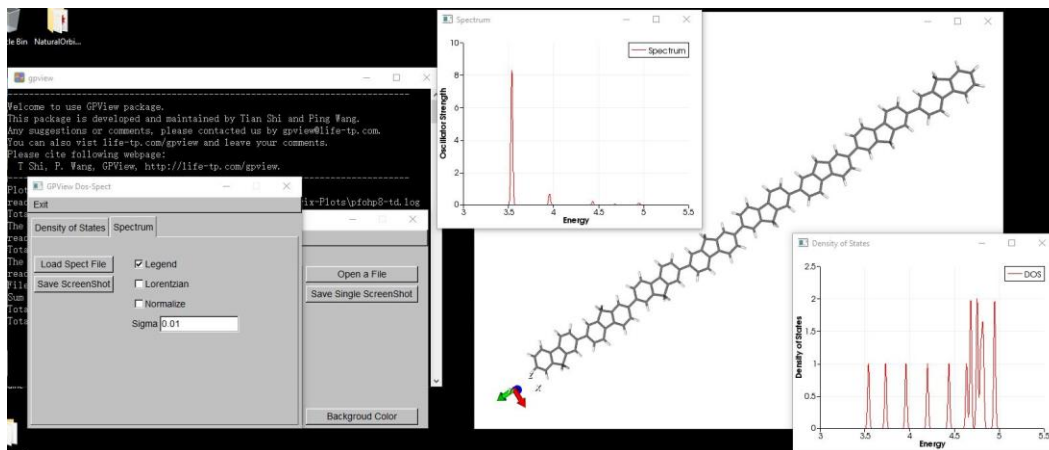


Figure 17 Density of States and Spectrum

The functions in GUI include:

- Save Screenshot – You can save screenshots to PNG files.
- Legend – Show/hide legend.
- Lorentzian – Broaden functions. Default is Gaussian function. You can choose Lorentzian function.
- Normalize – The largest value will be normalized to 1. The spectrum and DOS will be normalized by dividing the max value.
- Sigma – Parameters used in Gaussian and Lorentzian.

6. Extract Data from Gaussian 09 RWF dumped file

There are a lot of useful information stores in the Gaussian RWF files. A complete list of links of the RWF file can be found in

http://www.gaussian.com/g_tech/g_ur/e_progdev.htm.

GPView program can extract data from RWF dumped files and store them in a user-friendly format.

First, there is an application named `rwfdump`, which can extract data from RWF files, in Gaussian 09 installation directory. You can run `"rwfdump file.rwf output.txt 514R"` to

get dumped files. Here 514R is an example. '514' corresponding to overlap matrices, 'R' means they are stored as real numbers. Alternately, by running `"rwfdump file.rwf output.txt 633R"`, we can get the state density matrices and transition density matrices.

GPView can interface with dumped files and write the results into txt files in matrix or vector format, which can be easily used for analysis. Here is a list of matrices that are compatible with GPView.

- 514R – Overlap matrix
- 522R – Eigenvalues
- 524R – MO coefficients Alpha
- 526R – MO coefficients Beta
- 528R – SCF density Alpha
- 530R – SCF density Beta
- 532R – SCF density Real Total
- 534R – SCF density Real Spin
- 536R – Fock matrix Alpha
- 538R – Fock matrix Beta
- 571R – Energy weighted density matrix
- 633R – Excited State CI densities (Total Rho-CI DM and 1TDM)
- 635R – CIS coefficients

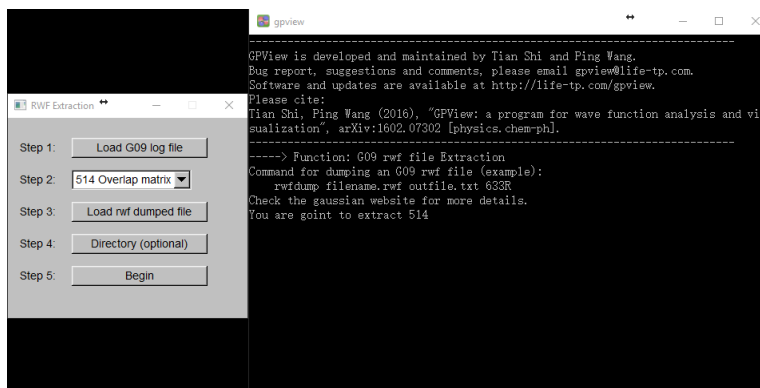


Figure 18 Extract rwf dumped files

The above figure shows the GUI of extracting RWF dumped files. Just follow these steps to extract data.

- Step 1: Load G09 log file – Parameters are stored in Gaussian 09 log files.
- Step 2: Choice – Choose a Task, e.g. 514 Overlap matrix.
- Step 3: Load rwf dumped file – Choose an RWF dumped files, e.g. 514.txt
- Step 4: Directory – Output directory. It is optional. By default, GPView will put them in the same directory as RWF dumped files or log files.
- Step 5: Begin – Begin to extract.

In G09 input file, you need to add following commands

`%chk=file.chk`

`%rwf=file.rwf`

`%save`

in order to save the RWF file (It is a large file).

7. Analysis of Transition Density Matrices

GPView can perform DM based analysis given overlap matrices and transition density matrices. In this function, we need Gaussian fchk or fch files as well, which provides us necessary parameters relate with basis functions. The following figure shows the GUI of this function.

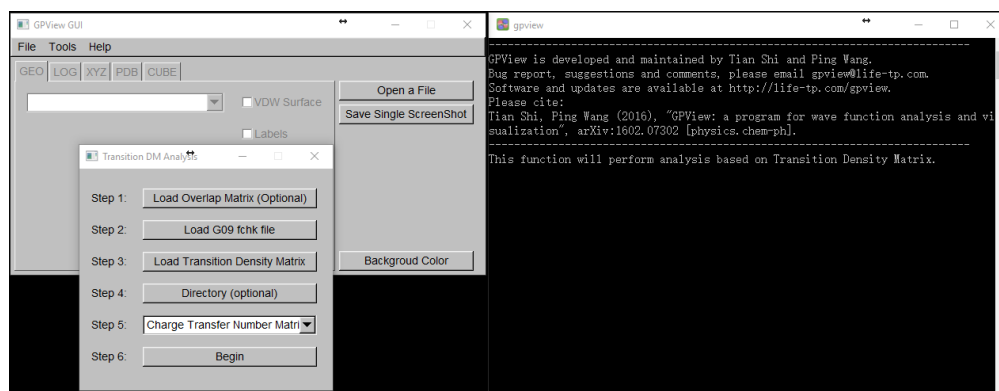


Figure 19 TDM Analysis

- Step 1: Load Overlap Matrix (Optional) – Load the overlap matrix.
- Step 2: Load G09 fchk file – Load the fchk/fch file.
- Step 3: Load Transition Density Matrix – Load one or many TDM.
- Step 4: Directory (Optional) – Used for outputting files.
- Step 5: Tasks – Example: the first option is for CTNM in subsection 1.
- Step 6: Begin – Begin the analysis.

There are two types of analysis so far:

1) Calculate Charge Transfer Number Matrices (CTNM)

This method is proposed in the following reference by Plasser and Lischka:

- Plasser, F. and H. Lischka, *Analysis of Excitonic and Charge Transfer Interactions from Quantum Chemical Calculations*. Journal of Chemical Theory and Computation, 2012. **8**(8): p. 2777-2789.

The charge transfer number between segment A and B can be represented by

$$\Omega_{AB} = \sum_{i \in A, j \in B} (D^{0\alpha} S)_{ij} (S D^{0\alpha})_{ij}$$

Where the $D^{0\alpha}$ is one electron transition density matrix (RWF 633R) from ground state to α excited state. The S is the overlap matrix (RWF 514R). GPView treats each atom as a 'segment', and we can get a matrix where X and Y represent the label of electrons and holes, respectively. Every element in the matrix represents the charge transfer number from one atom to another. This matrix has two advantages compared with original transition density matrix:

- The size is much smaller than the original one. For large conjugated systems, it is much easier to store and share.
- It can be visualized directed by matrix color maps or contours.

In addition to charge transfer number matrix, GPView will also calculate:

- The charge transfers number matrix for heavy atom only.
- A matrix whose elements are squared roots of the elements in charge transfer number matrix for heavy atom only (SCTNM), i.e. $\sqrt{\Omega_{AB}}$, where A and B are heavy atoms.

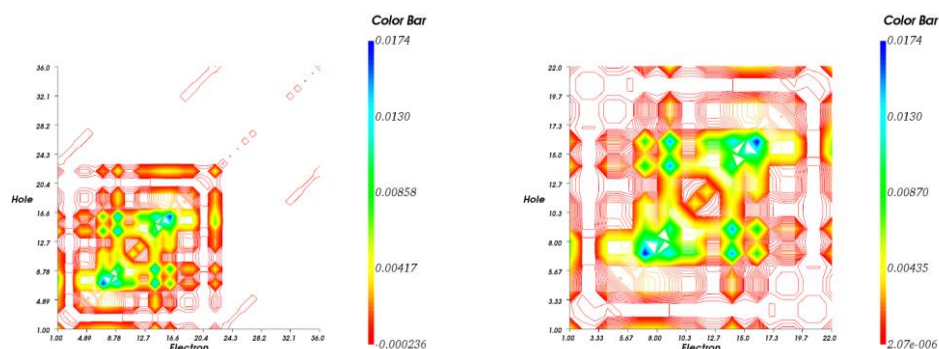


Figure 20 Contour Plots of Charge Transfer Number Matrices. Right: CTNM for heave atoms only.

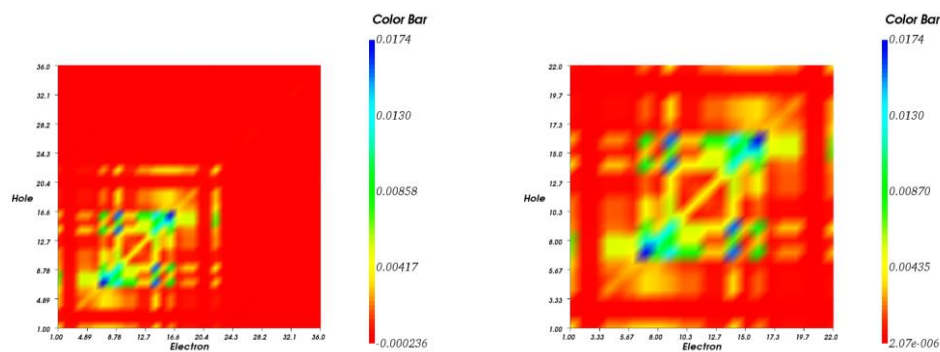


Figure 21 Color maps of Charge Transfer Number Matrices. Right: CTNM for heave atoms only.

2) Calculate Contracted Transition Density Matrices

To get a fully understanding of transition density matrices and contour plots. You can read the following reference:

- Tretiak, S. and S. Mukamel, *Density matrix analysis and simulation of electronic excitations in conjugated and aggregated molecules*. Chemical Reviews, 2002. **102**(9): p. 3171-3212.

The contracted transition density matrices (nuclear base) are calculated by two methods in GPView.

- Method 1:

$$D_{AB}^S = \sqrt{\sum_{i \in A, j \in B} (D^{0\alpha})_{ij} (D^{0\alpha})_{ij}}$$

- Method 2:

$$D_{AB}^S = \sum_{i \in A, j \in B} |(D^{0\alpha})_{ij}|$$

where A and B represent atom A and B, respectively. These contracted matrices can be shown as either contours or color maps.

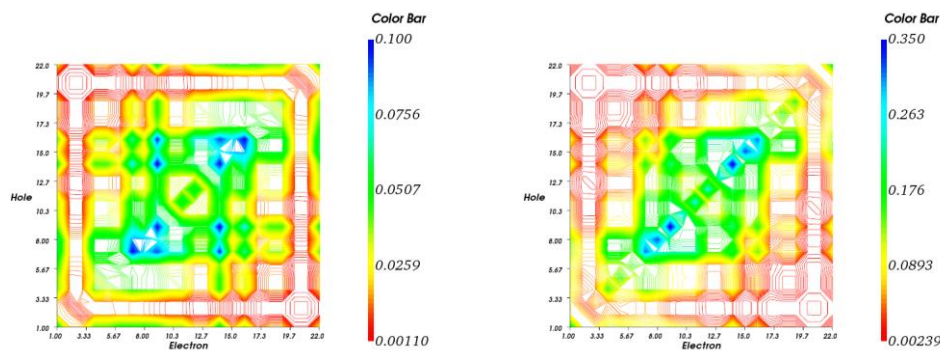


Figure 22 Contour Plots of Contracted Transition Density Matrices. Left: Method 1. Right: Method 2.

8. Wave functions based analysis

1) Overview of fchk, wfx and cube files

In this section, we will briefly summarize the websites which talk in details about file format of fchk, wfx and cube files. Then, we will show how GPView interfaces with these files and generates 3D cubes.

The format of fchk or fch files is discussed in details in Gaussian 09 website:

http://www.gaussian.com/g_tech/g_ur/u_formchk.htm

http://www.gaussian.com/g_tech/g_ur/f_formchk.htm

The format and use of wfx file can be found in Gaussian 09 and AIMALL websites:

http://www.gaussian.com/g_tech/g_ur/k_output.htm

<http://aim.tkgristmill.com/knownissues.html>

<http://aim.tkgristmill.com/wfxformat.html>

To generate a cube file, you can use the Gaussian cubegen utility. Here is a link that introduces the format of cube files.

http://www.gaussian.com/g_tech/g_ur/u_cubegen.htm

The wave function analysis part relies on three modules in GPView: Interface with fchk files, Interface with wfx files and Interface with cube files.

GPView can read molecular structures and electronic wave functions from fchk files. Then, it can perform wave function based analysis and output results to gjf, xyz and wfx files. We will outline the options here, and provide many examples in following sections.

- Get Current Coordinates -> xyz – Extract geometry to a XYZ file.
- Get Current Coordinates -> gjf – Extract geometry to a gjf file.
- Get Mulliken Charges -> txt – Extract Mulliken Charges to a text file.
- Fchk2wfx -> wfx (Restrict Closed Shell) – Read wave function from an fchk file. Export orbital information to a wfx file.
- Fchk2wfx -> wfx (MO) – Read wave function from an fchk file. Read parameters from a log file. Export orbital parameters to a wfx file.
- Fchk2wfx -> wfx (LCMO) -- Read wave function from an fchk file. Read parameters from a log file. Read linear transformation matrix from a text file. Read eigenvalues from a text file. Calculate the linear combination of molecular orbitals. Export orbital parameters to a wfx file.
- Fchk2wfx -> wfx (NTO, CIS) – Read wave function from an fchk file. Read CIS coefficients, i.e. X+Y and X-Y vectors (RWF 635R). Read parameters from a G09 log file. Calculate Natural Transition orbitals. Export results into a wfx file.
- Fchk2wfx -> wfx (NO, Total-CI) – Read wave function from an fchk file. Read Total CI density of current state (command in G09: Density=current) from the fchk file. Read Overlap Matrix (RWF 514R) from a text file. Read parameters from G09 log file. Calculate Natural Orbitals. Export results to a wfx file.
- Fchk2wfx -> wfx (NDO, Total-CI) – Read wave function from an fchk file. Read Total CI density of current state (command in G09: Density=current) from the fchk file. Read Overlap Matrix (RWF 514R) from text file. Read Total SCF density from the fchk file. Read parameters from a G09 log file. Calculate Natural Different Orbitals. Export results to a wfx file.
- Fchk2wfx -> wfx (NTO2NTO) – Read wave function from an fchk file that stores parameters of Natural Transition Orbitals. Read Parameters from a G09 log file. Export results to a wfx file.

- Fchk2wfx -> wfx (NO, Total-Rho-CI) – Read wave function from an fchk file. Read Total Rho CI density (RWF 633R) for all excited states from text files. Read Overlap Matrix (514R) from a text file. Read parameters from a G09 log file. Calculate Natural Orbitals. Export results to a wfx file.
- Fchk2wfx -> wfx (NDO, Total-Rho-CI) – Read wave function from an fchk file. Read Total Rho CI density (RWF 633R) of all excited states from text files. Read Total SCF density from the fchk file. Read Overlap Matrix (RWF 514R) from a text file. Read parameters from a G09 log file. Calculate Natural Different Orbitals. Export results to a wfx file.
- Fchk2wfx ->wfx (NTO, TDMAO) – Read wave function from an fchk file. Read Transition Density Matrices (RWF 633R) from text files. Read Overlap Matrix (RWF 514R) from a text file. Read parameters from a G09 log file. Calculate the Natural Transition Orbitals. Export results to a wfx file.

The following figure is the GUI of Interface G09 fchk file:

- Step 1: Load G09 fchk file – Load an fchk file.
- Step 1: Restricted – Restricted Closed Shell System.
- Step 1: UnRestricted – Unrestricted System.
- Step 1: Restricted-Open – Restricted Open Shell System (No available in current version).
- Step 2: Load G09 log file – Load parameters from G09 log file.
- Step 3: LinearTransMatrix – Load a coefficient matrix used to do linear transformation of MOs.
- Step 4: EigenValues – Load a user defined vector.
- Step 5: Directory – Output files.
- Step 6: Tasks.
- Step 7: Begin – Begin to process.

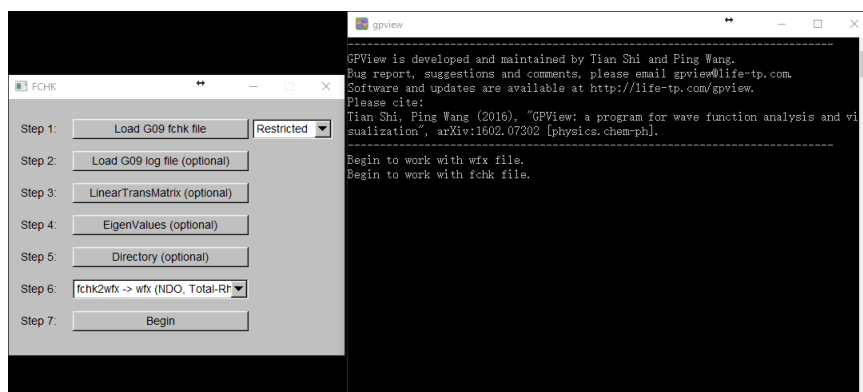


Figure 23 Interface with fchk file

GPView can read molecular structures and wave function parameters from wfx files. The geometries will be written to gif or XYZ files. The wave functions are used to calculate

molecular orbitals and electron densities. Finally, GPView generates grids and export results to cube files. Here is a list of options in tasks (Step 3)

- Get Cartesian Coordinates -> xyz – Extract the geometry to a XYZ file.
- Get Cartesian Coordinates -> gjf – Extract the geometry to a gjf file.
- Generate Molecular Orbitals -> cub – Read wave function parameters from a wfx file. Write MO (Molecular Orbitals, Natural Transition Orbitals, Natural Orbitals, and Natural Difference Orbitals) into the cube file.
- Generate SCF Density -> cub – Read wave function parameters from a wfx file. Calculate SCF electron density. Generate and write grids to a cube file.
- Generate Hole, Particle and Transition Densities (1NTO) -> cub – Read wave function parameters of NTO from a wfx file. Calculate Hole, Particle and Transition Densities. Generate and write grids to a cube file. Here, we only use the NTOs of highest contribution, i.e. largest λ_i .
- Generate Hole, Particle and Transition Densities (AllNTO) -> cub – Read wave function parameters of NTO from a wfx file. Calculate Hole, Particle and Transition Densities. Generate and write grids into a cube file. Here, we use the NTOs of non-zero λ_i .
- Generate Attachment and Detachment Densities (NDOs) -> cub – Read wave function parameters of NDO from a wfx file. Calculate Attachment and Detachment density. Generate and write grids into a cube file.

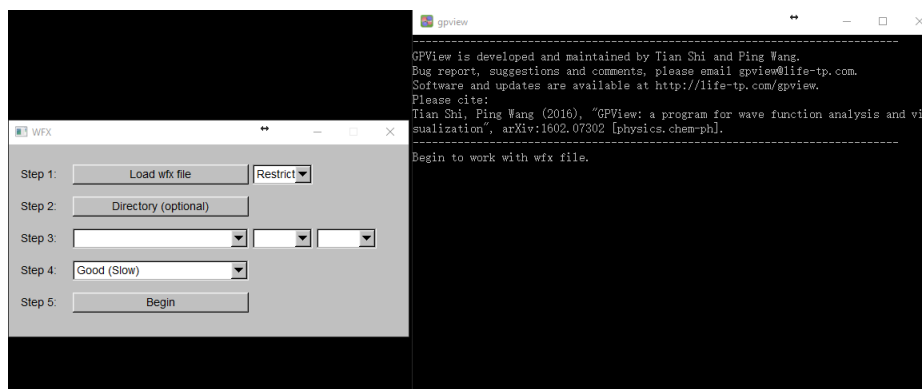


Figure 24 Interface with wfx file

- Step 1: Load wfx file – Load a wfx file.
- Step 1: Restricted – Restricted Closed Shell System.
- Step 1: UnRestricted – Unrestricted System.
- Step 1: Restricted-Open – Restricted Open Shell System (No available).
- Step 2: Directory – Output files.
- Step 3: Tasks.
- Step 3: Choice for MOs – All, HOMO-LUMO, HOMO-2-LUMO+2, Orbital Label.

- Step 3: Choice for Density -- Total Density, alpha density, beta density and spin density.
- Step 4: Quality – Bad (Fast), Medium and Good (Slow) (default).
- Step 5: Begin.

GPView can also read grids from cube files, perform some simple calculations based on elements and export results into another cube file. These calculations include square and scale of the elements of cube, difference and sum of elements from two cube files. This function has not been optimized so there may be bugs.

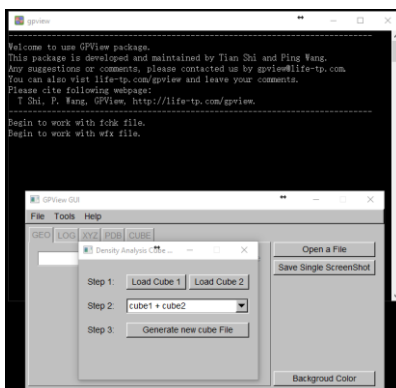


Figure 25 Interface with cube file

2) Molecular Orbitals

The wave function of molecular orbitals $\phi_i(r)$ can be expressed as

$$\phi_i(r) = \sum_j C_{ij} \chi_j(r)$$

where C_{ij} is the molecular orbital coefficient and $\chi_j(r)$ is the basis function (atomic orbital).

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with fchk file. Choice: 'fchk2wfx -> wfx (Restrict Closed Shell)'.
- Interface with wfx file. Choice: 'Generate Molecular Orbitals -> cub', 'HOMO-2~LUMO+2', 'Good (Slow)'.
- Read a pair of cube files, with HOMO and LUMO. Save the screenshot.

Output

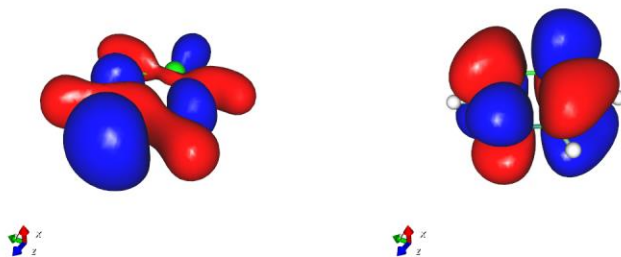


Figure 26 Visualize HOMO-LUMO for the Pyridine molecule

3) Total SCF Density (Closed Shell Systems)

The total SCF density $\rho(r)$ is expressed as

$$\rho(r) = \sum_i \lambda_i |\phi_i(r)|^2$$

where λ_i denotes the occupation number. $\lambda_i = 2$ for restricted closed shell system. $\phi_i(r)$ is the wave function of molecular orbital i .

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with fchk file. Choice: 'fchk2wfx -> wfx (Restrict Closed Shell)'.
- Interface with wfx file. Choice: 'Generate Total Density (Restrict Closed Shell) -> cub', 'Good (Slow)'.
- Read the cube file. Save the screenshot.

Output

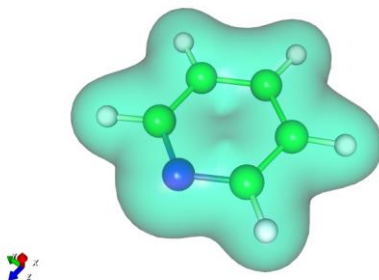


Figure 27 Visualize Total SCF density for the Pyridine molecule

4) Natural Transition Orbitals

Let D^{MO} denotes the transition density matrix in MO space from CIS, RPA or TDDFT (TDA approximation, Casida Ansatz) calculations. The dimension of this matrix is $M \times N$, where M and N represent the number of occupied and virtual MOs, respectively. The NTOs are obtained by the linear transformation of canonical MOs, with the unitary transformation matrices U ($M \times M$) and V ($N \times N$) obtained by singular value decomposition (SVD) of D^{MO} , i.e.

$$D^{MO} = U^T \Lambda V$$

where $\Lambda = \text{diag}\{\sqrt{\lambda_i}\}$ is a diagonal matrix and λ_i denotes the contribution of each pair of NTOs. The λ_i is essential to calculate the hole-particle densities.

The natural transition orbital participation ratio is defined as

$$PR_{NTO} = \frac{(\sum_i \lambda_i)^2}{\sum_i \lambda_i^2}$$

You can read these references for more details.

- Martin, R.L., *Natural transition orbitals*. Journal of Chemical Physics, 2003. **118**(11): p. 4775-4777.
- Plasser, F. and H. Lischka, *Analysis of Excitonic and Charge Transfer Interactions from Quantum Chemical Calculations*. Journal of Chemical Theory and Computation, 2012. **8**(8): p. 2777-2789.
- Plasser, F., et al., *New tools for the systematic analysis and visualization of electronic excitations. II. Applications*. Journal of Chemical Physics, 2014. **141**(2).
- Plasser, F., M. Wormit, and A. Dreuw, *New tools for the systematic analysis and visualization of electronic excitations. I. Formalism*. Journal of Chemical Physics, 2014. **141**(2).

Method 1:

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Use G09 to calculate the NTO. Here, we get the fchk for each excited state.

Procedure:

- Interface with fchk file. Choice: 'fchk2wfx -> wfx (Restrict Closed Shell)'.
- Interface with wfx file. Choice: 'Generate Molecular Orbitals -> cub', 'HOMO-2~LUMO+2', 'Good (Slow)'.
- Read a pair of cube files, with Hole and Particle. Save the screenshot.

Output

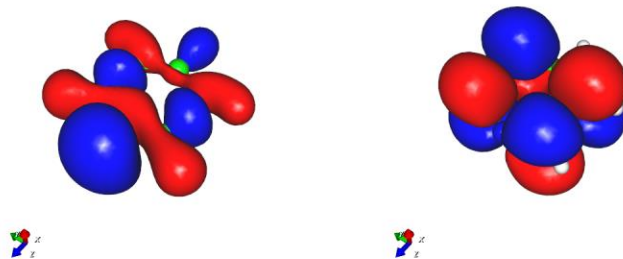


Figure 28 NTO of the 2nd excited state for the Pyridine molecule. $\sqrt{\lambda} = 9.9959$

Method 2:

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with rwf dumped file. Extract 635R.
- Interface with fchk file. Choice: 'fchk2wfx -> wfx (NTO, CIS)'.
- Interface with wfx file. Choice: 'Generate Molecular Orbitals -> cub', 'HOMO-2~LUMO+2', 'Good (Slow)'.
- Read a pair of cube files, with Hole and Particle. Save the screenshot.

Output

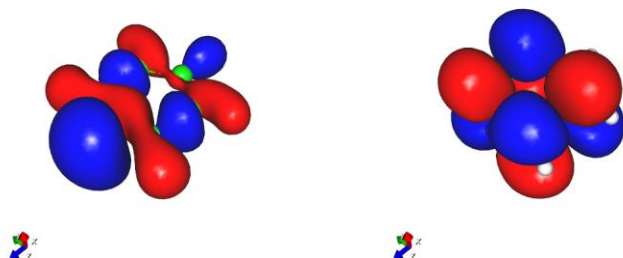


Figure 29 NTO of the 2nd excited state for the Pyridine molecule. $\sqrt{\lambda} = 9.9959$

Method 3:

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with rwf dumped file. Extract 514R, 633R.
- Interface with fchk file. Choice: 'fchk2wfx -> wfx (NTO, TDMAO)'.
- Interface with wfx file. Choice: 'Generate Molecular Orbitals -> cub', 'HOMO-2~LUMO+2', 'Good (Slow)'.
- Read a pair of cube files, with Hole and Particle. Save the screenshot.

Output

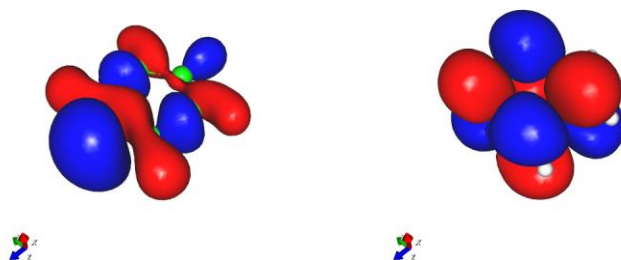


Figure 30 NTO of the 2nd excited state for the Pyridine molecule. $\sqrt{\lambda} = 9.9959$

5) Hole, Particle and Transition Densities

Let's denote hole and particle NTOs as $\psi_i^H(r)$ and $\psi_i^P(r)$, respectively, and then we can define hole $\rho_H(r)$ and particle $\rho_P(r)$ densities as

$$\rho_H(r) = \lambda_m |\psi_m^H(r)|^2$$

$$\rho_P(r) = \lambda_m |\psi_m^P(r)|^2$$

$$\rho_T(r) = \sqrt{\lambda_m} \psi_m^H(r) \psi_m^P(r)$$

if we only consider the NTOs corresponding to the largest eigenvalue (contribution), i.e. the largest λ_m .

The full expressions for the above definitions are

$$\rho_H(r) = \sum_i \lambda_i |\psi_i^H(r)|^2$$

$$\rho_P(r) = \sum_i \lambda_i |\psi_i^P(r)|^2$$

$$\rho_T(r) = \sum_i \sqrt{\lambda_i} \psi_i^H(r) \psi_i^P(r)$$

Please read these references for more details:

- Plasser, F., et al., *New tools for the systematic analysis and visualization of electronic excitations. II. Applications*. Journal of Chemical Physics, 2014. **141**(2).
- Plasser, F., M. Wormit, and A. Dreuw, *New tools for the systematic analysis and visualization of electronic excitations. I. Formalism*. Journal of Chemical Physics, 2014. **141**(2).

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with wfx file with NTO wave function information. Choice: 'Generate Hole, Particle and Transition Densities (1NTO)', 'Good (Slow)'.
- Read a pair of cube files, with Hole and Particle Densities. Save the screenshot.

Output

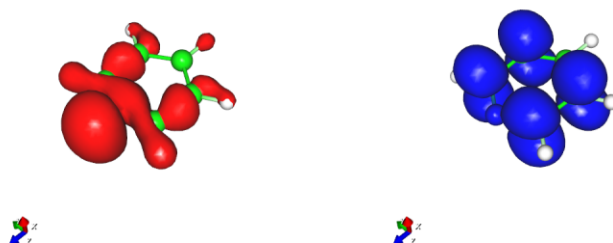


Figure 31 Hole and Particle Densities of the second excited state for the Pyridine molecule.

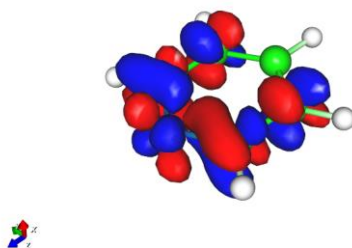


Figure 32 Transition Density of the second excited state for the Pyridine molecule.

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with wfx file with NTO wave function information. Choice: 'Generate Hole, Particle and Transition Densities (AllNTO)', 'Good (Slow)'.

- Read a pair of cube files, with Hole and Particle Densities. Save the screenshot.

Output

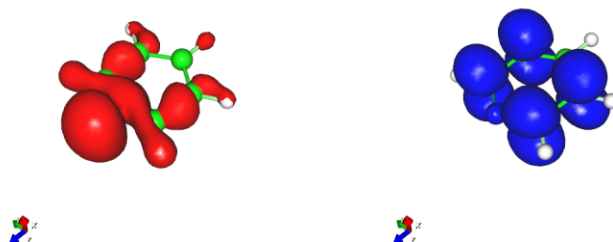


Figure 33 Hole and Particle Densities of the second excited state for the Pyridine molecule.

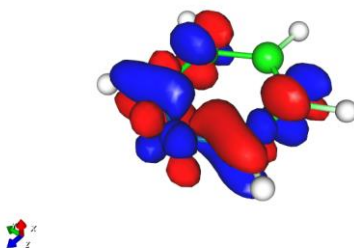


Figure 34 Transition Density of the second excited state for the Pyridine molecule.

6) Natural Orbitals (Use Total CI Density)

Natural Orbitals are obtained by the linear transformation of MOs, where the transformation matrix is obtained by diagonalizing a state density matrix. Let's consider a density matrix D_α for state α with respect to an orthonormal basis. It can be diagonalized as

$$U^T D_\alpha U = \Lambda$$

where $\Lambda = \text{diag}(\lambda_i)$ is a diagonal matrix and λ_i represents the occupation number of a natural orbital. As the reference pointed out, the sum of occupation numbers is identical to the number of electrons, i.e. $n = \sum_i \lambda_i$. The unitary matrix U gives the linear transformation from MOs to Natural Orbitals.

Please read these references for more details:

- Plasser, F., et al., *New tools for the systematic analysis and visualization of electronic excitations. II. Applications*. Journal of Chemical Physics, 2014. **141**(2).

- Plasser, F., M. Wormit, and A. Dreuw, *New tools for the systematic analysis and visualization of electronic excitations. I. Formalism*. Journal of Chemical Physics, 2014. **141**(2).

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with rwf dumped file. Extract 514R.
- Interface with fchk file. Choice: 'fchk2wfx -> wfx (NO, Total-CI)'.
- Interface with wfx file. Choice: 'Generate Molecular Orbitals -> cub', 'All', 'Good (Slow)'.
- Read a pair of cube files. Save the screenshot.

Output

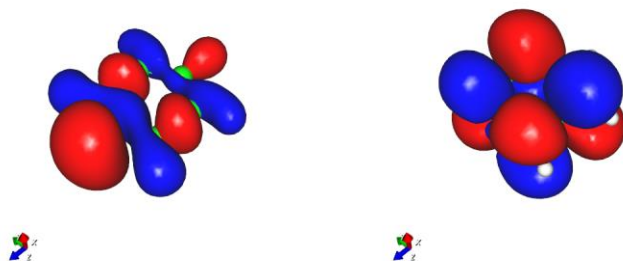


Figure 35 Natural Orbitals (Second excited state). Left: $n_i = 1.0116$. Right: $n_i = 0.9941$.

7) Natural Difference Orbitals (Use Total CI Density)

The Natural Difference Orbitals (NDOs) are built in the same way as Natural Orbitals. For NDOs, we have to diagonalize a difference matrix of two state density matrices instead of a state density matrix. Let's denote the state density matrix of α excited state as D_α and of ground state as D_0 , respectively. Then, the difference matrix $D_{0\alpha}$ is expressed as

$$D_{0\alpha} = D_\alpha - D_0$$

By diagonalizing this matrix, we obtain a linear transformation matrix from MOs to NDOs. The eigenvalues are important in defining the Detachment and Attachment Densities.

Please read these references for more details:

- Plasser, F., et al., *New tools for the systematic analysis and visualization of electronic excitations. II. Applications*. Journal of Chemical Physics, 2014. **141**(2).
- Plasser, F., M. Wormit, and A. Dreuw, *New tools for the systematic analysis and visualization of electronic excitations. I. Formalism*. Journal of Chemical Physics, 2014. **141**(2).

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with rwf dumped file. Extract 514R.
- Interface with fchk file. Choice: 'fchk2wfx -> wfx (NDO, Total-Cl)'.
- Interface with wfx file. Choice: 'Generate Molecular Orbitals -> cub', 'All', 'Good (Slow)'.
- Read a pair of cube files. Save the screenshot.

Output

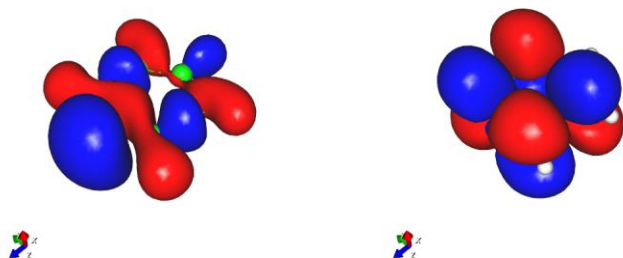


Figure 36 Natural Difference Orbitals. Left: $d_1 = -1.0100$. Right: $a_1 = 1.0043$.

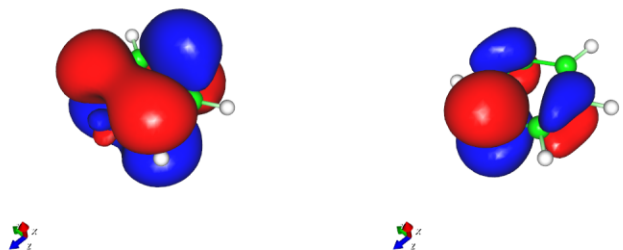


Figure 37 Natural Difference Orbitals. Left: $d_2 = -0.2720$. Right: $a_2 = 0.2722$.

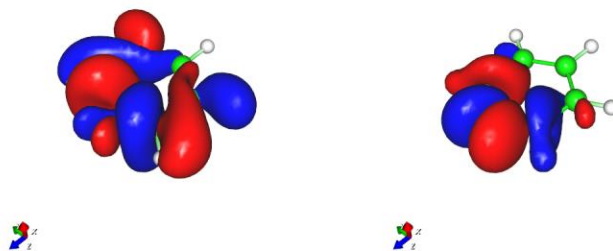


Figure 38 Natural Difference Orbitals. Left: $d_3 = -0.0943$. Right: $a_3 = 0.0941$.

8) Natural Orbitals (Use Total Rho CI Density)

Instead of using Total CI Density, we are using Total Rho CI Density here to get Natural Orbitals.

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with rwf dumped file. Extract 514R, 633R.
- Interface with fchk file. Choice: 'fchk2wfx -> wfx (NO, Total-Rho-CI)'.
- Interface with wfx file. Choice: 'Generate Molecular Orbitals -> cub', 'All', 'Good (Slow)'.
- Read a pair of cube files. Save the screenshot.

Output

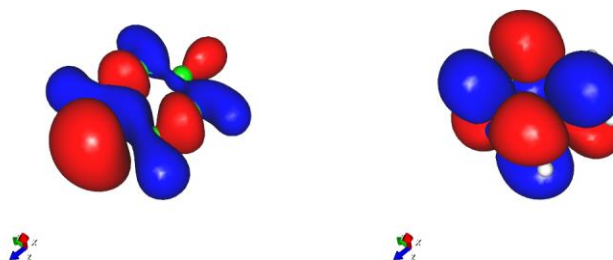


Figure 39 Natural Orbitals (Second excited state). Left: $n_i = 1.0008$. Right: $n_i = 0.9921$.

9) Natural Difference Orbitals (Use Total Rho CI Density)

Instead of using Total CI Density, we are using Total Rho CI Density here to get the NDOs.

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- Interface with rwf dumped file. Extract 514R, 633R.
- Interface with fchk file. Choice: 'fchk2wfx -> wfx (NDO, Total-Rho-Cl)'.
- Interface with wfx file. Choice: 'Generate Molecular Orbitals -> cub', 'All', 'Good (Slow)'.
- Read a pair of cube files. Save the screenshot.

Output

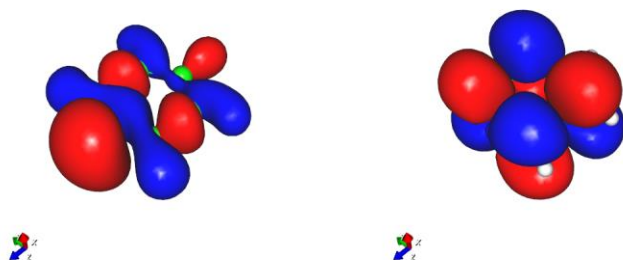


Figure 40 Natural Difference Orbitals. Left: $d_1 = -0.9992$. Right: $a_1 = 0.9992$.

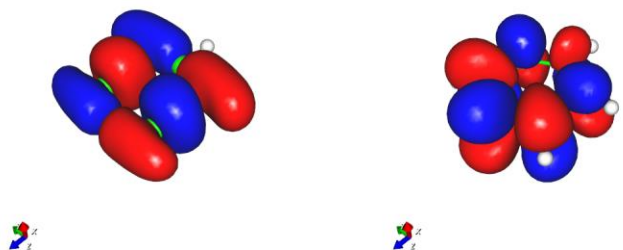


Figure 41 Natural Difference Orbitals. Left: $d_2 = -0.0007$. Right: $a_2 = 0.0007$.

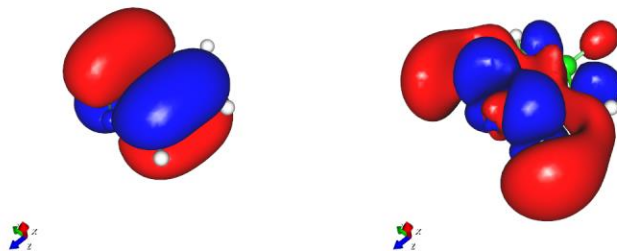


Figure 42 Natural Difference Orbitals. Left: $d_3 = -0.0003$. Right: $a_3 = 0.0002$.

10) Detachment and Attachment Densities

Suppose the eigenvalues obtained by diagonalizing a difference matrix $D_{0\alpha}$ are represented by a diagonal matrix $diag(\lambda_i)$. Let build two other diagonal matrices: $diag(d_i)$ and $diag(a_i)$, where

$$d_i = -\min(\lambda_i, 0)$$

$$a_i = \max(\lambda_i, 0)$$

The NDOs are denoted by $\psi_i(r)$. Now, we can define the detachment and attachment densities using following equations

$$\rho_d(r) = \sum_i d_i |\psi_i(r)|^2$$

$$\rho_a(r) = \sum_i a_i |\psi_i(r)|^2$$

Please read these references for more details:

- Plasser, F., et al., *New tools for the systematic analysis and visualization of electronic excitations. II. Applications*. Journal of Chemical Physics, 2014. **141**(2).
- Plasser, F., M. Wormit, and A. Dreuw, *New tools for the systematic analysis and visualization of electronic excitations. I. Formalism*. Journal of Chemical Physics, 2014. **141**(2).

Example: In this example, we perform TDDFT calculation for Pyridine molecule using Gaussian 09 at B3LYP/6-31G(d) level of theory.

Procedure:

- **Interface with wfx file with NDO wave function information. Choice: ‘Generate Attachment and Detachment Densities (NDOs)’, ‘Good (Slow)’.**
- **Read a pair of cube files, with Detachment and Attachment Densities. Save the screenshot.**

Output

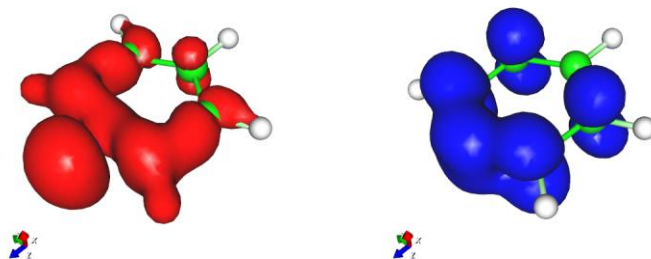


Figure 43 Detachment and Attachment Densities of the second excited state for the Pyridine molecule.

11) Generate Grids Efficiently

Suppose we have finished TDDFT calculations for a large conjugated molecule with more than 200 carbon atoms. If we want to visualize HOMO-LUMO, it may take hours to generate grids for such system. In this situation, we want to generate grids efficiently, which means fast and keep most important properties.

GPView can achieve this goal in two ways:

- Make the grids sparse.
- Make the 'box' smaller.

a) Sparse Grids

The default setting for GPView is generating points every 0.3 a.u in each direction. There are three options to change this setting:

- Bad (fast) – 0.9 a.u.
- Medium – 0.6 a.u.
- Good (Slow) – 0.3 a.u.

The first choice will be 27 times faster while the second is 8 times faster; therefore, it may take you up to 5 minutes to generate grids with more than 200 atoms.

Note: if your choice is Medium or Bad, you need to use other software, like GaussView to visualize final cube file, because there is an unsolved bug in GPView so far.

b) Smaller Box

Sometimes we don't want to lose any accuracy, however, generate grids for the whole molecule may not be necessary. Therefore, we can consider generate grids only for smaller area we are interested in. There is a hidden function that allows you to specify an area in GPView.

You can create a file 'local.lpar' in the directory contains other input files. In the 'local.lpar' file, you need to type in some coordinates (x, y, z). If you only give one coordinate, it will be the center of the box (default volume $12 \text{ a.u.} \times 12 \text{ a.u.} \times 12 \text{ a.u.}$). Otherwise, GPView will recalculate the center and boundary of the box.

Example, if you create a file local.lpar with input

```
0.0 0.0 0.0
```

Then, the center of the cube is (0.0, 0.0, 0.0).

If the input is

```
0.0 0.0 0.0
```

```
1.0 1.0 1.0
```

The center will be (0.5, 0.5, 0.5).

If you give a bunch of coordinates of atoms of the large molecule, GPView will calculate the center first, and then calculate the boundary of the box.

Examples

Acknowledgement

We would like to express our sincerely gratitude to

- Dr. Vladimir Chernyak
- Dr. H. Bernhard Schlegel

for their help.