

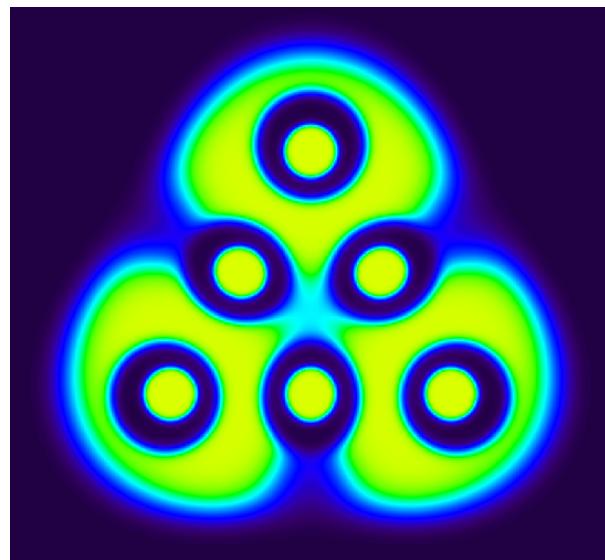
MULTIWFN

A Multifunctional Wavefunction Analyzer

Version 2.3.2

Software Manual

2012-Mar-27



Tian Lu

(Sobereva@sina.com)

School of Chemical and Biological Engineering,

University of Science and Technology Beijing, Beijing, China

All users must read

I know, most users are unwilling to read lengthy manual. Since Multiwfn is a heuristic and user-friendly program, it is unnecessary to read through the whole manual before using. However, the contents in this section you should never skip.

1. If the error “No executable for file 11.exe” appears in screen when Multiwfn is invoking Gaussian, you should set up Gaussian environment variable first. For Windows version, you can refer Appendix 1.
2. For getting started quickly, you can directly read Chapter 1 & 4. After that if you want to learn Multiwfn in detail, read Chapter 2 & 3.
3. The so-called “current directory” in this manual is the path where you invoking Multiwfn. If you boot up Multiwfn by clicking the icon of executable file, the “current directory” is where the executable file is posited in. As for command-line mode, if you are in “D:\study\” directory when invoking Multiwfn, then “D:\study\” is “current directory”.
4. Don’t put Multiwfn to the directory with too long name, including executable file name the path should less than 80 characters. So putting Multiwfn into “desktop” folder of Windows system is deprecated, the corresponding path is always too long and tends to exceed the threshold.
5. Please feel free to ask any question about using Multiwfn by emailing the author (Sobereva@sina.com), but ensure that you have read this manual carefully to find resolution. \(^{o}\)^(^o)/

Linux users must read

1. See Section 2.1.2 on how to install Linux version of Multiwfn.
2. When graphical user interface (GUI) appears, the graph will not be shown automatically, you have to do something to active the graph first, e.g. clicking "up" button or dragging a scale bar on GUI.
3. Due to an annoying bug in current version of DISLIN graphical library, if you would like to view the isosurface of just generated grid data in Multiwfn, you have to make sure that the number of grid points in x,y,z directions are identical, e.g. 80*80*80. So, don't choose 1, 2 or 3 in grid setting stage (namely low/middle/high quality grid) if you want to preview isosurface in Multiwfn, because in these three modes Multiwfn will determine grid points in each directions automatically according to molecule size.
4. There is no evident difference in calculation speed between prebuilt Windows and Linux version, since they are both compiled by intel fortran compiler with similar options.
5. I cannot guarantee that Linux version is as robust as Windows version, since all of my developments and most debugging are performed in Windows environment. Windows version is more recommended to be used.

About the manual

This manual is organized as following sequence:

Chapter 1 Overview: Briefly describe what is Multiwfn and what can do by Multiwfn.

Chapter 2 General informations: Introduce all other aspects beyond specific functions of Multiwfn, such as how to install, how to use, supported file types, limitations, etc.

Chapter 3 Functions: Describe all functions of Multiwfn in detail.

Chapter 4 Tutorial: Plenty of practical examples for helping users to use Multiwfn.

Chapter 5 Skills: Some useful skills in using Multiwfn.

Appendix

Italics font is used for input commands through this manual.

This manual doesn't only tell users how to use Multiwfn, but also introduces related theories, I hope people who don't have any interesting in Multiwfn can also benefit from this manual.

Content

All users must read	i
Linux users must read	ii
About the manual	iii
1 Overview	1
2 General informations	4
2.1 Install	4
2.1.1 Windows Version	4
2.1.2 Linux Version	4
2.2 Using Multiwfn	4
2.3 The files of Multiwfn	5
2.4 Parallel implementation	5
2.5 Supported wavefunction types and file types	5
2.6 Supported real space functions	7
2.7 Graphic formats and picture size	14
3 Functions	16
3.1 Defining molecular fragment (-3, -4)	16
3.2 Showing molecular structure and previewing orbitals / isosurfaces(0)	16
3.3 Outputting all properties at a point (1)	18
3.4 Outputting and plotting specific property in a line (3)	18
3.5 Outputting and plotting specific property in a plane (4)	19
3.5.1 Graph type	20
3.5.2 Setting up grid, plane and plotting region	24
3.5.3 Setting contour lines	25
3.5.4 Plot critical points, paths and interbasin paths on plane graph	25
3.6 Outputting and plotting specific property within a spatial region (5)	27
3.7 Custom operation, promolecular and deformation properties (subfunction 0, -1, -2 in main function 3, 4, 5)	29
3.7.1 Custom operation for multiple wavefunctions (0)	29
3.7.2 Promolecular and deformation properties (-1, -2)	29
3.7.3 Generation of atom wavefunctions	30
3.7.4 Sphericalization of atom wavefunction	31
3.8 Checking & Modifying wavefunction (6)	31
3.9 Population analysis (7)	33
3.9.1 Hirshfeld population (1)	33

3.9.2 Voronoi deformation density (VDD) population (2)	34
3.9.3 Mulliken atom & basis function population analysis (5)	34
3.9.4 Löwdin population analysis (6)	36
3.9.5 Modified Mulliken population defined by Ros & Schuit (SCPA) (7)	36
3.9.6 Modified Mulliken population defined by Stout & Politzer (8)	37
3.9.7 Modified Mulliken population defined by Bickelhaupt (9)	37
3.9.8 Becke population (10)	38
3.9.9 Atomic dipole moment corrected Hirshfeld population (ADCH, 11)	38
3.9.10 Charges from electrostatic potentials using a grid based method (CHELPG, 12)	38
3.9.11 Merz-Kollmann charge	40
3.10 Orbital composition analysis (8)	40
3.10.1 Output basis function, shell and atom composition in a specific orbital by MPA, Stout-Politzer and SCPA approaches (1, 2, 3)	40
3.10.2 Define fragment 1 and 2 (-1, -2)	41
3.10.3 Output composition of fragment 1 and inter-fragment composition by MPA, Stout-Politzer and SCPA approaches (4, 5, 6)	42
3.10.4 Decompose orbitals to atomic contribution by Hirshfeld partition (7)	43
3.10.5 Orbital composition analysis by natural atomic orbital approach (8)	43
3.11 Bond order analysis (9)	44
3.11.1 Mayer 2-center bond order analysis (1)	44
3.11.2 Multi-center bond order analysis (2)	45
3.11.3 Wiberg bond order analysis in Löwdin orthogonalized basis (3)	46
3.11.4 Mulliken bond order analysis (4) and decomposition (5)	47
3.12 Plotting total, partial, and overlap population density-of-states (10)	47
3.12.1 Theory	47
3.12.2 Input file	49
3.12.3 Options for plotting DOS	49
3.13 Plotting IR/Raman/UV-Vis spectrum (11)	50
3.13.1 Theory	51
3.13.2 Input file	52
3.13.3 Options for plotting	53
3.14 Topology analysis (2)	53
3.14.1 Theory	53
3.14.2 Search critical points	56
3.14.3 Generate paths	57
3.14.4 Generate interbasin surfaces	58
3.14.5 Visualize, analyze, modify and export results	59
3.15 Quantitative analysis of molecular surface (12)	60
3.15.1 Theory	60
3.15.2 Basic algorithm	62
3.15.3 Setting parameters	64
3.16.4 Options in post-process stage	64
3.16 Process grid data (13)	66
3.16.1 Output present grid data to Gaussian cube file (0)	66

3.16.2 Output value of all points with x/y/z coordinate (1)	66
3.16.3 Output data in a XY/YZ/XZ plane (2,3,4)	66
3.16.4 Output average data of XY/YZ/XZ planes in a range of Z/X/Y (5,6,7)	66
3.16.5 Output data in any plane by specifying three atom indices or three points (8,9)	66
3.16.6 Output data in specified value range (10)	67
3.16.7 Grid data calculation (11)	67
3.16.8 Map values of a cube file to specified isosurface of present grid data (12)	67
3.16.9 Set value of those grid points that are distant from some atoms (13)	68
3.16.10 Set value of those grid points outside overlap region of two fragments (14)	68
3.16.11 If data value is within certain range, set it to a specified value (15)	68
3.16.12 Scale data range (16)	68
3.16.13 Show statistic data (17)	69
3.100 Utilities (100)	69
3.100.1 Draw scatter graph between two functions and generate their cube files (1)	69
3.100.2 Output molecular structure to a pdb file (2)	70
3.100.3 Calculate molecular van der Waals volume (3)	70
3.100.4 Integrate a function in whole space (4)	71
3.100.5 Show overlap integral of alpha and beta orbital (5)	72
3.100.6 Monitor SCF convergence process (6)	72
3.100.7 Generate Gaussian input file with initial guess from converged wavefunction (7)	73
3.100.8 Generate Gaussian input file with initial guess from fragment wavefunctions (8)	73
3.100.9 Evaluate coordination number of all atoms	75
3.100.10 Analyze charge-transfer	75
3.100.11 Calculate overlap between two orbitals in whole space	77

4 Tutorial 79

4.0 Viewing orbitals and structure	79
4.0.1 Viewing molecular orbitals of monofluoroethane	79
4.0.2 Viewing natural bond orbitals (NBO) of ethanol	80
4.1 Showing all properties of triplet water at given point	82
4.2 Topology analysis	83
4.2.1 QTAIM analysis for 2-pyridoxine 2-aminopyridine	83
4.2.2 LOL topology analysis for acetic acid	85
4.3 Plotting the spin density curve of triplet formamide along carbon and oxygen atoms	87
4.4 Plot properties in a plane	88
4.4.1 The color-filled map of electron density of hydrogen cyanide	88
4.4.2 Shaded surface map with projection of localized orbital locator in a plane of monofluoroethane	89
4.4.3 Contour map of electron density laplacian of uracil without contributions of two atoms	90
4.4.4 Contour map of electrostatic potential of chlorine trifluoride	91
4.4.5 Gradient+contour map with topology paths of electron density of hydrogen peroxide	92
4.4.6 Deformation map of electron density of acetyl chloride	95
4.5 Generate cube file and view isosurface	97

4.5.1 Study electron localization function of chlorine trifluoride	97
4.5.2 Study electron density laplacian of Sn ₂ H ₄	97
4.5.3 Study aromaticity of benzene by ELF- α and ELF- π	99
4.5.4 Study Fukui function for phenol	102
4.5.5 Study electron transfer of imidazole coordinated magnesium porphyrin	103
4.6 Modifying wavefunction	105
4.6.1 Delete certain Gauss functions	105
4.6.2 Remove contributions from certain orbitals to real space functions	106
4.6.3 Translate and duplicate graphene primitive cell wavefunction to periodic system	107
4.7 Calculate Hirshfeld charges for chlorine trifluoride	108
4.8 Molecular orbital composition analysis	109
4.8.1 Analysis on acetamide by Mulliken method	109
4.8.2 Analysis on water by natural atomic orbital method	112
4.9 Three-center bond order analysis of Li ₆ cluster	114
4.10 Plot total, partial and overlap density-of-state (DOS) graph of water molecule	116
4.11 Plot infrared spectrum for NH ₃ BF ₃	116
4.12 Quantitative analysis of molecular surface of phenol	117
4.13 Process grid data	122
4.13.1 Extract data points in a plane	123
4.13.2 Perform mathematical operation on grid data	123
4.13.3 Scaling numerical range of grid data	124
4.13.4 Screen isosurfaces outside overlap region of two fragments	124
4.100 Other functions	126
4.100.1 Visual study of weak interaction	126
4.100.2 Calculate kinetic energy and nuclear attraction potential energy of phosgene by numerical integration	133
4.100.3 Perform simple energy decomposition by using combined fragment wavefunctions	134
4.100.4 Analyze charge-transfer during electron excitation	135
5 Skills	138
5.1 Make Multiwfn support more quantum chemistry programs	138
5.2 Running Multiwfn silently	138
5.3 Running Multiwfn in batch mode (for Windows)	139
5.4 Copy outputs from command-line window to clipboard (for Windows)	140
5.5 Make command-line window capable to record more outputs (for Windows)	141
Appendix	143
1 Setting up environment variable for Gaussian in Windows	143
2 Routine names of real space functions	143

1 Overview

Multiwfn is a free, open-source, user-friendly, powerful and flexible program, aims for general wavefunction analysis, current version is running on Windows (32/64bit Windows XP/Vista/7) and 64bit Linux platform. The latest version can be downloaded at Multiwfn website <http://multiwfn.codeplex.com> without registering. Multiwfn accepts several kinds of files as wavefunction input: .wfn/.wfx (Conventional / Extended PROAIM wavefunction file), .31~.40 (NBO plot file), .fch (Gaussian formatted check file).

Special points of Multiwfn

- (1) Very user-friendly. Multiwfn is designed as an interactive program, prompts shown in each step clearly instructs users what need to do next, Multiwfn also never print obscure messages, hence there is no any barrier even for beginners. Besides, there are about thirty practical examples in the manual, which would be very helpful for new users.
- (2) No third-part plotting softwares are required. A high-level graphical library DISLIN is invoked internally and automatically by Multiwfn for visualizing results, most of plotting parameters are adjustable by users in an interactive interface. Thus the procedure of wavefunction analysis is remarkably simplified, especially for studying distribution of real space function.
- (3) High efficiency. The code of Multiwfn is substantially optimized. Most parts are parallelized by OpenMP technology. For time-consuming tasks, the efficiency of Multiwfn exceeds analogous programs significantly. Meanwhile, the memory requirement is very low.

Basic functions of Multiwfn

- 1) Showing molecular structure and previewing orbitals (MO, NBO, natural orbital, etc.).
- 2) Outputting all supported properties at a point.
- 3) Outputting properties in a line and plotting curve map.
- 4) Outputting properties in a plane and plotting graph. Supported graph types including filled-color map, contour map, relief map (with/without projection), gradient map, vector field map.
- 5) Outputting properties in a spatial scope, data can be exported to Gaussian-type grid file (.cub) and can be viewed in terms of isosurface. Setting up grid is very flexible and convenient.
- 6) For calculation of real space function in one-, two- and three-dimensions, the operations between data generated from multiple files can be defined. One can draw difference map for electron density very easily.
- 7) Promolecule and deformation properties for all built-in real space functions are supported.
- 8) User can decide the wavefunction informations belong to which atoms and which orbitals will be used for subsequent analysis, uninteresting parts can be discarded.
- 9) Topology analysis for electron density (AIM analysis), ELF/LOL, laplacian etc. Critical points and gradient paths can be searched and visualized in terms of 3D or plane graph. Interbasin surfaces can be drawn.

- 10) Checking and modifying wavefunction. Including assigning, exchanging, translating & duplicating, etc.
- 11) Population analysis. Hirshfeld, VDD (Voronoi deformation density), Mulliken, Löwdin, Modified MPA (SCPA, Stout & Politzer, Bickelhaupt), Becke, ADCH (Atomic dipole moment corrected Hirshfeld), CHELPG and Merz-Kollmann methods are supported.
- 12) Orbital composition analysis. Mulliken, Stout & Politzer, Ros & Schuit (SCPA), Hirshfeld and natural atomic orbital (NAO) methods are supported to partition orbital composition.
- 13) Bond order analysis. Mayer bond order, multi-center bond order (up to 6-centers), Wiberg bond order in Löwdin orthogonalized basis and Mulliken bond order (can be decomposed to orbital contributions) are supported.
- 14) Plotting Total/Partial/Overlap population density-of-states (DOS).
- 15) Plotting IR/Raman/UV-Vis spectrum, parameters (broadening function, FWHM, etc.) can be controlled by user.
- 16) Quantitative analysis of molecular surface. Surface properties such as surface area, average value and std. of mapped functions can be computed, local minima and maxima can be located.
- 17) Processing Gaussian-type cube files. User can perform mathematical operations on grid data, set value in certain range, extract data in specified plane, etc.
- 18) Some utilities functions: Plotting scatter map for two functions in specific spatial scope, integrating a real space function in whole space by Becke method, evaluating overlap integral between alpha and beta orbital, monitoring SCF convergence process, generating Gaussian input file with initial guess from converged wavefunction or multiple fragment wavefunctions, calculate van der Waals volume, analyzing charge-transfer, etc.

The real space functions supported by Multiwfn

- 1 Electron density
- 2 Gradient norm of electron density
- 3 Laplacian of electron density
- 4 Value of orbital wavefunction
- 5 Electron Spin density
- 6 Hamiltonian kinetic K(r)
- 7 Lagrangian kinetic G(r)
- 8 Electrostatic potential from nuclear / atomic charges
- 9 Electron Localization Function (ELF) defined by Becke and the one defined by Tsirelson
- 10 Localized orbital locator (LOL) defined by Becke and the one defined by Tsirelson
- 11 Local information entropy
- 12 Total electrostatic potential (ESP)
- 13 Reduced density gradient (RDG)
- 14 Reduced density gradient with promolecular approximation
- 15 Sign(lambda2)*rho (The product of the sign of the second largest eigenvalue of electron density Hessian matrix and electron density)
- 16 Sign(lambda2)*rho with promolecular approximation

1 Overview

17 Fermi hole / correlation factor function (available for single determinant wavefunction)

18 Average local ionization energy

Multiwfn also reserves a custom function, the code can be filled by user for extending the function of Multiwfn.

Citing Multiwfn

If Multiwfn is used in your research, please cite as:

Tian Lu, Feiwu Chen, J. Comp. Chem. 33, 580 (2012)

I sincerely hope you can send me your accepted paper by E-mail (sobereva@sina.com), your paper will be listed on Multiwfn website

2 General informations

2.1 Install

2.1.1 Windows Version

What you need to do is just uncompress the program package. You can put Multiwfn to anywhere, however the name of the path (including file name) should less than 80 characters, else some unexpected problems may occur. Some functions in Multiwfn rely on Gaussian, you may need to setup environment variable for Gaussian manually, see Appendix 1.

2.1.2 Linux Version

1 Uncompress the program package, e.g. to /sob/Multiwfn. Notice that the path (including file name) should less than 80 characters.

2 Make sure that you have installed openmotif package, which provides libXm.so.4, Multiwfn cannot boot up without this file. If you haven't installed openmotif, download corresponding rpm package (e.g. openmotif-2.3.3-1.el5.3.x86_64.rpm) from <http://www.openmotif.org> and install it. openmotif 2.3.3-1 is compatible with Multiwfn, other openmotif version may also works.

3 Add below two sentences to your .bashrc file, e.g. /root/.bashrc, then reboot your shell

```
export KMP_STACKSIZE=40000000
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/sob/Multiwfn
```

KMP_STACKSIZE defines stacksize (in bytes) for each thread in parallel implementation, see Section 2.4 for detail.

4 Run `/sbin/sysctl -a|grep shmmmax` to check if the size of SysV shared memory segments is big enough (unit is in bytes), if the value is too small, Multiwfn may crashes when analyzing big wavefunction. In order to enlarge the size, for example you can add `kernel.shmmmax = 5000000000` to /etc/sysctl.conf and reboot system, then the upper limit will be enlarged to about 500MB.

Linux version of Multiwfn works well on Red Hat Enterprise Linux 6 Update 1 (64bit), I can not guarantee that the program is completely compatible with other Linux distributions. If system prompts you that some dynamical link libraries (.so files) are missing when booting up Multiwfn, try to find and install the packages which contain corresponding .so files.

2.2 Using Multiwfn

Using Multiwfn is very easy, simply read the prompts printed on screen, you will know what should input next. If you are stuck, please read corresponding section carefully in Chapter 3 or corresponding practical examples in Chapter 4.

The first thing to do after boot up Multiwfn is inputting the path of input file, e.g. `E:\lwd\bitboys.wfn`. If the input file is in current directory, you can just input file name without

the path of directory. If the input file of this time running Multiwfn is the same as last time, you can just input the letter *o*. (The path of the input file successfully readed at last time is recorded in settings.ini)

You can press CTRL+C or click “ \times ” button at right-top of command-line window any time to exit Multiwfn. When graphical window is showing on screen, you can click “RETURN” button to close the window, if there is no such button, click right mouse button on the graph to close it. If you click “ \times ” button at right-top of the window, the whole Multiwfn program rather than only the graphical window will be killed.

Multiwfn also support silent mode, in which user needn’t to click any button during running, it is useful for batch process or repeating similar tasks, please consult Chapter 5.

2.3 The files of Multiwfn

You will find following files after uncompressing Multiwfn package, only the bolded files are indispensable for running Multiwfn:

Multiwfn.exe (Windows) or **Multiwfn** (Linux) : The executable file of Multiwfn.

disdll_d.dll : Dynamic link library file of DISLIN graphical library. This file is not presented in Linux Version.

libiomp5md.dll (Windows) or **libiomp5.so** (Linux) : Intel OpenMP Runtime library.

settings.ini : All detail parameters for running Multiwfn are recorded here, most of them do not need to be modified often. If Multiwfn can’t find this file in current directory, a warning appears and default parameters will be used. The meanings of all parameters are not documented in this manual systematically, since they have already been commented in detail in settings.ini, only those important will be mentioned in this manual. I suggest users read through settings.ini.

“examples” folder : Some useful files and input files of examples in Chapter 4.

2.4 Parallel implementation

Most time-consuming codes of Multiwfn have been parallelized by OpenMP. If your computer is SMP architecture, you can feel much benefit from parallelization (For grid calculation, the speed up ratio versus the number of CPU cores is nearly linear). To enable parallel mode, just modify “nthreads” parameter in settings.ini to your situation. For example, your computer have a 12-cores CPU installed, then change “nthreads” to 12.

If Multiwfn crashes during parallel calculation, try to enlarge “ompstacksize” in settings.ini (for Windows version) or enlarge the value of the environment variable KMP_STACKSIZE (for Linux version).

2.5 Supported wavefunction types and file types

Wavefunction types supported by Multiwfn include restricted/unrestricted single-determinant wavefunction, restricted open-shell wavefunction and post-HF wavefunction (natural orbital formalism).

2 General informations

For basis function, cartesian or spherical harmonic gauss functions with angular moment up to g are supported.

There is no upper limit of the number of atoms / basis functions / GTFs / orbitals in Multiwfn, the actual upper limit is only decided by available memory on your computer.

Multiwfn determines the input file type by extension, currently supported file types include:

1 PROAIM wavefunction file (.wfn) : This format was first introduced by Bader's AIMPAC program, and currently supported by a lot of mainstream quantum chemistry softwares, such as Gaussian, GAMESS-US/UK, Firefly, Q-Chem. The informations of .wfn files include atomic coordinates/types, orbital energies/occupations/expansion coefficients respect to cartesian gauss type functions (GTF). Supported angular momentum of GTF is up to f . Notice that Multiwfn does not allow the .wfn file contains virtual orbitals (In general, .wfn files only contain occupied orbitals). The generation method of .wfn files is documented in the head of Chapter 4.

2 AIM extended wavefunction files (.wfx) : This is a newly introduced format as an extension of .wfn, the file can be generated by Gaussian09 since B.01 revision. Relative to .wfn format, .wfx supports higher data precision and infinite high angular moment of GTF. The most special point is the newly added electron density function (EDF) field, that is using multiple GTFs to represent inner core electron density of the wavefunction in which effective core potential (ECP) is used. Thus, the results of electron density analysis of the wavefunction used ECP are nearly identical to full electron wavefunction. Note that currently the real space functions supported EDF in Multiwfn are electron density, its gradient and laplacian, topology analysis for electron density and laplacian also support EDF. Currently the only supported GTF in EDF field is S-type (actually S-type is enough for fitting inner density, since which is nearly spherical symmetry). Likewise .wfn, Multiwfn does not allow virtual orbitals.

3 Gaussian formatted checkpoint file (.fch/.fchk) : This file type contains richer informations than .wfn/.wfx files, virtual orbital wavefunctions are also present, and this is the only supported file type for the functions in the Multiwfn that needed basis function informations. If you want to use .fch file as carrier for post-HF wavefunction, read the head of Chapter 4 carefully!

Notice that for single-determinant wavefunctions, before some calculations involving real space function (e.g. main function 3, 4, 5), virtual orbitals higher than LUMO+10 will be deleted automatically to speed up calculations, therefore you cannot analyze those orbitals after the calculation, unless you reboot Multiwfn. If you want to avoid this treatment, set "idelvirorb" in settings.ini to 0.

4 Plot files of NBO program (.31~.40) : The main purpose of supporting these file types is for visualizing PNAO/NAO/PNHO/NHO/PNBO/NBO/PNLMO/NLMO (their orbital coefficients are recorded in .32~.40 respectively), .31 recorded basis function informations. After boot up Multiwfn, you should input the path of .31 file first, and then input the path of one of .32~.40 files. Multiwfn can do almost anything that NBOview (a commerical software) can do and much more convenient to use.

5 Protein data bank format (.pdb) : This is the most widely used format for recording atom coordinates. pdb files don't carry any wavefunction information, but for the functions which only require atom coordinates, using pdb files as input is enough.

6 Charge files (.chg) : These plain text files are generated by some Multiwfn population functions, they carry element names (less than or equal to two characters), atom coordinates (first three columns, in angstrom) and charges (the fourth column), users can modify them manually. The main use of this format is for visualizing electrostatic potential from atomic charges.

7 Gaussian-type cube file (.cub or .cube) : This is the most prevalent volumetric data format, can be generated by vast computational chemistry softwares and can be recognized by the majority of molecular graphics programs. Atom coordinates and a set of grid data of real space function or multiple sets of grid data of molecular orbitals could be recorded in this file. Multiwfn only supports cubic grid, that means the three translation vectors must be parallelized with Cartesian axes. After cube file is loaded into Multiwfn, one can choose main function 0 to visualize isosurfaces, or use main function 13 to process the grid data.

7 Plain text file: This file type is only used for special functions, such as plotting DOS graph, plotting spectrum, generating Gaussian input file with initial guess. See explanations in corresponding sections.

Notice that different tasks need different types of informations, you should choose proper type of input file, see the table below. For example, the wavefunction represented by GTFs is enough for Hirshfeld population, so you can use .fch/.31~.40/.wfn/.wfx file as input, but .pdb, .chg and .cub files don't carry any wavefunction information hence cannot be used. While generating grid data of RDG function with promolecular approximation only requires atom coordinates, so all supported file formats can be used (except for plain text file). The requirements of information types by each function are described in Chapter 4.

File Format	Contained information types			
	Basis functions	GTFs	Atom coordinates	Grid data
	√	√	√	✗
.31~.40	✗	√	√	✗
.wfn	✗	√	√	✗
.wfx	✗	√	√	✗
.pdb	✗	✗	√	✗
.chg	✗	✗	√	✗
.cub/.cube	✗	✗	√	√
Plain text file	✗	✗	✗	✗

2.6 Supported real space functions

Real space function analysis is one of the most important function of Multiwfn, the supported real space functions are listed below. All wavefunctions are assumed to be real type, units are a.u.

Notice that for speeding up calculation, especially for big system, when evaluating a exponential function (except for some real space functions, such as 12, 14 and 16), if the exponent

2 General informations

is more negative than -40, then this evaluation will be skipped. The default cutoff value is safe enough and cannot cause detectable loss of precision even in quantitative analysis, you can also disable this treatment or adjust cutoff, see “expcutoff” in settings.ini.

1 Electron density

$$\rho(\mathbf{r}) = \sum_i \eta_i |\varphi_i(\mathbf{r})|^2 = \sum_i \eta_i \left| \sum_l C_{l,i} \chi_l(\mathbf{r}) \right|^2$$

Where η_i is occupation number of orbital i , χ is basis function. C is coefficient matrix, the element of i th row j th column corresponds to the expansion coefficient of orbital j respect to basis function i .

2 Gradient norm of electron density

$$|\nabla \rho(\mathbf{r})| = \sqrt{\left(\frac{\partial \rho(\mathbf{r})}{\partial x} \right)^2 + \left(\frac{\partial \rho(\mathbf{r})}{\partial y} \right)^2 + \left(\frac{\partial \rho(\mathbf{r})}{\partial z} \right)^2}$$

3 Laplacian of electron density

$$\nabla^2 \rho(\mathbf{r}) = \frac{\partial^2 \rho(\mathbf{r})}{\partial x^2} + \frac{\partial^2 \rho(\mathbf{r})}{\partial y^2} + \frac{\partial^2 \rho(\mathbf{r})}{\partial z^2}$$

The positive and negative value of this function correspond to electron density is locally depleted and locally concentrated respectively. The relationships between $\nabla^2 \rho$ and valence shell electron pair repulsion (VSEPR) model, chemical bond type, electron localization and chemical reactivity have been built by Bader and many other researchers.

If “laplfac” in settings.ini is set to other value rather than the default one 1.0, $\nabla^2 \rho(\mathbf{r})$ will be multiplied with this value. Setting it to negative value is convenient for analysis of electron density concentration.

4 Value of orbital wavefunction

$$\varphi_i(\mathbf{r}) = \sum_l C_{l,i} \chi_l(\mathbf{r})$$

When you select this function, you will be prompted to input the index of orbital i .

5 Electron spin density

$$\rho^s(\mathbf{r}) = \rho^\alpha(\mathbf{r}) - \rho^\beta(\mathbf{r})$$

If “ipolarpara” in settings.ini is set to 1, then spin polarization parameter function will be used instead

$$\zeta(\mathbf{r}) = \frac{\rho^\alpha(\mathbf{r}) - \rho^\beta(\mathbf{r})}{\rho^\alpha(\mathbf{r}) + \rho^\beta(\mathbf{r})}$$

2 General informations

The absolute value of ζ going from zero to unity corresponds to the local region going from unpolarized case to completely polarized case.

6 Hamiltonian kinetic energy density $K(\mathbf{r})$

The kinetic energy density is not uniquely defined, since the expected value of kinetic energy operator $\langle \varphi | -(1/2)\nabla^2 | \varphi \rangle$ can be recovered by integrating kinetic energy density from alternative definitions. One of commonly used definition is

$$K(\mathbf{r}) = -\frac{1}{2} \sum_i \eta_i \varphi_i^*(\mathbf{r}) \nabla^2 \varphi_i(\mathbf{r})$$

7 Lagrangian kinetic energy density $G(\mathbf{r})$

Relative to $K(\mathbf{r})$, the local kinetic energy definition given below guarantee positiveness everywhere, hence the physical meaning is more clear and be more commonly used. $G(\mathbf{r})$ is also known as positive definite kinetic energy density.

$$G(\mathbf{r}) = \frac{1}{2} \sum_i \eta_i |\nabla \varphi_i(\mathbf{r})|^2$$

8 Electrostatic potential from nuclear / atomic charges

$$V_{nu}(\mathbf{r}) = \sum_A \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|}$$

Where \mathbf{R}_A and Z_A denote position vector and nuclear charge of atom A, respectively. If pseudo-potential is used, then Z is the number of explicitly expressed electrons. When *chg* file is used as input, Z will stand for the atomic charges recorded in the file (the fourth column), at this time V_{nu} is useful for analyzing the difference between exact electrostatic potential and the electrostatic potential reproduced by atomic charges.

9 Electron localization function (ELF)

The larger the electron localization is in a region, the more likely the electron motion is confined within it. If electrons are completely localized, then they can be distinguished from the ones outside. Bader found that the regions which have large electron localization must have large magnitudes of Fermi hole integration. However, the Fermi hole is a six-dimension function and thus difficult to be studied visually. Becke and Edgecombe noted that spherically averaged like-spin conditional pair probability has direct correlation with the Fermi hole and then suggested electron localization function (ELF) in the paper *J. Chem. Phys.*, 92, 5397. The ELF used in Multiwfn is generalized for spin-polarized system, see *Acta Phys. -Chim. Sin.*, 27, 2786 (<http://www.whxb.pku.edu.cn/EN/abstract/abstract27788.shtml>) for derivation.

$$ELF(\mathbf{r}) = \frac{1}{1 + (D(\mathbf{r}) / D_0(\mathbf{r}))^2}$$

Where

2 General informations

$$D(\mathbf{r}) = \frac{1}{2} \sum_i \eta_i |\nabla \varphi_i(\mathbf{r})|^2 - \frac{1}{8} \left[\frac{|\nabla \rho_\alpha(\mathbf{r})|^2}{\rho_\alpha(\mathbf{r})} + \frac{|\nabla \rho_\beta(\mathbf{r})|^2}{\rho_\beta(\mathbf{r})} \right]$$

$$D_0(\mathbf{r}) = \frac{3}{10} (6\pi^2)^{2/3} [\rho_\alpha(\mathbf{r})^{5/3} + \rho_\beta(\mathbf{r})^{5/3}]$$

For close-shell system, since $\rho_\alpha = \rho_\beta = (1/2)\rho$, D and D_0 terms can be simplified as

$$D(\mathbf{r}) = \frac{1}{2} \sum_i \eta_i |\nabla \varphi_i(\mathbf{r})|^2 - \frac{1}{8} \frac{|\nabla \rho(\mathbf{r})|^2}{\rho(\mathbf{r})}$$

$$D_0(\mathbf{r}) = (3/10)(3\pi^2)^{2/3} \rho(\mathbf{r})^{5/3}$$

Savin *et al.* have reinterpreted ELF in the view of kinetic energy, see *Angew. Chem. Int. Ed. Engl.*, 31, 187, which made ELF also meaningful for Kohn-Sham DFT wavefunction even in post-HF wavefunction. They indicated that $D(\mathbf{r})$ reveals the excess kinetic energy density caused by Pauli repulsion, while $D_0(\mathbf{r})$ can be considered as Thomas-Fermi kinetic energy density. Since $D_0(\mathbf{r})$ is introduced into ELF as reference, what the ELF reveals is actually a *relative* localization.

ELF is within the range of [0,1]. A large ELF value means that electrons are greatly localized, indicating that there will be a covalent bond, a lone pair or inner shells of an atom involved. ELF has been widely used for a wide variety of systems, such as organic and inorganic small molecules, atomic crystals, coordination compounds, clusters, and for different problems, such as the atomic shell structure, classifications of chemical bonding modes, verifications of charge-shift bond.

Notice that there is a deficiency of ELF, sometimes with \mathbf{r} going beyond from molecular boundary, $D(\mathbf{r})$ decreases faster than $D_0(\mathbf{r})$ and then ELF reaches 1 (completely localized). To overcome the problem, Multiwfn automatically adds a minimal value 10^{-5} to $D(\mathbf{r})$, this treatment almost doesn't affect the ELF value in interesting regions. You can also disable this treatment by modifying “ELF_addminimal” in settings.ini to 0.

Tsirelson and Stash put forward an approximate version of ELF in *Chem. Phys. Lett.*, 351, 142, in which the actual kinetic energy term in $D(\mathbf{r})$ is replaced by Kirzhnits type second-order gradient expansion, that is

$$(1/2) \sum_i \eta_i |\nabla \varphi_i(\mathbf{r})|^2 \approx D_0(\mathbf{r}) + (1/72) |\nabla \rho(\mathbf{r})|^2 / \rho(\mathbf{r}) + (1/6) \nabla^2 \rho(\mathbf{r})$$

so that ELF is totally independent from wavefunction, and then can be used to analyze electron density from X-ray diffraction data. Of course Tsirelson's ELF can also be used to analyze electron density from quantum chemistry calculation, but is not as good as the ELF defined by Becke owing to the approximation introduced in kinetic energy term, however, qualitative conclusions can still be recovered in general.

If you want to use Tsirelson's definition of ELF, change “ELFOL_type” in settings.ini from 0 to 1. By the way, if “ELFOL_type” is set to 2, another formalism will be used:

$$\frac{1}{1 + D(\mathbf{r}) / D_0(\mathbf{r})}.$$

10 Localized orbital locator (LOL)

This is another function for locating high localization regions likewise ELF, defined by Schmider and Becke in the paper *J. Mol. Struct. (Theo.)*, 527, 51.

$$LOL(\mathbf{r}) = \frac{\tau(\mathbf{r})}{1 + \tau(\mathbf{r})}$$

Where

$$\tau(\mathbf{r}) = \frac{D_0(\mathbf{r})}{(1/2) \sum_i \eta_i |\nabla \varphi_i(\mathbf{r})|^2}$$

$D_0(\mathbf{r})$ for spin-polarized system and close-shell system are defined in the same way as in ELF.

LOL has similar expression compared to ELF. Actually, the chemically significant regions that highlighted by LOL and ELF are generally qualitative comparable, while Jacobsen pointed out that LOL conveys more decisive and clearer picture than ELF, see *Can. J. Chem.*, 86, 695. Obviously LOL can be interpreted in kinetic energy way as for ELF, however LOL can also be interpreted in view of localized orbital. Small (large) LOL value usually appears in boundary (inner) region of localized orbitals because the gradient of orbital wavefunction is large (small) in this area. The value range of LOL is identical to ELF, namely [0,1].

Multiwfn also supports the approximate version of LOL defined by Tsirelson and Stash (*Acta Cryst.*, B58, 780), namely the actual kinetic energy term in LOL is replaced by second-order gradient expansion, as what they do for ELF. This Tsirelson's version of LOL can be activated by setting "ELFLOL_type" to 1.

For special reason, if "ELFLOL_type" in settings.ini is changed from 0 to 2, another formalism will be used: $\frac{1}{1 + (1/\tau(\mathbf{r}))^2}$.

11 Local information entropy

Information entropy is a quantification of information, this theory was proposed by Shannon in his study of information transmission in noise channel, nowadays its application has been largely widened to other areas, including theoretical chemistry. For example, Aslangul and coworkers attempted to decompose diatomic and triatomic molecules into mutually exclusive space by minimizing information entropy (*Adv. Quantum Chem.*, 6, 93), Parr *et al.* discussed the relationship between information entropy and atom partition as well as molecular similarity (*J. Phys. Chem. A*, 109, 3957). The formula of Shannon's information entropy for normalized and continuous probability function is

$$S = - \int P(x) \ln P(x) dx$$

For chemical system, if $P(x)$ is replaced by $\rho(\mathbf{r})/N$, then the integrand may be called local information entropy of electrons

$$S(\mathbf{r}) = - \frac{\rho(\mathbf{r})}{N} \ln \frac{\rho(\mathbf{r})}{N}$$

Where N is the total number of electrons in current system. Integrating this function in whole space yields information entropy.

12 Total electrostatic potential (ESP)

Z_A is nuclear charge, if pseudopotential is used, then Z_A is the number of explicitly expressed electrons.

$$V_{tot}(\mathbf{r}) = V_{nu}(\mathbf{r}) + V_{ele}(\mathbf{r}) = \sum_A \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} - \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}'$$

This function measures the electrostatic interaction between a unit point charge placed at \mathbf{r} and the system of interest. A positive (negative) value implies that current position is dominated by nuclear (electronic) charges. Molecular electrostatic potential (ESP) has been widely used for prediction of nucleophilic and electrophilic sites for a long time. It is also valuable in studying hydrogen bonds, halogen bonds, molecular recognitions and the intermolecular interaction of aromatics. Moreover, based on statistical analysis, Murray and coworkers found a set of functions called GIPF, see *J. Mol. Struct. (THEOCHEM)*, 307, 55, which connects ESP in molecular surface and macroscopic properties. The links between ESP and covalent radii, electronegativity and system energy were reviewed in *Theor. Chem. Acc.*, 108, 134.

Notice that evaluating ESP is much more time-consuming than evaluating other functions. Also note that ESP in Multiwfn is evaluated exactly by nuclear attractive integrals rather than using approximate methods (such as multipole expansion, numerical Poisson equation), hence you may find the results generated by Multiwfn are somewhat different from those outputted by other quantum chemistry codes.

In order to speed up ESP evaluation, Multiwfn ignores some integrals that have little contributions. The threshold for ignoring is controlled by "espprecutoff" in settings.ini, enlarging this parameter results in more accurate ESP value, but also brings more computational cost. The ESP evaluated under default value is accurate enough in general cases.

13 Reduced density gradient (RDG)

RDG and Sign(lambda2)*rho are a pair of very important functions for revealing weak interaction region, for detail see *J. Am. Chem. Soc.*, 132, 6498 and the article *Visual research of weak interaction by Multiwfn* (in Chinese, can be downloaded at website of Multiwfn), the basic principles are also introduced in Section 4.100.1. RDG is defined as

$$RDG(\mathbf{r}) = \frac{1}{2(3\pi^2)^{1/3}} \frac{|\nabla\rho(\mathbf{r})|}{\rho(\mathbf{r})^{4/3}}$$

Notice that there is a parameter "RDG_maxrho" in settings.ini, if the value is set to x , then RDG function will be set to an arbitrary big value (100.0) where the electron density is larger than x . This mechanism allows uninteresting regions to be shielded when viewing isosurfaces of weak interaction regions. By default x is 0.05, you can nullify this treatment by setting the parameter to zero.

14 Reduced density gradient (RDG) with promolecular approximation

Weak interaction has significant influence on conformation of macromolecules, binding mode of proteins and ligands, unfortunately reproduction of electron density by *Ab initio* and grid

2 General informations

data calculation of RDG for such huge systems are always too time-consuming. Fortunately, it is found that weak interaction analysis under promolecular density is still reasonable. Promolecular density is simply constructed by superposing electron densities of free-state atoms and hence can be evaluated extremely quickly

$$\rho^{pro}(\mathbf{r}) = \sum_A \rho_A^{free,fit}(\mathbf{r} - \mathbf{R}_A)$$

Where $\rho_A^{free,fit}(\mathbf{r})$ is pre-fitted electron density function of atom A . The fitting coefficients at B3LYP/6-31G* level come from the supplement of *J. Am. Chem. Soc.*, 132, 6498, for the element in the first, second and third row the free-state atomic density is expanded by one, two and three S-type STOs respectively. If any atom in current system is heavier than Ar, the approximation is not available due to the lack of pre-fitted coefficients.

For efficiency consideration, if contribution from a H, C, N or O atom to the function value at a specific point is less than 0.00001, then the contribution will not be calculated, for huge system this treatment improves efficiency several times and result is almost not perturbed. You can also disable this treatment by setting “atomdenscut” in settings.ini to 0.

The parameter “RDGprodens_maxrho” in settings.ini is just the counterpart of “RDG_maxrho” for promolecular approximation situation.

15 Sign(lambda2)*rho

$$\Omega(\mathbf{r}) = \text{Sign}(\lambda_2(\mathbf{r}))\rho(\mathbf{r})$$

Where $\text{Sign}(\lambda_2(\mathbf{r}))$ means the sign of the second largest eigenvalue of electron density Hessian matrix at position \mathbf{r} .

16 Sign(lambda2)*rho with promolecular approximation

The real density used to determine $\text{Sign}(\lambda_2(\mathbf{r}))$ is approximated by promolecular electron density, see the description in function 14.

17 Fermi hole / correlation factor function

Fermi hole function reveals the decrease of probability of finding another like-spin electron at \mathbf{r}_2 owing to Pauli repulsion when a electron present at \mathbf{r}_1 . This function is only available for single determinant wavefunction in current version, the Fermi hole from σ spin is defined as

$$h^{\sigma\sigma}(\mathbf{r}_1, \mathbf{r}_2) = \frac{\rho_2^{\sigma\sigma}(\mathbf{r}_1, \mathbf{r}_2)}{\rho^\sigma(\mathbf{r}_1)} - \rho^\sigma(\mathbf{r}_2) = -\frac{\sum_{i \in \sigma} \sum_{j \in \sigma} \varphi_i(\mathbf{r}_1) \varphi_j^*(\mathbf{r}_1) \varphi_i^*(\mathbf{r}_2) \varphi_j(\mathbf{r}_2)}{\sum_{k \in \sigma} |\varphi_k(\mathbf{r}_1)|^2}$$

Value of 1, 2 and 3 of “fermiholespin” in settings.ini correspond to $\sigma=\alpha$, $\sigma=\beta$ and $\sigma=both$ respectively. The fermi hole in $\sigma=both$ circumstance can be written as

$$h(\mathbf{r}_1, \mathbf{r}_2) = -\frac{\sum_{i \in \alpha} \sum_{j \in \alpha} \varphi_i(\mathbf{r}_1) \varphi_j^*(\mathbf{r}_1) \varphi_i^*(\mathbf{r}_2) \varphi_j(\mathbf{r}_2) + \sum_{m \in \beta} \sum_{n \in \beta} \varphi_m(\mathbf{r}_1) \varphi_n^*(\mathbf{r}_1) \varphi_m^*(\mathbf{r}_2) \varphi_n(\mathbf{r}_2)}{\sum_{k \in \alpha} |\varphi_k(\mathbf{r}_1)|^2 + \sum_{l \in \beta} |\varphi_l(\mathbf{r}_1)|^2}$$

In Multiwfn, \mathbf{r}_1 is seen as reference point and \mathbf{r}_2 is seen as variable, to define the coordinate of reference point, modifying “refx”, “refy” and “refz” in settings.ini.

Fermi correlation factor is a function closely related to fermi hole,

$$f^\sigma(\mathbf{r}_1, \mathbf{r}_2) = \frac{h^\sigma(\mathbf{r}_1, \mathbf{r}_2)}{\rho^\sigma(\mathbf{r}_2)}$$

To use this function with $\sigma=\alpha$, $\sigma=\beta$ and $\sigma=\text{both}$, set "fermiholespin" to 4, 5 and 6 respectively.

18 Average local ionization energy

Average local ionization energy is written as

$$\bar{I}(\mathbf{r}) = \frac{\sum_i \rho_i(\mathbf{r}) |\varepsilon_i|}{\rho(\mathbf{r})}$$

Where $\rho_i(\mathbf{r})$ and ε_i are the electron density function and orbital energy of the i th molecular orbital, respectively. This function has many uses, for example reproducing atomic shell structure, measuring electronegativity, quantifying local polarizability and hardness, predicting sites for electrophilic or radical attack. There are many potential uses of $\bar{I}(\mathbf{r})$ waiting for further investigation. A review of $\bar{I}(\mathbf{r})$ has been given by Politzer *et al*, see *J. Mol. Model.*, 16, 1731.

100 User defined function

Multiwfn provides a custom function for users, this function corresponds to "userfunc" routine in function.f90. By filling codes by yourself, the function of Multiwfn can be extended. For examples, simply fill the code `userfunc=fgrad(x,y,z,'t')**2/8/fdens(x,y,z)` and recompile

Multiwfn, you get the integrand of Weizsacker functional $\tau_w[\rho] = \int \frac{|\nabla \rho(\mathbf{r})|^2}{8\rho(\mathbf{r})} d\mathbf{r}$. Don't worry

about how to write the code, consulting Appendix 2 and the codes of other functions in function.f90 you will find answer. If you want to define several custom functions, you can fill their codes in branch statement "if (iuserfunc==1) then ... else if (iuserfunc==2) then ... end if", where "iuserfunc" is an external parameter in settings.ini, by which you can select which custom function to be used, see comment in "userfunc" routine for example.

In order to cut down the length of the function list, potential energy density $V(\mathbf{r}) = -K(\mathbf{r}) - G(\mathbf{r})$ and energy density $E(\mathbf{r}) = G(\mathbf{r}) + V(\mathbf{r}) = -K(\mathbf{r})$ don't appear in the supported function list, since they can be obtained by simple computation of the functions listed above. However, if you want to directly calculate them, you can set "iuserfunc" parameter in settings.ini to 10 and 11, then the user defined function will be equivalent to potential energy density and energy density respectively.

2.7 Graphic formats and picture size

Benefiting from DISLIN graphical library, Multiwfn supports a lot of mainstream graphic

2 General informations

formats, including:

- 1 Postscript (ps)
- 2 Encapsulated postscript (eps)
- 3 Portable Document Format (pdf)
- 4 Windows metafile format (wmf)
- 5 Graphics Interchange Format (gif)
- 6 TIFF (tiff)
- 7 Portable Network Graphics (png)
- 8 Windows Bitmap format (bmp)

The graphic format of the picture exported by Multiwfns is controlled by “graphformat” parameter in settings.ini, you can set this parameter to the texts in the parentheses listed above, the default format is “png”.

For curve maps, the height and weight of picture are controlled by “graphlinefilewidth” and “graphlinefileheight” parameters in settings.ini respectively, “graph2Dfilewidth” and “graph2Dfileheight” are responsible for two-dimension data plotting (color-filled map, contour map, relief map, etc.), “graph3Dfilewidth” and “graph3Dfileheight” are responsible for three-dimension data plotting (isosurface graph).

3 Functions

This Chapter introduces each function of Multiwfn in detail, the numbers in the parentheses of secondary titles are corresponding indices in main menu, the ones in tertiary titles are corresponding indices in corresponding submenu. For helping user to choose proper type of input file (see Section 2.5), the informations needed by each function are shown in the final line of corresponding section.

3.1 Defining molecular fragment (-3, -4)

By this function, you can define the molecular fragment you are interested in (-3) and the fragment you are uninterested in (-4). When you enter function -3, input the total number of atoms in the fragment, and then input the atom indices in turn. The GTFs whose centers are not attributed to any atom in the fragment will be discarded (can be imagined as corresponding coefficients in all orbitals are set to zero), in the other words the wavefunction informations of all other atoms are discarded. This function affects all the following tasks which make use of GTF informations. Function -4 is the reverse of functon -3, the fragment you defined is the one will be discarded.

Why use this function? For large molecules, what you are interested in may be only a small part, by discarding GTFs of distant atoms, the calculation speed of time-consuming task such as generating cube file can increases evidently. In other circumstances, you may want to obtain the property only contributed by certain atoms, this function can be used to fulfill your object.

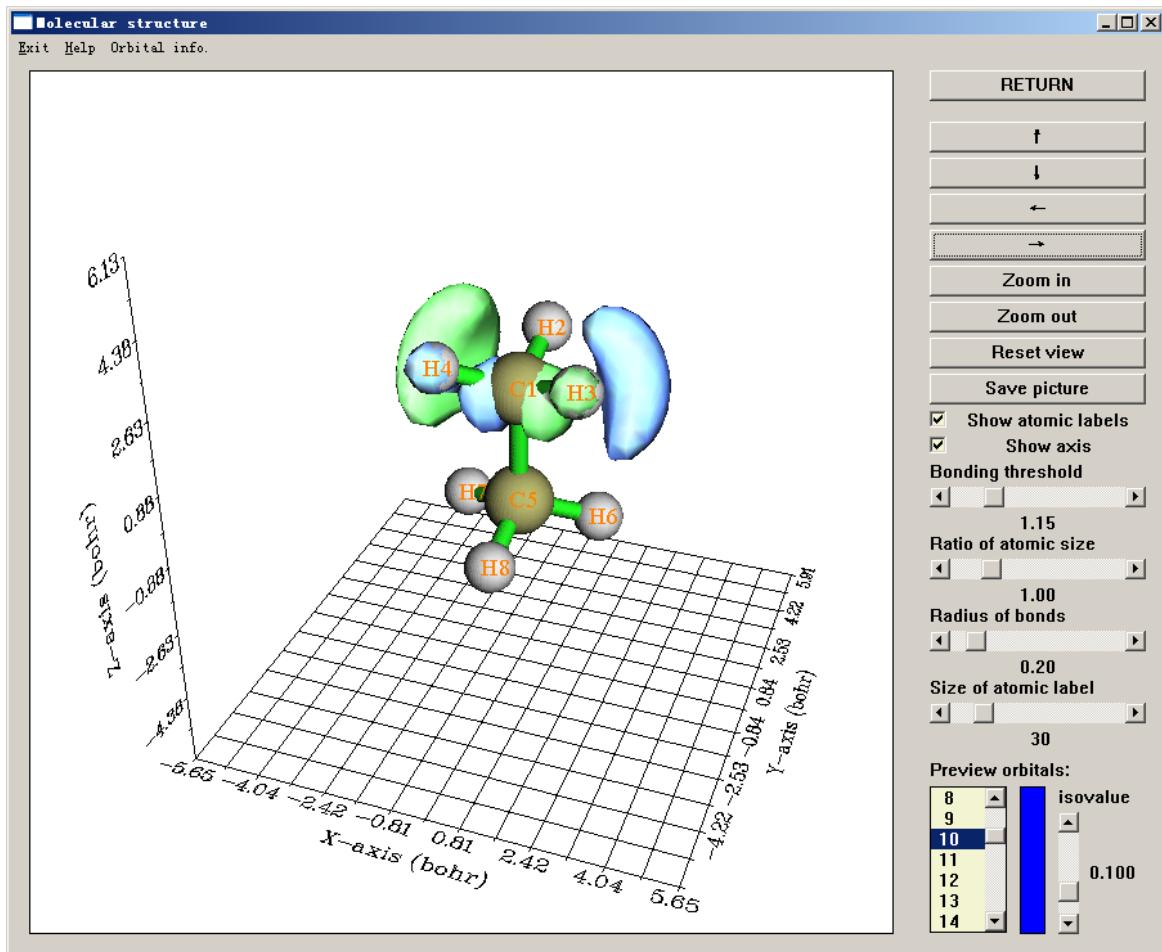
This function can be safely used together with function 6. For other task, please don't use this function, otherwise the results will be weird or meaningless. Notice that this function is irreversible, the discarded GTFs cannot be recovered, unless you reboot the program and reload the wavefunction file.

Informations needed: GTFs, atom coordinates.

3.2 Showing molecular structure and previewing orbitals / isosurfaces(0)

When function 0 is selected, all atom coordinates will be printed on text window. If the input file is Gaussian cube file, you will enter a GUI window, in which the isosurfaces can be visualized. If the input file is other types, below GUI window will pops up

3 Functions



In this graphical window, you can view molecular structure, preview orbital wavefunctions, most buttons are self-explained, I only explain some worthnoting options here.

By clicking “**Orbital info.**” and select the only option “show”, basic orbital informations will be printed on text window.

“**Ratio of atomic size**” is the ratio of one-fourth of atom radius shown on screen to its van der Waals (vdW) radius, so if the slide bar is dragged to 4.0, then what is shown is vdW surface.

Multiwfn determines if atoms are bonded by empirical distance criteria, if the distance between two atoms is short than 1.15 times of the sum of their vdW radii, they will be considered as bonded. You can adjust this criteria by dragging “**Bonding threshold**” slide bar.

Previewing orbitals: If inputted file carries wavefunction information, then all orbital indices will be listed on the right-bottom box, default selection is “none” (no orbital is shown). If you click one of orbital indices, Multiwfn will calculate grid data of wavefunction value for corresponding orbital, the progress is shown by blue bar. Once the bar is full, the orbital isosurface shows up immediately, the green and blue part correspond to positive and negative regions respectively. The “**isovalue**” slide bar controls the value of isosurface. Notice that this interface is not intended for generating high-quality grid data for orbital wavefunction but only for previewing, for efficiency consideration, the quality of grid data is very coarse, however you can controls the precision by changing “**nprevorbgrid**” in settings.ini. Sometimes you may see some regions of isosurfaces are blocked by black polygons, or green and blue regions are difficult to distinguish, you can rotate molecule to solve this problems.

Informations needed: GTFs (only for previewing orbitals), atom coordinates.

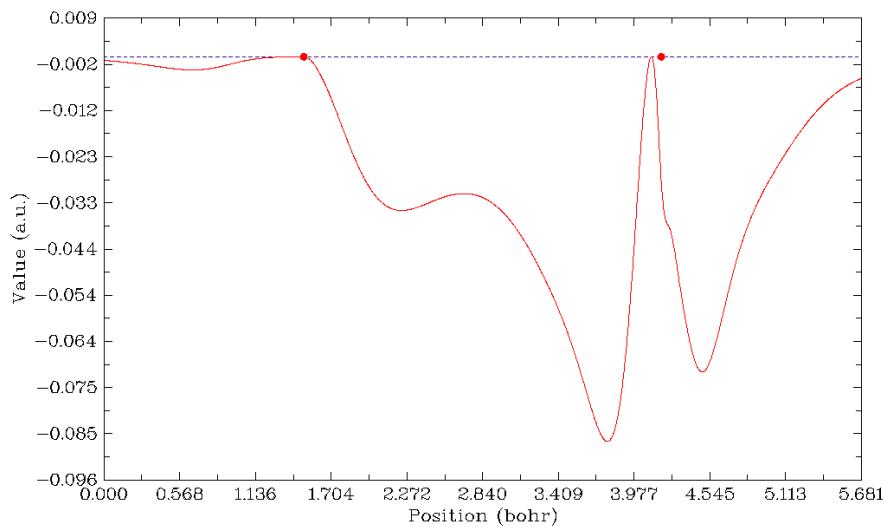
3.3 Outputting all properties at a point (1)

Input coordinate of a point or index of an atom, the value of all real space functions at given point or corresponding nuclear position will be printed on screen, as well as each component of gradient and Hessian matrix of specified function (default is electron density). The function can be specified by *f?*, for example *f9* select ELF, you can input *allf* to check all available functions. The orbital whose wavefunction value will be outputted can be selected by command *o?*, for example *o4* choose the fourth orbital. If the input file merely contains atom coordinates (such as *pdb* file), then only limited functions based on electron density will be outputted, the electron density used is promolecular density constructed from fitted free atom density, see the introduction of real space function 14 in Section 2.6.

Informations needed: GTFs, atom coordinates.

3.4 Outputting and plotting specific property in a line (3)

What you should do is just select a real space function and then define a line. There are two ways to define the line: (1) By inputting indices of two atoms, the line will be automatically extended by a small distance in each side, the extended distance can be adjusted by “aug1D” in settings.ini or by the option “0 Set size extension for mode 1”, default value is 1.5 bohr. (2) By inputting the coordinates of the two endpoints. Generally, the calculation only takes a few seconds, and then curve map pops up immediately, like this



The gray dash line indicates the position of Y=0. If the line is defined by the second way, two red circles with Y=0 will appear in the graph, they indicate the position of the two nuclei. Click

right button on the graph and then you can select what to do next, you can redefine the scale of Y-axis, export the data to line.txt in current directory, save the graph to a file, locate minimal and maximal positions and so on. Note that the process for searching stationary points and the position where Y equals to specified value is based on the data you have calculated, that means the finer the points, the more accurate X position you will get.

The data points are evenly distributed in the line, the number of points is 3000 by default, which is fine enough for most cases. The number can also be adjusted by “num1Dpoints” parameter in settings.ini. Of course, the more points the more time is needed for calculating data. Notice that for ESP calculation, the number of points is decreased to one-sixth automatically, because it is much more time-consuming than other task.

Informations needed: GTFs (except for ESP from nuclear/atomic charges and promolecular approximation version of RDG and $\text{sign}(\lambda_2) * \rho$), atom coordinates.

3.5 Outputting and plotting specific property in a plane (4)

The basic steps for using this function are listed below

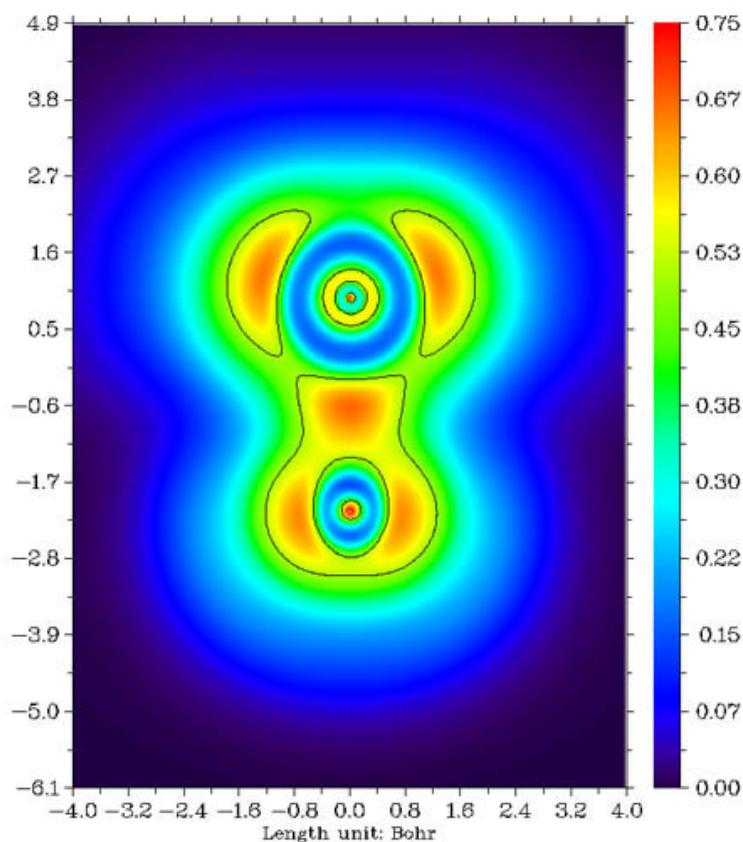
- 1 Select a real space function
- 2 Select graph type
- 3 Set the number of grid points in both dimensions
- 4 Define a plane
- 5 View the graph
- 6 Post-process (Adjust plotting parameters and replot, save graph, export data to plain text file, etc.)

Users can finish all operation only by following prompts, there are only several key points needed to be described.

3.5.1 Graph type

Currently Multiwfn supports seven graph types for showing data in a plane.

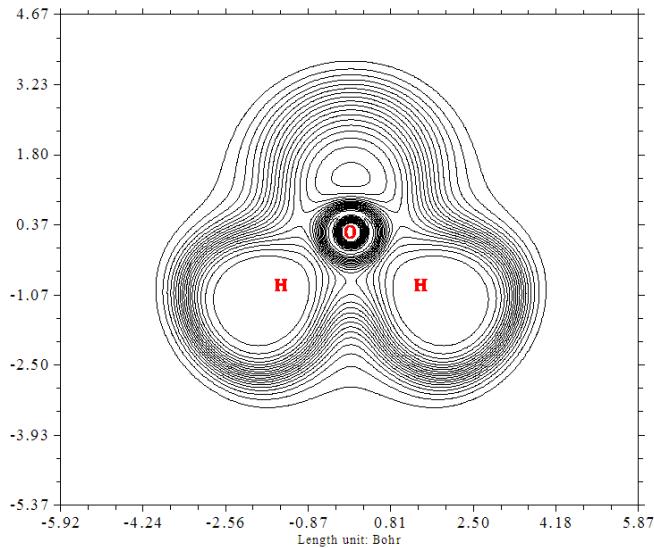
1 Color-filled map. Use color scale to map value of real space functions in every region, if the value exceeds lower limit or upper limit, then the region will be filled by black and white respectively. By selecting “Show contour lines“ in post-process step, contour lines could be plotted on the graph.



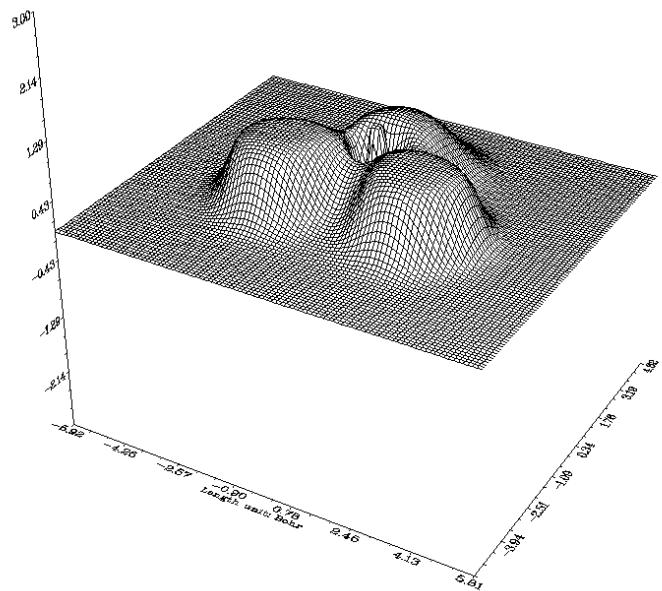
3 Functions

2 Contour line map. Solid lines indicate positive regions, dash lines indicate negative regions. The number of isovalue lines can be adjusted by user (see Section 3.5.3), you can also mark the isovales on the contour lines by using “Enable showing isovale on contour lines” in post-process step. If the vertical distance between an atom and plane is smaller than 0.3 bohr (you can modify this criteria by “disslabel” in settings.ini), the atom label will be shown on the graph.

Option 15 in post-process is used to plot a contour line corresponding to vdW surface (electron density=0.001 a.u., which is defined by R. F. W Bader). This is useful to analyze distribution of electrostatic potential on vdW surface. Such a contour line can be plotted in gradient line and vector field map too by the same option.

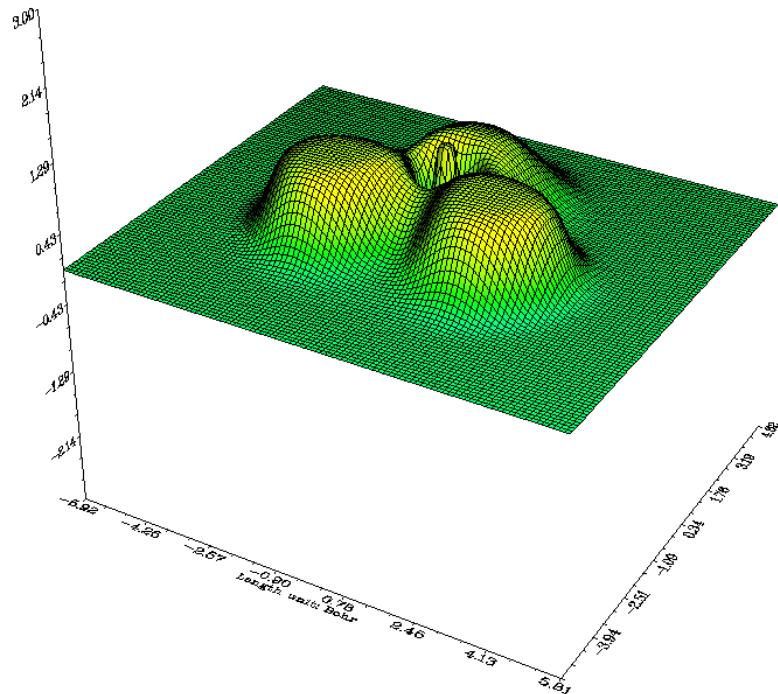


3 Relief map. Use height to represent value at every region. If values are too large they will be truncated in the graph, you can choose to scale the data with a factor to avoid truncation. The graph is shown on interactive interface, you can rotate, zoom in/out the graph.

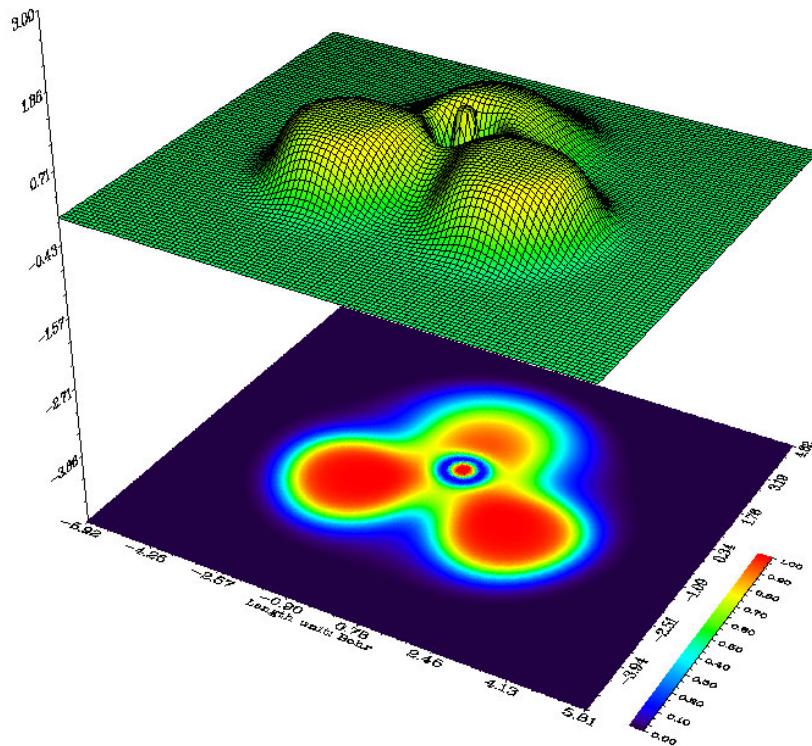


3 Functions

4 Shaded surface map. Similar to relief map, except that different value is also represented by shading different color. If you like, you can disable meshes on the surface in post-process step.

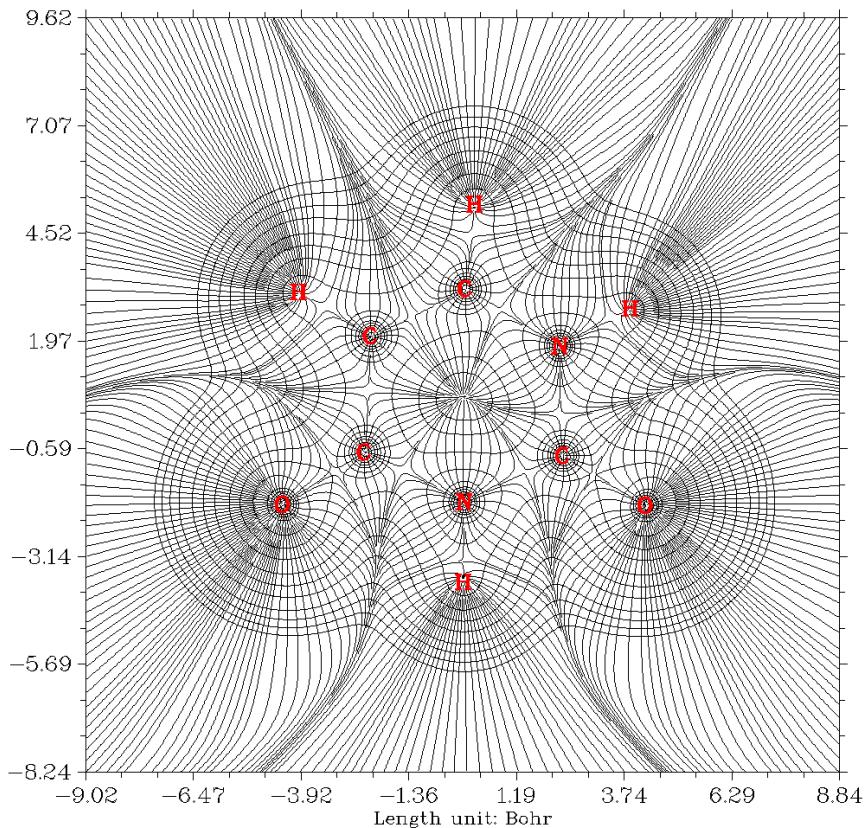


5 Shaded surface map with projection. Simply “shaded surface map”+“Color-filled map”.

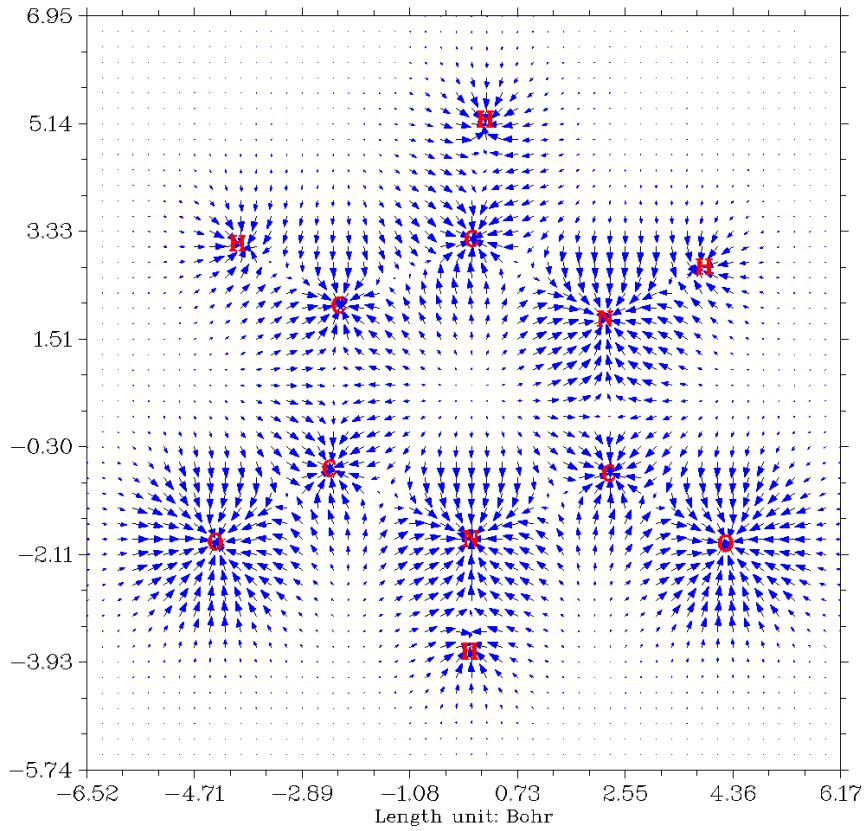


6 Gradient line map with/without contour lines (Not available for total ESP). This graph type represents gradient direction of real space function, you can determine if the contour lines also be shown on the graph. Note that since gradients of real space function are needed to be evaluated, and graphical library needs to take some time to generate gradient lines, the computational cost is evidently higher than other graph type.

By option 11 at post-process stage you can control the integration step for gradient lines, the smaller value you set the graph looks finer. By option 12 you can set interstice between gradient lines, the smaller value you set the lines become denser. By option 13 you can set the criteria for plotting new gradient line, try to play with it and you will know how the parameter affects the graph.



7 Vector field map with/without contour lines (Not available for total ESP). This graph type is very similar to last graph type, however the gradient lines are replaced by arrows, which distribute on grids evenly and represent gradient vectors at corresponding point. You can set color of arrows, or map different colors on arrows according to magnitude of function value, you can also invert the direction of arrows. The option 10 is worth mentioning, if you set upper limit for scaling to x by this option, then if the norm of a gradient vector exceeds this value, the vector will be scaled so that its norm equals to x .



3.5.2 Setting up grid, plane and plotting region

When program asking you to input the number of grid points in both dimensions, you can input such as *100,150*, which means in dimension 1 and 2 the number of grid points is 100 and 150 respectively, so total number is $100 \times 150 = 15000$, they are evenly distributed in the plotting region. For “Relief map”, “Shaded surface map” and “Shaded surface map with projection”, commonly I recommend *100,100*, if this value is exceeded, lines in the graph looks too crowd. For other graph type I recommend *200,200*, of course the picture will become more beautiful if you set the value to larger, but you have to wait more time for calculation. Bear in mind total ESP calculation is very time-consuming, you’d better use less grid points, for previewing purpose I recommend *80,80* in general.

Multiwfn provides six modes to define the plotting plane:

- 1 XY plane: User input Z value to define a XY plane uniquely.
- 2 XZ plane: User input Y value to define a XZ plane uniquely.
- 3 YZ plane: User input X value to define a YZ plane uniquely.
- 4 Define by three atoms: Input indices of three atoms to define a plane by their nuclear coordinates.
- 5 Define by three points: Input coordinates of three points to define a plane.
- 6 Input origin and transitional vector: This way is only suitable for expert, the two inputted translation vectors must be orthogonal.

Actual plotting region is a subregion of the plane you defined. Multiwfn automatically sets the plotting region to tightly enclose the whole molecule (for mode 1, 2 and 3) or cover the three nuclei / points you inputted (for mode 4 and 5), finally the plotting region is extended by a small distance to avoid truncating the interesting region. the extended distance is 4.5 bohr by default, if you find the region you interested in is still be truncated, just enlarge the value by option “0 Set size extension for plane type 1~5”, you can also directly modify the default value, which is controlled by “Aug2D” parameter in settings.ini.

3.5.3 Setting contour lines

For graph type 1, 2, 6 and 7, there is an option “Change contour line setting” in the post-process step. In this interface, current isovalues are listed (by default there are 62 isovalues) and you can modify them by using the subfunctions in this interface:

Option 1: Save current setting and return to upper menu. Then if you select “Show the graph again”, the graph with new isovalue setting will appears.

Option 2: Input a new value to replace old isovalue of a contour line.

Option 3: Add a new contour line and input the isovalue for it. Notice that total number of contour lines should not exceed 100.

Option 4: Delete a contour line. The indices of the contour lines behind it will move forward to fill the gap.

Option 5: Delete contour lines whose index is in a certain range.

Option 6: Export current isovalue setting to a plain text file, you can use this function to save multiple sets of your favourite isovalue settings for different systems and real space functions.

Option 7: Load isovalue setting from external file, the format should be identical to the file outputted by subfunction 6.

Option 8: Clean current setting and generate isovalues according to arithmetic sequence, user need to input initial value, step size and total number (should less than 100). For plotting ELF/LO, I suggest user input 0,0.05,21 to generate isovalues in the range 0.0~1.0 with step size 0.05.

Option 9: Likewise function 8, but according to geometric series.

Option 10: Some contour lines can be bolded with this function, by default no line is bolded. To bold some lines, select this function and input how many lines you want to bold, then input indices of them in turn. If there are some lines already been bolded, selecting this function will unbold all of them.

Option 11: Set color for positive contour lines, you need to input a color index.

Option 12: Set line style for positive contour lines, you need to input two integer number, the first one denotes the length of line segment, the second one denotes the length of interstice between line segment. For example 10,15 means positive contour lines are composed of line segments with length of 10 and spaces with length of 15 alternatively.

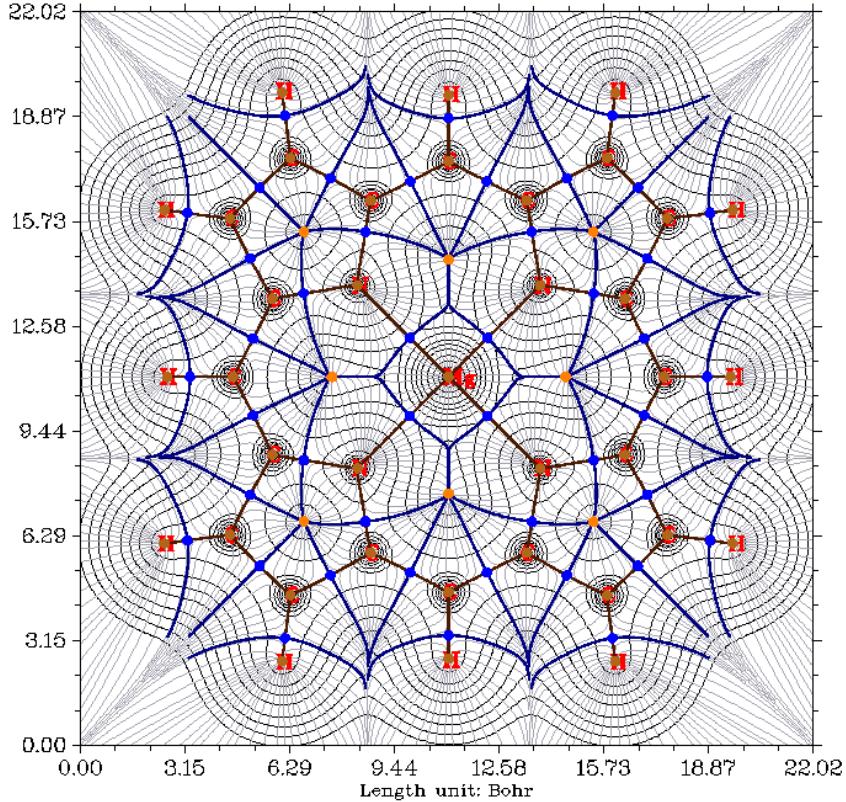
Option 13, 14: Likewise option 11 and 12, but for negative part.

3.5.4 Plot critical points, paths and interbasin paths on plane graph

CPs and paths can be plotted on plane graph type 2, 6 and 7. In order to do that, before drawing a plane graph, you need to go into topology analysis module (main function 2), search CPs and generate paths. After that, return to main menu and draw plane graph as usual, you will

3 Functions

find that the CPs and paths have appeared on the graph automatically, such as below (without deep blue lines). Brown, blue, and orange circles denote (3,-3), (3,-1) and (3,+1) critical points respectively, deep brown lines depict bond paths. Notice that if the distance between a CP and the plane exceeds "disshowlabel" in settings.ini, the CP will not be shown.



In the option “Set marks of critical points and paths” at post-process stage, you can set which types of CPs are permitted to be shown, size of markers, thickness and color of path lines.

Interbasin paths (deep blue lines in above graph) are derived from (3,-1) CPs, these paths dissect the whole space into individual basins, within each of which a (3,-3) CP can be found. In order to draw interbasin paths, you should confirm first that at least one (3,-1) CP has been found in topology analysis module and it is close enough to current plane (less than "disshowlabel" in settings.ini), then you can find a option "Generate and show interbasin paths" in post-process stage, choose it, wait until the generation of interbasin paths is completed, then show the plane graph again, you will find these interbasin paths have already presented.

If you expect the interbasin paths to be shorter or longer, choose the option "Set stepsize and maximal iteration for interbasin path generation" in post-process stage before generating interbasin paths, you will be prompted to input two values, the length of interbasin paths equals to the product of the two value. Note that if distance between a point in the path and the given plane exceeds "disshowlabel" in settings.ini, corresponding segment of the path will not be shown on the plane graph.

Informations needed: GTFs (except for ESP from nuclear/atomic charges and promolecular approximation version of RDG / sign(lambda2)*rho function), atom coordinates.

3.6 Outputting and plotting specific property within a spatial region (5)

The main purpose of this function is generating Gaussian-type grid file (.cub) for specific property, the .cub file is supported by a lot of chemistry visualization softwares, such as VMD, GaussView, Chemcraft, and Molekel. The isosurface of generated grid data can be previewed directly in Multiwfn too.

When using this function, select a real space function first, and then set grid point. Multiwfn provides seven modes for setting up grid point:

1 Low quality grid, about 125000 points in total (corresponding to 50*50*50 grid assuming the spatial region is cubic), this mode is recommended for previewing.

2 Medium quality grid, about 512000 points in total (corresponding to 80*80*80 grid assuming the spatial region is cubic). For small molecular, this quality is enough for most analysis.

3 High quality grid, about 1728000 points in total (corresponding to 120*120*120 grid assuming the spatial region is cubic).

For mode 1, 2 and 3, the actual number of points in each direction is automatically determined by Multiwfn so that spacings in each direction are nearly equal. The spatial scope of grid data is also determined automatically so that entire system can be completely covered. By viewing isosurface, if you found the boundary region is truncated too much and valuable informations are missing, you can set extension distance larger by using option “-10 Set size extension of grid range”, or directly set default value by modifying “Aug3D” in settings.ini.

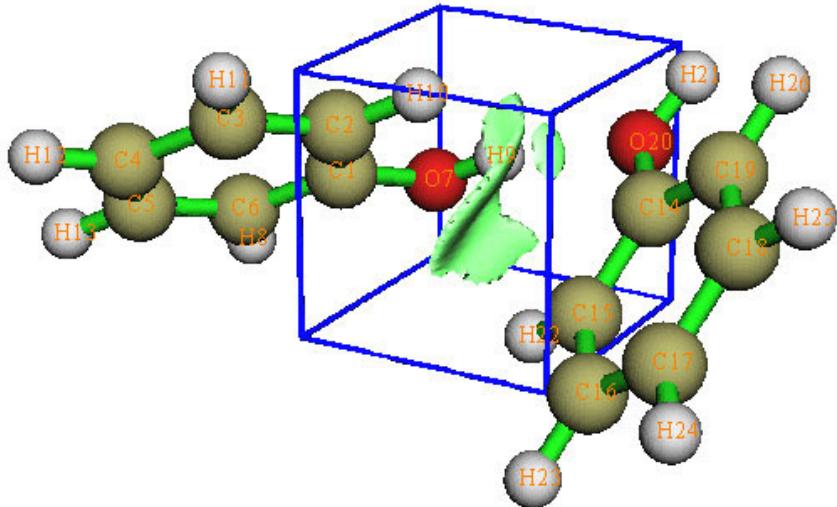
The so-called “Low quality”, “Medium quality” and “High quality” are **only relative to small system**, if spatial scope is large, then density of grid points of mode 3 is not high and should be called “Medium quality”, mode 2 should be call “Low quality” at this time. While for very small system, the mode 1 is enough for general purposes.

4 Specify the number of points in X, Y and Z directions by yourself, the spatial scope is determined automatically as above.

5 Specify all details of grid setting by user, including the number of points in X, Y and Z directions, initial point, translation vectors. This mode is useful for reproducing existed cube files.

6 Specify the center coordinate, number of points and extension distance in X, Y and Z directions. For example, the center coordinate you inputted is 2.3,1.0,5.5, the extension distance in X, Y and Z is k,k,m , then the coordinates of two most distant endpoints are (2.3- k ,1.0- k ,5.5- m) and (2.3+ k ,1.0+ k ,5.5+ m). This mode is useful for analyzing local properties.

7 Likewise function 6, but input indices of two atoms instead of inputting center coordinate, the midpoint between the two nuclei will be set as center. This mode is very useful for weak interaction analysis by RDG function, for examples, we want to study the weak interaction region between the dimer shown below, and we found C1 and C14 may enclose this region, so we input 1,14. see Section 4.100.1 for example. If the two atom indices are identical, then the nuclear coordinate will be set as center.



Notice that the more grid points you set, the finer isosurface graph you will gain, however the more time is needed for calculating data and generating isosurface graph, and the size of exported cube file will be bigger. Except for total ESP function, the increase of calculation time with the number of points is nearly linear.

After setting grid points, the program start to calculate grid data, once the calculation is finished, the minimum/maximum and corresponding coordinate, the sum of all/positive/negative data multiplied by differential element are printed on screen immediately. If what you calculated is electron density, the molecular dipole moment evaluated from grid data is printed too.

If the option “-1 Show the graph of isosurface” is selected, a GUI window will pop up, which shows the isosurface of grid data, all widgets are self-explained, you can change isosurface value by both dragging slide bar (upper and lower limit are -5 and +5 respectively) or inputting precise value in text box (Then press “Enter” button in you keyboard); if “Show data range” is selected, the spatial scope will be marked by a blue frame as the one in above picture (if spatial scope exceeds the range of coordinate axis, the frame will no be displayed). The isosurfaces with the same and reverse sign of current iso value are in green and blue respectively.

Function 1 can export isosurface graph to graphic file in current directory. Function 2 can export grid data to .cub file in current directory. By selecting function 3 the grid data will be exported to plain text file “output.txt” in current directory. Using function 4 you can set iso value without entering GUI window and dragging slide bar, it is useful for batch process and in command-line environment.

Informations needed: GTFs (except for ESP from nuclear/atomic charges and promolecular approximation version of RDG / sign(lambda2)*rho function), atom coordinates.

3.7 Custom operation, promolecular and deformation properties (subfunction 0, -1, -2 in main function 3, 4, 5)

3.7.1 Custom operation for multiple wavefunctions (0)

In function 3, 4 and 5, there is an subfunction allow you to set custom operation for multiple wavefunctions. Supported operators include + (add), - (minus), x (multiply), / (divide), there is no upper limit of the number of wavefunctions involved in custom operation. If the first loaded wavefunction is named “a.wfn”, in setting step of custom operation you input 2 (means there are two wavefunctions will be put into “custom operation list” and then will be operated with a.wfn in turn), then you input $-b.wfn$ and $x,c.wfn$, the property you get will be $[(\text{property of } a.wfn) - (\text{property of } b.wfn)] * (\text{property of } c.wfn)$.

Let see a practical example, assuming you want to view Fukui function of phenol, the first way is to calculate and export cube file for neutral state and +1 state separately, and then minus the former one by latter one by using program such as gsgrid (<http://gsgrid.codeplex.com>) and cubman, after that load the new cube file to visualization software, evidently this way is complex. By using “custom operation” in Multiwfn, this aim is able to be achieved more easily, all the steps are performed internally in Multiwfn. First load neutral state wavefunction in Multiwfn and enter main function 5, select subfunction 0, input 1, then input $-phenol_+1.wfn$, where “phenol_+1.wfn” is the wavefunction file of +1 state, then select electron density function and set grid points as usual, finally the isosurface graph you will see is the Fukui function of phenol.

Sometimes the molecular structure of first loaded file and the other files (that is those in the custom operation list) are not identical, the grid points you set is for the first loaded molecule, the automatical adjustment of spatial scope (if concerned) is also relative to the first loaded molecule, all the other files share this grid setting.

Avoid combining use of custom operation with main function -4, -3 and 6, otherwise you may get unwanted result.

3.7.2 Promolecular and deformation properties (-1, -2)

If you selected subfunction -1 in main function 3 or 4 or 5 before choosing a property (real space function), what you will get finally is promolecular property. Promolecular property is the superposition property of atoms in free-state

$$P^{pro}(\mathbf{r}) = \sum_A P_A^{free}(\mathbf{r} - \mathbf{R}_A)$$

If the property you chose is electron density, then the promolecular property is generally referred as promolecular density

$$\rho^{pro}(\mathbf{r}) = \sum_A \rho_A^{free}(\mathbf{r} - \mathbf{R}_A)$$

This is a pseudo-density corresponding to when molecule has formed but the density hasn't relaxed.

Deformation property is the difference between actual property and promolecular property of

a molecule in the same geometry

$$P^{def}(\mathbf{r}) = P^{mol}(\mathbf{r}) - P^{pro}(\mathbf{r})$$

If the property is chosen as electron density, then the deformation property is generally called deformation density, which is very useful for analyzing charge transfer and bonding nature.

3.7.3 Generation of atom wavefunctions

After you select subfunction -1, Multiwfn check whether all .wfn files corresponding to the elements in current system have presented in “atomwfn” subdirectory of current directory, if not, Multiwfn automatically invoke Gaussian to generate them. If the path of Gaussian executable file (“gaupath” parameter in settings.ini) hasn’t been defined or incorrect, Multiwfn will ask you to provide the path of Gaussian executable file. The basis-set for generating atom wavefunction can be specified randomly by user, however it is suggested to be in agreement with the basis-set used for molecular wavefunction, especially for yielding deformation property.

The newly generated atom wavefunctions or those carried from “atomwfn” directory (see below) are stored in “wfntmp” subdirectory in current directory. They will be sphericalized and translated to the position of each atom in present system according to element type, then corresponding atom index will be added to filename as suffix. These are the files directly be used to calculate promolecular and deformation properties.

If you want to avoid the step of generating atom wavefunctions every time, you can move the atom .wfn files that without number suffix (such as “C.wfn”) from “wfntmp” directory to “atomwfn” directory (If “atomwfn” directory is non-existing, build it by yourself), next time if Multiwfn finds all the needed atom .wfn files are already existing within “atomwfn” folder then Multiwfn will not annoy you again. Reversely, If you want to regenerate atom .wfn files with new basis-set, ensure “atomwfn” folder is non-existing in current directory, or you can change its name to avoid Multiwfn borrowing files in it.

“atomwfn” subdirectory in “examples” directory in Multiwfn compressed package contains all first four-rows atom wavefunction files generated under 6-31G*, if you want to use them, copy the directory to where “Multiwfn.exe” is posited in. Besides, there is a quick way to generate all first four-rows atom wavefunction files under specific basis-set, the file “example\genatmwfnpdb” contains all first four-rows atoms, load it into Multiwfn and generate any promolecular property, when the calculation of all atom wavefunctions is finished, copy the .wfn files that without suffix in “wfntmp” directory to “atomwfn” directory.

If your system involves elements heavier than Kr, Multiwfn is unable to generate atom wavefunctions by invoking Gaussian automatically, in that case you must calculate and sphericalize atom wavefunctions manually, then put the .wfn files to “atomwfn” directory.

The default theoretical method for generating wavefunction for main group elements in first four-rows (index from 1 to 20 and 31 to 36) is ROHF, for the transition metals in the fourth row, UB3LYP is used. In general, the promolecular and deformation properties are not sensible to theoretical method, if you want to specify theoretical method by yourself, you can input theoretical method and basis-set at the same time with slash as separator, for example *BLYP/6-311G**. Don’t add “RO” or “U” prefix, they will be added automatically for main group elements and transition metals in first four-rows. If exceptions occur during generation of atom

wavefunctions, please check Gaussian input and output files in “wfntmp” directory carefully.

Notice that the maximum character length of the path of .wfn file allowed by Gaussian is only 60! The path will be truncated if the length exceeds this threshold, and will result in error because Multiwfn can't find atom .wfn files. So don't put Multiwfn in the directory with too long path!

3.7.4 Sphericalization of atom wavefunction

The main purpose of Multiwfn supporting promolecular and deformation property is for generating promolecular and deformation density, however, electron density of most elements in free and ground state is not spherical, which leads to orientation dependence problem. To tackle it, atom electron density must be sphericalized, however there is no unique way to do this. In Multiwfn, atom electron density is sphericalized by modifying atom wavefunction artificially, here I describe the detail. If you want to skip the sphericalization step, simply set the “ispheratm” in settings.ini to zero.

For elements in IV A group, Multiwfn uses sp^3 configuration to replace s^2p^2 ground state by default. This treatment is reasonable, since in most molecules these atoms are in sp^3 hybridization. For VI A, VII A and Fe, Co, Ni, Multiwfn equalizes the occupation number of orbitals within the same shell, for example, oxygen has two singly occupied 2p orbitals and one doubly occupied 2p orbitals in ground state, the number of electrons in this shell is 4, so Multiwfn sets the occupation number of all the three orbitals to 4/3. This method works because the shape of occupied orbitals are always nearly identical, regardless of the original occupation number is one or two. However, the difference between virtual orbital and occupied orbital is remarkable, thus Multiwfn uses another method to sphericalize electron density of elements in III A group, Sc, Ti and V. In this method, the singly occupied orbitals are duplicated and rotated. Taking boron as example, assume that the singly occupied 2p orbital is directing along Z-axis, Multiwfn replicates this orbital twice and turn them toward X-axis and Y-axis respectively, finally the occupation numbers of the three orbitals are set to 1/3. (Users needn't to check the orientation of singly occupied orbital by themselves). If you hope that Multiwfn sphericalizes atoms in IV A group in such manner instead of using sp^3 configuration, set “SpherIVgroup” in settings.ini to 1.

Notice that the sphericalization methods used in Multiwfn are closely related to wavefunction type, the methods fail if unrestricted wavefunction is used for main group elements or restricted open-shell wavefunction is used for transition metals. The methods also fail if hartree-fock method is used for transition metals, because orbital order is different from most DFT cases (the HF's order is wrong, 4s is higher than 3d).

Informations needed: GTFs, atom coordinates.

3.8 Checking & Modifying wavefunction (6)

This function provides a lot of subfunctions for checking and modifying current loaded wavefunction, all subfunctions take effect immediately, so after modification you can print related informations to check if your operations are correct. Once you finished all modifications of wavefunction, you can save current wavefunction to “new.wfn” file or return to main menu, all

following tasks which make use of GTF informations will be affected. The title in this interface shows the number of GTFs, orbitals, atoms and alpha/beta electrons of current wavefunction.

Actually, the function -1, -2, -3 and -4 in main menu (see introduction above) should be put into this category, the reason why I still deploy them in main menu is only because they are invoked more frequently.

Below is the explanation of each subfunction in this category.

Subfunction -1: Return to main menu.

Subfunction 0: Save current modified wavefunction to “new.wfn” file in current directory.

Multiwfn can be used as a fch→wfn format converter through this function. Notice that the orbitals with zero occupation number and the atoms without any GTF posited on them will be automatically discarded during saving. Although GTFs with angular moment of g is not formally supported by original wfn format, g -type GTFs are still outputted by Multiwfn if present. 21~35 in "TYPE ASSIGNMENT" correspond to ZZZZ, YZZZ, YYZZ, YYYZ, YYYY, XZZZ, XYZZ, XYYZ, XYYY, XXZZ, XXYZ, XXYY, XXXZ, XXXY, XXXX respectively.

Subfunction 1: Print information of all GTFs, including the centers they are posited on, GTF types and exponents.

Subfunction 2: Print information of all basis functions, including the shells/centers they attributed to, types, and corresponding GTF index ranges.

Subfunction 3: Print basic information of all orbitals, including energies, occupation numbers, orbital types (Alpha, beta or alpha+beta).

Subfunction 4: Print detail information of an orbital, including the expansion coefficients with respect to GTFs (along with GTF informations), note that the coefficients include GTF normalization constants.

Subfunction 5: Print coefficient matrix in basis function (not the coefficients with respect to GTFs), only available when the input file contains basis function informations.

Subfunction 6: Print one-particle density matrix in basis functions, only available when the input file contains basis function informations.

Subfunction 7: Print overlap matrix in basis functions, the eigenvalues are printed together for checking linear dependence. Only available when the input file contains basis function informations.

Subfunction 11 to 15: Swap the centers or types or exponents or orbital expansion coefficients of two GTFs, or swap all properties of two GTFs (identical to swap the record order of them, thus doesn't affect any analysis result)

Subfunction 21 to 24: Set the center, type, exponent and expansion coefficient of a specific GTF in an orbital respectively.

Subfunction 25: Set the expansion coefficients of some GTFs that satisfied certain conditions in some orbitals. The conditions you can set for GTFs include: Index range of GTFs, index range of atoms, GTF types (you can input such as YZ, XXZ). The selected GTFs are intersection of these conditions. When program is asking you for inputting range, you can input such as 3,8 to select those from 3 to 8, input 6,6 to select only 6, especially, input 0,0 to select all.

Subfunction 26: Set occupation number of some orbitals. **This function is very useful for shielding the contributions from certain orbitals to real space functions, namely setting their occupation numbers to zero before calculating real space functions.**

Subfunction 27: Set orbital type of some orbitals, then wavefunction type will be automatically updated, please check the prompts shown on the screen.

Subfunction 31: Translate the whole wavefunction and all atom coordinates of current system by inputting translation vectors and their units.

Subfunction 32: Translate and duplicate the whole wavefunction and all atom coordinates of current system, users need to input translation vectors, their units and how many times the system will be translated and duplicated according to the translation vectors. This function is very useful for extending the primitive cell wavefunction outputted by Gaussian PBC function to multiple cells wavefunction, of course you can calculate multiple cells wavefunction directly by Gaussian, but much more computational time will be consumed and may lead to some accumulated errors.

Subfunction 34: Delete all inner orbitals of all atoms in present system, only valence orbitals will be reserved. Notice that for ECP wavefunction, this function do nothing.

Informations needed: Basis function (only for subfunction 2, 5, 6, 7), GTFs, atom coordinates.

3.9 Population analysis (7)

3.9.1 Hirshfeld population (1)

Hirshfeld is a very popular population method based on deformation density partition, Hirshfeld charge is defined as (*Theor. Chim. Acta (Berl.)* 44, 129)

$$q_A = - \int w_A^{Hirsh}(\mathbf{r}) \rho_{def}(\mathbf{r}) d\mathbf{r}$$

Where

$$\begin{aligned}\rho^{def}(\mathbf{r}) &= \rho(\mathbf{r}) - \rho^{pro}(\mathbf{r}) \\ \rho^{pro}(\mathbf{r}) &= \sum_A \rho_A^{free}(\mathbf{r} - \mathbf{R}_A) \\ w_A^{Hirsh}(\mathbf{r}) &= \frac{\rho_A^{free}(\mathbf{r} - \mathbf{R}_A)}{\rho^{pro}(\mathbf{r})}\end{aligned}$$

The advantages of Hirshfeld population are:

1. Result is qualitatively consistent with general chemical concepts such as electronegativity rule.
2. The weighting function w for space partition has clear physical meaning.
3. Unlike the methods based on integrating electron density such as AIM charge, what the Hirshfeld charge reflects is the amount of transferred electron density during molecule formation, the density not transferred is not involved.
4. Insensitive to the quality of wavefunction.
5. Although calculating Hirshfeld charge needs integration in real space, due to the smooth integrand, sophisticated density functional theory (DFT) grid-based integration schemes can be directly used, so Hirshfeld population is high-efficient.
6. The wide application field. Deformation density data can also be obtained by X-ray crystallography experiments. Moreover, the applicability of Hirshfeld population is not

3 Functions

constrained by the type of wavefunction, the method can be directly applied to solid system, where the wavefunction generally be described by plane-wave functions.

The disadvantages of Hirshfeld population are the charge is always too small and the poor reproducibility of observable quantities, such as molecular dipole moment and ESP, the reason is Hirshfeld population completely ignores atomic dipole moments.

When calculation is finished, along with Hirshfeld charges, following contents are printed

1 Dipole of atom: Atomic dipole moment, the simplest representation of density anisotropy around corresponding atom.

2 Summing up all charges: The value of summing up all Hirshfeld charges, if the value is very close to integer, means the quadrature is accurate; if not, means the outputted Hirshfeld charges are unreliable, you need to increase the density of integration points by setting “radpot” and “sphpot” to larger value and calculate again. For balancing computational time and accuracy, the default value of “radpot” and “sphpot” is 75 and 434 respectively, you can set them to 100 and 590 respectively for more accurate results.

3 Total dipole from atomic charges: Molecular dipole moment evaluated from Hirshfeld charges, following the components in X, Y and Z directions.

4 Total atomic dipole: The sum of all atomic dipole moments, following the components in X, Y and Z directions.

5 Total corrected dipole: The sum of “Total dipole from atomic charges” and “Total atomic dipole” following the components in X, Y and Z directions. This value should be very close to the molecular dipole moment directly evaluated from molecular wavefunction.

Finally, you can choose to output Hirshfeld charges to [name of loaded file].chg file in current directory, see Section 2.5 for detail about chg format.

Notice that this function requires wavefunctions of atom in free-state for generating $\{\rho^{free}\}$, see Section 3.7.3.

3.9.2 Voronoi deformation density (VDD) population (2)

The only difference between VDD and Hirshfeld population is the weighting function w . In VDD population, the Voronoi cell-like partition is used, see *J. Comp. Chem.*, 25, 189.

The results of VDD population are similar to Hirshfeld population in common, because the magnitude of deformation density is always small, so there is no significant change in charges when different weighting functions are used. The outputted terms are identical to Hirshfeld population. Personally, I suggest you use Hirshfeld population instead of VDD, since in present Multiwfn version the numerical integral precision for VDD is not as high as for Hirshfeld population.

3.9.3 Mulliken atom & basis function population analysis (5)

Mulliken population (MPA) is the oldest population method based on orbital wavefunction, supported by almost all quantum chemistry packages.

Orthonormality condition of spin orbital wavefunction entails $1 = \int |\phi(\mathbf{r})|^2 d\mathbf{r}$, if we assume

3 Functions

that the orbital is real type and insert the linear combination equation $\varphi_i(\mathbf{r}) = \sum_a C_{a,i} \chi_a(\mathbf{r})$ into it, we get

$$1 = \int \left(\sum_a C_{a,i} \chi_a(\mathbf{r}) \right)^2 d\mathbf{r} = \sum_a C_{a,i}^2 + \sum_a \sum_{b \neq a} C_{a,i} C_{b,i} S_{a,b} = \sum_a C_{a,i}^2 + 2 \sum_a \sum_{b > a} C_{a,i} C_{b,i} S_{a,b}$$

Where $S_{a,b} = \int \chi_a(\mathbf{r}) \chi_b(\mathbf{r}) d\mathbf{r}$, the normality of basis functions are used in derivation.

The first term is “local term”, denotes the net population of each basis function in orbital i , the second term is “cross term”, denotes the shared electrons between basis function pairs in orbital i . Certainly the local terms should be completely attributed to corresponding basis functions, however for cross terms the partition method is not unique. Mulliken defined the population of basis function a in spin orbital i as

$$\Theta_{i,a} = C_{a,i}^2 + \sum_{b \neq a} C_{a,i} C_{b,i} S_{a,b}$$

That is each cross term $2C_{a,i} C_{b,i} S_{a,b}$ is equally partitioned to corresponding two basis functions.

The population number of atom A is simply the sum of population numbers of all basis functions attributed to atom A in all orbitals. Mulliken atomic charge is defined as

$$q_A = Z_A - \sum_i n_{i,A} = Z_A - \sum_i \eta_i \sum_{a \in A} \Theta_{i,a}$$

Where η is orbital occupation number, $n_{i,A}$ is the contribution from orbital i .

In **subfunction 1** (“Output Mulliken charges and decompose them to MO's contribution”), not only atomic charges but also contributions of basis functions and shells are outputted, the latter are the sum of contributions from basis functions within corresponding shells. You can choose to decompose atom population number to orbitals' contributions.

In **subfunction 2** (“Output gross atomic population matrix and decompose it”), the gross atom population matrix is printed, from which you can get local terms of each atom (diagonal element) and cross terms between each atom pair (non-diagonal element multiplies 2). The matrix is defined as

$$\Omega_{A,B} = \sum_i \Omega_{A,B}^i = \sum_i \eta_i \sum_{a \in A} \sum_{b \in B} C_{a,i} C_{b,i} S_{a,b}$$

Notice that the last row of outputted matrix is the sum of corresponding column elements, that is the total population number of corresponding atom. You can also choose to decompose it to orbital's contribution ($\Omega_{A,B}^i = \eta_i \sum_{a \in A} \sum_{b \in B} C_{a,i} C_{b,i} S_{a,b}$), owing to the very large amount of

informations, results will be outputted to “groatmdcp.txt” in current directory rather than printing on screen.

By the way, the quantity $2 * \Omega_{A,B}^i$ is just the Mulliken bond order between atom A and B

contributed from orbital i , as we will see in Section 3.11.4.

In **subfunction 3** (“Output gross basis function population matrix & decompose it”), the gross basis function population matrix can be outputted for analyzing detail informations further, the matrix element is defined as

$$\Gamma_{a,b} = \sum_i \Gamma_{a,b}^i = \sum_i \eta_i C_{a,i} C_{b,i} S_{a,b}$$

Likewise in subfunction 2, the last row of outputted matrix is total population number of corresponding basis function. You can also select to output orbital's contribution ($\Gamma_{a,b}^i = \eta_i C_{a,i} C_{b,i} S_{a,b}$) to “grobasdcp.txt” in current directory.

MPA is deprecated for practical application due to the serious shortcomings: 1. Poor reproducibility of observable properties 2. The “equal partition” of cross term have no strict physical meaning 3. Very high basis-set dependence 4. Occasionally meaningless result occurs (population number is negative).

3.9.4 Löwdin population analysis (6)

The only difference between Löwdin population and MPA is if Löwdin orthogonalization is performed first. In Löwdin orthogonalization, the linear transformation matrix is

$$X = U s^{0.5} U^H$$

Where U is eigenvector matrix of overlap matrix, matrix s is $\text{diag}(\lambda_1, \lambda_2, \dots)$, here $\{\lambda\}$ is eigenvalue set of overlap matrix. After Löwdin orthogonalization, the overlap matrix becomes identity matrix, and new coefficient matrix is $X^{-1}C$.

Ostensibly, Löwdin population avoids the partition for cross terms since they have become zero, actually, the partition step is no more than hidden in the orthogonalization. Since Löwdin orthogonalization still has no strong physical meaning, it cannot make conclusion that Löwdin population is better than MPA in principle. In view of practical results, one also found Löwdin charges have no any evident advantages relative to Mulliken charges, though some people argued that Löwdin charges have better basis-set stability and reproducibility of molecular dipole moment. Besides, Mayer et. al. found Löwdin population has rotation dependence to some extent when cartesian type gauss basis functions are used, however the dependency can be safely ignored in generally, for detail please see *Chem. Phys. Lett.*, 393, 209 and *Int. J. Quantum. Chem.*, 106, 2065.

3.9.5 Modified Mulliken population defined by Ros & Schuit (SCPA) (7)

Some people had proposed several different partition methods of cross term to improve MPA, they are generally called modified MPA (MMPA). In the method proposed by Ros and Schuit (*Theo. Chim. Acta*, 4, 1), the composition of basis function a in orbital i is defined as

$$\Theta_{i,a} = \frac{C_{a,i}^2}{\sum_b C_{b,i}^2}$$

3 Functions

only the square of coefficients are presented in the formula, so this method is also called C-squared Population Analysis (SCPA). By inserting the identity

$$1 = \sum_a C_{a,i}^2 + \sum_a \sum_{b \neq a} C_{a,i} C_{b,i} S_{a,b}$$

into the right most of above formula, Θ can be rewritten as

$$\Theta_{i,a} = C_{a,i}^2 + \frac{C_{a,i}^2}{\sum_b C_{b,i}^2} \sum_a \sum_{b \neq a} C_{a,i} C_{b,i} S_{a,b}$$

It is clear that when calculating composition of basis function a in orbital i , what is partitioned is not the cross terms between atom a and other atoms, but the *total* cross term of all atom pairs. The SCPA atomic charges can be calculated in the same manner as MPA charges by using the newly defined Θ . Relative to MPA, the advantage of SCPA is that negative value of population number is never occurs.

3.9.6 Modified Mulliken population defined by Stout & Politzer (8)

Stout and Politzer defined the Θ as (*Theo. Chim. Acta*, 12, 379)

$$\Theta_{i,a} = C_{a,i}^2 + \sum_{b \neq a} \frac{C_{a,i}^2}{C_{a,i}^2 + C_{b,i}^2} 2C_{a,i} C_{b,i} S_{a,b}$$

That is cross terms are partitioned according to the ratio of the squares of corresponding coefficients. Ostensibly, this definition has more consideration on the unbalanced nature of cross term, however in practical applications the results are even worse than MPA, therefore this method is rarely used now. Besides, Grabenstetter and Whitehead had pointed out that this MMPA definition has unitary transformation dependence, so if the molecule is rotated the results changed (the dependence is remarkable and cannot be ignored).

3.9.7 Modified Mulliken population defined by Bickelhaupt (9)

The total population number of a basis function defined in *Organometallics*, 15, 2923 is

$$\Lambda_a = \sum_i \eta_i C_{a,i}^2 + \sum_{b \neq a} w_{a,b} \sum_i 2\eta_i C_{a,i} C_{b,i} S_{a,b}$$

Where the weight of basis function a for partitioning the total cross term between a and b in all orbitals is

$$w_{a,b} = \frac{\sum_k \eta_k C_{a,k}^2}{\sum_i \eta_i C_{a,i}^2 + \sum_j \eta_j C_{b,j}^2}$$

Essentially, it is equivalent to define Θ as

$$\Theta_{i,a} = C_{a,i}^2 + \sum_{b \neq a} w_{a,b} 2C_{a,i} C_{b,i} S_{a,b}$$

This method is similar to the MMPA defined by Stout and Politzer, the difference is in the latter the weight $w_{a,b}$ is only related to local terms of basis functions a and b in *current* orbital, while in present method the weight is related to *total* local terms of basis functions a and b .

3.9.8 Becke population (10)

In the paper *J. Chem. Phys.*, 88, 2547, Becke proposed a weighting function for converting whole space integral to multiple single-center spherical integrals, although the weighting function is not intended for population analysis, Multiwfn still makes an attempt to use this weighting function to obtain atomic charges. The Becke charge can be defined as

$$q_A = Z_A - \int w_A(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}$$

For details about Becke weighting function please see original paper. I found the resultant charges are not always good, the weighting function is needed to be adjusted further.

3.9.9 Atomic dipole moment corrected Hirshfeld population (ADCH, 11)

The main reason why Hirshfeld charges are too small and have poor reproducibility of observable quantity is that atomic dipole moments are completely neglected. In ADCH (*J Theor Comput Chem*, 11, 163), Hirshfeld charges are corrected by expanding atomic dipole moments to correction charges placed at neighbour atoms. ADCH atomic charges are very reasonable in chemical sense, molecular dipole moment is exactly reproduced, the reproducibility of ESP is close to the atomic charges from fitting ESP. Compared with another method that corrects Hirshfeld charges, namely Hirshfeld-I, the computational cost of ADCH correction is almost zero! Owing to its many advantages, ADCH is a highly recommended atomic charge model. For an extensive comparison of atomic charge models, see *Acta Phys -Chim Sin*, 28, 1 (in Chinese, <http://www.whxb.pku.edu.cn/EN/abstract/abstract27818.shtml>)

Before doing ADCH correction, Hirshfeld charge will be calculated first. In the summary field, “corrected” and “before” correspond to ADCH charge and Hirshfeld charge respectively. At final stage, the “Error” means the difference between molecular dipole moment produced by ADCH charges and the one produced by ab initio, “Error” is always equals to or very close to zero, because ADCH charge preserves molecular dipole moment.

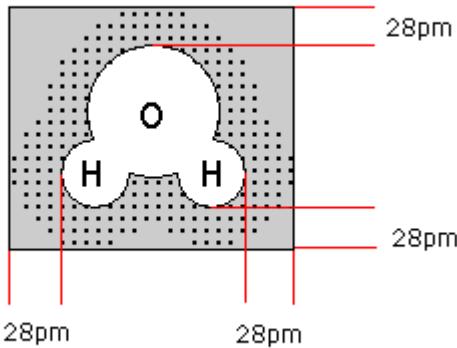
If you would like to know the detail of charge transfer in the ADCH correction process, you can set “ADCHtransfer” in settings.ini to 1.

3.9.10 Charges from electrostatic potentials using a grid based method (CHELPG, 12)

Theory

CHELPG (*J. Comp. Chem.*, 11, 361) is one of the most widely used electrostatic potentials (ESP) fitting charge model. Compared to CHELP and Merz-Kollman methods, CHELPG charges have the best rotational invariance, mostly due to the fitting points are distributed in cubic grid manner.

In CHELPG model, a box is defined first to enclose the whole molecule, extension distance in each side is 28pm, see the picture below



Fitting points are evenly scatter in the box, the default spacing is 0.3 angstrom. For any fitting point, if the distance between the point and any nucleus is smaller than vdw radius of corresponding atom, or the distances between the point and all nuclei are larger than 28pm, then the fitting point will be discarded. The black dots shown above are final fitting points.

Like other ESP fitting methods, in CHELPG, the deviation function shown below is minimized to make the ESP calculated by atomic charges (V_q) close to the ESP calculated by quantum chemistry (V) as well as possible.

$$F(q_1, q_2 \dots q_N) = \sum_i (V_q(\mathbf{r}_i) - V(\mathbf{r}_i))^2$$

Where \mathbf{r}_i is the coordinates of fitting point i . q are fitted point charges, the position are called fitting center. Notice that q doesn't necessarily mean atomic charge, the fitting center can be defined beyond nucleus.

Usage in Multiwfn

In Multiwfn, you can find the options used to set grid spacing of fitting points and box extension. The coordinates of additional fitting centers, which are beyond atomic nuclei, can be readed from external file by option 4, the format of the file is

```
numdata
X Y Z           ← For additional fitting center 1
X Y Z           ← For additional fitting center 2
...

```

numdata denotes how many entries are there in this file. X, Y and Z are coordinates (in bohr).

For flexibility consideration, coordinates of fitting points are allowed to be readed from external file to replace the CHELPG fitting points. The format of the file is

```
numdata
X Y Z [ESPval]   ← For fitting point 1
X Y Z [ESPval]   ← For fitting point 2
...

```

The *ESPval* is an optional term, which denotes precalculated ESP value at corresponding point. If *numdata* is a negative value, then the ESP values used in charge fitting will be readed from the fourth column rather than calculated by Multiwfn internally.

When you choose selection 1, Multiwfn will start to calculate ESP value at each fitting point, the process may take long time for big system, then fitted charges of all fitting centers will be

outputted on screen. The coordinates and ESP value calculated by quantum chemistry of all fitting points can optionally be exported to ESPfitpt.txt in current folder.

RMSE and RRMSE are outputted automatically, they measure the quality of fitting, the smaller value suggest that the fitted charges have better ESP reproducibility. RMSE and RRMSE are defined as (where N is the number of fitting points)

$$\text{RMSE} = \sqrt{\frac{\sum_i [V_q(\mathbf{r}_i) - V(\mathbf{r}_i)]^2}{N}}$$

$$\text{RRMSE} = \sqrt{\frac{\sum_i [V_q(\mathbf{r}_i) - V(\mathbf{r}_i)]^2}{\sum_i V(\mathbf{r}_i)^2}}$$

Notice that only small part of vdW radii are predefined in original and related papers, if your system contains other elements, Multiwfn will ask you to input vdW radius. If you don't have proper data in hand, you can input a somewhat arbitrary value, 3.4 bohr is recommended.

3.9.11 Merz-Kollmann charge

Merz-Kollmann (MK) charge is another well known charge model derived from ESP fitting, see *J. Comp. Chem.*, 11, 431. The only difference between MK and CHELPG is grid setting. In MK, the fitting points are evenly distributed on the layers of 1.4, 1.6, 1.8 and 2.0 times the vdW radii of each atoms, all fitting points enclosed by vdW surface are discarded.

In Multiwfn, the number of layers, the ratio of vdW radii used to define the layers, and the density of points per square angstrom on the shells are controllable. For detail of selections and outputs, see CHELPG section.

Informations needed: Basis functions (MPA, Löwdin, MMPA), GTFs (Hirshfeld, VDD, Becke, ADCH, CHELPG, MK), atom coordinates.

3.10 Orbital composition analysis (8)

Notice that the word “orbital” here is not restricted to molecular orbital, for example, if the input file carries natural bond orbitals (NBO), then what will be analyzed is NBOs. There is an excellent paper compared pros and cons of various orbital composition analysis approaches, see *Acta Chim. Sinica*, 69, 2393.

3.10.1 Output basis function, shell and atom composition in a specific orbital by MPA, Stout-Politzer and SCPA approaches (1, 2, 3)

MPA, SCPA and Stout-Politzer methods support decomposing orbital to basis function, shell and atom compositions. Actually I have introduced the theories in Section 3.9.5, 3.9.6 and 3.9.7, $\Theta_{i,a} * 100\%$ is just the composition of basis function a in orbital i , if we sum up all the compositions of basis functions that within a shell we will get shell composition, and if we sum up all the compositions of shells that attributed to the same atom we will get atom composition.

3 Functions

These approaches rely on basis expansion, in current Multiwfn version you must use .fch as input file.

When you entered “Orbital composition analysis” submenu from main menu, select which method you want to use for decomposition, and then input the index of orbital, the result will be printed on screen immediately, you can also input -1 to print basic informations of all orbitals to find which one you are interested in. By default, only those terms with composition larger than 0.5% will be printed, this threshold can be adjusted by “comphres” in settings.ini.

If the basis functions stored in .fch file are spherical harmonic type, then the label of basis functions printed will look like D+1, F-3 rather than XX, XYY. The labels of spherical harmonic basis functions used in Multiwfn are completely identical to Gaussian program, the conversion relationship is:

D 0=-0.5*XX-0.5*YY+ZZ

D+1=XZ

D-1=YZ

D+2= $\sqrt{3/2}^*(XX-YY)$

D-2=XY

F 0=-3/2/ $\sqrt{5}$ *(XXZ+YYZ)+ZZZ

F+1=- $\sqrt{(3/8)}^*$ XXX- $\sqrt{(3/40)}^*$ YYY+ $\sqrt{(6/5)}^*$ XZZ

F-1=- $\sqrt{(3/40)}^*$ XXY- $\sqrt{(3/8)}^*$ YYY+ $\sqrt{(6/5)}^*$ YZZ

F+2= $\sqrt{3/2}^*(XXZ-YYZ)$

F-2=XYZ

F+3= $\sqrt{(5/8)}^*$ XXX-3/ $\sqrt{8}$ *XYY

F-3=3/ $\sqrt{8}$ *XXY- $\sqrt{(5/8)}^*$ YYY

G 0=ZZZ+3/8*(XXXX+YYYY)-3* $\sqrt{(3/35)}^*$ (XXZZ+YYZZ-1/4*XXYY)

G+1=2* $\sqrt{(5/14)}^*$ XZZ-3/2* $\sqrt{(5/14)}^*$ XXXZ-3/2/ $\sqrt{14}$ *XYYZ

G-1=2* $\sqrt{(5/14)}^*$ YZZ-3/2* $\sqrt{(5/14)}^*$ YYYZ-3/2/ $\sqrt{14}$ *XXYZ

G+2=3* $\sqrt{(3/28)}^*$ (XXZZ-YYZZ)- $\sqrt{5/4}$ *(XXXX-YYYY)

G-2=3/ $\sqrt{7}$ *XYZZ- $\sqrt{(5/28)}^*$ (XXXY+YYYY)

G+3= $\sqrt{(5/8)}^*$ XXXZ-3/ $\sqrt{8}$ *XYYZ

G-3=- $\sqrt{(5/8)}^*$ YYYZ+3/ $\sqrt{8}$ *XXYZ

G+4= $\sqrt{35/8}$ *(XXXX+YYYY)-3/4* $\sqrt{3}$ *XXYY

G-4= $\sqrt{5/2}$ *(XXXY-YYYY)

3.10.2 Define fragment 1 and 2 (-1, -2)

Before doing composition analysis for fragments by Mulliken, Stout-Politzer and SCPA approaches, you have to define fragment in advance. If what you are interested in is only composition of one fragment rather than the composition between two fragments (cross term composition), you only need to define fragment 1. The content of fragment can be chosen to basis functions, shells, atoms or mixture of them, whatever you choose, only the indices of corresponding basis functions are recorded eventually. Notice that the "fragment" I referred here has *no any relationship* with the "fragment" involved in Section 3.1, the fragment defined here doesn't disturb wavefunction at all.

All supported commands in the interface of defining fragment are self-explained, so I will not reiterate them but only give an examples.

Define fragment as all P-shells of atom 3: First, type command “*all*”, all basis functions are listed, now find out the shells which are attributed to center 3 and only contain basis functions of type X, Y and Z (namely P-shell), assume that the indices of such shells are 3, 6 and 7, then input “*s 3,6,7*” to add them into fragment If you want to verify your operation, use “*all*” again and check if asterisks have appeared in the leftmost of corresponding rows, the marked basis functions are those which have been included in the fragment. Finally, type letter “*q*” to save setting and return to last menu, the indices of basis functions in the fragment will be printed at the same time.

By default, fragments have no any content. Each time you enter the fragment defining interface, the status of fragment is the same as the one when you leave the interface last time, so if you have defined the fragment before and you want to completely redefine it but to modify, don’t forget to use “*clean*” command first.

3.10.3 Output composition of fragment 1 and inter-fragment composition by MPA, Stout-Politzer and SCPA approaches (4, 5, 6)

After you defined fragment 1, the fragment composition analysis based on MPA, Stout-Politzer and SCPA approaches is available. The fragment composition is the sum of all basis function compositions within the fragment, in this function the fragment compositions of all orbitals are printed on screen at the same time. If the analysis method you chose is Mulliken (subfunction 4) or Stout-Politzer (subfunction 5), below component terms are outputted together with total composition:

c^2 term: The sum of square of coefficients of basis functions within fragment 1, namely $\sum_{a \in frag1} C_{a,i}^2 * 100\%$.

Int.cross: The sum of internal cross terms in fragment 1, namely $\sum_{a \in frag1} \sum_{b \in frag1} C_{a,i} C_{b,i} S_{a,b} * 100\%$.

Ext.cross: Fragment 1 part of the total cross term between fragment 1 and all other atoms, namely $\sum_{a \in frag1} \sum_{b \notin frag1} w_{a,b} 2C_{a,i} C_{b,i} S_{a,b} * 100\%$.

It is clear that total composition of fragment 1 equals to c^2 term + Int.cross + Ext.cross.

If the fragment 2 is also defined (you *must* have already defined fragment 1), in subfunction 5 (Mulliken) or subfunction 5 (Stout-Politzer) the cross term between fragment 1 and fragment 2 in each orbital, namely $\sum_{a \in frag1} \sum_{b \in frag2} 2C_{a,i} C_{b,i} S_{a,b} * 100\%$ will be outputted too. “Frag1 part” and

“Frag2 part” correspond to the components of cross term attributed to fragment 1 and fragment 2 respectively, for Mulliken analysis the two terms are of course exactly equal due to the “equal partition”.

3.10.4 Decompose orbitals to atomic contribution by Hirshfeld partition (7)

Hirshfeld weighting function (see Section 3.7.1) can also be used for decomposing orbital to atom compositions, the composition of atom A in orbital i is $\int \varphi_i^2(\mathbf{r}) w_A^{Hirsh}(\mathbf{r}) d\mathbf{r} \times 100\%$. This method has great basis-set stability and is always more reliable and reasonable than MPA and MMPA.

Before the calculation of atom composition, Multiwfn generates free-state atom wavefunctions first (see Section 3.7.3) for constructing weighting function, after that initialization step will take a while, then you input the orbital index that you are interested in. Because numerical quadrature always introduces some errors, so the sum of all atom compositions is not exactly equals to 100%, the deviation might be relatively significant in rare cases, so Multiwfn normalizes results automatically and prints them under the title "After normalization".

If you want to view composition of an atom in specific range of orbitals at the same time, you can choose subfunction -2, then input the atom index and the range of orbital index.

3.10.5 Orbital composition analysis by natural atomic orbital approach (8)

Theory

The first step of the famous natural bond orbital (NBO) analysis is converting original basis functions to natural atomic orbitals (NAOs) by use of density matrix. Resulting NAOs can be classified into three categories: (1) Core-type NAOs, describing inner core densities, their occupation numbers are almost equal to integer (2) Valence-type NAOs, describing valence densities, generally they have high occupation numbers (3) Rydberg-type NAOs, mainly displaying characteristics of polarization and delocalization of electrons, the occupation numbers of them are very low. Core and valence NAOs are collectively named as minimal-set, they have strong physical meaning and have one-to-one relationship with "actual" atomic orbitals, so they are what we should be most concerned. Occupied MOs are almost exclusively contributed by minimal-set NAOs. Rydberg NAOs don't have clear physical interpretations, their contribution can be ignored in occupied MOs, however they often have great contribution to virtual orbitals.

Since NAOs is a orthonormal set, if we have MO coefficient matrix in NAO basis, we can get contribution from a NAO to specific MO by simply squaring corresponding expansion coefficient and then multiply it by 100%. Composition of an atom can be calculated by summing up composition of minimal-set NAOs in this center.

This NAO approach has great basis-set stability as Hirshfeld approach, especially suitable for analyzing composition of occupied orbitals. However for virtual orbitals, contribution from Rydberg NAOs is often large, the NAO approach is no longer works well.

Input file

The MO coefficient matrix in NAO basis cannot be generated by Multiwfn itself, you need to provide a output file of NBO program containing this matrix as Multiwfn input file. By default NBO program doesn't output this matrix, you need to manually add *NAOMO* keyword between

3 Functions

\$NBO ... \$END field in NBO input file. The NBO program we referred here may be stand-alone NBO program (also known as GENNBO), or NBO module embedded in quantum chemistry software such as L607 in Gaussian. The output from NBO3.1 and NBO5.0 are formally supported, I estimate that no problem for other NBO versions.

Options

You will find following options in the interface:

-1 Define fragment: This option is used to define fragment, which is needed by fragment contribution analysis (option 1). All commands are self-explained. You can also refer to Section 3.10.2, the difference is that shell information cannot be utilized here.

0 Show composition of an orbital: Print contribution from some or all NAOs to specific MO. Since the numerical precision of expansion coefficients recorded in NBO output is not high, if you choose to output all NAOs, the sum may be slightly deviate from 100%, it is unnecessary to take care of this problem. If you choose to output core and valence NAOs only, the larger deviation of the sum from 100% denotes the larger Rydberg composition is. The printed NAO contributions are also summed up according to the atom center they attributed to, and then be outputted as "Condensed above result to atoms".

1 Show fragment contribution in a range of orbitals: Contribution from the fragment you defined to specified range of orbitals will be printed. The total contribution consist of contribution from core, valence, Rydberg NAOs in the fragment.

2 Select output mode: This option controls which set of NAOs will be outputted by option 0, there are three modes:

- (1) Show all NAOs
- (2) Only show core and valence NAOs: This is default mode, Rydberg NAOs will be ignored.
- (3) Show NAOs whose contribution is larger than specified criteria: Only the NAO (any type) whose contribution to specific MO is larger than you specified criteria will be outputted.

3 Switch spin type: One can find this option only if the NBO output file is for open-shell system. One can select which spin type of MOs to be analyzed.

Informations needed: Basis functions (Mulliken, Stout-Politzer and SCPA), atom coordinates and GTFs (Hirshfeld approach), MO coefficients in NAO basis (NAO approach)

3.11 Bond order analysis (9)

In bond order analysis module, you can directly select corresponding option to calculate bond order by corresponding method. If you want to obtain total bond order between atoms in two molecular fragments, you can manually sum up corresponding terms, however this work is tedious, the better way is using option -1 to define fragment 1 and 2 first before bond order analysis, after that the inter-fragment bond order will be outputted alongside two-center bond orders.

3.11.1 Mayer 2-center bond order analysis (1)

The Mayer bond order between atom *A* and *B* is defined as (*Chem. Phys. Lett.*, 97, 270)

3 Functions

$$I_{AB} = I_{AB}^\alpha + I_{AB}^\beta = 2 \sum_{a \in A} \sum_{b \in B} [(P^\alpha S)_{ba} (P^\alpha S)_{ab} + (P^\beta S)_{ba} (P^\beta S)_{ab}]$$

Where P^α and P^β are alpha and beta density matrix respectively, S is overlap matrix. Above formula can be equivalently rewritten using total density matrix $P = P^\alpha + P^\beta$ and spin density matrix $P^S = P^\alpha - P^\beta$

$$I_{AB} = \sum_{a \in A} \sum_{b \in B} [(PS)_{ba} (PS)_{ab} + (P^S S)_{ba} (P^S S)_{ab}]$$

For restricted close-shell circumstance, since spin density matrix is zero, the formula can be simplified to

$$I_{AB} = \sum_{a \in A} \sum_{b \in B} (PS)_{ab} (PS)_{ba}$$

Generally, the value of Mayer bond order is in agreement with empirical bond order, for single, double and triple bond the value is close to 1.0, 2.0 and 3.0 respectively. For unrestricted or restricted open-shell wavefunction, alpha, beta and total Mayer bond orders will be outputted separately. By default, only the bonds whose bond order exceed 0.1 will be printed on screen, the threshold can be adjusted by “bndordthresmayer” parameter in settings.ini, you can also select to export full bond order matrix.

Moreover, Multiwfn outputs total and free valences, the former is defined as

$$V_A = 2 \sum_{a \in A} (PS)_{aa} - \sum_{a \in A} \sum_{b \in A} (PS)_{ab} (PS)_{ba}$$

The latter is defined as

$$F_A = V_A - \sum_{B \neq A} I_{AB} = \sum_{B \neq A} \sum_{a \in A} (P^S S)_{ab} (P^S S)_{ba}$$

Where P^S is spin density matrix, namely P^α minus P^β . It is evident that for restricted close-shell wavefunction free valences are zero, since $P^S=0$. Total valences measures atomic bonding capacities, while free valences denote the remaining ability of forming new bonds by sharing electron pairs.

For unrestricted or restricted open-shell system, there is another way to calculate total bond order rather than summing up alpha and beta bond orders, that is summing up alpha and beta density matrices to form total density matrix first and then calculate Mayer bond order by using restricted close-shell formula, this treatment is sometimes called “generalized Wiberg bond order”, these total bond orders are printed following the title “Mayer bond order from mixed alpha&beta density matrix”.

Mayer bond order and multi-center bond order described below are sensitive to basis-set, don't use basis-set with diffusion functions, otherwise the bond order result will be unreliable.

3.11.2 Multi-center bond order analysis (2)

Mayer two-center bond order had been extended to multi-center, Multiwfn supports it up to six-center. Three/four/five/six-center bond orders are defined respectively as

$$I_{ABC} = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} [(PS)_{ab} (PS)_{bc} (PS)_{ca}]$$

$$I_{ABCD} = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \sum_{d \in D} [(PS)_{ab} (PS)_{bc} (PS)_{cd} (PS)_{da}]$$

3 Functions

$$I_{ABCDE} = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \sum_{d \in D} \sum_{e \in E} [(PS)_{ab} (PS)_{bc} (PS)_{cd} (PS)_{de} (PS)_{ea}]$$

$$I_{ABCDEF} = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \sum_{d \in D} \sum_{e \in E} \sum_{f \in F} [(PS)_{ab} (PS)_{bc} (PS)_{cd} (PS)_{de} (PS)_{ef} (PS)_{fa}]$$

Similarly, infinity-center bond orders can be written as

$$I_{ABCDEF...K} = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \dots \sum_{k \in K} [(PS)_{ab} (PS)_{bc} (PS)_{cd} \dots (PS)_{ka}]$$

You can hack “multicenter” routine in “bondanalysis.f90” to make Multiwfn support more than six-center bond order, by consulting existing codes, you must feel it is very easy.

For unrestricted or restricted open-shell wavefunction, the result of multi-center bond order analysis consists of four terms:

1 The bond order from mixed alpha&beta density matrix: The density matrix used in above formulas is total density matrix, that is $P^\alpha + P^\beta$.

2 The bond order from alpha density matrix: The density matrix used is P^α , that is the Mayer bond order for alpha density.

3 The bond order from beta density matrix: The density matrix used is P^β , that is the bond order for beta density.

4 The sum of bond order from alpha&beta density matrix: The sum of bond order for alpha density and for beta density. Likewise term 1, this term measures total strength of multi-center bond, however the value is different from term 1. This term has stronger physical meaning than term 1, while the value always tends to be smaller.

Due to the ambiguity of definition, the value of multi-center bond order may relies on input direction if those atoms lack of symmetry. For example, the bond order of ABCDEF may be slightly or significantly different from FEDCBA (namely the opposite input direction of ABCDEF). In my own viewpoint, in order to obtain reasonable six-center bond order for atom A-B-C-D-E-F one should calculate ABCDEF and FEDCBA respectively and take their average value. While some people argued that one need to take all possible permutation into account to get definitive result, see *J. Phys. Org. Chem.*, 18, 706, that means the bond order of BCADEF, CABDEF, DBCAFE and so on ($6! = 720$ in total) are all needed to be calculated and then sum up.

If you input -3 when Multiwfn asking how many center you want to calculate, all three-center bond order will be calculated, only those larger than the threshold you inputted will be printed. Similarly, four-, five- and six-center bonds can be searched by inputting -4, -5 and -6 respectively, however, due to efficiency consideration, the search may be not exhaustive.

3.11.3 Wiberg bond order analysis in Löwdin orthogonalized basis (3)

Wiberg bond order is defined as (footnote in *Tetrahedron*, 24, 1083)

$$I_{AB} = \sum_{a \in A} \sum_{b \in B} P_{ab}^2$$

Wiberg bond order is only suitable for the wavefunction represented by orthogonal basis functions such as most of semiempirical wavefunctions, and only defined for restricted close-shell system. Actually, Mayer bond order can be seen as a generalization of Wiberg bond order, under restricted close-shell system and orthonormal basis functions (namely S matrix is identity matrix) their results are completely identical.

In this function, Multiwfn orthogonalizes basis functions by Löwdin method first and then

performs usual Mayer bond order analysis, the threshold for printing is controlled by “bndordthresmayer” in settings.ini too.

3.11.4 Mulliken bond order analysis (4) and decomposition (5)

Mulliken bond order is the oldest bond order definition, it is defined as

$$I_{AB} = \sum_i \eta_i \sum_{a \in A} \sum_{b \in B} 2C_{a,i} C_{b,i} S_{a,b}$$

Mulliken bond order has low agreement with empirical bond order, it is deprecated for quantifying bonding strength, for which Mayer bond order always performs better. However, Mulliken bond order is a good qualitative indicator for bonding (positive value) and antibonding (negative value). The threshold for printing results is controlled by “bndordthresmul” parameter in settings.ini.

Mulliken bond order is easy to decompose to orbital contributions, the contribution from orbital i to bond order AB is

$$I_{AB}^i = \eta_i \sum_{a \in A} \sum_{b \in B} 2C_{a,i} C_{b,i} S_{a,b}$$

From the decomposition, we can know which orbitals are favourite and unfavourite for specific bonding.

Informations needed: Basis functions, atom coordinates.

3.12 Plotting total, partial, and overlap population density-of-states (10)

3.12.1 Theory

Density-of-states (DOS) is an important concept of solid physics, which means number of states in unit energy interval, since energy levels are contiguous, so DOS can be plotted as curve map. In isolated system (such as molecule), the energy levels are discrete, the concept of DOS is questionable and some people argued that DOS is completely valueless in this situation. However, if the discrete energy levels are broadened to curve artificially, DOS graph can be used as a valueable tools for analyzing the nature of electron structure.

The original total DOS (TDOS) of isolated system can be written as

$$TDOS(E) = \sum_i \delta(E - \varepsilon_i)$$

Where $\{\varepsilon\}$ is eigenvalue set of single-particle Hamilton, δ is dirac delta function. If δ is replaced by broadening function $F(x)$, such as Gaussian, Lorentzian and pseudo-Voigt function, we get broadened TDOS.

The normalized Gaussian function is defined as

$$G(x) = \frac{1}{c\sqrt{2\pi}} e^{-\frac{x^2}{2c^2}} \quad \text{Where} \quad c = \frac{\text{FWHM}}{2\sqrt{2\ln 2}}$$

3 Functions

FWHM is acronym of “full width at half maximum”, it is an adjustable parameter in Multiwfn, the larger FWHM the TDOS graph looks more smooth and analysis is easier to perform, but more fine-structure is masked.

The normalized Lorentzian function is defined as

$$L(x) = \frac{\text{FWHM}}{2\pi} \frac{1}{x^2 + 0.25 * \text{FWHM}^2}$$

Pseudo-Voigt function is weighted linear combination of Gaussian function and Lorentzian function:

$$P(x) = w_{gauss} G(x) + (1 - w_{gauss}) L(x)$$

Obviously, if $G(x)$ and $L(x)$ are normalized, normalization condition for $P(x)$ always holds whatever w_{gauss} .

The curve map of broadened partial DOS (PDOS) and overlap DOS (OPDOS) are valuable for visualizing orbital composition analysis, PDOS function of fragment A is defined as

$$PDOS_A(E) = \sum_i \Xi_{i,A} F(E - \varepsilon_i)$$

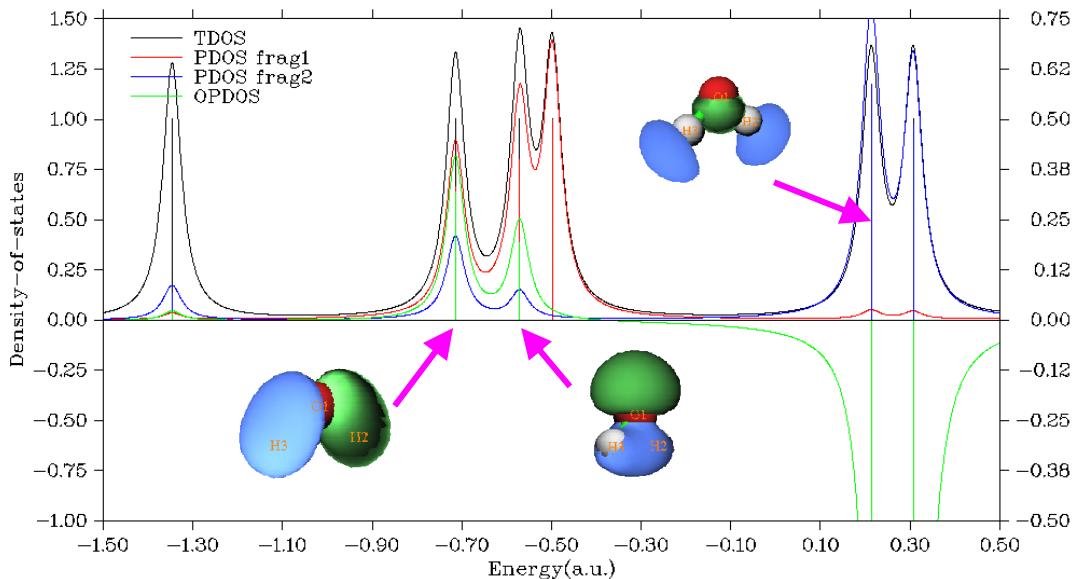
Where $\Xi_{i,A}$ is the composition of fragment A in orbital i . The OPDOS between fragment A and B is defined as

$$OPDOS_{A,B}(E) = \sum_i X_{A,B}^i F(E - \varepsilon_i)$$

Where $X_{A,B}^i$ is the composition of total cross term between fragment A and B in orbital i . I have discussed how to calculate Ξ and X in Section 3.10.3.

Let us see an example, from which we can know what new informations can be revealed by DOS map. This is a water molecule under HF/6-31G* wavefunction in ground state, the orbitals are canonical MOs, fragment 1 is defined as P-shells of oxygen (correspond to 2p atomic orbitals), fragment 2 is defined as two hydrogens, both original and broadened TDOS/PDOS/OPDOS are shown in the graph below. Notice that the height is only meaningful for lines (original data) but not for curves, left-axis and right-axis correspond to TDOS/PDOS and OPDOS respectively. The original DOS graph is discrete comb-like lines, from which we cannot obtain any additional information other than energy level distribution, it is impossible to distinguish different type of lines and degenerate energy levels owing to the overlapping. However if the discrete lines are broadened, from the height of black curve (TDOS) we can clearly know how dense the energy levels are distributed everywhere. Besides, the curves corresponding to TDOS, PDOS (red line for fragment 1, blue line for fragment 2) and OPDOS (green line) no longer overlap, we can clearly identify characters of each orbital by observing these curves. For example, the red curve is high and nearly approaches black line in the region of -0.9 a.u. to -0.3 a.u., so we can be concluded that 2p atomic orbitals of oxygen have significant contribution to corresponding MOs. Since the green curve is greater than or less than zero respectively denote corresponding MOs are favorable or unfavorable for forming chemical bond between oxygen 2p orbitals and hydrogens, it is shown that the orbital pointed by the leftmost arrow is very helpful for bonding, while the high-energy state orbitals (> 0.1 a.u.) are not conducive for bonding (namely antibonding character),

fortunately they haven't been occupied, otherwise the molecule must be broken. These conclusions can be confirmed further by observing isosurfaces of corresponding MOs.



3.12.2 Input file

.fch files can be used as input. You can also use output file of single point task of Gaussian program as input (“pop=full” keyword must be specified). For generality, Multiwfn supports plain text file as input too, the format is

```

nmo inp
energy [strength] [FWHM]           ← For orbital 1
energy [strength] [FWHM]           ← For orbital 2
energy [strength] [FWHM]           ← For orbital 3
...
energy [strength] [FWHM]           ← For orbital nmo

```

nmo is the number of orbitals recorded in this file, *inp* is input type, if is set to 1, only energies (in a.u.) will be readed, strengths and FWHMs of all orbitals are set to 1.0 and 0.25 respectively and automatically; if is set to 2, you have to specify strength and FWHM for each orbital too. If the strength of an orbital is set to *k*, then the broadened curve from this orbital will be normalized to *k* rather than 1 (default value). The format of this file is free, there is no upper limit of the number of orbitals.

3.12.3 Options for plotting DOS

When you entered the interface of plotting DOS, you will see below options:

-4 Show all orbital information: Print information of all orbitals on screen, also point out Fermi energy (when occupation number information is available).

-3 Output energy levels, strengths, FWHMs to plain text file: Output energy, strength and FWHM of each orbital to “orginfo.txt” in current directory, this file complies the format

introduced in last section, so can be directly used as input file.

-2 Define fragment 2: Only when fragment 2 is defined (fragment 1 must have been defined), the DOS graph will contain the lines and curves of “PDOS frag2” and “OPDOS”. The interface for defining fragment is the same as the one described in Section 3.10.2.

-1 Define fragment 1: DOS graph contains the line and curve of “PDOS frag 1” only when fragment 1 is defined.

0 Draw TDOS graph: Draw TDOS graph right now, if fragment 1 or both 1 and 2 are defined, the texts in this option will become “TDOS+PDOS” and “TDOS+PDOS+OPDOS” automatically.

1 Select broadening function: Select which broadening function will be used, you can select Lorentzian, Gaussian or Pseudo-Voigt function. Default is Lorentzian.

2 Set energy range: Set the energy range for plotting, that is set X-axis range.

3 Set full width at half maximum (FWHM): As the title says.

4 Set scale ratio for DOS curve: If this option is set to k , then height of all curves will multiply k (in full energy range).

5 Set Gaussian-weighting coefficient: This option sets w_{gauss} which is mentioned in last section, this option only appears when Pseudo-Voigt function is chosen.

6 Switch spin: Appears only when the loaded wavefunction is unrestricted .fch file, this option determines which set of energy levels (alpha or beta) will be used for analysis.

7 Switch method for calculating PDOS: Switch method of composition analysis between SCPA and Mulliken.

Notice that option -1, -2 and 7 only appear when the input file contains basis function informations, that is if you want to draw PDOS and OPDOS graph, you have to use .fch file as input.

Once you choose option 0, Multiwfn starts to calculate data set and then DOS graph pops up, after clicking right mouse button on the graph to close it, several options will be shown on screen, they control the range of Y-axis and if the lines or curves corresponding to TDOS, PDOS and OPDOS will be shown on DOS graph, etc. There is an option named “Set scale factor of Y-axis for OPDOS”, if the value is set to k and the range of current left-axis (for TDOS/PDOS) is [-3.5,2.0], then the range of right-axis (for OPDOS) will become [-3.5*k,2.0*k]. The reason why Multiwfn uses double axis is because the magnitude of OPDOS is generally much smaller than TDOS and PDOS.

You can also choose to export the picture you have seen, or export X-Y data set of DOS to plain text files for replotting the graph by external softwares, such as Origin. By choosing option 0 you can return to initial interface, the quality of DOS graph can be gradually improved by repeating the adjustments until you are satisfied.

Informations needed: Basis functions and atom coordinates (only for PDOS and OPDOS), energy levels.

3.13 Plotting IR/Raman/UV-Vis spectrum (11)

Although plotting spectrum is beyond the scope of wavefunction analysis, however due to the

3 Functions

mechanism is similar to plotting DOS, and this function is very useful for theoretical spectrum research, this function is added to Multiwfn.

3.13.1 Theory

For comparing theoretical results with experimental spectrum, the discrete lines corresponding to each transition mode have to be broadened to emulate the situation in real world, the broadening function for IR and Raman spectrum is Lorentzian function in common, for UV-Vis spectrum Gaussian function is commonly used, see Section 3.12.1 for details. Unlike DOS graph, in which the strengths for each energy level are always simply set to 1.0 (namely broadened curves from each energy level are normalized to 1.0), the strengths of transitions are very important data for plotting spectrum.

The unit of transition energy of IR and Raman spectrum is cm^{-1} in common, for UV/Vis spectrum all of eV ($1\text{eV}=8.0655*\text{1000cm}^{-1}$), nm and 1000cm^{-1} are common units. IR intensity, Raman activity and oscillator strength of each transition mode outputted by quantum chemistry programs are proportional to the area under the broadened curve. We discuss IR spectrum first. The frequently used unit of molar absorptivity ϵ , that is L/mol/cm can be rewritten as

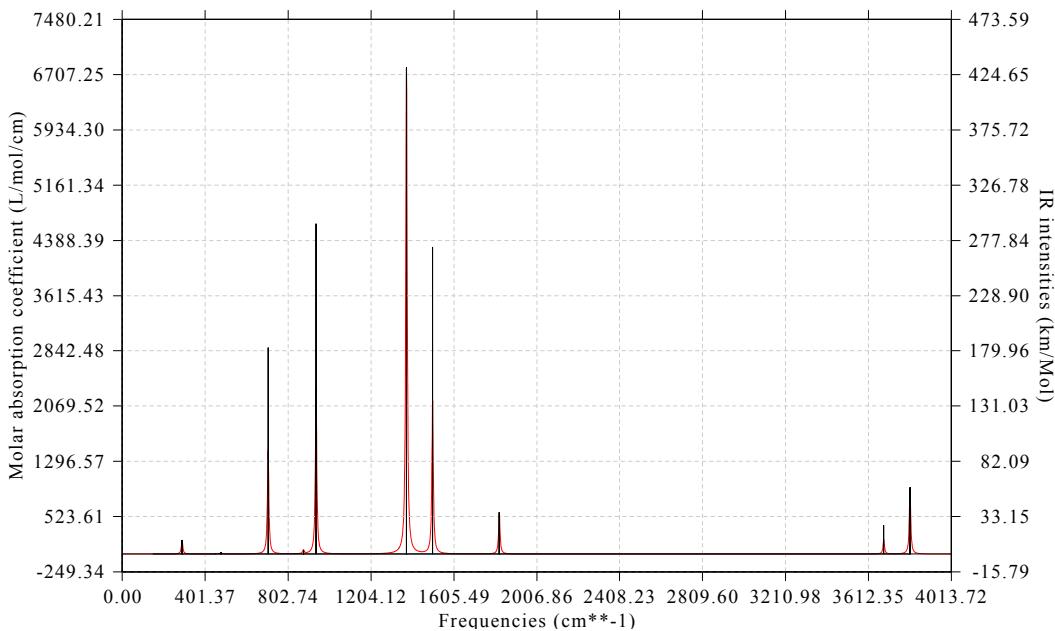
$$\frac{\text{L}}{\text{mol} * \text{cm}} = \frac{1000\text{cm}^3}{\text{mol} * \text{cm}} = \frac{1000\text{cm}^2}{\text{mol}}$$

Since IR intensity $\zeta = \int \epsilon(\nu) d\nu$, where ν is transition frequency, the unit of ζ should be

$$\frac{1000\text{cm}^2}{\text{mol}} * (\text{cm}^{-1}) = \frac{1000\text{cm}}{\text{mol}} = \frac{0.01\text{km}}{\text{mol}}$$

However km/mol is the more commonly used unit for IR intensity, therefore if IR intensity of a vibration mode is p km/mol, then the broadened curve from which should be normalized to $100*p$ (in the other words, the area under the curve is $100*p$). Sometimes the unit esu^2*cm^2 is used for IR intensity, the relationship with km/mol is $1\text{esu}^2*\text{cm}^2 = 2.5066 \text{ km/mol}$.

Below is an example of IR spectrum plotted by Multiwfn. Notice that the left axis corresponds to the curve (artificially broadened data), the right axis corresponds to the discrete lines (original transition data).



In theoretical chemistry field, oscillator strength is always used for representing transition strength in UV/Vis spectrum, however this is a dimensionless quantity and the precise relationship with molar absorptivity haven't been built. By comparing large number of theoretical data and experimental spectrums, one found an empirical relation: If the unit 1000cm^{-1} and $\text{L/mol}/\text{cm}$ are used for X-axis and Y-axis respectively, then the area under the curve that broadened from per unit oscillator strength should be $1/4.32 \times 10^6$. If eV is used as X-axis unit, the value should be $1/4.32/8.0655 \times 10^6 = 28700$. By this relation, UV/Vis spectrum can be emulated by theoretical data.

3.13.2 Input file

Currently two types of input file are supported:

1 Gaussian output file: For IR spectrum, use the output file of “freq” task as input; for Raman spectrum, use the output file of “freq=raman” task as input; for UV/Vis spectrum, use the output file of TDDFT or CI task as input.

2 Plain text file: For generality, Multiwfn supports plain text file as input, you can extract transition data from output files of other quantum chemistry packages other than Gaussian, and then fill them into a file according to the format shown below

```

numdata inptype
energy strength [FWHM]           ← For transition 1
energy strength [FWHM]           ← For transition 2
energy strength [FWHM]           ← For transition 3
...
energy strength [FWHM]           ← For transition numdata

```

numdata denotes how many entries are there in this file. If *inptype* is set to 1, then only *energy* and *strength* will be readed, and FWHMs for all transitions will be set to 8 cm^{-1} for IR and Raman spectrum and to $2/3\text{ eV}$ for UV-Vis spectrum. If *inptype* is set to 2, then *FWHM* will be readed too. The unit of both *energy* and *FWHM* is cm^{-1} for IR and Raman spectrum, eV for UV/Vis. The unit of strength is km/mol and A^4/AMU for IR and Raman spectrum respectively.

3.13.3 Options for plotting

After you select the type of spectrum, you will see below options. The meaning of some options is different for different types of spectrum.

-2 Export original transition data to plain text file: Output energies, intensities and FWHMs of all transitions to “transinfo.txt” in current directory, this file complies with the format introduced in last section, so can be directly used as input file.

-1 Show original transition data: Print energy and intensity data of all transitions on screen.

0 Plot spectrum: Plot spectrum right now.

1 Save picture: Export the spectrum to graphic file in current directory.

2 Export X-Y data set of lines and curves to plain text file: Export X-Y data set of lines and broadened spectrum to "spectrum_line.txt" and "spectrum_curve.txt" in current directory respectively, you can replot the curve and discrete line graph directly by the two files via external program, such as Origin.

3 Set lower and upper limit of X-axis: As the title says. By default the range of X-axis is adjusted automatically according to minimum and maximum transition energies.

4 Set left Y-axis: Set origin value, number of ticks and step size for left Y-axis. By default the range of Y-axis is adjusted automatically according to the maximum peak, the number of ticks is 10.

5 Set right Y-axis: Likewise option 4, but for right Y-axis.

6 Select broaden function: Gaussian, Lorentzian and Pseudo-Voigt function can be selected for broadening discrete lines.

7 Set scale ratio for curve: If the value is set to k , then the height of curve will multiply k in full range. For IR and Raman spectrum the default value is 1.0, for UV/Vis spectrum an empirical value 28700.0 is used when energy unit is eV or nm, when unit is 1000cm^{-1} the value $1/(4.32 \times 10^6)$ is used.

8 Set FWHM for all transitions: As the title says. The unit of FWHM is the same as X-axis.

9 Switch ON/OFF showing discrete lines: Choose if show the discrete lines corresponding to each transition on the spectrum graph.

10: For IR spectrum, switch the unit of IR intensities between km/mol (default) and esu^2cm^2 . For UV/Vis spectrum, choose the unit of transition energy from eV, nm and 1000cm^{-1} .

11 Set Gaussian-weighting coefficient: Sets w_{gauss} , which is mentioned in Section 3.12.1, this option only appears when Pseudo-Voigt function is chosen.

Informations needed: Transition energies and strengths.

3.14 Topology analysis (2)

3.14.1 Theory

Some real space functions in Multiwfn are available for topology analysis, such as electron density, its laplacian, orbital wavefunction, ELF, LOL etc. One can select real space functions by selection -11. Notice that once this selection is chosen, all previous topology analysis results will

be clean.

The topology analysis techniques proposed by Bader is firstly used for analyzing electron density in "atoms in molecules" (AIM) theory, this techniques have also been extended to other real space functions, e.g. the first topology analysis research of ELF for small molecules is given by Silvi and Savin, see *Nature*, 371, 683. In topology analysis language, the points at where gradient norm of function value is zero (except at infinity) are called as critical points (CPs), CPs can be classified into four types according to how many eigenvalues of Hessian matrix of real space function are negative.

(3,-3): All three eigenvalues of Hessian matrix of function are negative, namely the local maximum. For electron density analysis and for heavy atoms, the position of (3,-3) are nearly identical to nuclear positions, hence (3,-3) is also called nuclear critical point (NCP). Generally the number of (3,-3) is equal to the number of atoms, only in rarely cases the former may more than (e.g. Li₂) or less than (e.g. KrH⁺) the latter.

(3,-1): Two eigenvalues of Hessian matrix of function are negative, namely the one-order saddle point (likewise transition state in potential energy surface). For electron density analysis, (3,-1) generally appears between attractive atom pairs and hence commonly called bond critical point (BCP). The value of real space functions at BCP have great significance, for example the value of ρ and the sign of $\nabla^2 \rho$ at BCP are closely related to bonding strength and bonding type respectively in analogous bonding type (*The Quantum Theory of Atoms in Molecules-From Solid State to DNA and Drug Design*, p.11); the potential energy density at BCP has been shown to be highly correlated with hydrogen bond energies (*Chem. Phys. Lett.*, 285, 170); local information entropy at BCP is a good indicator of aromaticity (*Phys. Chem. Chem. Phys.*, 12, 4742).

(3,+1): Only one eigenvalue of Hessian matrix of function is negative, namely two-order saddle point. For electron density analysis, (3,+1) generally appears in the center of ring system and displays steric effect, hence (3,+1) is often named ring critical point (RCP).

(3,+3): None of eigenvalues of Hessian matrix of function are negative, namely the local minimum. For electron density analysis, (3,+3) generally appears in the center of cage system (e.g. pyramid P₄ molecule), hence is often referred as cage critical point (CCP).

The positions of CPs are searched by Newton method, one need to assign an initial guess point, then the Newton iteration always converge to the CP that is closest to the guess point. By assigning different guesses and doing iteration for each of them, all CPs could be found. Once searches of CPs are finished, one should use Poincaré-Hopf relationship to verify if all CPs may have been found, the relationship states that (for isolated system)

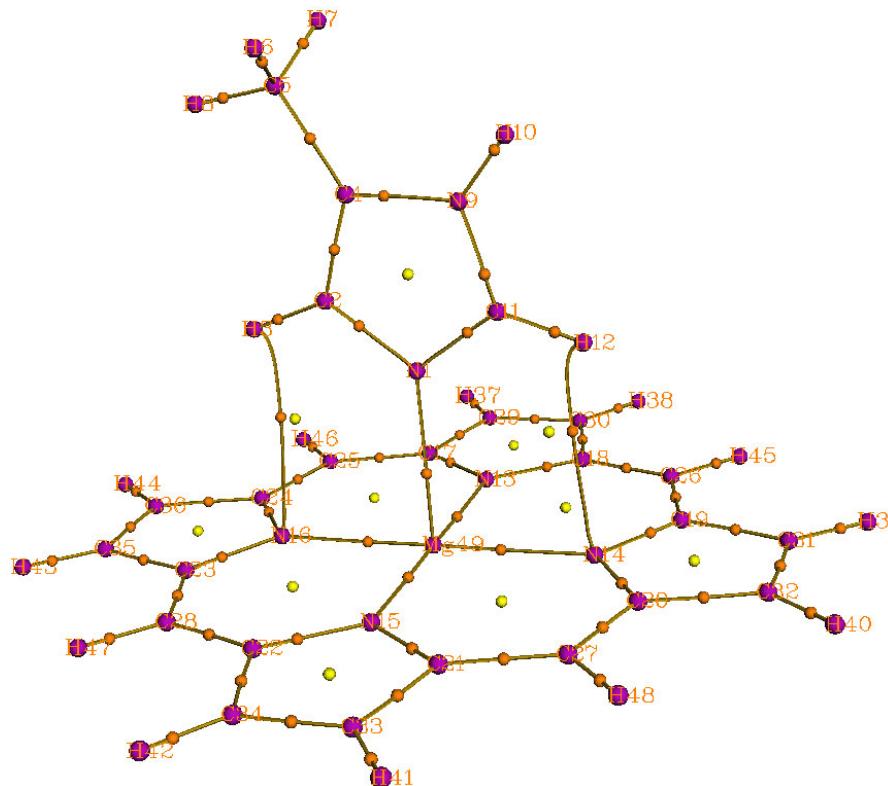
$$n_{(3,-3)} - n_{(3,-1)} + n_{(3,+1)} - n_{(3,+3)} = 1$$

If the relationship is unsatisfied, then some of CPs must be missing, you may need to try to search those CPs by different guesses. However even if the relationship is satisfied, it doesn't necessarily mean that all CPs have been found. Notice that the function spaces of ELF/LOL and laplacian of ρ are much more complex than ρ , it is very difficult to locate all CPs for these functions, especially for middle and large system, so, you can stop trying for searching CPs once all CPs that you are interested in have been found.

The maximal gradient path linking BCP and associated two local maxima of density is termed "bond path", which reveals atomic interaction path for all kinds of bonding. The collection

of bond paths is known as molecular graph, which provides an unambiguous definition of molecular structure. Bond path can be straight line or curve, obviously for the latter case the length of bond path is longer than the sum of the distances between BCP and associated two (3,-3) CPs.

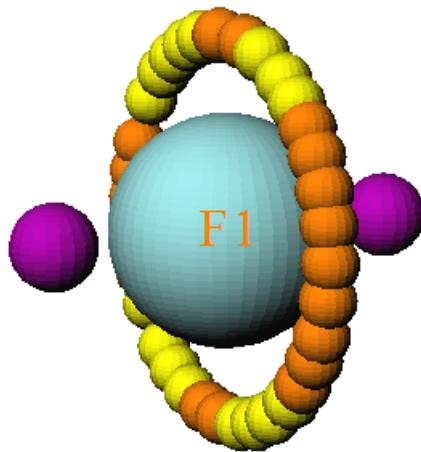
Let us see an example. In the complex shown below, the imidazole plane is vertical to magnesium porphyrin plane, the nitrogen in imidazole coordinated to magnesium. Magenta, orange and yellow spheres correspond to (3,-3), (3,-1) and (3,+1) critical points, brown lines denote bond paths.



The topology paths for other real space functions can be generated too in Multiwfn, they are very helpful to clarify intrinsic relationship between CPs.

About degenerated CP

Appearance of degenerated CPs in general system is very rare, so I mention it in the final of this section. The so-called degenerated CP is the CP have one or two zero eigenvalues of Hessian matrix. These CPs are very instable, slight perturbation of molecular geometry can break them into other type CPs. However, degenerated CPs are commonly occur in axisymmetric system. For example, in fluorine atom at groundstate, one of p orbital is singly occupied while other two are doubly occupied, topology analysis on LOL gives below picture:



The two (3,-3) correspond to the two valves of the singly occupied p orbital, while the locally maximum of electron localization arise from the other two p orbitals are represented by a circle of CPs. Ostensibly, the CPs in the circle are either (3,-1) or (3,+1) type, but in fact they are the same type degenerated CP, in which one eigenvalue is zero. Simply because the reasons of numerical convergence of the eigenvalue, they are formally recorded as different CP types in Multiwfn.

3.14.2 Search critical points

In Multiwfn several modes used to assign initial guesses are provided for Newton method:

(1) Search CPs from a given starting point: If you already know the position of CPs or you are able to guess where CPs may appear, this mode is suitable for you, the Newton iteration will start from the coordinate you inputted.

(2) Search CPs from nuclear positions: Use all nuclear positions in turn as starting points, this mode is very suitable for searching all (3,-3) for ρ , and those (3,-3)s in innermost of atom for ELF/LOL and laplacian. For electron density analysis, if you found n CPs after you used this mode, while there are more than n atoms in your system, it generally turns out that Newton method missed some (3,-3) CPs.

(3) Search CPs from midpoint of atom pairs: Use midpoint of all atom pairs in turn as starting points. This mode is very suitable for searching all (3,-1) for ρ .

(4) Search CPs from triangle center of three atoms: Likewise mode 3, but use triangle center of all combinations of three atoms in your system. Suitable for searching all (3,+1).

(5) Search CPs from pyramid center of four atoms: Likewise mode 3, but use pyramid center of all combinations of four atoms in your system. For electron density analysis, if you have tried mode 2 and mode 3 but still cannot find certain (3,+3), this mode is worth to try. This mode is more expensive than mode 3.

(6) Search CPs from a batch of points within a sphere:

You need to set sphere center, radius and the number of points first, then select option 0, the specified number (not the exact number you have set) of points will randomly distributed in the sphere as starting points for Newton method. The sphere center can be defined in very flexible way by option 2~6. If you choose option -1, then the search starts and each atom center will be used as sphere center in turn, assume that there are n atoms and the number of points in sphere you set is m , then about $n*m$ points will be used as initial guesses.

This mode is very appropriate for searching the CPs that are difficult to be located by other

modes. **For ELF, LOL and laplacian, it is highly recommended to use option -1 in this mode to locate all possible CPs.** Note that each time you carry out the search the positions of starting points are different, if some expected CPs were not found in previous searches, try to launch the search again and again to locate missing CPs.

The default searching parameters are appropriate for most systems, however in some cases you have to adjust parameters manually to ensure expected CPs could be found. Those parameters can be adjusted in the option "Set CP searching parameters".

(1) Set maximal iterations: If the number of Newton iteration exceeds this value before converged to a CP or before Hessian matrix became singular (that is the iteration cannot continue), then the Newton iteration will be terminated.

(2) Set scale factor for stepsize: The default value is 1.0, that is using original form of Newton method. Sometimes reduce the stepsize is benefit for locate CPs. For example the "Uracil.wfn" in example folder, by using mode 2 with default parameter the BCP linking N6 and H12 cannot be found, if you reduce the stepsize to 0.5, then the problem solved.

(3)/(4) Criteria for gradient-norm/displacement convergence: If both gradient-norm in current position and displacement in last step is smaller than the two values then the iteration stops and current position will be regarded as a critical points. Notice that for the elements heavier than Ar, during the search of their nuclear critical points, the convergence criteria are automatically multiplied by 10000, because the cusps of electron density at these nuclear positions are very sharp and hence hard to be located by default criteria.

(5) Minimal distance between CPs: If an iteration converged to a CP, however the distance between this CP and any CP that has been found is smaller than this value, then the CP just found will not be considered as a new CP, and hence discarded.

(6) Skip search if distance between atoms is longer than the sum of their vdW radius multiplied by: In CP searching mode 3, 4 and 5, if the distance of any two atoms that involved in the combination is longer than the sum of their vdW radii multiplied this value then current search will be skipped. The purpose of this option is to reduce the number of searches and hence computational cost of huge system. For example the imidazole--magnesium porphyrin complex mentioned earlier, there are $49*48/2=1176$ atom pairs, hence if this cutoff strategy is not employed when you choose searching mode 2, Multiwfn will try as many as 1176 searches. While if this strategy is used, since only closely related atom pairs will be considered, only 274 searches are needed be performed, it can be found that their results are identical.

(7) If print details of CP searching procedure: This option is present only in serial mode. User can select output level of details of CP searching procedure. This option is mainly used for debugging.

(8) Criteria for determining if Hessian matrix is singular: If the absolute value of determinant of Hessian matrix is lower than this value, then the Hessian matrix will be regarded as singular and terminate search. Too large of this value may lead some CPs be omitted, while too small of this value may cause numerical instability. Default value is appropriate for most cases.

3.14.3 Generate paths

Once CPs are found, you can choose to generate paths. In fact, the so-called "path" is a bunch of points evenly distributed in a curve.

(1) Generate the path connected (3,-3) and (3,-1): As the title says. Arithmetically, firstly displace the coordinate of each (3,-1) forward and backward respectively along the eigenvector that correspond to the positive eigenvalue of Hessian, and then go uphill following the gradient vector until encounters a (3,-3), the resultant trajectories compose the bond paths.

(2) Generate the path connected (3,+1) and (3,+3): Likewise above, the difference is that the starting points are (3,+1), and their coordinates are firstly moved forward and backward along the eigenvector correspond to the negative eigenvalue of Hessian, and then go downhill following the gradient vector.

In the option "Set path searching parameters", there are some suboptions used to adjust parameters for generating paths.

(1) Maximal number of points of a path: In each generation of paths, if the number of steps is reached this value before encounters a CP, then the trajectory will be discarded. For generating very long path, default value may need to be enlarged.

(2) Stepsize: The space between neighbour points that composed the paths. For paths with large curvature, sometimes they cannot be generated under default stepsize, you need to properly decrease the stepsize and regenerate paths. If you set maximal number of points and stepsize of paths as m and n respectively, then the maximal length of path is $m*n$.

(3) Stop generation if distance to any CP is smaller than: During the generation of paths, if the distance between current position and a CP that has been found, then it will be regarded as that the path has already reached the CP.

3.14.4 Generate interbasin surfaces

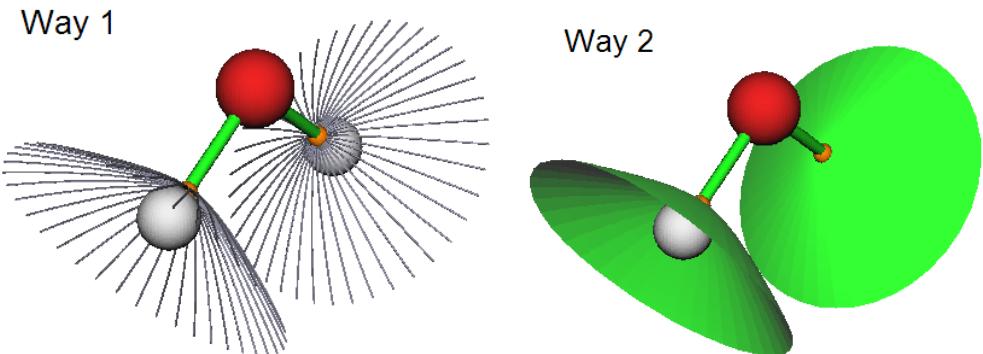
The interbasin surfaces (IBS) generated by Multiwfn actually consist of a bunch of paths derived from (3,-1) CPs, these surfaces divide the whole space into respective region for each (3,-3) CPs. By the function "Add or delete interbasin surfaces", you can generate, delete and check interbasin surfaces. Notice that before generating IBS, generation of CPs should be completed first, and at least one (3,-1) must be found.

If you want to generate the IBS from the (3,-1) CP with index of 15, then simply input *15* (you will find it is useful to visualize CPs by function 0 first to get the CP index). To delete this surface, input *-15* (negative sign means "delete"). If there is no IBS presented and you hope to generate all IBSs, input *0*. To delete all already generated IBS, also input *0*. A list of generated IBS can be printed by inputting the letter *l*. If you need to export the paths of a specific IBS (e.g. corresponding to the (3,-1) CP with index of 4) to external file, input *o 4*, then the coordinates of all paths derived from the 4th CP will be saved to surpath.txt in current folder. Input letter *q* can return to upper level menu.

Parameters for generating IBSs can be adjusted by the option "Set interbasin surface generating parameters". Enlarging number of paths in each IBS or lowering stepsize make IBSs looks more smooth. The length of paths in IBSs, or say the area of IBSs, is proportional to product of stepsize and number of points in each IBS path. Notice that once the parameters are changed, all generated IBSs will be lost.

There are three ways to portray IBSs in Multiwfn, which can be controlled by "isurfstyle" in settings.ini. Way 1 represents the IBS directly by the paths derived from corresponding (3,-1) CP; By Way 2 IBSs are shown by solid surfaces, this is default style; Way 3 is uncommonly used, you

can try it by yourself.



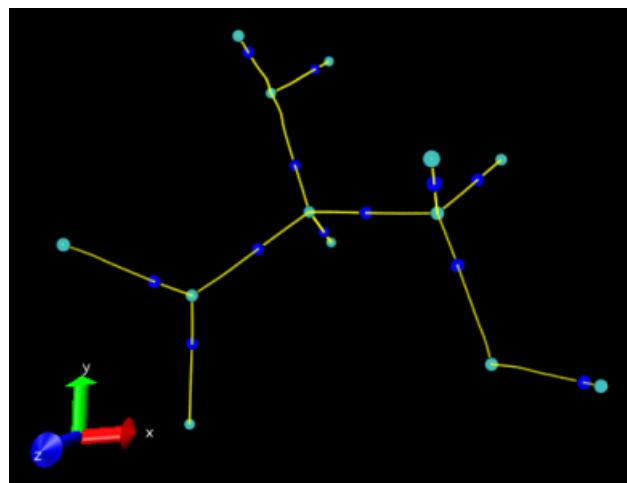
3.14.5 Visualize, analyze, modify and export results

If you select the function “Print and visualize all generated CPs, paths and surfaces”, a GUI will pop up, you can control if showing each type of CPs, paths, interbasin surfaces, molecular structure, labels etc. Meanwhile, a summary of found CPs and paths are printed in the command-line window. The satisfaction of Poincaré-Hopf relationship is also checked.

You can obtain values of all real space functions as well as gradient and Hessian matrix of the real space function you selected at a given CP by function “7 Show a variety of properties of a specified CP”.

In the option “Modify or export CPs”, you can print, delete, add and export CPs. The positions and types of all found CPs can be saved to a formatted text file “CPs.txt” in current folder by choosing suboption 4, the informations of CPs can also be loaded from an external formatted text file by suboption 5 (the found CP at current session will be clean), notice that the file format must be identical to the one outputted by suboption 4. All CPs can be exported as pdb file by suboption 6, so that CPs can be conveniently visualized by external visualization softwares, element C/N/O/F correspond to (3,-3)/(3,-1)/(3,+1)/(3,+3) respectively.

In the option “Modify, print detail or export paths”, you can print summary of generated paths, print positions of all points in specified path or delete some paths. By selecting suboption 4 and 5 you can respectively export out the detail informations of all paths to “paths.txt” in current folder or import in paths from an external file, the file format must be identical to the one outputted by suboption 4. By suboption 6, all points in all paths can be exported to a pdb file in current folder, so that paths can be conveniently visualized by external visualization softwares. For example, we use Multiwfn to perform QTAIM analysis for serine, then export CPs and paths as pdb file, then load them into VMD program and rendered as "Points" style and slightly adjust point size and color, we get below graph. The yellow paths are bond paths, blue and cyan points correspond to (3,-3) and (3,-1) CPs.



Distance, angle and dihedral angle between atoms and CPs can be conveniently measured by using option -9.

If the real space function you selected is electron density, option 20 will be presented. This is a utility to calculate Shannon aromaticity (SA) index, see *Phys. Chem. Chem. Phys.*, 12, 4742 for detail. The formula can be briefly written as

$$SA = \ln(N) + \sum_i^N (-p_i \ln p_i) \quad \text{Where } p_i = \frac{\rho(\mathbf{r}_{BCP_i})}{\sum_i^N \rho(\mathbf{r}_{BCP_i})}$$

In above formula, N is the total number of the BCPs in the ring you want to study aromaticity, \mathbf{r}_{BCP} is the position of BCP. In option 20, you need to input N and the index of these CPs in turn, then Shannon aromaticity index will be printed immediately.

Informations needed: GTFs, atom coordinates.

3.15 Quantitative analysis of molecular surface (12)

3.15.1 Theory

Molecular electrostatic potential (ESP), $V(\mathbf{r})$, has been widely used for prediction of nucleophilic and electrophilic sites, as well as molecular recognition mode for a long time, the theoretical basis is that molecules always tend to approach each other in a complementary manner of ESP. These analyses of ESP are common performed on molecular van der Waals (vdW) surface. Although the definition of such a surface is arbitrary, most people prone to take the 0.001 isosurface of electron density as vdW surface, since this definition reflects the specific features of the particular molecule, such as lone pairs, π electrons etc., this is also what the definition used in our analyses.

The analysis of ESP on vdW surface has been further quantified to extract more informations. It is shown that the strength and orientation of weak interaction, including such as hydrogen bonding, dihydrogen bonding and halogen bonding, can be well predicted and explained by

3 Functions

analyzing the magnitude and positions of minima and maxima on the surface. Politzer and coworkers have defined a set of molecular descriptors based on ESP on vdW surface, which are taken as independent variables of general interaction properties function (GIPF). GIPF successfully connects distribution of ESP on vdW surface and many condensed phase properties, including boiling point, surface tension, heats of vaporization and sublimation, LogP, etc. Below I brief these descriptors.

\bar{V}_S^+ and \bar{V}_S^- denote average of positive and negative ESP on vdW surface respectively

$$\bar{V}_S^+ = (1/m) \sum_{i=1}^m V(\mathbf{r}_i) \quad \bar{V}_S^- = (1/n) \sum_{j=1}^n V(\mathbf{r}_i)$$

Where i and j are indices of sampling points in positive and negative regions respectively. The S subscript means "Surface". The average of ESP over the entire surface is

$$\bar{V}_S = (1/t) \sum_{k=1}^t V(\mathbf{r}_i)$$

Π is the average deviation over the surface, which is viewed as a indicator of charge separation:

$$\Pi = (1/t) \sum_{k=1}^t [V(\mathbf{r}_k) - \bar{V}_S]$$

The total ESP variance can be written as the sum of positive and negative parts:

$$\sigma_{tot}^2 = \sigma_+^2 + \sigma_-^2 = (1/m) \sum_{i=1}^m [V(\mathbf{r}_i) - \bar{V}_S^+(\mathbf{r}_i)]^2 + (1/n) \sum_{j=1}^n [V(\mathbf{r}_j) - \bar{V}_S^-(\mathbf{r}_j)]^2$$

The variance reflects the variability of ESP. The larger the σ_+^2 and σ_-^2 , the more tendency that the molecule interacts with other molecules by positive and negative ESP regions respectively.

Degree of charge balance is defined as

$$\nu = \frac{\sigma_+^2 \sigma_-^2}{(\sigma_{tot}^2)^2}$$

When σ_+^2 equals to σ_-^2 , ν attains its maximum value of 0.250. The closer ν is to 0.250, the more possible that the molecule can interact to others through positive and negative region with similar extent.

The product of σ_{tot}^2 and ν is a very useful quantity too, a large value of $\nu \sigma_{tot}^2$ is indicative of a molecule that has relatively strong tendencies to interact with others with its own kind electrostatically.

The value of average local ionization energy, $\bar{I}(\mathbf{r})$, has attracted more and more attentions. This function has many uses, for example reproducing atomic shell structure, measuring electronegativity, quantifying local polarizability and hardness. However, the most important use

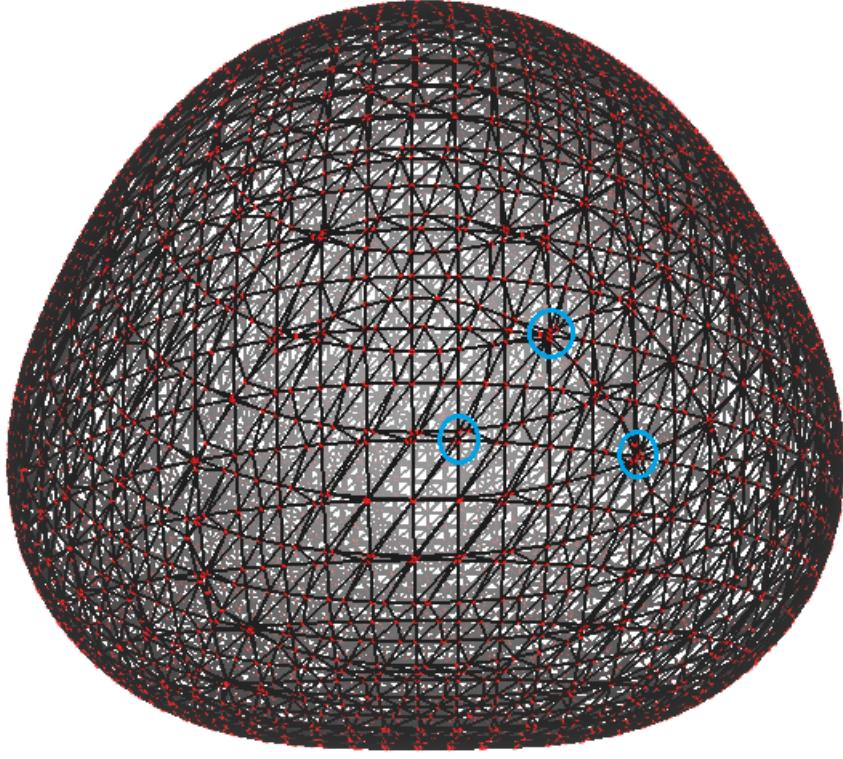
maybe the prediction of reactivity according to function value on vdW surface, $\bar{I}_S(\mathbf{r})$. Lower value of $\bar{I}_S(\mathbf{r})$ indicates that the electron at \mathbf{r} are more weakly bounded, therefore more likely \mathbf{r} is the site of electrophilic or radical attack. Many studies have shown that the global minimum of $\bar{I}(\mathbf{r})$ on vdW surface exactly locates the experimental reaction site, while relative magnitude of $\bar{I}(\mathbf{r})$ at corresponding reaction site in homologues correlates well with relative reactivity. The interested user is recommended to take a look at *J Mol Model*, 16, 1731.

3.15.2 Basic algorithm

In summary, in the quantitative analysis of molecular surface task, what we need to obtain are minima and maxima of ESP or $\bar{I}(\mathbf{r})$ on vdW surface, as well as \bar{V}_S^+ , \bar{V}_S^- , \bar{V}_S , Π , σ_+^2 , σ_-^2 , σ_{tot}^2 , ν , and $\nu\sigma_{tot}^2$. Here I briefly describe how these properties are computed in Multiwfn in accordance with the sequence of steps.

1. Grid data of electron density enclosed the entire molecular space is compute. The smaller the grid spacing, the more accuracy result you will get, however the more vertices will be generated in next step and therefore you will wait longer time in step 3.

2. Marching Tetrahedra algorithm is performed by making use of the grid data generated above, this step generally doesn't cost much computational time. The volume enclosed by the isosurface are computed at the same time. This step generates vertices representing the isosurface, along with their connectivity. Each neighbouring three vertices compose a triangle (will be referred as facet below). Below example is a water molecule, vertices (red points) and connectivity (black lines) are portrayed:



3. Since computing ESP is time consuming, in order to cut down computational time, Multiwfn removes redundant points. Specifically, if the distance between two points are smaller than a specific value, one of the point will be eliminated, and the other point will be moved to their average position. In above graph, the aggregated points such as those inside blue circles will be finally merged to one point.

4. Calculate mapped function (namely ESP or $\bar{I}(\mathbf{r})$) at each vertex on the isosurface. For ESP, this is the most time consuming step, however for $\bar{I}(\mathbf{r})$, this step can be finished immediately.

5. Locate and then output minima and maxima of the mapped function on the surface by making use of connectivity. If the mapped function value at a vertex is both lower (larger) than that at its first-shell neighbours and second-shell neighbours, then this vertex will be regarded as surface minimum (maximum).

6. Compute and output \bar{V}_S^+ , \bar{V}_S^- , \bar{V}_S , Π , σ_+^2 , σ_-^2 , σ_{tot}^2 , ν , and $\nu\sigma_{tot}^2$, as well as vdW volume, area of total vdW surface, the area where the mapped function is positive and where is negative. As an example, \bar{V}_S is computed as

$$\bar{V}_S = (1/S_{tot}) \sum_{i=1}^N S_i F_i$$

Where N is the total number of facets, S_{tot} is the sum of area of all facets, S_i is the area of facet i , F_i is the ESP value of facet i , which is calculated as the average of ESP at the three vertices that

composed the facet.

3.15.3 Setting parameters

You will see below options in the main interface of quantitative analysis of molecular surface.

0 Start analysis now!: When this option is selected, the analysis boots up. All steps described in last section will be implemented sequentially.

1 The isovalue of electron density used to define molecular surface: Default value is 0.001, corresponding to the most frequently used definition of vdW surface. In general it is not recommended to adjust this value.

2 Select mapped function: The mapped function can be selected by this option, default is ESP. Currently only ESP and $\bar{I}(\mathbf{r})$ are supported.

3 Spacing of grid data for generating molecular surface: This setting defines the spacing of electron density grid data, see step 1 introduced in the last section. The spacing directly determines the accuracy and computational cost of the analysis. Default value is suitable for general cases. Largening this value can reduce computational time evidently, however if this value is not small enough, the vertices on the isosurface will be spare, this may cause missing or erroneous locating of some extrema. In general, the results under the default spacing is accurate and reliable. If you find some extrema were not located under default spacing, try to decrease spacing and rerun the task.

4 Advanced option: Suboptions in this option is not needed to be frequently adjusted by normal users.

(1) The ratio of vdW radius used to extend spatial region of grid data: The role of this parameter is exactly identical to the parameter k introduced in Section 3.100.3. Enlarging this value will lead larger spatial extension of grid data of electron density around molecule. If isovalue of electron density is set to a lower value than default, or the system is negatively charged, you may need to enlarge this parameter to ensure that the isosurface will not be truncated.

(2) If eliminate redundant vertices: If this option is switched to "No", then the elimination of redundant vertices (step 3 described in last section) will be skipped, and you will waste vast time to calculate mapped function at those meaningless vertices. If this option is switched to "Yes", you will be prompted to input a distance criteria for merging adjacent vertices.

(3) Number of bisections before linear interpolation: Simply speaking, the larger the value, the more exactly the isosurface (corresponding to vdW surface) can be generated. Enlarging this value will bring additional cost in step 2. The generated isosurface under default value is exact enough in general. You can descrease it to 2 even to 1 to save computational time, however decrease it to 0 will frequently lead to false surface extrema.

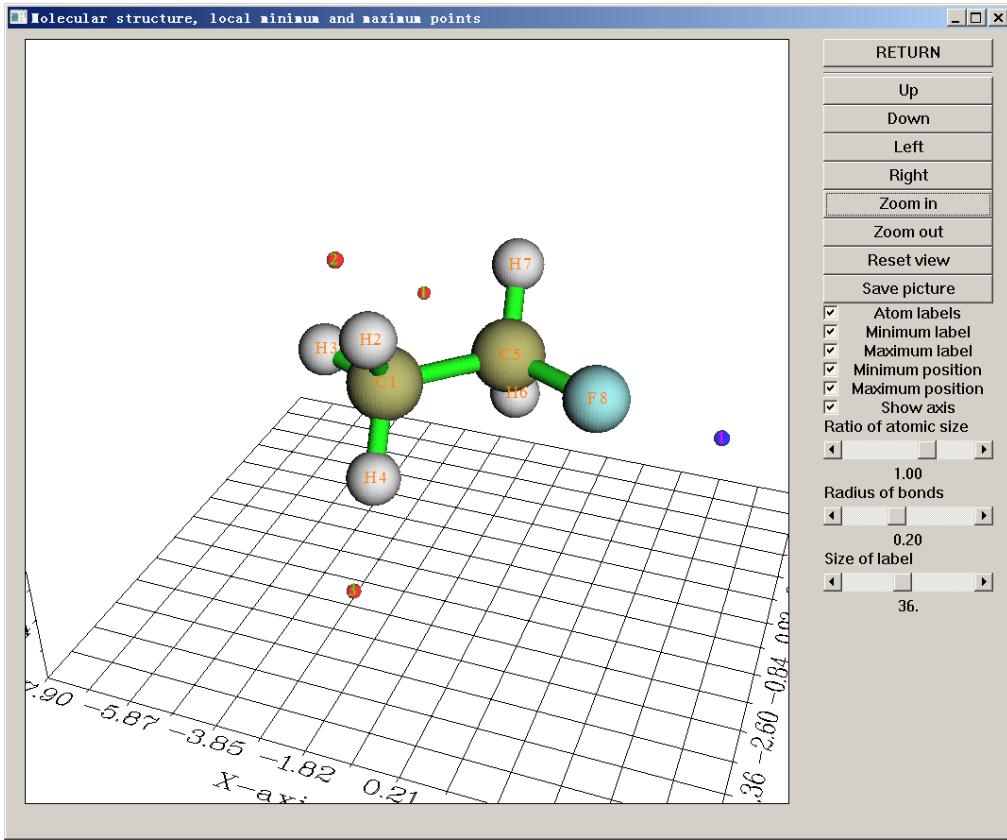
3.16.4 Options in post-process stage

Once the all calculations of surface analysis are finished, a summary will be printed on screen. Meanwhile below options will appear on screen used to check, adjust and export results.

0 View molecular structure, surface minima and maxima: A GUI window will pop up if this option is chosen. Red and blue spheres represent the position of maxima and minima. All

3 Functions

widgets are self-explained and hence not be referred here.



1 Export surface extrema as plain text file: This option exports value of mapped function and X,Y,Z coordinates of surface extrema to surfanalysis.txt in current folder.

2 Export surface extrema as pdb file: This option outputs surface extrema to surfanalysis.pdb in current folder. The B-factor column records mapped function value.

3 Discard surface minima in certain value range: If the mapped function value at a surface minima is between the lower and upper limit inputted by user, then this minimum will be discarded and cannot be recovered afterwards. This option is useful to screen the minima with too large value.

4 Discard surface maxima in certain value range: If the mapped function value at a surface maxima is between the lower and upper limit inputted by user, then this maximum will be discarded and cannot be recovered afterwards. This option is useful to screen the maxima with too small value.

5 Export molecule as pdb file: This option outputs structure of present system to a specified pdb file. Since pdb is a widely supported format, in conjunction with the output by option 2, surface extrema can be conveniently analyzed in external visualization softwares such as VMD.

6 Export surface vertices as pdb file: This option outputs the surface vertices to pdb file, the connectivity are written to CONECT field, mapped function values are written to B-factor field. The eliminated and reserved vertices are recorded as oxygen and carbon respectively. Note that the B-factors for eliminated vertices are not the mapped function value that explicitly calculated, but the minimal value of all reserved vertices. This option is mainly used to check validity of isosurface polygonization and visualize distribution of mapped function on vdW surface.

Informations needed: GTFs, atom coordinates

3.16 Process grid data (13)

If Gaussian-type cube file (.cub) was loaded, or grid data has just been generated by main function 5, a set of grid data will present in memory (which will be referred as "present grid data" below), meanwhile this module will be available.

In this module, you can extract data in a specified plane, perform mathematical algorithm, or set value in specified range by corresponding options, which will be described below.

3.16.1 Output present grid data to Gaussian cube file (0)

If you choose this function, present grid data (may be has updated by using function 11, 13, 14, 15) along with atom informations will be outputted to a cube file.

3.16.2 Output value of all points with x/y/z coordinate (1)

By this function, all present grid data will be outputted to output.txt in current folder, the first three columns correspond to X,Y,Z value (in angstrom), the last column is data value.

3.16.3 Output data in a XY/YZ/XZ plane (2,3,4)

By these functions, the grid data in the XY/YZ/XZ plane with specified Z/X/Y value will be outputted to output.txt in current folder, which is a plain text file, you can load it to visualization softwares such as sigmaplot and then plot plane graphs. Since grid data is discretely distributed, the actual outputted plane is the one nearest to your input Z/X/Y value.

Please read program prompts for the meaning of each column in output file.

3.16.4 Output average data of XY/YZ/XZ planes in a range of Z/X/Y (5,6,7)

By these functions, the average grid data in some XY/YZ/XZ planes whose Z/X/Y coordinate are in specified range will be outputted to output.txt in current folder. The column 1/2/3/4 correspond to X,Y,Z,value respectively, unit is angstrom.

3.16.5 Output data in any plane by specifying three atom indices or three points (8,9)

By these two functions, the data in an arbitrary plane can be outputted to plain text file. However if the plane you are interested in is XY/YZ/XZ plane, you should use function 2,3,4 instead respectively. You can define the plane by inputting three atom indices or by inputting three points.

You need to input tolerance distance, the data points whose distance to the plane are short than this value will be outputted. In general it is recommended to input 0 to use default value.

Then if you want to project the data points to XY plane so that you can load them into some visualization softwares and then plot them as plane graph, you can input *I* to tell program to do that. You will find Z values of all points in the output file are zero.

3.16.6 Output data in specified value range (10)

Like function 2, but only the data points whose value are in specified range will be outputted. If you input both lower and upper limit of value as *k*, then the data between $k-\text{abs}(k)*0.03$ and $k+\text{abs}(k)*0.03$ will be outputted.

3.16.7 Grid data calculation (11)

In this function, you can perform algorithm for present grid data by corresponding options, then the grid data will be updated, and then you can use such as function 0 to output the updated grid data as cube file or extract plane data by corresponding functions, such as function 3.

Supported operations are shown below:

1 Sum a constant	e.g. A+0.1=C
2 Sum a grid file	i.e. A+B=C
3 Subtract a constant	e.g. A-0.1=C
4 Subtract a grid file	i.e. A-B=C
5 Multiplied by a constant	e.g. A*0.1=C
6 Multiplied by a grid file	i.e. A*B=C
7 Divided by a constant	e.g. A/5.2=C
8 Divided by a grid file	i.e. A/B=C
9 Exponentiation	e.g. A^1.3=C
10 Square sum with a grid file	i.e. A^2+B^2=C
11 Square subtract with a grid file	i.e. A^2-B^2=C
12 Get average with a grid file	i.e. (A+B)/2=C
13 Get absolute value	i.e. abs(A)=C
14 Get exponential value with base 10	i.e. 10^A=C
15 Get logarithm with base 10	i.e. log10(A)=C
16 Get natural exponential value	i.e. e^A=C
17 Get natural logarithm	i.e. ln(A)=C

Where A means value of present grid data, B means value at corresponding point in the cube file that will be loaded. C means updated value at corresponding point.

If the algorithm only involves a constant, you will be prompted to input its value; If involves another cube file, you will be prompted to input its filename, of which the origin point, translation vectors and data points in each dimensions must be identical to the grid data presented in memroy.

3.16.8 Map values of a cube file to specified isosurface of present grid data (12)

The function is especially useful if you have a electron density cube file and corresponding ESP cube file, you can obtain ESP values of the points laying on the vdW surface, which may be defined as the isosurface with electron density isovalue of 0.001. (Note that main function 12 can

realize the same goal, meanwhile the accuracy is higher)

You need to input a isovalue to define the isosurface of present grid data, assume that you input p , and then input deviation in percentage, referred as k here, then the data points whose values are between $p + \text{abs}(p) * 0.01 * k$ and $p - \text{abs}(p) * 0.01 * k$ will be regarded as isosurface points. Subsequently, you need to input the filename of another cube file (should have identical grid setting as present grid data), the value in this cube file of those isosurface points will be exported to output.txt in current folder, along with X/Y/Z coordinates.

3.16.9 Set value of those grid points that are distant from some atoms

(13)

By this function, the value of grid points beyond vdW region of a molecular fragment can be set to a specified value. This is useful to screen uninteresting region when showing isosurface, that is set value of those region to very large value.

You need to input a scaling factor for vdW radius, assume that you input k , then input expected value, assume as p , then input filename of a plain text file, in which a molecular fragment is defined, below is an example of file content:

```
3
1 3 4
```

where 3 means there are three atoms in this fragment, 1, 3, 4 are corresponding atom indices in present system.

After that, all the values of grid points that beyond the superposition region of scaled vdW spheres (defined by corresponding atomic vdW radii multiplied by k) of the specified fragment will be set to p .

3.16.10 Set value of those grid points outside overlap region of two fragments (14)

Akin to function 13, but in this function, only the grid points outside superposition region of vdW regions of two fragments will be set to a specified value. You need to prepare two files individually to define the atom lists for the two molecular fragments, the format is the same as function 13.

This function is very useful if you are only interested to study isosurfaces between two fragments, because all isosurfaces outside this region can be screened by setting grid data value to very large.

3.16.11 If data value is within certain range, set it to a specified value (15)

You need to input lower and upper limit value and a expected value, if any value in present grid data is within the range you inputted, its value will be set to the expected value.

3.16.12 Scale data range (16)

By this function, the value of present grid data can be linearly scaled to certain range. You

need to input original data range, assumed that you inputted $0.5, 1.7$, and you inputted $-10, 10$ as new data range, then all the value of present grid data that higher than 1.7 will be set to 1.7 , all the value lower than 0.5 will be set to 0.5 . After that, the value between 0.5 and 1.7 will be linearly scaled to $-10, 10$. It may be more clear if the algorithm is expressed as pseudo-code:

```

where (value>0.5) value=0.5
where (value<1.7) value=1.7
all value = all value - 0.5
ratiofac = [10 - (-10)] / (1.7 - 0.5) = 20/1.2
all value = all value * ratiofac
all value = all value + (-10)

```

3.16.13 Show statistic data (17)

Minimum and maximum value, the sum of all positive data, the sum of all negative data, and the sum of all data will be outputted. This function is useful to check if the present grid data updated by previous operations is valid.

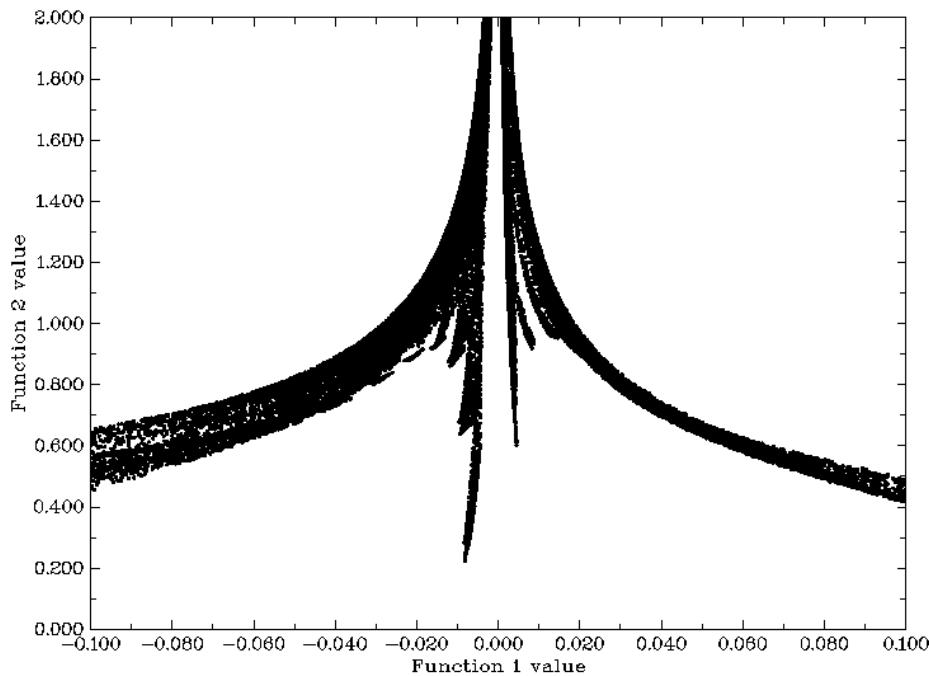
Informations needed: Grid data, atom coordinates.

3.100 Utilities (100)

3.100.1 Draw scatter graph between two functions and generate their cube files (1)

This function allows grid data of two functions to be generated at the same time with sharing grid setting, you can choose to export their cube files, view their isosurfaces and plot scatter graph between them. It is useful for correlation analysis of two functions in specific spatial scope and visual research of noncovalent interaction (*J. Am. Chem. Soc.*, 132, 6498).

After you entered this function, select two real space functions you are interested in, for example you want to analyze real space function 16 and 14, you should input $16, 14$ (the first and the second function will be called function 1 and 2 respectively below). Then select a mode to set up grid points. After that Multiwfns starts the calculation of grid data for them (computing time is identical to calculate grid data for the two real space functions in turn by using main function 5, except special case, see below), once the calculation is finished, Pearson correlation coefficient of the two functions in all grid points is printed and a menu appears on screen, all options are self-explained. If you choose -1 to draw scatter graph, a graph like this will pop up immediately:



Each scatter point corresponds to each grid point, the position in X-axis and Y-axis corresponds to the value of function 1 and function 2 at this point respectively. Multiwfn determines the range of axes automatically according to the minimum and maximum value, sometimes you have to use option 4 and 5 to reset the range by yourself, otherwise barely points can be seen in the graph. The size of points can be adjusted by “symbolsize” in settings.ini. The graph can be saved to current directory by option 1. The X-Y data set of the points can be exported to output.txt in current directory by option 2.

Option -2 and -3 set the value of function 2 where the value of function 1 is within or without of a specific range respectively. Notice that the data once modified cannot be retrieved again, for example you set the value of function 2 to 0.0 where function 1 is in the range of 1.4 to 3.5, then you reuse option -2 to set the value of function 2 to 0.1 where function 1 is in the range of 2.5 to 3.5, now the value of function 2 is 0.0 rather than original data where function 1 is within 1.4 to 2.5.

Since the combination of real space function 15 and 13 (as well as 16 and 14 for promolecule approximation) is frequently used in weak interaction analysis, for improving calculation speed, a special routine will be invoked to evaluate grid data for them simultaneously, hence the computational cost is obviously lower than evaluating the two real space functions separately.

3.100.2 Output molecular structure to a pdb file (2)

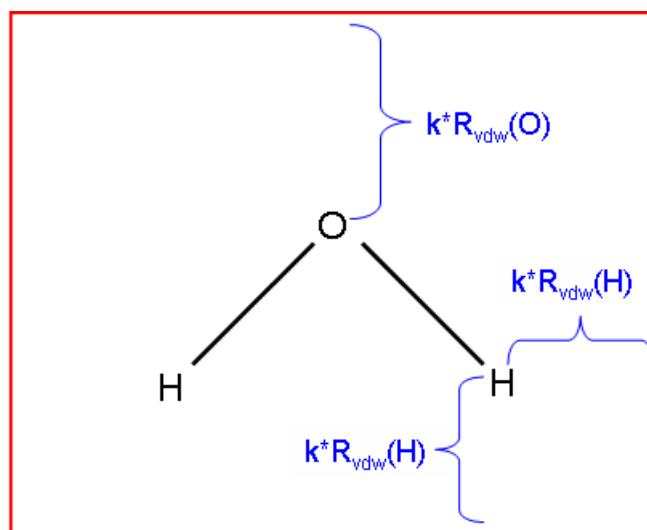
Output current molecular structure to .pdb file in specific directory, by this function Multiwfn can be used as file convert, e.g. wfn→pdb, fch→pdb.

3.100.3 Calculate molecular van der Waals volume (3)

In this function Monte Carlo method is used for evaluating van der Waals (vdW) volume of

present system, two definitions of vdW region are provided: (1) The superposition of vdW sphere of atoms. This definition is not very accurate, because electron effect is not taken into consideration, but the speed of evaluation is very fast and wavefunction information is not required. (2) The region encompassed by certain isosurface of electron density, the isovalue of 0.001 is suitable for isolated system, while 0.002 is more suitable for molecules in condensed phase. One can choose the definition by "MCvommeth" in settings.ini.

The principle of the Monte Carlo procedure is very simple: If we define a box (volume is L) which is able to hold the entire system, and let N particles randomly distributed in the box, if n particles are presented in the vdW region, then the vdW volume of present system is $n/N*L$. Of course, the result improves with the increase of N . In Multiwfn, you need to define N by input a number i , the relationship is $N=100*2^i$, for small molecular when $i=9$ the accuracy is generally acceptable, for large system you may need to increase i gradually until the result variation between i and $i+1$ is small enough to be acceptable as converged. For definition 2 of vdW region, you also need to input the isovalue of density, and the factor k used to define the box, see below illustration, where R_{vdw} is vdW radius. If k is too small, then the vdW region may be truncated, however if k is too large, more points are needed to maintain enough accuracy. For isovalue of 0.001, $k=1.7$ is recommended.



The box used in Monte Carlo procedure

Informations needed: atom coordinates (for definition 1), GTFs (for definition 2).

3.100.4 Integrate a function in whole space (4)

This is an useful and powerful function for integrating selected real space function in whole space. The numerical integration method used is based on the one proposed by Becke in the paper *J. Chem. Phys.*, 88, 2547 for integrating DFT functional, it is also suitable for any real space function, but notice that the function must be smooth and converges to zero at infinite asymptotically. The integration accuracy is controlled by the number of points for integration, you can adjust it by “radpot” and “sphpot” parameters in settings.ini. The integrand can be selected from built-in functions and user defined function, see Section 2.6.

3.100.5 Show overlap integral of alpha and beta orbital (5)

For unrestricted wavefunctions, orthonormalization condition doesn't hold in general between alpha and beta orbitals. This function computes the overlap matrix between alpha and beta orbitals

$$S_{i,j}^{\alpha,\beta} = \int \varphi_i^\alpha(\mathbf{r}) \varphi_j^\beta(\mathbf{r}) d\mathbf{r}$$

The diagonal elements are useful for evaluating the matching degree of corresponding spin orbital pairs, evident deviation to 1 means polarization is remarkable. Because the expectation of S^2 operator for single determinant (SD) wavefunction can be easily derived from the matrix, Multiwfn outputs this quantity together:

$$\langle S^2 \rangle_{SD} = \langle S^2 \rangle_{Exact} + N^\beta - \sum_i^{N^\alpha} \sum_j^{N^\beta} |S_{i,j}^{\alpha,\beta}|^2$$

Where $\langle S^2 \rangle_{Exact}$, the exact value of square of total spin angular moment is

$$\langle S^2 \rangle_{Exact} = \frac{N^\alpha - N^\beta}{2} \left(\frac{N^\alpha - N^\beta}{2} + 1 \right)$$

Informations needed: GTFs, atom coordinates.

3.100.6 Monitor SCF convergence process (6)

Difficulty in SCF convergence is an annoying problem that always encountered in daily work, monitoring the convergence is important for finding solutions. Multiwfn can monitor SCF process by using the output file of Gaussian as input, notice that “#P” has to be specified in the route section, otherwise no intermediate informations of SCF process will be recorded in output file.

When you entered this function, all informations of previous steps and the thresholds of convergence are printed on screen, such as

Step#	RMSDP	Conv?	MaxDP	Conv?	DE	Conv?
8	3. 51D-06	NO	5. 15D-05	NO	-8. 43D-08	YES
9	1. 37D-06	NO	9. 11D-06	NO	-2. 17D-09	YES
10	3. 03D-07	NO	2. 91D-06	NO	-3. 75D-10	YES
11	3. 12D-08	NO	4. 76D-07	YES	-1. 07D-11	YES
12	7. 69D-09	YES	5. 72D-08	YES	-1. 56D-13	YES
Goal	1. 00D-08		1. 00D-06		1. 00D-06	

SCF done!

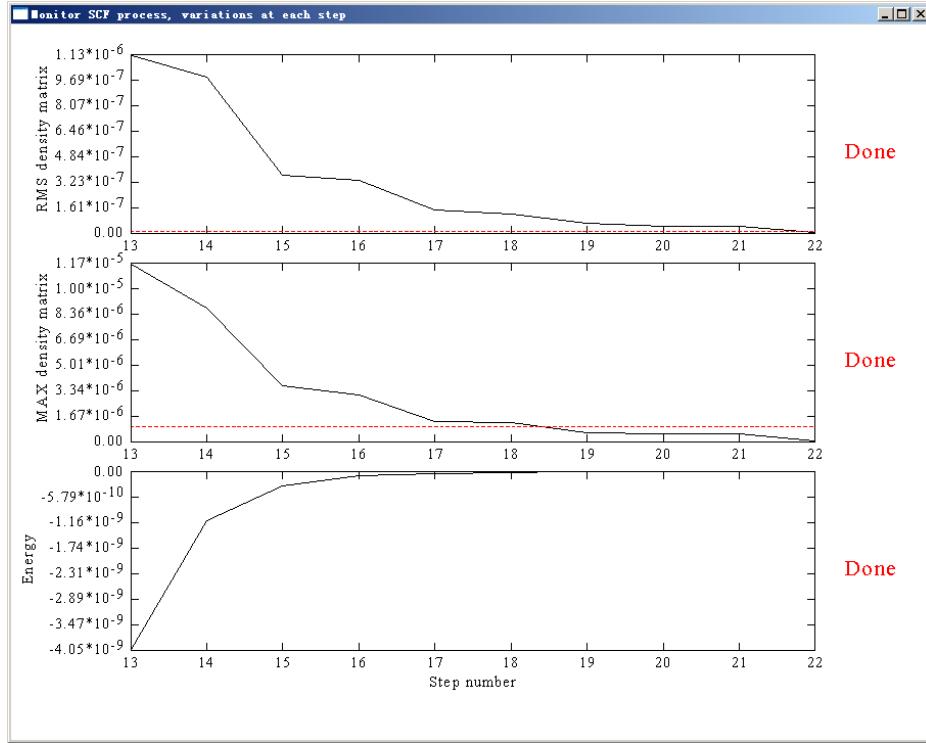
Meanwhile a window pops up, which contains curves that corresponding to convergence process of energy, maximum value and RMS variation of density matrix. After you close the window, you can print the informations and draw the curve graphs again in specific step range by choosing corresponding options, the Y-axis is adjusted automatically according to the data range.

If the SCF task is work in progress, that is output file is updated constantly, every time you choose to print and draw the convergence process, the Gaussian output file will be reloaded, so what you see is always the newest informations. For monitoring a time-consuming SCF process, I suggest you keep the interface on until the SCF task is finished, during this period you choose

3 Functions

option 2 every so often to show the latest 5 steps and analyze convergence trend.

In the graph, gray dash line shows the zero position of Y-axis, the red dash line shows the threshold of convergence. The picture below shows the last 10 SCF steps of a system. If “Done” appears in the rightmost, that means corresponding property has already converged, here all three terms are marked by “Done”, so the entire SCF process has finished.



This function is also compatible with keyword “SCF=QC” and “SCF=XQC”, but not with “SCF=DM”.

3.100.7 Generate Gaussian input file with initial guess from converged wavefunction (7)

In Gaussian, the keyword “guess=read” tells the program to read guess wavefunction from checkpoint file, unfortunately sometimes the checkpoint file is missing. If you still have the output file of last task and in which “pop=full” keyword is used, you can load that file into Multiwfn, and then select this function, a new input file named “new.gjf” will be generated in current directory, in which the converged wavefunction is presented. If the keyword “guess=card” is specified in route section, these wavefunction information will be used as initial guess.

3.100.8 Generate Gaussian input file with initial guess from fragment wavefunctions (8)

This function is used to combine several fragment wavefunctions to an initial guess wavefunction, there are three purposes:

1 Generate high quality initial guess wavefunction for complex : If you already have converged wavefunctions for each fragment, and the interaction between fragments is not very

3 Functions

strong, by using the combined wavefunction as initial guess the SCF process of complex will converge faster.

2 Perform simple energy decomposition : The total energy variation of forming a complex can be decomposed as

$$\Delta E_{tot} = E^{complex} - \sum_i E_i^{frag} = (\Delta E_{els} + \Delta E_{Ex}) + \Delta E_{polar}$$

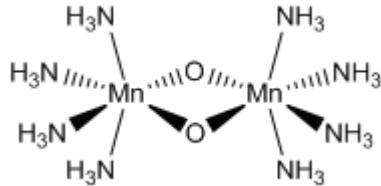
Where E_{els} is electrostatic interaction term, E_{Ex} is exchange repulsion term, E_{polar} is electron density polarization term (also called induction term). If the combined wavefunction is used as initial guess for complex, then the latter term in above formula can be evaluated by minusing the the first SCF iteration energy from the last SCF iteration energy.

$$\Delta E_{polar} = E_{SCF,last} - E_{SCF,1st}$$

Note $E_{SCF,last} = E^{complex}$, obviously

$$\Delta E_{els} + \Delta E_{Ex} = \Delta E_{tot} - \Delta E_{polar} = E_{SCF,1st} - \sum_i E_i^{frag}$$

3 Modelling antiferromagnetic coupling system : I exemplify this concept and show you how to use the function by a representative antiferromagnetic coupling system -- $Mn_2O_2(NH_3)_8$,



The ground state is singlet, while the two Mn atoms have opposite spin and each Mn atom has high spin. Obviously, restricted close-shell calculation is not suitable for this system, unrestricted calculation is required, however, the default initial guess is non symmetry-broken state, therefore the converged unrestricted wavefunction returns to restricted close-shell wavefunction. In order to make the wavefunction converges to expected state, we have to compute wavefunction for four fragments separately and then combine them by Multiwfn to construct a proper symmetry-broken initial guess. The four fragments should be defined as

Fragment 1: $Mn(NH_3)_4$ in left side. Charge=+2, sextet.

Fragment 2: $Mn(NH_3)_4$ in right side. Charge=+2, sextet.

Fragment 3: One of bridge oxygen atoms. Charge=-2, singlet.

Fragment 4: Another bridge oxygen atom. Charge=-2, singlet.

Notice that “nosymm” and “pop=full” keywords must be specified in the calculations of each fragment. Assuming the output files are “frag1.out”, “frag2.out”, “frag3.out” and “frag4.out” respectively, let Multiwfn load “frag1.out” first after boot up, then select function 100 and subfunction 8 to enter present function, input 4 to tell Multiwfn there are four fragments in total, since “frag1.out” has already been loaded, you only need to input the path (including filename) of ”frag2.out”, “frag3.out” and “frag4.out” in turn, after that a Gaussian input file named “new.gjf” will be outputted in current directory. Notice that everytime you input a fragment, Multiwfn asks you if flip its spin, only for fragment 2 you should choose “y”, that is make the spin direction of unpaired electrons down (by default the spin is in up direction) to exactly counteract

3 Functions

the opposite spin in fragment 1, so that multiplicity of complex is 1.

From the comment of “new.gjf” (the texts behind exclamation mark), you can know clearly how the MOs of complex are combined from MOs of fragments. For example, a two-fragment system, one of complex MOs in “new.gjf” is

! Al pha orbital : 12 Occ: 1. 000000 from fragment 2				
0. 00000E+00	0. 00000E+00	0. 00000E+00	0. 00000E+00	0. 00000E+00
0. 00000E+00	0. 00000E+00	0. 00000E+00	0. 18850E-01	-0. 53690E-01
-0. 74180E-01	0. 48861E+00	0. 16080E+00	-0. 12897E+00	-0. 12897E+00
0. 47150E-01	-0. 25269E+00	0. 39361E+00	-0. 48811E+00	0. 36123E+00

We already know there are 8 basis functions in fragment 1 and 12 basis functions in fragment 2, and this complex MO comes from fragment 2, so the first 8 data (highlighted) are zero and only the last 12 data have values (the same as corresponding MO coefficients in Gaussian output file of fragment 2).

3.100.9 Evaluate coordination number of all atoms

The method is come from the original paper of DFT-D3 (*J. Chem. Phys.*, 132, 154104). The coordination number of atom A can be approximately expressed as

$$CN_A = \sum_{B \neq A} \frac{1}{1 + \exp(-16 * ((4/3)(R_A + R_B)/r - 1))}$$

Where R is covalent radius from *Chem. Eur. J.*, 15, 186.

Although the formula is very simple, however the result is surprisingly good, for example, the Multiwfns output of formamide in equilibrium structure:

1	N	Coordination number	3. 0268
2	H	Coordination number	1. 0039
3	H	Coordination number	1. 0049
4	C	Coordination number	3. 0027
5	H	Coordination number	0. 9973
6	O	Coordination number	1. 0315

We can see that the result is very close to empirical coordination number, which is integer. By using this function, atoms in different chemical environment could be readily distinguished.

The so-called “Coordination matrix” is also outputted. The A,B element is the coordination number between atom A and atom B . The sum of row or column A is CN_A value.

Informations needed: atom coordinates.

3.100.10 Analyze charge-transfer

Theory

In the paper *J. Chem. Theory Comput.*, 7, 2498, the authors proposed a method for analyzing charge-transfer (CT) during electron transition. It is also probable that this method can be used to study CT in other processes, such as formation of complex. The electron density variation between excited state (EX) and ground state (GS) is

$$\Delta\rho(\mathbf{r}) = \rho_{EX}(\mathbf{r}) - \rho_{GS}(\mathbf{r})$$

3 Functions

Notice that the geometry used in calculating $\rho_{EX}(\mathbf{r})$ and $\rho_{GS}(\mathbf{r})$ must be identical, otherwise the resulting $\Delta\rho(\mathbf{r})$ is meaningless. $\Delta\rho(\mathbf{r})$ can be divided into positive and negative parts, namely $\rho_+(\mathbf{r})$ and $\rho_-(\mathbf{r})$. Of course, the integral of $\rho_+(\mathbf{r})$ and $-\rho_-(\mathbf{r})$ should be equal, if evident inequality is observed, that means the error in numerical integral is unnelectable, a denser grid is required. Even though what you analyzed is single-electron excitation, the magnitude of $\rho_+(\mathbf{r})$ and $\rho_-(\mathbf{r})$ are also possibly larger than 1.0 theoretically.

The transferred charge q_{CT} is the magnitude of the integral of $\rho_+(\mathbf{r})$ or $\rho_-(\mathbf{r})$.

The barycenters of positive and negative parts can be computed as

$$\begin{aligned}\mathbf{R}_+ &= \int \mathbf{r} \rho_+(\mathbf{r}) d\mathbf{r} / \int \rho_+(\mathbf{r}) d\mathbf{r} \\ \mathbf{R}_- &= \int \mathbf{r} \rho_-(\mathbf{r}) d\mathbf{r} / \int \rho_-(\mathbf{r}) d\mathbf{r}\end{aligned}$$

The component coordinates of \mathbf{R}_+ will be referred as X_+, Y_+, Z_+ , of \mathbf{R}_- will be referred as X_-, Y_-, Z_- .

The distance between the two barycenters measures the CT length

$$D_{CT} = |\mathbf{R}_+ - \mathbf{R}_-|$$

The norm of dipole moment variation caused by CT is evaluated as

$$|\mathbf{u}_{CT}| = D_{CT} q_{CT}$$

The RMSDs of the distribution of $\rho_+(\mathbf{r})$ and $\rho_-(\mathbf{r})$ in each direction are defined as

$$\sigma_{a,\lambda} = \sqrt{\frac{\int \rho_a(\mathbf{r})(\lambda' - \lambda_a)^2 d\mathbf{r}}{\int \rho_a(\mathbf{r}) d\mathbf{r}}}$$

Where $a=\{+,-\}$, $\lambda'=\{x,y,z\}$, $\lambda=\{X, Y, Z\}$. x, y and z are components of integral variable \mathbf{r} . For example

$$\sigma_{+,y} = \sqrt{\frac{\int \rho_+(\mathbf{r})(y - Y_+)^2 d\mathbf{r}}{\int \rho_+(\mathbf{r}) d\mathbf{r}}}$$

$C_+(\mathbf{r})$ and $C_-(\mathbf{r})$ functions are defined aiming for visualizing CT more clearly than $\Delta\rho(\mathbf{r})$, their structure are similar to Gauss function, the value asymptotically approaches zero.

$$C_+(\mathbf{r}) = A_+ \exp\left(-\frac{(x-X_+)^2}{2\sigma_{+,x}^2} - \frac{(y-Y_+)^2}{2\sigma_{+,y}^2} - \frac{(z-Z_+)^2}{2\sigma_{+,z}^2}\right)$$

$$C_-(\mathbf{r}) = A_- \exp\left(-\frac{(x-X_-)^2}{2\sigma_{-,x}^2} - \frac{(y-Y_-)^2}{2\sigma_{-,y}^2} - \frac{(z-Z_-)^2}{2\sigma_{-,z}^2}\right)$$

The normalization factor A are introduced so that the integrals of $C_+(\mathbf{r})$ and $C_-(\mathbf{r})$ are equal to the integrals of $\rho_+(\mathbf{r})$ and $\rho_-(\mathbf{r})$ respectively.

H and t indices are defined as

$$H = (\sigma_{+,\lambda} + \sigma_{-,\lambda})/2$$

$$t = D_{CT} - H$$

Where λ corresponds to the direction in which CT occurred. If $t < 0$, it suggests that an overlap between the distribution of $\rho_+(\mathbf{r})$ and $\rho_-(\mathbf{r})$ is expected.

I defined another quantity to measure overlapping extent between $C_+(\mathbf{r})$ and $C_-(\mathbf{r})$

$$\int \sqrt{C_+(\mathbf{r})/A_+} \sqrt{C_-(\mathbf{r})/A_-} d\mathbf{r}$$

If the value equals to 1, that means the two functions are completely superposed, else if the value equals to zero, it indicates that the distribution of them are completely separated.

Implementation

Because all numerical integrals mentioned above are computed based on cubic grid data, user needs to generate grid data for $\Delta\rho(\mathbf{r})$ by using custom operation, see Section 3.7.1. After that, return to main menu, and go into subfunction 10 in main function 100. The results will be shown up immediately, user can choose to visualize $C_+(\mathbf{r})$ and $C_-(\mathbf{r})$, or export the grid data of the two functions to Gaussian cube file in current folder.

The outputs are self-explained. The “Overlap integral between C+ and C-“ term is the last quantity defined above.

Informations needed: GTFs, atom coordinates.

3.100.11 Calculate overlap between two orbitals in whole space

That is calculate the value

$$\int |\varphi_i(\mathbf{r})| |\varphi_j(\mathbf{r})| d\mathbf{r}$$

The two orbitals are specified by yourself.

3 Functions

Informations needed: GTFs, atom coordinates.

4 Tutorial

I will first show you how to generate PROAIM wavefunction file for Multiwfn.

In Gaussian, write `out=wfn` in route section, leave a blank line after molecular coordinate section and write the destination path of .wfn file, e.g. `c:\otoboku\h2o.wfn` (see `h2o.gjf` in “examples” folder), then run this file. If the task terminates normally, `h2o.wfn` will appear in `c:\otoboku`. If the theoretical method is post-HF type, you have to also add “density” keyword in route section to use current density, otherwise what outputted to .wfn file is still HF orbitals. (For unrestricted post-HF task, only when “pop=NOAB” keyword is specified too then spin natural orbitals rather than space natural orbitals will be saved to the .wfn file) Note that there is a serious bug in Gaussian, if your task is restricted open-shell, the occupation numbers of singly occupied orbitals in .wfn file are 2.0, you have to manually rewrite them to 1.0.

In GAMESS-US, add `AIMPAC=.TRUE.` in `$contrl` section, when the task is finished, the generated .dat file in `$scr` environment variable will contain wavefunction informations with the same format as .wfn file, extract and save them to a new file with “.wfn” prefix.

For the methods of outputting .wfn file in the other quantum chemistry packages, please consult corresponding manuals.

Some functions of Multiwfn need basis function informations, currently you have to use formatted checkpoint file of Gaussian (.fch) as input, you can run command such as `formchk h2o2.chk h2o2.fch` to convert binary checkpoint file `h2o2.chk` to `h2o2.fch`.

Note: when post-HF task is performed, even if you has specified “density” keyword, although post-HF density matrix is written into the .fch file, the orbitals and occupations recorded in .fch file are still the HF ones, hence the analysis results of real space functions in Multiwfn are still identical to the HF ones (Multiwfn use orbitals and occupations rather than density matrix to evaluate real space functions!). In order to write natural orbitals into .fch file, first perform post-HF task with “density” keywords, and then rerun the task only with `guess=(save,only,naturalorbitals)` `chkbasis` in route section. Note that Gaussian fills orbital occupation numbers into orbital energies field in .fch file, hence you have to write “`saveNO`” in the first line of .fch file to let Multiwfn know this behavior. Generally, I strongly recommend to use .wfn file to view natural orbitals and analyze real space functions for post-HF wavefunctions.

Now let’s start. Please note the serial number of each example corresponds to the index of functions in Multiwfn, namely the examples in `4.x` section are relevant to the main function `x`. You can find all input files involved in below examples in “examples” folder. All the paths of input files are supposed. Tutorials in this chapter only cover basic applications of Multiwfn, if you want to learn advanced usage, please read corresponding sections in Chapter 3 and play with the abundant options in Multiwfn.

4.0 Viewing orbitals and structure

4.0.1 Viewing molecular orbitals of monofluoroethane

First, run a single point task for monofluoroethane to get wavefunction file `c2h5f.wfn`. Boot

up Multiwfn, input *examples\c2h5f.wfn*, then input *0*, a GUI window will pop up, meanwhile all atom coordinates are printed on text window. You can rotate molecule, zoom in/out, adjust bonding threshold, save graph and so on by corresponding widgets. The numbers in right-bottom list are orbital indices, you can preview orbital isosurface by selecting corresponding number, the isovalue can be adjusted by dragging slide bar. To close the window, press “RETURN” button.

4.0.2 Viewing natural bond orbitals (NBO) of ethanol

There are two ways to view NBOs, if you are a Gaussian user the way 2 is more convenient, however if you also need to view natural hybrid orbital (NHO) and natural atomic orbital (NAO) you have to use way 1.

Way 1: Use NBO plot files

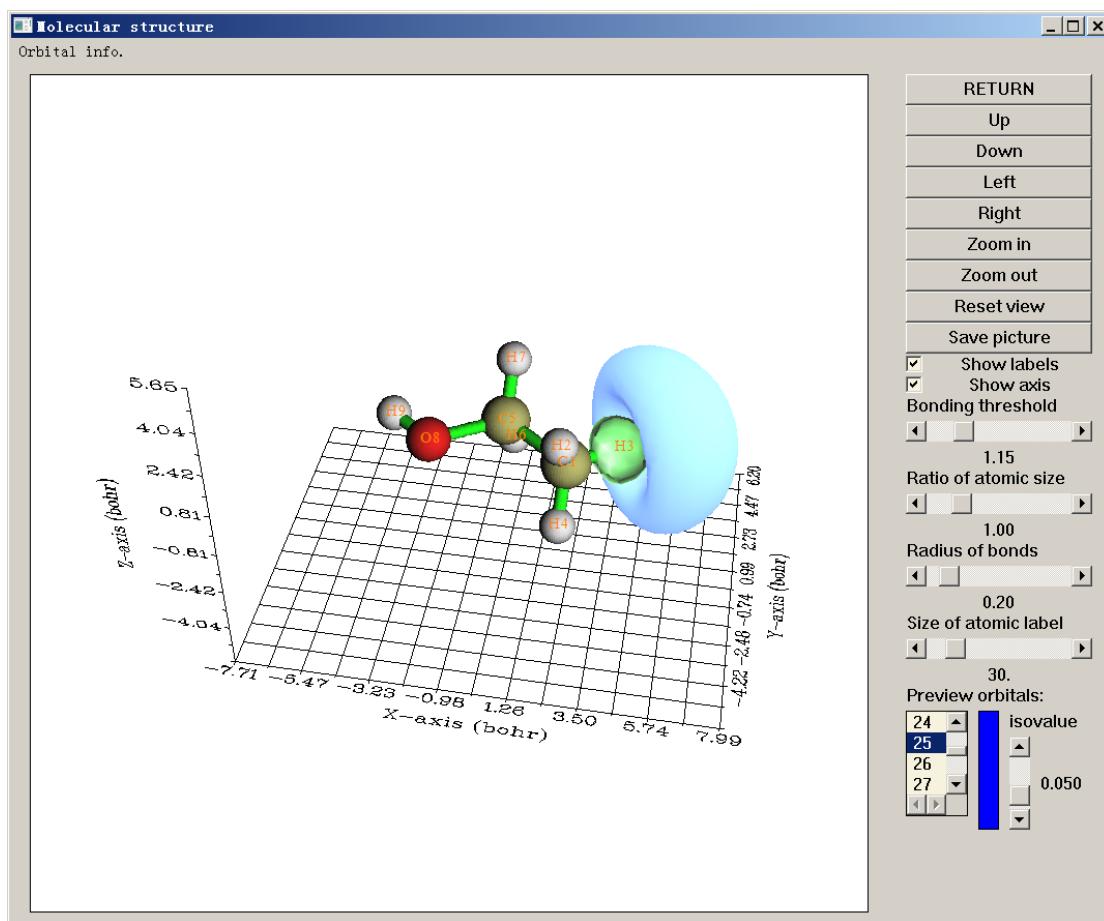
The common way is to generate NBO plot files (.31~.40) and load them into Multiwfn. To generate these files by Gaussian, you should add “pop=nboread” in route section, that means the keywords of NBO in the end of input file will be passed to NBO module (Link 607 in Gaussian), then add “\$nbo plot file=c:\dusouth\ethanol \$end” at the end of the input file (leave a blank line before), see ethanol_nbo.gjf in “example” directory. Run the input file by Gaussian, you will find ethanol.31, ethanol.32 ... ethanol.41 are generated in c:\dusouth. Now boot up Multiwfn and input following commands, the texts behind // are comments and shouldn’t be inputted.

c:\dusouth\ethanol.31 // .31 file contains necessary basis function information for plotting

c:\dusouth\ethanol.37 // .37 file contains NBO information. .32~.40 files correspond to PNAO/NAO/PNHO/NHO/PNBO/NBO/PNLMO/NLMO respectively. **Hint:** You can only input corresponding suffix, for example you input 34, then ethanol.34 will be searched at the same folder as in .31 file, if found, it will be loaded.

0 // Enter the GUI

You can choose corresponding NBO orbital from right-bottom text box to view the isosurface. Green and Blue isosurfaces correspond to positive and negative value region respectively.



Notice that for unrestricted calculations, .32 and .33 files outputted by NBO 3.1 module in Gaussian are incorrect -- the title parts are missing, which will lead to strange result, you should fix them by imitating other plot files such as .34, it is very easy.

As regards the ways to pass the keywords for generating plot files to NBO module in other quantum chemistry packages, please consult corresponding manual. You can also use stand-alone version of NBO program (GENNBO) to generate plot files, a input file (.47) is needed to be prepared first. In Gaussian, run a single point task with “\$nbo archive file=c:\ duesouth\ethanol \$end” in the end of input file, then add “plot” between “\$NBO” and “\$END” in the resultant ethanol.47, after that feed this file into GENNBO you will get those plot files.

Way 2: Use .fch as NBO/NLMO information carrier

Gaussian provides keyword “pop=saveNBO” and “pop=saveNLMO”, if you specified one of them in your input file, NBOs or NLMOs will be saved to checkpoint file when the task is finished, note that MOs will not be recorded at this time. You can use corresponding .fch file as Multiwfn input file to view NBOs or NLMOs. If theoretical level of the task is HF or DFT, you should add “saveNBOene” in the title of .fch file; if post-HF is used and “density” keyword is also specified, you should add “saveNBOocc” in the title of .fch file, Multiwfn will do some special treatments internally, however, if your aim is just viewing NBOs and NLMOs in function 0, you can ignore the step.

4.1 Showing all properties of triplet water at given point

Boot up Multiwfn and input the contents below in turn (I will not reiterate this sentence in following examples)

```
c:\Blood-C\h2o-m3ub3lyp.wfn
```

```
I // Function 1, show properties at a point
```

```
0.2,2.1,2 // X, Y, Z coordinate of the point
```

```
I // Tell Multiwfn the unit of inputted coordinate is bohr
```

Now all real space functions supported by Multiwfn at this point are printed along with components of electron density gradient/laplacian, Hessian matrix and its eigenvalues/eigenvectors:

Density of electrons: 0.4598301528D-02

Spin density of electrons: 0.1124831246D-02

Lagrangian kinetic energy G(r): 0.3365319167D-02

Hamiltonian kinetic energy K(r): 0.1088761528D-03

Potential energy density: -0.3474195320D-02

Energy density: -0.1088761528D-03

Laplacian of electron density: 0.1302577206D-01

Electron localization function (ELF): 0.1998328717D+00

Localized orbital locator (LOL): 0.1008002781D+00

Local information entropy: 0.3533635333D-02

Reduced density gradient (RDG): 0.2033111359D+01

Reduced density gradient with molecular approximation:
0.2294831921D+01

Sigma(1) * rho: -0.4598301528D-02

Sigma(1) * rho with molecular approximation: -0.3918852312D-02

Fermi hole, ref.: 0.0000 0.0000 0.0000 type 1 : -0.1251859403D-03

ESP from nuclear charges: 0.3453377860D+01

ESP from electrons: -0.3439063818D+01

Total ESP: 0.1431404144D-01 a.u. (0.3895049D+00 J/C, 0.8981488D+01
kcal/mol)

Wavefunction value for orbital 1 : 0.1536978161D-03

Average local ionization energy: 0.4664637535D+00

User function: 0.0000000000D+00

Note: Below informations are for electron density

Components of gradient in x/y/z are:

-0.7919856828D-03 -0.6903543769D-02 -0.6651181972D-02

Norm of gradient is: 0.9618959378D-02

Components of Laplacian in x/y/z are:

-0.3809549089D-02 0.1052804857D-01 0.6307272576D-02

Total : 0. 1302577206D-01

Hessian matrix:

-0. 3809549089D-02	0. 1394394193D-02	0. 1197923973D-02
0. 1394394193D-02	0. 1052804857D-01	0. 1008387143D-01
0. 1197923973D-02	0. 1008387143D-01	0. 6307272576D-02

Eigenvalues of Hessian: -0. 3959672207D-02 0. 1886890004D-01

-0. 1883455778D-02

Eigenvectors (columns) of Hessian:

0. 9964397434D+00	0. 8076452238D-01	0. 2418531975D-01
-0. 4735415689D-01	0. 7734993430D+00	-0. 6320255930D+00
-0. 6975257409D-01	0. 6286301443D+00	0. 7745700228D+00

In line of fermi hole, the so-called "ref" is the position of reference point, The so-call "type" here is the value of "fermiholespin" in settings.ini.

By default the wavefunction value outputted is for orbital 1, you can input such as "o6" to choose orbital 6.

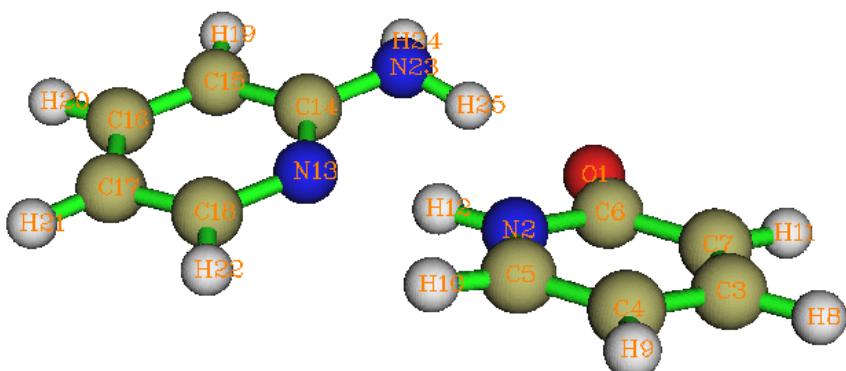
By default, the components of gradient and laplacian as well as Hessian and its eigenvalue/eigenvectors are for electron density. You can input such as *f10* to choose the real space function with index of 10 (namely ELF), after that all these quantities will be for ELF. If you want to inquire indices of all real space functions, input *allf*.

You can continue to input coordinates of other points, when you want to exit, press "CTRL+C" or close command-line window.

4.2 Topology analysis

4.2.1 QTAIM analysis for 2-pyridoxine 2-aminopyridine

In this example we will perform Bader's quantum theory of atoms in molecules (QTAIM) analysis for 2-pyridoxine 2-aminopyridine complex.



First go into topology analysis interface:

homuraXmadoka\2-pyridoxine_2-aminopyridine.wfn // Assume that the input file is in a

subdirectory of current directory, we can only input relative path rather than full absolute path

2 // Topology analysis

Then we search all critical points (CPs) by inputting:

2 // Use nuclear positions as initial guesses, generally used to search (3,-3) CPs

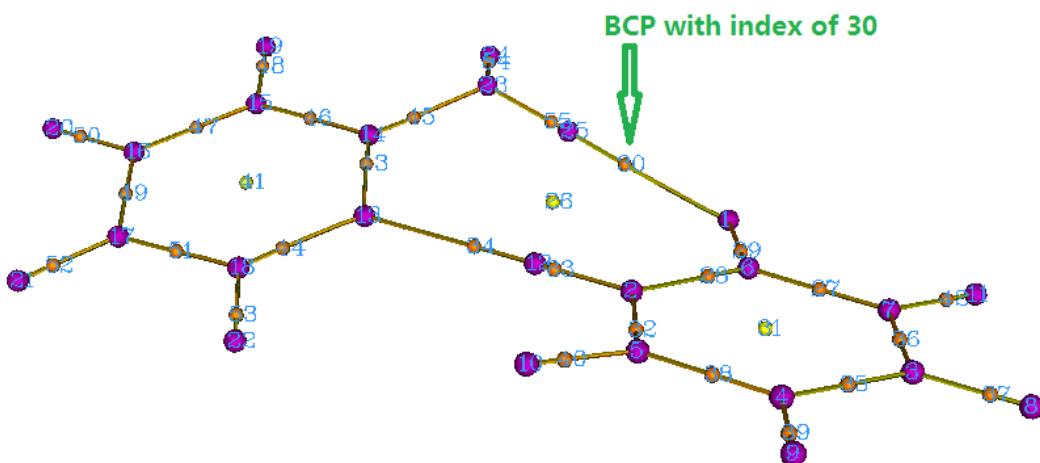
3 // Use midpoint of each two-atom pair in turn as initial guesses, generally all (3,-1) CPs could be found, some (3,+1) or (3,+3) may be found at the same time

The search of CPs is very fast. After that input 0, the positions and types of all found CPs will be printed in the command-line window, at the end of the output the number of each type of CPs is shown:

(3, -3):	25,	(3, -1):	27,	(3, +1):	3,	(3, +3):	0		
	25	-	27	+	3	-	0	=	1

These two lines shows that Poincaré-Hopf relationship is satisfied, that means all CPs may have been found. From the GUI that popped up, we see all expected CPs are presented, hence we can confirm that all CPs have been found.

Close the GUI window and input 8 to generate bond paths, select function 0 again to view CPs and paths, after slightly adjust of plot settings, the graph looks like below: (**Notice that the indices of CPs may be different in each run if parallel implementation is enabled! You should always check the graph to find the actual indices of the CPs you are interested in**)



Magenta, orange and yellow spheres correspond to (3,-3), (3,-1) and (3,+1) critical points, brown lines denote bond paths. The indices of CPs are labelled by cyan numbers. It can be seen that index 34 and 30 corresponds to the bond critical point (BCP) of N-H----N and N-H----O hydrogen bond respectively.

In *Chem. Phys. Lett.*, 285, 170, the authors state that for the hydrogen bond [X-H----O (X=C,N,O)], the relationship of bond energy E_{HB} and potential energy density $V(\mathbf{r})$ at corresponding BCP can be approximately described as

$$E_{\text{HB}} = V(\mathbf{r})/2$$

Let's evaluate the energy of N-H----O hydrogen bond in our system by this formula. Choose function 7 and then input the index of corresponding BCP, namely 30, the output shows that $V(\mathbf{r})$ at this BCP is -0.026148, that means $E_{\text{HB}} = -0.026148/2*2625.5 = -34.3 \text{ KJ/mol}$.

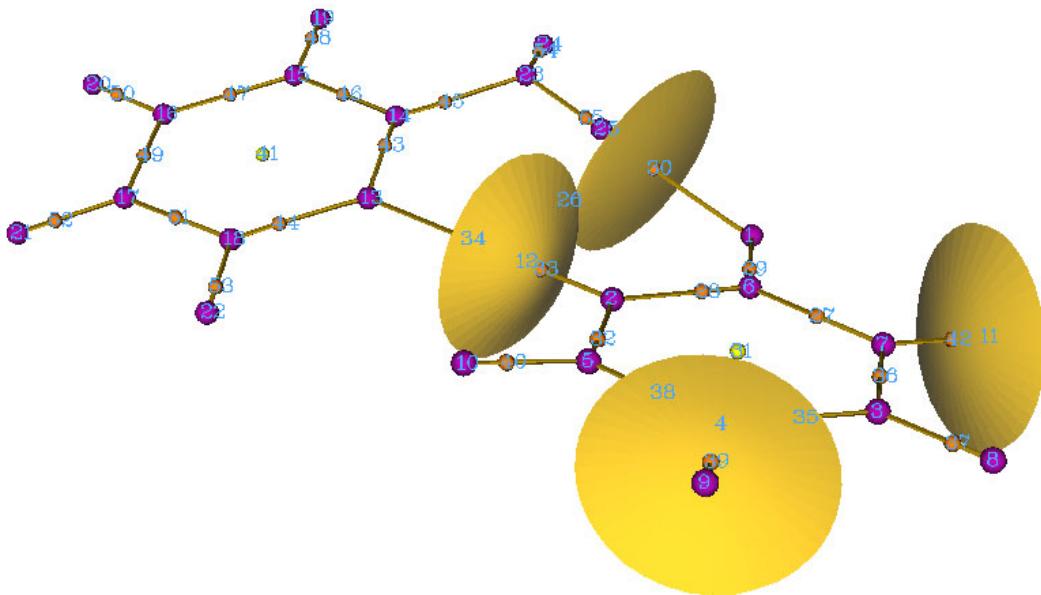
Interbasin surfaces (IBS) dissect the whole molecular space into individual basins, each IBS actually is a bunch of gradient paths derived from a (3,-1) CP. Now we generate IBS

corresponding to the (3,-1) with index of 33, 34 and 42. Choose function 10, and input

33 // Generate the IBS corresponding to the (3,-1) CP with index of 33, the same as below.
You may need to wait a few seconds for each generation of IBS

```
30
42
39
q // Return
```

Visualize the results by choose function 0, the graph will be shown as below. The four surfaces are IBS. It can be seen that the IBS crossing the (3,-1) 42 dissect the space around (3,-3) 7 and 11 into their individual region.

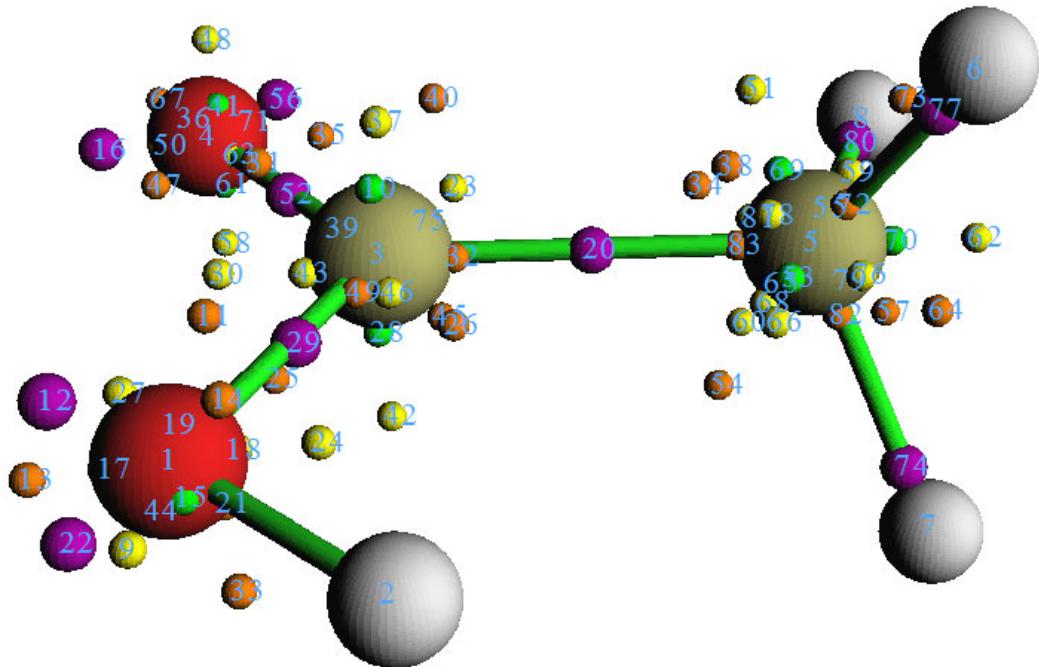


default value unchanged

-1 // Use each nucleus as sphere center in turn to search CPs. Since there are 8 atoms, and guessing points in each sphere is 1000, Multiwfn will try to search CPs from 8*1000 guessing points. Of course, the more the guessing points you set, the larger probability all CPs could be found in this search

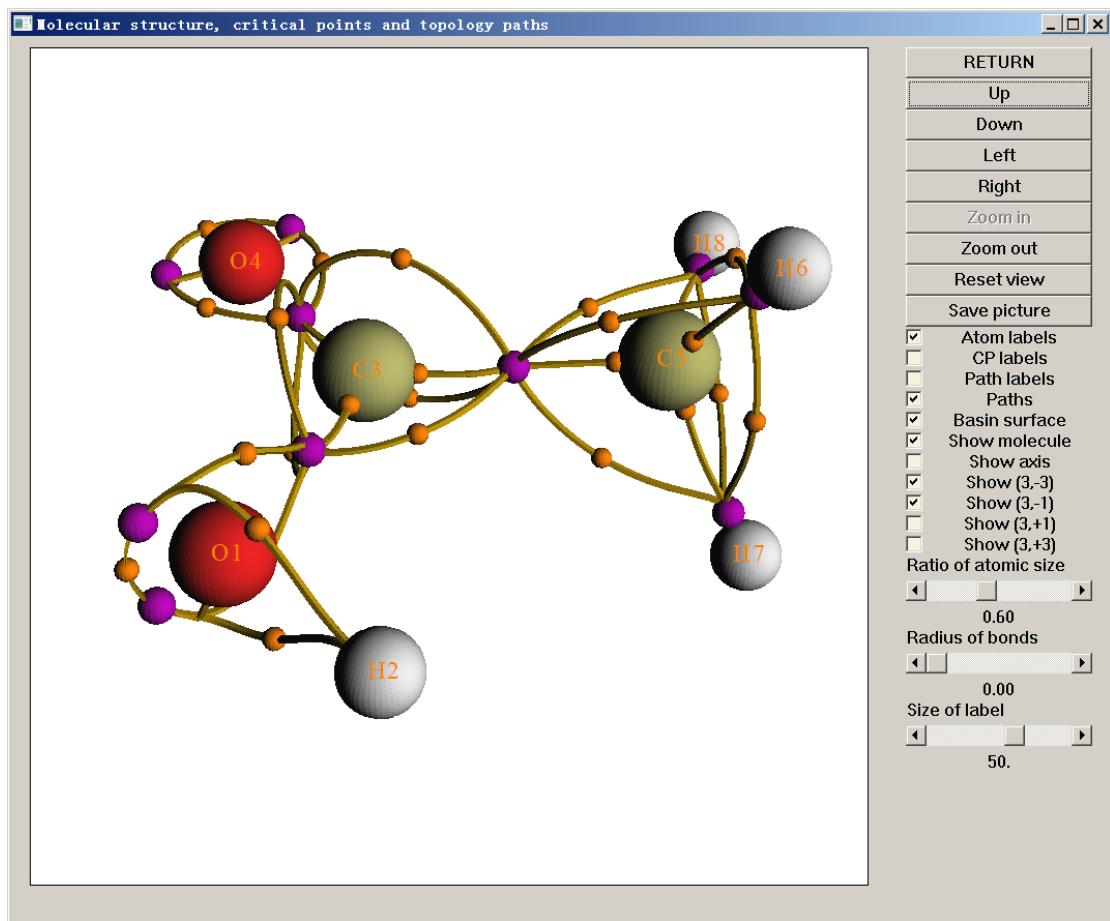
-9 // Return to upper menu

0 // Visualize the searching result



It is clear that the number of CPs in LOL function space is very large. Actually there are still some CPs were not been found in the search, if you repeat the search one time or several times, some missing CPs could be found. Since all interesting CPs have been found, repeat the search is unnecessary. From the graph, it can be seen that the three (3,-3) type CP, namely CP20, 29 and 52 correspond to the locally maximal electron localization in covalent bond regions. Note that the three CPs very close to hydrogens have no strong physical meaning, in my own viewpoint they are the consequence of failure of definition of LOL. CP12, 22, 16 and 56 correspond to lone pairs of oxygens.

Now choose option 8 to generate the topology paths linking (3,-1) and (3,-3) CPs, then choose option 0 to visualize the result again. The paths clarify the intrinsic relationship between CPs. You may find there are too many objects in the graph and they confused you visually. To relieve your visual burden, some uninteresting object can be screened by properly adjusting GUI widgets, as shown below.



4.3 Plotting the spin density curve of triplet formamide along carbon and oxygen atoms

Last_Exile\formamide-m3.wfn // Assuming this input file is in current directory, so the path of directory is not needed to input

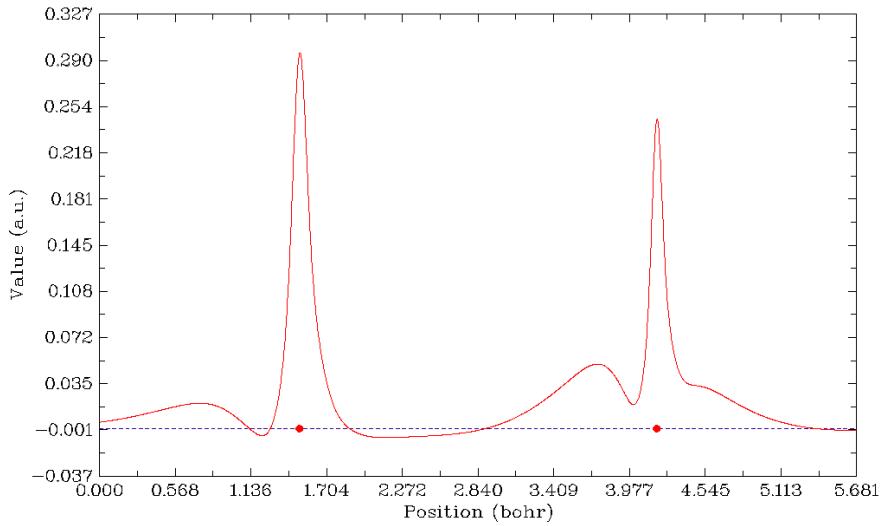
3 // Function 3, plot property in the line

5 // Spin density function

1 // Defining the line by nuclear coordinates of two atoms

1,6 // The indices of the two atoms, carbon and oxygen atoms correspond to 1 and 6 in present example

The graph shows up immediately:



The X-axis corresponds to the position in the line you defined, the dash line corresponds to the position of Y=0, two red circles denote the position of the two atom nuclei. After you click right mouse button on the graph to close it, a new menu appears on command-line interface, you can export the graph, adjust the range of Y-axis, export X-Y data set and replot the graph and so on by corresponding options. By selecting option 6, minimum and maximum positions can be located:

```

Local maximum X:      0. 753677  Value:      0. 20104903D-01
Local minimum X:      1. 219518  Value:      -0. 55161276D-02
Local maximum X:      1. 503567  Value:      0. 29655008D+00
Local minimum X:      2. 145518  Value:      -0. 70202704D-02
Local maximum X:      3. 736194  Value:      0. 51047629D-01
Local minimum X:      4. 006988  Value:      0. 18918295D-01
Local maximum X:      4. 184992  Value:      0. 24448996D+00
Total ly found      3 local minimum,      4 local maximum

```

4.4 Plot properties in a plane

4.4.1 The color-filled map of electron density of hydrogen cyanide

In this example we plot slice map for electron density of hydrogen cyanide

Blue_drop\HCN.wfn

4 // Plot graph in a plane

I // Electron density function

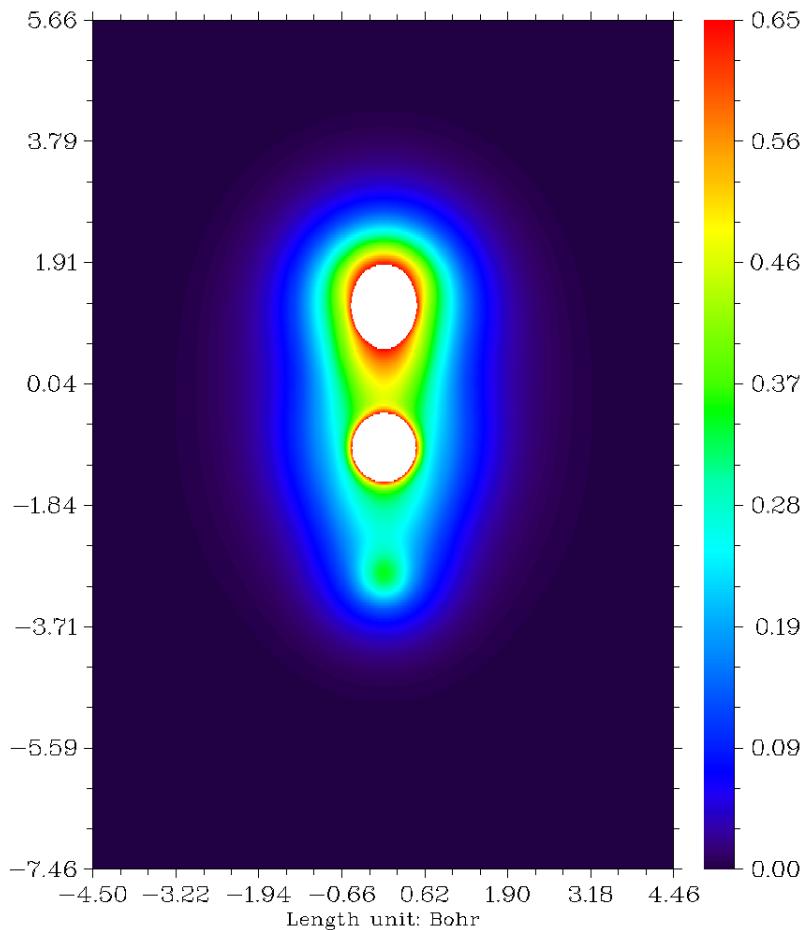
I // Select color-filled map as graph type

200,200 // Number of points in each dimension, 200,200 is recommended value for most cases. If you increase the value, graph will become finer, but you have to wait longer time for calculating data and plotting

2 // Select XZ plane (in hcn.wfn, Z-axis is molecular axis)

0 // The number of XZ planes is infinite, here we choose the only one with Y=0 bohr

After a few seconds the graph pops up



The central regions of carbon and nitrogen are white, suggesting that electron density exceeds the upper limit of color scale (0.65). Close the graph, then a menu appears, you can choose corresponding options to adjust plotting parameters and then use -1 to replot again, or export X-Y data set to plain text file for replotting the graph by external data visualization software (sigmaplot, matlab, etc.), or save the picture in current directory (the graphical format can be controlled by "graphformat" in settings.ini) and so on.

4.4.2 Shaded surface map with projection of localized orbital locator in a plane of monofluoroethane

NOIR\c2h5f.wfn

0 // Let's view the molecular structure first to find the plane we are interested in, suppose that the C-C-F plane is what we want, remember the indices of the three atoms: 1, 5, 8, and then press RETURN button

4 // Plot graph in a plane

10 // Localized orbital locator (LOL)

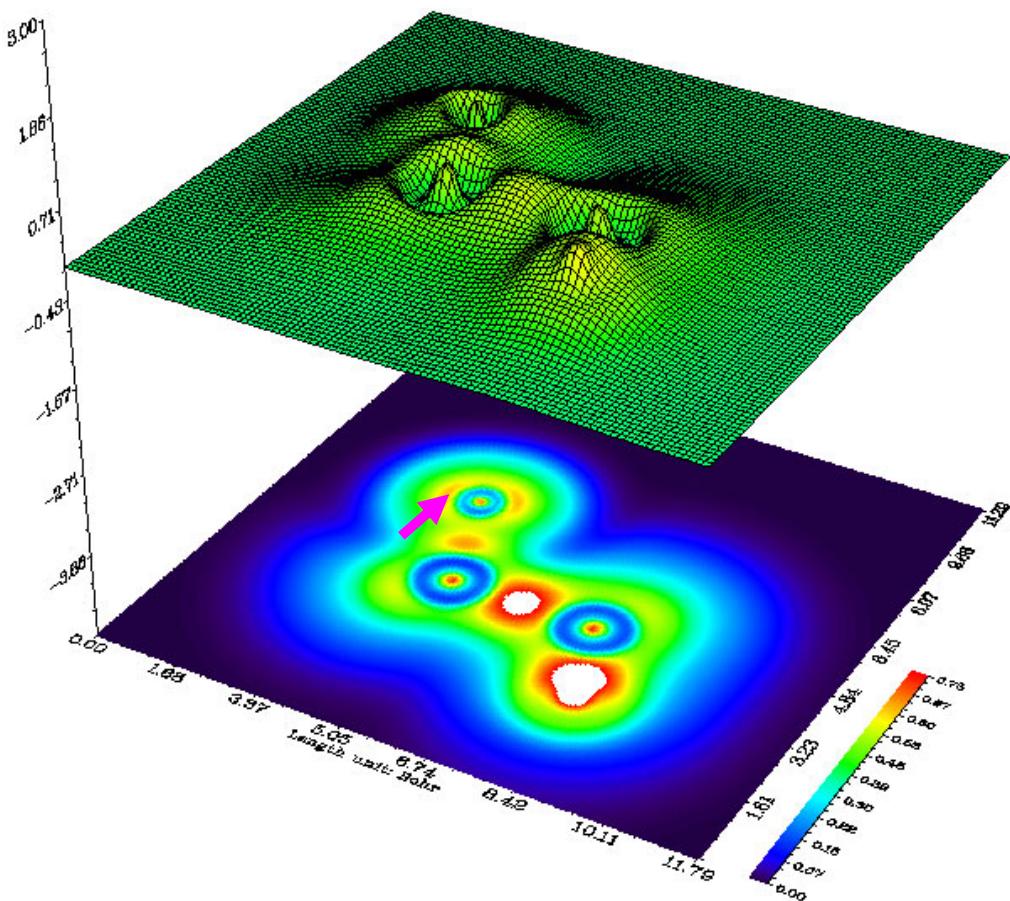
5 // Shaded surface map with projection

100,100 // For this type of graph, this is recommended value

4 // Define the plane by three atoms. If you want to enlarge plotting region, select 0 first and input a larger value than the default one (4.5 bohr)

1,5,8 // The indices of the three atoms

The graph immediately pops up:



We can see that covalent regions have high LOL value, the electron depletion regions between valence shell and inner shell are shown by the blue circles around nuclei. The lone pairs of fluorine atom is pointed out by purple arrow.

The graph you have seen in the GUI can be saved to graph file by option 0 at post-process interface, if you found the exported picture is truncated at edge, select option -1 to reenter the GUI window, zoom out the graph and then export the picture again.

4.4.3 Contour map of electron density laplacian of uracil without contributions of two atoms

Tri_sestry\Yukio_Nana\Uracil.wfn

-4 // Discard contribution of some atoms

2 // Assume that two atoms (C3 and C4) will be discarded

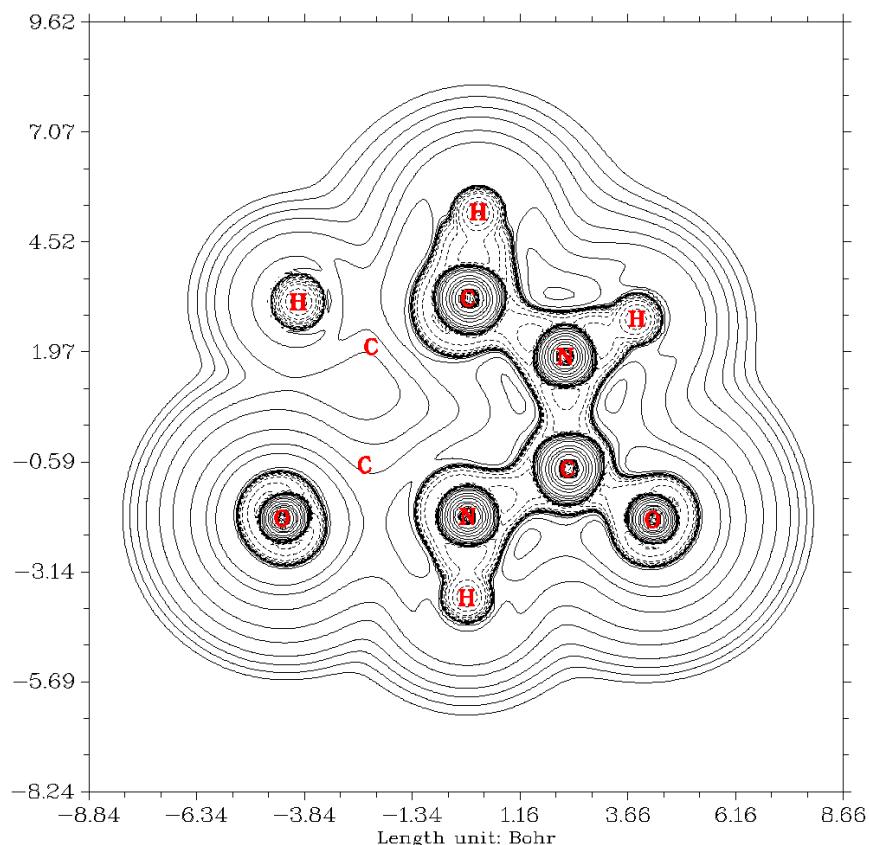
3 // The index of the first atom is 3

4 // The index of the second atom is 4

```

4 // Plot graph in a plane
3 // Laplacian function
2 // Contour line map
200,200
1 // XY plane will be plotted
0 // The Z-position of the XY plane is zero, that is molecular plane
Below is the resultant graph, solid and dash line correspond to positive and negative regions respectively.

```



From the graph you can see the contribution from the two carbons have been discarded as expected. By using the options shown on the command-line screen, you can save, export, redraw the picture, adjust plotting parameters, etc. If option 2 is chosen, the isovalues will be marked on corresponding contour lines. Once option 3 is chosen, you will enter an interface for setting up contour lines, please consult Section 3.5.3. For more about molecular fragment setting please consult Section 3.1.

4.4.4 Contour map of electrostatic potential of chlorine trifluoride

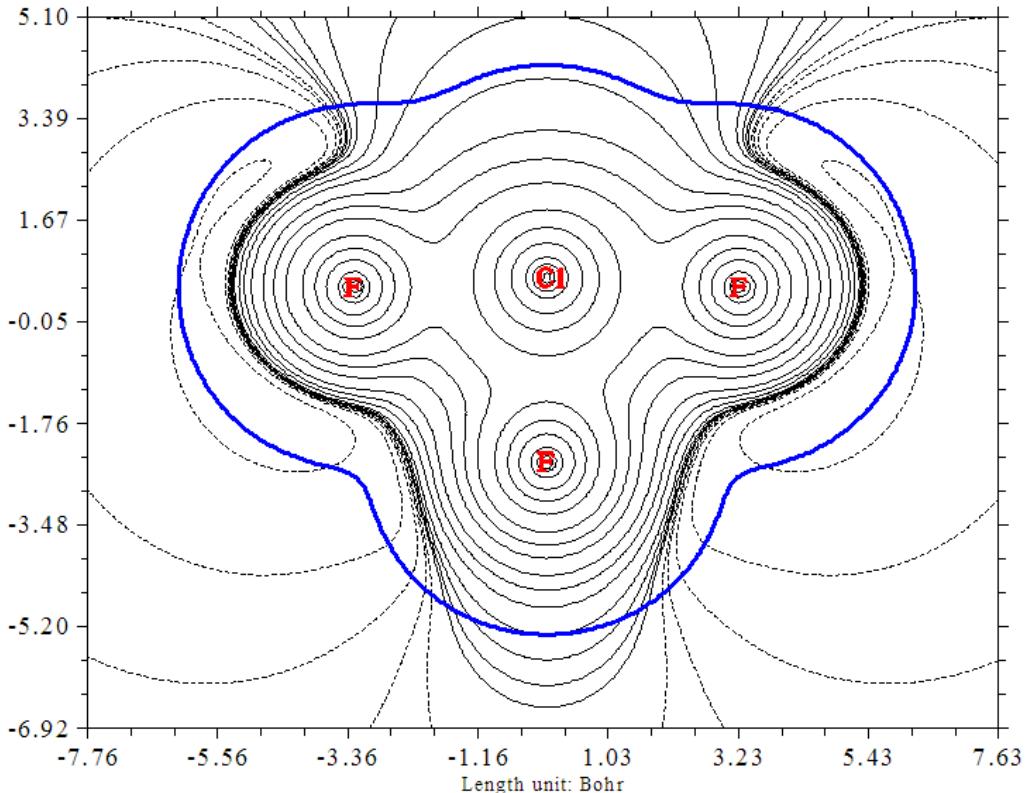
```

Billy_Herrington\ClF3.wfn // ClF3.wfn was generated at B3LYP/6-31G* level
4 // Plot graph in a plane
12 // Total electrostatic potential
2 // Draw contour line map
120,120
3 // YZ plane

```

0 // Set x coordinate of the YZ plane to 0

Because evaluation of electrostatic potential (ESP) is evidently more time-consuming than other real space functions, you need to wait for a while. Once the calculation is finished, ESP map will pop up. This map is inconvenient to be visually analyzed, since what we are interested in is often the ESP distributed on molecular vdW surface, so it is better to plot the vdW surface on this map simultaneously. In order to do this, we close the graph by clicking right mouse button, choose option 15, and then choose option -1 to replot the graph, you will see such a picture



The bold blue line corresponds to vdW surface (isosurface of electron density=0.001 a.u., as defined by R. F. W. Bader). From the graph it is clear that bromine atom is positively charged, because the local vdW surface closed to bromine atom largely intersected solid contour lines (correspond to positive ESP). For the same reason, we can see that the equatorial chlorin atom possesses more electrons than the two axial chlorin atoms, this point can be further verified when we calculate Hirshfeld atomic charges for this molecule in section 4.7.

4.4.5 Gradient+contour map with topology paths of electron density of hydrogen peroxide

vb1510s\kabat\h2o2.fch // Of course, you can also use .wfn file as input

4

I // Electron density

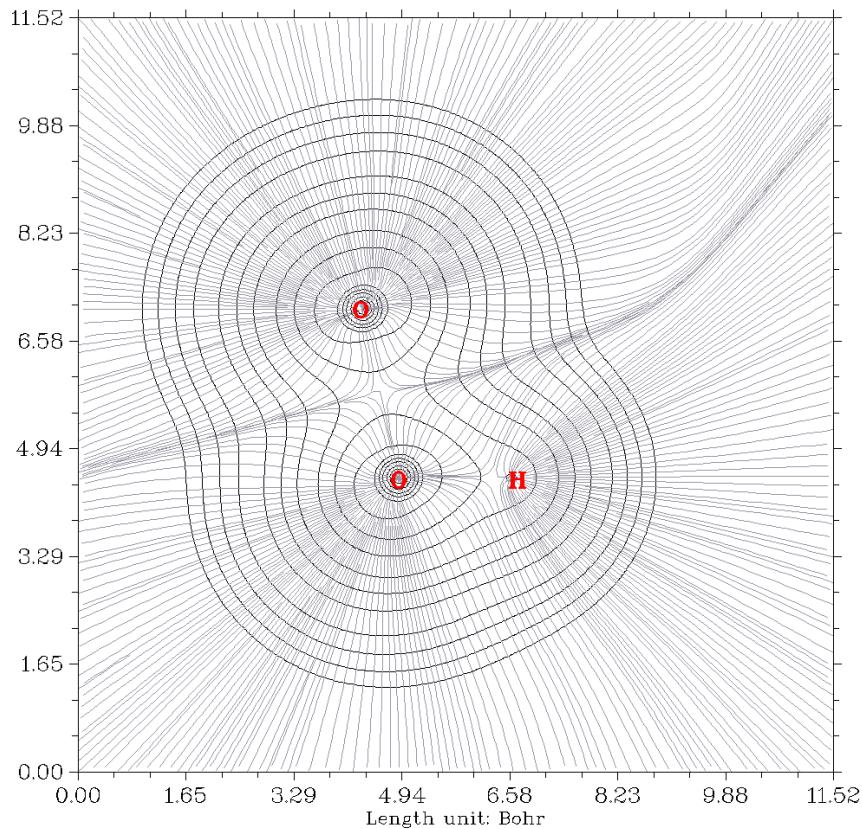
6 // Gradient line with/without contour line map

200,200

4

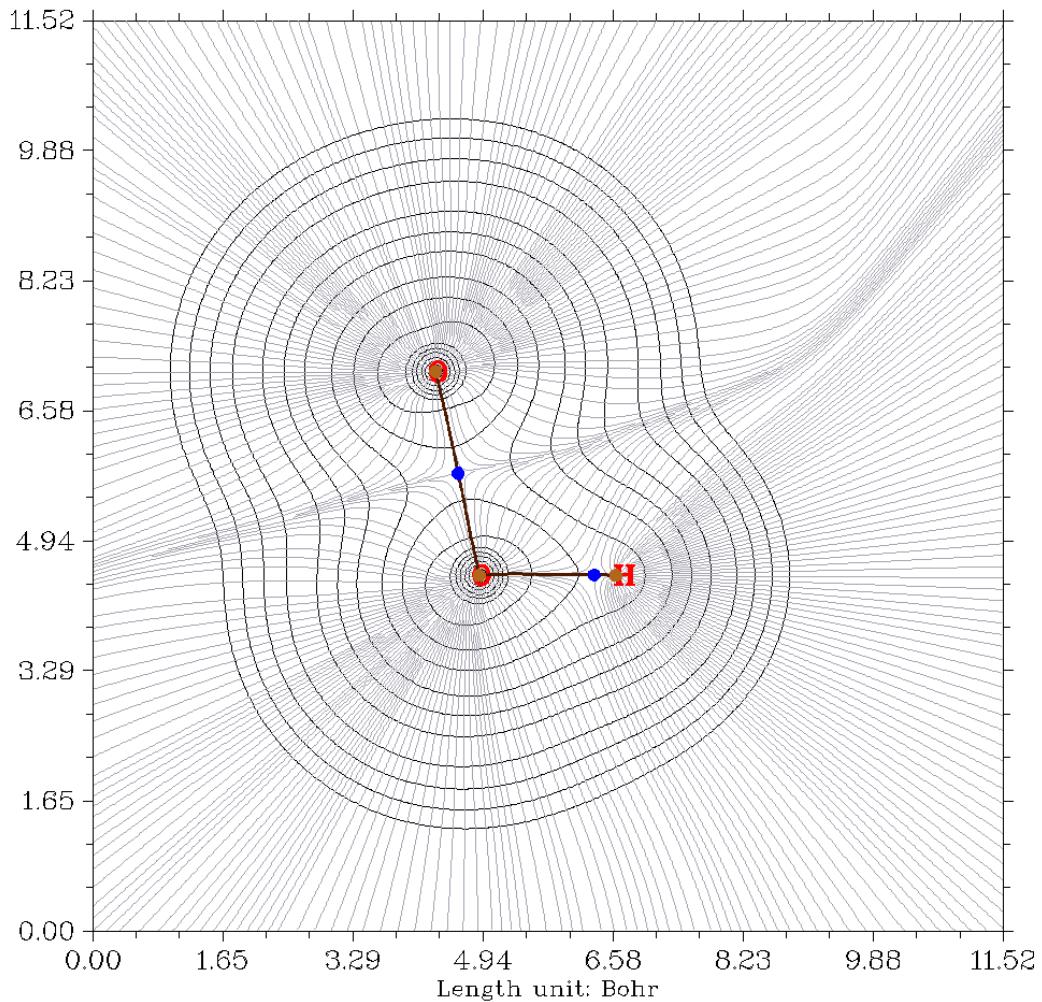
2,I,3 // Define the plane by nuclear coordinates of atom 1, 2 and 3

Generating data and plotting gradient map take more computational time relative to other graph type, however since present system is small and basis-set is only 6-31G*, resultant graph shows up immediately:

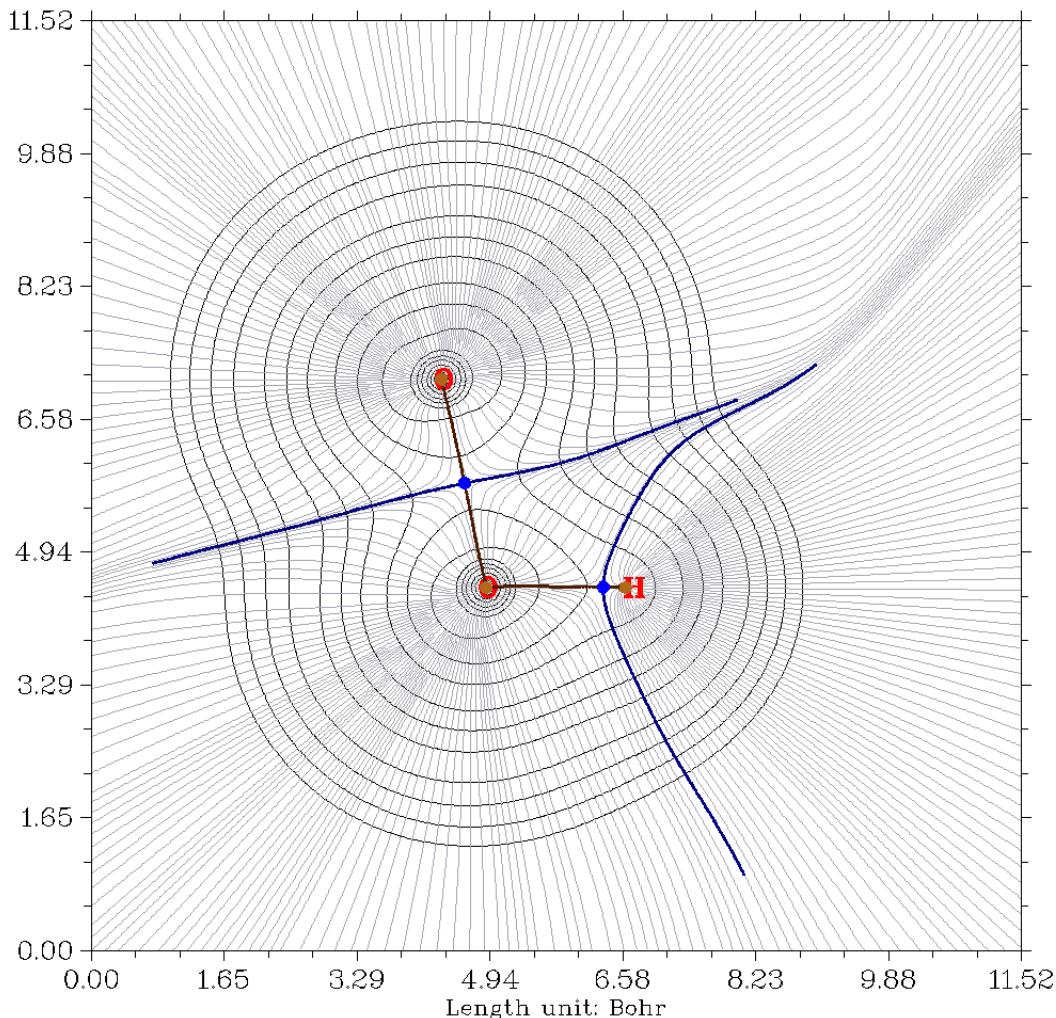


This type of graph is very useful in Bader's QTAIM analysis. You can also plot gradient+contour map for any other real space functions supported by Multiwfn.

If you want critical points and paths also to be portrayed on the graph, you need to do topology analysis as described in Section 4.2.1, then return to main menu, and then draw contour or gradient map as usual. The resultant graph looks like below. Brown, blue, and orange circles denote (3,-3), (3,-1) and (3,+1) critical points respectively, bold black lines depict bond paths.



Interbasin paths can also be drawn on the graph. If you have finished the search of CPs in topology analysis module, after you draw contour/gradient/vector field map, you could find a option named "Generate and show interbasin paths" in post-process stage, select it, and show the graph again, the interbasin paths will be drawn on the graph by bold deep blue lines:



4.4.6 Deformation map of electron density of acetyl chloride

Deformation map of electron density clearly shows you the electron density variation during formation of molecule, that is actual molecular electron density minus electron densities of each atom (in free state). It is a labor work to draw such a graph. However, with Multiwfn this task is very easy.

K-ON_Mio\ch3cocl.wfn

4

-2 // Tell Multiwfn you want to draw deformation map, then Multiwfn prepares free-state atom wavefunctions

6-31G* // The basis-set used to generate atom wavefunctions by Gaussian. Because the basis-set used for ch3cocl.wfn is 6-31G*, so here you should use 6-31G* too

d:\study\g09w\g09.exe // The path of executable file of Gaussian (you can also use G03). If you already set correct path in “gaupath” parameter in settings.ini, then Multiwfn will not ask you to input the path every time

Now Multiwfn starts to invoke Gaussian to calculate atom wavefunctions, then Multiwfn translates and sphericalizes them internally. These temporary wavefunction files are stored in “wfntmp” folder in current directory, after you get the expected graph you can delete the folder. Let's continue to input the remain commands.

I // Electron density function

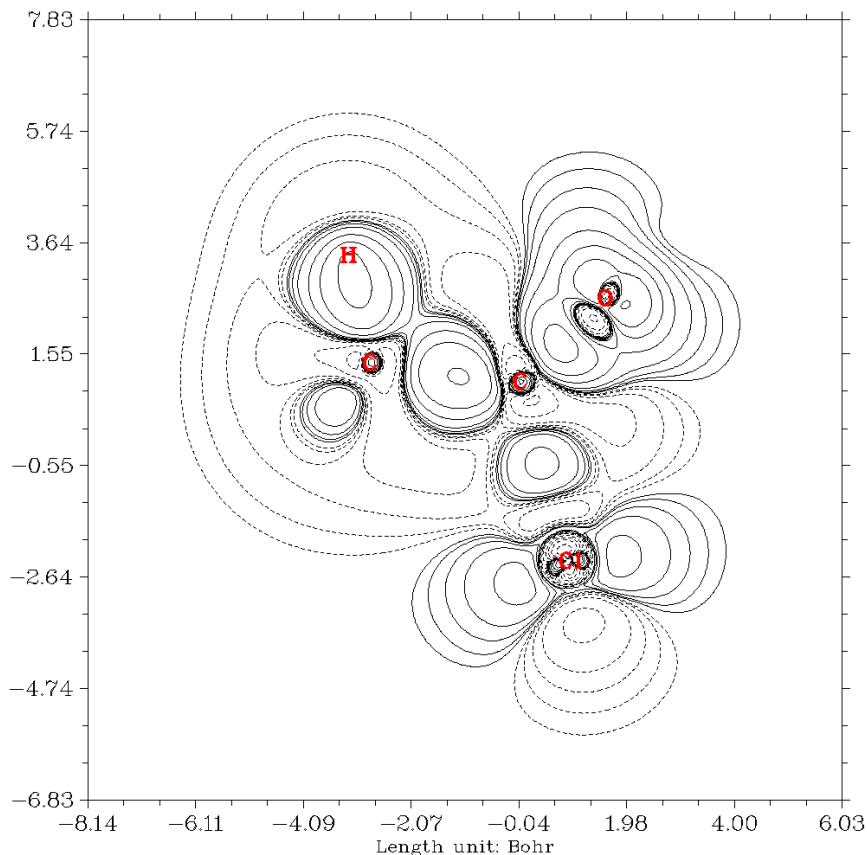
2 // Contour map

200,200

I

0 // The XY plane with Z=0 is the plane of acyl chloride

Then the deformation map pops up:



As we expected, electron density concentrates towards bonding regions, we also find the density distribution around chlorine atom is largely desphericalized, satisfied with hybrid orbital theory, chlorine atom forms somewhat sp^3 hybrid state.

If you want to avoid recalculating atom wavefunctions everytime, you can copy the .wfn files those without number suffix (such as "C.wfn") from "wfntmp" folder to "atomwfn" folder in current directory, if Multiwfn finds the atom wavefunctions that needed already exist in "atomwfn" folder, then Gaussian will not be invoked to calculate them again. You can also use "genatmwfnpdb" in "examples" direotory to generate all atom wavefunctions under specific basis-set in a single run, please consult Section 3.7.3. The "atomwfn" folder in "examples" direotory contains all atom wavefunctions under 6-31G*, you can copy the folder to current directory, after that you don't need Gaussian again.

You can also plot deformation maps for other functions by choosing corresponding real space function, however not all of them are meaningful.

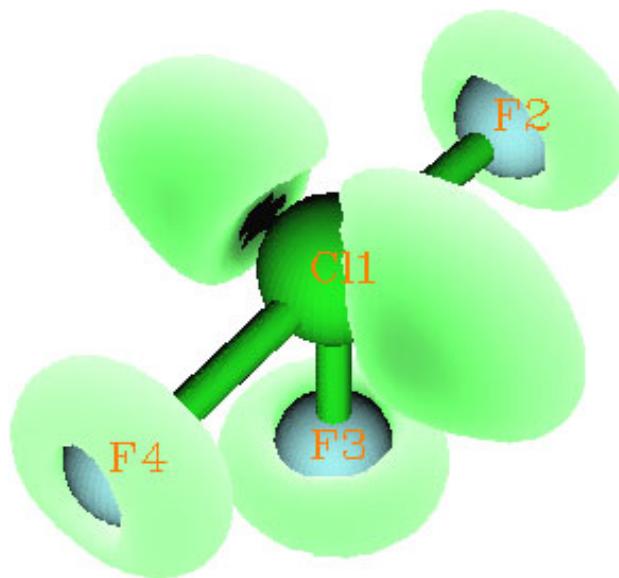
4.5 Generate cube file and view isosurface

4.5.1 Study electron localization function of chlorine trifluoride

```
H:\Ready Kirken\ClF3.wfn // Chlorine trifluoride at B3LYP/6-31G*
5 // Generate grid data and view isosurface
9 // Electron localization function (ELF)
2 // Medium quality grid, about 512000 points will be evaluated, this setting is fine enough
for small system. Please consult Section 3.6 for more about grid setting informations.
```

Now Multiwfn starts to calculate grid data, this is a time-consuming task, you have to wait for a while. Notice that if your computer have multiple CPU cores (SMP parallel architecture), I **strongly** suggest you enable the parallel mode, that is set “nthreads” parameter in settings.ini to the number of CPU cores, you will find calculation speed improved significantly! (By default, the parallel implementation is enabled for four cores)

When the calculation is finished, Multiwfn outputs some statistical informations, in the new menu you can draw the isosurface by option -1, then a GUI window pops up, input isoalue of 0.85 in the text box and press “Enter” button on your keyboard, the isosurface would looks like this (some ugly black polygons may appear, this is bug in current DISLIN graphical library, try to rotate molecule to eliminate them):



The ELF isosurface clearly points out the lone electron pair regions of fluorine and chlorine atoms. After click “Return” button to close GUI window, you can use option 1 to save the graph in current directory (the effect is the same as clicking the button “save pic” in GUI), or use 2 and 3 to export grid data to Gaussian cube file and to plain text file in current directory respectively.

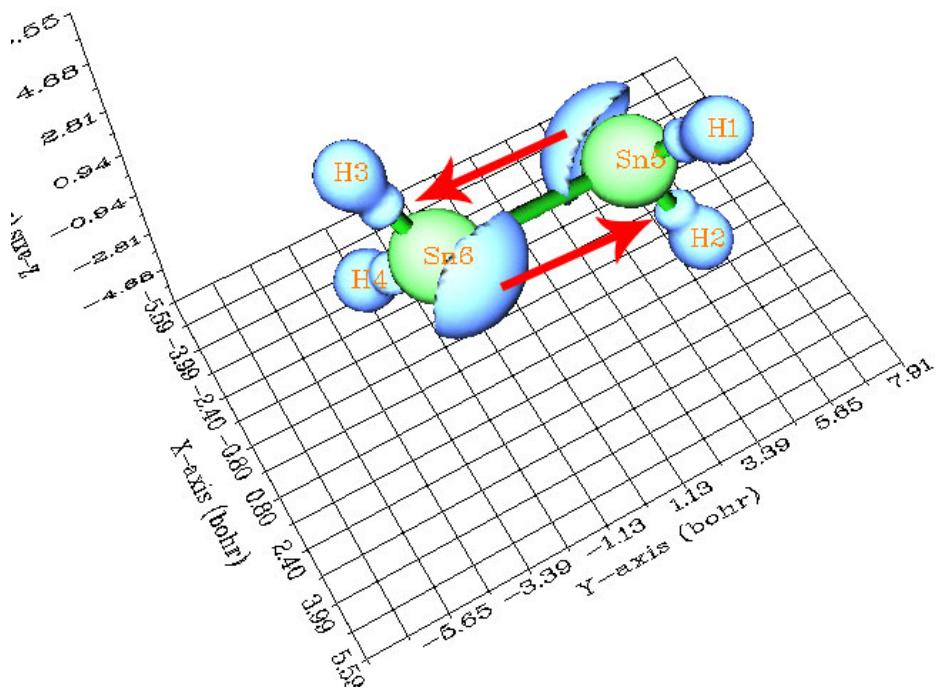
4.5.2 Study electron density laplacian of Sn₂H₄

Electron density laplacian is another localization indicator as ELF and LOL but with longer

history. Laplacian is not as good as ELF and LOL for highlighting localization region, its distinguishability is weak, for example, the shell structures of atoms heavier than krypton cannot be fully illustrated by laplacian, and if you attempt to use laplacian to analyze chlorine trifluoride, you will find the lone pair regions of fluorine atoms (circular isosurfaces) are hard to identify. Moreover, the value range of laplacian is too large, which brings difficulties to analysis. However for most system laplacian still works fine, and I will demonstrate its usefulness below.

Although both carbon and tin atoms are elements attributed to IV A group, however the structure of Sn_2H_4 is bend rather than planar as C_2H_4 . Some people pointed out that the reason is the energy of singlet SnH_2 is lower than triplet one, so tin atom forms sp^2 hybrid state in SnH_2 , when two SnH_2 fragments approach, lone pair electrons in each SnH_2 will coordinate with unoccupied p-atomic orbital of tin atom in another SnH_2 to form Sn_2H_4 . This viewpoint can be verified by laplacian function.

I have provided wavefunction file Sn2H4.wfn in “examples” folder, you can use this file to plot isosurface of laplacian function, the procedure is the same as in Section 4.5.1, the only difference is laplacian function should be chosen rather than ELF. The wavefunction file is generated at B3LYP/Lanl2DZ level, don’t worry about if the pseudo-potential treatment of inner electrons will artificialize the isosurface you are interested in, because the lone pair regions are concentrated only from electrons in valence shell, while Lanl2 pseudo-potential not only treats valence shell but also treats sub-valence shell explicitly, so it is safe. Below is the resultant isosurface:



The green and blue regions are the isosurfaces corresponding to isovalue of +0.07 and -0.07 respectively, that is the blue regions are where valence-shell electrons concentrated on. The lone pair regions appear on the graph as we expected, the red arrows indicate how the lone pairs

coordinate with the unoccupied 2p atomic orbital of neighbour tin atom.

4.5.3 Study aromaticity of benzene by ELF- α and ELF- π

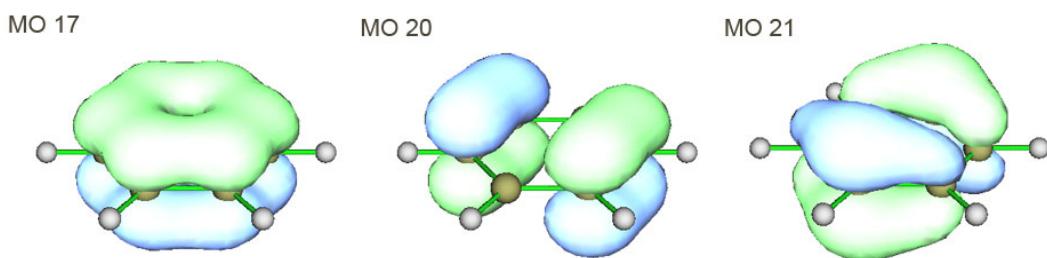
ELF- α and ELF- π are commonly used α and π aromaticity indices. ELF- α (ELF- π) are defined as the ELF value at bifurcation point (that is (3,-1) CP) of ELF basin that only contributed from α orbitals (π orbitals), see *J. Chem. Phys.*, 120, 1670 and *J. Chem. Theory Comput.*, 1, 83 for detail. The theoretical basis of these indices is that the ELF value at bifurcation point measures interaction between adjoining ELF domains, the larger value means electrons have better delocalization between these domains, which is commonly recognized as the nature of aromaticity. It is argued that if ELF- π is larger than 0.70, then the molecule has π aromaticity. While if the average of ELF- π and ELF- α is larger than 0.70, one can say that the molecule is global aromatic. In present example, we will calculate ELF- α and ELF- π for benzene.

In order to separate α and π orbitals, we need to know which orbitals are π orbitals first.

benzene.wfn // Optimized at B3LYP/6-311G* level, the same level used in *Chem. Phys. Lett.*, 443, 439

0 // Preview molecular orbitals (MO)

Now check orbital shape of each MO in turn, we found 17th, 20th and 21th MOs are π orbitals, as shown below. All other MOs are recognized as α orbitals.



We first calculate ELF- π . The contribution to ELF from α orbitals should be omitted, **this can be realized by setting occupation number of all α orbitals to zero**.

6 // Enter "Modify & Check wavefunction" interface

26 // Set occupation number for some orbitals

0,0 // 0,0 means select all orbitals

0 // Set occupation number of all orbitals to zero

26 // Set occupation number for some orbitals

17,17 // Input orbital range. We want to select MO 17, so input 17 for both lower limit and upper limit

2 // Set occupation number of MO 17 to 2.0, since this is a close-shell system

26 // Set occupation number for some orbitals

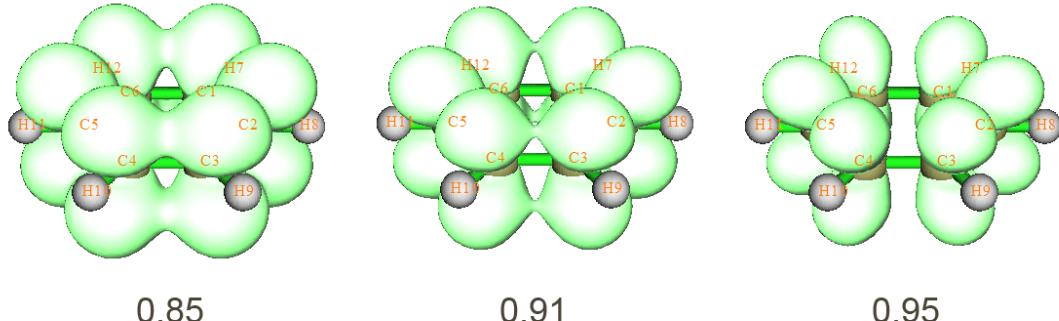
20,21 // Select MO 20 and MO 21

2 // Set their occupation number to 2.0 too. If you want to check if occupation numbers have been correctly set, choose option 3. You will find occupation number of all α orbitals have become zero, that means they will have no any contribution when we calculate any real space functions

-I // Return to main menu

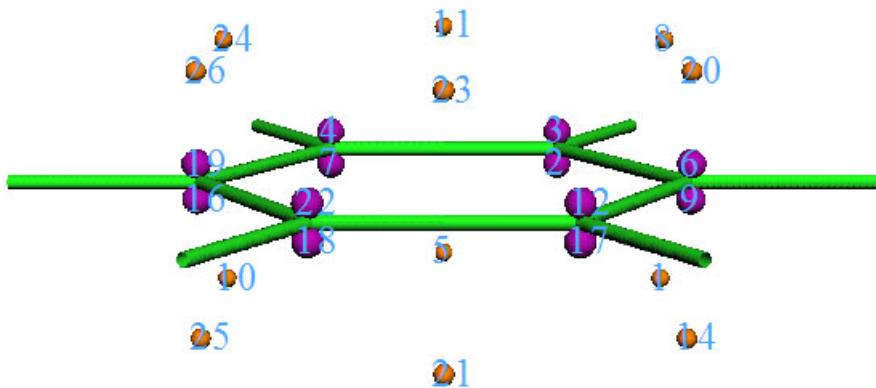
There are two ways to calculate ELF- π , way 1 is to examine ELF isosurface directly, this way is more intuitive but less accurate than way 2, in which topology analysis technique is used. We

use way 1 first. Generate and view isosurface for ELF by main function 5 as usual (likewise Section 4.5.1. High quality grid is recommended) This time the ELF isosurface only reflects π localization. Gradually increase isovalues, you will find the two circle-shape ELF basins are bifurcated to twelve spherical-like basins at about the iso value of 0.91 (see below graph), which implies ELF- π of benzene is about 0.91.



Let us use way 2 to calculate ELF- π again, in principle this way is more rigorous than way 1. Choose 0 to return to main menu.

- 2 // Topology analysis
- 11 // Select real space function
- 9 // Select ELF
- 6 // This searching mode is the most appropriate one for locating ELF CPs
- 1 // Start the search. The guessing points will be scattered around each atom in turn.
- 9 // Return to upper menu
- 0 // Visualize results. The resultant graph are shown below, two (3,+1) CPs are not shown



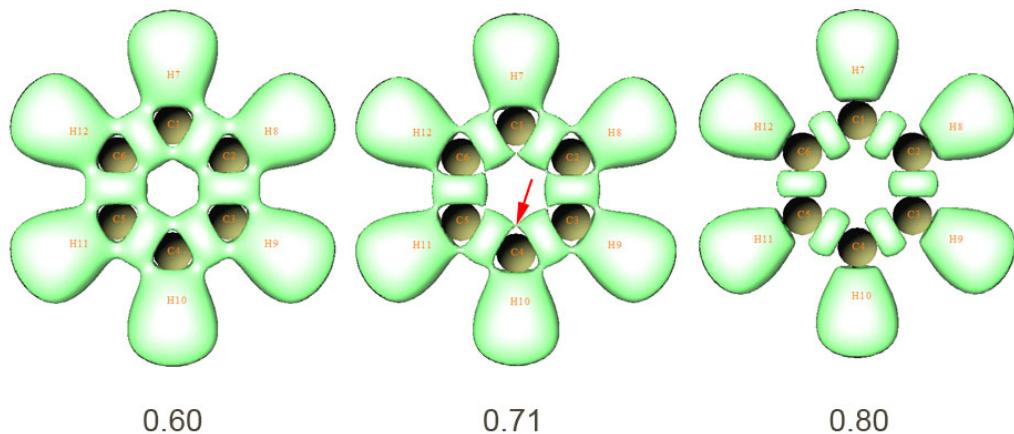
Compare this graph with ELF isosurface map, it clear that the (3,-1) CPs (orange) are bifurcation position of ELF basins. While (3,-3) CPs (purple) correspond to the maximal points of the twelves ELF basins. Now we check the ELF value at (3,-1) CPs, we can choose any one, since they are equivalent.

- 7 // Show all properties at a CP
- 23 // The CP with index of 23

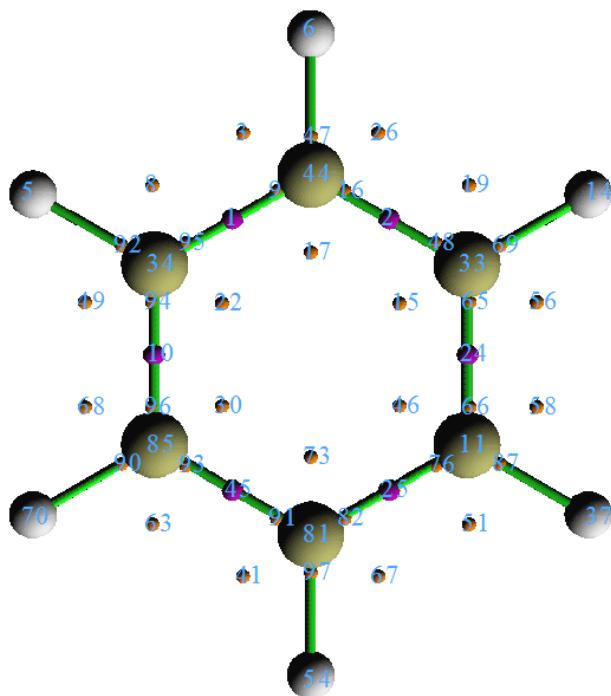
From the output, we find the ELF value at CP 23, namely ELF- π of benzene is 0.91247, this result is in very good agreement with the value 0.913 given in *Chem. Phys. Lett.*, 443, 439. Evidently, this value exceeds the criteria (0.70) of π aromaticity, which implies that benzene has

strong π aromaticity.

Now we calculate ELF- α for benzene. Reboot Multiwfn and load benzene.wfn, set occupation number of MO 17, 20 and 21 to zero. Then generate isosurface for ELF as usual, gradually adjust isovalue, find out at which isovalue the basins correspond to the α -bonds between carbon atoms are bifurcated. One can find at the isovalue equals to 0.71 the basin are bifurcated, suggesting that ELF- α is about 0.71, the bifurcation point are pointed by red arrow shown below:



Enter topology analysis module and search ELF CPs, like what we did in way 2 of ELF- π analysis. You will obtain below graph. For clarity, (3,+1) and (3,+3) CPs are hidden.



Compare positions of CPs with ELF isosurface map, it is clear that the (3,-1) CPs such as CP 73 correspond to bifurcation points of the α -bond basins between carbon atoms, check the ELF value at CP 73, we get 0.70907, this is the accurate ELF- α value. Our result agrees with the value 0.717 from *Chem. Phys. Lett.*, 443, 439.

The average of ELF- α and ELF- π is $(0.70907+0.91247)/2=0.81077$, which is larger than the criteria of global aromaticity. so benzene possesses global aromatic character.

4.5.4 Study Fukui function for phenol

Fukui function is a important concepts in conceptual density functional theory, it has been widely used in prediction of reactive site. Fukui function is defined as (*J. Am. Chem. Soc.*, 106, 4049)

$$f(\mathbf{r}) = \left[\frac{\partial \rho(\mathbf{r})}{\partial N} \right]_v$$

Where N is number of electrons in present system, the constant term v in the partial derivative is external potential. Generally the external potential only comes from nuclear charges, so v can be seen as nuclear coordinates for isolated system. It is argued that reactive sites have larger value of Fukui function than inert regions. We cannot directly evaluate the partial derivative due to the discontinuity when N is integer. In the finite difference approximation, Fukui function can be calculated unambiguously for three situations:

$$\text{Electrophilic attack: } f^-(\mathbf{r}) = \rho_N(\mathbf{r}) - \rho_{N-1}(\mathbf{r})$$

$$\text{Nucleophilic attack: } f^+(\mathbf{r}) = \rho_{N+1}(\mathbf{r}) - \rho_N(\mathbf{r})$$

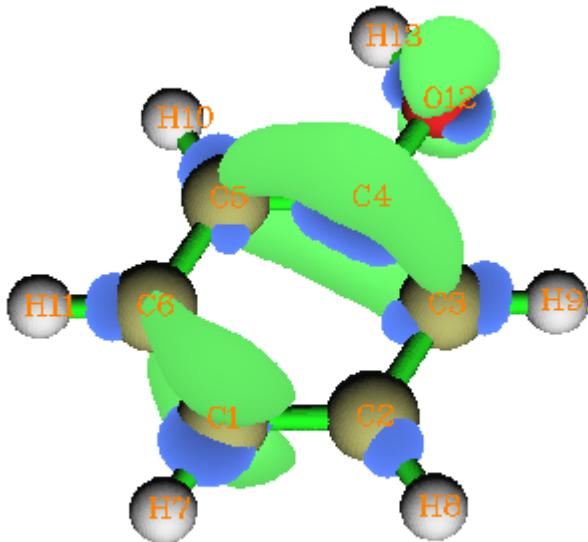
$$\text{Radical attack: } f^0(\mathbf{r}) = \frac{\rho_{N+1}(\mathbf{r}) - \rho_{N-1}(\mathbf{r})}{2}$$

In this example, we want to find out the reactive site of electrophilic attack of phenol. First, use quantum chemistry program to optimize structure and generate .wfn file for neutral state phenol, and then generate another .wfn file for phenol with charge of +1. Notice that at the second step you only need to change the charge and multiplicity to 1 and 2 respectively from 0 and 1, don't reoptimize molecule structure for +1 state phenol, because v (nuclear coordinates here) is defined as constant in the partial derivative of Fukui function. I assume the two .wfn files are called phenol.wfn and phenol+1.wfn, you can find them in “examples” directory.

For studying the isosurface of $f^-(\mathbf{r})$, you can generate cube files for the two .wfn files separately and obtain the difference by main function 13 of Multiwfn, and then visualize the cube file. However this work is more convenient to be achieved by make use of “custom operation” function in Multiwfn (please see Section 3.7.1 for detail), just follow the steps below:

```
phenol.wfn //Calculated at HF/6-31G* level
5
0 // Set custom operation
I // Only 1 file will be operated with the file has been loaded (namely phenol.wfn)
-, phenol+1.wfn // “-“ is minus operator, that means the property of phenol.wfn will minus
corresponding property of phenol+1.wfn, which is the wavefunction of the phenol with +1 net
charge (corresponds to N-1 electron state)
I // Electron density
2 // Medium quality grid
```

Now Multiwfn starts to calculate electron density grid data for phenol.wfn, and then calculate for phenol+1.wfn, finally the grid data of phenol.wfn, namely $\rho_N(\mathbf{r})$, will minus the grid data of phenol+1.wfn, namely $\rho_{N-1}(\mathbf{r})$, the result is $f^-(\mathbf{r})$. Let's check the isosurface, after adjusting the isovalue to a proper value (0.007), the graph is



Clearly, most positive part of $f^-(\mathbf{r})$ function are localized in O12, C1, C3, C4 and C5, that means para and ortho position of hydroxyl are favourite reactive sites of electrophilic attack, this conclusion is in agreement with empirical rule, that is hydroxyl is an ortho-para- director.

4.5.5 Study electron transfer of imidazole coordinated magnesium porphyrin

In present example, I will show you how to do fragmental electron density difference in Multiwfn. It can be imagined that during the course of imidazole coordinating to magnesium porphyrin, electron density transfer occurs, the variation can be made clear by subtracting electron density of imidazole (called NN below) and magnesium porphyrin (called MN below) in standalone state from the whole system (called MN-NN below). Of course, geometries of MN and NN have maintained the same states in MN-NN, otherwise the fragmental electron density difference is questionable.

MN-NN.gif in “examples” folder is Gaussian input file of MN-NN system (geometry has been optimized), run it, then MN-NN.wfn is yielded. After that delete the atoms in MN and NN part from MN-NN and save them to MN.gif and NN.gif respectively, then run MN.gif and NN.gif to get MN.wfn and NN.wfn. Since the size of wavefunction files are huge, they were not provided in “examples” folder. It should be paid attention that by default Gaussian always puts input orientation to standard orientation, that means the coordinates in MN.wfn and NN.wfn will not be

consistent with MN-NN.wfn, hence “nosymm” keyword must be specified in route section to avoid the automate adjustment of coordinates.

Now we generate cube file of fragmental electron density difference by Multiwfn.

MN-NN.wfn

5

0 // Set custom operation

2 // Two files will be operated with MN-NN.wfn

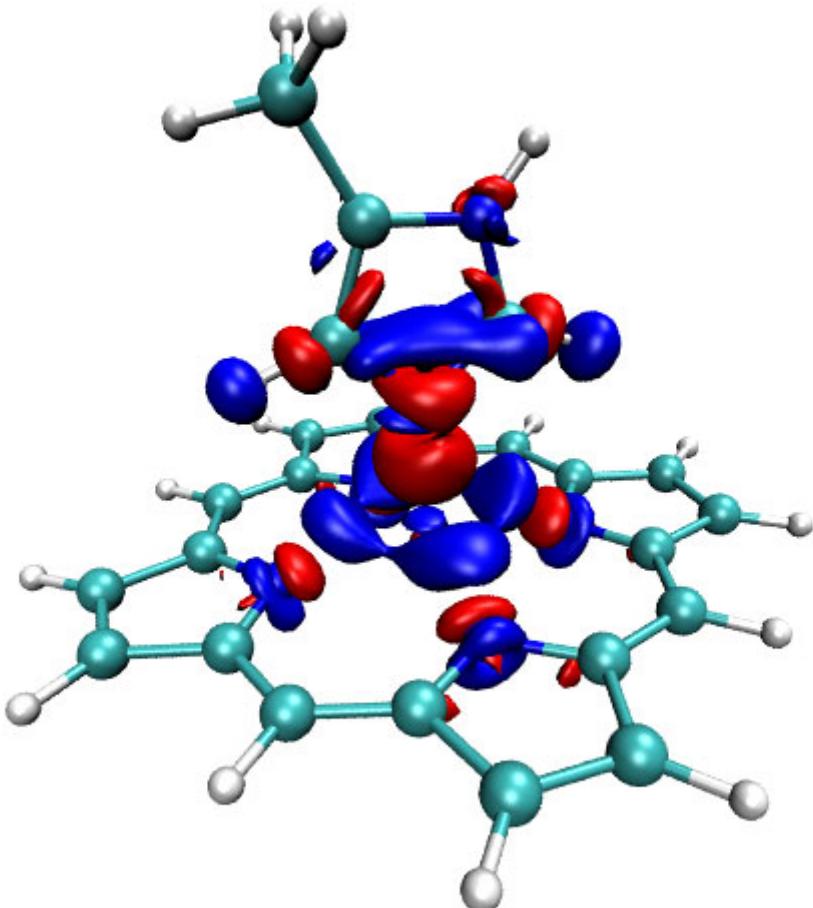
-,*MN.wfn* // Minus MN.wfn from MN-NN.wfn

-,*NN.wfn* // Then minus NN.wfn from MN-NN.wfn

1 // Electron density

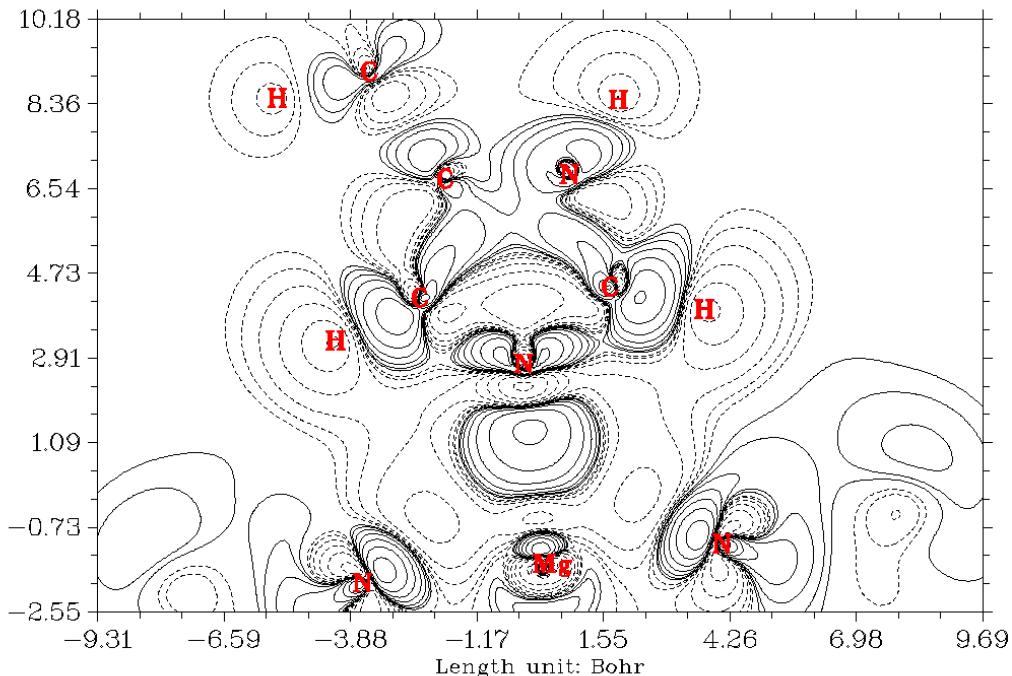
3 // Since present system is relative huge, we need more grid points than normal cases

Wait for a while, then select option 2 to export cube file. Because Multiwfn is not very professional at visualization of grid data, for large size grid data the speed is low, we use external tools (VMD).



The red and blue isosurface (± 0.0012) represent the region in which electron density is increased and decreased after NN coordinated to MN, it is obvious that electron density is shifted from backside of nitrogen in NN toward magnesium atom to strengthen the coordination bond. Besides, the appearance of NN not perturbed electron density distribution of porphyrin remarkably, only slight polarization effect occurs in the four coordination nitrogens of MN (by lowering

isovalue you may find richer informations, but they can be ignored due to the small magnitude). The picture below is contour map of fragmental electron density difference in NN plane, try to use custom operation in main function 4 (the same as the one in main function 5) to draw it by yourself, it is quite easy.



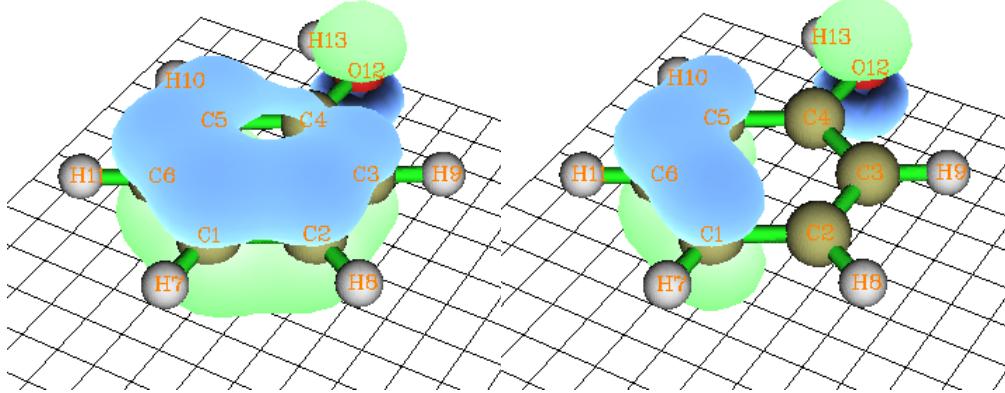
4.6 Modifying wavefunction

4.6.1 Delete certain Gauss functions

In subfunction 25 of main function 6, you can set orbital expansion coefficients of Gauss type functions (GTFs) which satisfied certain conditions. If the coefficients are set to zero, that means the informations of these GTFs are deleted. In this example, we delete all Z-type GTFs of atom 2,3 and 4 from orbital 23 of phenol and then plot isosurface for this molecular orbital

```
miku|MIG21|phenol.wfn
6 // Modify wavefunction
25 // Set the coefficients of some GTFs that satisfied certain conditions
0,0 // Set the index range of GTFs, only the GTFs satisfied this condition will be reserved
to next step. 0,0 tell Multiwfn the range is "ALL"
2,4 // Only the GTFs attributed to atom 2, 3, 4 will be reserved to next step
Z // Only reserve Z-type GTFs to next step
23,23 // Set lower and upper limit of orbital, if they are identical, then the only one orbital is
selected
0 // Set coefficients of selected GTFs in orbital 23 to zero, that is delete their informations
0 // Save current wavefunction to new.wfn in current directory
You can choose option 4 and input 23 to check expansion coefficient of orbital 23 to verify if
```

your operation is correct. Let's compare the isosurfaces of orbital 23 before and after modification. In order to plot the modified orbital 23, you can close Multiwfn and load new.wfn, or directly select option -1 to return main menu, and then enter main function 0 to plot the isosurface of orbital 23.



Left side is unmodified state, the molecular plane is parallel to XY plane, so if Z-type GTFs in some atoms are deleted, the corresponding part of isosurface should disappear, this is what we have seen at right side (modified state).

4.6.2 Remove contributions from certain orbitals to real space functions

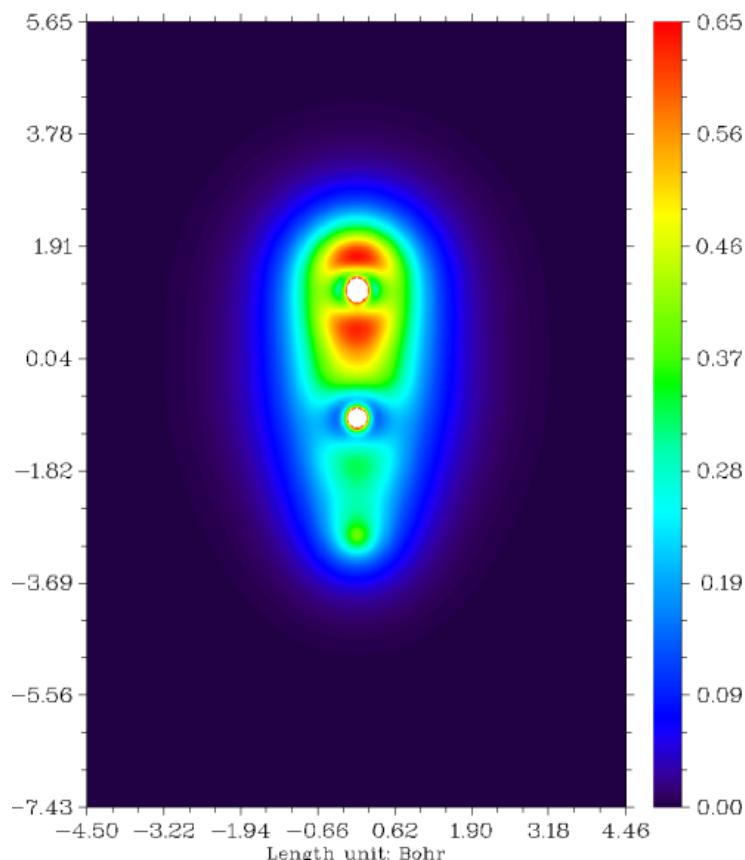
In Multiwfn, contributions from certain orbitals to real space functions can be removed. From formulae in Section 2.6 it is clear that if occupation numbers of certain orbitals are set to zero, they will have no contribution, as if they were not existed. Actually in Section 4.5.3 we have already used this trick to separate ELF as ELF- α and ELF- π . Notice that some real space functions are not linear, such as ELF and LOL, so they cannot be computed as the sum of contributions from each occupied orbitals. In contrast, some real space functions are linear, such as electron density and kinetic energy density.

In this example, we will remove contributions from the MOs consist of inner-core atomic orbitals to electron density for hydrogen cyanide, so that we can obtained valence electron density map.

The principle is very simple. Inner-core atomic orbitals always have very low energies, so they can only compose MOs with the lowest energies. Hydrogen has no inner-core atomic orbital, both carbon and nitrogen have one inner-core atomic orbital, hence what we need to do is to set occupations numbers of the first two MOs to zero (assume that this is close-shell wavefunction). Now boot up Multiwfn, and input following commands:

```
HCN.wfn
6 //Modifying wavefunction
26 // Set occupation numbers
1,2 // Select MO from 1 to 2
0 // Set their occupation numbers to zero
-1 // Return to main manu
```

Then, If we redo the example in Section 4.4.1, we will get below graph



From the picture we find that the white circles in carbon and nitrogen center are largely contracted, this is because the contributions from their 1s atomic orbitals are removed. Electron structures are exhibited more clearly in this picture than in Secton 4.4.1.

For convenience consideration, Multiwfn provides function 34 in main function 6, one can directly choose it to set occupation numbers of the MOs that consist of inner-core atomic orbitals to zero.

4.6.3 Translate and duplicate graphene primitive cell wavefunction to periodic system

The .fch or .wfn file generated by periodic boundary condition (PBC) calculation of Gaussian only contains wavefunction of primitive cell, so the analysis results don't show any periodic character, of course you can enlarge your inputted system to multiple cells before PBC calculation, but you have to wait much more time for computation. Multiwfn provides a way to convert primitive cell wavefunction to periodic wavefunction (at least in the region you are interested the new wavefunction shows periodic character).

In this section I use graphene as example. First, generate .fch file of graphene primitive cell. The content of “molecular specification” field in input file is

C	0. 000000	0. 000000	0. 000000
C	0. 000000	1. 429118	0. 000000
TV	2. 475315	0. 000000	0. 000000
TV	-1. 219952	2. 133447	0. 000000

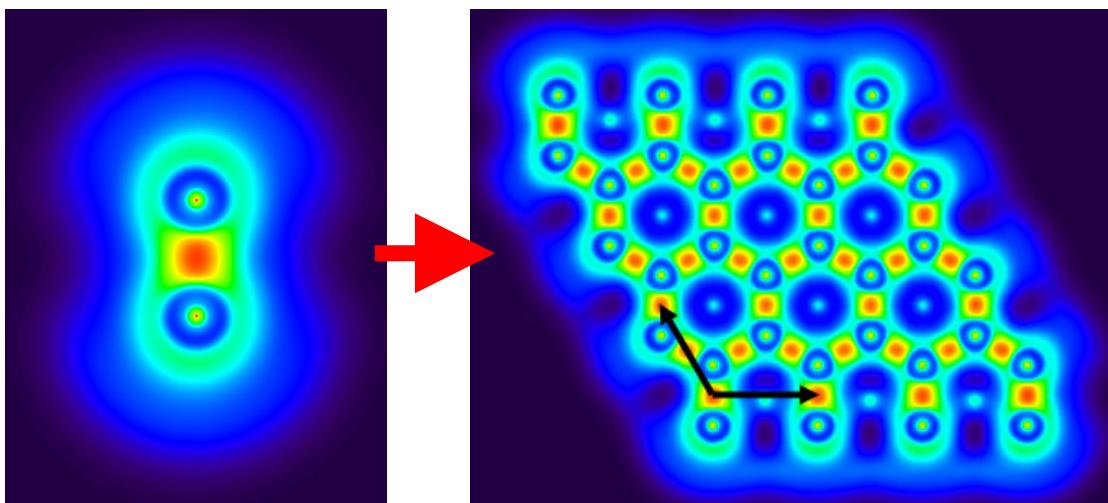
In route section, fill “#P PBEPBE/3-21g/Auto SCF=Tight”. Use Gaussian to run this input

file and then use formchk to convert the binary checkpoint file to graphene.fch. Then boot up Multiwfn and input:

```
graphene.fch
6
32 // Translate and duplicate primitive cell wavefunction
2.475315    0.000000    0.000000 // Translation vector 1
2 // Unit is angstrom
3 // Translate and duplicate present system three times in this direction
32 // Notice that current system already have four primitive cells, this time we will translate
and duplicate current system in another direction three times, so the final system will contain 16
primitive cells
-1.219952    2.133447    0.000000 // Translation vector 2
2 // Unit is angstrom
3 // Translate and duplicate present system three times in this direction
```

The left part of the picture below is LOL function of primitive cell. After above manipulation, we recalculate LOL function and then the right graph is obtained (black arrows denote translation vectors). Apparently, the central region of the extended system shows correct periodic character, however the behavior of boundary region is still incorrect, you can extend the system further to enlarge “correct” region.

Notice that if “nosymm” keyword is not specified in PBC calculation, Gaussian may automatically put the system into standard orientation, at this time you shouldn’t use the translation vectors in Gaussian input file as the translation vectors for translating and duplicating system in Multiwfn, but should use the content in “Translation vectors” field of .fch file or “PBC vector” segment in Gaussian output file.



4.7 Calculate Hirshfeld charges for chlorine trifluoride

I have introduced the theory of Hirshfeld population in Section 3.9.1, to calculate Hirshfeld charges for ClF3, input below commands in Multiwfn

```
yamibo\clf3.wfn
7 // Population analysis
```

I // Hirshfeld population

Because Hirshfeld population requires electron density of atoms in free-state, Multiwfn will prepare atom wavefunctions first, this procedure is the same as the one when plotting deformation map, please recall Section 4.4.6 and consult Section 3.7.3. Once the procedure is finished, the calculation of atomic charges is begun. The result not only includes atomic charges but also includes atomic dipole moments, the molecular dipole moment evaluated from atomic charges, the sum of atomic dipole moments and corrected molecular dipole moment, see Section 3.9.1 for detail explanation. From the result we find the charges of the three fluorine atoms are unequal, the equatorial one is -0.064 while the axial ones have more electrons, are -0.200.

Finally Multiwfn asks you if exporting the result, if you select ‘y’, the element names, atom coordinates and atomic charges will be outputted to a plain text file with .chg extension. You can use this file as Multiwfn input and use real space function “Electrostatic potential from atomic charges” to study the electrostatic potential from Hirshfeld charges.

4.8 Molecular orbital composition analysis

4.8.1 Analysis on acetamide by Mulliken method

The details about orbital composition analysis are documented in Section 3.10, in this example we first analyze the composition of the sixth molecular orbital of acetamide, and then analyze which orbitals have main contribution to the bonding between formamide part and methyl group.

touhou\CH3CONH2.fch // For composition analysis (except Hirshfeld analysis for atom composition) you have to use .fch file as input

8 // Orbital composition analysis

I // Use Mulliken partition

6 // The orbital index is 6

The composition of basis functions, shells and atoms are printed immediately, see below.

```
Threshold of absolute value: > 0.500000% // Only the basis functions
with composition larger than 0.5% will be printed, you can change the threshold
by "compthres" parameter in settings.ini.
```

Orbital :	6	Energy(a. u.):	-0. 90528995	Occ:	2. 00000000	Type:
<i>Al pha&Beta</i>						
Basis	Type	Atom	Shell	Local	Cross term	Total
23	S	5(C)	14	0. 44902%	0. 67507%	1. 12409%
24	X	5(C)	15	0. 31240%	0. 50522%	0. 81762%
25	Y	5(C)	15	4. 25271%	5. 88221%	10. 13493%
29	Y	5(C)	17	0. 00777%	-0. 61063%	-0. 60286%
38	S	6(O)	20	3. 50037%	2. 65507%	6. 15544%
42	S	6(O)	22	3. 29488%	1. 71316%	5. 00803%
53	S	7(N)	26	15. 20411%	15. 32688%	30. 53098%
57	S	7(N)	28	16. 89006%	17. 40040%	34. 29046%
61	XX	7(N)	30	0. 00774%	0. 58279%	0. 59053%

4 Tutorial

63	ZZ	7(N)	30	0. 02793%	-0. 98090%	-0. 95297%
67	S	8(H)	31	1. 27855%	3. 03091%	4. 30946%
69	S	9(H)	33	1. 52931%	3. 60949%	5. 13880%
Sum up listed above:				46. 75484%	49. 78967%	96. 54451%
Sum up all basis functions:				51. 95605%	48. 04395%	100. 00000%

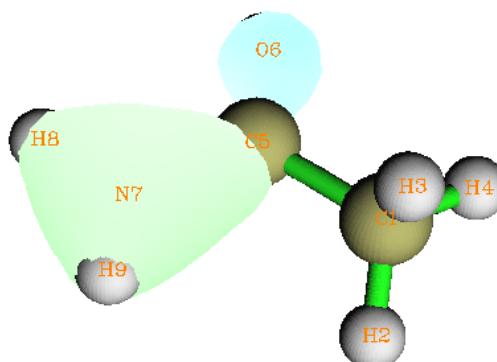
Composition of each shell, threshold of absolute value: > 0.50000%

Shell I	14	Type: S	in atom	5(C) :	1. 12409%
Shell I	15	Type: P	in atom	5(C) :	10. 95268%
Shell I	17	Type: P	in atom	5(C) :	-0. 97156%
Shell I	20	Type: S	in atom	6(O) :	6. 15544%
Shell I	22	Type: S	in atom	6(O) :	5. 00803%
Shell I	26	Type: S	in atom	7(N) :	30. 53098%
Shell I	28	Type: S	in atom	7(N) :	34. 29046%
Shell I	31	Type: S	in atom	8(H) :	4. 30946%
Shell I	33	Type: S	in atom	9(H) :	5. 13880%

Composition of each atoms:

Atom	1(C) :	1. 172494%
Atom	2(H) :	0. 054446%
Atom	3(H) :	0. 032119%
Atom	4(H) :	0. 008171%
Atom	5(C) :	11. 812452%
Atom	6(O) :	11. 632739%
Atom	7(N) :	65. 500850%
Atom	8(H) :	4. 470224%
Atom	9(H) :	5. 316506%

The result indicates that nitrogen has primary contribution (65.5%) to orbital 6, and the contribution consist of two S-shell (30.5% and 34.3%). P-shell of neighbour carbon and S-shell of oxygen have slight contribution too (both are about 12%). We can check if the result is reasonable by viewing isosurface (isovalue is set to 0.1 here):



From the graph, the region where the value of orbital wavefunction is large is mainly localized around nitrogen, and there is no nodal plane, so the orbital wavefunction in this region should be constructed from s-type orbitals. The isosurface also somewhat intrudes into the region of atom C5 and O6, so they should have small contribution to MO 6, moreover, because there is a nodal plane in C5, the atomic orbitals of C5 used to form MO 6 should be p-type. Obviously, these conclusions are in fairly agreement with composition analysis. The advantage of composition analysis is that the result can be quantified, while by visual study we can only draw qualitative conclusion, for some complex system we cannot draw even qualitative conclusion.

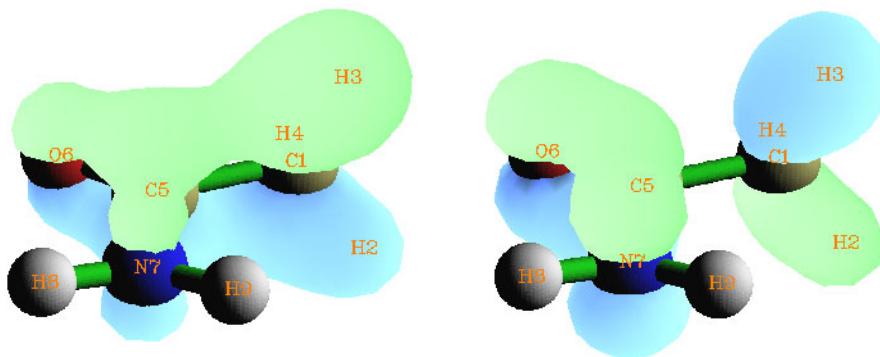
Now let's find which molecular orbitals have main contribution to the bonding between formamide part and methyl group.

```
touhou\CH3CONH2.fch
8 // Orbital composition analysis
-I // Define fragment 1
a 1,2,3,4 // Add all basis functions in atom 1, 2, 3, 4 (namely methyl group) into fragment1
q // Save fragment and return to upper menu
-2 // Define fragment 2
a 5,6,7,8,9 // Add all basis functions in atom 5, 6, 7, 8, 9 (namely formamide part) into
fragment 2
q
4 // Print composition of fragment 1 and the cross term between fragment 1 and 2 in all
orbitals by Mulliken analysis. If you only defined fragment 1, then only composition of fragment
1 will be printed
```

Amount of the information printed is huge, so I only extract cross term composition in all occupied orbitals:

Cross term between fragment 1 and 2 and individual parts:						
Orb#	Type	Energy	Occ	Frag1 part	Frag2 part	Total
1	AB	-19.104	2.000	-0.0013%	-0.0013%	-0.0026%
2	AB	-14.349	2.000	-0.0001%	-0.0001%	-0.0003%
3	AB	-10.282	2.000	0.0606%	0.0606%	0.1211%
4	AB	-10.184	2.000	0.0604%	0.0604%	0.1209%
5	AB	-1.038	2.000	0.3535%	0.3535%	0.7070%
6	AB	-0.905	2.000	0.7295%	0.7295%	1.4590%
7	AB	-0.739	2.000	7.9504%	7.9504%	15.9008%
8	AB	-0.588	2.000	-1.8755%	-1.8755%	-3.7510%
9	AB	-0.541	2.000	-0.7506%	-0.7506%	-1.5011%
10	AB	-0.466	2.000	3.8098%	3.8098%	7.6197%
11	AB	-0.447	2.000	5.5270%	5.5270%	11.0539%
12	AB	-0.402	2.000	-0.3983%	-0.3983%	-0.7965%
13	AB	-0.396	2.000	-0.4274%	-0.4274%	-0.8549%
14	AB	-0.367	2.000	-5.8826%	-5.8826%	-11.7653%
15	AB	-0.266	2.000	0.1192%	0.1192%	0.2383%
16	AB	-0.244	2.000	-6.0931%	-6.0931%	-12.1863%

The product of the cross term composition between fragment 1 and 2 in orbital i and corresponding orbital occupation number is the total Mulliken bond order between them contributed from orbital i . From above information we can see MO 7 and 11 are helpful for bonding, because the composition are relative large, while MO 14 and 16 are not conducive for bonding. The isosurfaces of MO 11 (left side) and 14 (right side) are shown below, it is clear that the result of composition analysis is reasonable.



4.8.2 Analysis on water by natural atomic orbital method

In this example we analyze molecular orbital composition of water by natural atomic orbital (NAO) method discussed in Section 3.10.5. NAO method has much better basis-set stability and stronger theoretical basis than MMPA class methods (such as Mulliken).

Performing NAO method requires MO coefficient matrix in NAO basis, this matrix cannot be generated by Multiwfn itself, but Multiwfn can utilize the output information containing this matrix by stand-alone NBO program or NBO module embedded in quantum chemistry softwares. The NBO 3.1 module embedded in Gaussian program is L607. Below is a Gaussian input file for water, which will output the matrix we needed.

```
# HF/6-31g* pop=nboread
```

```
Title Card Required
```

```
0 1
O          0. 00000000   0. 00000000   0. 11472000
H          0. 00000000   0. 75403100  -0. 45888100
H          0. 00000000  -0. 75403100  -0. 45888100
```

```
$NBO NAOMO $END
[space line]
[space line]
```

Where `pop=nboread` keyword indicates that the texts enclosed by `$NBO` and `$END`, namely `NAOMO`, will be passed to NBO module. `NAOMO` keyword tells NBO module to output MO coefficient matrix in NAO basis.

Assume that the output file is named as `H2O_NAOMO.out` (can be found in "example"

4 Tutorial

folder), we start Multiwfn and input:

```
8 // Enter orbital composition analysis module
8 // Enter NAO analysis function
```

You will find the default output mode is "Only show core and valence NAOs". Core and valence NAOs have one-to-one relationship with actual atomic orbitals, if the MO to be analyzed is occupied, in general we only need to concern these NAOs, while Rydberg NAOs can be ignored. Assume that we want to analyze MO 4, we input

```
0 // Show orbital composition of specific MO
4 // Analyze MO 4
```

Below information will appear on screen

NAO#	Center	Label	Type	Composition
1	1(O)	S	Cor(1S)	0. 031329%
2	1(O)	S	Val (2S)	8. 573184%
5	1(O)	px	Val (2p)	0. 000000%
7	1(O)	py	Val (2p)	0. 000000%
9	1(O)	pz	Val (2p)	84. 088900%
16	2(H)	S	Val (1S)	3. 541924%
18	3(H)	S	Val (1S)	3. 541924%

Summing up the compositions listed above: 99. 777261%

Rydberg composition: 0. 222739%

Condensed above result to atoms:

Center	Composition
1(O)	92. 693413%
2(H)	3. 541924%
3(H)	3. 541924%

According to the result, we can say for example, 2pz atomic orbital of oxygen has 84.09% contribution to MO 4 in current geometry. The contribution from the NAOs listed above (Rydberg composition is not included in present example) are also summed up to atom contribution according to which center they are attributed to.

You can further analyze specific orbital composition by inputting corresponding MO number, but here we input 0 to return to last menu. We will analyze contribution from the NAOs centered on the two hydrogens to MO 1~10.

Input -I to enter the interface for defining fragment, and then input *all* to check all NAOs, we will see:

NAO#	Atom	Label	Type	Occupancy	Energy
14	0	1	dx2y2	Ryd(3d)	0. 00073
... (Omission)					
15	0	1	dz2	Ryd(3d)	0. 00254
16	H	2	S	Val (1S)	0. 52321
17	H	2	S	Ryd(2S)	0. 00086
18	H	3	S	Val (1S)	0. 52321
19	H	3	S	Ryd(2S)	0. 00086
					0. 70497

We input a 2,3, that means all NAOs attributed to atom 2 and 3 will be added to current

fragment. Then input *q* to save and quit. From the prompt printed on screen you will know NAO 16, 17, 18 and 19 have presented in this fragment.

Then select option 1 and input *1,8*, the contribution from the four NAOs to MO 1~8 will be shown as below

Orb. #	Core	Val ence	Rydberg	Total
1	0. 000000%	0. 119072%	0. 001568%	0. 120640%
2	0. 000000%	18. 556232%	0. 054450%	18. 610682%
3	0. 000000%	26. 557472%	0. 018050%	26. 575522%
4	0. 000000%	7. 083848%	0. 012482%	7. 096330%
5	0. 000000%	0. 000000%	0. 000000%	0. 000000%
6	0. 000000%	35. 481888%	46. 831842%	82. 313730%
7	0. 000000%	27. 557888%	61. 538418%	89. 096306%
8	0. 000000%	32. 433458%	29. 568050%	62. 001508%

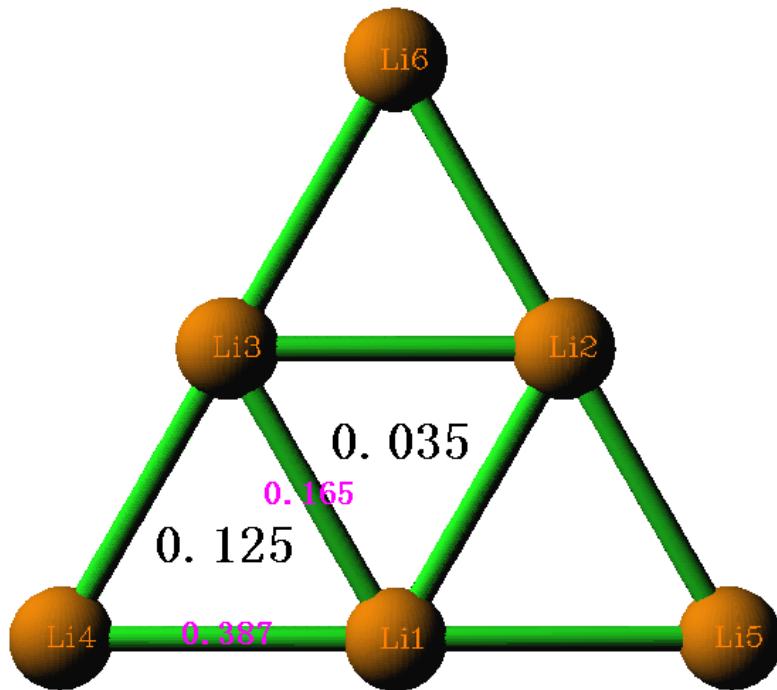
Since none of the four NAOs in the fragment is core-type, the Core term is 0% in all MOs. Valence and Rydberg terms correspond to the contribution from NAO 16, 18 and NAO 17,19 respectively. The first five MOs are doubly occupied in present system. It is clear that Rydberg NAOs have very low contribution to occupied MOs, while their contribution to virtual MOs are significant and no longer can be ignored. NAO 16 and 18 directly correspond to 1s atomic orbital of H2 and H3, so we can say the two hydrogens collectively contribute 26.56% to MO 3. However, physical meaning of Rydberg NAOs are difficult to be interpreted, although NAO 17 and 19 are centered on the two hydrogens formally, it may be questionable to say the two hydrogens contributed 32.43% (or 62.00%) to MO 8. So it is not generally recommended to use NAO method to analyze atom contribution to virtual MOs, for this case, Hirshfeld method introduced in Section 3.10.4 may be the best choice.

4.9 Three-center bond order analysis of Li₆ cluster

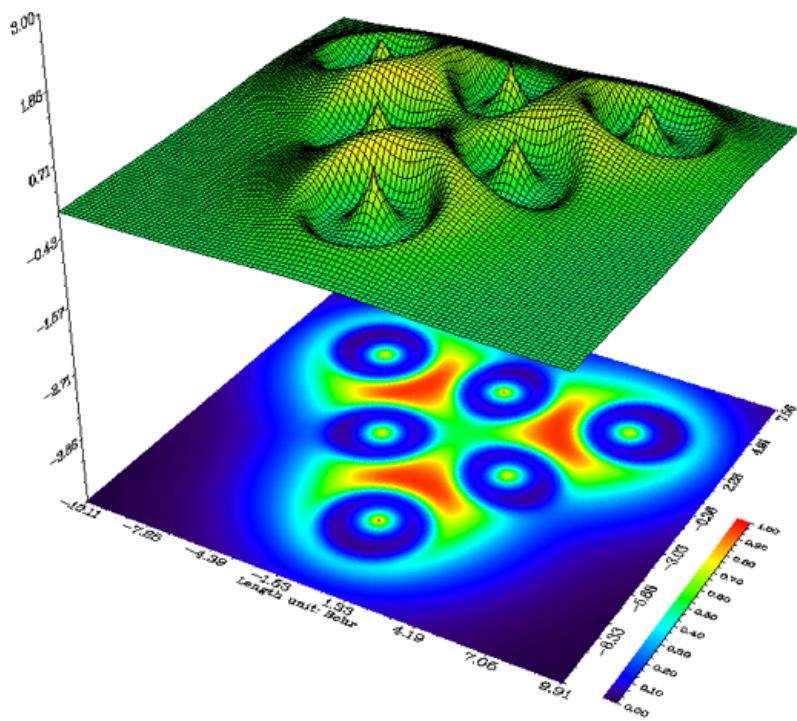
The electron structure character of complex system, such as cluster is hard to predict by simple empirical rules, we have to resort to wavefunction analysis methods. In this section we use three-center Mayer bond order to study which type of three-membered ring in Li₆ cluster is the most stable. Note that Mayer bond order analysis requires basis function information, so you have to use .fch file as input rather than .wfn file. We will calculate the three-member ring consist of Li1, Li4 and Li3 first.

```
Bakuretsu_Tenshi\Li6.fch
9 // Bond order analysis
2 // Multicenter Mayer bond order analysis
3 // Three centers
1 // Input the indices of the atoms constituted three-member ring in turn
4
3
```

Result is 0.12478846. Then repeat above procedure to calculate three-center bond order of central three-member ring, the result is 0.03517822. I marked the results in the graph below. The pink texts denote conventional two-center Mayer bond orders, you can calculate them by subfunction 1.



Evidently, the outer-side three-member rings are more stable than the central one, even with two-center bond order this conclusion can be drawn too. We can further demonstrate this viewpoint by ELF graph in molecular plane



It is clearly that electrons tend to localized in the outer-side three-member rings to stabilize them, the conclusion of real space function analysis is in fairly agreement with bond order analysis. By checking laplacian map, ELF map and electron density deformation map, you can get the same conclusion.

4.10 Plot total, partial and overlap density-of-state (DOS) graph of water molecule

I have detailed this example in Section 3.12.1, here I only gives operation steps. To plot PDOS and OPDOS, basis function informations are required, so you have to use .fch as input.

Freelancer\h2o.fch

10 // Plot TDOS/PDOS/OPDOS. If you only want to plot TDOS rather than PDOS and OPDOS, you can then select option 0 directly and ignore following commands

-1 // Define fragment 1

all // Print basis function list and find out which basis functions correspond to p-shell of oxygen, we find shell 3 and 5 should be chosen

s 3,5 // Add all basis functions in shell 3 and 5 into fragment 1

q // Quit interface and save fragment

-2 // Define fragment 2

a 2,3 // Atom 2 and 3 are hydrogens, add them into fragment 2

q

2 // Set X-axis range

-1.5,0.5 // From -1.5a.u to 0.5a.u

l // Set broadening function

1 // Use Lorentzian function. Corresponding DOS looks more smooth than the default one (Gaussian function)

0 // Plot TDOS+PDOS+OPDOS right now

The graph pops up immediately. After close the graph, you will find several options used to custom the graph.

4.11 Plot infrared spectrum for NH₃BF₃

Multiwfn can read in frequencies and intensities from output file of Gaussian vibration analysis task (“freq” keywords), NH3BF3.out in “examples” directory is such a file, both structure optimization and vibration analysis are performed at B3LYP/6-31G* level, the frequencies are already scaled by 0.9614 (“scale=0.9614” was specified in route section).

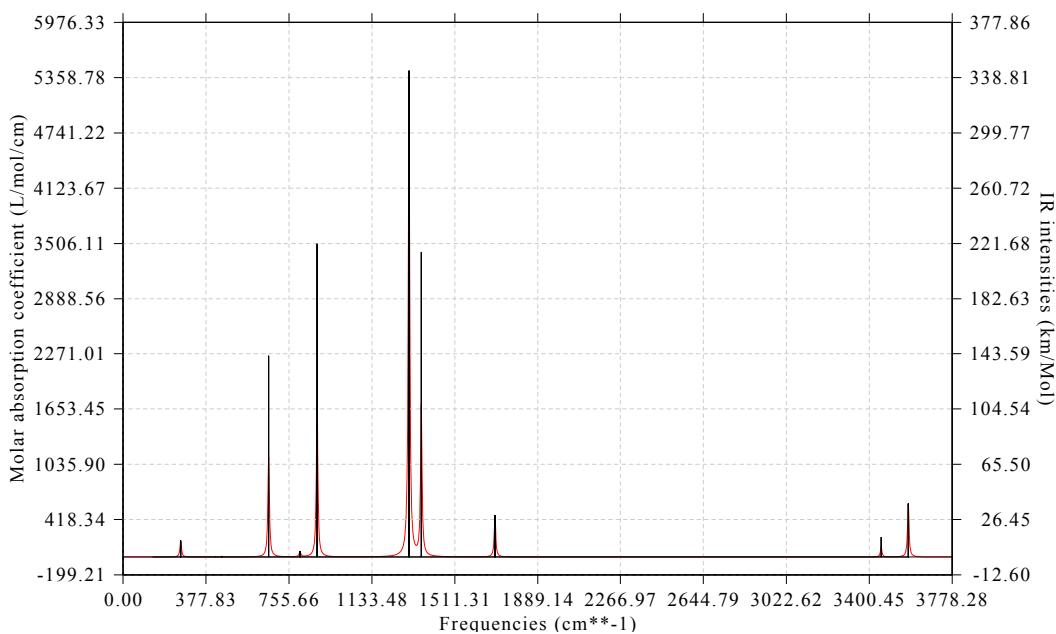
Saber\NH3BF3.out

11 // Plot IR/Raman/UV-Vis spectrum

I // Select IR spectrum

0 // Plot spectrum right now

You will get below graph



Left axis corresponds to curve (broadened data), right axis corresponds to discrete lines (original data). Plotting parameters such as full width at half maximum (FWHM), broadening function, unit and range of axes can be adjusted by corresponding options. The graph and X-Y data set of discrete lines/curve can be exported by option 1 and 2 respectively. If you find the gray dash lines are disappeared in exported graphic file, you can change “graphformat” parameter in settings.ini to “wmf” or “pdf”, then reenter Multiwfn and export the graph again.

The procedure of plotting Raman and UV/Vis spectrum is the same as plotting IR spectrum. If the quantum chemistry program you used for spectrum calculation is not Gaussian, you can manually extract frequencies and intensities from output file and then save them to a plain text file as Multiwfn input, see Section 3.13 for detail.

4.12 Quantitative analysis of molecular surface of phenol

In this example I will show the use of quantitative analysis of molecular surface by case study of phenol. The theoretical basis has been documented in Section 3.15.1 and will not be repeated here. We first analyze ESP on vdW surface:

phenol_DFT.wfn // phenol wavefunction produced at B3PW91/6-31G** level. For most systems this level can give satisfactory result. Hartree-Fock wavefunction is not recommended to use, since HF completely ignored electron correlation and hence lead to unreliable ESP distribution

12 // Quantitative analysis of molecular surface

0 // Start the analysis under default settings. By default the mapped function is ESP

Now the calculation starts. Since computing ESP is time consuming, you need to wait for a while. During the calculation some intermediate informations are printed, ordinary users don't need to concern them. Below results will be printed on screen once the calculation has been finally finished:

4 Tutorial

Global surface minimum: -0.041191 a.u. at 1.452428 3.347510 -0.008393 Ang.

Global surface maximum: 0.085761 a.u. at -1.936645 3.093464 0.021360 Ang.

Number of surface minima: 3

#	a.u.	eV	kcal/mol	X/Y/Z coordinate(Angstrom)		
1	-0.03046066	-0.828877	-19.112843	0.150202	-1.011077	-1.882004
2	-0.03046393	-0.828966	-19.114896	0.178356	-0.990252	1.877656
*	3	-0.04119057	-1.120852	-25.845433	1.452428	3.347510 -0.008393

Number of surface maxima: 5

#	a.u.	eV	kcal/mol	X/Y/Z coordinate(Angstrom)		
1	0.02275811	0.619280	14.279804	-3.345090	-2.281694	0.020475
*	2	0.08576096	2.333674	53.811572	-1.936645	3.093464 0.021360
3	0.01935782	0.526753	12.146259	0.066223	-4.286661	0.040555
4	0.01980285	0.538863	12.425498	3.340574	-2.325727	0.021485
5	0.01584879	0.431267	9.944481	3.431682	1.187721	-0.004660

===== Summary of surface analysis =====

Volume: 835.71041 bohr^3 (123.83953 Angstrom^3)

Overall surface area: 476.12414 bohr^2 (133.32836 Angstrom^2)

Positive surface area: 231.15545 bohr^2 (64.73013 Angstrom^2)

Negative surface area: 244.96869 bohr^2 (68.59823 Angstrom^2)

Overall average value: -0.00020123 a.u. (-0.12626308 kcal/mol)

Positive average value: 0.01877759 a.u. (11.78218611 kcal/mol)

Negative average value: -0.01810987 a.u. (-11.36322142 kcal/mol)

Overall variance (sigma_tot): 0.00041498 a.u. (163.38200949 (kcal/mol)^2)

Positive variance: 0.00031117 a.u. (122.51069875 (kcal/mol)^2)

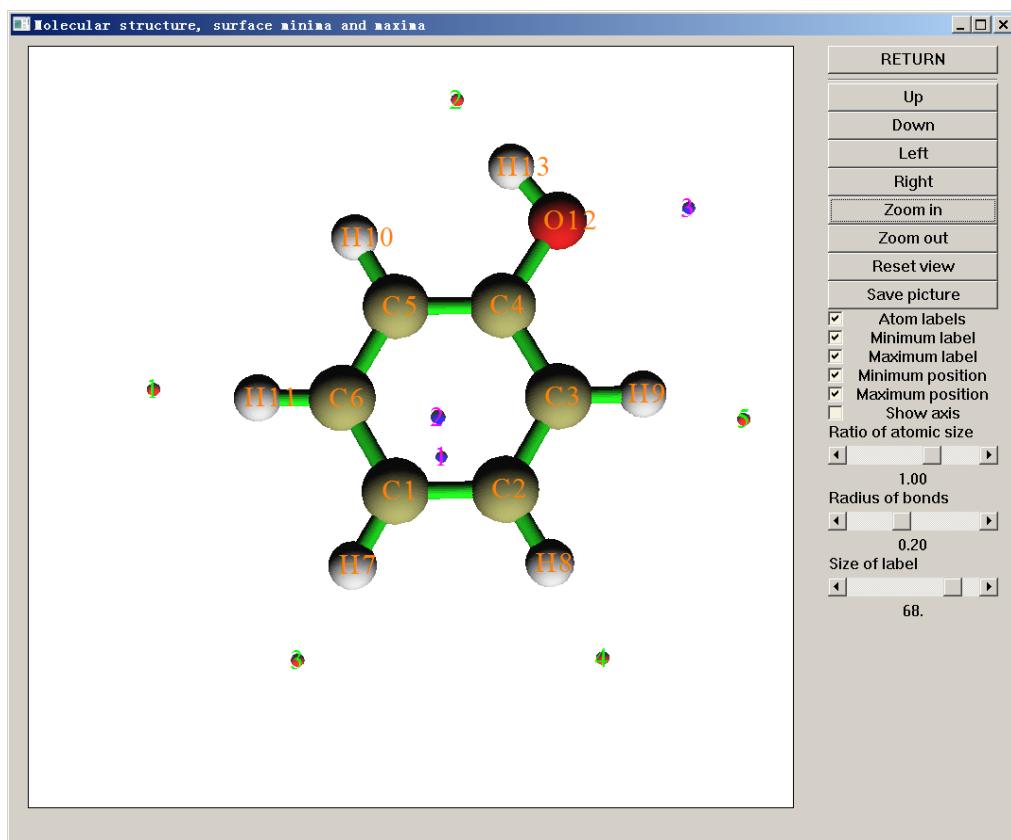
Negative variance: 0.00010381 a.u. (40.87131074 (kcal/mol)^2)

Balance of charges (miu): 0.18757896

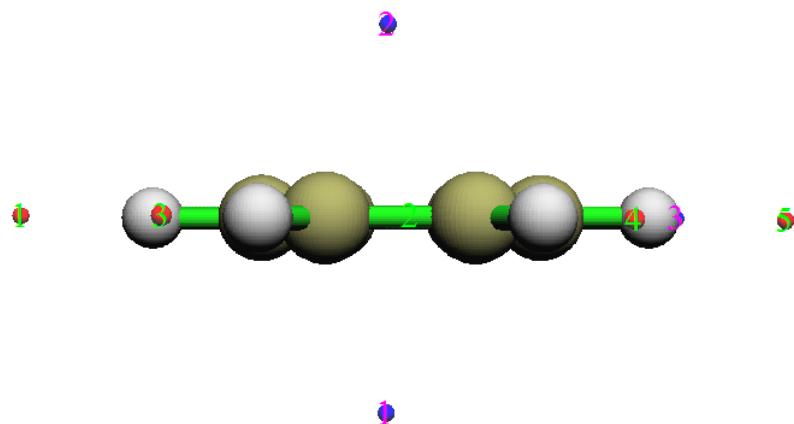
Product of sigma_tot and miu: 0.00007784 a.u. (30.64702689 (kcal/mol)^2)

Internal charge separation (Pi): 0.01842887 a.u. (11.56337916 kcal/mol)

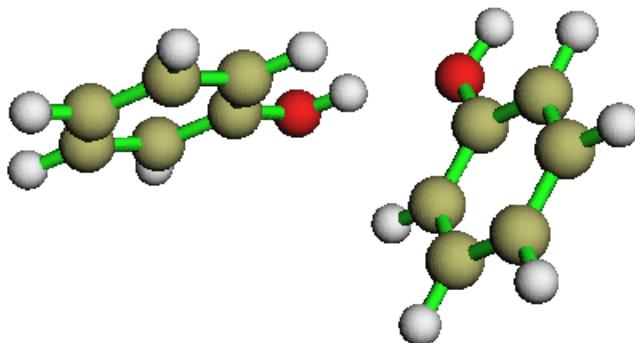
Above information includes all quantities that involving ESP mentioned in Section 3.15.1. In Multiwfn, the vdW volume can also be computed by Monte Carlo method through subfunction 3 of main function 100, see Section 3.100.3, the results match fairly well. Now select option 0 in the post-process interface to view molecular structure and surface extrema (red and blue spheres correspond to maxima and minima respectively):



In side view:



Minimum 3 (-25.84kcal/mol) is global minimum on the surface, its large negative value is owing to the lone pair of oxygen. Maximum 2 (53.81kcal/mol) is global maximum arise from the positively charged H13, the ESP at this point is much larger than that at other maxima (where the ESP ranges from 10 to 15kcal/mol). This is because the presence of oxygen, which attracted a great deal of electrons from H13. In complex, assume that only electrostatic interaction exists, monomers always contact each other in maximally ESP complementary manner. So we can expected that in phenol dimer, H13 and maximum 2 in a monomer, and O12 and minimum 3 in neighbour monomer will be in a straight line (resulting hydrogen bonding), this is the exactly situation in actual geometry of phenol dimer, see below graph. Notice that in the dimer, maximum 2 and minimum 3 shown above have cancelled each other out.



Minimum 1 and 2 (both are -19.11kcal/mol) are local minimum on the surface, mainly arise from the abundant π electrons above and below the ring. It is well known that nucleophiles always prefer to attack the atom which has very low ESP around it, so C1 should be a ideal reaction site for nucleophilic reaction. This conclusion is partially consistent with the general knowledge that hydroxyl is a ortho-para directing group. However, although the global minimum is closest to O12, O12 is not the nucleophilic reaction site, this contradiction reveals the inherent limitation of ESP analysis method.

Next, we analyze average local ionization energy \bar{I} on phenol vdW surface.

-1 // Return to the parameter setting interface from post-process stage

2 // Reselect mapped function

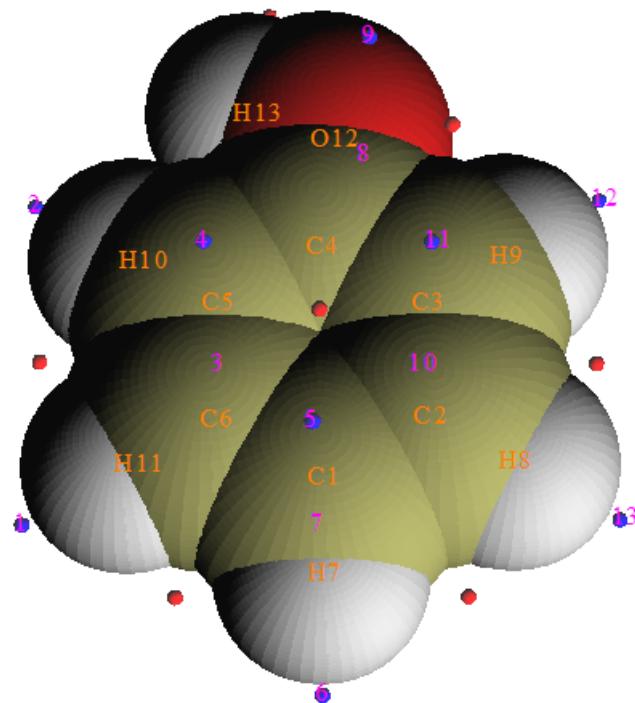
2 // Choose \bar{I} as mapped function

0 // Start the surface analysis.

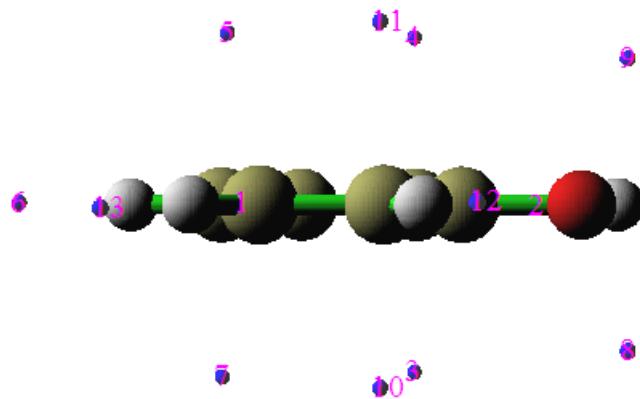
Since calculation of \bar{I} is much simple than ESP, the calculation will be finished soon.

Unlike surface analysis for ESP, at this time only vdW volume, surface area and average of \bar{I} on vdW surface are outputted alongside extrema information.

Choose 0 to visualize extrema, meanwhile drag the "Ratio of atomic size" scale bar to 4.0, we will see:



In side view (For clarity, only minima are shown)



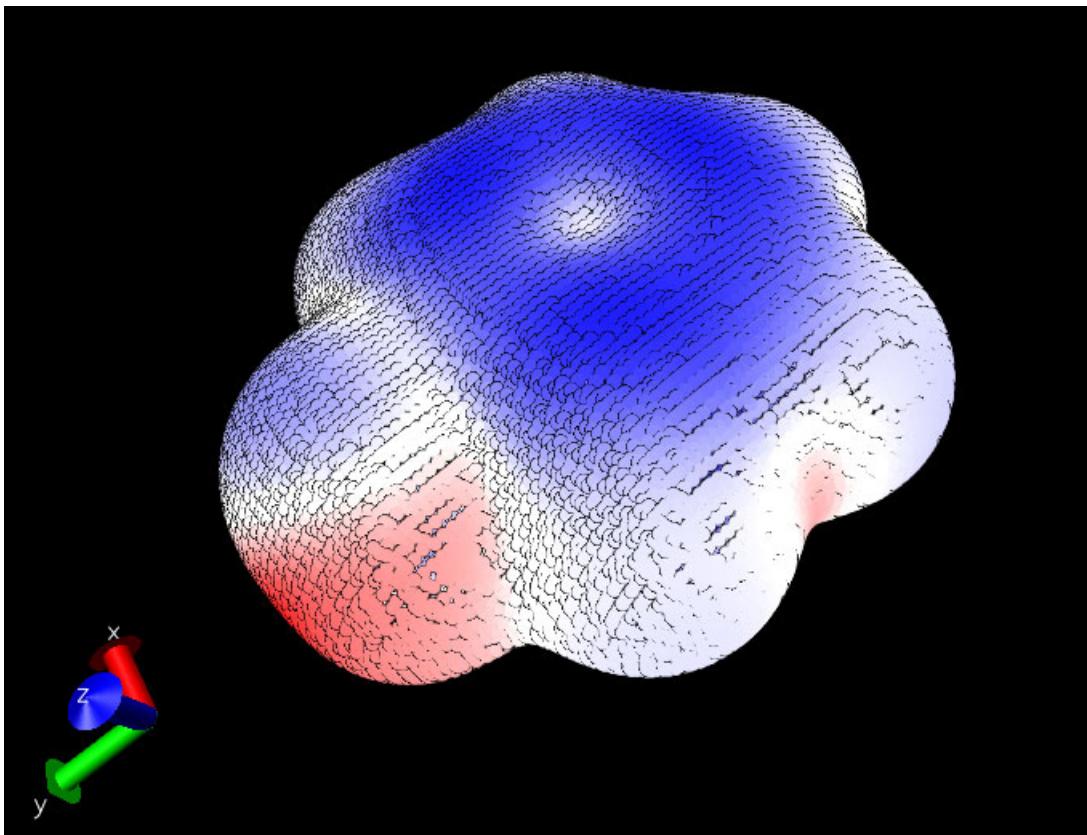
Low value of \bar{I} suggests the electron at this position is not tightly bounded, the site with lowest \bar{I} on vdW surface is usually recognized as the most vulnerable site to electrophilic or to free radical attack. The regions easy to be polarized such as π electron and lone pair generally have corresponding surface minima of \bar{I} . In present instance, minimum 8 and 9 correspond to lone pair of O12, both the \bar{I} value are 10.59eV. Minimum 4,5,11, and 3,7,10 correspond to π electrons, all of them the \bar{I} values are about 8.9eV and can be viewed as degenerated global minima. Worthnotingly, the minima above and below the conjugated ring only present at ortho- and para-carbon. These observations perfectly explained the effect of hydroxyl as

a ortho-para directing group. Since \bar{I} at minimum 8 and 9 are obviously larger than \bar{I} at the minima around the carbon ring, oxygen is not a vulnerable site for nucleophilic reaction.

By utilizing some third-part visualization softwares, variance of mapped function on the entire vdW surface can be visualized. Now let's plot \bar{I} distribution on vdW surface of phenol by VMD program, which can be freely downloaded at <http://www.ks.uiuc.edu/Research/vmd/>.

In the post-process interface in above example, select option 6 to export surface vertices to *vtx.pdb* in current folder. Boot up VMD, drag the pdb file into VMD main window, and then select "Graphics"->"Representations...". Input *name C* in "Selected Atoms" text box and press ENTER button (because each carbon atom in the pdb file corresponds to a surface vertex), set "Coloring method" to "Beta" (this is because Beta field in the pdb file corresponds to mapped function value), set "Drawing method" to "Points", set "Size" to 15 or larger. After that choose "Graphics"->"Colors..."->"Color scale", set "Method" to "BWR". The final image is shown below.

Blue, white and red correspond to \bar{I} varies from small value to larger value. The surface extrema found by Multiwfns can be exported as pdb file by option 2 at post-process stage, therefore they can also be conveniently visualized in a similar manner in VMD.



4.13 Process grid data

Main function 13 includes a bunch of subfunctions, by using them you can process the grid

data loaded from Gaussian-type cube file (.cub), or the grid data directly generated by main function 5 of Multiwfn. In this section I present several simple applications, please play with other subfunctions by yourself.

4.13.1 Extract data points in a plane

In this example we extract average XY-plane data between Z=28 and Z=32 angstrom to a plain text file.

dens.cub // A cube file generated by Multiwfn or by some external programs, since cube file is generally large, it is not provided in "example" folder. You can also use the grid data generated internally by Multiwfn instead, that is use main function 5 to calculate grid data first and then choose 0 to return to main menu (the just generated grid data is present in memory)

```
13 // Process grid data
5 // Extract average plane data
28,32 // Range of Z (in angstrom)
```

Now the data points are exported to output.txt in current folder, including X,Y coordinates and value. You can import this file to plotting software such as sigmaplot to draw plane graph.

Another example, we extract data point on the plane defined by atom 4,6,2.

```
dens.cub
13 // Process grid data
```

8 // Output data in a plane by specifying three atom indices. This function is commonly used to extract tilted plane, if the plane is parallel to XY, YZ or XZ, you should use function 1,2 or 3 instead respectively

0 // Use automatically determined tolerance distance. If vertical distance between any point and the plane you defined is smaller than tolerance distance, then the point will be outputted.

1 // Project the data points in the plane you defined to XY plane, so that you can directly import the outputted file to plotting software to draw plane graphs

Now the data value along with coordinates is exported to output.txt in current folder.

Notice that Multiwfn doesn't do interpolation during plane data extraction, hence if the quality of grid data is not fine enough (namely spacing between points is large), then the extracted plane data will be sparse (especially severe for the plane not parallel to XY, YZ or XZ plane).

4.13.2 Perform mathematical operation on grid data

Assume that we have two cube files MO1.cub and MO2.cub, each of them records wavefunction value of a orbital. In this example we will generate a cube file containing total electron density from these two orbitals.

```
MO1.cub // Load one of the two cubes file first
13 // Process grid data
11 // Grid data calculation
```

10 // Perform $A^2+B^2=C$ algorithm, where A is present grid data (MO1.cub), B is another cube file (MO2.cub), C is updated grid data. We perform this algorithm is because according to Born's interpretation, square of orbital wavefunction value is density probability.

MO2.cub // Load another cube file. This cube file must has identical grid setting as MO1.cub

After computing, the grid data stored in memory, namely A in above equation, has been updated to C.

```
0 // Output the updated grid data
totdes.cub // Filename of new cube file, which contains total electron density from those two orbitals
```

4.13.3 Scaling numerical range of grid data

The numerical range of ELF function is [0,1], in this example, we scale its numerical range to [0,65535] (which is value range of unsigned 16bit integer). We first compute ELF grid data in Multiwfn as described in Section 4.5.1, and then input

```
0 //Return to main menu from post-process interface of grid data calculation
13 // Process grid data
16 // Scale data range
0,1 // Original data range
0,65535 // The range after scaling. Please read Section 3.16.12 for the detail of scaling algorithm.
```

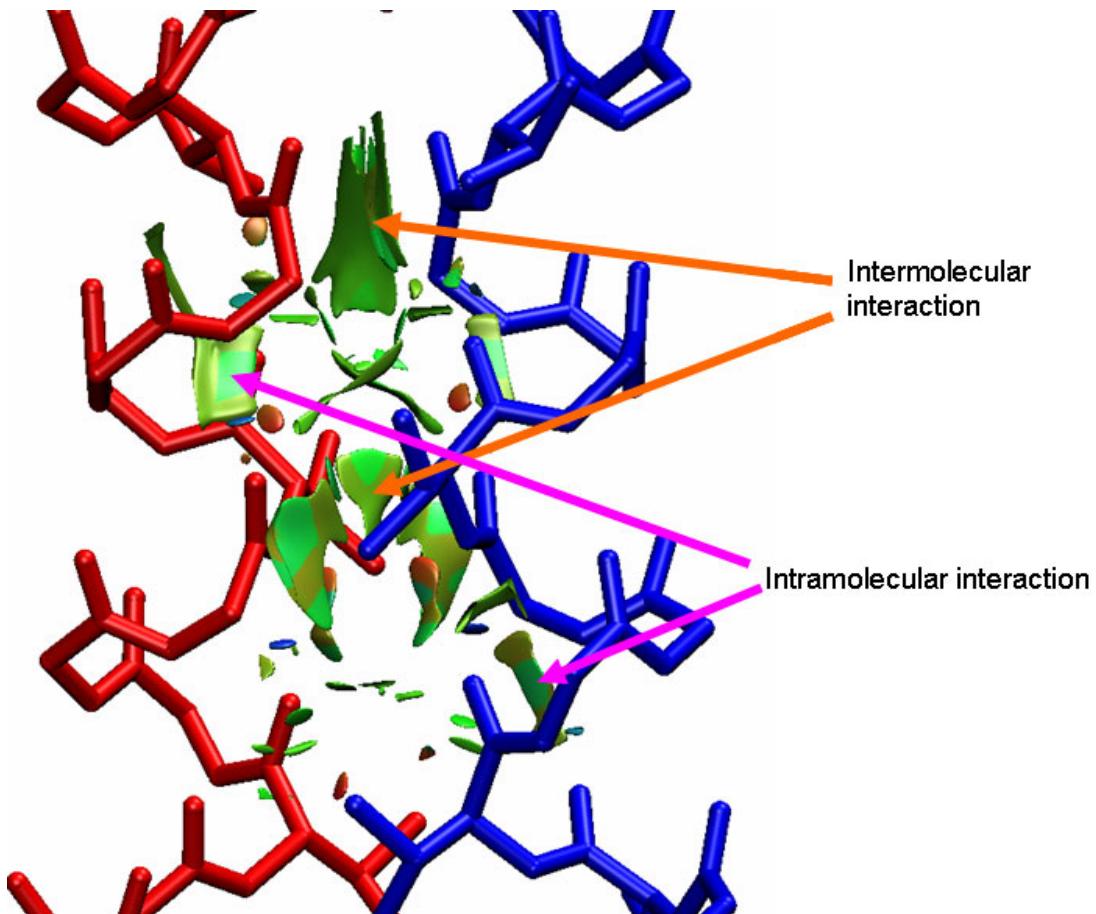
Now the grid data has been scaled. You can choose function 0 to export the updated grid data to Gaussian cube file, or extract plane data to plain text file by corresponding functions.

4.13.4 Screen isosurfaces outside overlap region of two fragments

Sometimes we don't want all isosurfaces in whole space are shown, because too many isosurfaces will confuse our vision. This is especially true during analysis of inter-molecular interaction by reduced density gradient, what we want to study is only those isosurfaces inside inter-molecular region. In order to screen isosurfaces in other regions, we can set the value of grid points outside superposition region of scaled vdW regions of two molecules to a very large value (at least larger than maximum value in current grid data). In this example I give you a practical example.

The so-called "scaled vdW regions" is the superposition region of scaled vdW spheres of all atoms in the fragment. While the "scaled vdW sphere" denotes the sphere correponding to the scaled vdW radii.

Below is a segment of dimeric protein plotted by VMD program (you will know how to draw a similar picture after you have read Section 4.100.1), red and blue represent backbone structure of the two chains respectively. Isosurfaces of reduced density gradient exhibit weak interaction region. However, these isosurfaces includes both intermolecular and intramolecular parts, they are interwinded and arise difficulty in visual study of weak interaction between the two chains.



In order to screen those intramolecular isosurfaces, we will use subfunction 14 in main function 13 of Multiwfn. First, we need to prepare two atom list files for the two chains (a chain corresponds to a fragment). atmlist1.txt includes atom indices of chain 1, the head and tail parts of the file:

```

159 <--- Total number of atoms in chain 1
1 <--- Atom index (relative to the whole system) of the first atom in this
chain
2 <--- Atom index (relative to the whole system) of the second atom in this
chain
3
...
159 <--- atom index (relative to the whole system) of the last atom in this
chain

```

Similarly, atmlist2.txt defines atom list for chain 2, its head and tail parts:

```

159 <--- Chain 2 have 159 atoms too
160 <--- Atom index (relative to the whole system) of the first atom in this
chain
161 <--- Atom index (relative to the whole system) of the second atom in
this chain
162
...
318 <--- atom index (relative to the whole system) of the last atom in this

```

chai n

Then boot up Multiwfn and input:

RDG.cub // The cube file of reduced density gradient corresponding to above graph

13 // Process grid data

14 // Set value of the grid points outside overlap region of the scaled vdW regions of the two fragments

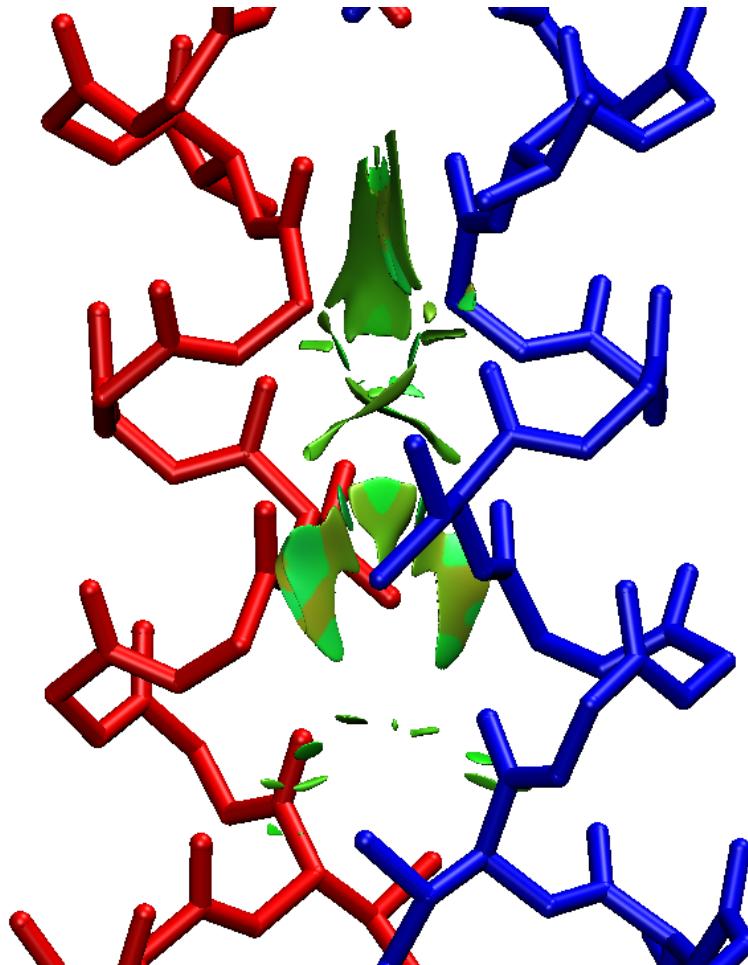
1.8 // The value for scaling vdW radius. In your practical studies, you may need to adjust this value again and again

1000 // Set value of those grid points to 1000, this value is large enough

atmlist1.txt // The name of the atom list file for chain 1

atmlist2.txt // The name of the atom list file for chain 2

Wait for a while, the grid data will be updated. Then choose function 0 to export it as cube file. Using this new cube file to redraw above picture, we find all of intramolecular isosurfaces have disappeared, the graph becomes more clear.



4.100 Other functions

4.100.1 Visual study of weak interaction

This is a very important application of Multiwfn, the theory has already been detailed in *J.*

Am. Chem. Soc., 132, 6498 and my article *Visual research of weak interaction by Multiwfn* (in chinese, can be downloaded at website of Multiwfn), so here I only sketch the key points. By the way, NCIplot written by Weitao Yang group is a program dedicated to this propose, however if you have used both NCIplot and Multiwfn, you will find the latter is much more convenient and more efficient to use.

Use isosurface to represent weak interaction region

The first thing is to find a way to distinguish weak interaction region from other regions. From the table below we can know if only the regions where the value of reduced density gradient (RDG) function is in the range of 0 to medium are reserved, then “around nuclei” and “edge of molecule” regions will be shielded. The RDG function is defined as

$$RDG(\mathbf{r}) = \frac{|\nabla\rho(\mathbf{r})|}{\rho^{4/3}(\mathbf{r})}$$

	Around nuclei	Around chemical bond	Weak interaction region	Edge of molecule
$ \nabla\rho(\mathbf{r}) $	Large	0~Minor	0~Small	Very small~Small
$\rho(\mathbf{r})$	Large	Medium	Small	0~Small
RDG(\mathbf{r})	Medium	0~Minor	0~ Medium	Medium ~Very large

Then from remained regions, if we only extract the region where $\rho(\mathbf{r})$ is small rather than middle, the region “around chemical bond” will be shielded and only weak interaction region is eventually remained. Now I use phenol dimer to exemplify this analysis method.

David_Koller\MIku\PhenolDimer.wfn

5 // Generate grid data

13 // RDG function

7 // Use middle point of two atoms as center of grid data, this way to define spatial scope is very suitable for weak interaction analysis

1,14 // The indices of the two atoms are set to 1 and 14, because from molecular structure (see below graph) we can estimate that the weak interaction region is between C1 and C14

40,40,40 // The weak interaction region is small, so 40*40*40 grid is fine enough

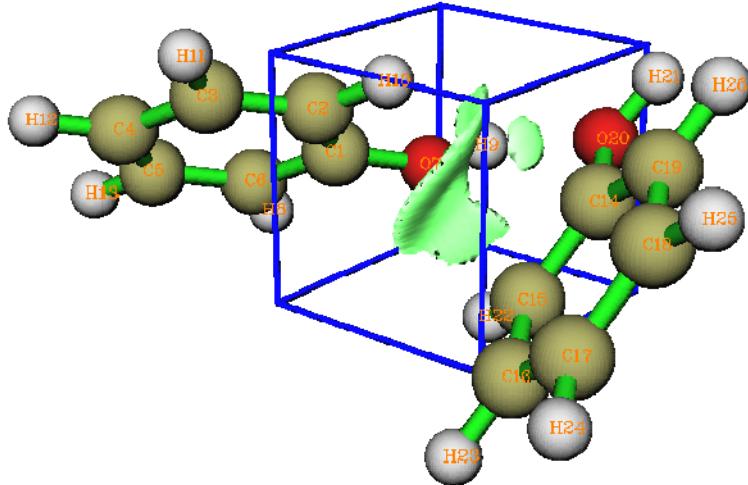
3,3,3 // Size extensions in all directions are 3 bohr, so the volume of spatial scope is $(2*3)^3=216$ bohr³. This extension value is generally too small, but enough for present study.

4 // After Multiwfn finished the calculation of grid data, set isovalue for viewing RDG isosurface

0.5 // Set isovalue to 0.5, this is recommended value for most cases, if too small, then isosurface is too small, if too big, then unwanted “Around nuclei” and “Around chemical bond” region will appear

-I // Show the isosurface

You will get below graph:

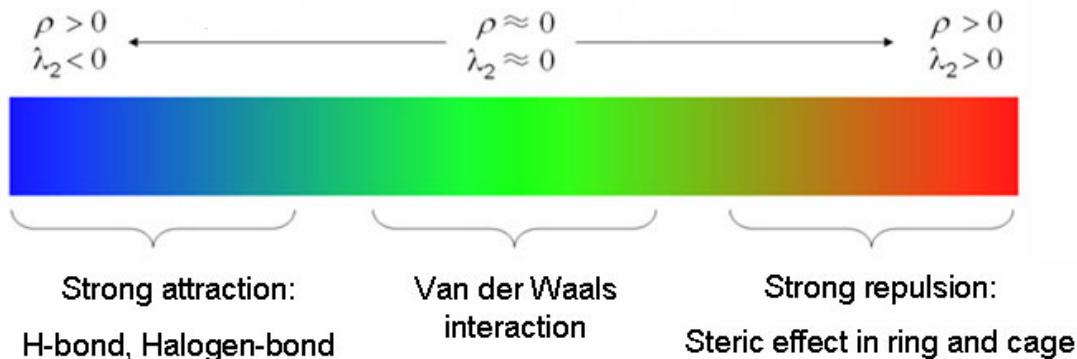


The isosurface represents the weak interaction region between phenol dimer very clearly. Notice that by default, RDG function is set to an arbitrary large value (100.0) where electron density is larger than or equal to 0.05, so that the isosurface of the regions “around chemical bond” will be shielded, the threshold is adjusted by “RDG_maxrho” parameter in settings.ini, 0.05 is suitable for most cases.

The cubic blue frame shows spatial scope of grid data, only appears when “Show data range” is selected in GUI window. Because the extension distance from center of grid data is set to 3.0 bohr, so the side length is $2 \times 3 = 6$ bohr.

Discriminate weak interaction types by filling color to RDG isosurface

In Bader’s QTAIM theory, the (3,-1) type critical point indicates electron density is aggregated at this point, appears in chemical bond path or between the atoms which have attractive interaction; while (3,+1) type critical point indicates electron density is depleted at this point and displays steric effect (also called nonbond overlap), generally appears in the center of ring system. The criterion for distinguishing (3,-1) and (3,+1) is the second largest eigenvalue of Hessian matrix of electron density (called λ_2 here), if λ_2 exceed zero, then (3,+1), else (3,-1). Besides, the strength of weak interaction has positive correlation with electron density ρ in corresponding region, van der Waals interaction regions always have very small ρ , while the regions correspond to strong steric effect and hydrogen bond always have relative large ρ . So we can define a real space function $\text{sign}(\lambda_2(\mathbf{r}))\rho(\mathbf{r})$, that is the sign of λ_2 multiples ρ . If we use different colors to represent this function value as below and fill them to RDG isosurfaces, we can not only know where is weak interaction region, but also know the interaction type.



Unfortunately, current Multiwfn doesn't support plotting color-filled isosurface graph, I recommend you to generate grid data for $\text{sign}(\lambda_2)\rho$ and RDG simultaneously by subfunction 1 of function 100 of Multiwfn, and then plot the graph by VMD, which is a free software and can be downloaded at <http://www.ks.uiuc.edu/Research/vmd>. I still use phenol dimer as example, this time I don't only show you the characteristic of weak interaction region, but also ring center region, so the spatial scope of grid data should be set to cover entire dimer.

examples\PhenolDimer.wfn

100 // Other functions

1 // Generate grid data for two real space functions in same spatial scope

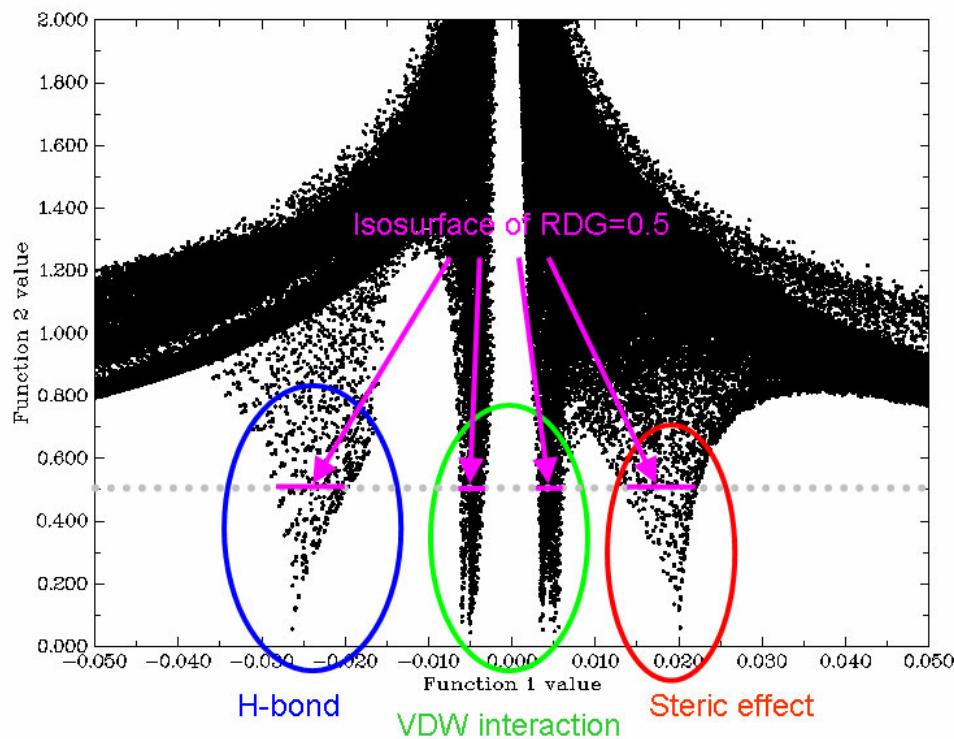
15,13 // Define $\text{sign}(\lambda_2)\rho$ as function 1 and RDG as function 2

-10 // Set extension distance in all directions

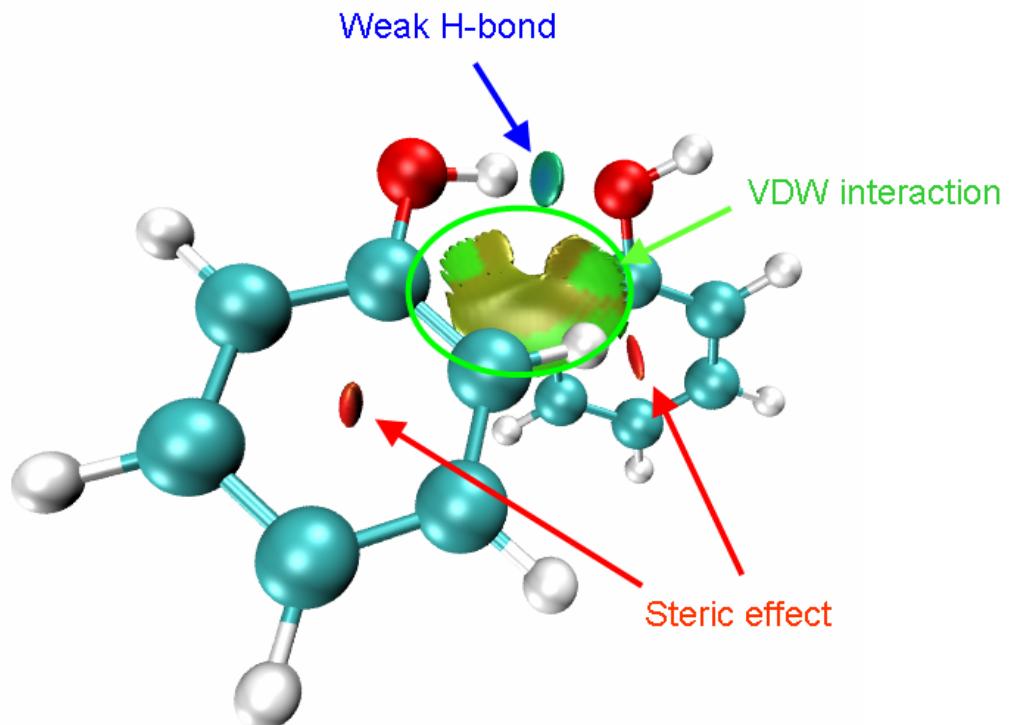
0 // Because weak interaction regions only appear in internal region of present system, we don't need to leave a buffer region at system boundary, so we set the extension distance to 0 bohr here.

2 // About 512000 points. Because the spatial scope is larger than last example, we need more grid points

Wait several minutes for the calculation. I will first show you the characteristic of different type regions in scatter graph, I think it is helpful to understand the internal mechanism. Select option -1, a scatter graph pops up (you can also select option 1 to export this graph). The X-axis and Y-axis are $\text{sign}(\lambda_2)\rho$ and RDG functions respectively, each point in the graph corresponds to each grid point. There are four spikes, and the points in their peaks are just critical points in AIM theory, if you draw a horizontal line, then the segments intersected with spikes are just RDG isosurfaces. So RDG analysis method can be seen as an extension of AIM theory for visual research. The spikes can be classified to three type, I marked them by blue, green and red circle, see below.



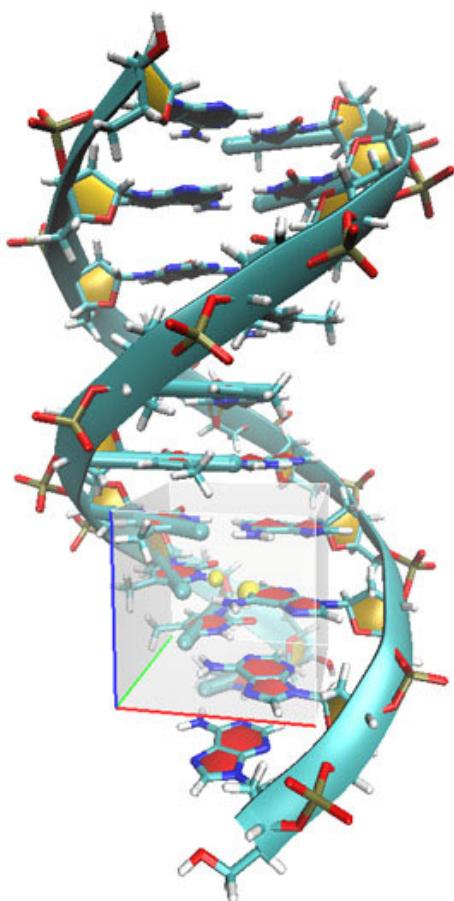
Now, select option 3 to export grid data of $\text{sign}(\lambda_2)\rho$ and RDG to func1.cub and func2.cub in current directory, then copy them and RDGfill.vmd in “examples” folder to VMD install directory, RDGfill.vmd is a script file of VMD. After that boot up VMD (here I use version 1.8.7), select “file”-“Load state”, choose “RDGfill.vmd”, you will see the graph below on OpenGL window:



From the color-filled RDG isosurface, we can identify different type regions by color. Recall the color scale bar I showed you, the more blue means the stronger interactive interaction, the elliptical slab between oxygen and hydrogen shows light blue color, so we can conclude that there is a hydrogen bond, but not strong. The interaction region marked by green circle can be identified as vdW interaction region, because the filled-color is green or light brown, that means the density electron in this region is low. Obviously, the regions in the center of rings show strong steric effect, since they are filled by red.

Promolecular approximation

Calculation of wavefunction and grid data for macrosystem is very time-consuming, however, the electron density can be approximately constructed by superposing electron densities of free-stated atoms (so-called “Promolecular approximation”), at this time the results of RDG analysis are shown to be still reasonable, the B3LYP/6-31G* densities of free-stated atoms (for first three rows) are built-in data of Multiwfn. To active promolecular approximation, simply use function 14 instead of function 13 and use function 16 instead function 15 when you selecting real space function. Since only atom coordinates information are required when promolecular approximation is used, you can use .pdb file as input. As an example, let's do RDG analysis for a DNA molecule with 10 base-pairs, what we are interested in is only the region in the box, see below.

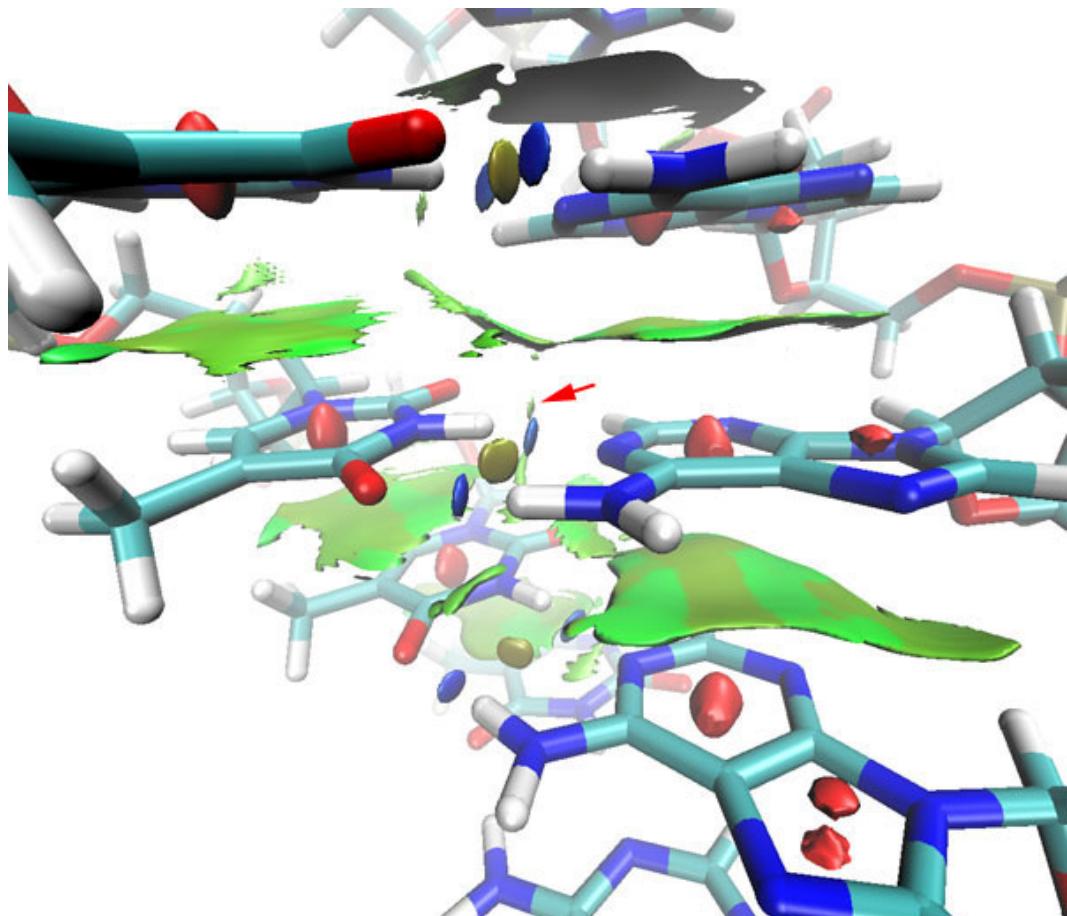


```

Krysof\DNA.pdb
100
I
16,14 // sign( $\lambda_2$ ) $\rho$  and RDG function with promolecular approximation
7
84,565 // Use middle point of atom 84 and 565 as center. You can view molecule structure in
main function 0 in advance to find two proper atoms used to define center
120,120,120 // Because the spatial scope of grid data is large, we need more grid points
9,9,9 // Set the extension distances in all directions to 9 bohr

When calculation is finished, export func1.cub and func2.cub and then plot the graph by
VMD as the procedure described in last section. From the graph you will see the default isovalue
0.5 is not suitable for plotting RDG isosurface when promolecular approximation is used, so you
need to adjust it manually, choose "Graphics"- "Representation", select the representation whose
style is "Isosurface", then reset isovalue by inputting expected value in text box or dragging scale
bar, for present system I recommend 0.25. For better visualization, you'd better adjust "cue mode"
and "Cue Start/End" in "Display"- "Display settings...". Finally, you will get the graph below

```



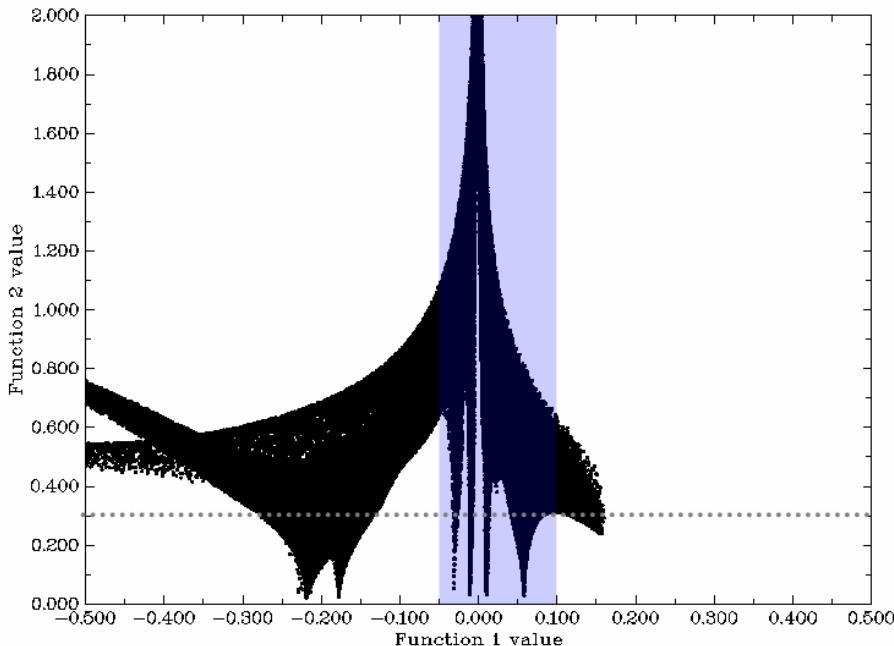
It is clear that there are π - π stacking interactions between neighbour base pairs (big pieces), and there are two strong hydrogen bonds within each base-pairs. The region pointed by red arrow seems to be hydrogen bond because it connects hydrogen and oxygen, however since the

filled-color is green, we can conclude that this is only vdW interaction.

Notice that, by default RDG value will be set to 100.0 where ρ is larger than 0.1 if promolecular approximation is used, this value may not be suitable for certain circumstance, you can set the default threshold by “RDGprodens_maxrho” in settings.ini.

Interactively set RDG value where $\text{sign}(\lambda_2)\rho$ is in specific range

Multiwfn allows you to interactively set RDG value where $\text{sign}(\lambda_2)\rho$ is in specified range. Here I use phenol dimer in promolecular approximation as example. Firstly, set “RDGprodens_maxrho” to 0.0 to disable automatic truncation, then generate grid data of RDG and $\text{sign}(\lambda_2)\rho$ for present system, after the calculation is finished, if you set X-axis range from -0.5 to 0.5 and then plot scatter graph, you will see (blue box and dash line are added by external tools)



Assume that you want to plot the RDG isosurfaces with isovalue of 0.3 only for weak interaction region, you should screen the region outside blue box first to avoid dashed line intersecting with the spikes around $\text{sign}(\lambda_2)\rho=-0.2$ (corresponding to chemical bond regions) and rightmost tail, that is we should set RDG function to a large value (e.g. 100) where $\text{sign}(\lambda_2)\rho$ is less than -0.5 or larger than 0.1. Now we choose option -3, input -0.05, 0.1 and 100 in turn, then draw scatter graph again, you will find only the region where $\text{sign}(\lambda_2)\rho$ is in -0.05~0.1 is reserved.

4.100.2 Calculate kinetic energy and nuclear attraction potential energy of phosgene by numerical integration

I suggest you read Section 3.100.4 first, the aim of this example is to show you the usefulness and universality of numerical integration. In quantum chemistry program, kinetic energy and nuclear attraction potential energy integration are evaluated analytically, analytical method has advantage in both speed and accuracy. In Multiwfn you can evaluate them by general numerical

integration function. Local kinetic energy (or called “kinetic energy density”) is a built-in function, we first integrate this function in whole space to get total kinetic energy.

COCl2.wfn // HF/6-31G wavefunction*

100

4 // Integrate a function in whole space

6 // Hamiltonian kinetic density K(r)

The result is 1031.1250, which is very close to the Gaussian outputted value 1031.1107.

Because I don’t want the list of real space function becomes lengthy, so rarely used functions such as local nuclear attraction potential energy are not chosen as built-in function, however you can easily make these functions available by hacking source code, please consult Appendix 2. First, find “userfunc” routine in function.f90, modify the default content to *userfunc = -nucesp(x,y,z)*fdens(x,y,z)*, then recompile Multiwfn, redo above procedure but select function 100 as integrand, you will get nuclear attraction potential energy -2839.1689, the value outputted by Gaussian is -2839.1629, they are rather close. If you want to obtain nuclear attraction potential energy contributed from a certain orbital, use main function -1 to reserve corresponding orbital and delete other orbitals, then do the integration.

To further illustrate the flexibility of numerical integration function in Multiwfn, assume that you want to calculate expectation of r^2 operator, you can modify the content of “userfunc” routine to *userfunc = (x*x+y*y+z*z)*fdens(x,y,z)*, then do the integration you will get 444.6514, which is in agreement with Gaussian outputted value 444.6524 (behind “Electronic spatial extent (au): <R**2>”).

4.100.3 Perform simple energy decomposition by using combined fragment wavefunctions

The principle of the simple energy decomposition has been introduced in Section 3.100.8, here we try to use the method to analyze components of interaction energy between NH₃ and BH₃ in adduct NH₃BH₃ at HF/6-31G* level. The Gaussian input files refered in this section can be found at "frag" subdirectory in "example" folder. The Gaussian version used here is G03W E.01, for other version, the energy composition you get may be slightly different.

(1) Perform structure optimization task for NH₃BH₃, and then create a new input file with the optimized coordinate, named NH3BH3.gjf.

(2) Copy NH3BH3.gjf as NH3.gjf and BH3.gjf, then remove BH3 fragment in NH3.gjf, remove NH3 fragment in BH3.gjf. Add "pop=full nosymm" keywords in both NH3.gjf and BH3.gjf.

(3) Run the two fragment input files by Gaussian respectively to generate NH3.out and BH3.out.

(4) Boot up Multiwfn, input following commands

NH3.out //Fragment 1

100

8

2 // There are two fragments in total

BH3.out //Fragment 2

Multiwfn generates new.gjf in current folder, this is the Gaussian input file for NH₃BH₃ with

fragment-combined wavefunction as initial guess.

(5) Make sure "nosymm" keyword is presented in new.gjf. In order to output SCF energy in each iteration, change "#" to "#P". "pop=full" keyword can be deleted.

(6) Run new.gjf by Gaussian to generate new.out.

From NH3.out, BH3.out and new.out, following informations can be found

E(NH₃) = -56.184295 a. u.

E(BH₃) = -26.368196 a. u.

E(NH₃BH₃) = E(SCF_I last) = -82.611815 a. u.

E(SCF_1st) = -82.535321 a. u.

Note that E_SCF_1st is the SCF energy in the first cycle of SCF process. According to the equations shown in Section 3.100.8, we found

$\text{delta_E_tot} = E(\text{NH}_3\text{BH}_3) - E(\text{NH}_3) - E(\text{BH}_3) = -0.059324 \text{ a.u.} = -155.76 \text{ KJ/mol}$

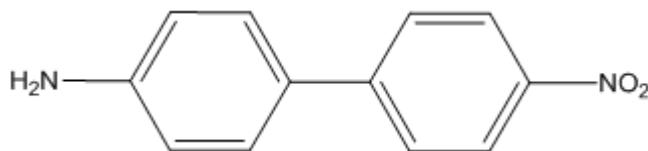
$\text{delta_E_polar} = E(\text{SCF_last}) - E(\text{SCF_1st}) = -0.076494 \text{ a.u.} = -200.83 \text{ KJ/mol}$

$\text{delta_}(E_{\text{els}} + E_{\text{Ex}}) = \text{delta_E_tot} - \text{delta_E_polar} = 0.017170 \text{ a.u.} = 45.08 \text{ KJ/mol}$

That is the total interaction energy between NH₃ and BH₃ is -155.76 KJ/mol, the polarization energy -200.83 KJ/mol stabilized the adduct, while the sum of electrostatic and exchanging energy destabilized the adduct by 45.08 KJ/mol.

4.100.4 Analyze charge-transfer during electron excitation

In this instance we will analyze charge-transfer (CT) between the first singlet excited state and ground state of the molecule shown below in ethanol solvent, which will be referred as P2. The related theory has been introduced in Section 3.100.10.



Since the .wfn files corresponding to the excited and ground state are huge, they are not provided. Instead, the input files of Gaussian for generating the two .wfn files are provided in “example” folder. I would like to remind you once again, the geometries in the wavefunction files of the two states must be exactly identical, otherwise the result is meaningless!

First, we calculate grid data of electron density variation $\Delta\rho$.

CT\extP2.wfn // Excited state wavefunction file

5 // Generate grid data

0 // Set custom operation

I // Electron density

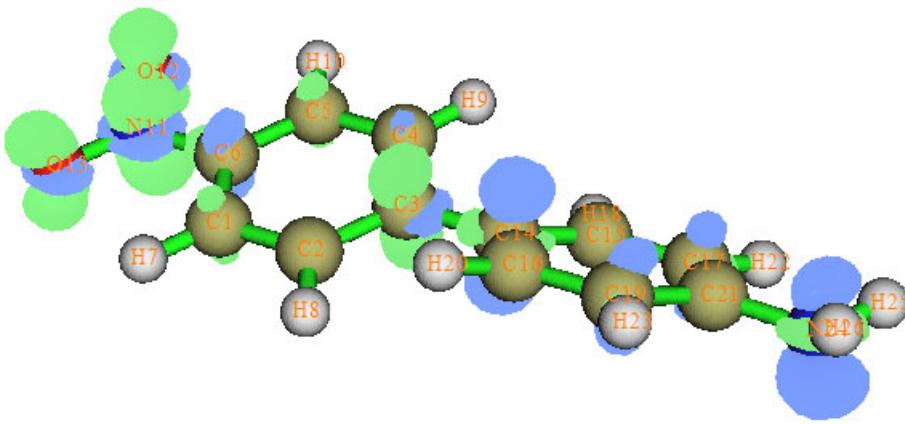
-,CT\basP2.wfn // Ground state wavefunction file. Corresponding density will be subtracted

from the excited one to generate $\Delta\rho$

I // Electron density

2 // If the system is much larger than present one, more grid points are required.

Once the calculation is normally terminated, you can choose option -1 to view the electron density variation before and after excitation (Default value is too large, 0.005 is recommended as isovalue).



However this difference graph is not intuitive, positive and negative parts intertwined together, as we will see C₊ and C₋ functions will make the picture much more clear.

0 // Return to main menu

100 // Utilities

10 // A special routine for analyzing CT

The following informations are displayed immediately

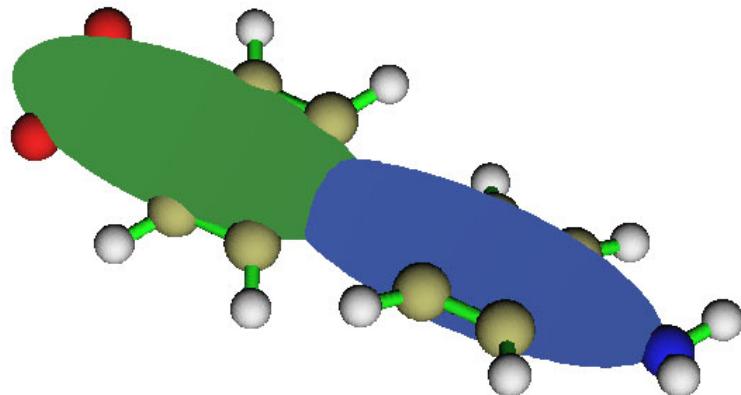
```

Transferred charge (positive and negative): 0.844 -0.844
Barycenter of positive charge in x, y, z: -5.024 -0.001 0.000
Barycenter of negative charge in x, y, z: 4.335 -0.018 -0.055
Distance of charge-transfer: 9.359 bohr ( 4.953 Angstrom )
Dipole moment between two states(a. u.): -7.896 0.014 0.046 Norm:
7.896
Dipole moment between two states(Debye): -20.070 0.035 0.117 Norm:
20.070
RMSD of positive charge in x, y, z: 5.656 2.362 1.551 Total: 6.323
RMSD of negative charge in x, y, z: 6.217 2.162 1.664 Total: 6.789
H index in x, y, z (bohr): 5.936 2.262 1.608 Norm: 6.553
H index in x, y, z (Angstrom): 3.141 1.197 0.851 Norm: 3.468
t index in x, y, z (bohr): 3.423 -2.245 -1.553 Norm: 4.378
t index in x, y, z (Angstrom): 1.811 -1.188 -0.822 Norm: 2.317
Overlap integral between C+ and C-: 0.742365

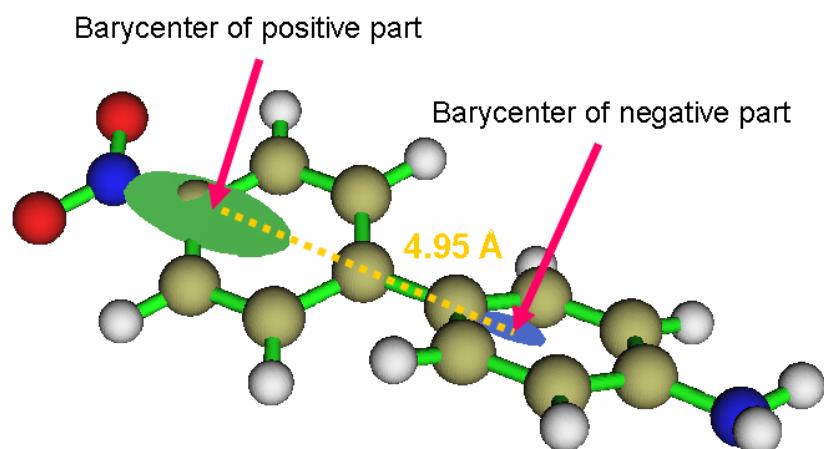
```

These informations are self-explained, if you are confused, recall Section 3.100.10.

By selecting option 1, C₊ (green) and C₋ (blue) functions can be shown up. The isovalue of the graph shown below is 0.0015.



If the isovalue is increased to 0.0024, the positions of barycenters can be approximately located.



From the graph it is evident that the direction of electron-transfer is from amino group side (electron donor) to nitrophenyl group side (electron acceptor). However the barycenters are not located on the two substituents exactly, and the displacement at amino group side are unneglectable, it suggests that electron depletion zone quite delocalized to nearby phenyl, the actual electron acceptor is not amino group but phenyl, this finding parallels that phenyl is a weak electron donor.

5 Skills

5.1 Make Multiwfn support more quantum chemistry programs

Some quantum chemistry programs don't support exporting .wfn file, however they support exporting Molden input file, such as Molpro, Molcas, ORCA and Dalton. Molden input file contains abundant wavefunction informations as .wfn file, you can use Molden2aim program written by W. Zou (<http://people.smu.edu/wzou/program/index.html>) to convert Molden input file to .wfn file. For example, in Molpro, add *put,molden,ltwd.mold* in tail of input file, the Molden input file *ltwd.mold* will be generated after running the task, then move *ltwd.mold* into the directory where *molden2aim.exe* is posited in, then boot up Molden2aim, press ENTER and input *ltwd.mold*, you will get *ltwd.mold.wfn*, this file can be directly used as Multiwfn input.

Please don't modify the title line of the generated wfn file.

Since Molden2aim v2.0.5, orbital spin-types are explicitly written at the end of the converted wfn file (\$MOSPIN \$END field), this information is automatically loaded by Multiwfn.

Molden2aim can output GTFs with angular moment up to *g* (although *g* GTFs is not formally defined in wfn format), these *g* GTFs can also be correctly identified and loaded into Multiwfn.

The "charge" field of the wfn file outputted by Molden2aim is the serial number of the element in periodic table, rather than the effective nuclear charges, even if effective core potential (ECP) is used. This treatment is inconsistent with the wfn file outpttuted by Gaussian, in which "charge" is effective core charges when ECP is used (e.g. the "charge" of Au under Lanl2DZ is 19.0). So, if ECP is used and you want to calculate electrostatic potential, don't forget to modify "charge" field in the wfn file outputted by Molden2aim to effective nuclear charges.

5.2 Running Multiwfn silently

Multiwfn aims for ease of use, so it is designed as an interactive program. However Multiwfn can also run in silent mode, in which you don't need to press any keys during Multiwfn running. For example, you want to obtain the graph in example 4.4.1 silently, you need to compile an inputstream file first, the content is (red texts are comments):

```
-1
4
4
1
1
200, 200
2
```

- 0
- 0 ← Option 0: save graph to current directory

I suppose that the inputstream file is named “4.4.1.txt”, I have already provided this file in “examples” directory. Now change “isilent” parameter in settings.ini from 0 to 1, this step tells Multiwfn to forbid any graph or GUI popping up automatically during running (otherwise you have to close the window by clicking mouse button). Then run this command in MS-DOS (to enter MS-DOS from Windows, click “Start”-“run” and type “cmd”): *Multiwfn hcn.wfn < 4.4.1.txt > medinfo.txt*, here I assume Multiwfn.exe, 4.4.1.txt and hcn.wfn are in current directory. After a few seconds, you will find the expected graphic file appears in current directory, from medinfo.txt you can also find all intermediate informations outputted by Multiwfn (the same as in interactive mode).

What does the content in inputstream file mean? The answer is easy: the texts in each line of inputstream file are just what you need to input in interactive mode, you can redo example 4.4.1 to ensure it is true. It is easy to write a new inputstream file by following the prompts in interactive mode. The symbol “<” and “>” are called redirection operator, they respectively tell Multiwfn the content in 4.4.1.txt is inputstream, meanwhile outputstream should be stored in medinfo.txt. This redirection mechanism is not provided by Multiwfn but provided by operation system. Notice that the name of input file is missing in inputstream file, because it appears as an argument.

You may have noted that when the task is finished, some errors like below appear, you can safely ignore them:

```
forrtl: severe (24): end-of-file during read, unit -4, file CONIN$  

Image          PC       Routine      Line      Source  

Multiwfn.exe   00588F1A Unknown      Unknown    Unknown  

Multiwfn.exe   00586438 Unknown      Unknown    Unknown  

Multiwfn.exe   00530B3A Unknown      Unknown    Unknown  

.....
```

Another example, assume that you want to save detail compositions of orbital 1 to 30 of COCl2.fch, just create an inputstream file dubbed “orbana_1_30.in” with below content:

```
8 ← Orbital composition analysis  

1 ← Mulliken method  

1 ← Orbital 1  

2 ← Orbital 2  

3 ← Orbital 3  

... ← Fill 4, 5, 6... by yourself, you can use some tricks to generate arithmetic sequence to  

avoid input them manually  

30 ← Orbital 30
```

Then run the command: *Multiwfn COCl2.fch < orbana_1_30.in > orbana_1_30.txt*.

5.3 Running Multiwfn in batch mode (for Windows)

If you are familiar with writing scripts and you have read last section, you must have known how to use Multiwfn to process a batch of files, it does very easy. Assume that you want to

generate cube files of ELF for these files: ultravox.wfn, chinaski.fch, strawberry_egg.wfn, compile a plain text file named “batchrun.bat” with below content:

```
Mul ti wfn ul travox.wfn < genELFcub.txt > nul |
move ELF.cub ul travox.cub
Mul ti wfn chinaski.fch < genELFcub.txt > nul |
move ELF.cub chinaski.cub
Mul ti wfn strawberry_egg.wfn < genELFcub.txt > nul |
move ELF.cub strawberry_egg.cub
del nul |
```

Where genELFcub.txt is inputstream file for generating ELF cube file, the content is

5
9
2

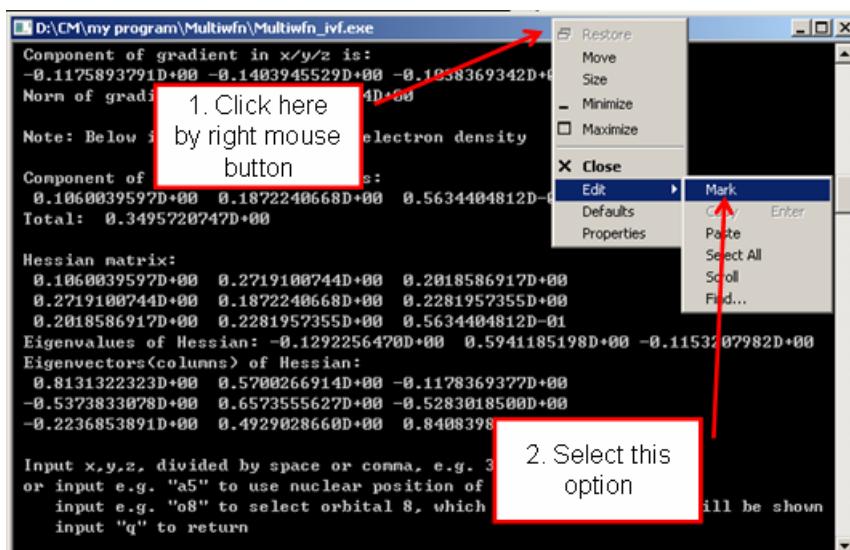
2 ← Option 2: Exporting cube file in current directory

If you don't understand the meaning, read last section and do example 4.5.1.

Now, double click the icon of batchrun.bat or input *batchrun* in MS-DOS, the work will start and then the three ELF cube files appear in turn.

5.4 Copy outputs from command-line window to clipboard (for Windows)

Sometimes the outputs of Multiwfn on command-line window are needed to be stored permanently or transported to third-part softwares via plain text file. Here I describe how to copy these outputs to Windows clipboard, assuming that you want to copy Hessian matrix of electron density.



After you select "Mark", drag left mouse button from point A to point B

```

Select D:\CM\my program\Multiwfn\Multiwfn_ivf.exe
Component of gradient in x/y/z is:
-0.1175893791D+00 -0.1403945529D+00 -0.1038369342D+00
Norm of gradient is: 0.2105231614D+00

Note: Below informations are for electron density

Component of Laplacian in x/y/z is:
0.1060039597D+00 0.1872240668D+00 0.5634404812D-01
Total: 0.3495720747D+00

Hessian matrix:
0.1060039597D+00 0.2719100744D+00 0.2018586917D+00
0.2719100744D+00 0.1872240668D+00 0.2281957355D+00
0.2018586917D+00 0.2281957355D+00 0.5634404812D-01
Eigenvalues of Hessian: -0.1292256470D+00 0.5941185178D+00 -0.1153207982D+00
Eigenvectors (columns) of Hessian:
0.8131322323D+00 0.5700266914D+00 -0.1178369377D+00
-0.5373833078D+00 0.6573555627D+00 -0.5283018500D+00
-0.2236853891D+00 0.4929028660D+00 0.8408398251D+00

Input x,y,z, divided by space or comma, e.g. 3.3,2.0,-0.3
or input e.g. "a5" to use nuclear position of atom 5
    input e.g. "o8" to select orbital 8, which wavefunction value will be shown
    input "q" to return

```

Then press ENTER button, the information highlighted by white rectangle will be stored to clipboard, you can paste them anywhere.

5.5 Make command-line window capable to record more outputs (for Windows)

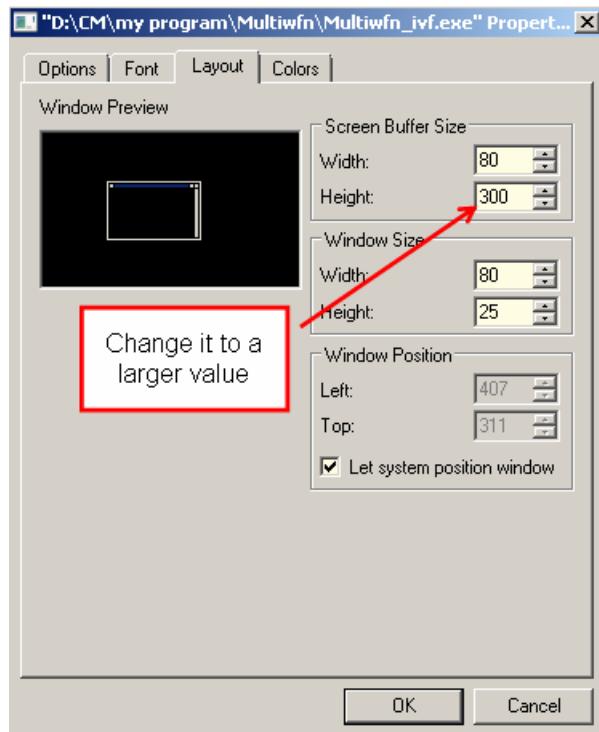
Occasionally you may find command-line window cannot record entire outputs of Multiwfn. For example, you select option 6 in wavefunction modification module to get density matrix for a relative big wavefunction, however only the last part of the matrix can be found in the command-line window. The solution of the problem is to enlarge buffer size of the window, please follow below steps.

Boot up Multiwfn, click title of the window by right mouse button, click "Properties", select "Layout" page, you will find the default buffer size of the window is 300 (see the screenshot below), that means only up to 300 lines can be recorded in the window, which is obviously too small. Change the value to a larger value, for example 10000, and then click OK button. After that you will find the window capable to record much more outputs (If the complete output still cannot be recorded, enlarge buffer size again).

The buffer size setting is saved permanently in system, you needn't to set this value again at next time you boot up Multiwfn.

For Linux, you can find similar option used to set buffer size of terminal too.

5 Skills



Appendix

1 Setting up environment variable for Gaussian in Windows

If the error “No executable for file 11.exe” occurs when Multiwfn is invoking Gaussian, that means the environment variable “GAUSS_EXEDIR“ has not been set correctly. To set it in Windows XP system, enter “control panel”-“System properties”-“Advanced”, click “Environment variables” button, then click “New” button (in “User variables” frame), input *GAUSS_EXEDIR* in variable name, input the install directory of Gaussian in variable value (e.g. *d:\study\g09w*, assuming g09.exe is in this folder). Afterwards when Gaussian is invoking, Gaussian will search executable files of each link (such as 11.exe) in this folder.

2 Routine names of real space functions

Below are routine names of all real space functions supported by Multiwfn, corresponding codes can be found in function.f90. You can make use of them to construct user defined function.

userfunc(x,y,z) : User defined function
 fmo(x,y,z,i) : Wavefunction value of orbital *i*
 fdens(x,y,z) : Electron density
 avglocion(x,y,z) : Average local ionization energy
 fspindens(x,y,z) : Electron spin density
 ffermihole(x,y,z,intype) : Fermi hole / correlation factor
 fgrad(x,y,z,'x') : X component of electron density gradient. 'y', 'z' for Y, Z component
 fgrad(x,y,z,'t') : Norm of electron density gradient
 fgrad(x,y,z,'r') : Reduced density gradient
 Lagkin(x,y,z,idir) : Lagrangian kinetic G(r). idir=0/1/2/3 means total/X/Y/Z kinetic energy
 Hamkin(x,y,z,idir) : Hamiltonian kinetic K(r). idir=0/1/2/3 means total/X/Y/Z kinetic energy
 flapl(x,y,z,"x") : 2-order derivative of electron density respect to X. 'y','z' for Y, Z counterpart
 flapl(x,y,z,"t") : Laplacian of electron density
 calchessmat_dens(x,y,z,elehess,elerho) : Return Hessian matrix of electron density (elehess)
 along with electron density (elerho)
 gencalehessmat(ifunc,x,y,z,hess,value,grad,intype) : A general routine returns Hessian matrix,
 gradient vector and value of a real space function at x,y,z. ifunc decides real space function
 ELF_LOL(x,y,z,"ELF") : Electron localization function
 ELF_LOL(x,y,z,"LOL") : Localized orbital locator
 RDGprodens(x,y,z) : Reduced density gradient with promolecular approximation
 signlambda2rho(x,y,z) : Sign(λ_2)* ρ

Appendix

signlambda2rho_prodens(x,y,z) : Sign(λ_2)* ρ with promolecular approximation
calchessmat_prodens(x,y,z,elehess,elerho) : "calchessmat" routine with promolecular approximation
infoentro(x,y,z) : Local information entropy
totesp(x,y,z) : Total electrostatic potential
nucesp(x,y,z) : Electrostatic potential from nuclear / atomic charges
eleesp(x,y,z) : Electrostatic potential from electrons