

MULTIWFN

- A Multifunctional Wavefunction Analyzer -

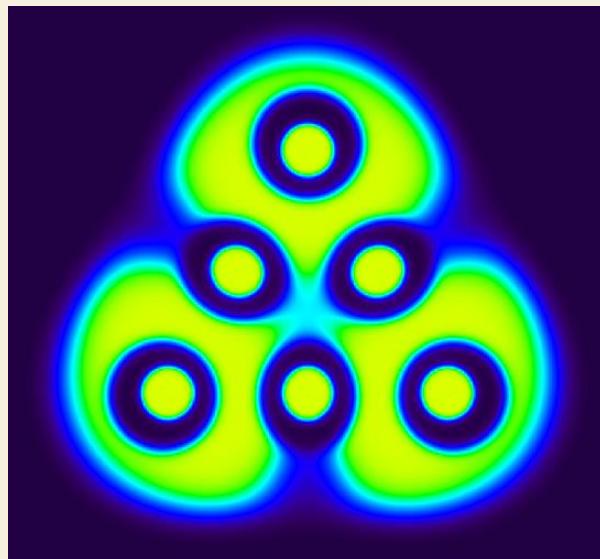
Software Manual

with tutorials and abundant examples in Chapter 4

Version 3.6(dev)

2019-Mar-30

Official site: <http://sobereva.com/multiwfn>



Tian Lu

sobereva@sina.com

Beijing Kein Research Center for Natural Sciences (www.keinsci.com)

!!!!!! ALL USERS MUST READ !!!!!!!

I know, most people, including myself, are unwilling to read lengthy manual. Since Multiwfn is a heuristic and very user-friendly program, it is absolutely unnecessary to read through the whole manual before using it. However, you should never skip reading following content!

1. Please feel free to ask questions about using Multiwfn by posting topics on Multiwfn English forum (<http://sobereva.com/wfnbbs>) or Multiwfn Chinese forum (<http://bbs.keinsci.com/wfn>). I always would like to provide as much help as I can for any Multiwfn user. \(^{o}\)^(^o)/
2. The BEST way to get started quickly is reading Chapter 1, Sections 2.1~2.5, and then follow the tutorials in Chapter 4. After that if you want to learn more about theoretical backgrounds and details of Multiwfn, then read Section 2.6~2.8 of Chapter 2 and Chapter 3. Note that the tutorials and examples given in Chapter 4 only cover most important and frequently used functions rather than all functions of Multiwfn.
3. Different functions of Multiwfn require different type of input file, please carefully read Section 2.5 for explanation.
4. Usually Multiwfn runs in interactive mode, however Multiwfn can also be run in command-line mode, and it is very convenient to write shell script to make Multiwfn deal with a large number of input files. See Sections 5.2 and 5.3 for detail.
5. If you do not know how to copy the output of Multiwfn from command-line window to a plain text file, consult Section 5.4. If you do not know how to enlarge screen buffer size of command-line window of Windows system, consult Section 5.5.
6. If 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. (Note: Most functions in Multiwfn DO NOT require Gaussian installed on your local machine)
7. The so-called ”current directory” in this manual is the path where you are invoking Multiwfn. If you boot up Multiwfn by clicking the icon of executable file in Windows, the ”current directory” is the position of the executable file. For the case of command-line mode, if you are in “D:\study\” directory when invoking Multiwfn, then “D:\study\” is ”current directory”.
8. Please often check Multiwfn website and update the program to the latest version. Multiwfn is always in active development, useful functions are continuously added, bugs are continuously fixed and efficiency is continuously improved.

Linux and Mac OS USERS MUST READ

1. If you are new to Multiwfn, using Windows version is recommended, since this version of Multiwfn is more stable than others, and you do not need to set up your system.
2. See Section 2.1.2 and 2.1.3 on how to install Linux and Mac OS versions of Multiwfn, respectively.
3. When graphical user interface (GUI) appears, the graph cannot be shown automatically, you have to do something to active the graph, e.g. clicking "up" button or dragging a scale bar in the GUI.
4. Transparent style does not work when showing isosurfaces in GUI window. But if you choose to save the picture, the transparent isosurfaces can be rendered normally in the resulting graphical file.
5. Using Mac OS version is deprecated, some functions may do not work normally.

About the manual

This manual is organized as following sequence:

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

Chapter 2 General information: Introduces all aspects beyond specific functions of Multiwfn, such as how to install, how to use, supported file types, etc.

Chapter 3 Functions: Describes all functions and related theories of Multiwfn in detail.

Chapter 4 Tutorials and examples: Plenty of practical examples are provided for helping users to use Multiwfn.

Chapter 5 Skills: Some useful skills in using Multiwfn.

Appendix

In general, italics font is used for input commands, journal name and filename throughout this manual. Very important contents are highlighted by red color.

You can directly jump to specific section by choosing corresponding entry in bookmark window of your pdf reader.

The purpose of this manual is not only helping users to use Multiwfn, but also introducing related theories. I hope any quantum chemist can benefit from this manual.

Albeit I have tried to write this manual as readable as possible for beginners, some topics request the readers have basic knowledges of quantum chemistry. If the readers have carefully read the book "Quantum Chemistry 7ed" written by Ira. Levine, then they will never encounter any difficulty during reading through the whole manual.

Contents

!!!!!! ALL USERS MUST READ !!!!!!!	i
Linux and Mac OS USERS MUST READ	ii
About the manual.....	iii
Contents	iv
1 Overview.....	1
2 General information.....	8
2.1 Install.....	8
2.1.1 Windows version	8
2.1.2 Linux version	8
2.1.3 Mac OS version	9
2.2 Using Multiwfn	9
2.3 Files of Multiwfn	10
2.4 Parallel implementation	10
2.5 Input files and wavefunction types.....	11
2.6 Real space functions.....	15
2.7 User defined real space function	28
2.8 Graphic formats and image size	40
3 Functions	41
3.2 Showing molecular structure and viewing orbitals / isosurfaces (0).....	41
3.3 Outputting all properties at a point (1)	43
3.4 Outputting and plotting specific property in a line (3).....	44
3.5 Outputting and plotting specific property in a plane (4)	45
3.5.1 Graph types	45
3.5.2 Setting up grid, plane and plotting region	50
3.5.3 Options in post-process interface	52
3.5.4 Setting up contour lines.....	53
3.5.5 Plot critical points, paths and interbasin paths on plane graph.....	54
3.6 Outputting and plotting specific property within a spatial region (5)	55
3.7 Custom operation, promolecular and deformation properties (subfunction 0, -1, -2 in main function 3, 4, 5).....	59
3.7.1 Custom operation for multiple wavefunctions (0)	59
3.7.2 Promolecular and deformation properties (-1, -2).....	59
3.7.3 Generation of atomic wavefunctions.....	60

3.7.4 Sphericalization of atom wavefunction	61
3.8 Checking & Modifying wavefunction (6).....	61
3.9 Population analysis and atomic charges (7)	64
3.9.1 Hirshfeld atom population (1).....	64
3.9.2 Voronoi deformation density (VDD) atom population (2).....	65
3.9.3 Mulliken atom & basis function population analysis (5)	66
3.9.4 Löwdin atom & basis function population (6)	67
3.9.5 Modified Mulliken atom population defined by Ros & Schuit (SCPA) (7)	68
3.9.6 Modified Mulliken atom population defined by Stout & Politzer (8).....	69
3.9.7 Modified Mulliken atom population defined by Bickelhaupt (9).....	69
3.9.8 Becke atomic charge with atomic dipole moment correction (10).....	69
3.9.9 Atomic dipole moment corrected Hirshfeld atomic charges (ADCH, 11)	70
3.9.10 CHELPG (Charges from electrostatic potentials using a grid based method) ESP fitting atomic charge (12).....	72
3.9.11 Merz-Kollmann (MK) ESP fitting atomic charge (13).....	74
3.9.12 AIM atomic charge (14)	75
3.9.13 Hirshfeld-I atom population (15)	75
3.9.14 CM5 atomic charge (16)	78
3.9.15 Electronegativity Equalization Method (EEM) atomic charge (17).....	79
3.9.16 Restrained ElectroStatic Potential (RESP) atomic charge (18).....	81
3.9.16.1 Theory	81
3.9.16.2 Usage.....	86
3.10 Orbital composition analysis (8)	88
3.10.1 Output basis function, shell and atom composition in a specific orbital by Mulliken, Stout-Politzer and SCPA approaches (1, 2, 3).....	89
3.10.2 Define fragment 1 and 2 (-1, -2)	90
3.10.3 Output composition of fragment 1 and inter-fragment composition by Mulliken, Stout-Politzer and SCPA approaches (4, 5, 6).....	90
3.10.4 Orbital composition analysis by natural atomic orbital approach (7)	91
3.10.5 Calculate atom and fragment contributions by Hirshfeld or Hirshfeld-I method (8,10)	93
3.10.6 Calculate atom and fragment contributions by Becke method (9)	93
3.10.7 Evaluate oxidation state by LOBA method.....	94
3.11 Bond order analysis (9)	94
3.11.1 Mayer bond order analysis (1).....	94
3.11.2 Multi-center bond order analysis (2, -2, -3)	96
3.11.3 Wiberg bond order analysis in Löwdin orthogonalized basis (3)	98
3.11.4 Mulliken bond order analysis (4) and decomposition (5).....	99
3.11.5 Orbital occupancy-perturbed Mayer bond order (6).....	99
3.11.6 Fuzzy bond order (7)	100
3.11.7 Laplacian bond order (8)	101
3.11.8 Decompose Wiberg bond order in NAO basis as atomic orbital pair contributions (9)	101
3.12 Plotting total, partial, and overlap population density-of-states (10)	102

3.12.1 Theory	102
3.12.2 Input file.....	104
3.12.3 Options for plotting DOS.....	105
3.12.4 Local DOS.....	106
3.13 Plotting IR, Raman, UV-Vis, ECD, VCD and ROA spectra (11).....	107
3.13.1 Theory	107
3.13.2 Input file.....	109
3.13.3 Usage and options	112
3.13.4 Plotting multiple files and weighted spectrum.....	114
3.14 Topology analysis (2)	115
3.14.1 Theory	115
3.14.2 Search critical points	118
3.14.3 Generate topology paths.....	120
3.14.4 Generate interbasin surfaces.....	120
3.14.5 Visualize, analyze, modify and export results.....	121
3.14.6 Calculate the aromaticity indices based on topology properties of electron density..	122
3.15 Quantitative analysis of molecular surface (12).....	123
3.15.1 Theory	123
3.15.2 Numerical algorithm	127
3.15.3 Parameters and options	129
3.15.4 Options at post-process stage	131
3.15.5 Special topic: Hirshfeld and Becke surface analyses	134
3.16 Process grid data (13).....	137
3.16.0 Visualize isosurface of present grid data (-2).....	138
3.16.1 Output present grid data to Gaussian cube file (0).....	138
3.16.2 Output all data points with value and coordinate (1)	138
3.16.3 Output data points in a XY/YZ/XZ plane (2, 3, 4).....	138
3.16.4 Output average data of XY/YZ/XZ planes in a range of Z/X/Y (5,6,7).....	138
3.16.5 Output data points in a plane defined three atom indices or three points (8,9).....	138
3.16.6 Output data points in specified value range (10).....	138
3.16.7 Grid data calculation (11).....	139
3.16.8 Map values of a cube file to specified isosurface of present grid data (12)	139
3.16.9 Set value of the grid points that far away from / close to some atoms (13)	140
3.16.10 Set value of the grid points outside overlap region of two fragments (14)	140
3.16.11 If data value is within certain range, set it to a specified value (15)	141
3.16.12 Scale data range (16).....	141
3.16.13 Show statistic data of the points in specific spatial and value range (17)	141
3.16.14 Calculate and plot integral curve in X/Y/Z direction (18)	141
3.17 Adaptive natural density partitioning (AdNDP) analysis (14)	142
3.17.1 Theory	142
3.17.2 Input file.....	145
3.17.3 Options	145
3.18 Fuzzy atomic space analysis (15).....	148
3.18.0 Basic concepts.....	148

3.18.1 Integration of a real space function in fuzzy atomic spaces (1)	151
3.18.2 Integration of a real space function in overlap spaces (8).....	152
3.18.3 Calculate atomic multipole moments (2)	152
3.18.4 Calculate atomic overlap matrix (3).....	154
3.18.5 Calculate localization and delocalization index (4).....	154
3.18.6 Calculate PDI (5).....	157
3.18.7 Calculate FLU and FLU- π (6,7).....	158
3.18.8 Calculate condensed linear response kernel (9)	159
3.18.9 Calculate para linear response index (10)	160
3.18.10 Calculate multi-center delocalization index (11).....	160
3.18.11 Calculate information-theoretic aromaticity index (12)	160
3.19 Charge decomposition analysis and plotting orbital interaction diagram (16).....	161
3.19.1 Theory	161
3.19.2 Input file	165
3.19.3 Usage.....	166
3.20 Basin analysis (17).....	168
3.20.1 Theory	168
3.20.2 Numerical aspects	169
3.20.3 Usage.....	172
3.21 Electron excitation analysis (18).....	177
Basic information about electron excitation analysis module	177
3.21.0 Check, modify and export configuration coefficients of an excitation (-1)	180
3.21.1 Analyze and visualize hole-electron distribution, transition density, and transition electric/magnetic dipole moment density (1)	181
3.21.1.1 Theory	181
3.21.1.2 Usage and Functions	189
3.21.2 Plot atom/fragment transition matrix of various kinds as heat map (2)	191
3.21.3 Analyze charge-transfer based on density difference grid data (3)	195
3.21.4 Calculate Δr index to measure charge-transfer length (4)	198
3.21.5 Calculate transition electric dipole moments between all states and electric dipole moment of each state (5).....	199
3.21.6 Generate natural transition orbitals (NTOs) (6)	200
3.21.7 Calculate ghost-hunter index (7)	202
3.21.8 Calculate interfragment charge transfer in electron excitation via IFCT method (8).203	203
3.21.9 Generate and export transition density matrix (9).....	206
3.21.10 Decompose transition dipole moment as molecular orbital pair contributions (10)	207
3.21.11 Decompose transition dipole moment as basis function and atom contributions (11)	208
3.21.12 Calculate Mulliken atomic transition charges (12)	209
3.21.13 Generate natural orbitals of specific excited states (13).....	210
3.21.14 Calculate Λ index to characterize electron excitation (14).....	211
3.22 Orbital localization analysis (19)	212
3.23 Visual study of weak interaction (20).....	217
3.23.1 Noncovalent interaction (NCI) anaylsis (1)	217

3.23.2 NCI analysis based on promolecular density (2).....	225
3.23.3 Averaged NCI analysis (NCI analysis for multiple frames. 3)	226
3.23.4 Density Overlap Regions Indicator (DORI) analysis (5)	227
3.23.5 Independent Gradient Model (IGM) analysis based on promolecular density (10) ...	228
3.24 Energy decomposition analysis (21)	232
3.24.1 Energy decomposition analysis based on molecular forcefield (EDA-FF).....	232
3.100 Other functions, part 1 (100).....	236
3.100.1 Draw scatter graph between two functions and generate their cube files	236
3.100.2 Export .pdb, .pqr, .xyz, .wfn, .wfx, .molden, .fch, NBO .47 or input file of mainstream quantum chemistry codes.....	237
3.100.3 Calculate molecular van der Waals volume	238
3.100.4 Integrate a function in whole space.....	239
3.100.5 Show overlap integral between alpha and beta orbitals	240
3.100.6 Monitor SCF convergence process of Gaussian.....	241
3.100.7 Generate Gaussian input file with initial guess from converged wavefunction	242
3.100.8 Generate Gaussian input file with initial guess from fragment wavefunctions.....	242
3.100.9 Evaluate interatomic connectivity and atomic coordination number	244
3.100.11 Calculate overlap and centroid distance between two orbitals	246
3.100.12 Perform biorthogonalization between alpha and beta orbitals	246
3.100.13 Calculate HOMA and Bird aromaticity index.....	248
3.100.14 Calculate LOLIPOP (LOL Integrated Pi Over Plane).....	250
3.100.15 Calculate intermolecular orbital overlap	250
3.100.18 Yoshizawa's electron transport route analysis	251
3.100.19 Generate promolecular .wfn file from fragment wavefunctions	253
3.100.20 Calculate Hellmann-Feynman forces	253
3.100.21 Calculate properties based on geometry information for specific atoms	254
3.100.22 Detect π orbitals and set occupation numbers	255
3.100.23 Fit function distribution to atomic value	256
3.100.24 Obtain NICS_ZZ value for non-planar or tilted system.....	257
3.200 Other functions, part 2 (200).....	258
3.200.1 Calculate core-valence bifurcation (CVB) index and related quantities	258
3.200.2 Calculate atomic and bond dipole moments in Hilbert space	264
3.200.3 Generate cube file for multiple orbital wavefunctions.....	266
3.200.4 Generate iso-chemical shielding surfaces (ICSS) and related quantities	266
3.200.5 Plot radial distribution function for a real space function	268
3.200.6 Analyze correspondence between orbitals in two wavefunctions	268
3.200.7 Parse output of (hyper)polarizability task of Gaussian	270
3.200.8 Calculate (hyper)polarizability by sum-over-states (SOS) method.....	273
3.200.9 Calculate average bond length and average coordinate number	277
3.200.10 Output various kinds of integral between orbitals.....	278
3.200.11 Calculate center, the first and second moments of a real space function	279
3.200.12 Calculate energy index (EI) or bond polarity index (BPI)	280
3.200.14 Domain analysis (Obtaining properties within isosurfaces of a function)	281
3.200.15 Calculate electron correlation index.....	282

3.200.16 Generate natural orbitals, natural spin orbitals and spin natural orbitals based on the density matrix in .fch/.fchk file	283
4 Tutorials and Examples	285
Prologue and generation of input files	285
4.0 View orbitals and structure.....	288
4.0.1 Viewing molecular orbitals of cycloheptatriene.....	288
4.0.2 Viewing natural bond orbitals (NBO) of ethanol.....	290
4.0.3 Using Multiwfn+VMD to rapidly plot high-quality orbital isosurface map	292
4.1 Calculate properties at a point.....	294
4.1.1 Show all properties of triplet water at a given point	294
4.1.2 Calculate ESP at nuclear positions to evaluate interaction strength of H ₂ O···HF	296
4.2 Topology analysis.....	298
4.2.1 Atoms in molecules (AIM) and aromaticity analysis for 2-pyridoxine 2-aminopyridine	298
4.2.2 LOL topology analysis for acetic acid.....	303
4.2.3 Plot real space function along bond path	305
4.2.4 Decompose properties at a critical point as orbital contributions	306
4.2.5 Easily plot high quality AIM topology map in VMD visualization program based on Multiwfn outputs.....	309
4.2.6 Topology analysis in special regions: G-C...G-C base pair as an example	312
4.3 Output and plot various properties in a line	314
4.3.1 Plot the spin density curve of triplet formamide along carbon and oxygen atoms.....	314
4.3.2 Study Fermi hole and Coulomb hole of H ₂	315
4.3.3 Study interatomic interaction via PAEM-MO method	318
4.4 Output and plot various properties in a plane	321
4.4.1 Color-filled map and contour line map of electron density of hydrogen cyanide	321
4.4.2 Shaded surface map with projection effect of electron localization function (ELF) of monofluoroethane	324
4.4.3 Plotting plane map without contributions from some atoms.....	325
4.4.4 Contour map of electrostatic potential of chlorine trifluoride.....	327
4.4.5 Contour map of two orbital wavefunctions	329
4.4.6 Gradient+contour map with topology paths of electron density of hydrogen peroxide	331
4.4.7 Deformation map of electron density of acetyl chloride	334
4.4.8 Draw difference map of electron density and ELF for water tetramer with respect to its constituent monomers	336
4.4.9 Draw LOL- π map for porphyrin to reveal favorable electron delocalization path.....	338
4.5 Generate grid data and view isosurface map.....	340
4.5.1 Electron localization function of chlorine trifluoride.....	340
4.5.2 Laplacian of electron density of 1,3-butadiene	341
4.5.3 Calculate ELF- α and ELF- π to study aromaticity of benzene.....	342
4.5.4 Use Fukui function and dual descriptor to study favorable site for electrophilic attack of phenol.....	346

4.5.5 Plot difference map of electron density to study electron transfer of imidazole coordinated magnesium porphyrin.....	350
4.5.6 Study electron delocalization range function EDR($r;d$) of anionic water dimer	353
4.5.7 Study orbital overlap distance function D(r) of thioformic acid	354
4.6 Modify and check wavefunction.....	355
4.6.1 Delete certain Gaussian functions	355
4.6.2 Remove contributions from certain orbitals to real space functions	356
4.6.3 Translate and duplicate graphene primitive cell wavefunction to periodic system.....	357
4.7 Population analysis and atomic charge calculation.....	359
4.7.0 Mulliken population analysis on triplet ethanol	359
4.7.1 Calculate Hirshfeld and CHELPG atomic charges as well as fragment charge for chlorine trifluoride	361
4.7.2 Calculate and compare ADCH atomic charges with Hirshfeld atomic charges for acetamide	362
4.7.3 Calculate condensed Fukui function and condensed dual descriptor	363
4.7.4 Illustration of computing Hirshfeld-I atomic charges	365
4.7.5 Calculating EEM atomic charges for ethanol-water cluster.....	367
4.7.6 Determining correspondence between basis functions and atomic orbitals via population analysis.....	368
4.7.7 Illustration of deriving RESP charges and normal ESP fitting charges with extra constraints	370
Example 1: Deriving RESP charges for dopamine in ethanol environment.....	370
Example 2: Taking multiple conformations into account during RESP charge calculation of dopamine	373
Example 3: Imposing equivalence constraint in ESP fitting of Dimethyl phosphate.....	375
Example 4: Evaluation of atomic charges of aspartic acid residue with equivalence and charge constraints.....	376
Example 5: Using two times of one-stage fitting to equivalently realize standard RESP two-stage fitting	378
4.7.8 Examine electrostatic potential reproducibility of atomic charges.....	380
4.8 Molecular orbital composition analysis	382
4.8.1 Analyze acetamide by Mulliken method	383
4.8.2 Analyze water by natural atomic orbital method.....	386
4.8.3 Analyze acetamide by Hirshfeld and Becke method	388
4.8.4 Calculate oxidation state by LOBA method.....	390
4.9 Bond order analysis.....	391
4.9.1 Mayer bond order and fuzzy bond order analysis on acetamide	391
4.9.2 Three-center bond order analysis on Li ₆ cluster	394
4.9.3 Calculate Laplacian bond order (LBO).....	395
4.9.4 Decomposition analysis of Wiberg bond order in NAO basis for formaldehyde	396
4.9.5 Study orbital contributions to Mulliken bond order for C-C bond of CH ₃ CONH ₂	398
4.10 Plot density of states (DOS).....	399
4.10.1 Plot total, partial and overlap DOS for N-phenylpyrrole	399
4.10.2 Plot local DOS for 1,3-butadiene	405

4.11 Plot various kinds of spectra	407
4.11.1 Plot infrared (IR) spectrum for NH_3BF_3	407
4.11.2 Plot UV-Vis spectrum and contributions from individual transitions for acetic acid .	408
4.11.3 Plot electronic circular dichroism (ECD) spectrum for asparagine.....	410
4.11.4 Plot conformational weighted UV-Vis and ECD spectra for plumericin.....	411
4.11.5 Plot Raman spectrum for 2-methyloxirane based on Raman intensity.....	414
4.11.6 Simultaneously plot multiple systems.....	415
4.11.7 Plot Raman optical activity (ROA) spectrum for chiral molecule S-methyloxirane..	417
4.12 Quantitative analysis of molecular surface	419
4.12.1 Electrostatic potential analysis on phenol molecular surface.....	419
4.12.2 Average local ionization energy analysis on phenol molecular surface	424
4.12.3 Local molecular surface analysis for acrolein.....	426
4.12.4 Fukui function distribution on local molecular surface of phenol.....	428
4.12.5 Becke surface analysis on guanine-cytosine base pair	429
4.12.6 Hirshfeld surface analysis and fingerprint plot analysis on urea crystal	431
4.12.7 Predict density of molecular crystal of FOX-7	436
4.12.8 Quantitative analysis of orbital overlap distance function $D(r)$ on thioformic acid molecular surface	437
4.13 Process grid data	438
4.13.1 Extract data points in a plane	438
4.13.2 Perform mathematical operation on grid data	439
4.13.3 Scaling numerical range of grid data.....	440
4.13.4 Screen isosurfaces in local regions.....	440
4.13.4.1 Screen isosurfaces inside or outside a region.....	440
4.13.4.2 Screen isosurfaces outside overlap region of two fragments	441
4.13.5 Acquire barycenter of a molecular orbital.....	443
4.13.6 Plot charge displacement curve	444
4.13.7 Evaluation of electron density overlap	447
4.14 Adaptive natural density partitioning (AdNDP) analysis.....	449
4.14.1 Analyze Li_5^+ cluster.....	449
4.14.2 Analyze B_{11} cluster	450
4.14.3 Analyze phenanthrene	454
4.15 Fuzzy atomic space analysis	456
4.15.1 Study delocalization index of benzene.....	456
4.15.2 Study aromaticity of phenanthrene by PDI, FLU, FLU- π and PLR	458
4.16 Charge decomposition analysis and plotting orbital interaction diagram	459
4.16.1 Closed-shell interaction case: COBH_3	459
4.16.2 Open-shell interaction case: CH_3NH_2	463
4.16.3 More than two fragments case: $\text{Pt}(\text{NH}_3)_2\text{Cl}_2$	465
4.17 Basin analysis.....	466
4.17.1 AIM basin analysis for HCN and Li_6	467
4.17.2 ELF basin analysis for acetylene.....	473
4.17.3 Basin analysis of electrostatic potential for H_2O	476
4.17.4 Basin analysis of electron density difference for H_2O	480

4.17.5 Study source function in AIM basins	482
4.17.6 Local region basin analysis for polyyne.....	484
4.17.7 Evaluate atomic contribution to population of ELF basins	485
4.18 Electron excitation analysis.....	486
4.18.1 Using hole-electron analysis to fully characterize electron excitations	487
4.18.1.1 Example 1: NH ₂ -biphenyl-NO ₂	487
4.18.1.2 Example 2: Ru(bpy) ₃ ²⁺ cation in water	497
4.18.2 Illustration of transition density and transition dipole moment density analysis	500
4.18.2.1 Analyzing transition density in real space for N-phenylpyrrole.....	500
4.18.2.2 Analyzing transition density in terms of transition density matrix (TDM) plot ..	503
4.18.2.3 Generate grid data of transition density between two excited states.....	503
4.18.3 Analyze charge-transfer during electron excitation based on electron density difference	504
4.18.4 Calculate Δ _r and Δ indices to characterize various electron excitations for N-phenylpyrrole	507
4.18.6 Generate and analyze natural transition orbitals (NTOs) for uracil	509
4.18.8 Using IFCT method to study interfragment charge transfer during electron excitation for 4-nitroaniline.....	512
4.18.10 Gain molecular orbital pair contributions to transition dipole moment	515
4.18.11 Plot transition dipole moment vector contributed by molecular fragments as arrows	517
4.18.13 Study electronic structure of a single excited state and difference between two excited states.....	519
4.19 Orbital localization analysis.....	522
4.19.1 Localizing molecular orbital of 1,3-butadiene by Pipek-Mezey method	522
4.19.2 Analyze variation of localized molecular orbitals for SN ₂ reaction	525
4.19.3 Characterize Re-Re bond of [Re ₂ Cl ₈] ²⁻ anion.....	526
4.19.4 Study bond dipole moment based on two-center LMOs for CH ₃ NH ₂	528
4.20 Visual study of weak interactions.....	529
4.20.1 Studying weak interaction in 2-pyridoxine 2-aminopyridine by NCI method	529
4.20.2 Studying weak interaction in DNA by NCI method based on promolecular density ..	532
4.20.3 Visually studying weak interaction for water in bulk environment by aNCI method ..	534
4.20.5 Simultaneously revealing covalent and noncovalent interaction in phenol dimer by DORI analysis	539
4.20.10 Visualize and quantify intermolecular and intramolecular interactions by means of Independent Gradient Model (IGM)	540
Example 1: Guanine-cytosine (GC) base pair.....	541
Example 2: C ₆₀ -coronene dimer.....	548
Example 3: Oxazolidinone trimer	549
4.21 Energy decomposition analysis.....	552
4.21.1 Examples of energy decomposition analysis based on forcefield (EDA-FF).....	552
Example 1: Water dimer.....	553
Example 2: Circumcoronene-Cytosine-Guanine trimer.....	556
4.100 Other functions (Part 1).....	562

4.100.4 Calculate kinetic energy and nuclear attraction potential energy of phosgene by numerical integration	562
4.100.8 Perform simple energy decomposition by using combined fragment wavefunctions	563
4.100.12 Perform biorthogonalization analysis for orbitals of unrestricted open-shell wavefunction.....	564
4.100.13 Calculate HOMA and Bird aromaticity index for phenanthrene.....	568
4.100.14 Calculate LOLIPOP for phenanthrene	569
4.100.15 Calculate intermolecular orbital overlap integral of DB-TTF.....	570
4.100.18 Yoshizawa's electron transmission route analysis for phenanthrene	571
4.100.19 ELF analysis on the promolecular wavefunction combined from fragment wavefunctions	572
4.100.21 Calculate molecular diameter and length/width/height.....	575
4.100.22 Analyze π electron character of non-planar system: cycloheptatriene	577
4.200 Other functions (Part 2).....	579
4.200.4 Study iso-chemical shielding surface (ICSS) and magnetic shielding distribution for benzene	579
4.200.5 Plot radial distribution function of electron density	583
4.200.7 Parse output file of “polar” task of Gaussian to obtain (hyper)polarizability and related quantities	587
4.200.8 Calculate polarizability and hyperpolarizability for NH_3 via sum-over-states (SOS) method.....	589
4.200.12 Calculate energy index (EI) and bond polarity index (BPI).....	592
4.200.14.1 Integrate real space functions within reduced density gradient (RDG) isosurface to study weak interaction quantitatively.....	593
4.200.14.2 Visualize molecular cavity and calculate its volume by domain analysis module	597
4.A Special topics and advanced tutorials.....	601
4.A.1 Study variation of electronic structure along IRC path	601
4.A.2 Calculation of spin population	606
4.A.3 Overview of methods for studying aromaticity	607
4.A.4 Overview of methods for predicting reactive sites	610
4.A.5 Overview of methods for studying weak interactions.....	612
4.A.6 Calculate odd electron density and local electron correlation function.....	615
4.A.7 Study (hyper)polarizability density	618
4.A.8 Analyze wavefunction higher than CCSD level	625
4.A.9 Calculate TrEsp (transition charge from electrostatic potential) charges and analyze excitonic coupling	627
4.A.10 Intuitively exhibiting atomic properties by coloring atoms.....	632
4.A.11 Overview of methods for studying chemical bonds	634
4.A.12 Overview of methods for analyzing electron excitation.....	640
4.A.13 Plot electrostatic potential colored van der Waals surface map and penetration graph of van der Waals surfaces	646
5 Skills.....	651

5.1 Make Multiwfn support more quantum chemistry programs.....	651
5.2 Running Multiwfn in silent mode	651
5.3 Running Multiwfn in batch mode	653
5.4 Copy outputs from command-line window to clipboard.....	655
5.5 Make command-line window capable to record more outputs	656
5.6 Rapidly load a file into Multiwfn.....	656
5.7 Make use of cubegen utility in Gaussian package to reduce computational time of electrostatic potential analyses	657

Appendix659

1 Setting up environment variable for Gaussian in Windows	659
2 The routines for evaluating real space functions.....	659
3 Detail of built-in atomic densities	661
4 Details about supplying inner-core electron density for the wavefunctions involving pseudo-potential.....	662
5 Check sanity of wavefunction.....	663
6 Special functions	663

1 Overview

Multiwfn is an extremely powerful wavefunction analysis program, it supports almost all of the most important wavefunction analysis methods. Multiwfn is free, open-source, high-efficient, very user-friendly and flexible. Multiwfn can be downloaded at Multiwfn official website <http://sobereva.com/multiwfn>.

Input files supported by Multiwfn

Multiwfn accepts several kinds of files for loading wavefunction information: .wfn/.wfx (Conventional / Extended PROAIM wavefunction file), .fch (Gaussian formatted check file), .molden (Molden input file), .31~.40 (NBO plot files) and .gms (GAMESS-US and Firefly output file). Other types such as Gaussian input and output files, .cub, .grd, .pdb, .xyz and .mol files can be used for specific functions.

Briefly speaking, Multiwfn can perform wavefunction analyses based on outputted file of almost all well-known quantum chemistry programs, such as Gaussian, ORCA, GAMESS-US, Molpro, NWChem, Dalton, xtb, PSI4, Molcas, Q-Chem, MRCC, deMon2k, Firefly, CFour, Turbomole ...

Special points of Multiwfn

(1) Very comprehensive functions. Almost all of the most important wavefunction analysis methods have been well supported by Multiwfn.

(2) Extremely user-friendly. Multiwfn is designed as an interactive program (but can also run silently), prompts shown on screen in each step clearly tell users what should input next. Multiwfn also never prints obscure messages, therefore there is no any barrier even for beginners. In addition, all wavefunction analysis theories are very detailedly documented, and there are more than one hundred of well-written examples in the manual, which are very helpful for new users. Moreover, The developer always very timely and patiently replies users' questions in Multiwfn official forum.

(3) High flexibility. The design of the overall framework, functions and user interface of Multiwfn is rather flexible, but this does not sacrifice ease of use. Different modules of Multiwfn are organically integrated together to make numerous analyses that single module cannot realize feasible

(4) High efficiency. The code of Multiwfn is substantially optimized. Most parts are parallelized by OpenMP technology. For computationally intensive tasks, the efficiency of Multiwfn exceeds analogous programs significantly.

(5) Results can be visualized directly. A high-level graphical library DISLIN is invoked internally and automatically by Multiwfn for visualizing results, most plotting parameters are controllable in interactive interface. This remarkably simplified wavefunction analysis, especially for studying distribution of real space functions.

Major functions of Multiwfn

- Showing molecular structure and viewing orbitals (MO, NBO, natural orbital, NTO, LMO, etc.). Speed of generating orbitals is extremely fast.
- Outputting all supported real space functions as well as gradient and Hessian at a point. Value can be decomposed to orbital contributions.
- Calculating real space function along a line and plot curve map.
- Calculating real space function in a plane and plot plane map. Supported graph types include filled-color map, contour map, relief map (with/without projection), gradient map and vector field map.
- Calculating real space function in a spatial scope, data can be exported to Gaussian-type cube grid file (.cub) and can be visualized as isosurface.
- For the calculation of real space functions in one-, two- and three-dimensions, user can define the operations between the data generated from multiple wavefunction files. Therefore one can calculate and plot such as Fukui function, dual descriptor and density difference very easily. Meanwhile promolecule and deformation properties for all real space functions can be calculated directly.
- Topology analysis for electron density (AIM analysis), Laplacian, ELF, LOL, etc. Critical points (CPs) and gradient paths can be searched and then be visualized in 3D window or plotted as plane graph. Interbasin surfaces can be drawn. Values of all supported real space functions can be calculated at critical points or along topology paths. CP properties can be decomposed as orbital contributions.
- Checking and modifying wavefunction. For example, print orbital and basis function information, manually set orbital occupation number and type, translate and duplicate system, discard wavefunction information from specified atoms.
- Population analysis. Hirshfeld, Hirshfeld-I, VDD, Mulliken, Löwdin, Modified Mulliken (including three methods: SCPA, Stout & Politzer, Bickelhaupt), Becke, ADCH (Atomic dipole moment corrected Hirshfeld), CM5, CHELPG, Merz-Kollmann, RESP (Restrained ElectroStatic Potential), AIM (Atoms-In-Molecules) and EEM (Electronegativity Equalization Method) are supported. Electrostatic interaction energy of two given fragments can be calculated based on atomic charges.
- Orbital composition analysis. Mulliken, Stout & Politzer, SCPA, Hirshfeld, Hirshfeld-I, Becke and natural atomic orbital (NAO) methods are supported to obtain orbital composition.
- Bond order analysis. Mayer bond order, multi-center bond order in AO or NAO basis (up to 12-centers), Wiberg bond order in Löwdin orthogonalized basis and Mulliken bond order are supported. Mayer and Mulliken bond order can be decomposed to orbital contributions. Wiberg bond order can be decomposed to contribution from various natural atomic orbital pairs.
- Plotting total, partial, overlap population density-of-states (TDOS, PDOS, OPDOS). Up to 10 fragments can be very flexibly and conveniently defined. Local DOS (LDOS) can also be plotted for a point as curve map or for a line as color-filled map.
- Plotting IR (infrared), normal/pre-resonance Raman, UV-Vis, ECD (electronic circular dichroism), VCD (vibrational circular dichroism) and Raman optical activity (ROA) spectra. Not only harmonic spectrum, but also anharmonic fundamental, overtone and

combination bands can be plotted. Spin-orbit coupling effect could be incorporated. Abundant parameters (broadening function, FWHM, scale factor, etc.) can be customized by users. Many data process functions are supported (*e.g.* finding maximum and minimum of the spectrum). Total spectrum can be decomposed to individual contribution of each transition. Spectrum of multiple systems can be conveniently plotted together. Plotting conformational weighted spectrum can be very easily realized.

- Quantitative analysis of molecular surface. Surface properties such as surface area, enclosed volume, average value and std. of mapped functions can be computed for overall molecular surface or local surfaces; local minima and maxima of mapped functions on the surface can be located.
- Processing grid data (can be loaded from .cub/.grd or generated by Multiwfn). User can perform mathematical operations on grid data, set value in certain range, extract data in specified plane, plot integral curve, etc.
- Adaptive natural density partitioning (AdNDP) analysis. The interface is interactive and the AdNDP orbitals can be visualized directly.
- Analyzing real space functions in fuzzy atomic spaces (defined by Becke, Hirshfeld or Hirshfeld-I partitions). These quantities can be computed: Integral of selected real space function in atomic spaces or in overlap regions of atomic spaces, atomic multipole moments, atomic overlap matrix (AOM), localization and delocalization index (DI), condensed linear response kernel, multi-center DI, as well as five aromaticity indices, namely FLU, FLU- π , PDI, PLR and information-theoretic index.
- Charge decomposition analysis (CDA) and extended CDA analysis. Orbital interaction diagram can be plotted. Infinite number of fragments can be defined.
- Basin analysis. Attractors can be located for any real space function, corresponding basins can be generated and visualized at the same time. All real space functions can be integrated in the generated basins. Electric multipole moments, orbital overlap matrix, localization index and delocalization index can be calculated for the basins. Atomic contribution to basin population can be obtained.
- Electron excitation analyses, including: Visualizing and analyzing hole-electron distribution, transition density, transition electric/magnetic dipole moment and charge density difference; calculating Coulomb attractive energy between hole and electron (exciton binding energy); calculating Mulliken atomic transition charges and TrEsp (transition charge from electrostatic potential); decomposing transition dipole moment to MO pair contribution or basis function/atom contribution; analyzing charge-transfer by the method proposed in *JCTC*, **7**, 2498; plotting atom/fragment transition density matrix, transition dipole moment matrix and charge transfer matrix as heat maps; calculating Δr index (*JCTC*, **9**, 3118) and Λ index (*JCP*, **128**, 044118) to reveal electron excitation character; calculating transition dipole moments between excited states; generating natural transition orbitals (NTOs); calculating ghost-hunter index (*JCC*, **38**, 2151); calculating amount of interfragment charge transfer via IFCT method; generating natural orbitals for a batch of excited states.
- Orbital localization analysis: Pipek-Mezey (based on Mulliken or Löwdin population) and Foster-Boys localization methods are supported. Composition, energy and dipole moment of the resulting LMOs can be derived, shape and center of the LMOs can be easily

visualized. Furthermore, based on the LMOs, oxidation states can be evaluated via LOBA method (*PCCP*, **11**, 11297).

- Visual study of weak interaction: NCI method (*JACS*, **132**, 6498), aNCI method (noncovalent interaction analysis in fluctuating environments, *JCTC*, **9**, 2226), DORI method (*JCTC*, **10**, 3745), independent gradient model (IGM) method (*PCCP*, **19**, 17928). Scatter map can be directly plotted, volume enclosed by isosurface of related real space function can be integrated for quantitative analysis. Becke and Hirshfeld surface analyses, as well as fingerprint analysis are also supported.
- Energy decomposition analysis (EDA): EDA based on UFF/AMBER/GAFF molecular force fields, as well as simple energy decomposition (relies on Gaussian).
- Other useful functions or utilities involved in quantum chemistry studies (incomplete list): Integrating a real space function over the whole space by Becke's multi-center method; evaluating overlap integral between alpha and beta orbitals; evaluating overlap and centroid distance between two orbitals; generating Gaussian input file with initial guess from converged wavefunction or multiple fragment wavefunctions; calculating van der Waals volume; calculating HOMA and Bird aromaticity indices; calculating LOLIPOP index; calculating intermolecular orbital overlap; Yoshizawa's electron transport route analysis; calculating atomic and bond dipole moment in Hilbert space; plotting radial distribution function for real space functions; plotting iso-chemical shielding surface (ICSS); calculating overlap integral between orbitals in two different wavefunctions; parsing output of (hyper)polarizability task of Gaussian; calculating polarizability and 1st/2nd/3rd hyperpolarizability by sum-over-states (SOS) method; outputting various kinds of integrals between orbitals; evaluating the first and second moments and radius of gyration for a real space function; exporting loaded structure/wavefunction to many popular formats such as .wfn, .wfx, .molden, .fch, NBO .47, .pdb, .xyz and yield input file for a lot of known quantum chemistry codes; calculating bond polarity index (BPI); Domain analysis (obtaining properties within isosurfaces defined by a real space function); calculating electron correlation indices; automatically detecting π orbitals; evaluating molecular diameter and length/width/height; perform biorthogonalization between alpha and beta orbitals to maximally pair them; Evaluating interatomic connectivity and atom coordination numbers based on geometry; Evaluating core-valence bifurcation (CVB) index.

Real space functions supported by Multiwfn

Real space function analysis is one of the most powerful features of Multiwfn, more than 100 real space functions are supported and listed below, detailed descriptions can be found in Section 2.6 and 2.7 of the manual:

- Electron density
- Gradient norm of electron density
- Laplacian of electron density
- Value of orbital wavefunction
- Electron spin density
- Hamiltonian kinetic energy density $K(\mathbf{r})$
- Lagrangian kinetic energy density $G(\mathbf{r})$

- Electrostatic potential from nuclear / atomic charges
- Electron localization function (ELF) defined by Becke and the one defined by Tsirelson
- Localized orbital locator (LOL) defined by Becke and the one defined by Tsirelson
- Local information entropy
- Electrostatic potential (ESP)
- Reduced density gradient (RDG) with/without promolecular approximation
- $\text{Sign}(\lambda_2)\rho$ (The product of the sign of the second largest eigenvalue of electron density Hessian matrix and electron density) with/without promolecular approximation
- Exchange-correlation density, correlation hole and correlation factor
- Average local ionization energy
- Source function
- Electron delocalization range function EDR($\mathbf{r};\mathbf{d}$) and orbital overlap distance function D(\mathbf{r})
(Related code was kindly contributed by Arshad Mehmood)
 - The δg function defined in Independent Gradient Model (IGM)

Others (incomplete list): potential energy density, electron energy density, strong covalent interaction index (SCI), shape function, local temperature, bond metallicity, linear response kernel, local electron affinity/electronegativity/hardness, ellipticity of electron density, eta index, on-top pair density, numerous forms of DFT exchange-correlation potential, numerous forms of DFT kinetic energy density, Weizsäcker potential, Fisher information entropy, Ghosh/Shannon entropy density, integrand of Rényi entropy, steric energy/potential/charge, Pauli potential/force/charge, quantum potential/force/charge, PAEM, density overlap regions indicator (DORI), region of slow electrons (RoSE), PS-FID, single exponential decay detector (SEDD), electron linear momentum density, electric/magnetic dipole moment density, local electron correlation function, magnitude of electric field.

Implementing a new real space function into Multiwfn is extremely easy, as illustrated in Section 2.7 of the manual.

Things that Multiwfn can do

The analyses that Multiwfn can realize for different topics are briefly listed below, you can easily find corresponding introduction and examples by searching manual. Do not forget to ask question in Multiwfn forum when you are confused!

- Visualizing various kinds of orbitals generated by various programs in various forms
- Characterizing chemical bonds: Various form of AIM analyses; studying real space functions (ELF, LOL, $\nabla^2\rho$, kinetic/potential energy density, valence ρ , fragment density difference, deformation density, source function, bond ellipticity, bond degree, eta index, $V(\mathbf{r})/G(\mathbf{r})$, SCI, DORI, PAEM, IGM...); various kinds of bond orders analysis (Mayer, Laplacian, Mulliken, Wiberg, Fuzzy and multi-center bond orders, as well as decomposition analysis for Mayer, Mulliken and Wiberg bond orders); localization/delocalization index; orbital localization analysis; various methods of measuring bond polarity and bond dipole moment; charge decomposition analysis (CDA); overlap population density-of-states (OPDOS); energy decomposition analysis and so on. See Section 4.A.11 of manual for an overview. Variation of various properties of chemical bonds during scan and IRC processes can also be easily studied via shell scripts, see Section 4.A.1.
- Characterizing electron distribution and variation: Atomic charges (AIM, Mulliken, SCPA, Hirshfeld, Hirshfeld-I, Voronoi, Löwdin, ADCH, CM5, EEM, CHELPG, MK, RESP...); total and

spin population analyses for basis functions/shells/atoms/fragments; atomic dipole and multipole moment analysis; plotting / basin analysis / domain analysis for density difference; charge displacement curve

- Aromaticity and electron delocalization analysis: ICSS; AdNDP; ELF- σ/π ; LOL- σ/π ; HOMA; Bird; multi-center bond order; NICS; Shannon aromaticity; FLU and FLU- π ; PDI; ATI; PLR; Δ DI; density curvature perpendicular to ring plane and so on. See Section 4.A.3 for an overview
- Characterizing intramolecular and intermolecular weak interactions: AIM analysis; visual analyses (NCI, IGM, DORI); plotting and quantitative molecular surface analysis for electrostatic potential (ESP); energy decomposition analysis based on forcefield; Hirshfeld/Becke surface analysis; LOLIPOP; mutual penetration distance and penetration volume analysis; atomic charge and multipole moment analysis; charge transfer analysis (density difference map, CDA, variation of population ...); ELF and core-valence bifurcation (CVB) index and so on. See Section 4.A.5 for an overview
 - Electron excitation analysis: Analysis of hole and electron (distribution, atom/fragment/orbital contribution, centroid position, displacement and overlap, exciton binding energy); charge transfer analysis (IFCT, density difference...); NTO; overlap and centroid distance between crucial MOs; plotting atom/fragment transition density matrix and charge transfer matrix; Δr index; decomposition of transition dipole moment to basis function/atom/fragment/MO pair contributions; transition dipole moment between various excited states; transition atomic charge; ghost-hunter index; revealing variation of electronic structure (bonding and population) during excitation and so on. See Section 4.A.12 for an overview
 - Prediction of reactive sites: ESP and ALIE analyses on molecular surface; Fukui function and dual descriptor as well as their condensed forms; atomic charges; orbital composition analysis for frontier molecular orbitals; population of π electron; orbital overlap distance function analysis. See Section 4.A.4 for an overview
 - Prediction properties of molecular condensed phase: Using ESP distribution on vdW surface to empirically predict heat of vaporization, heat of sublimation, density of molecular crystal, boiling point, heat of fusion, surface tension and so on. See Section 3.15.1
 - Plotting IR, Raman, UV-Vis, ECD, VCD and ROA spectra based on output file of quantum chemistry codes
 - Characterizing molecular structure: Evaluating molecular volume, length/height/weight, diameter, interatomic connectivity and atomic coordination number, average bond length of atomic cluster, cavity volume and so on
 - (Hyper)polarizability study: Parsing Gaussian output file of "polar" task; plotting (hyper)polarizability density; obtaining atomic contribution to (hyper)polarizability; calculating (hyper)polarizability by means of sum-over-states method
 - Many others: Teaching structure chemistry; converting file formats; studying electron correlation effect; realizing ELF-tuning and LOL-tuning for DFT functionals; evaluating oxidation state by LOBA method; electric conduction analysis (TDOS and PDOS; orbital overlap analysis between neighbouring monomer; Yoshizawa's transport route analysis); Studying distribution of real space functions (in terms of radial distribution function, centroid, first and second moments, integral over whole space and local region...) and so on

Citing Multiwfn

The best way of supporting development of Multiwfn is properly citing my related works, as shown below.

If Multiwfn is used in your research, **at least** this paper must be cited:

Tian Lu, Feiwu Chen, Multiwfn: A Multifunctional Wavefunction Analyzer, *J. Comput. Chem.* **33**, 580-592 (2012)

If quantitative molecular surface analysis module of Multiwfn is involved, citing below paper is also needed, in which the algorithm was described in detail

Tian Lu, Feiwu Chen, Quantitative analysis of molecular surface based on improved Marching Tetrahedra algorithm, *J. Mol. Graph. Model.*, **38**, 314-323 (2012)

If orbital composition analysis module of Multiwfn is involved, citing below paper is highly recommended, in which different orbital composition calculation methods are detailedly discussed

Tian Lu, Feiwu Chen, Calculation of Molecular Orbital Composition, *Acta Chim. Sinica*, **69**, 2393-2406 (2011) (in Chinese)

If charge decomposition analysis (CDA) module of Multiwfn is involved, citing below paper is recommended, in which the generalized CDA method that implemented in Multiwfn is introduced

Meng Xiao, Tian Lu, Generalized Charge Decomposition Analysis (GCDA) Method, *Journal of Advances in Physical Chemistry*, **4**, 111-124 (2015) (in Chinese)
<http://dx.doi.org/10.12677/JAPC.2015.44013>

Citing Multiwfn website in your work is also welcomed, but **do not ONLY** cite the website, thanks!

Discussion zones

There are two Multiwfn official forums, with different languages. You can discuss anything about Multiwfn and wavefunction analysis in either one. If you encountered problems in using Multiwfn, please do not hesitate to post topic on these forums!

Multiwfn English forum: <http://sobereva.com/wfnbbs>

Multiwfn Chinese forum: <http://bbs.keinsci.com/wfn>

2 General information

2.1 Install

2.1.1 Windows version

What you need to do is just uncompressing the program package. You can place Multiwfn anywhere, however the length of the path (including file name) should be less than 80 characters, otherwise some unexpected problems may occur. A few functions in Multiwfn rely on Gaussian, you need to setup environment variables for Gaussian manually, see Appendix 1.

2.1.2 Linux version

1 Uncompress the program package to for example /sob/Multiwfn. Notice that the path (including file name) should be less than 80 characters.

2 Make sure that you have installed motif package, which provides libXm.so.4, Multiwfn cannot boot up without this file. The motif is freely available at <http://motif.ics.com/motif/downloads>. If you are a CentOS or Red Hat Linux user and have not installed motif, download corresponding rpm package (*e.g.* motif-2.3.4-1.x86_64.rpm) and install it; alternatively, you can directly run command *yum install motif* to install it. If you are an Ubuntu user, download and install corresponding deb package (*e.g.* libmotif4_2.3.4-1_amd64.deb) or directly run *sudo apt-get install libmotif4* to install it.

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

```
export KMP_STACKSIZE=200000000
```

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 large enough (unit is in bytes), if the value is too small, Multiwfn may crash when analyzing big wavefunction. In order to enlarge the size, for example you can add *kernel.shmmmax = 512000000* to /etc/sysctl.conf and reboot system, then the upper limit will be enlarged to about 512MB.

5 Assume that you are using Bash shell, and you have decompressed the Multiwfn package as “/sob/Multiwfn_3.6_bin_Linux” folder, you should add below lines into ~/.bashrc file:

```
export Multiwfnprefix=/sob/Multiwfn_3.6_bin_Linux
export PATH=$PATH:/sob/Multiwfn_3.6_bin_Linux
```

Then after re-entering the terminal, you can normally boot up Multiwfn anywhere by simply run the command *Multiwfn*.

If you are using Ubuntu, you must add *ulimit -s unlimited* in your .bashrc file to set the available stacksize memory to be formally unlimited. Because default stacksize of Ubuntu system is quite small (merely 8MB in some versions!), Multiwfn will crash when dealing with large wavefunctions. Some other Linux distributions also have this severe problem.

Linux version of Multiwfn works well on Red Hat Enterprise Linux 6 & 7, CentOS 6 & 7 and

Ubuntu 12/14/16. I can not guarantee that the program is completely compatible with all 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.

If you encounter difficulty when running/compiling Multiwfn due to missing or incompatibility of some graphics related library files, and meantime you do not need any visualization function of Multiwfn, you can run/compile Multiwfn without GUI supported, all functions irrelevant to GUI and map plotting will still work normally. Please check document of compilation method in source code package on how to compile this version, the pre-compiled binary of this version can also be downloaded from Multiwfn website (termed as noGUI version).

2.1.3 Mac OS version

If you are using relatively old Mac OS version, e.g. OS X 10.8, please follow the instruction given in this section. If you intend to install Multiwfn in relatively new Mac OS systems, such as OS X 10.11 El Capitan and macOS 10.12 Sierra, please follow steps in this page: <https://wiki.ch.ic.ac.uk/wiki/index.php?title=Mod:multiwfn>.

1 Uncompress the program package to for example /Users/sob/Multiwfn. Notice that the path (including file name) should less than 80 characters.

2 Download .dmg file of Mac OS version of motif from http://www.ist-inc.com/downloads/motif_download.html and then install it. The motif package I installed is openmotif-compat-2.1.32_IST.macosx10.5.dmg.

If your system does not natively support X11 (i.e. OS X Mountain Lion), you should download XQuartz from <http://xquartz.macosforge.org/landing> and install it.

3 Add below sentence to your .profile file (e.g. /Users/sob/.profile) to make them take effect automatically, then reboot your terminal. If the .profile is unexisting, you should create it manually.
`export KMP_STACKSIZE=64000000`

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

4 Run `sysctl -a|grep shmmmax` to check if the size of SysV shared memory segments is large enough (unit is in bytes), if the value is too small, Multiwfn may crashes when analyzing big wavefunction. In order to enlarge the size, you should edit or create the file /etc/sysctl.conf, and add `kern.sysv.shmmmax = 512000000` to it and reboot system, then the upper limit will be enlarged to about 512MB.

5 Move libdislin_d.11.dylib in the Multiwfn package to /usr/local/lib folder.

6 Set `Multiwfnprompt` environment variable if needed, see point 5 of Section 2.1.2.

Now enjoy!

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 tutorials in Chapter 4.

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

path of the input file successfully read at last time was recorded in settings.ini). If the input file is in the same path as the one last time used, for convenience, the path can be replaced by symbol ?. For example, last time you loaded C:\sob\wives\K-ON\Mio.wfn, this time you can simply input ?Azusa.fch to load C:\sob\wives\K-ON\Azusa.fch. If you preferred to select file in GUI window, you can directly press ENTER button, then a GUI window will be shown used to select input file.

You can press CTRL+C or click “ \times ” button at right-top of command-line window any time to exit Multiwfnn, a more graceful way of exiting Multiwfnn is selecting option -10 (a hidden option) in main menu. 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. Another way (an elegant way) to exit Multiwfnn is selecting option -10 (an hidden option) in main function menu.

If you want to load another file into Multiwfnn, you can reboot Multiwfnn or start a new Multiwfnn instance. Alternatively, in main function menu you can select option -11 (a hidden option) to re-initialize Multiwfnn and load a new file, at the meantime the settings.ini is also reloaded. However, please remember that the safest way of loading a new file is rebooting Multiwfnn.

Multiwfnn also supports silent mode, by which users do not need to press any keyboard button during running, it is useful for batch processing or carrying out similar tasks, please consult Sections 5.2 and 5.3.

2.3 Files of Multiwfnn

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

Multiwfnn.exe (Windows) or **Multiwfnn** (Linux/Mac OS) : The executable file of Multiwfnn.

libomp5md.dll (Windows) : Intel OpenMP Runtime library.

settings.ini : All detail parameters for running Multiwfnn are recorded here, most of them do not need to be frequently modified. When booting up, Multiwfnn will try to find and use this file in current folder, if it is not presented in current folder, the file in the path defined by "Multiwfnnpath" environment variable will be used; if the file still cannot be found, default settings will be used instead. The meanings of all parameters in settings.ini are not documented in this manual systematically, since they have already been commented in detail, only those important will be mentioned in this manual. I suggest you read through settings.ini and find out the ones useful for you.

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

LICENSE.txt: The terms that all users must follow.

2.4 Parallel implementation

Most time-consuming codes of Multiwfnn have been parallelized by OpenMP. If your computer is SMP architecture, you can greatly 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 physical 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 or Mac OS version).

2.5 Input files and wavefunction types

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

For basis function, Cartesian or spherical harmonic Gaussian functions with angular moment up to \mathbf{h} 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 file extension. Notice that different function need different types of information, 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/.molden/.gms/.31~.40/.wfn/.wfx file as input, but .pdb, .xyz, .mol, etc. do not 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 requirement on information types by each function is commonly described at the end of corresponding section in Chapter 3 by red text.

File Format	Contained information types				
	Basis functions	GTFs	Atom coordinates	Grid data	Atomic charges
.fch/.fchk	✓	✓	✓	✗	✗
.molden	✓	✓	✓	✗	✗
.gms	✓	✓	✓	✗	✗
.31~.40	✗	✓	✓	✗	✗
.wfn	✗	✓	✓	✗	✗
.wfx	✗	✓	✓	✗	✗
.pdb .xyz .mol .gif	✗	✗	✓	✗	✗
.chg .pqr	✗	✗	✓	✗	✓
.cub/.cube	✗	✗	✓	✓	✗
DMol3 .grd	✗	✗	✗	✓	✗
Plain text file	✗	✗	✗	✗	✗

AIM 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 and NWChem. The information of .wfn files include atomic coordinates/types, orbital energies, occupation numbers, expansion coefficients of Cartesian Gaussian type functions (GTF). Supported angular momentum of GTF is up to f . The wfn file does not contain any virtual orbital. The generation method of .wfn file is documented at the beginning of Chapter 4.

Note: Although GTFs with angular moment of g and h are not formally supported by original .wfn format, if g and h-type GTFs are recorded in following manner, then Multiwfn is able to recognize them: 21~35 in "TYPE ASSIGNMENT" correspond to YZZZ, XYYY, XXYY, XYZZ, YZZZ, XYYZ, XXXX, XXXY, XZZZ, XXYZ, XXXZ, XXZZ, YYYZ, ZZZZ, respectively. 36~56 correspond to ZZZZ, YZZZ, YYZZ, YYZZ, YYYY, YYYY, XZZZ, XYZZ, XYYZ, XYYYY, XXZZ, XXYZ, XXYY, XXXY, XXXZ, XXXYZ, XXXYY, XXXXZ, XXXXY, XXXXX, respectively. The sequence shown here in fact is also the sequence used in the .wfn outputted by Molden2AIM and Gaussian09 since B.01.

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. Currently the real space functions supported EDFs in Multiwfn include: electron density, its gradient and Laplacian, local information entropy, reduced density gradient as well as $Sign(\lambda_2(\mathbf{r}))$. Meanwhile topology analyses of electron density and its Laplacian also take into account EDFs. **Notice that EDF informations have neither effect on ESP nor the real space functions that relied on wavefunction (e.g. kinetic energy density, ELF).** If you want to analyze these properties for heavy elements, you should use full-electrons basis sets, at least small-core ECP. Currently the only supported GTF type in EDF field is S-type (actually S-type is enough for fitting inner density, since which is nearly spherical symmetry). Like .wfn, Multiwfn does not allow virtual orbitals presented in .wfx file.

Multiwfn has a powerful built-in EDF library, taken from Molden2aim program developed by Wenli Zou. **As long as the input file contains GTF information (e.g. .fch, .wfn, .molden, .gms...), Multiwfn always automatically loads EDF information from this library for the atoms using pseudopotential.** Only when you use .wfx file as input and the .wfx itself already contains EDF field, the EDF information will be loaded from the .wfx file rather than from EDF library. See Appendix 4 for more details.

Note that although some programs other than Gaussian can also generate .wfx file (e.g. ORCA), these .wfx files are unable to provide EDF field.

Notice: For certain version of Gaussian (e.g. G09 B.01), I found that the EDF field recorded in .wfx for rare cases is problematic, namely the number of electrons represented by EDF field is unequal to the actual number of core electrons exhibited by ECP. In order to verify if the EDF field is correct, you can use subfunction 4 in main function 100 to obtain the integral of total electron density over the whole space, if the result is approximately equal to the total number of electrons (core+valence electrons), that means the EDF field is correct.

Gaussian formatted checkpoint file (.fch/.fchk) : Checkpoint file of Gaussian program (.chk) can be converted to formatted checkpoint file (.fch/.fchk) via *formchk* utility in Gaussian package. There is no any difference between .fch and .fchk. ".fch" ("fchk") is the default extension generated by Windows (Linux) version of *formchk*.

.fch/.fchk contains richer information than .wfn/.wfx files, virtual orbital wavefunctions are also recorded, and meanwhile it provides basis function information for Multiwfn. If you want to

use .fch/.fchk file as carrier for post-HF wavefunction, read the beginning of Chapter 4 carefully!

Notice that for single-determinant wavefunctions, before some calculations involving real space function (e.g. main function 2, 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 disable this treatment, set "idelvirorb" in settings.ini to 0.

The .fchk file generated by Q-Chem and PSI4 can also be used as input file of Multiwfn. Notice that before loading the .fchk file generated by Q-Chem into Multiwfn, you must set "ifchprog" in settings.ini to 2.

Molden input file (.molden or .molden.input or molden.inp) : Currently, a wide variety of quantum chemistry packages, such as Molpro, Molcas, ORCA, Q-Chem, CFour, Turbomole, PSI, MRCC and NWChem are able to produce input file of Molden visualization program. This type of file records atomic coordinates, basis set definition, information of all occupied and virtual orbitals (including expansion coefficient of basis functions, occupation number, spin, energy and symmetry), meanwhile there is no information only specific for Molden. So in fact, Molden input file can be regarded as a standard and general file format for exchanging wavefunction information. For Multiwfn, this type of file can provide atomic coordinate, basis function information and GTF information. **.molden file only supports basis functions up to g angular moment**.

Beware that the Molden input files produced by a lot of program are quite non-standard! **Currently Multiwfn only formally supports the Molden input file generated by Molpro, ORCA, xtb, Dalton, NWChem (only for spherical harmonic functions and meantime symmetry is disabled), MRCC (only for spherical harmonic functions), deMon2k, BDF**. If the Molden input file you used is generated by other programs, the analysis result may or may not be correct, you should first use the methods described in Appendix 5 to check if the wavefunction has been correctly loaded.

Hint: Multiwfn fully supports the Molden input file standardized by molden2aim utility (see Section 5.1 for detail), which is able to properly recognize Molden input files generated by many other quantum chemistry codes, such as CFOUR and Molcas.

Although Molden input file also supports Slater type orbital (STO), Multiwfn can only utilize the Molden input file recording Gaussian type basis functions.

One drawback of Molden input file is that it does not explicitly record nuclear charges as other formats such as wfn and fch, therefore the results relying on nuclear charges (e.g. electrostatic potential) will be problematic when pseudo-potential is used. To address this problem, Multiwfn loads atomic indices in the file (i.e. the third column in [Atoms] field) as nuclear charges, thus if you manually change the atomic index to the number of atomic valence electrons that explicitly represented in the quantum chemistry calculation, then the result will be correct. (Note that Multiwfn determines element of each atom according to atomic name in .molden file, therefore even after above modification, the elements can still be correctly identified)

A disadvantage of using Molden input file as wavefunction carrier is that its format is not as compact as .fch. Due to this reason, for the same wavefunction, loading speed of .molden file is much slower than .fch. Therefore, if you need to frequently analyze a .molden file, I suggest you use subfunction 2 of main function 100 to convert it to .fch format.

In analogy to .fch file, Multiwfn may delete virtual orbitals higher than LUMO+10. To avoid this, you should set "idelvirorb" in settings.ini to 0.

The way of generating Molden input file by a few quantum chemistry programs are described

at the beginning of Chapter 4.

PS: Detailed description about .molden format can be found on Molden official site:
http://www.cmbi.ru.nl/molden/molden_format.html.

GAMESS-US or Firefly output file (.gms): If you want to use GAMESS-US or Firefly (originally known as PC-GAMESS) output file as input file, you can change its suffix as .gms, then Multiwfn will properly recognize it. Currently, I can only guarantee that output file of HF/DFT/TDDFT calculation with default NPRINT option can be normally loaded by Multiwfn. The role of .gms is similar as .molden and .fch file, *i.e.* all of them provide atomic coordinates, GTF and basis function informations.

Notice that since I am not a experienced Firefly user, I cannot guarantee that the compatibility with Firefly output files is as good as GAMESS-US output files. For the former I only tested DFT single point task and TDDFT task.

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/MO (their orbital coefficients are recorded in .32~.40 respectively), .31 recorded basis function information. After boot up Multiwfn, you should input the path of .31 file first, and then input the path of one of .32~.40 files (for simplicity, you can only input the suffix when the filenames are identical).

Notice that among all types of the orbitals generated by NBO program, only using NBO or NLMO to calculate real space functions is meaningful!

Protein data bank format (.pdb), .xyz, MDL Molfile (.mol): These are the most widely used formats for recording atom coordinates. They do not carry any wavefunction information, but for the functions which only require atom coordinates, using these kind of files as input is enough. A unique advantage of .mol with respect to .pdb and .xyz is that it contains atomic connectivity table, which is need by a few functions of Multiwfn, *e.g.* Calculation of EEM atomic charges.

Notice that the .mol file supported by Multiwfn is V2000 version, both the maximum number of atoms and bonds that can be recorded are 999. More description about .mol format can be found in https://en.wikipedia.org/wiki/Chemical_table_file.

Gaussian job file (.gjf): This means input file of Gaussian program, it can provide atom coordinate information as well as the number of α and β electrons information to Multiwfn. Note that the atoms must be recorded as Cartesian coordinate.

Charge files (.chg) : This type of plain text file can be generated by some functions of Multiwfn (*e.g.* population analysis functions), it contains 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. This file is free-formatted, all fields must be delimited by white-space. This file can provide atomic charge information, the main use of which is to visualize electrostatic potential and analyze it on molecular surface based on atomic charges, the electrostatic interaction energy based on atomic charges can also be evaluated by subfunction -2 of main function 7 using .chg as input file. When .chg file is loaded, the sum of all atomic charges as well as electric dipole moment calculated using the atomic charges will be shown on screen.

An example of .chg file of water molecule is given below:

```
O    0.000000  0.000000  0.119308 -0.301956
H    0.000000  0.758953 -0.477232  0.150977
H    0.000000 -0.758953 -0.477232  0.150977
```

.pqr file : This format is very similar to .pdb format, but with different content. Behind the columns corresponding to atomic X/Y/Z coordinate, there are two columns recording atomic charges and atomic radii, respectively (the number of decimals of the two columns is not important, the fields must be delimited by white-space). This kind of file can provide atom information as well as atomic charge information to Multiwfn. Below is an example .pqr file of water. The REMARK field could exist to record comments, they will be skipped during loading the file.

```
REMARK From file m1charges.out
REMARK ESP charges
ATOM    1 O    0 1    1      0.000  0.123  0.000 -0.680698 2.9000
ATOM    2 H    0 1    1      0.757 -0.490  0.000  0.340338 2.6000
ATOM    3 H    0 1    1     -0.757 -0.490  0.000  0.340361 2.6000
```

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, 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.

DMol3 grid file (.grd) : .grd file is the volumetric data format mainly used by DMol3 program. This file is very similar to .cub file, but atomic information is completely omitted.

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. Gaussian output files also belong to this type.

2.6 Real space functions

The "Real space functions" in this manual referred to as the functions whose variables are coordinate of the three-dimension space of present system. Real space function analysis is one of the most important functions of Multiwfn, the supported real space functions are listed below. All wavefunctions are assumed to be real type, all units are in **atomic unit (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 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 , φ is orbital wavefunction, 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 . Atomic unit for electron density can be explicitly written as Bohr³.

When input file does not contain GTF information, this function will be calculated as promolecular density, which is approximate molecular electron density simply constructed by superposing built-in spherically averaged free-state atomic density of all atoms in the system. See Appendix 3 on how the built-in atomic density were derived.

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

Spin density is defined as the difference between alpha and beta 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 returned instead of spin density

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

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 clearer and more commonly used. $G(\mathbf{r})$ is also known as positive definite kinetic energy density.

$$G(\mathbf{r}) = \frac{1}{2} \sum_i \eta_i |\nabla \varphi_i(\mathbf{r})|^2 = \frac{1}{2} \sum_i \eta_i \left\{ \left(\frac{\partial \varphi_i(\mathbf{r})}{\partial x} \right)^2 + \left(\frac{\partial \varphi_i(\mathbf{r})}{\partial y} \right)^2 + \left(\frac{\partial \varphi_i(\mathbf{r})}{\partial z} \right)^2 \right\}$$

Since $\nabla \rho_i = \nabla(\eta_i \varphi_i^2) = 2\eta_i \varphi_i \nabla \varphi_i$, there is an equivalent form of $G(\mathbf{r})$: $\frac{1}{8} \sum_i \frac{\nabla \rho_i \cdot \nabla \rho_i}{\rho_i}$.

$K(\mathbf{r})$ and $G(\mathbf{r})$ are directly related by Laplacian of electron density

$$\nabla^2 \rho(\mathbf{r}) / 4 = G(\mathbf{r}) - K(\mathbf{r})$$

8 Electrostatic potential from nuclear / atomic charges

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

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_{nuc} is useful for analyzing the difference between exact electrostatic potential and the electrostatic potential reproduced by atomic charges.

Notice that at nuclear positions, this function will be infinite and may cause some numerical problems in program, hence at these cases this function always returns 1000 instead of infinity.

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 (1990). The ELF used in Multiwfnn is generalized for spin-polarized system, see [*Acta Phys. -Chim. Sin.*, 27, 2786 \(2011\)](#) for introduction. For a review, see Chapter 5 of *Theoretical Aspects of Chemical Reactivity* (2007).

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

where

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

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

For closed-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 (1992), which makes ELF also meaningful for Kohn-Sham DFT wavefunction and even multi-configuration wavefunction. The first term of $D(\mathbf{r})$ can be seen as the exact kinetic energy density of the noninteracting electron system defined by KS-DFT theory, namely $\tau_s(\mathbf{r}) = (1/2) \sum_i \eta_i |\nabla \varphi_i(\mathbf{r})|^2$, while the second term is equivalent to Weizsäcker kinetic energy

density $\tau_w(\mathbf{r}) = (1/8) |\nabla \rho|^2 / \rho$, therefore the $D(\mathbf{r}) = \tau_s(\mathbf{r}) - \tau_w(\mathbf{r})$ reveals the excess kinetic energy density caused by Pauli repulsion and it is known as Pauli kinetic energy density. The $D_0(\mathbf{r})$ can be interpreted as Thomas-Fermi kinetic energy density τ_{TF} , which is the exact kinetic energy density of noninteracting, uniform electron gas. Since $D_0(\mathbf{r})$ is introduced into ELF as reference, what the ELF reveals is actually degree of *relative* localization.

ELF is within the range of [0,1]. A large ELF value means that electrons are greatly localized, indicating that there is a covalent bond, a lone pair or inner shells of the 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 revealing atomic shell structure, classification of chemical bonding, verification of charge-shift bond, studying aromaticity.

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 does not 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 (2002), in which the actual kinetic energy term in $D(\mathbf{r})$ is replaced by Kirzhnits type second-order gradient expansion, that is

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

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.

If “ELFOL_type” is set to 2, another formalism will be used:

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

A real space function closely related to ELF is SCI index, it is very useful for identifying strong covalent bonds, see introduction of user-defined function 37 in Section 2.7.

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. (THEOCHEM)*, **527**, 51 (2000).

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

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 closed-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 (2008). 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 (2002)), 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 “ELFOL_type” to 1.

For special reason, if “ELFOL_type” in settings.ini is changed from 0 to 2, another formalism

will be used: $\text{LOL}(\mathbf{r}) = \frac{1}{1 + [1/\tau(\mathbf{r})]^2}$.

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), Noorizadeh and Shakerzadeh suggested using information entropy to study aromaticity (*Phys. Chem. Chem. Phys.*, **12**, 4742). 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 over whole space yields information entropy.

12 Total electrostatic potential (ESP)

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

where Z is nuclear charge, the index A cycles over all atoms. If pseudo-potential is used, then Z_A is the number of explicitly represented electrons. The atomic unit (a.u.) of ESP is Hartree/e, where e is elementary charge. Other commonly used unit are eV/e and kcal/(mol·e), however for simplicity, in Multiwfn they are printed as eV and kcal/mol, respectively. The value in eV/e unit is equal to the value in SI unit (J/C).

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. There are a lot of reviews on ESP, interested readers are suggested to consult *WIREs Comput. Mol. Sci.*, **1**, 153 (2011), *Theor. Chem. Acc.*, **108**, 134 (2002), Chapter 17

of the book *Chemical Reactivity Theory-A Density Functional View*, the entry "Electrostatic Potentials: Chemical Applications" (page 912) in the book *Encyclopedia of Computational Chemistry* and Chapter 7 of the book *Reviews in Computational Chemistry* vol.2.

By the way, if you only want to obtain the electrostatic potential contributed by electrons, namely $V_{\text{ele}}(\mathbf{r})$, you can use the 14th user-defined function. If you want to omit contribution of specific nucleus during evaluating $V_{\text{ESP}}(\mathbf{r})$, use 39th user-defined function. See corresponding entries in Section 2.7 for details.

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.

The calculation speed of ESP of cubegen utility in Gaussian package is much faster than Multiwfn. If you have Gaussian installed on your system, it is **strongly** recommended to set *cubegenpath* parameter in settings.ini file to actual path of cubegen, so that cubegen could be automatically invoked by Multiwfn to evaluate ESP. Please check Section 5.7 for detail.

13 Reduced density gradient (RDG)

RDG and $\text{sign}(\lambda_2)\rho$ are a pair of very important functions for revealing weak interaction regions, they are collectively employed in NCI method, see *J. Am. Chem. Soc.*, **132**, 6498 (2010) for detail, and application examples in Section 3.23.1. RDG is defined as

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

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; however reproduction of electron density by *ab initio* and grid 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 rapidly

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

where $\rho_A^{\text{free,fit}}(\mathbf{r})$ is pre-fitted spherically averaged electron density of atom A . The atomic densities for H~Lr are built-in data of Multiwfn, among which the data for H~Ar are taken from supplemental material of *J. Am. Chem. Soc.*, **132**, 6498 (2010), while those for other elements are evaluated according to the description in Appendix 3. For elements heavier than Lr the promolecular approximation is not currently available.

For efficiency consideration, if contribution from 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 the result is almost unperturbed. You can also disable this treatment by setting “atomdencut” in settings.ini to 0.

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

15 $\text{Sign}(\lambda_2)\rho$

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

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 $\text{Sign}(\lambda_2)\rho$ with promolecular approximation

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

17 Exchange-correlation density, correlation hole and correlation factor

These functions involve advanced topics, in order to clarify their physical meanings and to avoid confusion of the symbols used in a wide variety of literatures, I think it is worth to use much more texts to introduce theoretical background. For more detail discussions, please consult Chapter 2 of *A Chemist's Guide to Density Functional Theory 2ed* and Chapter 5 of *Methods of molecular quantum mechanics (2ed, McWeeny)*.

===== Theoretical basis =====

Pair density is defined as

$$\pi(\mathbf{r}_1, \mathbf{r}_2) = N(N-1) \int \int \cdots \int |\Psi(\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_N)|^2 d\sigma_1 d\sigma_2 d\mathbf{x}_3 d\mathbf{x}_4 \dots d\mathbf{x}_N$$

where Ψ is system wavefunction, \mathbf{r} is space coordinate, σ is spin coordinate, \mathbf{x} is space-spin coordinate. Pair density denotes the probability that finding an electron at \mathbf{r}_1 and another electron at \mathbf{r}_2 , regardless of the spin type. If we perform double-integration for pair density over the whole space, we will get $N(N-1)$, reflecting the nature that there are $N(N-1)$ electron pairs in present system. Obviously, pair density can be decomposed to contributions from different spin types of electron pairs

$$\pi(\mathbf{r}_1, \mathbf{r}_2) = \pi^{\alpha\alpha}(\mathbf{r}_1, \mathbf{r}_2) + \pi^{\alpha\beta}(\mathbf{r}_1, \mathbf{r}_2) + \pi^{\beta\alpha}(\mathbf{r}_1, \mathbf{r}_2) + \pi^{\beta\beta}(\mathbf{r}_1, \mathbf{r}_2)$$

If the electron motions are completely independent with each other, then the probability density of finding two electrons with spin σ_1 at \mathbf{r}_1 and with σ_2 at \mathbf{r}_2 respectively should simply be $\rho^{\sigma_1}(\mathbf{r}_1)\rho^{\sigma_2}(\mathbf{r}_2)$. Of course, in real world electrons always interacting with each other, so their motions are correlated. The pair density thereby should be corrected by *exchange-correlation density F*

$$\pi^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) = \rho^{\sigma_1}(\mathbf{r}_1)\rho^{\sigma_2}(\mathbf{r}_2) + \Gamma_{\text{xc}}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)$$

If we have already known that an electron with spin σ_1 presents at \mathbf{r}_1 , then the probability of finding another electron with spin σ_2 at \mathbf{r}_2 is known as *conditional probability* (This function is also known as Lennard-Jones function)

$$\Omega^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) = \frac{\pi^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)}{\rho^{\sigma_1}(\mathbf{r}_1)} = \rho^{\sigma_2}(\mathbf{r}_2) + \frac{\Gamma_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)}{\rho^{\sigma_1}(\mathbf{r}_1)}$$

Correlation hole reveals the decrease of probability of finding another electron with spin σ_2 at \mathbf{r}_2 when an electron with spin σ_1 presents at \mathbf{r}_1 owing to electron correlation effect

$$h_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) = \frac{\Gamma_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)}{\rho^{\sigma_1}(\mathbf{r}_1)} = \Omega^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) - \rho^{\sigma_2}(\mathbf{r}_2) = \frac{\pi^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)}{\rho^{\sigma_1}(\mathbf{r}_1)} - \rho^{\sigma_2}(\mathbf{r}_2)$$

Correlation factor is a function closely related to correlation hole

$$f_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) = \frac{h_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)}{\rho^{\sigma_2}(\mathbf{r}_2)} = \frac{\Gamma_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)}{\rho^{\sigma_1}(\mathbf{r}_1)\rho^{\sigma_2}(\mathbf{r}_2)}$$

Collectively, one can write out

$$\pi^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) = \rho^{\sigma_1}(\mathbf{r}_1)\rho^{\sigma_2}(\mathbf{r}_2) + \rho^{\sigma_1}(\mathbf{r}_1)h_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) = \rho^{\sigma_1}(\mathbf{r}_1)\rho^{\sigma_2}(\mathbf{r}_2)[1 + f_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)]$$

Γ_{XC} can be decomposed to the sum of exchange correlation (also called as Fermi correlation) part Γ_X and Coulomb correlation part Γ_C , therefore h_{XC} can be straightforwardly decomposed to *exchange hole* h_X (also called as Fermi hole) and *Coulomb hole* h_C as follows. Likewise, f_{XC} can be decomposed to f_X and f_C

$$\begin{aligned}\Gamma_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) &= \Gamma_X^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) + \Gamma_C^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) \\ h_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) &= h_X^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) + h_C^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) \\ f_{XC}^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) &= f_X^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2) + f_C^{\sigma_1\sigma_2}(\mathbf{r}_1, \mathbf{r}_2)\end{aligned}$$

Fermi correlation only presents between like-spin electrons; while Coulomb correlation occurs between any two electrons. Fermi correlation is much more important than Coulomb correlation, even at Hartree-Fock level, Fermi correlation is always well represented due to the anti-symmetry requirement of Slater determinant, while Coulomb is completely omitted. Only post-HF wavefunction is capable to simultaneously exhibit Fermi and Coulomb correlation effects. Commonly we only focus on Fermi hole while neglecting Coulomb hole. One can easily show that integration of h_X over whole space is exactly equal to -1, hence Fermi correlation perfectly avoided self-pairing problem, which may cause significant rise in system energy; while the integration for h_C is zero, this is mainly why Coulomb correlation has less influence on system energy.

It is also rather straightforward to obtain the pair density and conditional probability when only exchange correlation or Coulomb correlation is taken into account.

The total π , Ω , Γ_{XC} (or Γ_X , Γ_C) and h_{XC} (or h_X , h_C) for an electron with spin σ , regardless the spin of another electron can be defined as

$$\begin{aligned}\pi^{\sigma,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) &= \pi^{\sigma\alpha}(\mathbf{r}_1, \mathbf{r}_2) + \pi^{\sigma\beta}(\mathbf{r}_1, \mathbf{r}_2) \\ \Omega^{\sigma,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) &= \Omega^{\sigma\alpha}(\mathbf{r}_1, \mathbf{r}_2) + \Omega^{\sigma\beta}(\mathbf{r}_1, \mathbf{r}_2) \\ \Gamma_{XC}^{\sigma,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) &= \Gamma_{XC}^{\sigma\alpha}(\mathbf{r}_1, \mathbf{r}_2) + \Gamma_{XC}^{\sigma\beta}(\mathbf{r}_1, \mathbf{r}_2) \\ h_{XC}^{\sigma,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) &= h_{XC}^{\sigma\alpha}(\mathbf{r}_1, \mathbf{r}_2) + h_{XC}^{\sigma\beta}(\mathbf{r}_1, \mathbf{r}_2)\end{aligned}$$

===== Technical aspects =====

For single-determinant wavefunctions, exchange-correlation density for an α electron can be explicitly written as

$$\Gamma_{\text{XC}}^{\alpha,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) = \Gamma_{\text{XC}}^{\alpha\alpha}(\mathbf{r}_1, \mathbf{r}_2) = -\sum_{i \in \alpha}^{\text{occ}} \sum_{j \in \alpha}^{\text{occ}} \phi_i^*(\mathbf{r}_1) \phi_j^*(\mathbf{r}_2) \phi_j(\mathbf{r}_1) \phi_i(\mathbf{r}_2)$$

Note that $\Gamma_{\text{XC}}^{\alpha\beta}(\mathbf{r}_1, \mathbf{r}_2) = \Gamma_{\text{XC}}^{\beta\alpha}(\mathbf{r}_1, \mathbf{r}_2) = 0$ for this type of wavefunction. To obtain the expression for β electron, just replace α by β , similarly hereafterin.

For post-HF wavefunction, exact evaluation of pair density requires two-particle density matrix (2PDM). Unfortunately 2PDM is very difficult to be obtained, mainstream quantum chemistry packages including Gaussian can't directly output it. In Multiwfn, exchange-correlation density for post-HF wavefunction is approximately evaluated by natural orbital formalism. Note that the approximate method is not unique, see *J. Chem. Theory Comput.*, **6**, 2736 for discussion. The most popular form among them, which is firstly derived by Müller, is currently implemented in Multiwfn as below. See *Phys. Lett.*, **105A**, 446, also see *Mol. Phys.*, **100**, 401 for extensive discussion (especially equation 32).

$$\Gamma_{\text{XC,approx}}^{\alpha,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) = -\sum_{i \in \alpha} \sum_{j \in \alpha} \sqrt{\eta_i \eta_j} \phi_i^*(\mathbf{r}_1) \phi_j^*(\mathbf{r}_2) \phi_j(\mathbf{r}_1) \phi_i(\mathbf{r}_2)$$

Obviously, if occupation numbers of natural spin orbitals are integer (0 or 1), then $\Gamma_{\text{XC,approx}}^{\alpha,\text{tot}}$ reduces to single-determinant form. so $\Gamma_{\text{XC,approx}}^{\alpha,\text{tot}}$ can be regarded as a general form to evaluate exchange-correlation density. Note that post-HF wavefunction has taken Coulomb correlation between unlike-spin electrons into account, however there is no way to separate $\Gamma_{\text{XC,approx}}^{\alpha\alpha}$ and $\Gamma_{\text{XC,approx}}^{\alpha\beta}$ from $\Gamma_{\text{XC,approx}}^{\alpha,\text{tot}}$.

The exchange-only part of Γ for post-HF wavefunction can be approximately evaluated as below (of course, there is no exchange correlation between $\alpha\beta$ electron pair)

$$\Gamma_{\text{X,approx}}^{\alpha,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) = \Gamma_{\text{X,approx}}^{\alpha\alpha}(\mathbf{r}_1, \mathbf{r}_2) = -\sum_{i \in \alpha} \sum_{j \in \alpha} \eta_i \eta_j \phi_i^*(\mathbf{r}_1) \phi_j^*(\mathbf{r}_2) \phi_j(\mathbf{r}_1) \phi_i(\mathbf{r}_2)$$

So Coulomb-only part of Γ can be evaluated as (including both $\alpha\alpha$ and $\alpha\beta$ pair contributions)

$$\begin{aligned} \Gamma_{\text{C,approx}}^{\alpha,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) &= \Gamma_{\text{XC,approx}}^{\alpha,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2) - \Gamma_{\text{X,approx}}^{\alpha\alpha}(\mathbf{r}_1, \mathbf{r}_2) \\ &= \sum_{i \in \alpha} \sum_{j \in \alpha} [(\eta_i \eta_j - \sqrt{\eta_i \eta_j}) \phi_i^*(\mathbf{r}_1) \phi_j^*(\mathbf{r}_2) \phi_j(\mathbf{r}_1) \phi_i(\mathbf{r}_2)] \end{aligned}$$

Since we already have explicit expression to calculate Γ_{XC} term, other quantities introduced earlier can be easily computed according to the relationships between them and Γ_{XC} . Recall that $\rho^\sigma(\mathbf{r}) = \sum_{i \in \sigma} \eta_i |\phi_i(\mathbf{r})|^2$.

Postscript: One can show that $\Gamma_{\text{XC,approx}}^{\alpha,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2)$ also exactly holds the requirement that

integration of \mathbf{r}_2 over the whole space is equal to $-\rho^\alpha(\mathbf{r}_1)$. However, in common, integrating \mathbf{r}_2 over the whole space for $\Gamma_{X,\text{approx}}^{\alpha,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2)$ and $\Gamma_{C,\text{approx}}^{\alpha,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2)$ deviate from $-\rho^\alpha(\mathbf{r}_1)$ and zero, respectively, which are basic properties of exact form of $\Gamma_X^{\alpha,\text{tot}}$ and $\Gamma_C^{\alpha,\text{tot}}$.

===== Usage =====

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, just modifying “refxyz” in settings.ini before booting up.

“paircorrtype” parameter in settings.ini controls which type of correlation effect will be taken into consideration in calculation of Γ . Since correlation hole and correlation factor are calculated based on Γ , this setting also affects them. For single-determinant wavefunction, =1 and =3 are equivalent and =2 is meaningless, because Coulomb correlation is completely omitted.

- =1: Only consider exchange correlation
- =2: Only consider Coulomb correlation
- =3: Consider both exchange and Coulomb correlation

“pairfunctype” parameter in settings.ini controls which function and which spin will be calculated by real space function 17, see below, those enclosed by parentheses are for single-determinant wavefunction cases. Of course, for closed-shell system, the results for α spin are exactly identical to those for β spin. Note that correlation factor for post-HF wavefunction case is undefined.

- | | | | | | |
|----------------------------------|---------------------------|---------------------------------|-------------------------|----------------------|----------------------|
| =1: $h^{\alpha,\text{tot}}$ | $(h^{\alpha\alpha})$ | $=2: h^{\beta,\text{tot}}$ | $(h^{\beta\beta})$ | | |
| =4: Undefined | $(f^{\alpha\alpha})$ | =5: Undefined | $(f^{\beta\beta})$ | | |
| =7: $\Gamma^{\alpha,\text{tot}}$ | $(\Gamma^{\alpha\alpha})$ | =8: $\Gamma^{\beta,\text{tot}}$ | $(\Gamma^{\beta\beta})$ | | |
| =10: $\pi^{\alpha\alpha}$ | when paircorrtype =1 | $(\pi^{\alpha\alpha})$ | =11: $\pi^{\beta\beta}$ | when paircorrtype =1 | $(\pi^{\beta\beta})$ |
| =12: $\pi^{\text{any,any}}$ | | | | | |
| $(\pi^{\text{any,any}})$ | | | | | |

For example, if paircorrtype=1 and pairfunctype=2, for post-HF wavefunction, what will be calculated is $h_X^{\beta,\text{tot}}(\mathbf{r}_1, \mathbf{r}_2)$, which is equivalent to $h_X^{\beta\beta}(\mathbf{r}_1, \mathbf{r}_2)$ since $\beta\alpha$ electron pairs have no exchange correlation. This quantity can be interpreted as Fermi hole at \mathbf{r}_2 caused by a β electron present at \mathbf{r}_1 .

18 Average local ionization energy

Average local ionization energy is written as (*Can. J. Chem.*, **68**, 1440)

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

where $\rho_i(\mathbf{r})$ and ε_i are the electron density function and orbital energy of the i th molecular orbital, respectively. Hartree-Fock and typical DFT functionals such as B3LYP are both suitable for computing $\bar{I}(\mathbf{r})$. Lower value of $\bar{I}(\mathbf{r})$ indicates that the electrons at this point are more weakly bounded. $\bar{I}(\mathbf{r})$ has widespread uses, for example revealing atomic shell structures, measuring electronegativity, predicting pKa, quantifying local polarizability and hardness, but the most important one may be predicting reactive site of electrophilic or radical attack. It is proved that the minima of ALIE on vdW surface are good indicator to reveal which atoms are more likely to be the preferential site of electrophilic or radical attack.. There are also many potential uses of $\bar{I}(\mathbf{r})$ waiting for further investigation. Excellent reviews of $\bar{I}(\mathbf{r})$ have been given by Politzer *et al*, see *J. Mol. Model.*, **16**, 1731 and Chapter 8 of the book *Theoretical Aspects of Chemical Reactivity* (2007).

Since $\bar{I}(\mathbf{r})$ is dependent upon orbital energies, while orbital energy for post-HF wavefunction is undefined, therefore when post-HF wavefunction is used, $\bar{I}(\mathbf{r})$ will be simply outputted as zero everywhere.

If the parameter "iALIEdecomp" in settings.ini is set to 1, in main function 1, not only $\bar{I}(\mathbf{r})$ will be outputted, the contribution from each occupied MOs will also be outputted, the contribution due to MO i is defined as

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

19 Source function

Source function was proposed by Bader and Gatti, see *Chem. Phys. Lett.*, **287**, 233.

$$SF(\mathbf{r}, \mathbf{r}') = -\frac{\nabla^2 \rho(\mathbf{r}')}{4\pi \cdot |\mathbf{r} - \mathbf{r}'|}$$

It can be shown that

$$\rho(\mathbf{r}) = \int SF(\mathbf{r}, \mathbf{r}') d\mathbf{r}'$$

where \mathbf{r}' ranges entire space. This equation suggests that $SF(\mathbf{r}, \mathbf{r}')$ represents the effect of electronic Laplacian at \mathbf{r}' on electron density at \mathbf{r} . If at \mathbf{r}' the electron is concentrated (namely Laplacian is negative, also suggesting potential energy dominates kinetic energy), then \mathbf{r}' will be a source for the electron density at \mathbf{r} ; conversely, if at \mathbf{r}' the electron is depleted, then \mathbf{r}' diminish the electron density at \mathbf{r} . If the range of integration in above formula is restricted to a local region Ω and we get a value $S(\mathbf{r}, \Omega)$, then $S(\mathbf{r}, \Omega)/\rho(\mathbf{r}) \times 100\%$ can be regarded as the contribution from region

Ω to the electron density at \mathbf{r} . Source function has many uses, when it is used to discuss bonding problems, usually bond critical points are taken as \mathbf{r} . A very comprehensive review of theoretical background and applications of source function is given by Gatti in *Struct. & Bond.*, **147**, 193 (2010).

In Multiwfn, source function has two modes: (1) If "srcfuncmode" in settings.ini is set to 1, then \mathbf{r}' is regarded as variable, while \mathbf{r} is regarded as fixed reference point, whose coordinate is determined by "refxyz" in settings.ini. This is default mode, useful to study effect of electronic Laplacian at everywhere on specific point (2) If "srcfuncmode" is set to 2, then \mathbf{r} becomes variable and \mathbf{r}' becomes reference point, this is useful to study effect of electronic Laplacian at specific point on everywhere.

When $\mathbf{r}=\mathbf{r}'$, this function will return $-\nabla^2\rho(\mathbf{r}')/0.001$ to avoid numerical problem.

20.21 Electron delocalization range function EDR($\mathbf{r};d$) and orbital overlap distance function D(\mathbf{r})

Content of this section and all analysis code of EDR($\mathbf{r};d$) and D(\mathbf{r}) was kindly contributed by Arshad Mehmood and then slightly adapted by Tian Lu.

The electron delocalization range function EDR($\mathbf{r};d$) (*J. Chem. Phys.*, **141**, 144104 (2014); *J. Chem. Theory Comput.*, **12**, 3185 (2016); *Angew. Chem. Int. Ed.*, **56**, 6878 (2017)) quantifies the extent to which electrons at point \mathbf{r} in a wave function occupy orbital lobes of size d . EDR($\mathbf{r};d$) is built from the nonlocal one-particle reduced density matrix (1-RDM) $\gamma(\mathbf{r}, \mathbf{r}') = \sum_i \eta_i \varphi(\mathbf{r})\varphi(\mathbf{r}')$

as

$$\text{EDR}(\mathbf{r}; d) = \int g_d(\mathbf{r}, \mathbf{r}') \gamma(\mathbf{r}, \mathbf{r}') d\mathbf{r}'$$

$$g_d(\mathbf{r}, \mathbf{r}') = \left(\frac{2}{\pi d^2} \right)^{3/4} \rho^{-1/2}(\mathbf{r}) \exp\left(-\frac{|\mathbf{r}-\mathbf{r}'|^2}{d^2}\right)$$

Here $\rho(\mathbf{r})$ is the electron density at point \mathbf{r} . The prefactor ensures that the EDR is between -1 and +1. The Multiwfn implementation evaluates the EDR on grids, for a single global input value of distance d . Section 4.5.6 illustrates an example.

At each point, the orbital overlap distance function $D(\mathbf{r}) = \text{argmax}_d \text{EDR}(\mathbf{r}; d)$ corresponds to the distance d that maximizes EDR($\mathbf{r}; d$). Compact, chemically "hard" regions of small $D(\mathbf{r})$ are distinguished from diffuse, chemically "soft" regions of large $D(\mathbf{r})$. Atomic averages of valence-electron $D(\mathbf{r})$ complement the information obtained from atomic partial charges. Plots of $D(\mathbf{r})$ on density isosurfaces, and quantitative analysis of such surfaces, complements molecular electrostatic potentials. The Multiwfn implementation evaluates EDR($\mathbf{r}; d_i$) on a grid of distances d_i , then uses a three-point numerical fit to find the maximum. Sections 4.5.7 and 4.12.8 illustrate example calculations.

22 δg defined in Independent Gradient Model (IGM) method

$$\delta g = |\nabla \rho^{\text{IGM}}| - |\nabla \rho|$$

See Section 3.23.5 and original paper of Independent Gradient Model (IGM) *Phys. Chem. Chem. Phys.*, **19**, 17928 (2017) for introduction. This function is calculated based on built-in promolecular density, and its value at bond critical point in weak interaction region is shown to be closely related

to interaction strength. This function can also be plotted as plane map or isosurface map to reveal all kinds of bonding regions.

2.7 User defined real space function

In real space function selection menu you can find a term named "User defined real space function", this function corresponds to "userfunc" routine in source file function.f90. By filling codes by yourself, the functions of Multiwfn can be easily extended. For examples, after filling the code `userfunc=fgrad(x,y,z,'t')**2/8/fdens(x,y,z)` and recompile Multiwfn, you will get the integrand

of Weizsäcker kinetic energy functional $\tau_w[\rho] = \int \frac{|\nabla \rho(\mathbf{r})|^2}{8\rho(\mathbf{r})} d\mathbf{r}$. When you write code you can

consult Appendix 2 of this manual as well as the codes of other functions. If you would like 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 file, by which you can select which custom function to be used.

In addition, in order to avoid lengthy list of real space functions, numerous uncommonly used real space functions are not explicitly presented in the list. However, if you want to use them, you can set "iuserfunc" parameter in settings.ini to one of below values, then the user defined function will point to corresponding function. For example, before running Multiwfn, if you set "iuserfunc" to 2, then the user defined real space function will be equivalent to density of Beta electrons.

-2 Promolecular density calculated based on built-in sphericalized atomic densities, you can check Appendix 3 on how these atomic densities are produced. Options -3 and -4 of main function 6 could be used to exclude contribution of some atoms to promolecular density.

Please note that this promolecular density is slightly different to the promolecular density used in real space functions 14 and 16, because in which the atomic densities from H to Ar are directly taken from the original paper of NCI method rather than calculated according to the method described in Appendix 3.

-1 The function value evaluated by trilinear interpolation from grid data. The grid data can be generated by main function 5, or loaded from .cub/.grd file when Multiwfn boot up. This function is quite useful. For example, by making use of this function you can plot the grid data as curve map and plane map via main functions 3 and 4, respectively; you can also gain atomic contribution to the total value via subfunction 1 of fuzzy atomic space analysis module (main function 15)

0 The function corresponds to a constant value of 1.0

1 Alpha density: $\rho^\alpha(\mathbf{r}) = [\rho(\mathbf{r}) + \rho^s(\mathbf{r})]/2$

2 Beta density: $\rho^\beta(\mathbf{r}) = [\rho(\mathbf{r}) - \rho^s(\mathbf{r})]/2$

3 Integrand of electronic spatial extent $\langle R^{**2} \rangle$: $\rho(\mathbf{r}) \times (x^2 + y^2 + z^2)$

4 Weizsäcker potential (closed-shell form): $V_w(\mathbf{r}) = \frac{1}{8} \frac{|\nabla \rho(\mathbf{r})|^2}{\rho^2(\mathbf{r})} - \frac{1}{4} \frac{\nabla^2 \rho(\mathbf{r})}{\rho(\mathbf{r})}$

5 Integrand of Weizsäcker functional (closed-shell form): $\tau_w(\mathbf{r}) = |\nabla \rho(\mathbf{r})|^2 / [8\rho(\mathbf{r})]$, which is the exact kinetic energy density of any one-orbital system (one or two electrons, or any number of bosons). If you need spin polarized form, use user-defined function 1200.

6 Radial distribution function of electron density: $4\pi \times \rho(\mathbf{r}) \times (x^2 + y^2 + z^2)$. Clearly, spherical symmetry of electron density is assumed.

7 Local temperature, in k_B^{-1} (*PNAS*, **81**, 8028): $T(\mathbf{r}) = \frac{2}{3} \frac{G(\mathbf{r})}{\rho(\mathbf{r})}$

8 Average local electrostatic potential (*J. Chem. Phys.*, **72**, 3027): $V_{ESP}(\mathbf{r}) / \rho(\mathbf{r})$

9 Shape function: $\rho(\mathbf{r}) / N$, where N is the total number of electrons

10 Potential energy density (Virial field): $V(\mathbf{r}) = -K(\mathbf{r}) - G(\mathbf{r}) = (1/4)\nabla^2 \rho(\mathbf{r}) - 2G(\mathbf{r})$

11 Electron energy density: $E(\mathbf{r}) = G(\mathbf{r}) + V(\mathbf{r}) = -K(\mathbf{r})$ (Sometimes E is written as H)

12 Local nuclear attraction potential energy: $-\rho(\mathbf{r}) \times V_{nuc}(\mathbf{r})$

13 Kinetic energy density per electron: $G(\mathbf{r}) / \rho(\mathbf{r})$ This quantity at bond critical point is useful to discriminate covalent bonding and closed-shell interaction

14 Electrostatic potential from electrons: $V_{ele}(\mathbf{r}) = V_{ESP}(\mathbf{r}) - V_{nuc}(\mathbf{r}) = -\int \rho(\mathbf{r}') / |\mathbf{r} - \mathbf{r}'| d\mathbf{r}'$

15 Bond metallicity: $\xi_J(\mathbf{r}) = \rho(\mathbf{r}) / \nabla^2 \rho(\mathbf{r})$ At bond critical point, $\xi_J > 1$ indicates metallic interaction, see *J. Phys.: Condens. Matter*, **14**, 10251.

16 Dimensionless bond metallicity: $\xi_m(\mathbf{r}) = \frac{36(3\pi^2)^{2/3}}{5} \frac{\rho(\mathbf{r})^{5/3}}{\nabla^2 \rho(\mathbf{r})}$ At bond critical point, larger value corresponds to stronger metallicity of the bond, see *Chem. Phys. Lett.*, **471**, 174.

17 Energy density per electron: $E(\mathbf{r}) / \rho(\mathbf{r})$, this value at BCP is called as bond degree parameter (BD), see *J. Chem. Phys.*, **117**, 5529 (2002) for detail, in which the authors advocate that for covalent bonds (*viz.* $E_{BCP} < 0$), the BD renders covalence degree, and the stronger the interaction the greater the BD magnitude; while for closed-shell interactions (*viz.* $E_{BCP} > 0$), the BD can be viewed as softening degree, and the weaker the interaction the larger the BD magnitude.

18 Region of Slow Electrons (RoSE), which is defined in *Chem. Phys. Lett.*, **582**, 144 (2013) and has a pattern very similar to ELF, with value space of [-1,1]: $\nu_{\pm} = \frac{D_0(\mathbf{r}) - G(\mathbf{r})}{D_0(\mathbf{r}) + G(\mathbf{r})}$

19 Single exponential decay detector (SEDD), which is highly analogous to ELF. Its updated definition in *J. Chem. Theory Comput.*, **10**, 3745 (2014) is implemented:

$$\text{SEDD}(\mathbf{r}) = \ln \left\{ 1 + \left[\frac{\nabla(\nabla\rho(\mathbf{r})/\rho(\mathbf{r}))^2}{\rho(\mathbf{r})} \right]^2 \right\}$$

20 Density overlap regions indicator (DORI), defined in *J. Chem. Theory Comput.*, **10**, 3745 (2014):

$\text{DORI}(\mathbf{r}) = \theta(\mathbf{r})/[1 + \theta(\mathbf{r})]$, where $\theta(\mathbf{r}) = [\nabla(\nabla\rho(\mathbf{r})/\rho(\mathbf{r}))^2]^2 / [\nabla\rho(\mathbf{r})/\rho(\mathbf{r})]^6$. DORI is mainly used to reveal interatomic interaction regions, see Section 3.23.3 for more information

21 Integrand of X component of electric dipole moment: $-x \times \rho(\mathbf{r})$

22 Integrand of Y component of electric dipole moment: $-y \times \rho(\mathbf{r})$

23 Integrand of Z component of electric dipole moment: $-z \times \rho(\mathbf{r})$

24 Approximate form of DFT linear response kernel for closed-shell (*Phys. Chem. Chem. Phys.*, **14**,

$$3960): \chi(\mathbf{r}_1, \mathbf{r}_2) \approx 4 \sum_{i \in \text{occ}} \sum_{j \in \text{vir}} \frac{\varphi_i^*(\mathbf{r}_1) \varphi_j(\mathbf{r}_1) \varphi_j^*(\mathbf{r}_2) \varphi_i(\mathbf{r}_2)}{\varepsilon_i - \varepsilon_j}$$

25 Magnitude of fluctuation of the electronic momentum: $\tilde{P}(\mathbf{r}) = \frac{1}{2} \frac{|\nabla\rho(\mathbf{r})|}{\rho(\mathbf{r})}$, which is useful to

discuss bonding and very similar to reduced density gradient, see *Theor. Chim. Acc.*, **127**, 393

26 Integrand of Thomas-Fermi kinetic energy functional (closed-shell form):

$\tau_{\text{TF}}(\mathbf{r}) = C_{\text{TF}} \rho(\mathbf{r})^{5/3}$, where $C_{\text{TF}} = (3/10)(3\pi^2)^{2/3} = 2.871234$. This is the exact kinetic energy density of noninteracting, uniform electron gas. If you need spin polarized form, use user-defined function 1200.

27 Local electron affinity: $EA_L(\mathbf{r}) = \frac{-\sum_{i \in \text{vir}} |\varphi_i(\mathbf{r})|^2 \varepsilon_i}{\sum_{i \in \text{vir}} |\varphi_i(\mathbf{r})|^2}$, which is very similar to average local

ionization energy, but i cycles all unoccupied orbitals. See *J. Mol. Model.*, **9**, 342

28 Local Mulliken electronegativity: $\chi_L(\mathbf{r}) = [\bar{I}(\mathbf{r}) + EA_L(\mathbf{r})]/2$, see *J. Mol. Model.*, **9**, 342

29 Local hardness: $\eta_L(\mathbf{r}) = [\bar{I}(\mathbf{r}) - EA_L(\mathbf{r})]/2$, see *J. Mol. Model.*, **9**, 342

Note: In order to use EA_L , χ_L and η_L , the wavefunction must contain both occupied and unoccupied orbitals, therefore such as .fch, .molden and .gms rather than .wfn/.wfx file must be used. Meanwhile, "idelvirorb" in settings.ini must be set to 0. Besides, these quantities are meaningful only when minimal basis set is used.

30 Ellipticity of electron density: $\varepsilon(\mathbf{r}) = [\lambda_1(\mathbf{r})/\lambda_2(\mathbf{r})] - 1$, where λ_1 and λ_2 are the lowest and the second lowest eigenvalues of Hessian matrix of ρ , respectively. At bond critical point, λ_1 and λ_2 are both negative and exhibit the curvature of electron density perpendicular to the bond.

31 eta index: $\eta(\mathbf{r}) = |\lambda_1(\mathbf{r})|/\lambda_3(\mathbf{r})$, where λ_1 and λ_3 are the lowest and the highest eigenvalues

of Hessian matrix of ρ , respectively. It was argued that the value of η at bond critical point is less than unity for closed shell interactions and increases with increasing covalent character, see *Angew. Chem. Int. Ed.*, **53**, 2766 (2014), as well as *J. Phys. Chem. A*, **114**, 552 (2010) for discussions.

32 Modified eta index by Tian Lu: $\eta'(\mathbf{r}) = |\lambda_1(\mathbf{r})| / \lambda_3(\mathbf{r}) - 1$. Similar to η , but negative value of η' corresponds to closed shell interactions.

33 Potential acting on one electron in a molecule (PAEM), see *J. Comput. Chem.*, **35**, 965 (2014):

$$V_{\text{PAEM}}(\mathbf{r}) = -\sum_A \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} + \frac{1}{\rho(\mathbf{r})} \int \frac{\pi(\mathbf{r}, \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}', \quad \pi \text{ is pair density. It can be further written as}$$

$$V_{\text{PAEM}}(\mathbf{r}) = -V_{\text{ESP}}(\mathbf{r}) + V_{\text{XC}}(\mathbf{r}) = -V_{\text{ESP}}(\mathbf{r}) + \frac{1}{\rho(\mathbf{r})} \int \frac{\Gamma_{\text{XC}}^{\alpha, \text{tot}}(\mathbf{r}, \mathbf{r}') + \Gamma_{\text{XC}}^{\beta, \text{tot}}(\mathbf{r}, \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}', \quad \text{where}$$

V_{ESP} is the total molecular electrostatic potential and V_{XC} is exchange-correction potential. In Multiwfn, the exchange-correlation density Γ is evaluated in terms of Müller approximation, see part 17 of Section 2.6 for detail. It was shown in the original paper that PAEM may be useful to distinguish covalent and noncovalent interactions. An example of application of PAEM can be found in Section 4.3.3.

34 The same as **33**, but now V_{XC} directly corresponds to DFT exchange-correlation potential. Its specific form can be chosen via "iDFTxcsel" parameter, see the end of this section for detail. This form of PAEM is several times more computationally economical than **33** but only supports closed-shell wavefunction.

35 $|V(\mathbf{r})| / G(\mathbf{r})$. In *J. Chem. Phys.*, **117**, 5529 (2002) it was proposed that this quantity at BCP can be used to discriminate interaction types. <1 corresponds to closed-shell interaction; >2 corresponds to covalent interaction; while >1 and <2 corresponds to intermediate interaction.

36 On-top pair density, namely the two positions of the pair density are identical: $\pi(\mathbf{r}, \mathbf{r})$. See such as *Int. J. Quantum Chem.*, **61**, 197 (1995) for discussion. "paircorrtype" parameter in settings.ini mentioned earlier affects the result.

37 The strong covalent interaction index (SCI) defined in *J. Phys. Chem. A*, **122**, 3087 (2018) and further examined in *J. Mol. Model.*, **24**, 213 (2018). This function was shown to be very useful for identifying very strong covalent bonds. SCI is expressed as $\text{SCI}(\mathbf{r}) = 1/\zeta(\mathbf{r})$, where

$\zeta(\mathbf{r}) = [\tau_s(\mathbf{r}) - \tau_w(\mathbf{r})] / \tau_{\text{TF}}$. The meaning of τ_s , τ_w and τ_{TF} are described in Section 2.6 where ELF is introduced. It is easy to find SCI index is closely related to ELF, which can be written as $\text{ELF}(\mathbf{r}) = 1/[1 + \zeta^2(\mathbf{r})]$.

38 The angle between the second eigenvector of Hessian of electron density and the vector perpendicular to a given plane, which can be defined by option 4 of main function 1000; the unit vector normal to the plane will be shown on screen, assume that you use three points A, B, C to define the plane and you get vector \mathbf{u} , but what you really want is $-\mathbf{u}$, you can then input the points again but in reverse sequence, *i.e.* C, B, A. In *J. Phys. Chem. A*, **115**, 12512 (2011) this quantity along bond paths was used to reveal π interaction.

39 Electrostatic potential without contribution of a specific nucleus:

$$V_n(\mathbf{r}) = \sum_{A \neq K} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} - \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}', \text{ where the nucleus } K \text{ can be set by option 3 of main function}$$

1000 (which is hidden in main interface but can be chosen). V_n is a useful quantity, for example if K is chosen as index of a hydrogen, then the value correlates with its pK_a , because in this case V_n approximately reflects the binding energy of a proton at the position of K and rest of the system; In addition, *J. Phys. Chem. A*, **118**, 1697 (2014) showed that V_n can be used to quantitatively predict interaction energy of the weak interactions dominated by electrostatic effect (*viz.* H-bonds, halogen-bonds, dihydrogen bonds), see Section 4.1.2 for introduction and example.

It is worth to mention that in main function 1, when you request Multiwfn to print properties at nuclear position of an atom, the electrostatic potential without contribution of nuclear charge of this atom is automatically printed.

40 Steric energy density: $|\nabla\rho(\mathbf{r})|^2 / [8\rho(\mathbf{r})]$, which is equivalent to integrand of Weizsäcker functional.

41 Steric potential: $\nu_s(\mathbf{r}) = \frac{1}{8} \frac{|\nabla\rho(\mathbf{r})|^2}{[\rho(\mathbf{r}) + \delta]^2} - \frac{1}{4} \frac{\nabla^2\rho(\mathbf{r})}{\rho(\mathbf{r}) + \delta}$. The negative of this quantity is also

known as one-electron potential (OEP). Notice that the δ is a very small term artificially introduced to avoid the denominator converges to zero faster than nominator. The value of δ can be determined by "steric_addminimal" in settings.ini. If δ is set to 0, then the original expression of steric potential is recovered.

42 Steric charge: $q_s(\mathbf{r}) = \nabla^2\nu_s(\mathbf{r}) / (-4\pi)$.

43 The magnitude of steric force: $F_s(\mathbf{r}) = |-\nabla\nu_s(\mathbf{r})|$.

Notice that the δ term, which is mentioned above, also affects steric force and steric charge. Discussions of steric energy/potential/force/charge can be found in *J. Chem. Phys.*, **126**, 244103.

44,45,46 Damped Steric potential, steric force based on damped steric potential, directly damped Steric force: Documented privately

50 Shannon entropy density: $s_S(\mathbf{r}) = -\rho(\mathbf{r}) \ln \rho(\mathbf{r})$.

51 Fisher information density: $i_F(\mathbf{r}) = |\nabla\rho(\mathbf{r})|^2 / \rho(\mathbf{r})$. Note that this quantity is only different from Weizsäcker functional by a constant factor of 1/8.

52 Second Fisher information density: $i'_F(\mathbf{r}) = -\nabla^2\rho(\mathbf{r}) \ln \rho(\mathbf{r})$. The relationship between Shannon entropy and Fisher information can be found in *J. Chem. Phys.*, **126**, 191107. The integration of this function over the whole space is exactly identical to that of Fisher information density.

53 Ghosh entropy density (or Ghosh-Berkowitz-Parr entropy), in k_B (*PNAS*, **81**, 8028): $s(\mathbf{r}) = (3/2)\rho(\mathbf{r})\{\lambda + \ln[t(\mathbf{r})/t_{TF}(\mathbf{r})]\}$, where $\lambda = 5/3 + \ln(4\pi c_k / 3)$, c_k is Thomas-Fermi

constant $(3/10)(3\pi^2)^{2/3} = 2.871234$, Lagrangian kinetic energy density $G(\mathbf{r})$ is chosen to be

the kinetic energy density term $t(\mathbf{r})$.

54 The same as 53, but $G(\mathbf{r}) - \nabla^2 \rho(\mathbf{r})/8$, which is the kinetic energy density exactly corresponding to Eq. 22 of *PNAS*, **81**, 8028, is employed as the $t(\mathbf{r})$. In rare cases this definition of kinetic energy density leads to a very small negative value, since at these points $t_{\text{TF}}(\mathbf{r})$ is also very close to zero, in order to normally get result, $\ln[t(\mathbf{r})/t_{\text{TF}}(\mathbf{r})]$ in this case is simply set to zero.

55 Integrand of quadratic form of Rényi entropy: $\rho^2(\mathbf{r})$

56 Integrand of cubic form of Rényi entropy: $\rho^3(\mathbf{r})$

60 Pauli potential: $V_\theta = V_{\text{ESP}} - V_{\text{XC}} - V_{\text{W}}$, where V_{ESP} is total electrostatic potential (as shown in Section 2.6), V_{XC} is exchange-correlation potential (its specific form is determined by "iDFTxcsel" in settings.ini, see below), V_{W} is Weizsäcker potential. This quantity corresponds to Eq. 17 of *Comput. Theor. Chem.*, **1006**, 92-99 and is only applicable to closed-shell cases.

61 The magnitude of Pauli force: $F_\theta(\mathbf{r}) = |-\nabla V_\theta(\mathbf{r})|$.

62 Pauli charge: $q_\theta(\mathbf{r}) = \nabla^2 V_\theta(\mathbf{r}) / (-4\pi)$.

63 Quantum potential: $V_q = V_\theta + V_{\text{XC}}$

64 The magnitude of quantum force: $F_q(\mathbf{r}) = |-\nabla V_q(\mathbf{r})|$

65 Quantum charge: $q_q(\mathbf{r}) = q_\theta(\mathbf{r}) + q_{\text{XC}}(\mathbf{r})$

66 The magnitude of electrostatic force: $F_{\text{ESP}}(\mathbf{r}) = |-\nabla V_{\text{ESP}}(\mathbf{r})|$

67 Electrostatic charge: $q_{\text{ESP}}(\mathbf{r}) = \nabla^2 V_{\text{ESP}}(\mathbf{r}) / (-4\pi)$

70 Phase-space-defined Fisher information density (PS-FID): $i_f(\mathbf{r}) = \frac{3\rho(\mathbf{r})}{k_B T(\mathbf{r})} = \frac{9}{2} \frac{\rho^2(\mathbf{r})}{G(\mathbf{r})}$,

where $T(\mathbf{r})$ is local temperature as shown above. This function has very similar characters to ELF and LOL, the spatial localization of electron pairs can be clearly revealed. See *Chem. Phys.*, **435**, 49 (2014) for introduction and illustrative applications.

71, 72, 73, 74 Electron linear momentum density (EMD) in 3D representation. The electron linear momentum operator is $-i\nabla$, for an orbital the expectation value of linear momentum in X is $-i \langle \varphi | \nabla | \varphi \rangle$, so X component of EMD for a wavefunction can be defined as (the imaginary sign is ignored) $p_x(\mathbf{r}) = -\sum_i \eta_i \varphi_i^*(\mathbf{r}) \frac{\partial \varphi_i(\mathbf{r})}{\partial x}$, similar for Y and Z components. The 71, 72, 73th

user-defined functions correspond to X, Y, Z component, respectively. The magnitude (74) of EMD is defined as $p_{\text{tot}}(\mathbf{r}) = \sqrt{p_x^2(\mathbf{r}) + p_y^2(\mathbf{r}) + p_z^2(\mathbf{r})}$.

75, 76, 77, 78 Magnetic dipole moment density (MDMD). The operator for magnetic dipole moment is the angular momentum operator (see *Theoret. Chim. Acta*, **6**, 341)

$$-i(\mathbf{r} \times \nabla) = -i \left[\hat{\mathbf{i}} \left(y \frac{\partial}{\partial z} - z \frac{\partial}{\partial y} \right) + \hat{\mathbf{j}} \left(z \frac{\partial}{\partial x} - x \frac{\partial}{\partial z} \right) + \hat{\mathbf{k}} \left(x \frac{\partial}{\partial y} - y \frac{\partial}{\partial x} \right) \right]$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are unity vectors in X, Y and Z directions, respectively. Therefore, X, Y and Z component of MDMD may be defined as follows (the imaginary sign is ignored)

$$m_x(\mathbf{r}) = -\sum_i \eta_i \varphi_i^*(\mathbf{r}) \left[y \frac{\partial \varphi_i(\mathbf{r})}{\partial z} - z \frac{\partial \varphi_i(\mathbf{r})}{\partial y} \right]$$

$$m_y(\mathbf{r}) = -\sum_i \eta_i \varphi_i^*(\mathbf{r}) \left[z \frac{\partial \varphi_i(\mathbf{r})}{\partial x} - x \frac{\partial \varphi_i(\mathbf{r})}{\partial z} \right]$$

$$m_z(\mathbf{r}) = -\sum_i \eta_i \varphi_i^*(\mathbf{r}) \left[x \frac{\partial \varphi_i(\mathbf{r})}{\partial y} - y \frac{\partial \varphi_i(\mathbf{r})}{\partial x} \right]$$

The 75, 76, 77th user-defined functions correspond to X, Y, Z component, respectively. The magnitude (78) of MDMD is $m_{\text{tot}}(\mathbf{r}) = \sqrt{m_x^2(\mathbf{r}) + m_y^2(\mathbf{r}) + m_z^2(\mathbf{r})}$.

79 Gradient norm of electron energy density: $|\nabla E(\mathbf{r})|$

80 Laplacian of electron energy density: $\nabla^2 E(\mathbf{r})$

81, 82, 83 X, Y, Z component of Hamiltonian kinetic energy density, respectively

84, 85, 86 X, Y, Z component of Lagrangian kinetic energy density, respectively

87 Local total electron correlation function: $I_T(\mathbf{r}) = \frac{1}{4} \sum_i [\eta_i(1-\eta_i)]^{1/2} |\varphi_i(\mathbf{r})|^2$. The i denotes

index of natural spin orbital, the η is corresponding occupation number. This and below two functions are very useful for revealing electron correlation in different regions, see *J. Chem. Theory Comput.*, **13**, 2705 (2017) for introduction, illustrative example can also be found in Section 4.A.6. Note that in some cases, η may be marginally larger than 1.0 or negative, Multiwfn automatically set it to 1.0 and 0.0 respectively to make the calculation feasible.

88 Local dynamic electron correlation function:

$$I_D(\mathbf{r}) = \frac{1}{4} \sum_i \{[\eta_i(1-\eta_i)]^{1/2} - 2\eta_i(1-\eta_i)\} |\varphi_i(\mathbf{r})|^2$$

$$\mathbf{89} \text{ Local nondynamic electron correlation function: } I_{ND}(\mathbf{r}) = \frac{1}{2} \sum_i \eta_i(1-\eta_i) |\varphi_i(\mathbf{r})|^2$$

It is clear that $I_T(\mathbf{r}) = I_D(\mathbf{r}) + I_{ND}(\mathbf{r})$

100 Disequilibrium (also known as semi-similarity): $D_r(\mathbf{r}) = \rho^2(\mathbf{r})$. See *Int. J. Quantum. Chem.*,

113, 2589 (2013) for illustrative applications.

101 Positive part of ESP: V_{ESP}^+ . In the region where V_{ESP} is positive, $V_{\text{ESP}}^+ = V_{\text{ESP}}$; where ESP is negative, $V_{\text{ESP}}^+ = 0$.

102 Negative part of ESP: V_{ESP}^- .

103 Magnitude of electric field $|\mathbf{F}|$. Since electric field vector is simply negative gradient vector of ESP, therefore this quantity corresponds to norm of gradient of ESP.

900, 901, 902 X, Y and Z coordinate variables, respectively: x, y, z .

1000 Integrand of DFT exchange-correlation functionals, which is also known as exchange-correlation energy density.

1100 DFT exchange-correlation potential, *i.e.* $V_{\text{XC}}(\mathbf{r}) = \frac{\delta E_{\text{XC}}[\rho(\mathbf{r})]}{\delta \rho(\mathbf{r})}$

For user-defined function 1000 and 1100, "iDFTxcsel" in settings.ini is used to select XC functional. 0~29 are X part only, 30~69 are C part only, 70~99 are whole XC. For example, if iDFTxcsel=32, then iuserfunc=1000 corresponds to integrand of LYP, and iuserfunc=1100 corresponds to LYP potential.

Notice that both 1000 and 1100 are applicable for closed-shell cases. But currently only 1000 can be applied to open-shell cases.

[See below for details]

Available options of "iDFTxcsel" parameter

Currently only LSDA and some GGA functionals are supported

0 LSDA exchange: $-(3/2)[3/(4\pi)]^{1/3}[\rho_\alpha(\mathbf{r})^{4/3} + \rho_\beta(\mathbf{r})^{4/3}]$. For closed-shell cases the equivalent form is $-(3/4)(3/\pi)^{1/3}\rho(\mathbf{r})^{4/3}$.

1 Becke 88 (B88) exchange, see *Phys. Rev. A*, **38**, 3098-3100 (1988).

2 Perdew-Burke-Ernzerhof (PBE) exchange, see *Phys. Rev. Lett.*, **77**, 3865-3868 (1996).

3 Perdew-Wang 91 (PW91) exchange, see Electronic Structure of Solids '91; Ziesche, P., Eschig, H., Eds.; Akademie Verlag: Berlin, 1991; p. 11.

30 Vosko-Wilk-Nusair V (VWN5) correlation, see *Can. J. Phys.*, **58**, 1200-1211 (1980).

31 Perdew 86 (P86) correlation, see *Phys. Rev. B*, **33**, 8822-8824 (1986).

32 Lee-Yang-Parr (LYP) correlation, see *Phys. Rev. B*, **37**, 785-789 (1988).

33 Perdew-Wang 91 (PW91) correlation, see the reference of its exchange counterpart.

34 Perdew-Burke-Ernzerhof (PBE) correlation, see the reference of its exchange counterpart.

70 Becke 97 (B97) exchange-correlation, see *J. Chem. Phys.*, **107**, 8554-8560 (1997).

71 Hamprecht-Cohen-Tozer-Handy with 407 training molecules (HCTH407) exchange-correlation, see *J. Chem. Phys.*, **114**, 5497-5503 (2001).

80 SVWN5 exchange-correlation.

81 BP86 exchange-correlation.

82 BLYP exchange-correlation.

83 BPW91 exchange-correlation.

84 PBEPBE exchange-correlation.

85 PW91PW91 exchange-correlation.

About iuserfunc=1200

User-defined function 1200 is a collection of all kinds of kinetic energy density (KED) supported by Multiwfn. The so-called KED denotes the integrand of kinetic energy functional. The "iKEDsel" in settings.ini is used to choose the form of KED.

Note that if "iKEDsel" is not at its default value (0), then the τ_S term (Lagrangian kinetic energy density) of ELF and strong covalent interaction index (SCI) will be replaced with corresponding KED.

There is an subfunction 92 in the main function 1000 (hidden in the main menu), it can calculate integral of all KEDs over the whole space, namely evaluating kinetic energy based on the kinetic energy functionals. The GEA4 KED is calculated only when "iKEDsel" is set to 24 since it needs Laplacian of electron density, while other KEDs only need electron density and its gradient. Integral of Hamiltonian KED is not calculated by this function since the result is identical to that of Lagrangian KED.

- iKEDsel=1: Hamiltonian KED, identical to real space function 6

- iKEDsel=2: Lagrangian KED, identical to real space function 7
- iKEDsel=3: Thomas-Fermi KED: $\tau_{\text{TF}}(\mathbf{r}) = \sum_{\sigma=\alpha,\beta} \tau_{\text{TF}}^{\sigma} = \sum_{\sigma=\alpha,\beta} C_{\text{TF}} [\rho_{\sigma}(\mathbf{r})]^{5/3}$,

where $C_{\text{TF}} = \frac{3}{10} (6\pi^2)^{2/3} = 4.557799872$ is Thomas-Fermi constant in spin polarized case.

- iKEDsel=4: Weizsäcker KED: $\tau_{\text{W}}(\mathbf{r}) = \sum_{\sigma=\alpha,\beta} \frac{|\nabla \rho_{\sigma}(\mathbf{r})|^2}{8\rho_{\sigma}(\mathbf{r})}$

Most of below mentioned KEDs can be represented using a general form $\tau(\mathbf{r}) = \sum_{\sigma=\alpha,\beta} \tau_{\text{TF}}^{\sigma}(\mathbf{r}) F^{\sigma} [s^{\sigma}(\mathbf{r})]$, where $s^{\sigma}(\mathbf{r}) = |\nabla \rho^{\sigma}(\mathbf{r})| / \rho^{\sigma}(\mathbf{r})^{4/3}$, F is called as enhancement factor, see *J. Chem. Phys.*, **127**, 144109 (2007) for detail. This paper systematically introduced and compared a variety of existing KEDs. Citation of below KEDs are not given, since they can be found in the *J. Chem. Phys.* paper. Note that a lot of expressions given in the *J. Chem. Phys.* paper are wrong, while the formulae given below are absolutely correct, and all of them are explicitly written as spin polarized form. More information and comparison about KEDs can be found in *Phys. Rev. A*, **46**, 6920 (1992) and *J. Chem. Phys.*, **100**, 4446 (1994).

- iKEDsel=5: Second order gradient expansion approximation, $F_{\text{GEA}_2}^{\sigma} = \left[1 + \frac{1}{72C_{\text{TF}}} s^{\sigma}(\mathbf{r})^2 \right]$
- iKEDsel=6: Thomas-Fermi + 1/5 Weizsäcker KED, $F_{\text{TF5W}}^{\sigma} = \left[1 + \frac{1}{40C_{\text{TF}}} s^{\sigma}(\mathbf{r})^2 \right]$
- iKEDsel=7: Thomas-Fermi + Weizsäcker KED, $F_{\text{TFvW}}^{\sigma} = \left[1 + \frac{1}{8C_{\text{TF}}} s^{\sigma}(\mathbf{r})^2 \right]$
- iKEDsel=8: Thomas-Fermi + $b/9$ Weizsäcker KED, $F_{\text{TF9W}}^{\sigma} = \left[1 + \frac{1.067}{72C_{\text{TF}}} s^{\sigma}(\mathbf{r})^2 \right]$
- iKEDsel=9: N -dependent Thomas-Fermi KED, $F_{\text{TF-N}}^{\sigma} = 1 + \frac{0.313}{N^{1/3}} - \frac{0.187}{N^{2/3}}$, where N is the number of total electrons in the system

- iKEDsel=10: Pearson KED, $\tau_{\text{Pear}}^{\sigma} = \tau_{\text{TF}}^{\sigma} + \frac{1}{1 + [s_r^{\sigma}(\mathbf{r})/\zeta]^6} \frac{1}{72} \frac{|\nabla \rho^{\sigma}(\mathbf{r})|^2}{\rho^{\sigma}(\mathbf{r})}$, where ζ is fixed to 1, $s_r^{\sigma}(\mathbf{r}) = s^{\sigma}(\mathbf{r})/[2(6\pi^2)^{1/3}]$, similarly hereinafter

- iKEDsel=11: DePristo-Kress Pade KED, $F_{\text{DK Pade}}^{\sigma} = \frac{9b_3x^4 + a_3x^3 + a_2x^2 + a_1x + 1}{b_3x^3 + b_2x^2 + b_1x + 1}$, where $x = [s^{\sigma}(\mathbf{r})]^2 / (72C_{\text{TF}})$, $a_1=0.95$, $a_2=14.28111$, $a_3=-19.57962$, $b_1=-0.05$, $b_2=9.99802$ and $b_3=2.96085$

- iKEDsel=12: Lee-Lee-Parr KED, $F_{\text{LLP}}^\sigma = 1 + \frac{0.0044188[s^\sigma(\mathbf{r})]^2}{1 + 0.0253s^\sigma(\mathbf{r})\operatorname{arcsinh}[s^\sigma(\mathbf{r})]}$
- iKEDsel=13: Ou-Yang-Levy 1 KED, $F_{\text{OL1}}^\sigma = 1 + \frac{1}{72C_{\text{TF}}} [s^\sigma(\mathbf{r})]^2 + 0.00187s^\sigma(\mathbf{r})$
- iKEDsel=14: Ou-Yang-Levy 2 KED, $F_{\text{OL2}}^\sigma = 1 + \frac{1}{72C_{\text{TF}}} [s^\sigma(\mathbf{r})]^2 + \frac{0.0245s^\sigma(\mathbf{r})}{1 + 2^{5/3}s^\sigma(\mathbf{r})}$
- iKEDsel=15: Thakkar KED, $F_{\text{Thak}}^\sigma = 1 + \frac{0.0055[s^\sigma(\mathbf{r})]^2}{1 + 0.0253s^\sigma(\mathbf{r})\operatorname{arcsinh}[s^\sigma(\mathbf{r})]} - \frac{0.072s^\sigma(\mathbf{r})}{1 + 2^{5/3}s^\sigma(\mathbf{r})}$
- iKEDsel=16: Becke 86A KED, $F_{\text{B86A}}^\sigma = 1 + 0.0039 \frac{[s^\sigma(\mathbf{r})]^2}{1 + 0.004[s^\sigma(\mathbf{r})]^2}$
- iKEDsel=17: Becke 86B KED, $F_{\text{B86B}}^\sigma = 1 + 0.00403 \frac{[s^\sigma(\mathbf{r})]^2}{\{1 + 0.007[s^\sigma(\mathbf{r})]^2\}^{4/5}}$
- iKEDsel=18: DePristo-Kress 87 KED,
$$F_{\text{DK87}}^\sigma = 1 + 0.00132327[s^\sigma(\mathbf{r})]^2 \frac{1 + 0.861504s^\sigma(\mathbf{r})}{1 + 0.044286[s^\sigma(\mathbf{r})]^2}$$
- iKEDsel=19: Perdew-Wang 86 KED,
$$F_{\text{PW86}}^\sigma = \{1 + 1.296[s_r^\sigma(\mathbf{r})]^2 + 14[s_r^\sigma(\mathbf{r})]^4 + 0.2[s_r^\sigma(\mathbf{r})]^6\}^{1/15}$$
- iKEDsel=20: Perdew-Wang 91 KED,
$$F_{\text{PW91}}^\sigma = \frac{1 + a_1 s_r^\sigma(\mathbf{r}) \operatorname{arcsinh}[b \times s_r^\sigma(\mathbf{r})] + \{a_2 - a_3 e^{-10a_1 s_r^\sigma(\mathbf{r})^2}\} [s_r^\sigma(\mathbf{r})]^2}{1 + a_1 s_r^\sigma(\mathbf{r}) \operatorname{arcsinh}[b \times s_r^\sigma(\mathbf{r})] + a_4 [s_r^\sigma(\mathbf{r})]^4}, \text{ where } a_1=0.19645,$$

$a_2=0.2743, a_3=0.1508, a_4=0.004$ and $b=7.7956$

- iKEDsel=21: Lacks-Gordon 94 KED
$$F_{\text{LG94}}^\sigma = \frac{\{1 + a_2 [s_r^\sigma(\mathbf{r})]^2 + a_4 [s_r^\sigma(\mathbf{r})]^4 + a_6 [s_r^\sigma(\mathbf{r})]^6 + a_8 [s_r^\sigma(\mathbf{r})]^8 + a_{10} [s_r^\sigma(\mathbf{r})]^{10} + a_{12} [s_r^\sigma(\mathbf{r})]^{12}\}^b}{1 + 10^{-8} [s_r^\sigma(\mathbf{r})]^2}$$

, where $a_2=(10^{-8}+0.1234)/0.024974, a_4=29.790, a_6=22.417, a_8=12.119, a_{10}=1570.1, a_{12}=55.944$ and $b=0.024974$

- iKEDsel=22: Acharya-Bartolotti-Sears-Parr KED, $F_{\text{ABSP}}^\sigma = \frac{1}{8C_{\text{TF}}} [s^\sigma(\mathbf{r})]^2 + 1 - \frac{1.412}{N^{1/3}}$
- iKEDsel=23: Gázquez-Robles KED, $F_{\text{GR}}^\sigma = \frac{1}{8C_{\text{TF}}} [s^\sigma(\mathbf{r})]^2 + \left(1 - \frac{2}{N}\right) \left(1 - \frac{1.303}{N^{1/3}} + \frac{0.029}{N^{2/3}}\right)$
- iKEDsel=24: Fourth order gradient expansion approximation,

$$\tau_{\text{GEA } 4}^{\sigma} = \tau_{\text{GEA } 2}^{\sigma} + \frac{(6\pi^2)^{-2/3}}{540} [\rho^{\sigma}(\mathbf{r})]^{1/3} \left\{ \left[\frac{\nabla^2 \rho^{\sigma}(\mathbf{r})}{\rho^{\sigma}(\mathbf{r})} \right]^2 - \frac{9}{8} \frac{\nabla^2 \rho^{\sigma}(\mathbf{r})}{\rho^{\sigma}(\mathbf{r})} \left| \frac{\nabla \rho^{\sigma}(\mathbf{r})}{\rho^{\sigma}(\mathbf{r})} \right|^2 + \frac{1}{3} \left| \frac{\nabla \rho^{\sigma}(\mathbf{r})}{\rho^{\sigma}(\mathbf{r})} \right|^4 \right\}$$

Note that if the "uservar" in settings.ini is not equal to zero, all above mentioned KEDs will be added by " $\nabla^2 \rho / \text{uservar}$ " term. For example, when iuserfunc=1200, iKEDsel=5 and uservar=6, the user-defined function will correspond to $\tau_{\text{GEA2}} + \nabla^2 \rho / 6$ (which corresponds to the KED employed by Tsirelson type of ELF and LOL, as shown in Section 2.6).

1201 Difference between KED selected by "iKEDsel" and Weizsäcker KED.

1202 Difference between KED selected by "iKEDsel" and Lagrangian KED.

1203 Absolute difference between KED selected by "iKEDsel" and Lagrangian KED.

2.8 Graphic formats and image size

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

Tip: If the graph is mainly composed of lines, e.g. contour line map, the best format is pdf.

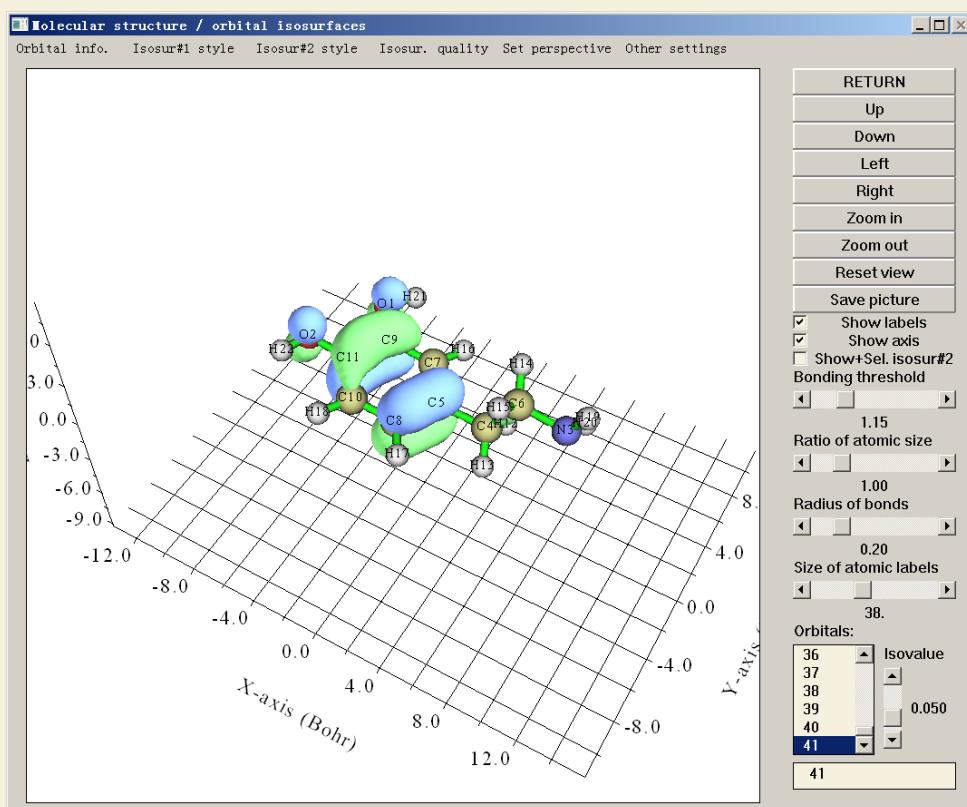
For curve maps, the height and weight of the image file are controlled by “graph1Dsize” parameter in settings.ini. “graph2Dsize” is responsible for two-dimension data plotting (color-filled map, contour map, relief map, etc.). “graph3Dsize” is responsible for three-dimension data plotting (isosurface graph, molecular structure graph).

3 Functions

This chapter introduces all functions of Multiwfn in detail, the numbers in the parentheses of secondary titles are indices of corresponding functions in main menu, the numbers in the parentheses of tertiary titles are indices of corresponding options in corresponding submenus. Different functions of Multiwfn require different types of input file, the information needed by each function are shown in the final line of corresponding section. Please choose proper type of input file according to the table in Section 2.5.

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

When main function 0 is selected, information of all atoms as well as basic information of featured orbitals (e.g. index of HOMO and LUMO, HOMO-LUMO gap) will be printed on text window, meanwhile molecular structure will be shown in a GUI window. If the input file contains orbital information, the orbitals can be viewed by selecting corresponding orbital index in the list at right-bottom corner of the window, or by directly inputting orbital index in the text box, see below screenshot. If the input file is .cub or .grd format, then one can view isosurface of the grid data in this interface.



Since most widgets in this GUI are self-explained, I only mention some worthnoting points here.

Viewing structure

"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 to be shown is vdW surface.

There are three predefined drawing styles of system structure, you can activate one of them via "Use CPK style", "Use vdW style" and "Use line style" in "Other settings" in the menu.

The perspective can be adjusted by "Left", "Right", "Up", "Down", "Zoom in" and "Zoom out" buttons. To exactly control the viewpoint, you can also directly input value of rotation angle by selecting "Set rotation angle" option in "Set perspective" in the menu. In addition, you can rotate your mouse wheel in the drawing region to zoom in and zoom out.

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

The color of bonds, atomic labels and atom spheres can be set by parameters "bondRGB", "atmlabRGB" and "atmcolorfile" in settings.ini, respectively. See corresponding comments for detail. After adjusting these parameters, you need to reboot Multiwfn to make them take effect. You can also directly change atomic label color in the GUI via "Other settings"- "Set atomic label color".

Viewing orbitals

By clicking "**Orbital info.**" and then selecting corresponding options, basic orbital information can be printed on text window.

All orbital indices are listed in the right-bottom box, default selection is "none" (no orbital is shown). If you click an orbital index, Multiwfn will calculate grid data of wavefunction value for corresponding orbital, and then the orbital isosurface appears immediately. Green and blue parts correspond to positive and negative regions, respectively. The "Isovalue" slide bar controls the isovalue of the isosurface.

For efficiency consideration, the default quality of grid data is relatively coarse, the number of grid points can be set by "**Isosur. quality**", the larger number leads to the smoother isosurface (Note that after you adjust the number, present isosurface will be deleted). You can also set default number of grid points through changing "nprevorbgrid" in settings.ini.

For unrestricted wavefunction, the alpha and beta orbitals are recorded separately. Assume that there are N_a alpha orbitals and N_b beta orbitals, then the first N_a and latter N_b orbitals in the orbital selection list correspond to alpha and beta orbitals, respectively. Whereas in the orbital selection box, negative index corresponds to beta orbitals. For example, if you input -9, then the 9th beta orbital will be shown, and the orbital selection list will be automatically switched to the N_a+9 term.

Sometimes it is useful to display two orbitals simultaneously; for example, analysis of phase overlapping of two NBOs. This can be realized by "**Show+Sel. Isosur#2**" (Sel.=Select) check box. This check box is inactive by default, once you have chosen an orbital (corresponding isosurface will be referred to as Isosurface #1), this check box will be actived. After you clicked the check box, the isosurface of the orbital you newly selected (which will be referred to as Isosurface #2) will be shown together with the orbital you previously selected, yellow-green and purple parts correspond to positive and negative regions respectively. If the check box is deselected, Isosurface #2 will disappear, then if you reselect the check box the same Isosurface #2 will be redrawn.

The representation styles of Isosurface #1 and #2 can be adjusted by suboptions in "**Isosur.**

"**#1**" and "**"Isosur. #2"**" individually, available styles include solid face (default), mesh, points, solid face+mesh and transparent face. (Notice that Multiwfn cannot plot only one isosurface as transparent face). The colors for face and mesh/points can be set by corresponding suboptions too, users will be prompted to input Red, Green, Blue components for positive and negative parts in turn. The component values should between 0.0 and 1.0, for instance, 1.0,1.0,0.0 corresponds to Yellow. Opacity for transparent face can also be customly set, valid range is from 0.0 (completely transparent) to 1.0 (completely opaque).

Before showing orbital isosurfaces, Multiwfn first sets up a box internally, then calculates orbital wavefunction value at the points evenly placed in the box. The default extension distance for setting up the box is controlled by "Aug3D" parameter in *settings.ini*. The default value is suitable and efficient for general cases; however, in rare cases (*e.g.* visualizing Rydberg orbitals) you may need to manually enlarge the extension distance, you can either change the Aug3D to modify the default setting, or select "**Other settings**"-"**Set extension distance**" and then input a value to modify extension distance for present instance.

When orbital isosurface is portrayed as solid face, in rare cases the plotting effect is not quite good, for example positive and negative regions are difficult to be distinguished, in these cases you can try to adjust lighting setting by choosing "**Other settings**"-"**Set lighting**" or properly rotate the system to bypass this issue.

Information needed: GTFs (only for viewing orbitals), atom coordinates, grid data (only for viewing isosurface of grid data)

3.3 Outputting all properties at a point (1)

Input coordinate of a point or index of an atom, then the values of all real space functions supported by Multiwfn at the 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.

If input *d*, value of a real space function at a given point can be decomposed into orbital contributions. The contribution from orbital *i* is evaluated as follows: First set occupation number of all orbitals to zero except for orbital *i*, and then calculate real space function as usual. Note that sum of contribution of all orbitals may be different to the result when all orbitals are simultaneously into account, since many real space functions contain non-linear operators, such as ELF.

Electrostatic potential is the most expensive one among all of the real space functions supported by Multiwfn. If you are not interested in it, you can set "ishowptESP" parameter in "settings.ini" to 0 to skip calculation of electrostatic potential.

Information needed: GTFs (depends on the choice of real space function), atom coordinates

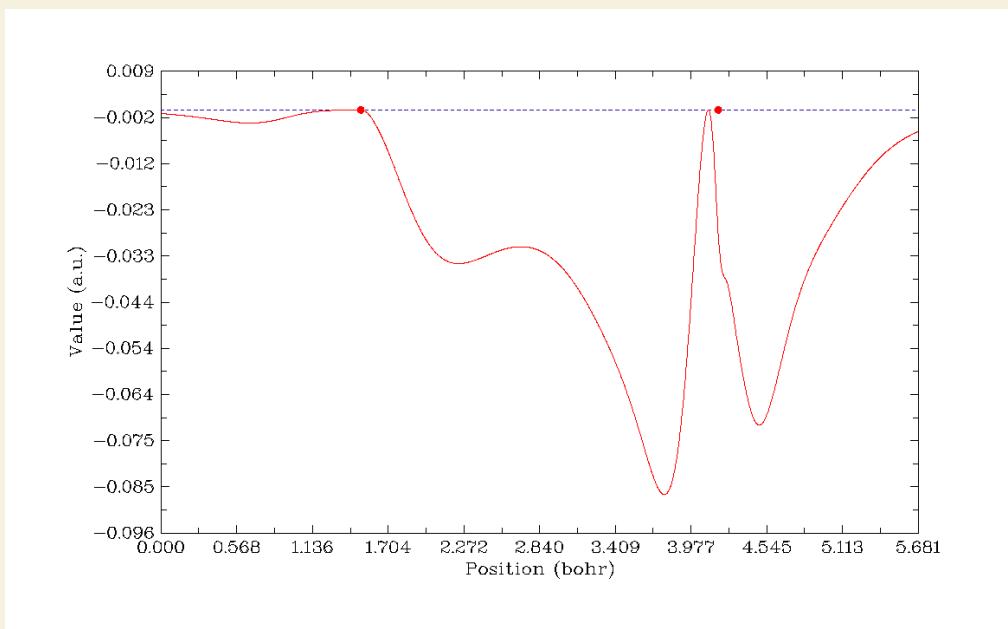
3.4 Outputting and plotting specific property in a line (3)

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 extension distance 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, then curve map pops up immediately, like this



The gray dashed 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.

Information 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 of using this function are listed below. Users can finish all operations by simply following program prompts, there are only a few key points are needed to be described.

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

Worthnotingly, after you enter this function, you will see real space function selection menu; if you select option 111 (a hidden option), then the real space function to be plotted will be Becke weight of an atom or Becke overlap weight between two atoms. If you select option 112 (another hidden option), Hirshfeld weight of a given atom or fragment will be plotted.

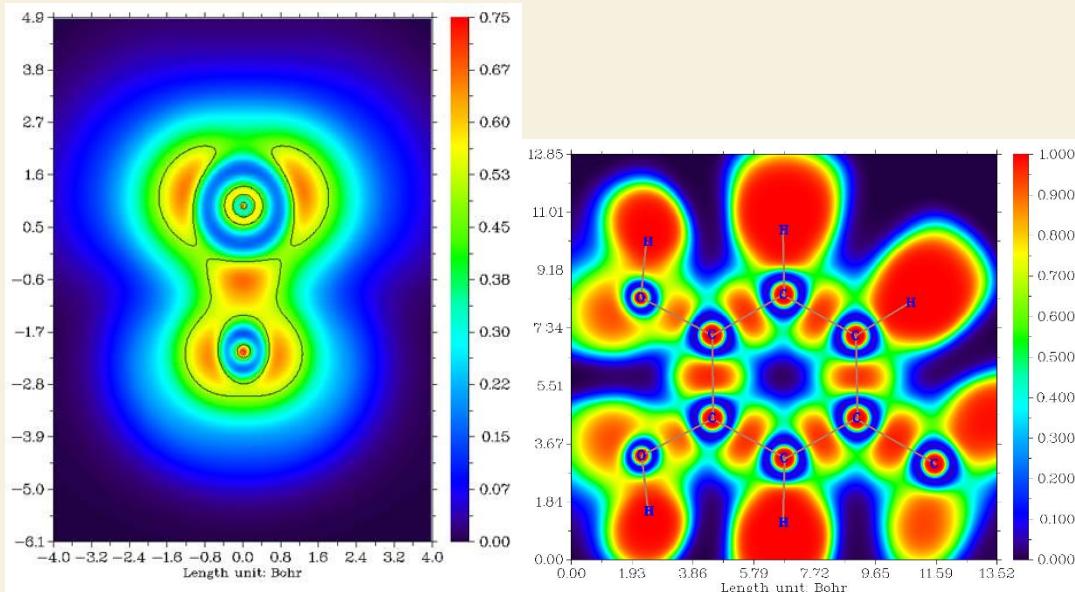
If the parameter "iplaneextdata" in settings.ini is set to 1, then the data in the plane will not be calculated by Multiwfn internally but directly loaded from an external plain text file. The coordinate of the points in the plane will be automatically outputted to current folder, you can use third-party tools (*e.g.* cubegen utility of Gaussian) to calculate function value at these points.

3.5.1 Graph types

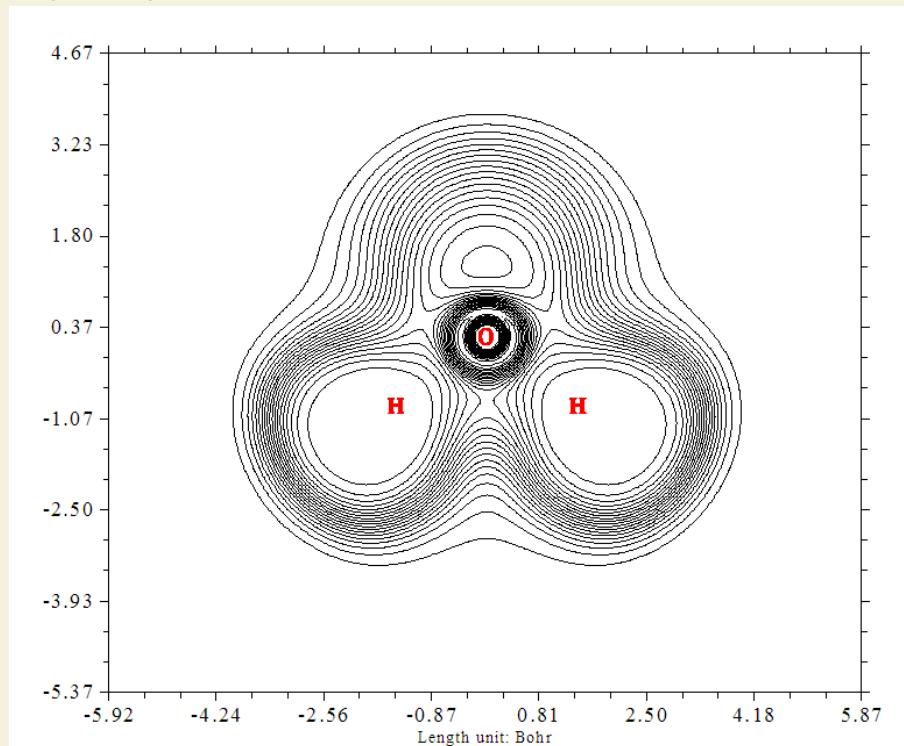
Currently Multiwfn supports seven graph types for exhibiting data in a plane.

1 Color-filled map. This type of map uses different colors to represent real space function value in different region. By default, if the function value exceeds lower (upper) limit of color scale, then the region will be filled by black (white); if you want to avoid this treatment, simply set the parameter "inowhiteblack" in settings.ini to 1. If you want to reverse color scale as red-white-blue, in post-process menu you can set the lower limit smaller than upper limit, *e.g.* inputting *I,-I*.

By selecting “Enable showing contour lines” in post-process menu, contour lines can be plotted together on the graph.



2 Contour line map. This map uses solid lines to represent positive regions, and dashed lines to exhibit negative regions.



The number of isovalue lines can be adjusted by user (see Section 3.5.4), you can also mark the isovalue on the contour lines by using the option “2 Enable showing isovalue on contour lines” in post-process step.

Only for contour line map, if the real space function you selected to be plotted is orbital wavefunction, you can not only plot one orbital by inputting one orbital index, but also plot two orbitals simultaneously by inputting two orbital indices (e.g. 3,5), see Section 4.4.5 for example.

Option 4 in post-process menu is used to toggle if showing atoms labels or reference point (involved in some real space functions such as correlation hole and source function) in the graph. The size and form (i.e. if showing atom index or element name) of the atom labels can be set by "pleatmlabsize" and "iatmlabtype" parameters in settings.ini, respectively. The reference point is represented as a blue asterisk on the graph. By default only when the distance between the atoms and the plane is smaller than "disshowlabel" in settings.ini the corresponding atom labels will be shown. To change this distance threshold, you can either adjust this value in setting.ini or choose option 17 in post-process menu. If "iatom_on_plane_far" parameter in settings.ini is set to 1, then even if the distance is larger than the threshold, the label will still be shown, but in thin text rather than bold text.

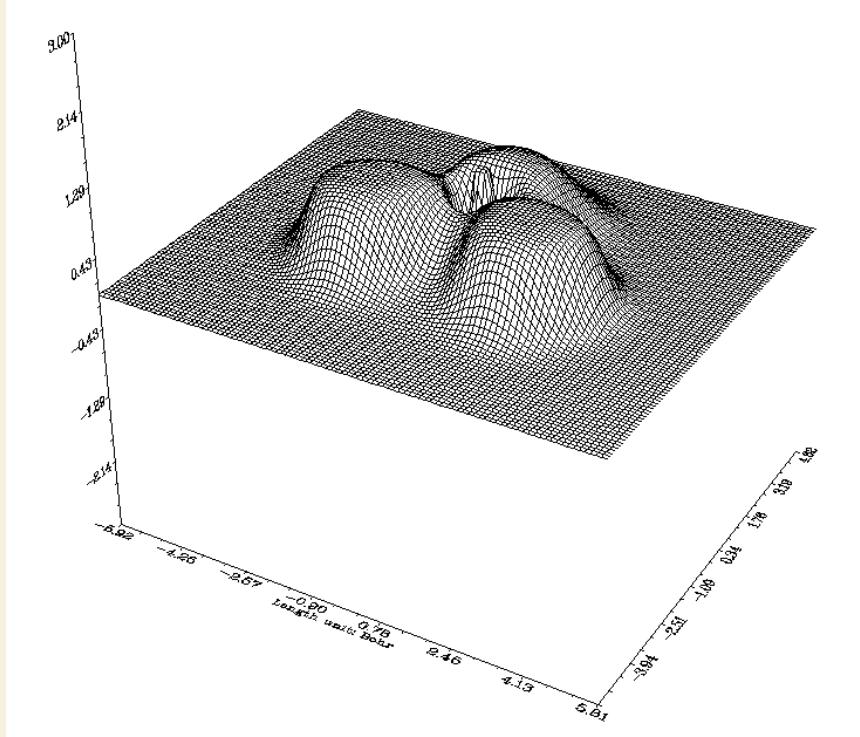
Option 8 in post-process menu is used to toggle if showing bonds. The bond is shown for an atom pair if these two conditions are satisfied: (1) The distance between the two atoms is smaller than bonding threshold, which can be adjusted in the GUI window of main function 0 using corresponding scale bar. (2) Both atoms are close enough to the plotting plane (smaller than the "disshowlabel" parameter mentioned above).

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 in *J. Am. Chem. Soc.*, **109**, 7968 (1987)). This

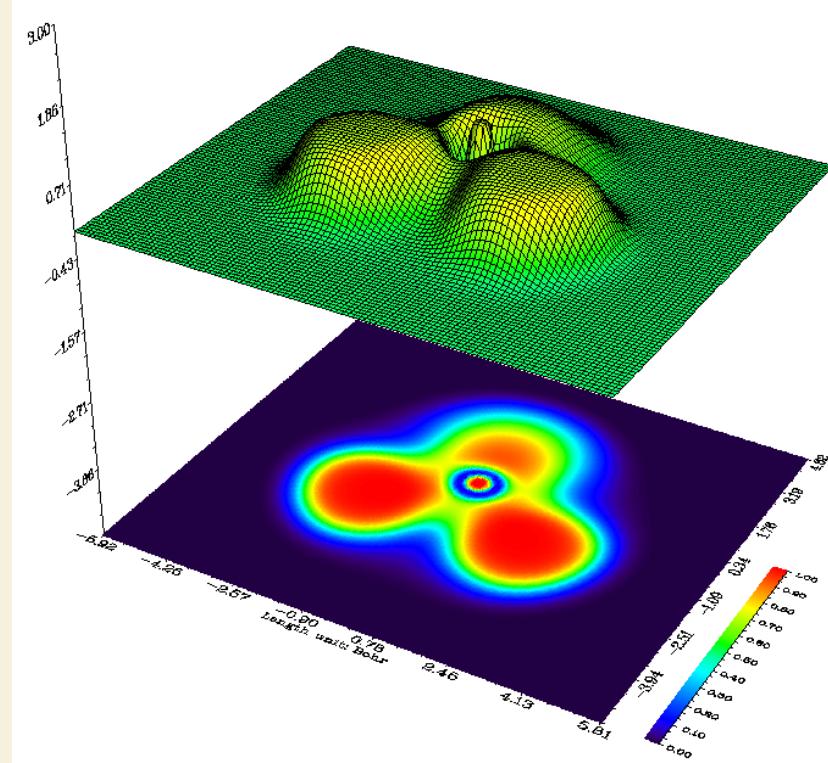
is useful to analyze distribution of electrostatic potential on vdW surface. Such a contour line can be plotted in gradient line and vector field map too by the same option. Color, label size and line style of the contour line can be adjusted by Option 16.

Notice that the discussion in above two paragraphs also applies to color-filled map, gradient lines map and vector field map (see below).

3 Relief map. Use height to represent value at every point. 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.

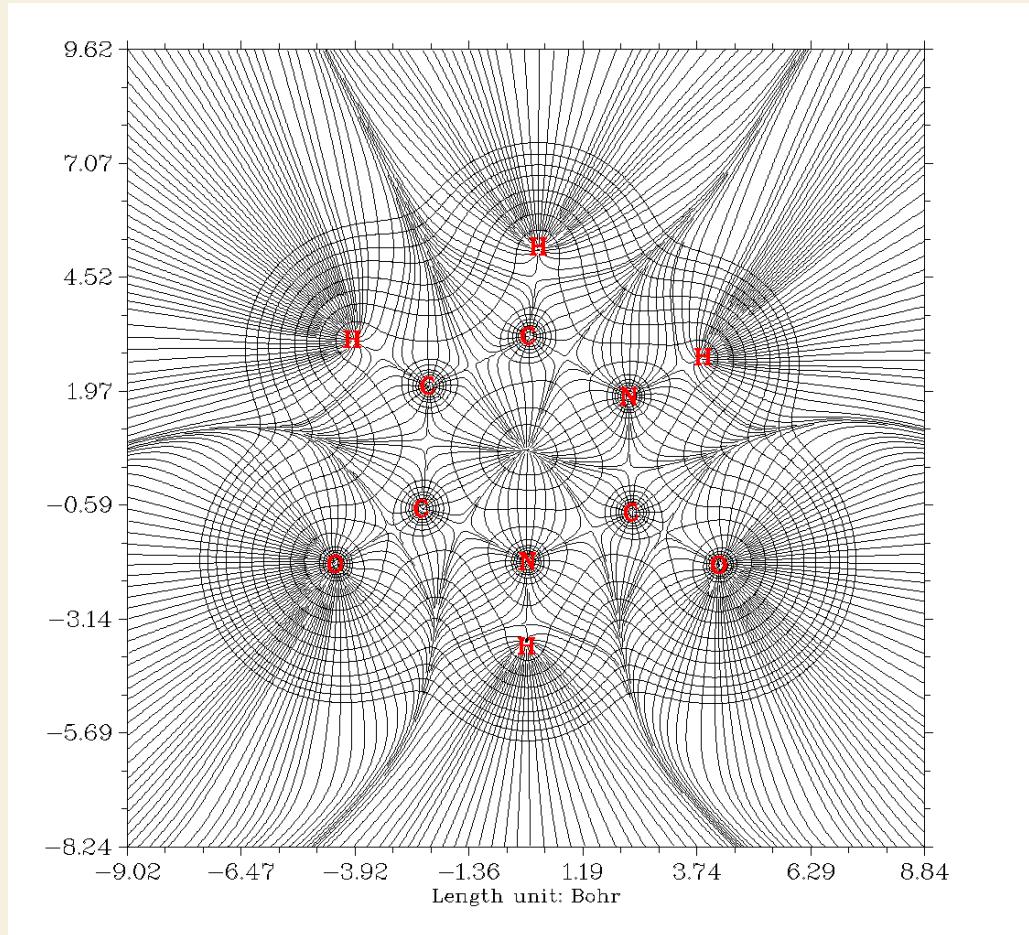


4 Shaded surface map and 5 Shaded surface map with projection. The relief map is shaded in these two types. The latter also plots color-filled map as projection. The meshes on the surface can be disabled at post-process stage.

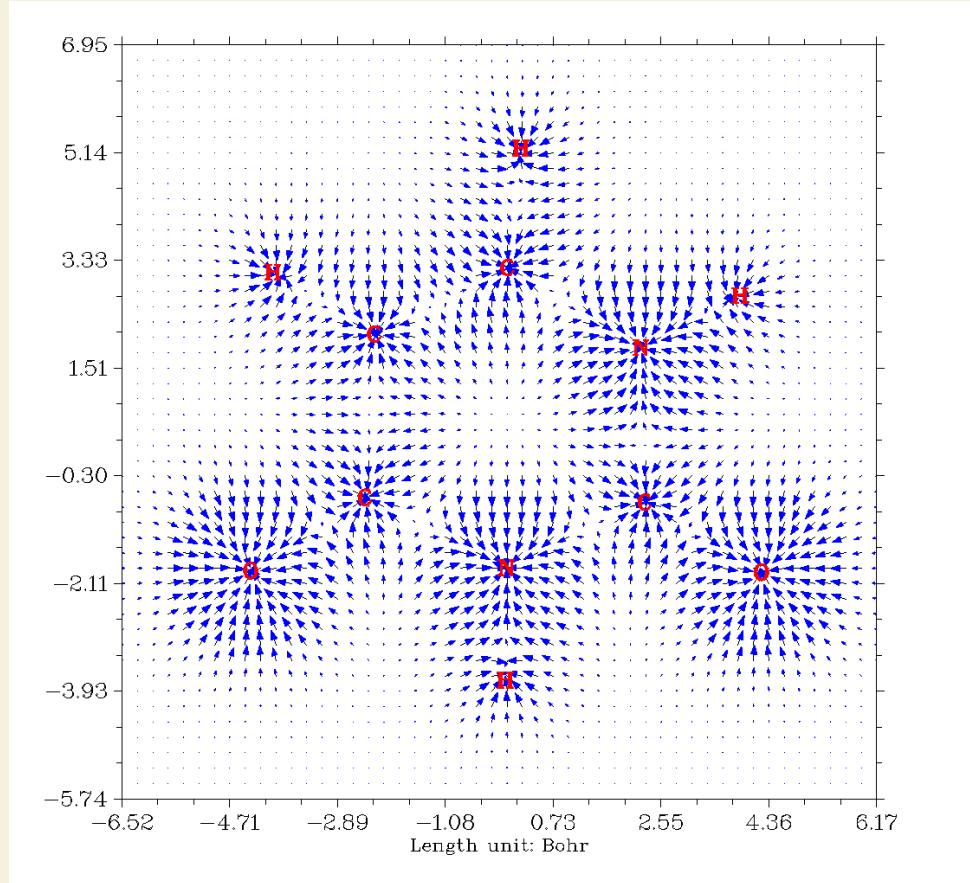


6 Gradient line map with/without contour lines. This graph type represents gradient direction of real space function, you can determine if the contour lines also be shown on the graph. Note that since gradients of real space function are needed to be evaluated, and graphical library needs to take some 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. 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 dimensions 1 and 2 the number of grid points are 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, the lines in the graph will look too crowd. For other graph types I recommend *200,200*. Of course the picture will become more pretty and smoother if you set the value larger, but you have to wait more time for calculation. Bear in mind that total ESP calculation is very time-consuming, you’d better use less grid points, for previewing purpose I recommend *80,80* or less.

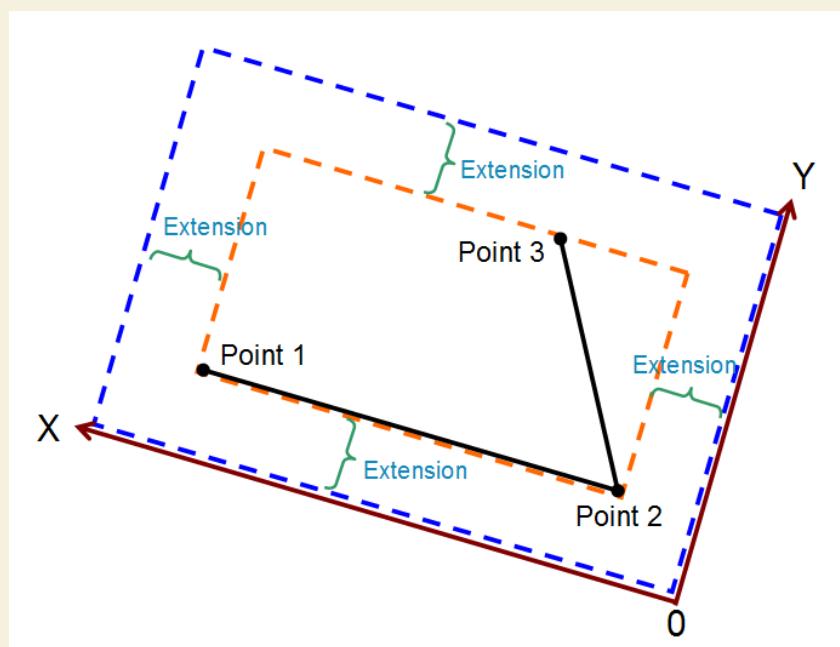
Multiwfn provides 7 modes to define the plotting plane:

1. XY plane: User inputs Z value to define a XY plane uniquely.
2. XZ plane: User inputs Y value to define a XZ plane uniquely.
3. YZ plane: User inputs 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.

7. Define a plane parallel to a bond and meantime normal to a plane defined by three atoms

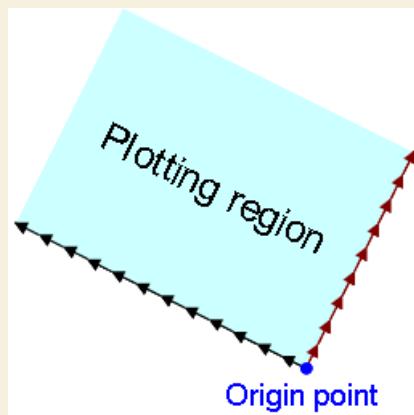
For modes 1~5, the actual plotting region is a subregion of the plane you defined. Multiwfn automatically sets the plotting region to tightly enclose the whole molecule (for modes 1, 2 and 3) or cover the three nuclei / points you inputted (for modes 4 and 5), finally the plotting region is extended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

Below diagram illustrates how the actual plotting region is determined when you select mode 4 or 5 to define the plotting plane. The X and Y axes shown in the graph correspond to the actual X and Y axes you finally see. Evidently, the input sequence of the three points or atoms directly affects the graph.

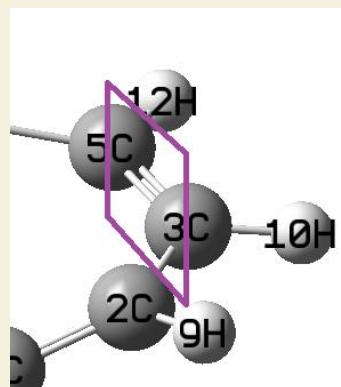


For the modes 4 and 5, if you find the content is skewed in the final graph, or the interesting part is not located at the center of the graph, you can choose “**-1: Set translation and rotation of the map for plane types 4 and 5**” before selecting mode 4 or 5. For example, if you find the content in the graph your previously plotted should be translated by (-3,1.5) Bohr and then rotated by 35°, then in this option you should first input -3,1.5 and then input 35, the resulting graph will meet your expectation. (A practical instance of using this option was posted on <http://bbs.keinsci.com/thread-11037-1-1.html>)

For mode 6, the plotting region is determined as follows, in which each black arrow denotes translational vector 1, each brown arrow denotes translational vector 2, blue point denotes origin point. The number of arrows is the number of grids set by users. Evidently, this mode enables users to fully control the plotting plane setting.



Mode 7 is very useful when you want to define a plotting plane cutting a bond but there is no proper reference atoms to use mode 4, below graph is such an instance, the purple rectangle is the plane you want to plot:



To define this plane, you should select mode 7, and then input 3,5 to use these two atoms to define the axis that the plotting will be parallel to, and then input 2,3,10 (or 2,5,10 etc.) to use them to define a plane that the plotting plane will be normal to. After that you need to input the length of X and Y axes, e.g. 10 Bohr and 7 Bohr, respectively.

3.5.3 Options in post-process interface

After plotting the graph, you will see a menu, in which there are a lot of options used to adjust or improve the quality of the graph. Since many of them have already been introduced above and some of them will be mentioned in next sections, and lots of them are self-explained, only a few will be mentioned here.

-9 Only plot the data around certain atoms: Sometimes in the graph only a few regions are interesting; if you want to screen other regions, you may find this option useful. After selecting this option, assume that you input 2,4,8-10, then only the real space function around atoms 2,4,8,9,10 will be plotted (the data to be plotted then in fact is the original plane data multiplied by the Hirshfeld weight of the fragment you inputted). Next time you select this option, the original data will be recovered.

-7 Multiply data by a factor: This is mainly used to scale the range of the plane data.

-6 Export calculated plane data to plane.txt in current folder: After using this option to export the plane data, you can very conveniently using third-part plotting softwares such as SigmaPlot to redraw the data.

-2 Set stepsize in X,Y,(Z) axes: This option determines the spacing between the labels in the coordinate axes. If axis labels are not shown, that means you need to decrease the stepsizes.

-1 Show the graph again: After adjusting plotting parameters, choose this option to replot the graph to check the effect.

0 Save the graph to a file: Export the graph to a graphic file in current folder. See Section 2.8 on how to determine the graphic format and size.

3.5.4 Setting up contour lines

For graph types 1, 2, 6 and 7, the contour lines can be plotted (if not shown, select "2 Enable showing contour lines" in the post-process menu). There is also an option "Change setting of contour lines" in the post-process menu. In this interface, values of current contour lines are first listed and you can modify them by using below suboptions:

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 value of a contour line.

Option 3: Add a new contour line and input the isovalue for it.

Option 4: Delete some contour lines.

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: Generate isovalue according to arithmetic sequence, user need to input initial value, step size and total number. For plotting ELF/LOL, I suggest user input 0,0.05,21 to generate isovalue in the range 0.0~1.0 with step size 0.05. You can choose if cleaning existing contour lines, if you select n , then new contour lines will be appended to old ones.

Option 9: Like 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.

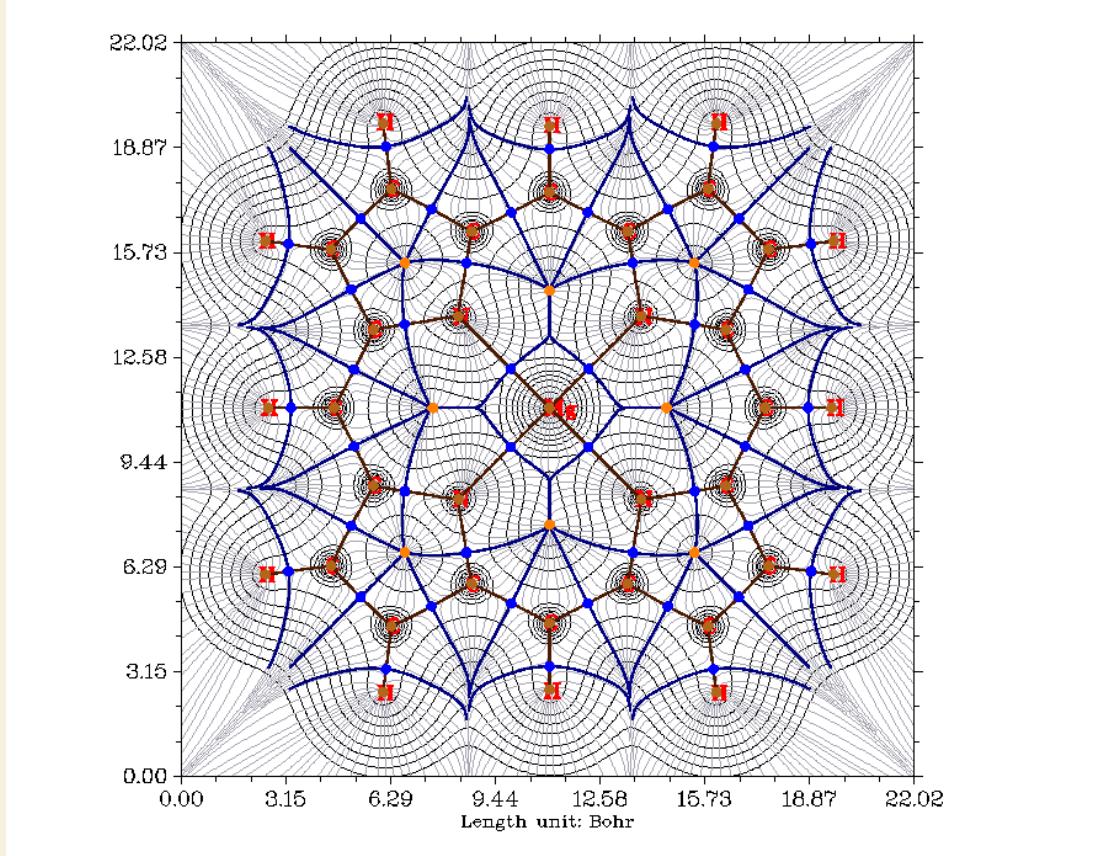
Option 12: Set line style for positive contour lines, you need to input two integer number, the first one denotes the length of line segment, the second one denotes the length of interstice between line segment. For example 10,15 means positive contour lines are composed of line segments with length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

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

Option 15: If this option is selected, the positive and negative lines will be set to "crimson, width=6, solid line" and "blue, width=6, long dashed line", respectively. Then the saved picture will be very suitable for publication; as you can see in the resulting graphical file, the two kinds of lines are clear and can be distinguished easily.

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

CPs and paths can be plotted on color-filled map, contour line map, gradient line map and vector field map. Below is a typical electron density gradient line map containing critical points, topology paths and interbasin paths.



In order to append the CPs and paths on plane map, before drawing the plane graph using main function 4, you need to enter topology analysis module (main function 2), then search CPs and generate paths. After that, return to main menu and draw plane graph as usual, you will find that the CPs and paths have appeared on the graph. In the graph, brown, blue, orange, green dots denote (3,-3), (3,-1), (3,+1), (3,+3) critical points, respectively. Bold dark brown lines depict bond paths.

In the option “4 Set details of plotting critical points and paths” at post-process menu, you can choose which types of CPs are allowed to be shown, and you can set size of markers, thickness, distance threshold and color of path lines. By default, if the vertical distance between a CP or a point in a path and the plotting plane exceeds 0.5 Bohr, the CP or path point will not be shown, the thresholds can be altered by “8 Set distance threshold for showing CPs” and “9 Set distance threshold for showing paths”.

Interbasin paths (deep blue lines in above graph) are derived from (3,-1) CPs, these paths dissect the whole space into individual atomic basins. You can also append the interbasin paths on contour line map, gradient line map and vector field map. In order to draw interbasin paths, you should first confirm that at least one (3,-1) CP has been found in topology analysis module and it is close enough to current plotting plane (smaller 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 replot the plane graph again, you will find these interbasin paths have already presented.

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

Information needed: GTFs (depending on the choice of real space function), atom coordinates

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

The main purpose of this function is calculating grid data or generating Gaussian-type cube file (.cub) for specific real space function. 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 viewed directly in Multiwfn too. This function can also be used to calculate function values for a set of points recorded in an plain text file.

The basic procedure of using this function is:

- (1) Select a real space function
- (2) Set up grid
- (3) Use post-process options to visualize isosurfaces, modify and export data

In the real space function selection menu, if you select option 111 (a hidden option), then the real space function to be calculated will be Becke weight of an atom or Becke overlap weight between two atoms. If you select option 112 (another hidden option), Hirshfeld weight of a given atom or fragment will be calculated.

Setting up grid

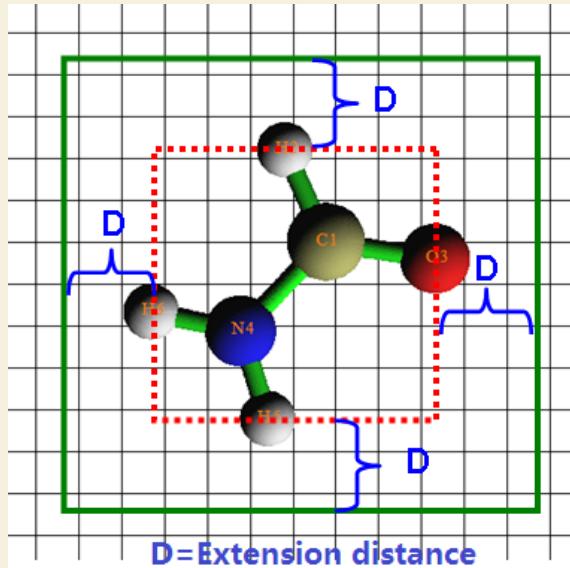
Multiwfn provides many modes for setting up grid point:

Mode 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 purpose.

Mode 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.

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

For modes 1, 2 and 3, the actual number of points in each direction is automatically determined by Multiwfn so that grid spacing in each direction are nearly equal. The method of determining spatial scope of grid data is illustrated below in two-dimension case.



First, Multiwfn sets up a box (red dashed line) to just enclose the entire molecule, and then the box is suitably extended in each dimension to avoid truncating boundary part of isosurfaces, the green rectangle in below graph is the actual spatial scope of grid data. If the extension distance is inappropriate, you can customly set the value by using option “-10 Set extension distance of grid range”, or directly set default value by modifying “Aug3D” in settings.ini.

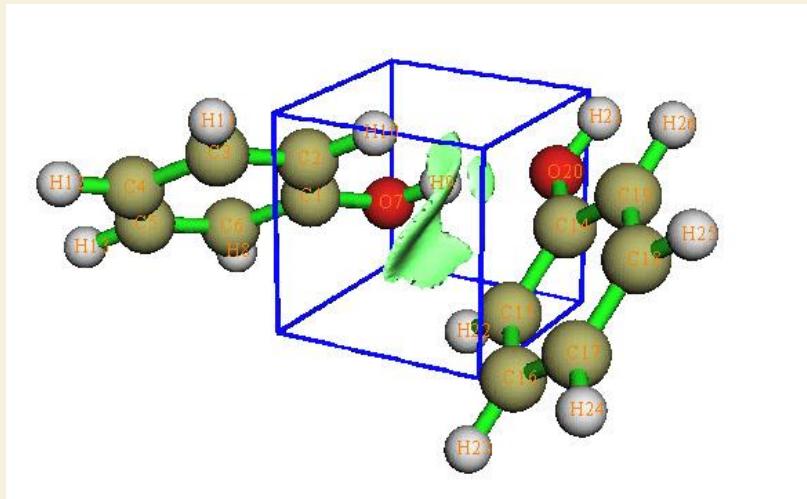
The so-called “Low quality”, “Medium quality” and “High quality” are **only relative to small system**. If spatial range of the grid data is wide, then density of grid points of mode 3 will not be high and should be called “Medium quality”, and mode 2 should be called “Low quality” at this time. While for very small system, the density of mode 1 is already adequate for general purposes.

Mode 4: Specify the number of grid points or grid spacing in X, Y and Z directions by yourself, the spatial scope is determined automatically as shown above.

Mode 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 experienced users.

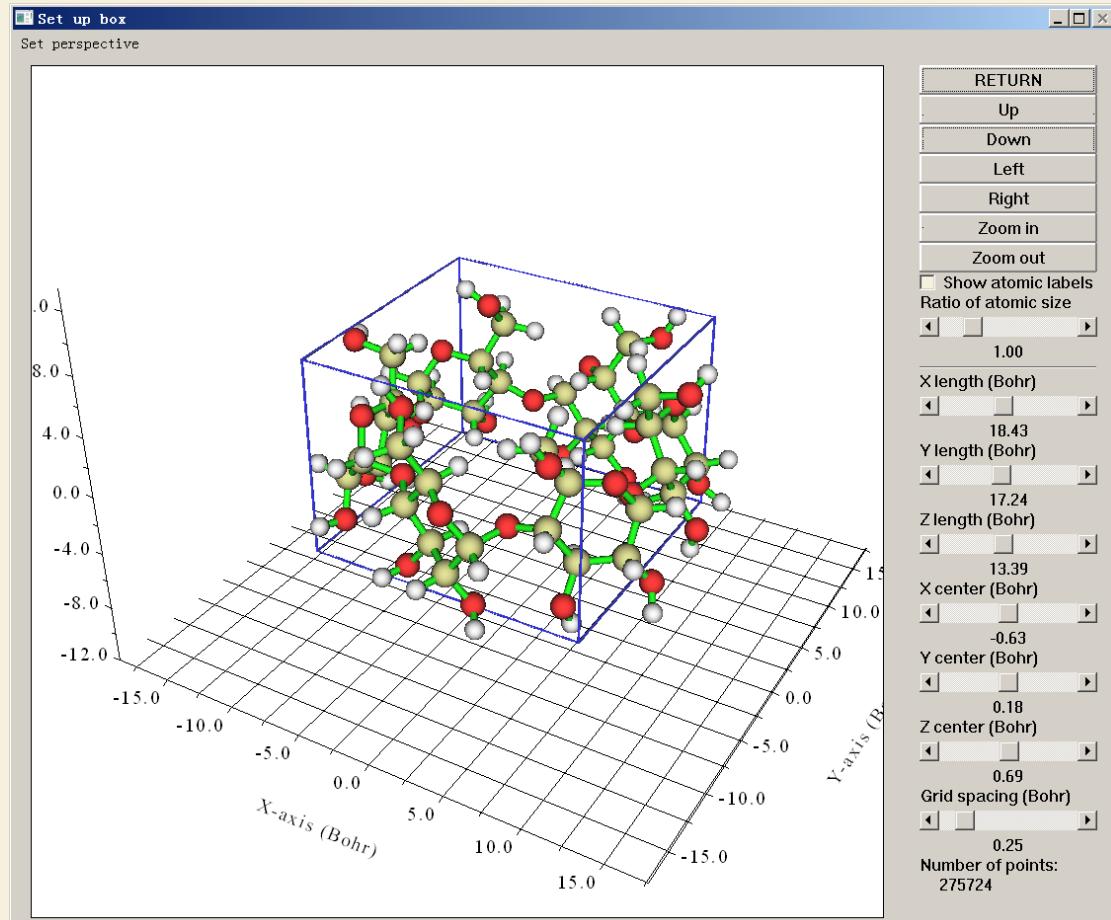
Mode 6: Specify the center coordinate, number of points and extension distance in X, Y and Z directions. For example, the center coordinate you inputted is 2.3,1.0,5.5, the extension distance in X, Y and Z is k,k,m , then the coordinates of two most distant endpoints are $(2.3-k,1.0-k,5.5-m)$ and $(2.3+k,1.0+k,5.5+m)$.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

Mode 7: Like function 6, but inputting 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 3.23.1 for example. If the two atom indices are identical, then the nuclear coordinate will be set as center.



Mode 8 Use grid setting (origin, number of points and grid spacing) of an existing cube file. You will be prompted to input the file name. This mode is useful to generate multiple cube files with exactly identical grid setting.

Mode 10 Set up the grid data in a GUI window. This mode is quite convenient, the box is clearly visible in the graphic window, and you can directly change box length, box center and grid sapcing by dragging corresponding scale bars (the X,Y,Z directions share the same grid spacing). The total number of grids under present grid setting can be directly seen at the right bottom side of the window.



Notice that the more grid points you set, the finer the isosurface graph you will get, however the more time is needed for calculating grid data and generating isosurface graph. The increase of calculation time with the number of points is nearly linear.

After setting grid points, the program starts 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 based on grid data is also printed out. Then you will see post-process menu.

Post-process options

In the post-process menu if the option “-1 Show isosurface graph” 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 your 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 not be displayed). The isosurfaces with the same and reverse sign of current isovalue are in green and blue respectively.

Option 1 can export isosurface graph to graphic file in current directory. Option 2 can export grid data to .cub file in current directory. By selecting option 3 the grid data will be exported to plain text file “output.txt” in current directory. Using option 4 you can set isovalue without entering GUI window and dragging slide bar, it is useful for batch process and in command-line environment. With options 5~8, you can perform addition, subtraction, multiplication and division operations on the grid data, respectively.

Special note: Calculate data for a set of points

If you want to use Multiwfn to calculate real space function value for a set of points (may be arbitrarily distributed), in the interface for setting up grid, you should choose "100 Load a set of points from external file" and then input the path of the plain text file recording the points. In the file the first line should be the number of points recorded, and followed by X/Y/Z coordinates of all points. For example

1902

```
-3.3790050  -2.0484700   0.0274117
-3.3844930  -1.9472468  -0.2500274
-3.4064601  -1.9221635  -0.1287148
-3.4118258  -1.9232203  -0.0003350
-3.4059106  -1.9218280   0.1351179
...
```

The coordinates must be given in Bohr. The format is in free format, and can have more than four columns of data, but the columns after the fourth one will be simply ignored.

Multiwfn will load coordinates of the points from this file, and then calculate function value for them. Finally, the X/Y/Z coordinate and function value of all points will be outputted to a plain text file, whose path is specified by users.

Information needed: Atom coordinates, GTFs (depends on the choice of real space function)

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 main function 3, 4 and 5, there is a subfunction allow you to set custom operation for multiple wavefunctions. Supported operators include + (add), – (minus), * (multiply), / (divide), there is no upper limit of the number of wavefunctions involved in custom operation. For example, if the first loaded wavefunction after booting up Multiwfn is *a.wfn*, then in the setting step of custom operation you inputted 2 (viz. there are two wavefunctions will be put into “custom operation list” and thus will be operated with *a.wfn* in turn), then you inputted *-b.wfn* and **c.wfn*, the property finally you get will be [(property of *a.wfn*) - (property of *b.wfn*)] * (property of *c.wfn*). If you are confused, you can consult the example in Sections 4.5.4 and 4.5.5.

Sometimes the molecular structure in the first loaded file and that in the subsequently loaded files are not identical, the grid points you set will be for the first loaded file, all of the other files will share the same grid setting.

Avoid using custom operation in conjunction with main functions -4, -3 and 6, otherwise you may get absurd result.

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

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

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

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

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

This is an artificial density that corresponds to the state when molecule has formed but the density has not relaxed.

Deformation property is the difference between actual property and promolecular property of a molecule under the same geometry

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

If the property is chosen as electron density, then the deformation property is generally called deformation density or known as electronic bonding charge distributions (BCD), which is very useful for analyzing charge transfer and bonding nature. Application example of deformation

density can be found from my paper [Acta Phys. -Chim. Sin., 34, 503 \(2018\)](#).

3.7.3 Generation of atomic wavefunctions

Evaluation of promolecular and deformation properties and some functions in Multiwfn request atomic wavefunction files, such as calculating Hirshfeld, VDD and ADCH charges, fuzzy atomic space analysis and orbital composition analysis based on Hirshfeld partition, etc. The process of generating atomic .wfn files are exactly the same.

After you select subfunctions -1 or -2 to study promolecular and deformation properties, Multiwfn checks whether .wfn files of all elements involved in present system have been presented in “atomwfn” subdirectory of current directory, if not, Multiwfn automatically invokes Gaussian to generate the missing element .wfn files and sphericalizes their densities. If the path of Gaussian executable file (“gaupath” parameter in settings.ini) is incorrect or has not been defined, Multiwfn will ask you to input the path of Gaussian executable file. The basis set for generating the element wavefunctions can be arbitrarily set by user, however it is suggested to use the same basis set as the molecular wavefunction.

The newly generated element wavefunction files or those taken from “atomwfn” directory are stored in “wfntmp” subdirectory in current directory. They will be translated to actual position of the atoms in present system, meanwhile atomic indices will be added to the .wfn filename (e.g. "Cr 30.wfn"). These are the files will directly be used to calculate promolecular and deformation properties.

Details and skills

If you want to avoid the step of generating element wavefunctions every time, you can move the.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 detected that all needed element .wfn files have already been presented in“atomwfn” folder, then Multiwfn will directly use them. Multiwfn only invokes Gaussian to calculate the missing element .wfn files.

“atomwfn” subdirectory in “examples” directory contains element wavefunction files of all first four rows, they were generated under 6-31G*, and they have been sphericalized. If you want to use them, simply copy the "atomwfn" directory to current folder.

There is a quick way to generate all first four-rows element wavefunction files: the file “examples\genatmwfnpdb” contains all first four-rows atoms, load it into Multiwfn and generate promolecular property, after the calculation of element 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 wavefunction files by invoking Gaussian and sphericalize their density automatically; in that cases you have to calculate and sphericalize atom wavefunctions manually, and then put them into “atomwfn” directory, Multiwfn will directly use them.

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 insensitive 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**. Do not add “RO” or “U” prefix since they will be added automatically. If error occurs 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 permitted by Gaussian is only 60! The path will be truncated and cause error if the length exceeded this threshold. So do not 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 in spherical symmetry, hence will lead to orientation dependence problem. To tackle it, atomic 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 0.

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, as well as 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. These methods also fail if Hartree-Fock method is used for transition metals, because orbital order produced by HF is different from most DFT cases (the HF’s order is wrong, 4s is higher than 3d).

3.8 Checking & Modifying wavefunction (6)

This function provides a lot of subfunctions for checking and modifying loaded wavefunction, all subfunctions take effect immediately, so after modification you can print related information 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 wavefunction information will be affected. The title in this interface shows the number of GTFs, orbitals, atoms and alpha/beta electrons of current wavefunction.

Below is explanation of each subfunction in this category.

Subfunction -1: Return to main menu.

Subfunction -3 and -4: Defining molecular fragment. In -3 you directly define fragment by inputting atomic indices; while in -4 you should input the indices of the atoms to be excluded, the remaining atoms will constitute the fragment. The GTFs whose centers do not belong to this fragment will be discarded (can be imagined as corresponding GTF coefficients in all orbitals are set to zero). This function affects all the following tasks which make use of GTF information, in particular, the calculations involving real space functions.

When use these two functions? For large molecules, what you are interested in may be only a small region, by discarding atoms in other region, the calculation speed of time-consuming task such as generating grid data can be increased evidently. Besides, sometimes you may only want to obtain real space function contributed by certain atoms, these two functions can fulfill your purpose.

This function can be safely used together with main function 1~6. For other tasks, DO NOT use this function if you do not understand what you are exactly doing, otherwise you may obtain wrong or meaningless results. Notice that this function is irreversible, the discarded GTFs cannot be recovered, unless you reboot the program and reload the wavefunction file.

Subfunction 0: Save current modified wavefunction to “new.wfn” file in current directory. Multiwfn can be used as an .fch/.molden/.gms→wfn format converter through this function. Notice that the orbitals with zero occupation number and the atoms do not have GTFs will be automatically discarded during saving.

Subfunction 1: Print information of all GTFs, including the centers they are belonging to, 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 information), 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 information.

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

Subfunction 7: Print various kinds of integral matrix between basis functions, such as overlap integrals, electric/magnetic dipole moment integrals, kinetic energy integrals and velocity integrals. For overlap matrix, eigenvalues are printed together, which are useful for checking linear dependency. Only available when the input file contains basis function information.

Subfunction 11: Swap centers or types or exponents or orbital expansion coefficients of two GTFs, or swap all information of two GTFs at once (identical to swap record order of two GTFs, thus does not 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.

If the input file also contains basis function information, by this option you can also set expansion coefficients of basis functions.

Subfunction 26: Set occupation number of some orbitals. You can directly set them to a specific value, or add, minus, multiply, divide them by a given value. **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 28: Set energy of some orbitals. You can directly set them to a specific value, or add, minus, multiply, divide them by a given value. This function is useful when you want to rectify the orbital energies using a given relationship (e.g. *J. Am. Chem. Soc.*, **121**, 3414 (1999)) before plotting density-of-states (DOS) map.

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 useful for extending the primitive cell wavefunction outputted by Gaussian PBC function to supercell wavefunction, of course you can calculate supercell wavefunction directly by Gaussian, but much more computational time will be consumed. Notice that constructing wavefunction of supercell by this way is only an approximation, because the mix between orbitals of neighbouring primitive cell to new supercell orbitals is completely ignored.

Subfunction 33: Rotate wavefunction, namely X→Y, Y→Z, Z→X

Subfunction 34: Set occupation number of all inner molecular orbitals in present system to zero, namely only contribution from valence orbitals will be reserved for following studies. The number of inner MOs is automatically determined, however, the atoms using pseudo-potential are not taken into account.

Subfunction 35: This function is very useful if you want to discard contribution of orbitals with certain irreducible representation (IRREP) in all kinds of analyses. After you enter this subfunction, IRREP of all occupied orbitals will be shown, the "N_orb" denotes the number of occupied orbitals belonging to the corresponding IRREP. If you want to discard contribution of some IRREPs in the succeeding analyses, you can select option 1 and input the index of the IRREPs, then the occupation number of corresponding orbitals will be set to zero and thus their contributions are eliminated, and you will also see their status are changed from "Normal" to "discarded". Then you can choose option 0 to save wavefunction and quit. You can also choose option 2 to recover the original occupation number of all orbitals, or choose 3 to reverse the status of every IRREP between "Normal" and "Discarded". Note that this subfunction only works for .molden and .gms files, because only these files record orbital IRREPs (beware that the IRREPs in the .molden file produced by many quantum chemistry programs are missing or incorrect). Besides, only restricted and unrestricted SCF wavefunctions are supported.

Subfunction 36: Invert phase of some orbitals (i.e. replacing expansion coefficients of basis functions and GTFs of these orbitals with their negative values).

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

3.9 Population analysis and atomic charges (7)

This module is used to calculate population number of basis functions, shells, atoms, fragments, or atomic charges. There are some points should be noted:

Regardless of which subfunction you choose, if you want to evaluate fragment charge (the sum of charge of the atoms in a fragment), you should select option "-1 Define fragment" first and input atom indices to define the fragment. Then once the calculation of atomic charges is finished, fragment charge will be printed along with atomic charges.

After atomic charge calculation, you can choose to output atomic coordinates with calculated charges to *[name of loaded file].chg* file in current directory, see Section 2.5 for detail about .chg format.

If the file loaded when Multiwfn boots up is .chg format, you can use option "-2 Calculate interaction energy between fragments based on atomic charges" in present function to evaluate interfragment electrostatic interaction energy between two given fragments by classic Coulomb formula (see below) using the atomic charges in this .chg file.

$$E_{\text{Coul-int}} = \sum_{A \in \text{frag 1}} \sum_{B \in \text{frag 2}} \frac{q_A q_B}{|\mathbf{R}_A - \mathbf{R}_B|}$$

This feature is useful for studying electrostatic interaction component in total molecular interaction energy, as well as evaluating intermolecular exciton coupling energy based on atomic transition charges (see Section 4.A.9 on how to derive them).

3.9.1 Hirshfeld atom population (1)

Hirshfeld is a very popular atomic population method based on deformation density partition, Hirshfeld charge is defined as (*Theor. Chim. Acta (Berl.)*, **44**, 129 (1977))

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

where

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

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 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.

After calculation, if the printed sum of all Hirshfeld charges is very close to integer, that means the quadrature is accurate; if not, that 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" are 75 and 434, respectively, you can set them to 100 and 590 respectively to obtain more accurate results.

An example is given in Section 4.7.1.

Information needed: GTFs, atom coordinates

3.9.2 Voronoi deformation density (VDD) atom 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, each cell corresponds to an atom, see *J. Comput. Chem.*, **25**, 189 (2004) for details.

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.

Note that if the sum of VDD atomic charge deviates from molecular net charge evidently, that means the numerical integration accuracy is not satisfactory, and hence you should enlarge "sphpot" parameter in settings.ini to improve the result. "radpot" parameter also influences the result, but not so significantly as "sphpot".

Information needed: GTFs, atom coordinates

3.9.3 Mulliken atom & basis function population analysis (5)

Theory

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

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

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 .

Mulliken analysis is not ideal for practical application due to these shortcomings:

1. Poor reproducibility of observable properties such as molecular dipole moment
2. The “equal partition” of cross term has no strict physical meaning
3. Very high basis set dependence. In particular, **diffuse function must not be presented, otherwise the result will be misleading.**
4. Occasionally, meaningless result occurs (population number is negative).

Usage

Since amount of outputted information is huge for large systems, before selecting below subfunctions, you can use option -1 to change the default output destination from screen to a specific plain text file.

- **Subfunction 1** (Output Mulliken population and atomic charges): Population of basis functions, population of basis function shells, population of each angular moment of atomic orbitals,

as well as atomic charges are printed.

- **Subfunction 2** (Output gross atomic population matrix and decompose it): 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

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

Note 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 the matrix to contribution of each occupied orbital, the Ω^i matrices will be outputted to *groatmdcp.txt* in current directory.

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.

- **Subfunction 3** (Output gross basis function population matrix and decompose it): Gross basis function population matrix can be outputted by this option for further analyzing detail of electron distribution, 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 the outputted matrix is total population number of corresponding basis function. You can also select to output contribution of each occupied orbital, namely $\Gamma_{a,b}^i$, to *grobasdcp.txt* in current directory (notice that this file is extremely large even for medium-sized system).

- **Subfunction 4** (Output orbital contributions to atomic populations to *atmpopdcp.txt*): This option is used to output contributions of occupied orbitals to atomic populations, namely all $\sum_i \eta_i \sum_{a \in A} \Theta_{i,a}$ terms, to *atmpopdcp.txt* in current folder.

An example of Mulliken population analysis is given in Section 4.7.0.

Information needed: Basis functions

3.9.4 Löwdin atom & basis function population (6)

The only difference between Löwdin and Mulliken population analysis is whether Löwdin orthogonalization is performed first. In Löwdin orthogonalization, the linear transformation matrix is

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

where U is eigenvector matrix of overlap matrix, matrix s is $\text{diag}(\lambda_1, \lambda_2, \dots)$, where $\{\lambda\}$ is eigenvalue set of overlap matrix. After Löwdin orthogonalization, the overlap matrix becomes identity matrix,

and new coefficient matrix is $\mathbf{X}^{-1}\mathbf{C}$.

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 Mulliken. 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 Gaussian basis functions are used, however the dependency can be safely ignored in generally, for detail please see *Chem. Phys. Lett.*, **393**, 209 (1968) and *Int. J. Quantum Chem.*, **106**, 2065 (2006).

Information needed: Basis functions

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

Some people had proposed several different partition methods of cross term to improve Mulliken analysis, they are generally called as modified Mulliken population analysis (MMPA). In the method proposed by Ros and Schuit (*Theo. Chim. Acta*, **4**, 1 (1966)), the composition of basis function a in orbital i is defined as

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

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 Mulliken charges by using the newly defined Θ . Relative to Mulliken population, the advantage of SCPA is that negative value of population number never occurs.

Information needed: Basis functions

3.9.6 Modified Mulliken atom 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 Mulliken, 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).

Information needed: Basis functions

3.9.7 Modified Mulliken atom 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 .

Information needed: Basis functions

3.9.8 Becke atomic charge with atomic dipole moment correction (10)

In the paper *J. Chem. Phys.*, **88**, 2547 (1988), Becke proposed a weighting function for converting whole space integral to multiple single-center spherical integrals, although the weighting function is not intend for population analysis, Multiwfn still makes an attempt to use this weighting function as atomic space to obtain atomic charges. The Becke charge can be defined as

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

The radii used for evaluating the Becke weighting function (or say Becke atomic space) can be controlled by uses, see corresponding options shown on screen. For details about Becke weighting function please see *J. Chem. Phys.*, **88**, 2547 or Section 3.18.0. According to my experiences, by using the default "modified CSD" radii, Becke charge is reasonable for typical organic systems, but not very appropriate for ionic systems. For detail about "modified CSD" radii, see the end of Section 3.18.0.

After the Becke charges are calculated, atomic dipole moment correction will be performed automatically. The correction process is identical to the one used for correcting Hirshfeld charge (see next section). After the correction the charges will have better electrostatic potential reproducibility and can exactly reproduce molecular dipole moment.

Information needed: GTFs, atom coordinates

3.9.9 Atomic dipole moment corrected Hirshfeld atomic charges (ADCH, 11)

Basic characteristic and Usage

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 the ADCH method proposed by me ([J. Theor. Comput. Chem., 11, 163 \(2012\)](#)), atomic dipole moment of each atom is expanded to correction charges placed at neighbour atoms, then ADCH charge is just the sum of original Hirshfeld charge and correction charge. 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 obtained from fitting ESP. Compared to another method that try to improve Hirshfeld charges, namely Hirshfeld-I (see Section 3.9.13), the computational cost of ADCH charge is negligible. 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 \(2012\)](#).

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 the difference between molecular dipole moment produced by ADCH charges and the one produced by actual electron density, “Error” is always equals to or very close to zero, because ADCH charges in principle exactly reproduce molecular dipole moment.

If you would like to obtain the detail of charge transfer between atoms in the atomic dipole moment correction process, you can set “ishowchgtrans” in settings.ini to 1.

An example of calculating ADCH charges is given in Section 4.7.2.

Algorithm detail about expanding atomic dipole moment

Full details of ADCH charge can be found from its original paper, here I only mention how the atomic dipole moment correction (ADC) is realized.

Assume that we want to expand atomic dipole moment of atom *A* as correction charges, clearly,

the sum of all correction charges should be zero, and the dipole moment evaluated based on the correction charges should be exactly equal to atomic dipole moment, namely below conditions should be satisfied

$$\sum_B \Delta q_{A \rightarrow B} = 0$$

$$\mathbf{\mu}_A = \sum_B \Delta q_{A \rightarrow B} \mathbf{R}_B$$

where $\Delta q_{A \rightarrow B}$ is transferred charge from atom A to B , in other words, it is the correction charge on atom B due to A . The index B cycles all atoms in the system. \mathbf{R}_B is relative coordinate (column vector) of atom B with respect to atom A

The correction charges are expected to be distributed only around atoms neighbouring to A , this could be realized by minimizing function F :

$$F = \sum_B \frac{(\Delta q_{A \rightarrow B})^2}{v_{AB}} + \alpha \sum_B \Delta q_{A \rightarrow B} + \beta (\mathbf{\mu}_A - \sum_B \Delta q_{A \rightarrow B} \mathbf{R}_B)$$

where α and β are Lagrangian multipliers used to satisfy the two constraint conditions, the v_{AB} is a function decreases rapidly as increase of distance between A and B , and its detailed form is dependent of atomic radii of A and B , see original paper of ADCH for more information. Obviously, this design of F suppresses occurrence of large correction charge on the atoms far away from atom A .

It is shown that after some manipulations, the working equation of evaluating $\Delta q_{A \rightarrow B}$ is

$$\Delta q_{A \rightarrow B} = \frac{v_{AB}}{\sum_B v_{AB}} [(\mathbf{R}_B - \langle \mathbf{R}_B \rangle)^T \cdot \mathbf{\Lambda}^{-1} \cdot \mathbf{\mu}_A]$$

where superscript "T" is sign of transpose, and

$$\mathbf{\Lambda} = \langle \mathbf{R}_B \cdot (\mathbf{R}_B)^T \rangle - \langle \mathbf{R}_B \rangle \langle \mathbf{R}_B \rangle^T$$

$$\langle \mathbf{R}_B \rangle = \frac{\sum_B v_{AB} \mathbf{R}_B}{\sum_B v_{AB}}, \quad \langle \mathbf{R}_B \cdot (\mathbf{R}_B)^T \rangle = \frac{\sum_B v_{AB} \mathbf{R}_B \cdot (\mathbf{R}_B)^T}{\sum_B v_{AB}}$$

If the atom A is in local planar region, the $\mathbf{\Lambda}$ matrix will be exactly or almost singular matrix, in this case inversed matrix of $\mathbf{\Lambda}$ obviously cannot be obtained. Notice that the solution to this problem in current implementation is slightly different to the one introduced in the ADCH original paper. In current Multiwfn, the $\mathbf{\Lambda}$ matrix is first diagonalized, the eigenvalues with absolute value less than 10^{-5} will be simply set to zero (the corresponding eigenvector typically perpendicular to the local plane of atom A), and remaining eigenvalues are inverted, now this matrix has corresponded to the $\mathbf{\Lambda}^{-1}$ in the new local coordinate. Then $\mathbf{R}_B - \langle \mathbf{R}_B \rangle$ and $\mathbf{\mu}_A$ are transformed to the new local coordinate by doing left multiplication of transpose of eigenvector matrix of $\mathbf{\Lambda}$ on them. Finally, the $\Delta q_{A \rightarrow B}$ is obtained using above formula.

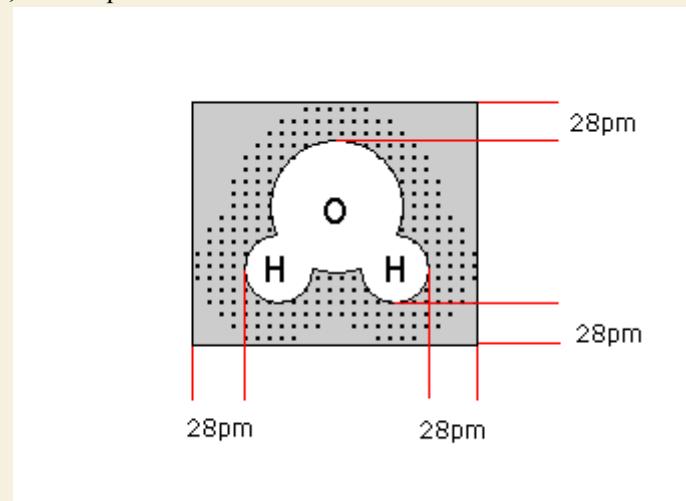
Information needed: GTFs, atom coordinates

3.9.10 CHELPG (Charges from electrostatic potentials using a grid based method) ESP fitting atomic charge (12)

Theory

CHELPG (*J. Comput. Chem.*, **11**, 361 (1990)) is one of most widely used electrostatic potentials (ESP) fitting charge models. Compared to CHELP and Merz-Kollman methods, CHELPG charges have better 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 28 pm, 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 based on wavefunction (V) as good 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 coordinate of fitting point i . $\{q\}$ are fitted point charges, their positions are referred to as fitting centers. Notice that q does not necessarily correspond to atomic charge, the fitting centers can be defined at arbitrary positions.

It can be shown that the minimization of F could be formulated to below matrix equation

$$\mathbf{A}\mathbf{q} = \mathbf{B}$$

which can be further explicitly written as

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} & 1 \\ A_{21} & A_{22} & \dots & A_{2N} & 1 \\ \vdots & \vdots & \ddots & \vdots & 1 \\ A_{N1} & A_{N2} & \dots & A_{NN} & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \lambda \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \\ q_{\text{tot}} \end{bmatrix}$$

with

$$\mathbf{A}_{A,B} = \sum_i \frac{1}{r_{iA} r_{iB}} \quad \mathbf{B}_A = \sum_i \frac{V(\mathbf{r}_i)}{r_{iA}} \quad q_{\text{tot}} = \sum_A q_A$$

where A and B are indices of fitting centers and N is total number of fitting centers. The column vector \mathbf{q} is what we need, its first N elements correspond to charges of the fitting centers. The \mathbf{q} could be easily evaluated as $\mathbf{q} = \mathbf{A}^{-1} \mathbf{B}$.

Usage

In the interface there are many options, which are introduced below.

- Option 1: If selecting this option, Multiwfn will start to calculate ESP value at each fitting point, then fitted atomic charges of all fitting centers will be outputted on screen. RMSE and RRMSE are also outputted automatically, they measure quality of fitting, the smaller value suggests that the fitted charges have better ESP reproducibility. RMSE and RRMSE are defined as (where N is the number of fitting points)

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

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

- Options 2~4: These options are used to set parameters for distributing fitting points. The default values are reasonable and should not be changed without special reasons.

- Option 5: During the ESP fitting, three kinds of ESP can be calculated and used, namely "Nuclear + Electronic" (default), "Electronic" and "Transition electronic". Commonly you should not change this option. The "Transition electronic" should be chosen if you want to evaluate the so-called TrEsp (transition charge from electrostatic potential), please check Section 4.A.9 for details.

- Option 6: If you have chosen this option once to switch its status to "Yes", then after calculation, coordinates with ESP values of all fitting points can be exported to *ESPfitpt.txt* or *ESPfitpt.pqr* in current folder, the former is more readable, while the latter can be directly loaded into VMD program to visualize the fitting points (and can be colored according to the "charge" column, which records ESP values). In addition, fitting points with absolute difference between exact ESP and the ESP evaluated based on atomic charges can be exported to *ESPerr.pqr* so that ESP reproduction error in various molecular regions can be visualized in VMD. See Section 4.7.8 for illustration.

- Option 10: Choose the atomic radii used in fitting. The atomic radii employed in fitting affects the distribution of fitting points, and thus influence the resulting charges. There are three modes can be chosen to set the radii:

(1) This is default mode, the radii defined in original paper of CHELPG are used. However, only the elements in the first three rows are defined, if your system contains other elements, Multiwfn will ask you to input their vdW radii in turn. If you do not have proper radii in hand, you can directly press ENTER button, then vdW radius of UFF forcefield of corresponding element multiplied by 1/1.2 will be used, which is commonly a reasonable choice.

(2) Employing the UFF radii scaled by 1/1.2 for all elements. The UFF radii can be find in Table 1 of UFF original paper *J. Am. Chem. Soc.*, **114**, 10024 (1992).

(3) Load radii of all elements involved in present system from external file. Multiwfn will ask you to input the file path during fitting, the format of the file should like follows, the unit of the radii should be in Å, all elements in the system must be defined:

```
H 1.2
O 1.8
Cu 2.2
```

- Option -1: For flexibility consideration, by using this option, coordinates of fitting points are allowed to be read from external file to replace the CHELPG fitting points. The format of the file should be

<i>numdata</i>	
<i>X Y Z [ESPval]</i>	← For fitting point 1
<i>X Y Z [ESPval]</i>	← For fitting point 2
...	

where 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 read from the fourth column rather than calculated by Multiwfn.

- Option -2: If you have special reasons (*e.g.* you want to place additional fitting centers at lone pair or σ -hole region to enhance description of ESP around corresponding region), then coordinates of additional fitting centers can be read from external file by using this option. The format of the file should be

<i>numdata</i>	
<i>X Y Z</i>	← For additional fitting center 1
<i>X Y Z</i>	← For additional fitting center 2
...	

where *numdata* denotes how many entries are in this file. X, Y, Z are coordinates (in Bohr).

- Option -3: If you simply want to examine reproducibility for ESP at the fitting points of given atomic charges, then you can use this option to load atomic charges from specific .chg file. Then when you select option 1 to start the ESP fitting process, no ESP fitting charges will be yielded, only the RMSE and RRMSE of the given atomic charges will be outputted. (If you have chosen option -2 to load additional fitting centers, then the number of charges in the loaded .chg file should be identical to the total number of fitting centers)

An example is given in Section 4.7.1.

Information needed: GTFs, atom coordinates

3.9.11 Merz-Kollmann (MK) ESP fitting atomic charge (13)

Merz-Kollmann (MK) charge is another well known charge model derived from ESP fitting, see *J. Comput. Chem.*, **11**, 431 (1990). 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, if distance between a fitting point and any atom is smaller than 1.4 times of its

vdW radius, then this fitting points will be discarded.

In the MK module of Multiwfn, the density of points per Å² on the MK layers can be set by option 2, the number of layers and the scale factor of atomic vdW radii used to define the layers can be set by option 3, the atoms used to construct the MK fittings points can be set by option 4. For description of other options and outputs, see last section.

Like the CHELPG module, in the MK module you can also choose the mode for determining atomic radii used in the fitting by option 10. Notice that the original paper of MK method does not explicitly present atomic radii. In Multiwfn, the default MK radii for the first and second rows and P, S, Cl are in line with those in MK code of Gaussian, while the default radii for Na~Si are defined as 1.57, 1.65, 1.65, 1.80 Å, respectively.

In the example in Section 4.7.8, I exemplified how to use the MK module to investigate ESP reproducibility of given atom charges on all MK fitting points and on the fitting points corresponding to specific atoms. In addition, the example showed how to visualize ESP reproduction error at various molecular regions.

Information needed: GTFs, atom coordinates

3.9.12 AIM atomic charge (14)

AIM (Atoms in molecules) population is the number of electrons in AIM basin, and accordingly, nuclear charge minus AIM population yields AIM charge, which is also known as Bader charge. AIM charges can be calculated in basin analysis module, please check the example given in 4.17.1 on how to do this. Related theories and algorithms of basin analysis module are introduced in Section 3.20.

Information needed: GTFs, atom coordinates

3.9.13 Hirshfeld-I atom population (15)

Hirshfeld-I (HI) method was proposed in *J. Chem. Phys.*, **126**, 144111 (2007), it is an important extension of Hirshfeld method. It is believed that the atomic space defined by Hirshfeld is not quite ideal, because it does not respond actual molecular environment. In HI, atomic spaces are gradually refined via an iterative scheme. After convergence, the final HI atomic spaces are evidently more physically meaningful than the Hirshfeld ones.

There are some variants of HI method, including the Hirshfeld-E proposed in *J. Chem. Theory Comput.*, **9**, 2221 (2013), the Hirshfeld- λ proposed in *J. Comput. Chem.*, **32**, 1561 (2011) and the fractional occupation Hirshfeld-I (FOHI) proposed in *J. Chem. Theory Comput.*, **7**, 1328 (2011). Also, there is an iterative atomic space method name iterated stockholder atoms (ISA), which is closely related to HI, see *Chem. Commun.*, **2008**, 5909. However, since in my personal viewpoint these variants and ISA are not quite important, I do not intend to implement them in Multiwfn.

Basic theory of HI method

In HI method, the weighting function of atom A at iteration n is defined as

$$w_A^{(n)}(\mathbf{r} - \mathbf{R}_A) = \frac{\rho_A^{(n-1)}(\mathbf{r} - \mathbf{R}_A)}{\sum_B \rho_B^{(n-1)}(\mathbf{r} - \mathbf{R}_B)} \equiv \frac{\rho_A^{(n-1)}(\mathbf{r} - \mathbf{R}_A)}{\rho_{\text{pro}}^{(n-1)}(\mathbf{r})}$$

where \mathbf{r} and \mathbf{R} denote spatial coordinate and position of atomic nucleus, respectively. The atomic density involved in HI is obtained by linear interpolation between spherically averaged free-state atomic density of adjacent charge states:

$$\rho_A^{(n)}(r) = (q_{\text{high}} - q_A^{(n-1)})\rho_{A,\text{low}}^{\text{free}}(r) + (q_A^{(n-1)} - q_{\text{low}})\rho_{A,\text{high}}^{\text{free}}(r)$$

where q_{high} and q_{low} are ceiling and floor integer of atomic charge of atom A at $n-1$ iteration, while $\rho_{A,\text{high}}^{\text{free}}$ and $\rho_{A,\text{low}}^{\text{free}}$ are spherically averaged atomic density of atom A at these two charge states, respectively. For example, if atom A carries 0.2 charge at $n-1$ step, then its radial atomic density at step n will be computed as

$$\rho_A^{(n)}(r) = 0.8 \times \rho_{A,q=0}^{\text{free}}(r) + 0.2 \times \rho_{A,q=+1}^{\text{free}}(r)$$

Before the HI iteration, radial density of all atoms are initialized to their neutral state.

Based on HI atomic weighting functions, HI atomic charges can be straightforwardly obtained. The HI iteration continues until all atomic charges converged to a given criterion.

Note: It is clear that the HI atomic charges yielded at the first iteration are in principle identical to Hirshfeld charges. However, this is not exactly true in Multiwfn, because currently the rule of generating spherically averaged free-state atomic density involved in HI is not completely identical to that used in Hirshfeld, but the difference is very small and can be ignored. It is never incorrect to directly take the atomic charges printed at the first iteration of HI as Hirshfeld charges.

Usage

After you entered the HI analysis module, you can adjust convergence criterion and the maximum number of cycles, also you can switch the algorithm for realizing HI iterations (see below), as well as choose if printing atomic charges every iteration. If you select option 0, Multiwfn will first try to yield atomic radial density file (.rad) for all elements in the systems by invoking Gaussian, and then start the HI iteration. After convergence, HI atomic charges and fragment charge (if you have defined fragment) are printed. Note that the printed HI atomic charges have been properly normalized (i.e. making sum of HI population of all atoms equal to total number of electrons) to eliminate noise of numerical integration. Finally, you can select if outputting resulting atomic charges as .chg file.

Procedure of preparing atomic radial density files

Before igniting HI iteration, atomic radial density files (.rad files) must be first prepared. The steps are detailed below:

(1) **Generating atomic .wfn files in "atmrad" folder:** Multiwfn generates Gaussian input files (.gjf) in "atmrad" subdirectory of current folder and invokes Gaussian to run them to generate atomic .wfn files for all elements involved in present system. The charge states range from -2 to +2 are taken into account. Calculation of useless charge states are skipped. For example, it's meaningless to calculate -2 and +2 charge states for alkali elements, since in practical systems it is impossible that an alkali atom has charge between -1 to -2, or between +1 to +2. The multiplicity of each charge state of each element is set to ground state (this cannot be well guaranteed for all cases, for example,

ground state at different calculation level is even different. However, this is never an important problem, so please do not concern it too much). The path of Gaussian executable file should be set by "gaupath" in settings.ini file, if it is not properly set, you will be prompted to input it in the Multiwfn interface. You also need to input the keywords of Gaussian used to calculate the atomic .wfn files. After Gaussian calculation finished, the atomic .wfn files are generated in "atmrad" folder, the file name directly corresponds to element name and charge state. For example, the file corresponding to -1, 0, +1 and +2 charge states of Be will be generated in "atmrad" folder as "Be-1.wfn", "Be_0.wfn", "Be+1.wfn" and "Be+2.wfn", respectively. If corresponding file has already been found in "atmrad" folder, then the file will be retained and not be recalculated. If you find Gaussian calculation is failed (mostly due to SCF unconvvergence problem), you should carefully check Gaussian output file and properly adjust the keyword used.

(2) **Converting atomic .wfn files to .rad files:** As mentioned earlier, HI calculation requires spherically averaged atomic densities. However, the electron density corresponding to the atomic .wfn files generated at last step often does not meet this requirement. For example, carbon at its neutral ground state has s^2p^2 configuration, and thus "C_0.wfn" corresponds to elliptical density distribution. To get spherically averaged representation of atomic densities, Multiwfn automatically loads each atomic .wfn files in "atmrad" folder, calculates spherically averaged radial density, and write the data as .rad file, whose name is identical to its parental .wfn file. i.e. "C_0.wfn" will be converted to "C_0.rad". The .rad file is plain text file, the first line is the number of data points, the first and second columns respectively correspond to radial distance with respect to nucleus (in Bohr) and corresponding electron density. Note that if needed .rad file is already presented in "atmrad" folder, then the conversion of .wfn→.rad will be skipped.

It is suggested (but not compulsory) that the atomic .wfn files are generated at the same calculation level as the molecule under study, so that the result has good physical meaning. You should always make sure that the atomic .wfn and .rad files in "atmrad" folder were generated at expected level. For example, in the "atmrad" folder if some .wfn/.rad files were generated at B3LYP/cc-pVDZ level, and some were generated at M06-2X/def2-TZVP level, while the molecular wavefunction file was obtained at PBE/6-311G** level, then the resulting HI charges must be problematic. If you want to let Multiwfn regenerate all needed atomic .wfn/.rad files at given level during HI analysis, simply cleaning up the "atmrad" folder before calculation.

Note: If you are using Windows version of Multiwfn, while Gaussian cannot be invoked properly, please read Appendix 1 to set environment variable for Gaussian.

Two numerical algorithms of HI

Multiwfn provides below two algorithms to realize HI methods, the results are exactly the same, they only differ by efficiency and memory requirement.

(1) Fast & large memory requirement (default): This algorithm computes as much as possible data before starting iteration. Therefore, once initialization stage is done, the HI iteration can be finished rapidly. The drawback of this algorithm is that large amount of memory is needed, the memory consumed is at least $5 \times n_{\text{atm}} \times n_{\text{atm}} \times n_{\text{rad}} \times n_{\text{ang}}$, where n_{atm} is the number of total atoms, n_{rad} and n_{ang} corresponds to the number of radial and angular integration grid per atom. Evidently, this algorithm is unable to be applied for very large systems unless you have huge physical memory.

(2) Slow & low memory requirement: This algorithm utilizes very low amount of memory, however, the data needed to be calculated in each iteration is much more than algorithm (1), thus the total computational cost is much higher.

For both the two algorithms, the higher number the integration grid, the more accurate the result. By default n_{rad} and n_{ang} are properly set by Multiwfn. If the system only consists of first two rows elements, then $n_{\text{rad}} \times n_{\text{sph}}$ will be $30 \times 170 = 5100$. If you want to manually set n_{rad} and n_{sph} , you can set "iautointgrid" in settings.ini to 0 and change "radpot" and "sphpot" parameters.

An example of computing HI charges is given in Section 4.7.4.

Information needed: GTFs, atom coordinates

3.9.14 CM5 atomic charge (16)

CM5 charge was proposed by Truhlar et al. in *J. Chem. Theory Comput.*, **8**, 527 (2012). This charge is somewhat akin to the ADCH charge proposed by me, both of them are defined as applying additional correction to Hirshfeld charges. Unlike ADCH, which is free of empirical parameters, CM5 method contains global parameters as well as parameters for individual elements. The CM5 parameters are optimized for best reproduction of highly accurate experimental or theoretical molecular dipole moment.

Worthnotingly, there is a remarkable difference between the basic feature of ADCH and CM5: ADCH charges can exactly reproduce molecular dipole moment corresponding to present calculation level, that means if the level used is very high (e.g. CCSD/aug-cc-pVTZ), then the dipole moment calculated by the resulting ADCH charge must be very close to the molecular dipole moment in real world; while if the used level is poor (e.g. HF/6-31G), then the ADCH charges will be almost useless. In contrast, the CM5 charges do not attempt to reproduce molecular dipole moment at present calculation level, but to reproduce real molecular dipole moment, therefore even if low level such as B3LYP/6-31G* is used, which is certainly unable to give good dipole moment result, the CM5 charges derived at this level commonly are still able to yield molecular dipole moment at acceptable accuracy.

The expression of CM5 charge is

$$q_i^{\text{CM5}} = q_i^{\text{Hirsh}} + \sum_{j \neq i} T_{ij} B_{ij}$$

$$B_{ij} = \exp[-\alpha(r_{ij} - R_i - R_j)]$$

where r_{ij} is distance between atom i and j , B_{ij} may be regarded as their Pauling bond order, R_i and R_j are their atomic covalent radii, which are defined as follows: For $Z=1\sim 96$, the average between CSD radii and Pyykkö radii are used, while for $Z=97\sim 118$, the Pyykkö radii are employed. The global parameter α equals to 2.474 \AA^{-1} . The T_{ij} is defined as D_{ij} if both i and j are attributed to H, C, O, N, note that $D_{ij}=0$ when i and j belong to the same element and $D_{ij}=-D_{ji}$. All the involved six D_{ij} parameters (H-C, H-N, H-O, C-N, C-O, N-O) are tabulated in the original paper. For other cases, T_{ij} is defined as $D_i - D_j$, the optimized D parameters for all elements through out the whole periodic table are provided in the supplemental material of CM5 original paper.

Like usual Hirshfeld and ADCH calculations, after you enter this function, I suggest you select option 1 to use the build-in sphericalized atomic densities in free-states, since it is the most convenient. Then Multiwfn starts calculation of Hirshfeld charges, and then print CM5 charges. If you want to gain detailed information about the CM5 correction process during the calculation, you

can set "ishowchgtrans" parameter in settings.ini to 1.

Information needed: GTFs, atom coordinates

3.9.15 Electronegativity Equalization Method (EEM) atomic charge (17)

Theory

The EEM method is also known as Full Equalization of Orbital Electronegativity (FEOE), it was firstly proposed in *J. Am. Chem. Soc.*, **107**, 829 (1985). The idea of EEM is very clear. The electronegativity of atom i can be written as

$$\chi_i = (\chi_i^0 + \Delta\chi_i) + 2(\eta_i^0 + \Delta\eta_i)q_i + \sum_{j \neq i} \frac{q_j}{r_{i,j}}$$

where χ_i^0 and η_i^0 are Sanderson electronegativity and Parr-Pearson hardness of corresponding element, respectively, while $\Delta\chi_i$ and $\Delta\eta_i$ are fitted parameters. q_i denotes atomic charge of atom i , and $r_{i,j}$ is distance between atoms i and j . For convenience, above formula is commonly expressed as

$$\chi_i = A_i + B_i q_i + \kappa \sum_{j \neq i} \frac{q_j}{r_{i,j}}$$

where κ is global parameter, A and B are element parameters. It can be seen that, atomic electronegativity in a chemical system is function of atomic charges.

EEM method assumes that in a chemical system, all atoms have equal electronegativity, hence the EEM charges can be determined by solving linear equations according to below conditions

$$\chi_1 = \chi_2 = \dots = \chi_N = \chi_{\text{mol}}$$

$$\sum_i q_i = Q$$

where Q stands for net charge of the whole system, χ_{mol} is molecular electronegativity

It is easy to show that the working matrix equation for solving EEM charges can be written as follows

$$\begin{bmatrix} B_1 & \frac{\kappa}{r_{1,2}} & \dots & \frac{\kappa}{r_{1,N}} & -1 \\ \frac{\kappa}{r_{2,1}} & B_2 & \dots & \frac{\kappa}{r_{2,N}} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\kappa}{r_{N,1}} & \frac{\kappa}{r_{N,2}} & \dots & B_N & -1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \chi_{\text{mol}} \end{bmatrix} = \begin{bmatrix} -A_1 \\ -A_2 \\ \vdots \\ -A_N \\ Q \end{bmatrix}$$

Since EEM matrix element is very simple and the above matrix equation can be solved easily, EEM charges could be evaluated rather rapidly even for very large systems.

Parameters

The result of EEM charges are directly dependent of EEM parameters κ , $\{A\}$ and $\{B\}$. There is no unique way to determine the parameters, the most common way to obtain them is fitting, so

that the resulting EEM charges are close to quantum chemistry atomic charges (e.g. NPA, CHELPG, Mulliken) as much as possible. Many papers presented EEM parameters fitted to various kinds of atomic charges calculated at different levels. In Multiwfn, below EEM parameters can be directly chosen:

- Parameters fitted to Mulliken charges at HF/STO-3G, see *Int. J. Mol. Sci.*, **8**, 572-582 (2007). Available for H, C, N, O, S
- Parameters fitted to CHELPG charges at B3LYP/6-31G* and at HF/6-31G*, see *J. Comput. Chem.*, **30**, 1174 (2009). Available for H, C, N, O, F, S, Cl, Br, Zn
- Parameters fitted to NPA charges at B3LYP/6-311G*, they were extracted from SI of *J. Cheminform.*, **8**, 57 (2016). Available for H, C, N, O, F, P, S, Cl, Br

Note that for some elements, their A and B parameters also depend on multiplicity, namely the maximal formal bond order between this atom and its neighboring atoms, so that influence of chemical environment can be taken into account.

With above mentioned parameters, EEM charges have good reproducibility of target atomic charges for typical organic systems, but do not expect EEM method can work well for systems with complicated electronic structure, since common training set of EEM parameters only include organic systems with typical bonding.

Usage

Since formal bond order is involved in the EEM calculation, you must use MDL molfile (.mol) as input file, because in all file types that supported by Multiwfn, only this file contains connectivity information between atoms. The .mol file can be outputted by many programs, such as GaussView and OpenBabel. OpenBabel is recommended to use, since the format of .mol file outputted by OpenBabel is very standard. Do not use GaussView to generate .mol file if dashed bond is presented (e.g. for benzene, each C-C bond is described by default as a single bond with a dashed bond, the .mol file is unable to record such non-Lewis representation of bonding).

After you load a .mol file and then enter present function, you can directly choose option 0 to calculate EEM charges, the molecular electronegativity will also be outputted together. The default EEM parameters are those fitted for reproducing B3LYP/6-31G* CHELPG charges.

Before calculation, you can choose option "1 Choose EEM parameters", present EEM parameters will be shown on screen, and then you can select a built-in parameter set that you want to use. Alternatively, you can load parameters from external file using suboption 0, the format of parameter file should mimic to this:

```
0.302000
H 1 2.38500 0.73700
C 1 2.48200 0.46400
C 2 2.46400 0.39200
N 1 2.59500 0.46800
N 2 2.55600 0.37700
O 1 2.82500 0.84400
O 2 2.78900 0.83400
```

The first line is κ , after that, defining parameter for each element of each multiplicity. The second, third and fourth columns are multiplicity, A and B , respectively. Free format is used. Using corresponding option, the present parameters can also be exported to EEMparm.txt in current folder.

If the system under study is an ionic system, do not forget to use option "2 Set net charge" to set net charge to actual status before calculation!

Notice for GaussView users: The .mol file generated by GaussView, at least for version 6.0.16, is not completely correct when there are aromatic rings. In the connectivity field of .mol file, the aromatic ring should be represented as alternate variation of single and double bonds, however in the .mol generated by GaussView, the multiplicity of the bonds in aromatic rings is recorded as 4, clearly it is unreasonable. This issue could be fixed if you use OpenBabel to convert the .mol to a new .mol file.

An example of calculating EEM charges is given in Section 4.7.5.

Information needed: Atom coordinates with connectivity (.mol file)

3.9.16 Restrained ElectroStatic Potential (RESP) atomic charge (18)

This module is quite powerful and flexible, it can calculate the standard RESP charge proposed by Kollman and coworkers in *J. Phys. Chem.*, **97**, 10269 (1993), and can also calculate ESP fitting charges under various extra constraint conditions. Below I first describe basic ideas of RESP charge and related special treatments, and then introduce usage of this module. If you are not interested in theory, you can directly jump to Section 3.9.16.2. However, if you have not read Section 3.9.10, you should read it first to gain basic knowledges about ESP fitting procedure. Some more discussions about this topic can be found in my blog article "Principle of RESP charge and its calculation in Multiwfn" (in Chinese, <http://sobereva.com/441>).

3.9.16.1 Theory

Theory-Part 1: Multiple conformations, equivalence constraint and penalty function

The ESP fitting charges of MK and CHELPG types introduced in Section 3.19.11 and 3.19.12 can be perfectly used for molecular modeling purpose of rigid molecules, however, they are not suitable for modeling flexible molecules, due to below reasons:

(1) The results are highly dependent on conformation. Flexible molecules have many different conformations, the conformation often changes during molecular dynamics simulation, while the ESP fitting charges are highly sensitive to the conformation. If only one conformation is used to calculate the ESP fitting charges and the simulation is conducted based on these charges, the dynamic behavior of the molecule may be wrong and the relative energies between different conformations may be seriously incorrect, since single set of fixed atomic charges is unable to equitably well describe all relevant conformations.

(2) The atomic charges fitted at single conformation do not faithfully reflect chemical equivalency of atoms. For example, three hydrogens on the methyl group of methanol are chemically equivalent. During molecular dynamic simulation in room temperature, the methyl group can rotate frequently, therefore, three hydrogens should share the same charge. However the calculated charges are not identical regardless of choice of conformation used in the ESP fitting procedure (since this system does not have triple rotational symmetry along the methyl bond axis), clearly this problem also brings some irrationality to the simulation phenomenon.

(3) The quality of fitted charge of buried atoms is poor. The ESP fitting points are distributed in the vicinity of and outside the van der Waals surface. For atoms connected to multiple atoms (such as sp^3 hybridized carbon), especially the atoms inside a macromolecule, because of their far

distances to fitting points, their atomic charges have low fitting quality and large numerical uncertainty. Moreover, as the conformation changes, the charge fluctuation of these buried atoms tends to be significant, therefore the presence of these atoms further aggravate the conformational dependence of the ESP fitting charges.

ESP fitting charges cannot be well used for modeling flexible molecules without solving above mentioned problems.

For the above problem (1), a good solution is to simultaneously consider multiple conformations during the ESP fitting process. One first determines the weight of each conformation, then constructs **A** matrix and **B** vector using fitting points of various conformers with consideration of conformational weights, then the solved ESP fitting charges can at least be able to well reproduce the ESP of those conformations with relatively large weights. This idea has been examined in *J. Am. Chem. Soc.*, **114**, 9075 (1992). Of course, this way of considering multiple conformations can be very expensive for flexible molecules with many rotatable bonds, because the number conformations increases exponentially with the increase in rotatable bonds.

For the above problem (2), equivalence constraints can be imposed on chemically equivalent atoms in the fitting process so that their atomic charges are the same (Another way is to calculate the ESP fitting charges as usual, and then average the charges of chemical equivalent atoms. However, the charge obtained in this way is not as good as employing equivalence constraints).

For the above problem (3), the solution proposed in Kollman's RESP paper is to add a hyperbolic penalty function $\chi_{\text{str}}^2 = a \sum_A [(q_A^2 + b^2)^{1/2} - b]$ to the function of measuring

reproducibility of the ESP calculated based on wavefunction, where index *A* corresponds to atomic index of non-hydrogen atoms. The penalty function involves a tightness parameter *b* and a restraint strength parameter *a*. The former is generally set to 0.1, while the latter can be adjusted in the actual calculation. The larger the *a*, the stronger the tendency of the atomic charge to be pulled down, and meantime the worse the ESP reproducibility becomes. Obviously, the parameter *a* should be properly selected, generally a value less than or equal to 0.001 is employed. It has been found that introduction of this form of penalty function significantly lowers charges of buried atoms, while other atoms, in particular polar atoms, are not evidently affected. Kollman believes that this treatment also significantly reduces the conformational dependence of the ESP fitting charge. After introducing the hyperbolic penalty function, the ESP fitting procedure can no longer be solved in one step, iteration is needed until changes of all atomic charges are small enough.

Below I give detailed derivation of the equation used to calculate ESP fitting charges under special considerations. When equivalence constraint is employed, the function to be minimized using least square method in the ESP fitting procedure will be

$$F = \left[\sum_i \left(V_i - \sum_A q_A \sum_{a \in A} \frac{1}{r_{ia}} \right) \right]^2 + \lambda \left(\sum_A n_A q_A - q_{\text{tot}} \right)$$

where *i* cycles fitting points, V_i is the ESP calculated based on wavefunction at point *i*, $\{q\}$ is the set of uniquely derived atomic charges, r_{ia} denotes distance between point *i* and atom *a*, which belongs to equivalence constraints *A*. n_A is the number of atoms constrained to be equivalent in batch *A*. If $n_A=1$, that means *A* just corresponds to an atom without equivalence constraint.

Minimization of *F* with respect to variables yields

$$\frac{\partial F}{\partial \lambda} = 0 = \sum_A n_A q_A - q_{\text{tot}}$$

$$\frac{\partial F}{\partial q_B} = 0 = -2 \left(\sum_{b \in B} \frac{1}{r_{ib}} \right) \sum_i \left(V_i - \sum_A q_A \sum_{a \in A} \frac{1}{r_{ia}} \right) + n_B \lambda$$

The second equation can be further reorganized as follows, with considering the fact that the value of Lagrangian multiplier λ is arbitrary

$$\frac{\partial F}{\partial q_B} = 0 = \left(\sum_{b \in B} \frac{1}{r_{ib}} \right) \sum_i \left(V_i - \sum_A q_A \sum_{a \in A} \frac{1}{r_{ia}} \right) + \lambda$$

$$\sum_A q_A \sum_{a \in A} \sum_{b \in B} \frac{1}{r_{ia} r_{ib}} + \lambda = \sum_{b \in B} \sum_i \frac{V_i}{r_{ib}}$$

The set of linear equations can be formulated as a matrix equation

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1N} & n_1 \\ A_{21} & A_{22} & \dots & A_{2N} & n_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{N1} & A_{N2} & \dots & A_{NN} & n_N \\ n_1 & n_2 & \dots & n_N & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \lambda \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \\ q_{\text{tot}} \end{bmatrix} \Rightarrow \mathbf{Aq} = \mathbf{B}$$

with

$$A_{A,B} = \sum_{a \in A} \sum_{b \in B} \sum_i \frac{1}{r_{ia} r_{ib}} \quad B_A = \sum_{a \in A} \sum_i \frac{V_i}{r_{ia}} \quad q_{\text{tot}} = \sum_A q_A$$

where A and B are atomic indices, and there are totally N atoms. Once construction of the \mathbf{A} and \mathbf{B} is completed, the charge vector can be easily obtained as $\mathbf{q} = \mathbf{A}^{-1} \mathbf{B}$.

Now we consider the case that penalty function is added to the function F . Given that

$$\partial \chi_{\text{str}}^2 / \partial q_A = a q_A (q_A^2 + b^2)^{-1/2}$$

by incorporating it into the above expression of $\partial F / \partial q_B = 0$, we finally get

$$\sum_A q_A \sum_{a \in A} \sum_{b \in B} \sum_i \frac{1}{r_{ia} r_{ib}} + a q_A (q_A^2 + b^2)^{-1/2} + \lambda = \sum_{b \in B} \sum_i \frac{V_i}{r_{ib}}$$

The diagonal elements of \mathbf{A} should thus be $A_{A,A} = \sum_{a,b \in A} \sum_i \frac{1}{r_{ia} r_{ib}} + a(q_A^2 + b^2)^{-1/2}$, while the non-diagonal terms of \mathbf{A} should keep unchanged. In practical calculation, the $\{q\}$ in initial \mathbf{A} is set to zero, then updated $\{q'\}$ is obtained by solving the matrix equation, after that $\{q'\}$ is used to construct the \mathbf{A} of the second iteration. The iteration is repeated until charge variation of all atoms is smaller than a given threshold.

The easiest way of taking multiple conformations into the ESP fitting procedure is replacing the \mathbf{A} matrix and \mathbf{B} vector with their weighted averaged counterparts, as suggested in *J. Am. Chem. Soc.*, **114**, 9075 (1992):

$$\mathbf{A} = \sum_{\mu} w_{\mu} \mathbf{A}^{\mu} \quad \mathbf{B} = \sum_{\mu} w_{\mu} \mathbf{B}^{\mu}$$

where μ is index of conformer and w_{μ} is corresponding weight. The weights could be determined in different ways, the common way is calculating relative Gibbs free energies and then evaluating weights according to Boltzmann distribution.

Theory-Part 2: The standard RESP charge

The Restrained ElectroStatic Potential (RESP) charge proposed by Kollman *et al.* in *J. Phys. Chem.*, **97**, 10269 (1993) is arguably the most suitable atomic charge module for molecular simulation of flexible small molecules. It takes advantage of many aforementioned ideas and largely solves the above mentioned problems in MK/CHELPG charges. The derivation of the RESP charges is divided into the following two fitting steps.

- Step 1: During the charge fitting, a hyperbolic penalty function with $a=0.0005$ is used to impose a weak charge restraint on non-hydrogen atoms. Charges of all atoms are fitted, and no equivalence constraint is employed. This step allows atomic charges to change with the greatest degree of freedom to make polar atoms fit the ESP as well as possible.
- Step 2: Using the hyperbolic penalty function with $a=0.001$ to impose a strong restraint on non-hydrogen atoms. This step only allows charge of sp^3 hybridized carbons, methylene carbons and hydrogens attached on them to be fitted, while charges of all the other atoms keep fixed at the value obtained at step 1. Equivalence constraint is applied to hydrogens on each $-CH_3$, $=CH_2$, $-CH_2-$ group.

The reason why the RESP charge is divided into two steps is because the authors found that only by doing so, the problems of normal ESP fitting charges in modeling flexible molecules could be largely solved without causing too much damage on the reproducibility of ESP. Since RESP charge is fairly ideal for molecular dynamic modeling purpose, it has been employed by many famous forcefields, such as AMBER, GAFF and GLYCAM.

Notice that although conformation dependency has been diminished to large extent in the subtly designed RESP fitting process, if you want to obtain a set of atomic charges that can equally well describe all important conformations, you still need to explicitly taking multiple conformations in the RESP fitting procedure.

The MK type of fitting points are employed in original paper of RESP, however, changing to CHELPG type fitting points is also completely reasonable.

Theory-Part 3: ESP fitting with charge constraints

When calculating the ESP fitting charges, various constraints can be added via the Lagrangian multiplier method. The most significant one should be constraint on net charge of specific fragments. The charge constraints can achieve many special purposes:

(1) Biomacromolecules, polymers and other systems are all polymerized one by one. Each component of such a macromolecule is called residue. The atomic charges of the whole system are made up of that of individual residues. It is obvious that the net charge of each residue should be an integer. If we want to derive atomic charges of a given residue, we can cap the two terminals of the residue with appropriate group or fragment, and then impose charge constraint on the residue segment during ESP fitting process so that its net charge exactly corresponds to a desired integer.

(2) Some force fields, such as GROMOS, use the charge-group concept to reduce the error of

the electrostatic interaction evaluated via cut-off method. Each charge group contains several atoms, and all the atomic charges are summed to an integer. For example, the total charge of a carboxyl group should be 0, and its charge should become -1 after dissociation of the proton. In order to obtain a set of ESP fitting charges that compatible with the charge group concept, charge constraints can be utilized to maintain the charge of each segment as a specified integer value.

(3) Sometimes one wants to calculate ESP fitting charges based on wavefunction of dimer or multimer, and hopes that the charge of each monomer is exactly integer, this purpose could be realized by employing charge constraints.

Technically, adapting charge constraint into ESP fitting in terms of Lagrangian multiplier is straightforward, we only need to properly modify the form of matrix equation. For example, we want to add below constraints:

$$\begin{aligned} q_1 + q_N &= 0.5 \\ q_2 &= -0.2 \end{aligned}$$

Then below terms should be added to the function F , which is to be minimized:

$$\lambda'(q_1 + q_N - 0.5) + \lambda''(q_2 + 0.2)$$

correspondingly, two new equations appears

$$\frac{\partial F}{\partial \lambda'} = 0 = q_1 + q_N - 0.5 \quad \frac{\partial F}{\partial \lambda''} = 0 = q_2 + 0.2$$

and

$$\frac{\partial F}{\partial q_1} = 0 = \frac{\partial F_0}{\partial q_1} + \lambda' \quad \frac{\partial F}{\partial q_2} = 0 = \frac{\partial F_0}{\partial q_2} + \lambda'' \quad \frac{\partial F}{\partial q_N} = 0 = \frac{\partial F_0}{\partial q_N} + \lambda'$$

where F_0 is the function F without charge constraint. Clearly, the current ESP fitting problem in matrix equation form can be given as

$$\left[\begin{array}{ccccccc|cc} A_{11} & A_{12} & \dots & A_{1N} & n_1 & 1 & 0 \\ A_{21} & A_{22} & \dots & A_{2N} & n_2 & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & 0 & 0 \\ A_{N1} & A_{N2} & \dots & A_{NN} & n_N & 1 & 0 \\ n_1 & n_2 & \dots & n_N & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \lambda \\ \lambda' \\ \lambda'' \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \\ q_{\text{tot}} \\ 0.5 \\ -0.2 \end{bmatrix}$$

In practical programming implementation, when multiple conformations, equivalence constraint, charge constraint and penalty function are simultaneously taken into account, the ESP fitting calculation is carried out in following process: Because of introduction of the hyperbolic form of penalty function, the **A** and **q** should be updated alternately until convergence criterion is reached. In each iteration, only the first $N_{\text{atom}} \times N_{\text{atom}}$ block of **A** matrix and first N_{atom} elements of **B** vector are constructed with consideration of conformation weights and penalty function, then remainder parts of **A** and **B** are filled according to charge constraint. Finally, according to equivalence constraint, the corresponding rows of **A** are combined together (*e.g.* if atoms 3, 6, 7 are constrained to be equivalent, then these three rows should be summed up) to form a temporary matrix, whose

columns are further properly combined according to equivalence constraint to form \mathbf{A}_{eqv} matrix. Similarly, the rows of \mathbf{B} vector are transformed to \mathbf{B}_{eqv} according to equivalence constraint. After that, solving the equation $\mathbf{q}_{\text{eqv}} = \mathbf{A}_{\text{eqv}}^{-1} \mathbf{B}_{\text{eqv}}$ and correspondingly updating atomic charges according to the given equivalency relationship. In the next cycle, the diagonal terms of \mathbf{A} matrix are updated using the atomic charges obtained in last cycle, while non-diagonal terms of \mathbf{A} and all elements of \mathbf{B} vector are not needed to be changed. Multiwfn simply uses zero as initial charges for the atoms to be fitted.

3.9.16.2 Usage

Relationship between the two sets of ESP fitting codes

It is noteworthy that currently Multiwfn has two sets of codes used to perform ESP fitting.

(1) Subfunction 12 and 13 of main function 7. As introduced in Sections 3.9.10 and 3.9.11, they are mainly used to calculate regular CHELPG and MK charges, respectively.

(2) The present module, namely subfunction 18 of main function 7.

Relatively speaking, the (2) is far more powerful and general than (1), it also has more options, as will be described below. However, it should also be recognized that (1) has a few unique features, which have not been supported in (2):

- Loading additional fitting centers from external file is supported, therefore it is possible to replace extra fitting centers at *e.g.* lone pairs and σ -hole regions to enhance description of these features

- Loading additional fitting points from external file is supported.

Any input file carrying GTF information could be used for the two sets of ESP fitting modules.

Hint: Since calculation of ESP on fitting points is a computationally demanding step, while calculation speed of ESP by internal code of Multiwfn is currently not as fast as the cubegen utility in Gaussian, it is strongly recommended to enable Multiwfn invoke cubegen to evaluate ESP to significantly reduce cost of deriving ESP fitting charges. You simply need to set *cubegenpath* parameter in settings.ini to actual path of cubegen executable file in your machine. See Section 5.7 for detail.

Options in the RESP module

After loading input file and entering the RESP module, you will find many options.

- Option 1: If you just want to calculate the standard RESP charge defined by Kollman, simply select this option, the RESP charges will be calculated and outputted. Since this calculation contains two steps, this calculation is also referred to as "two-stage RESP fitting".

- Option 2: If you simply need to calculate normal ESP fitting charges, select this option. This process only includes one step, therefore it is also called "one-stage ESP fitting". Before the calculation, you can set the atomic equivalence constraints with option 5, or set the charge constraints with option 6. By default, penalty function with $b=0.1$ and $a=0.0005$ are employed to non-hydrogen atoms, hydrogens in each CH_3 and CH_2 group are constrained to be equivalent, and no charge constraint is imposed to any atom.

- Option -1: In the calculation of standard RESP charges and normal ESP fitting charges with/without additional constraints, multiple conformations could be taken into the fitting process. By selecting this option, you will be requested to input path of a plain text file containing conformer information, each line of this file consisted of file path and weight of each conformer. For example:

```
D:\a\conf1.fch 0.2
```

```
D:\a\conf2.fch 0.75
```

```
D:\b\conf.fch 0.05
```

Evidently, the sum of all weights must be exactly or very close to 1.0. After that, in the ESP fitting charge calculation, all files involved in this file will be loaded and calculated in turn (In fact, when you use this feature, the input file loaded when Multiwfn boots up will no longer important, it can even only contain structure information of present system, therefore you can also use formats such as .pdb and .xyz as input file).

- Option 3: By default, MK type fitting points is employed, if you want to change to CHELPG type fitting points, or you want to modify detailed settings of distribution of fitting pionts (such as point density), you can use this option. Note that the density of fitting points under default setting is already high enough, and it does not need to be further enhanced without special reason.
- Option 4: This option is used to set parameters of hyperbolic penalty function. The a and b used in one-stage fitting, as well as the respective a parameters used in the first and second stages of standard RESP fitting can be customized.
- Option 5: This option is use to set equivalence constraint in one-stage fitting (namely option 2). By default hydrogens in each CH₃ and CH₂ group are required to be equivalent, you can remove the constraint, or customize the constraint by providing a plain text file containing entries of equivalence constraints. For example, if the file content is

```
4,6,9-11
```

```
5,7
```

Then there will be two equivalence constraints, the first constraint requires that atoms 4, 6, 9, 10, 11 share the same charge, the second one requires that atoms 5 and 7 share the same charge. There is no upper limit on the number of entries.

Note: There must not be intersection between equivalence constraint entries. For example, if the first entry is 2-7, while the second entry is 5,8-10, the result will be completely meaningless.

- Option 6: This option is use to set charge constraint in one-stage fitting (namely option 2). By default no charge constraint is employed, however in this option you can provide a plain text file to customize the rule of charge constraint. For example, if the file content is

```
4,6,9-11 0.8
```

```
5,7 -0.3
```

Then total charge of atom 4, 6, 9, 10, 11 will be constrained to 0.8 during the fitting, while sum of charge of atoms 5 and 7 will be constrained to -0.3. There is no upper limit on the number of entries.

Note: Intersection between charge constraint entries is allowed, for example, you can require charge of atom 5 keep at 0.35 while sum of charges of atoms 3~8 is 1.0. Charge constraint could also be used in combination with equivalence constraint; however, there should not be intersection between any charge constraint entry and equivalence constraint entry. For example, charges of atoms 2, 5, 9 are required to be identical, and meantime you constrain total charge of 5,10~15 to be 0.1, such a combination will never be accepted since both of them involves atom 5

- Option 7 and details about determination of connectivity: To calculate the standard RESP charge, or to calculate the normal ESP fitting charges but requiring the charges of the hydrogens in each CH₃ and CH₂ group to be equivalent, interatomic connectivity is needed for automatically determining which atomic charges should be fitted and which hydrogens should be constrained to be equivalent. By default, if distance between two atoms is less than 1.15 times the sum of their CSD covalent radii, then they will be regarded as bonded. If you feel that the current connection

relationship does not match your expectation, you can select option 7 to read the connectivity from a specific .mol file, the .mol format contains a field recording connectivity information and can be generated by many visualization programs such as GaussView. (Alternatively, you can modify the threshold for judging bonding in main function 0, in which you can gradually change the bonding threshold until the bonding relationship shown in the graphical window completely in line with your expectation, the threshold will be retained and applied to the calculation in the RESP module).

- Option 8: This option enables Multiwfn directly load fitting points and ESP values from Gaussian output file. If you have selected this option once, then during the ESP fitting charge calculation process, Multiwfn will no longer attempts to determine position of fitting points and calculate ESP values, but prompt you to input path of a Gaussian output file of *pop=MK* or *pop=CHELPG* task in combination with *IOp(6/33=2)* keyword. In addition, Gaussian also has a keyword *IOp(6/42=x)*, where *x* is the number of fitting points per Å² for pop=MK task. *x* is recommended to set to 6, which corresponds to Multiwfn default setting. Notice that since the code in Multiwfn and in Gaussian for generating MK fitting points is different, the result calculated with and without loading Gaussian pop=MK output file must have slight deviation.

In general, this option is not needed, but if you prefer to perform time-consuming calculations on server and use Multiwfn to realize analyses on a poorly configured PC, then this function will be useful. In addition, if you may calculate ESP fitting charges for a system many times (due to some reasons, such as each time of calculation uses a different setting), then if Gaussian output file containing fitting points information is generated, calculation of ESP values in each time of ESP fitting can be avoided and thus a lot of time could be saved.

- Option 10: This option is used to set the atomic radii employed in the fitting, please check corresponding description in CHELPG section (Section 3.9.10) for detail. Notice that the default mode is "automatic", in this case, if the fitting points of MK type are employed, then MK radii will be adapted (see Section 3.9.11 on how the MK radii are defined in Multiwfn); if fitting points of CHELPG type are employed, then CHELPG radii will be adapted.

- Option 11: This option is used to choose the type of ESP that you want to fit. Commonly, this option should not be changed, the default type of ESP is the ESP defined in usual way. However, if you intend to use this RESP module to derive atomic transition charges, you should choose this option and change the ESP type to "3 Transition electronic". Please check Section 4.A.9 for more information and example.

Some examples of evaluating standard RESP charges and ESP fitting charges with extra constraints can be found in Section 4.7.7.

Information needed: GTFs, 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 various orbital composition analysis approaches, see *Acta Chim. Sinica*, **69**, 2393 (2011) (in Chinese, http://sioc-journal.cn/Jwk_hxxb/CN/abstract/abstract340458.shtml).

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

Mulliken, SCPA and Stout-Politzer methods support decomposing orbital to basis function, shell and atom compositions. Actually I have introduced the theories in Sections 3.9.5, 3.9.6 and 3.9.7, $\Theta_{i,a} \times 100\%$ 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.

These approaches rely on basis expansion, in current Multiwfn version you must use .fch, .molden and .gms 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 information 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/.molden 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= √ 3/2*(XX-YY)

D-2=XY

F 0=-3/2 / √ 5*(XXZ+YYZ)+ZZZ

F+1=- √ (3/8)*XXX- √ (3/40)*YYY+ √ (6/5)*XZZ

F-1=- √ (3/40)*XXY- √ (3/8)*YYY+ √ (6/5)*YZZ

F+2= √ 3/2*(XXZ-YYZ)

F-2=XYZ

F+3= √ (5/8)*XXX-3/ √ 8*YYY

F-3=3/ √ 8*XXY- √ (5/8)*YYY

G 0=ZZZZ+3/8*(XXXX+YYYY)-3* √ (3/35)*(XXZZ+YYZZ-1/4*XXYY)

G+1=2* √ (5/14)*XZZZ-3/2* √ (5/14)*XXXZ-3/2/ √ 14*XYZ

G-1=2* √ (5/14)*YZZZ-3/2* √ (5/14)*YYYZ-3/2/ √ 14*XXYZ

G+2=3* √ (3/28)*(XXZZ-YYZZ)- √ 5/4*(XXXX-YYYY)

G-2=3/ √ 7*XYZZ- √ (5/28)*(XXXY+XXXX)

G+3= √ (5/8)*XXXZ-3/ √ 8*XYZ

G-3=- √ (5/8)*YYYZ+3/ √ 8*XXYZ

G+4= √ 35/8*(XXXX+YYYY)-3/4* √ 3*XXYY

G-4= √ 5/2*(XXXY-XXXX)

H 0=ZZZZZ-5/ √ 21*(XXZZZ+YYZZZ)+5/8*(XXXXZ+YYYYZ)+ √ (15/7)/4*XXYYZ

```

H+1= √(5/3)*XZZZZ-3* √(5/28)*XXXZZ-3/ √28*XYZZZ+ √15/8*XXXXX+ √(5/3)/8*YYYYY+ √(5/7)/4*XXXYY
H-1= √(5/3)*YZZZZ-3* √(5/28)*YYYZZ-3/ √28*XXYZZ+ √15/8*YYYYY+ √(5/3)/8*XXXXY+ √(5/7)/4*XXYYY
H+2= √5/2*(XXZZZ-YYZZZ)- √(35/3)/4*(XXXZZ-YYYYZ)
H-2= √(5/3)*XYZZZ- √(5/12)*(XXXYZ+XYZZY)
H+3= √(5/6)*XXXZZ- √(3/2)*XYZZZ- √(35/2)/8*(XXXXX-YYYYY)+ √(5/6)/4*XXXYY
H-3=- √(5/6)*YYYZZ+ √(3/2)*XXYZZ- √(35/2)/8*(XXXXY-YYYYY)- √(5/6)/4*XXYYY
H+4= √35/8*(XXXZZ+YYYYZ)-3/4* √3*XXYYZ
H-4= √5/2*(XXXYZ-XYZZY)
H+5=3/8* √(7/2)*XXXXX+5/8* √(7/2)*YYYYY-5/4* √(3/2)*XXXYY
H-5=3/8* √(7/2)*YYYYY+5/8* √(7/2)*XXXXY-5/4* √(3/2)*XXYYY

```

An example is given in Section 4.8.1.

Information needed: Basis functions

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 does not 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, that is define fragment as all P-shells of atom 3: First, type command *all*, information of all basis functions are listed, find out the shells that attributed to center 3 and contain X, Y and Z type of basis functions (viz. PX, PY and PZ). 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, input *all* again and check if asterisks have appeared in the leftmost of corresponding rows, the marked basis functions are those that have been included in the fragment. Finally, input the letter *q* to save current fragment and return to last menu, the indices of basis functions in the fragment will be printed at the same time.

By default, fragments do not have any content. Each time you enter the fragment definition interface, the status of fragment is identical to that when you leave the interface last time. So, if you have defined the fragment earlier and you want to completely redefine it, do not forget to use "clean" command to empty the fragment first.

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

After you defined fragment 1, the fragment composition analysis based on Mulliken, 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

$$\sum_{a \in frag1} C_{a,i}^2 \times 100\% .$$

Int.cross: The sum of internal cross terms in fragment 1, namely

$$\sum_{a \in frag1} \sum_{b \in frag1} C_{a,i} C_{b,i} S_{a,b} \times 100\% .$$

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

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 $\sum_{a \in frag1} \sum_{b \in frag2} 2C_{a,i} C_{b,i} S_{a,b} \times 100\%$ 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 Orbital composition analysis by natural atomic orbital approach

(7)

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 do not have clear physical interpretations, their contribution can be ignored in occupied MOs, however they often have great contribution to virtual orbitals.

Since NAOs is an 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 does not output this matrix, you need to manually add *NAOMO* keyword between \$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 consists 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.

An example is given in Section 4.8.2.

Information needed: MO coefficients in NAO basis

3.10.5 Calculate atom and fragment contributions by Hirshfeld or Hirshfeld-I method (8,10)

Hirshfeld and Hirshfeld-I weighting function (see Sections 3.9.1 and 3.9.13, respectively) can also be used for decomposing orbital to atom and fragment compositions, the composition of atom A in orbital i is $\int \varphi_i^2(\mathbf{r}) w_A(\mathbf{r}) d\mathbf{r} \times 100\%$. The composition of a fragment is simply the sum of the compositions of the atoms that belongs to the fragment. These methods have great basis set stability and are always more reliable and reasonable than Mulliken and MMPA. In fact the Hirshfeld partition is already good enough, the more sophisticated and computationally demanding Hirshfeld-I partition is not necessary.

If you choose to use Hirshfeld partition, you will be prompted to select the way to generate atomic densities for constructing Hirshfeld weighting function, I strongly suggest using the built-in atomic densities rather than using atomic .wfn files, since the former is much more convenient. If you choose to use Hirshfeld-I partition, regular HI iterations will be performed first to yield converged atomic weighting functions (if you are confused by the operations, please consult the example of computing HI charges in Section 4.7.4 and the implementation details of Hirshfeld-I introduced in Section 3.9.13).

Before calculating orbital composition, data initialization is automatically carried out. Once it is finished, you can 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, choose option -2, then input the atom index and the index range of orbitals.

If you wish to study contribution of a fragment to orbitals, use -9 to define a fragment first, then when you input an orbital index, the contribution of the fragment will be outputted along with the contributions of all atoms. Also, you can choose -3 to calculate the contribution from the fragment you defined to a range of orbitals.

If selecting option -4, program will calculate composition of every atom in every orbitals and then export all of them to orbcomp.txt in current folder.

An example is given in Section 4.8.3.

Information needed: Atom coordinates and GTFs

3.10.6 Calculate atom and fragment contributions by Becke method (9)

This function is very similar to the function introduced in Section 3.10.5, the only difference is that Becke partition is used instead of Hirshfeld partition. For most cases, their results are in qualitative agreement with each other. Using Becke partition instead of Hirshfeld partition has a prominent advantage, namely the atomic wavefunction files are not needed, since the Becke atomic space can be simply constructed based on atomic radius. For more detail about Becke partition, see Section 3.18.0. An example is given in Section 4.8.3.

Information needed: Atom coordinates and GTFs

3.10.7 Evaluate oxidation state by LOBA method

This function is an implementation the LOBA method proposed in *Phys. Chem. Chem. Phys.*, **11**, 11297 (2009). LOBA (localized orbital bonding analysis) is a method used to evaluate atomic oxidation state based on orbital composition of localized MOs (LMOs). The idea is very simple: if an atom has N nuclear charges, and its composition in M occupied LMOs are larger than a threshold (e.g. 50%. In this case the electrons in these LMOs can be approximately viewed as completely attributed to the atom. If a LMO is doubly occupied, it should be counted twice), then the oxidation state of the atom is $N-M$.

In my opinion, this idea can also be extended to define fragmental oxidation state, namely if the sum of nuclear charge in a fragment is N , and the fragment contribution to M LMOs are larger than a certain threshold, then the fragment oxidation state will be $N-M$.

To use this function, you should provide .fch or .molden file containing LMOs (or NBOs). For example, if you are a Gaussian user, you can use the .fch file resulting from pop=saveNBO or pop=saveNLMO task as input. Alternatively, you can directly use Multiwfn to carry out orbital localization to generate .fch containing localized orbitals. An example is given in Section 4.8.4.

The result of LOBA method somewhat depends on the choice of orbital composition analysis method. For simplicity in Multiwfn the SCPA method is used for LOBA analysis, therefore diffuse function should not be used!

Information needed: Basis functions

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. This feature is not available for multi-center bond order analysis and orbital occupancy-perturbed Mayer bond order.

3.11.1 Mayer bond order analysis (1)

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

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

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 closed-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 0.05 will be printed on screen, the threshold can be adjusted by “bndordthres” 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

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

where P^S is spin density matrix, namely P^α minus P^β .

For restricted closed-shell wavefunctions free valences are zero since $P^S=0$, thus total valence of an atom is simply the sum of the related bond orders

$$V_A = \sum_{B \neq A} I_{AB}$$

Total valence (also known as atomic valence) measures atomic bonding capacity, while free valence characterizes 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 closed-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”.

Similar to Mulliken population, Mayer bond order and the multi-center bond order described below are sensitive to basis set, so do not use the basis sets having diffuse functions, otherwise the bond order result will be unreliable.

Although Mayer bond order was originally defined for single-determinant wavefunctions, for post-HF wavefunctions, Multiwfn calculates Mayer bond orders via exactly the same formulae as shown above based on corresponding post-HF density matrix. The reasonableness of this treatment has been validated in *Chem. Phys. Lett.*, **544**, 83 (2012).

Some applications of Mayer bond order can be seen in *J. Chem. Soc., Dalton Trans.*, **2001**, 2095.

Information needed: Basis functions

3.11.2 Multi-center bond order analysis (2, -2, -3)

In main function 9 there are three options (2, -2, -3) used to calculate multi-center bond order, they are very similar. I first carefully introduce option 2, then discuss the difference between options -2 and 2. Finally, -3 is briefly mentioned.

Option 2

The multi-center bond order, which is also known as multi-center index, was originally proposed in *Struct. Chem.*, **1**, 423 (1990), in some sense it may be viewed as an extension of Mayer bond order to multi-center cases. Multiwfn supports it up to 12 centers. Three/four/five/six-center bond orders are defined respectively as

$$\begin{aligned} I_{ABC} &= \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} (PS)_{ab} (PS)_{bc} (PS)_{ca} \\ I_{ABCD} &= \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \sum_{d \in D} (PS)_{ab} (PS)_{bc} (PS)_{cd} (PS)_{da} \\ I_{ABCDE} &= \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \sum_{d \in D} \sum_{e \in E} (PS)_{ab} (PS)_{bc} (PS)_{cd} (PS)_{de} (PS)_{ea} \\ I_{ABCDEF} &= \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \sum_{d \in D} \sum_{e \in E} \sum_{f \in F} (PS)_{ab} (PS)_{bc} (PS)_{cd} (PS)_{de} (PS)_{ef} (PS)_{fa} \end{aligned}$$

Similarly, infinite-center bond order can be written as

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

For open-shell cases, there are two definitions of the multi-center bond order, the first one is the sum of alpha part and beta parts:

$$\begin{aligned} I_{ABCDEF\dots K} &= I_{ABCDEF\dots K}^\alpha + I_{ABCDEF\dots K}^\beta \\ &= 2^{n-1} \left[\sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \dots \sum_{k \in K} (P^\alpha S)_{ab} (P^\alpha S)_{bc} (P^\alpha S)_{cd} \dots (P^\alpha S)_{ka} \right] \\ &\quad + 2^{n-1} \left[\sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \dots \sum_{k \in K} (P^\beta S)_{ab} (P^\beta S)_{bc} (P^\beta S)_{cd} \dots (P^\beta S)_{ka} \right] \end{aligned}$$

Another definition is using the mixed density matrix, this is not rigorous as above:

$$\begin{aligned} I_{ABCDEF\dots K} &= \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \dots \sum_{k \in K} (P^{\text{mixed}} S)_{ab} (P^{\text{mixed}} S)_{bc} (P^{\text{mixed}} S)_{cd} \dots (P^{\text{mixed}} S)_{ka} \\ P^{\text{mixed}} &= P^\alpha + P^\beta \end{aligned}$$

For unrestricted or restricted open-shell wavefunction, the output of multi-center bond order analysis consists of four terms, which have been explained above: (1) The result from alpha density matrix (2) The result from beta density matrix (3) The sum of the result of alpha and beta parts (4) The result from mixed alpha&beta density matrix.

Notice that the multi-center bond orders for different number of centers are not directly comparable, since the result is not in the same magnitude. However, in *Phys. Chem. Chem. Phys.*, **118**, 11839 (2016), it was shown that the normalized multi-center bond order is comparable for different ring size and can be simply calculated as $\text{MCI}^{1/n}$, where n is the number of centers and MCI is the multi-center bond order defined above. For example, at B3LYP/6-31G* level, the MCI for H_3^+ , benzene (6 centers) and naphthalene (12 centers) are 0.2963, 0.0863 and 0.0080,

respectively, while the normalized result are 0.667, 0.665 and 0.617, respectively. When MCI is negative, the normalized value should be calculated as $-|MCI|^{1/n}$.

Due to the ambiguity of definition and the use of non-orthogonal basis functions, the value of multi-center bond order may relies on input direction if the atoms lack of symmetry. For example, the result of ABCD can be different from DCBA (the opposite input direction of ABCD). The reason is clear: The term in the summation in the case of ABCD is $(PS)_{ab}(PS)_{bc}(PS)_{cd}(PS)_{da}$, while if we invert the input direction, the term will become $(PS)_{dc}(PS)_{cb}(PS)_{ba}(PS)_{ad}$. Although both P and S are symmetry matrices, their product PS is not necessarily symmetry, so the two terms are not equivalent. In my own viewpoint, in order to obtain reasonable result, for example A-B-C-D-E-F, one should calculate ABCDEF and FEDCBA respectively and take their average. While some people advocated that it is needed to take all possible permutations into account to get a 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 required to be calculated and then summed up. This treatment does not have clear physical meaning and is absolutely unnecessary, it also elevates the computational cost significantly.

The computational cost for multi-center bond order increases exponentially with the number of centers; for ring size larger than 12 centers the cost will be prohibitively high, so Multiwfn doesn't support higher number of centers.

Automatical search of multi-center bonds: If you input -3 when Multiwfn asks you to input atom combination, 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. Due to efficiency consideration, the search may be not exhaustive. Also note that the search is based on mixed alpha&beta density matrix for open-shell cases.

Option -2

The most severe drawback of multi-center bond order is its high basis set dependency. In particular, if diffuse functions are involved, then multi-center bond order may be completely wrong. In order to tackle this problem, I proposed an alternative way (to be published) to calculate the multi-center bond order, and the idea is implemented as option -2.

Option -2 is very similar to option 2 (as introduced above), the only difference is that the multi-center bond order is calculated based on natural atomic orbital (NAO) basis rather than based on the basis functions originally defined by the basis set. Since NAO is an orthonormal set and thus overlap matrix S is an identity matrix, the formula can be simplified as (using closed-shell form for example)

$$I_{ABCDEF\dots K} = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \dots \sum_{k \in K} P_{ab} P_{bc} P_{cd} \dots P_{ka}$$

The multi-center bond order calculated in this manner has very good stability with respect to change of basis set. Even when diffuse functions are presented the result is still highly reliable. According to my experience, if no basis functions show diffuse character, the results given by option 2 and -2 will be very similar, though not exactly identical.

Another advantage of employing NAO basis is that the ambiguity of multi-center bond order due to the input direction of atom indices is completely eliminated, that means the result of ABCD must be identical to DCBA. This is because the matrix involved in the summation (P) is symmetry.

In order to use option -2, the output file of NBO module embedded in Gaussian or standalone NBO program should be used as input file, and *DMNAO* keyword must be used to make NBO print density matrix in NAO basis. If you are a Gaussian user, for example, you can use output file of below instance as input file of Multiwfn (DO NOT use .fch file for this analysis!).

```
#p PBE1PBE/6-311G** pop=nboread
```

```
opted
```

```
0 1
C          0.00000000  1.38886900  0.00000000
... [ignored]
H          -2.14060700  1.23588000  0.00000000
```

```
$NBO DMNAO $END
```

In this function, if you only input indices of two atoms, then the result is just Wiberg bond order under NAO basis, which is completely identical to that printed by *bndidx* keyword of NBO program.

Option -3

There is a hidden option -3 in main function 9, it is used to calculate multi-center bond order under Löwdin orthogonalized basis (see next section for more information). Since this method should not have evident advantages over original multi-center bond order definition, this option is invisible in the interface. However, if you have interesting, you can have a try.

Information needed: Basis functions (options 2, -3), NBO output file with DMNAO keyword (option -2)

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

Wiberg bond order is defined as follows, see footnote in *Tetrahedron*, **24**, 1083 (1968)

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

The original definition of 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 closed-shell system. Actually, Mayer bond order can be seen as a generalization of Wiberg bond order, for restricted closed-shell system and orthonormal basis functions (namely *S* matrix is identity matrix) cases their results are completely identical.

In this function, Multiwfn first orthogonalizes basis functions by Löwdin method and then performs usual Mayer bond order analysis. The threshold for printing is controlled by “*bndordthres*” in settings.ini too.

As shown in *J. Mol. Struct. (THEOCHEM)*, **870**, 1 (2008), the Wiberg bond order calculated in this manner, say W_L , has much less sensitivity to basis set than Mayer bond order (whereas for small basis sets, their results are closed to each other). One should be aware that W_L tends to overestimates bond order for polar bonds in comparison with Mayer bond order.

Commonly, if there is not special reason, using Mayer bond order is more preferred.

Notice that numerous literatures used NBO program to calculate Wiberg bond order, the result must be somewhat different to that produced by present function, because in NBO program the Wiberg bond orders are calculated in the basis of natural atomic orbitals (NAO), which are generated by OWSO orthogonalization method. Multiwfn is also possible to calculate Wiberg bond order under NAO basis, and furthermore, the result can be decomposed as atomic orbital pair contributions, see Section 3.11.8 for details.

Information needed: Basis functions

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

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

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

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 “bndordthres” parameter in settings.ini.

Mulliken bond order is easy to be decomposed 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.

Information needed: Basis functions

3.11.5 Orbital occupancy-perturbed Mayer bond order (6)

Orbital occupancy-perturbed Mayer bond order was firstly proposed in *J. Chem. Theory Comput.*, **8**, 908. Put simply, by using this method one can obtain how large is the contribution from specific orbital to Mayer bond order.

Orbital occupancy-perturbed Mayer bond order can be written as

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

The only difference between this definition and Mayer bond order shown in Section 3.11.1 is that P^α and P^β have been replaced by P_X^α and P_X^β respectively. P_X stands for the density matrix generated when occupation number of a specific orbital is set to zero. The difference between $I_{A,B}^*$ and Mayer bond order can be regarded as a measure of contribution from the orbital to Mayer

bond order. Bear in mind, because Mayer bond order is not a linear function of density matrix, the sum of $I_{A,B}^*$ for all orbitals is not equal to Mayer bond order generally.

In Multiwfn, you only need to input indices of two atoms, then $I_{A,B}^*$ for all occupied orbitals and the difference between $I_{A,B}^*$ and Mayer bond order will be outputted. The more negative (positive) the difference, the more beneficial (harmful) to the bonding due to the existence of the orbital.

You can also use another way to calculate $I_{A,B}^*$, that is using wavefunction modification module (main function 6) to manually set occupation number of a specific orbital to zero, and then calculate Mayer bond order as usual, but this manner may be tedious if you want to calculate $I_{A,B}^*$ for many orbitals.

This kind of analysis is illustrated in Section 4.9.1 and Section 4.19.3.

Information needed: Basis functions

3.11.6 Fuzzy bond order (7)

Fuzzy bond order (FBO) was first proposed by Mayer in *Chem. Phys. Lett.*, **383**, 368 (2004):

$$B_{AB} = B_{AB}^\alpha + B_{AB}^\beta = 2 \sum_{\mu} \sum_{\nu} [(\mathbf{P}^\alpha \mathbf{S}^A)_{\mu\nu} (\mathbf{P}^\alpha \mathbf{S}^B)_{\nu\mu} + (\mathbf{P}^\beta \mathbf{S}^A)_{\mu\nu} (\mathbf{P}^\beta \mathbf{S}^B)_{\nu\mu}]$$

$$S_{\mu\nu}^A = \int w_A(\mathbf{r}) \chi_\mu^*(\mathbf{r}) \chi_\nu(\mathbf{r}) d\mathbf{r}$$

where \mathbf{S} is overlap matrix of basis functions in fuzzy atomic spaces. In Multiwfn, Becke's fuzzy atomic space with sharpness parameter $k=3$ in conjunction with modified CSD radii is used for calculating FBO. (See Section 3.18.0 for introduction of fuzzy atomic space).

Commonly the magnitude of FBO is close to Mayer bond order, especially for low-polar bonds, but much more stable with respect to the change in basis set. According to the comparison between FBO and delocalization index (DI) given in *J. Phys. Chem. A*, **109**, 9904 (2005), FBO is essentially the DI calculated in fuzzy atomic space. See Section 3.18.5 for detail about DI.

Calculation of FBO requires performing Becke's DFT numerical integration, due to which the computational cost is larger than evaluation of Mayer bond order. By default, 40 radial points and 230 angular points are used for numerical integration. This setting is able to yield accurate enough results in general. If you want to further refine the result, you can set the number of radial and angular points by "radpot" and "sphpot" in settings.ini manually, and ensure that "iautointgrid" has been set to 0.

The threshold for printing results is controlled by "bndordthres" parameter in settings.ini.

Information needed: GTFs, atom coordinates

3.11.7 Laplacian bond order (8)

In *J. Phys. Chem. A*, **117**, 3100 (2013) (<http://pubs.acs.org/doi/abs/10.1021/jp4010345>), I proposed a novel definition of covalent bond order based on the Laplacian of electron density $\nabla^2 \rho$ in fuzzy overlap space, called Laplacian bond order (LBO). The LBO between atom A and B can be simply written as

$$L_{A,B} = -10 \times \int_{\nabla^2 \rho < 0} w_A(\mathbf{r}) w_B(\mathbf{r}) \nabla^2 \rho(\mathbf{r}) d\mathbf{r}$$

where w is a smoothly varying weighting function proposed by Becke and represents fuzzy atomic space, hence w_{AWB} corresponds to fuzzy overlap space between A and B. Note that the integration is only restricted to negative part of $\nabla^2 \rho$. The physical basis of LBO is that the larger magnitude the integral of negative $\nabla^2 \rho$ in the fuzzy overlap space, the more intensive the electron density is concentrated in the bonding region, and therefore, the stronger the covalent bonding.

The reasonableness and usefulness of LBO were demonstrated by applying it to a wide variety of molecules and by comparing it with many existing bond order definitions. It is shown that LBO has a direct correlation with the bond polarity, the bond dissociation energy and the bond vibrational frequency. The computational cost of LBO is low, also LBO is insensitive to the computational level used to generate electron density. In addition, since LBO is inherently independent of wavefunction, one can in principle obtain LBO by making use of accurate electron densities derived from X-ray diffraction data.

In Multiwfn, Becke's fuzzy atomic space with sharpness parameter $k=3$ in conjunction with modified CSD radii is used for calculating LBO. (See Section 3.18.0 for detail about fuzzy atomic space). The threshold for printing results is controlled by “bndordthres” parameter in settings.ini.

Note that in current implementation, LBO is particularly suitable for organic system, but not for ionic bonds since in these cases a better definition of atomic space should be used to faithfully exhibit actual atomic space. LBO is also not very appropriate for studying the bond between two very heavy atoms (heavier than Ar), because these bonds are often accompanied by insignificant charge concentration in the fuzzy overlap space, even though the bonding is doubtless covalent.

Information needed: GTFs, atom coordinates

3.11.8 Decompose Wiberg bond order in NAO basis as atomic orbital pair contributions (9)

As mentioned in Section 3.11.3, Wiberg bond order is expressed as

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

The data is calculated for atom pair. Since the expression is simply a linear summation of square of density matrix element, it is straightforward to decompose Wiberg bond order as basis function pair

contribution (this idea is to be published). For example, P_{ab}^2 is simply the contribution from interaction between basis functions a and b . Since one-to-one correspondence between basis function and atomic orbital is lacking when extended basis set is used, in order to make the decomposition method full of physical meaning, the decomposition is best to be carried out under natural atomic orbitals (NAOs). Each non-Rydberg type of NAO uniquely corresponds to an atomic orbital, thus, by above decomposition method, Wiberg bond order *at the atomic orbital scale* can be obtained.

In addition, contribution from interaction between atomic orbital shells i and j can be obtained as $\sum_{a \in \text{shell } i} \sum_{b \in \text{shell } j} P_{ab}^2$.

After you enter this function, simply input indices of two atoms, the non-neglectable contribution from NAO pairs and NAO shell pairs together with total Wiberg bond will be printed.

This function currently is limited for closed-shell systems.

The input file of this function is completely identical to the option -2 described in Section 3.11.2, namely the output file of NBO program containing density matrix information (*i.e.* DMNAO keyword is required).

By the way, it is also possible to decompose multi-center bond order as atomic orbital contribution. For example, three-center bond order is expressed as

$$I_{ABC} = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} P_{ab} P_{bc} P_{ca}$$

Clearly, $P_{ab} P_{bc} P_{ca}$ can be regarded as contribution from interacting between NAOs a , b and c .

However, decomposition of multi-center bond order under NAO basis has not been implemented.

An illustrative example of this decomposition analysis is given in Section 4.9.4.

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 represents the 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 valuable tool for analyzing the nature of electron structure.

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

$$\text{TDOS}(E) = \sum_i \delta(E - \varepsilon_i)$$

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

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

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

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

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 regardless of the select of w_{gauss} .

The curve map of broadened partial DOS (PDOS) and overlap DOS (OPDOS) are very valuable for visual study of orbital composition. PDOS of fragment A is defined as

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

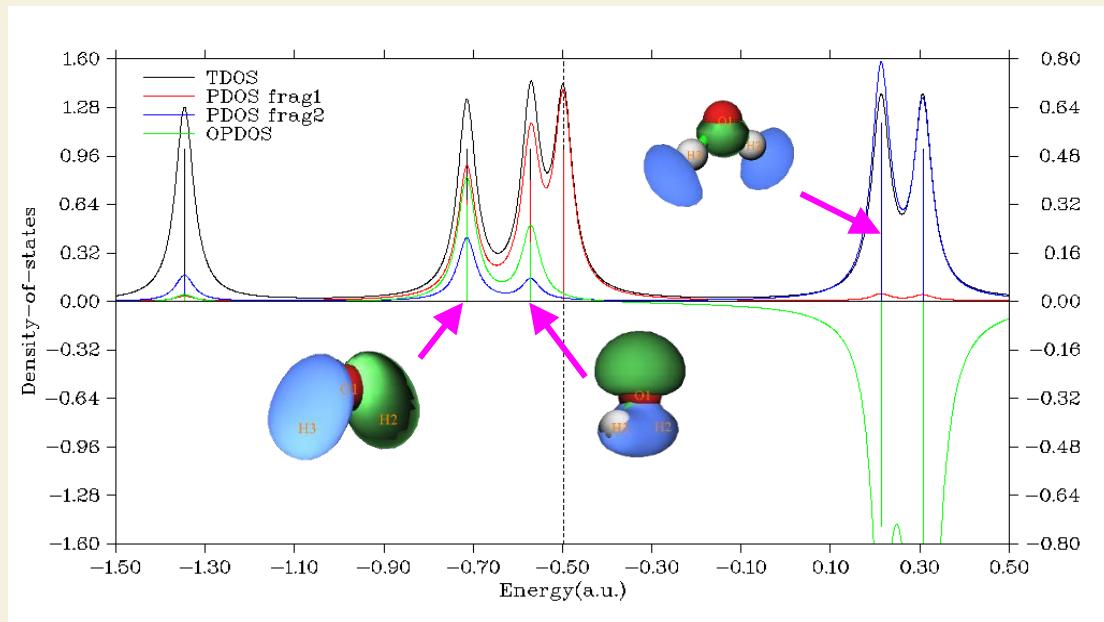
where $\Xi_{i,A}$ is the composition of fragment A in orbital i . Note that the word "projected DOS" used in some literatures is essentially equivalent to the partial DOS. The OPDOS between fragment A and B is defined as

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

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 information 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 vertical dashed line indicates the position of HOMO level. 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/.molden/.gms 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 occ [strength] [FWHM]           ← For orbital 1
energy occ [strength] [FWHM]           ← For orbital 2
energy occ [strength] [FWHM]           ← For orbital 3
...
energy occ [strength] [FWHM]           ← For orbital nmo

```

where *energy* and *occ* denote orbital energy and occupation number. *nmo* is the number of orbitals recorded in this file. *inp* is input type, if it is set to 1, only energies (in a.u.) and occupation numbers will be loaded, strengths and FWHMs of all orbitals will be set to 1.0 and 0.25 respectively and automatically; if it is set to 2, you have to specify strength and FWHM for each orbital. 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. The most important ones among them may be option 0 and option -1.

-5 Customize energy levels, occupations, strengths and FWHMs for specific MOs: By this option, you can manually set energies, occupations, strengths and FWHMs for specific orbitals. For example, if you would like only to plot DOS for a few MOs, you can set strengths of the other MOs to zero (by default, strengths for all MOs are 1.0).

-4 Show all orbital information: Print information of all orbitals on screen, also point out Fermi energy (when occupation number information is available). Note that “Fermi energy” is a concept ill-defined for isolated systems, any energy that $\geq E_{\text{HOMO}}$ and $< E_{\text{LUMO}}$ may be regarded as Fermi energy. In the output, E_{HOMO} is simply taken as Fermi energy.

-3 Export energy levels, strengths, FWHMs to plain text file: Export 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.

-1 Define fragments: You can define up to 10 fragments in this option. After you entered this option, the information of present fragments are shown on screen. You can input *x* to define fragment *x*, and input *-x* to unset fragment *x*, or input *i,j* to exchange the definition of fragment *i* and *j*. To leave this interface, input 0. The PDOS will be plotted for all of the fragments defined in this option. OPDOS will be drawn only when both fragments 1 and 2 have been defined.

DOS graph will contain the line and curve of “PDOS frag 1” only when fragment 1 is defined.

0 Draw TDOS graph!: Draw DOS graph right now! If any fragment has been defined, the texts in this option will become “TDOS+PDOS”; if both fragments 1 and 2 have been defined, the texts will become “TDOS+PDOS+OPDOS”.

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

2 Set energy range: This option is used to set the lower limit, upper limit and step between labels of X-axis.

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/.molden/.gms 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.

8 Switch unit between a.u. and eV

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

Once you choose option 0, Multiwfns starts to calculate data set and then DOS graph pops up, after clicking right mouse button on the graph to close it, post-process menu will appear on the

screen, which has many options. They control the range and stepsize of Y-axis, if the lines or curves corresponding to TDOS, PDOS and OPDOS will be shown on DOS graph, if a vertical dashed line will be drawn to highlight HOMO level, the texts in the legends, etc. Worthnotingly, there is an option named “Set scale factor of Y-axis for OPDOS”, if the value is set to k and the range of left-axis (for TDOS/PDOS) is set to e.g. [-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 DOS map you have seen as graphical file in current folder, 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.

A very detailed example of plotting TDOS, PDOS and OPDOS is given in Section 4.10.1.

3.12.4 Local DOS

There is a special kind of DOS named local DOS (LDOS). LDOS curve for a given point \mathbf{r} is evaluated as follows:

$$\text{LDOS}(\mathbf{r}, E) = \sum_i \varphi_i^2(\mathbf{r}) f(E - \varepsilon_i)$$

To plot such a map, choose "10 Draw local DOS for a point", you will be prompted to input the coordinate of the point \mathbf{r} .

One can also plot LDOS for a set of points placed evenly in a line as color-filled map, the X-axis correspond to energy, while the Y-axis corresponds to coordinate in the line relative to starting point. To plot this kind of LDOS map, choose "11 Draw local DOS along a line", you will be prompted to input the coordinate of starting point and end point defining the line, as well as the number of points consisting of the line.

In the DOS module, the options controlling FWHM, energy unit, energy range and scale ratio affect the resulting LDOS graphs.

Note that fragment definition does not affect the result of LDOS, *i.e.* the LDOS always corresponds to total DOS. However, if you want to separate angular moment contribution to LDOS, you can use subfunction 25 of main function 6 to set coefficient of unwanted GTF in all MOs to zero, then they will not contribute to LDOS.

An example of plotting LDOS can be found in Section 4.10.2.

Information needed: Basis functions and atom coordinates (only for PDOS, OPDOS and local DOS), energy levels and occupation numbers (.fch/.molden/.gms file, Gaussian output file or the plain text file recording them)

3.13 Plotting IR, Raman, UV-Vis, ECD, VCD and ROA spectra (11)

Although spectrum plotting is beyond the scope of wavefunction analysis, due to this function is very useful for theoretical spectrum research, the spectrum plotting function is also included in Multiwfn. Multiwfn is able to plot IR (Infrared), normal Raman / Pre-resonance Raman, UV-Vis (ultraviolet-visible), ECD (electronic circular dichroism), VCD (vibrational circular dichroism) and Raman optical activity (ROA) spectra.

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 commonly used broadening function for vibrational spectra (IR, Raman, VCD and ROA) is Lorentzian function, while for electronic spectra (UV-Vis and ECD), Gaussian function is often 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, Raman, VCD and ROA spectra is cm^{-1} in common, for UV-Vis and ECD spectra all of eV ($1\text{eV}=8.0655*1000\text{cm}^{-1}$), nm and 1000cm^{-1} are common units. The strength data of each transition mode outputted by quantum chemistry programs is proportional to the area under the broadened curve. Below we discuss some details of the six types of spectra supported by Multiwfn respectively.

IR: The frequently used unit of molar absorptivity ϵ , namely L/mol/cm , can be rewritten as

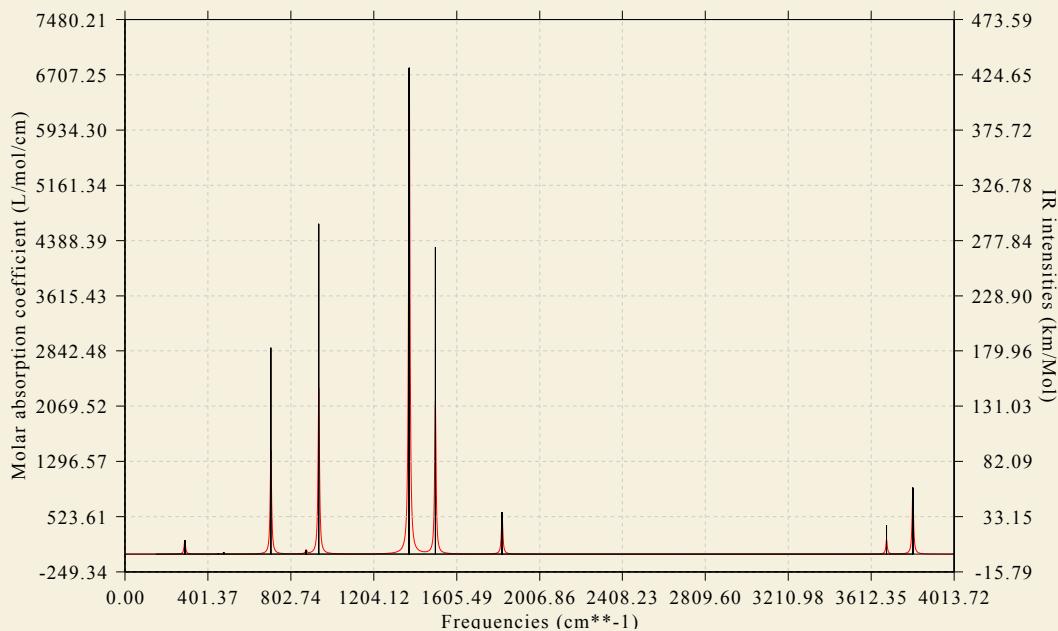
$$\frac{\text{L}}{\text{mol} \times \text{cm}} = \frac{1000\text{cm}^3}{\text{mol} \times \text{cm}} = \frac{1000\text{cm}^2}{\text{mol}}$$

Since IR intensity $\varsigma = \int \epsilon(\nu) d\nu$, where ν is transition frequency, the unit of ς 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 vibrational 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).



Raman (normal or pre-resonance): Raman spectrum measures intensity of scattered light. Beware that for a specific vibration mode i , its Raman activity S_i and Raman intensity I_i are two different quantities. Raman activity is a intrinsic property of each molecular vibrational mode, while Raman intensity is directly related to experimental Raman spectrum, and its value is dependent of the choice of wavenumber of incident light ν_0 as well as temperature. The conversion relationship can be found in some literatures, e.g. *J. Mol. Struct.*, **702**, 9-21 (2004):

$$I_i = \frac{C(\nu_0 - \nu_i)^4 S_i}{\nu_i B_i} \quad B_i = 1 - \exp\left(-\frac{hc\nu_i}{kT}\right)$$

where C is a suitably chosen common normalization factor for all peak intensities, ν_i is vibrational frequency. Only the Raman spectrum broadened based on Raman intensities is strictly comparable to the experimental one; however, as the peak position of Raman activity and intensity are identical, and for a specific vibrational mode its intensity is proportional to its activity, the Raman spectrum broadened from Raman activities is also useful in some sense.

Almost all quantum chemistry codes only output Raman activities, therefore by default, Multiwfn emulates the Raman spectrum in terms of broadening Raman activities. However you can also let Multiwfn to first convert the activities to intensities via option 19 (ν_0 and T are provided by users) based on above formula, then the Raman spectrum will be obtained by broadening Raman intensities. The normalization coefficient C is fixed to 10^{-12} in Multiwfn (which is unimportant since what we are interested in is the shape rather than the absolute height of Raman spectrum). The integral of the peak broadened by one unit of Raman activity or intensity is equal to 1.

UV-Vis: 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 have not been built. However, there is a well-known relation between theoretical data and experimental spectra: If the unit 1000cm^{-1} and L/mol/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 $1/4.32 \times 10^6$. Equivalently, if eV is used as X-axis unit, the value should be

$1/4.32/8.0655*10^6=28700$. By this relation, UV-Vis spectrum can be simulated by theoretical data.

Fluorescence (or phosphorescence, as long as you can obtain oscillator strength by means of spin-orbit coupling calculation) spectrum can also be plotted with exactly the same manner as UV-Vis, the only difference is that according to Kasha's rule, you should only take the first singlet excited state into account.

ECD: The significance of rotatory strengths in ECD spectrum is analogous to oscillator strengths in UV-Vis spectrum, each electron transition mode corresponds to a rotatory strength. If the rotatory strengths are broadened, after somewhat scaling and shifting, the resultant curve will be comparable to experimental ECD spectrum. The integral of the peak broadened by one unit of rotatory strength is equal to 1. In quantum chemistry programs, such as Gaussian, rotatory strengths can be calculated in length representation or in velocity representation, in the former case the strengths are origin-dependent, while in the latter case the strengths are origin-independent. For complete basis set situation, the results in the two representations converge to the same values. Commonly velocity representation is recommended to be used.

VCD: VCD is akin to ECD, each vibrational mode has a rotatory strength. After broadening them, the curve will be comparable to experimental VCD spectrum.

ROA: The ROA spectrum measures the difference between scattering intensity of right and left circularly polarized light

$$\text{ROA intensity} \equiv I_i^R - I_i^L \propto \frac{(\nu_0 - \nu_i)^4 A_i}{\nu_i B_i} \quad B_i = 1 - \exp\left(-\frac{hc\nu_i}{kT}\right)$$

The ROA strength data outputted by Gaussian ROA task in fact is the A_i term, which should be converted to actual ROA intensity according to above equation. The A_i term is dependent of frequency of incident light. The integral of the peak broadened by one unit of ROA intensity/strength is equal to 1. There are several different forms of ROA, including ROA SCP(180), ROA SCP(90), ROA DCP(180). The 90 and 180 denote the angle between incident light and scattered light. The SCP (scattered circular polarization) means the incident light is linearly polarized light while the scattered light is circularly polarized light; DCP (dual circular polarization) corresponds to the case that both incident and scattered lights are circularly polarized light. The ROA SCP(180) is commonly employed, and it is also known as SCP backscattered ROA.

The ROA task of Gaussian also simultaneously output frequency-dependent Raman strength, which corresponds to the R_i term of below equation

$$\text{Raman intensity} \equiv I_i^R + I_i^L \propto \frac{(\nu_0 - \nu_i)^4 R_i}{\nu_i B_i} \quad B_i = 1 - \exp\left(-\frac{hc\nu_i}{kT}\right)$$

correspondingly, there are also Raman SCP(180), Raman SCP(90) and Raman DCP(180) data. Since the current used incident light should be far from electron excitation energy, such Raman spectrum is called as far from resonance Raman.

3.13.2 Input file

Only the input files mentioned in this section are supported by Multiwfn for plotting spectrum purpose. DO NOT use such as .fch, .molden and .wfn as input file, evidently the data needed for plotting spectrum are not recorded in these files.

1 Gaussian output file:

- IR spectrum: Use output file of *freq* task as input. If you use Gaussian 09 D.01 or later revisions and meantime *freq=anharm* keyword was specified to carry out anharmonic analysis, Multiwfn will prompt you to choose if loading anharmonic frequencies and IR intensities instead of the harmonic ones.
- Raman spectrum: Use output file of *freq=raman* task as input. If you would like to plot pre-resonance Raman spectrum, you should at the same time use *CPHF=rdfreq* keyword and write the frequencies of the incident lights after a blank line under the geometry specification, e.g. *300nm 400nm 500nm*. If you hope to plot anharmonic Raman spectrum, use *freq(raman,anharm)* keywords, then Multiwfn will prompt you to choose if loading anharmonic frequencies and Raman activities instead of the harmonic ones.
- VCD spectrum: Use output file of *freq=VCD* task as input. If you hope to plot anharmonic Raman spectrum (supported by Gaussian since G16), use *freq(VCD,anharm)* keywords, then Multiwfn will prompt you to choose if loading anharmonic frequencies and rotatory strengths instead of the harmonic ones.

For anharmonic IR, Raman and VCD spectra, you can choose if only loading anharmonic fundamental data, or simultaneously loading anharmonic overtone band or combination band data.

- UV-Vis or ECD spectrum: Use output file of TDDFT, TDHF, CIS or ZINDO task as input, no any additional keywords are required. The rotatory strengths in both length and velocity representation (the data under "R(length)" and "R(velocity)", respectively) can be chosen to be loaded by Multiwfn. Output file of optimization task for excited state can also be used, only the last output for transition information will be loaded by Multiwfn, therefore the resulting spectrum corresponds to final geometry.
- ROA spectrum: Use output file of *freq=ROA* task as input. The frequencies of incident lights should be specified after a blank line under the geometry specification, e.g. *0.02, 0.03, 0.04, 0.05*, the unit is default to a.u.; or write e.g. *500nm 520nm 550nm*.

2 ORCA output file:

- IR spectrum: Use *freq* keyword.
- Raman spectrum: Use keywords like below

```
! b3lyp def2-SVP numfreq
%elprop Polar 1 end
```
- UV-Vis and ECD spectrum: Use keywords like below (CIS, TDHF and ZINDO calculations are also supported)

```
! b3lyp def2-SVP
%tddft
nroots 20
TDA false
end
```

Notice that for TDDFT calculation, ORCA by default uses Tamm-Dancoff Approximation (TDA), in this case the oscillator strength and rotatory strengths are evidently not as good as that produced by TDDFT, therefore in above example *TDA false* is used to make ORCA use standard TDDFT formalism. Also it is worth noting that ORCA employs length representation for rotatory strengths in ECD.

Output file of sTDA or sTD-DFT task of ORCA (see below for details) is also supported by

Multiwfn for plotting UV-Vis or ECD spectrum purpose. Below is example keywords:

```
! PBE0 def2-SVP def2/J RIJCOSX
%maxcore 6000
%pal nprocs 36 end
%otddft
Mode sTDDFT
Ethresh 10.0
maxcore 6000
end
```

Since ORCA 4.1 spin-orbit coupling (SOC) effect can be taken into account during TDDFT calculation, and all data needed for plotting SOC corrected UV-Vis and ECD spectra are automatically outputted. Therefore, when output file of SOC-TDDFT task is loaded into Multiwfn, when you enter UV-Vis and ECD plotting option, you can select if loading SOC corrected data instead of the one without SOC consideration. It is very easy to conduct SOC-TDDFT, simply adding *dosoc true* in *%otddft* section, for example:

```
%otddft nroots=40 TDA false dosoc true end
```

- VCD and ROA spectra: Currently not supported by ORCA.

3 Grimme's sTDA output file

The sTDA proposed by Grimme in *J. Chem. Phys.*, **138**, 244104 (2013) is a method approximately solve the TDDFT equation and thus reduces the computational cost of the electron excitation part of TDDFT calculation by about two or three orders of magnitude. The correpsonding STDA program can be freely downloaded at

<https://www.chemie.uni-bonn.de/pctc/mulliken-center/software/stda/stda>

Although the Grimme's sTDA code has already been implanted into ORCA program, the *tda.dat* file outputted by standalone sTDA code can also be used as input file of Multiwfn for plotting UV-Vis or ECD purpose.

When ECD is to be plotted, you can choose which representation of rotatory strength will be used. The length and velocity representations have been mentioned above, while the mixed-form of representation, which is recommended in the sTDA original paper, is defined as $R_M = R_V \times f_L / f_V$, where f_L and f_V are length and velocity representations of oscillator, respectively, and R_V is the velocity representation of rotatory strength.

4 Grimme's xtb output file

The xtb program written by Grimme is mainly used to carry out GFN-xTB calculation (*J. Chem. Theory Comput.*, 13, 1989 (2017)), which may be viewed as a semiempirical version of DFT method. It is not only robust but also rather fast, it can be conveniently applied to systems consisted of hundreds of atoms. Note that although the accuracy of xtb frequency has been verified to be basically reasonable, the quality of IR intensities outputted by xtb is not good (according to my experiences). xtb program can be freely obtained via <https://www.chemie.uni-bonn.de/pctc/mulliken-center/software/xtb/xtb>.

By using following command, xtb will optimize the structure in *test.xyz* and then perform frequency analysis: *xtb test.xyz -ohess |tee test.out*. Then the output file *test.out* will contain harmonic frequencies and IR intensities, it can be used as input file of Multiwfn for plotting IR

spectrum purpose.

Note that xtb outputs $3N$ modes, where N is the number of atoms. When plotting spectrum, Multiwfn will ask you how many lowest modes will be removed to discard the overall movement modes, you should input 5 and 6 for linear and non-linear system, respectively.

5 Plain text file:

For generality, Multiwfn supports plain text file as input, you can extract transition data from output files of quantum chemistry packages other than Gaussian/ORCA/sTDA, 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
```

where *numdata* denotes how many entries in this file. If *inptype* is set to 1, then only *energy* and *strength* will be read, and FWHMs for all transitions will be automatically set. If *inptype* is set to 2, then FWHMs will be read too. The transitions should be sorted according to the energies from low to high. The unit of both *energy* and *FWHM* should be in cm^{-1} for IR, Raman, VCD and ROA spectra, in eV for UV-Vis and ECD spectra. The unit of strength should be in km/mol , $\text{\AA}^4/\text{amu}$, cgs (10^{-40} erg-esu-cm/Gauss), 10^{-44} esu 2 cm 2 , 10^4 K for IR, Raman, ECD, VCD, ROA spectra, respectively (oscillator strength of UV-Vis is dimensionless).

An example of the plain text file is shown below (for IR)

```
6 2
81.32920    0.72170    8.0
417.97970   3.58980    8.0
544.67320   21.06430   8.0
583.12940   41.33960   8.0
678.66900   91.47940   8.0
867.37410   2.94480   8.0
```

3.13.3 Usage and options

After boot up Multiwfn, first input path of Gaussian/ORCA/sTDA/xtb output file or the plain text file containing transition data, then enter main function 11, and you will be prompted to select the type of spectrum (for pre-resonance Raman and ROA, you also need to choose the interested frequency), after that you will see below options. The meaning of some options may be different for various types of spectra.

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

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

0 Plot spectrum: Plot spectrum right now!

1 Save graphical file of the spectrum in current folder: As the title says.

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. The stepsize between ticks also needs to be inputted. By default the range of X-axis is adjusted automatically according to minimum and maximum transition energies.

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

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

In many cases, after you adjusting setting for Y-axis at one side, the position of zero point of this Y-axis deviates from that of Y-axis at another side, which makes the graph weird; To address this problem, options 4 and 5 enable you to choose if correspondingly adjusting range of Y-axis at another side. If you input *y*, then lower/upper limit and stepsize of Y-axis at another side will be proportionally scaled to make zero point of left and right Y-axes are in the same horizontal line.

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

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 Raman, ECD, VCD and ROA spectra the default value is 1.0, for IR the value is 100, 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*10⁻⁶) is used.

8 Input full width at half maximum (FWHM): As the title says.

9 Toggle showing discrete lines: Choose if show the discrete lines corresponding to transitions on the spectrum graph.

10 Switch the unit of infrared intensity / Set the unit of excitation energy: For IR spectrum, switch the unit of IR intensities between km/mol (default) and esu²*cm². For UV-Vis and ECD spectra, choose the unit of energies between 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.

12 Set shift value in X: If the value is set to *k*, then the final curve and discrete lines will be shifted by *k* in X direction.

13 Set colors of curve and discrete lines: As the title says.

14 Multiply the transition energies (or vibrational frequencies) by a factor: The selected transition energies (or vibrational frequencies) can be scaled by a given factor. This is option is mainly used to apply frequency scaling factor onto the calculated harmonic frequencies.

15 Output contribution of individual transition to the spectrum: If select this option, user will be prompted to input a criterion (e.g. *k*), then not only the total spectrum (as option 2), but also the contributions from the individual transitions whose absolute value of strength larger than *k* will be outputted to *spectrum_curve.txt* in current folder. This feature is particularly useful for identifying the nature of total spectrum.

16 Find the positions of local minima and maxima: After you select this option, the positions of the maxima and minima of the spectrum will be located.

17 Toggle showing dashed grid lines: Choose if plotting gray dashed grid lines on the

spectrum.

18 Toggle weighting spectrum of each system: See explanation in next section.

19 Convert Raman (or ROA) activities to intensities: As mentioned earlier, this option is used to convert Raman (or ROA) activities to intensities. After that, the Raman (or ROA) spectrum plotted by option 0 will be the one broadened based on Raman (or ROA) intensities rather than Raman (or ROA) activities. The wavenumber of incident light and temperature are inputted by users.

20 Modify strengths: You can select some transitions and change their strengths (depending on spectrum type) to specific value. This option is very useful if you would like to plot fluorescence spectrum, in this case you need to set oscillator strength of all transitions except for the lowest singlet excitation to zero (Kasha's rule).

21 Toggle showing weighted curve: As shown in next section, Multiwfn is able to plot weighted spectrum according to Boltzmann distribution. This option controls if plotting corresponding curve in due time.

22 Set thickness of curves/lines/texts/axes: As the title says.

Note that the actual number of points constituting the spectrum is controlled by "num1Dpoints" parameter in settings.ini. The default value is commonly large enough and thus need not to be adjusted.

3.13.4 Plotting multiple files and weighted spectrum

In Multiwfn, it is possible to simultaneously plot spectrum for multiple files, and meantime taking weights into account. This feature is quite useful for obtaining actual spectrum of flexible molecule with many thermally accessible conformations.

For example, there is a molecule containing four accessible conformations, their distribution ratio have been determined according to Boltzmann's method based on calculated free energies. If you want to plot its weighted spectrum and spectrum of each conformation, you should write a plain text file named *multiple.txt* (other file name cannot be recognized by Multiwfn), the content is:

```
boltz\Excit\ a.out 0.6046  
boltz\Excit\ b.out 0.1950  
boltz\Excit\ c.out 0.1686  
boltz\Excit\ d.out 0.0317
```

where the first and second column correspond to path of input file and weight of each conformation. Different type of input file can be presented together in this file. If you use this file as input file, after you entered spectrum plotting module, Multiwfn will load data from these files in turn. When plotting spectrum, spectrum of all the four conformations will be separately calculated and drawn with different colors. In addition, the weighted spectrum is plotted on the graph as thick red curve, it is simply evaluated as follows:

$$\text{weighted spectrum} = 0.6046 \times a + 0.1950 \times b + 0.1686 \times c + 0.0317 \times d$$

In the graph you can also observe many black discrete lines, these are collection of discrete lines corresponding to different conformations. Notice that their heights have already been multiplied by corresponding weights.

If before plotting spectrum you have chosen option 18 once, then the spectrum of various conformations shown on graph are the weighted ones; in other words, these curves represent

contribution of each conformation to actual spectrum (thick red curve). In this case, the color of weighted discrete lines are no longer all black but in accordance with color of curves, so that user can easily recognize correspondence between curve and discrete among different systems.

If you simply want to simultaneously plot spectrum for multiple systems but do not want to draw weighted spectrum, then the second column of *multiple.txt* should not be weights, but customized legends, for example

```
boltz\Excit\a.out Molecular A  
boltz\Excit\b.out Molecular B  
boltz\Excit\c.out Molecular C  
boltz\Excit\d.out Molecular D
```

Then plot spectrum as usual, curve of all the four systems will be shown together, and the legends will be "Molecular A", "Molecular B", etc.

When the legend is completely composed of numbers, the legend will be regarded as weight. If you the legend should be a number, you should add \$ in front of the legend to let Multiwfn know it is legend rather than weight. For example, \$50 will be recognized as legend of "50".

Note that the files included in *multiple.txt* may contain different number of transitions, however, in this case the file containing the maximum number of transitions must be placed at the beginning of *multiple.txt*.

If you are using Linux platform and some file paths in the *multiple.txt* contains / symbol or space, you should add double quotation marks at the two ends of the file path, so that the path could be properly loaded.

Many examples of plotting various kinds of spectra are given in Section 4.11.

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 technique proposed by Bader was firstly used for analyzing electron density in "atoms in molecules" (AIM) theory, which is also known as "the quantum theory of atoms in molecules" (QTAIM), this technique has 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 can be more than

(e.g. Li_2) or less than (e.g. KrH^+) the latter.

(3,-1): Two eigenvalues of Hessian matrix of function are negative, namely the second-order saddle point. For electron density analysis, (3,-1) generally appears between attractive atom pairs and hence commonly called as 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*, p11); 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 first-order saddle point (like transition state in potential energy surface). For electron density analysis, (3,+1) generally appears in the center of ring system and displays steric effect, hence (3,+1) is often named as 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 P_4 molecule), hence is often referred to as cage critical point (CCP).

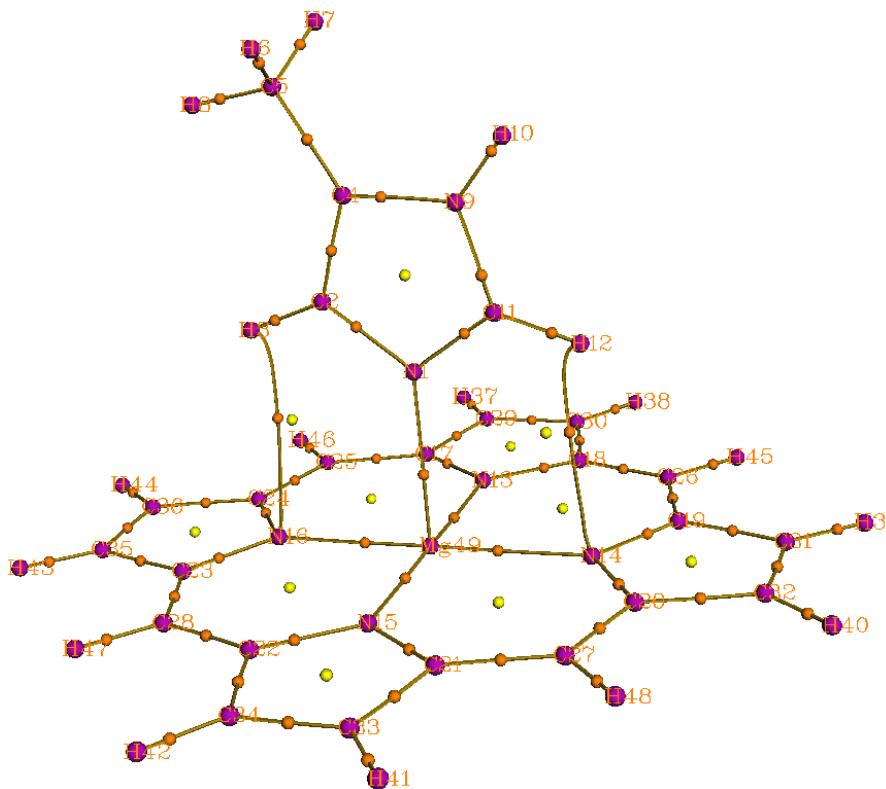
The positions of CPs are searched by Newton method, one need to assign an initial guess point, then the Newton iteration always converge to the CP that is closest to the guess point. By assigning different guesses and doing iteration for each of them, all CPs could be found. Once searches of CPs are 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 does not 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 as “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.

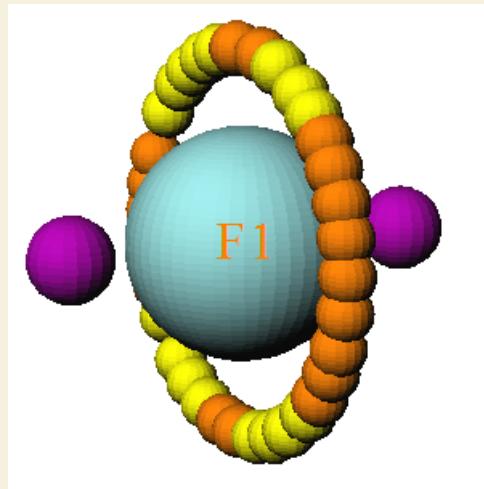
Let us see an example. In the complex shown below, the imidazole plane is vertical to magnesium porphyrin plane, the nitrogen in imidazole coordinated to magnesium. Magenta, orange and yellow spheres correspond to (3,-3), (3,-1) and (3,+1) critical points, brown lines denote bond paths.



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 degenerate CP

Appearance of degenerate CPs in general system is very rare, so I mention it in the final of this section. The so-called degenerate CP is the CP have one or two zero eigenvalues of Hessian matrix. These CPs are very unstable, slight perturbation of molecular geometry can break them into other type CPs. However, degenerate 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 degenerate 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

Searching modes

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

(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. You can also input two atomic indices, then corresponding midpoint will be taken as starting point for locating CP.

(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 taken 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. If you choose option -2, then you will be prompted to input indices of atoms, the nuclei of these atoms will be used as sphere center in turn.

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 try to locate possible CPs.** Note that each time you carry out the search the positions of starting points are different, if some interesting CPs were not found in previous searches, try to launch the search again and again to locate missing CPs.

Beware that the index of the CPs located by searching mode 6 may be different in each time of execution!

CP searching parameters

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. These parameters can be set in the option "-1 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 of 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 will be 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 and occurrence of CPs in the regions far from system. Default value is appropriate for most cases.

(9) Set value range for reserving CPs: By making use of this option, during the CP searching, only the CP with value within user-defined range will be reserved, and thus unnecessary CPs could be ignored. By default all CPs are reserved. Via this option you can for example only search the CPs in weak interaction region (corresponding to low electron density).

(10) Set the atoms to be considered in searching modes 2, 3, 4, 5: If you only want to search CPs in a local molecular region, you can use this option to define the fragment, then only the atoms in this fragment will be taken into account when using searching modes 2, 3, 4, 5.

3.14.3 Generate topology paths

Path generating methods

Once CPs have been found, you can choose to generate topology paths, which connect various CPs. In Multiwfn, the paths are essentially represented as a bunch of points evenly distributed in a curve. Using options 8 and 9, two kinds of paths can be generated:

(1) Generate the paths connecting (3,-3) and (3,-1) CPs: Arithmetically, firstly displacing the coordinate of each (3,-1) forward and backward respectively along the eigenvector that corresponds to the positive eigenvalue of Hessian, and then go uphill following the gradient vector until encounter a (3,-3), the resulting trajectories constitute the bond paths.

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

Path generating parameters

In the option "-2 Set path generating parameters", there are some suboptions used to adjust parameters for generating paths.

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

(2) Stepsize: The space between neighbour points that constitutes the paths. For paths with large curvature, sometimes they cannot be generated under default stepsize, you need to properly decrease the stepsize and try to regenerate paths. If you set maximum number of points and stepsize of paths as m and n respectively, then the maximum length of paths will not longer than $m \times n$.

(3) Stop generation if distance to any CP is smaller than: During the generation of paths, if the distance between current point and a located CP is found to be smaller than this value, then the path will be regarded as connected to the CP.

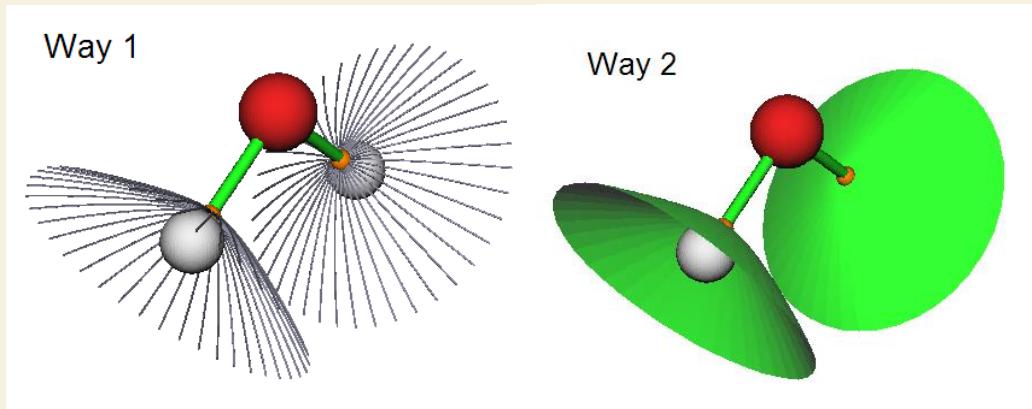
3.14.4 Generate interbasin surfaces

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

Visualize results (option 0)

If you select the function "0 Print and visualize all generated CPs, paths and interbasin 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.

Evaluate CP properties (option 7)

You can obtain values of all real space functions supported by Multiwfn (as well as gradient and Hessian matrix of the selected real space function) at a given CP or at all CPs by function "7 Show real space function values at specific CP or all CPs". Note that electrostatic potential is the most expensive one among all of the real space functions, if you are not interested in it, you can set "ishowptESP" parameter in "settings.ini" to 0 to skip calculation of electrostatic potential.

In this function, you can also decompose a selected real space function at a given CP into contributions from a range of orbitals. See corresponding example in Section 4.2.4.

Measure geometry (option -9)

Distance, angle and dihedral angle between atoms and CPs can be conveniently measured by using the option "-9 Measure distances, angles and dihedral angles between CPs or atoms".

Manipulate CPs (option -4)

In the option “-4 Modify or export CPs”, you can print, add, delete and export CPs. In the suboption 2, the CPs can be deleted according to various condition, including: (1) CP type (2) CP index range (3) Distance to specific molecular fragment (thus you can remove CPs in irrelevant regions) (4) Range of electron density (using this feature you can solely delete CPs with low density and thus remove the CPs in weak interaction regions, or delete CPs with high density and thus remove those in the chemical bond regions).

The positions and types of all found CPs can be saved to a formatted text file *CPs.txt* in current folder by suboption 4. The information 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 such as VMD. In this file, element C/N/O/F correspond to (3,-3)/(3,-1)/(3,+1)/(3,+3) respectively.

Manipulate topology paths (option -5)

In the option “-5 Modify or print detail or export paths, or plot property along a path”, you can print summary of generated paths and print coordinate of all points in a specific path via suboptions 1 and 2, respectively.

By suboption 4 you can export out the detail information of all paths to *paths.txt* in current folder. By suboption 5, the paths can be imported 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.

By suboption 7, value of real space functions along selected topology paths can be plotted as curve map or exported as plain text file. You can also take the data corresponding to "Dist." and "Value" column as X and Y axes respectively to plot curve graph by third-part plotting softwares.

Paths can be manually deleted via suboption 3 by directly inputting their indices. Suboption 8 is also used to delete paths, it is designed for only retaining bond paths (and corresponding BCPs) connecting two fragments but removing all other bond paths, so that one can more easily study interfragment interactions via AIM method. The usefulness of this option will be illustrated in the example in Section 4.2.6.

3.14.6 Calculate the aromaticity indices based on topology properties of electron density

If the real space function you selected is electron density, options 20 and 21 will be visible, they are utilities used to analyze aromaticity.

Shannon aromaticity index

Option 20 is used to calculate Shannon aromaticity (SA) index, see *Phys. Chem. Chem. Phys.*, **12**, 4742 (2010) for detail. The formula can be briefly written as

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

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. The smaller the SA index, the more aromatic is the molecule. The range of $0.003 < SA < 0.005$ is chosen as the boundary of aromaticity/antiaromaticity in original paper.

Curvature of electron density perpendicular to ring plane at RCP

In *Can. J. Chem.*, **75**, 1174 (1997), the authors showed that electron density at RCP is closely related to aromaticity of corresponding ring. The larger the density, the stronger the aromaticity. They also pointed out that the curvature of electron density perpendicular to the ring plane at the RCP has more significant correlation with the ring aromaticity. The more negative the curvature, the stronger the aromaticity. Assume that the ring is strictly perpendicular to a Cartesian axis, for example, a ring perpendicular to Z axis (viz. the ring is in XY plane), then the curvature is just ZZ component of Hessian matrix of electron density at the RCP, which can be directly obtained by using option 7 to print out properties of a given RCP. However, if the ring is distorted, or is not exactly perpendicular to any Cartesian axis, you should use option 21 to calculate the curvature. In option 21, you need to input the index of the RCP (or directly input coordinate of a point), and then input indices of three atoms, their nuclei position will be used to define the ring plane. Then, the electron density, the gradient and the curvature of electron density perpendicular to the ring plane at the RCP will be outputted on screen. In addition, unit normal vector, the coordinates of the two points above and below 1 Angstrom of the RCP in normal direction will be outputted together, which can be taken as the point used to calculate NICS(1).

Information needed: GTFs, atom coordinates

3.15 Quantitative analysis of molecular surface (12)

Quantitative analysis of molecular surface is a powerful tool, it has a lot of practical applications, such as predicting reactive sites, predicting molecular properties, interpreting intermolecular weak interaction. The theory and numerical algorithm involved in present module have been detailedly described in my paper [*J. Mol. Graph. Model.*, **38**, 314 \(2012\)](#). In the next two sections, these two aspects will only be briefly introduced.

3.15.1 Theory

In Multiwfn, in principle, distribution of any real space function on molecular surface (or the surface defined by isosurface of a certain function) can be quantitatively studied. Electrostatic potential and average local ionization energy on molecular vdW surface are particularly useful, therefore they will be discussed detailed in this section. Same kind of quantitative analyses could

also be applied to other real space functions, such as user-defined functions, electron delocalization range function (EDR) and even Fukui function and dual descriptor.

Electrostatic potential on vdW surface

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 commonly performed on molecular van der Waals (vdW) surface. Although the definition of such a surface is arbitrary, most people tend to take the 0.001 a.u. isosurface of electron density as vdW surface, since this definition reflects specific electron structure features of a molecule, such as lone pairs and π electrons, this is also what the definition used in our analyses.

The analysis of ESP on vdW surface has been further quantified to extract more information. 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 analyzing the magnitude and positions of minima and maxima on the surface. Politzer and coworkers (*J. Mol. Struct. (THEOCHEM)*, **307**, 55) 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 density, boiling point, surface tension, heats of vaporization and sublimation, LogP, impact sensitivity, diffusion constant, viscosity, solubility, solvation energy and so on. Below I enumerate and briefly describe these descriptors.

A^+ and A^- indicate the surface area in which the ESP has positive and negative value, respectively. Total surface area A is the sum of them.

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

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

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

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

Π is the average deviation over the surface, which is viewed as an indicator of internal 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:

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

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

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

When σ_+^2 equals to σ_-^2 , ν attains its maximum value of 0.250. The closer the 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 $\nu \sigma_{\text{tot}}^2$ is an indicative of a molecule that has relatively strong tendencies to interact with others of its own kind electrostatically.

In Multiwfn, above mentioned surface descriptors can be calculated not only on the whole vdW surface, but also on the subregions corresponding to atoms or user-defined fragments. The detail of the theory is to be published, and thus not documented here at the moment. In addition, these surface descriptors can be calculated for other kinds of real space functions.

Some practical applications of GIPF descriptors

- Predicting heat of vaporization and heat of sublimation

A practical application of above GIPF descriptors is presented in *J. Phys. Chem. A*, **110**, 1005 (2006). The authors showed that for a series of molecules containing C, H, N and O elements, the heat of vaporization can be well evaluated as

$$\Delta H_{\text{vap}} = a\sqrt{A} + b\sqrt{\nu \sigma_{\text{tot}}^2} + c$$

with least-squares fit coefficients $a = 2.130$, $b = 0.930$ and $c = -17.844$. The heat of sublimation can be predicted as

$$\Delta H_{\text{sub}} = aA^2 + b\sqrt{\nu \sigma_{\text{tot}}^2} + c$$

with $a = 0.000267$, $b = 1.650087$ and $c = 2.966078$. In above equations the surface area A is in \AA^2 , ΔH_{sub} is in kcal/mol, σ_{tot}^2 is in (kcal/mol)². Note that the coefficients are more or less dependent on the calculation level used. The author used B3LYP/6-31G* to optimize geometry and B3LYP/6-311++G(2df,2p) to calculate ESP.

- Predicting density of molecular crystal

Another typical application of statistical data of ESP on vdW surface is predicting crystal density of organic molecules containing C, H, N and O elements. The crystal density is a crucial property of energetic compounds. In *J. Phys. Chem. A*, **111**, 10874 (2007), it was shown that the density can be estimated by $\rho = M / V_m$, where M is molecular mass and V_m is molecular vdW volume defined by $\rho=0.001$ isosurface; for ionic crystal (e.g. ammonium azide), M and V_m

correspond the sum of mass and volume of the cation and anion comprising a formula unit of the compound. Although the relationship is quite simple, it indeed works well for most neutral species, but the error is evidently larger for ionic species. In order to improve the prediction accuracy for neutral ones, in *Mol. Phys.*, **107**, 2095 (2009), the authors introduced GIPF descriptors into the formula:

$$\rho = \alpha \frac{M}{V_m} + \beta(\nu\sigma_{tot}^2) + \gamma$$

At B3PW91/6-31G** level, the fitted coefficients are $\alpha = 0.9183$, $\beta = 0.0028$ and $\gamma = 0.0443$. This formula has proven to have improved accuracy, because intermolecular electrostatic interaction is somewhat effectively taken into considered. In a succeeding paper *Mol. Phys.*, **108**, 1391 (2010), the author showed that the crystal density of ionic compounds can be estimated much better than

$\rho = M / V_m$ if GIPF descriptors are introduced:

$$\rho = \alpha \frac{M}{V_m} + \beta \left(\frac{\bar{V}_{S(\text{cation})}^+}{A_{(\text{cation})}^+} \right) + \gamma \left(\frac{\bar{V}_{S(\text{anion})}^-}{A_{(\text{anion})}^-} \right) + \delta$$

with least-squares fit coefficients $\alpha = 1.0260$, $\beta = 0.0514$, $\gamma = 0.0419$ and $\delta = 0.0227$ at B3PW91/6-31G** level. In the equation, $\bar{V}_{S(\text{cation})}^+$ and $A_{(\text{cation})}^+$ denote the \bar{V}_S^+ and A^+ of the cation; $\bar{V}_{S(\text{anion})}^-$ and $A_{(\text{anion})}^-$ denote the \bar{V}_S^- and A^- of the anion. For 30 test cases, the average absolute error is merely 0.033g/cm³.

Noticed that above relationships are only appropriate for small organic compounds containing C, H, N, O elements, the error are significantly larger for other kinds of systems.

- Predicting boiling point

In *J. Phys. Chem.*, **97**, 9369 (1993), it is shown that boiling point can be predicted as

$$T_{bp} = \alpha A + \beta \sqrt{\nu\sigma_{tot}^2} + \gamma$$

where $\alpha = 2.736$, $\beta = 33.31$, $\gamma = -72.05$ were fitted at HF/STO-5G*/HF/STO-3G* level. This paper also showed equations for predicting critical temperature, volume and pressure.

- Predicting solvation free energy

In *J. Phys. Chem. A*, **103**, 1853 (1999), the prediction equation for solvation free energy is presented as (V_{min} denotes the ESP value at its global minimum in the whole space):

$$\begin{aligned} \Delta G_{\text{solv}} (\text{kJ/mol}) = & 0.17201 V_{\text{min}} - 2.6412 \times 10^{-5} (V_{\text{S,max}} - V_{\text{S,min}})^3 \\ & + 0.051892 A^- \bar{V}_S^- + 9704.2 / (A^- \bar{V}_S^-) + 46.827 \end{aligned}$$

- Predicting other properties

In addition, the equations used to predict heat of fusion, surface tension and crystal/liquid density can be found in *J. Phys. Chem.*, **99**, 12081 (1995), the equations used to predict lattice energy for ionic crystal containing NH₄⁺, K⁺ and Na⁺ are given in *J. Phys. Chem. A*, **102**, 1018 (1998). More

formulae used to predict physical properties of organic molecules based on GIPF descriptors are summarized in Table 3 of *J. Mol. Struct. (THEOCHEM)*, **425**, 107 (1998). GIPF also has many important uses in study of biochemical systems, see *Int. J. Quantum Chem.*, **85**, 676 (2001) for a review.

Average local ionization energy and local electron affinity on vdW surface

Average local ionization energy, $\bar{I}(\mathbf{r})$, has attracted more and more attentions, see corresponding part of Section 2.6 for brief introduction. 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 users are recommended to take a look at *J. Mol. Model.*, **16**, 1731 and Chapter 8 of the book *Theoretical Aspects of Chemical Reactivity* (2007).

Local electron affinity EA_L is a quantity very similar to $\bar{I}(\mathbf{r})$, the only difference is that the MOs under consideration is not all occupied ones, but all unoccupied ones. It was shown that EA_L on molecular surface is useful for analyzing nucleophilic attacking, for detail see *J. Mol. Model.*, **9**, 342.

Quantitative analysis on molecular surface for Fukui function and orbital overlap distance function $D(\mathbf{r})$ have also been proven to be fairly useful, see Sections 4.12.4 and 4.12.8 for example, respectively.

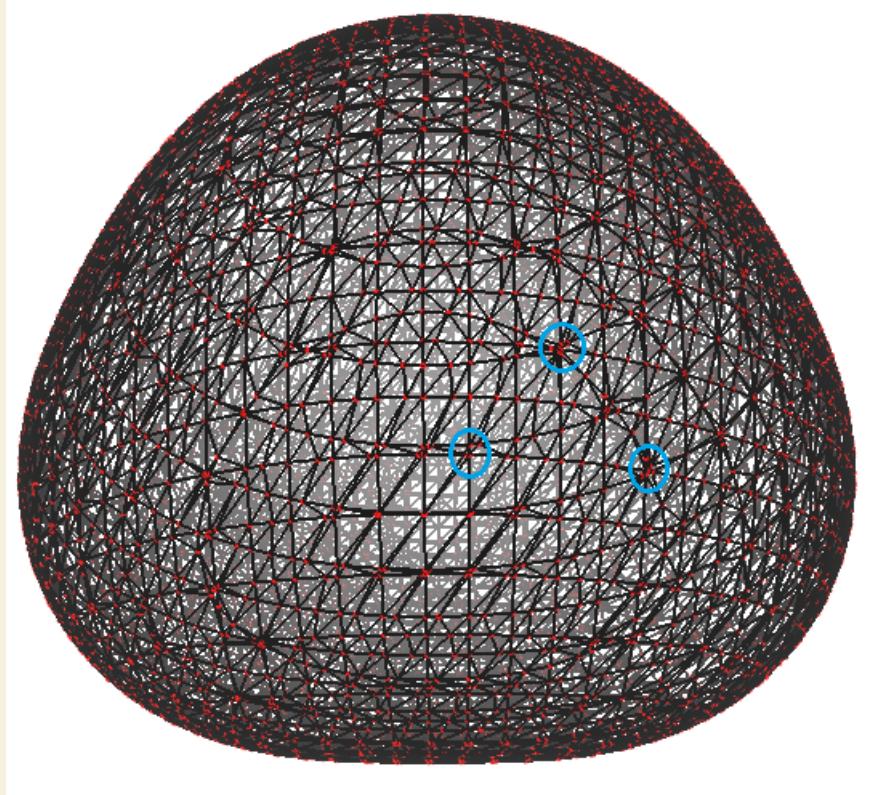
3.15.2 Numerical algorithm

In summary, in the quantitative analysis of molecular surface task, what we need to obtain are minima and maxima of a selected real space function (*i.e.* mapped function) on vdW surface (or isosurface of a specific real space function), as well as \bar{V}_S^+ , \bar{V}_S^- , \bar{V}_S , Π , σ_+^2 , σ_-^2 , σ_{tot}^2 , ν , and $\nu\sigma_{\text{tot}}^2$. 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 is used, the more accurate 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 does not cost much computational time. The volume enclosed by the isosurface

is computed at the same time. This step generates vertices representing the isosurface, along with their connectivity. Each neighbouring three vertices constitute a triangle (will be referred to 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 overall computational time, Multiwfn eliminates redundant points automatically. Specifically, if the distance between two points is smaller than a specific value, one of the points 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 (ESP, $\bar{I}(\mathbf{r})$ and so on) at each vertex on the isosurface. For ESP, this is the most time consuming step; however for such as $\bar{I}(\mathbf{r})$ and EAL , 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 , v , and $v\sigma_{\text{tot}}^2$, 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

$$\bar{V}_S = (1 / A_{\text{tot}}) \sum_{i=1}^N A_i F_i$$

where N is the total number of facets, A_{tot} is the sum of area of all facets, A_i is the area of facet i , F_i is the ESP value (or value of other kind of mapped function) of facet i , which is calculated as the average of ESP at the three vertices composing the facet.

3.15.3 Parameters and options

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. Currently ESP, $\bar{I}(\mathbf{r})$, EA_L and user defined real space function (Section 2.7) are directly supported and can be automatically computed by internal module of Multiwfn during the surface analysis. Alternatively, if you choose "A function loaded from external file", then the value of mapped functions at all surface vertices will be loaded from an external file, so that you can analyze other mapped functions (e.g. Fukui function), see the description of option 5 below for detail about the format of the external file.

Note that by default the ESP is evaluated based on wavefunction, this process may be quite time-consuming for large systems. However, if you choose "Electrostatic potential from atomic charge", then Multiwfn will evaluate ESP based on atomic charges, the computational time will be reduced by several orders. The atomic charges will be loaded from a .chg file, you can use main function 7 to produce the .chg file, or you can write the file manually, see corresponding part of Section 2.5 on the .chg file format. Note that the analysis result will be reasonable only when the the way to generate atomic charges can reproduce ESP well (e.g. CHELPG, MK and ADCH methods).

3 Spacing of grid points 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 are accurate and reliable. If you find some extrema were not located under default spacing, try to decrease spacing and rerun the task.

4 Advanced options: 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 criterion for merging adjacent vertices. Commonly, 0.4~0.5 times of grid spacing is recommended to be used as the criterion.

(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 decrease it to 2 even to 1 to save computational time, however decrease it to 0 will frequently lead to false surface extrema.

5 If load mapped function values from external file

Via this option, during surface analysis, the value of mapped function at all surface vertices can be loaded from external file rather than directly calculated by Multiwfn. This option has two purposes: (1) Reduce overall analysis time (2) Analyze some special functions that cannot be directly produced by Multiwfn, or those requiring mathematical operations and thus not directly available in present module (e.g. dual descriptor). This option has four suboptions:

(0) Do not load mapped function but directly calculate by Multiwfn: This is default case.

(1) Load mapped function at all surface vertices from plain text file: If this option is selected, then after generation of molecular surface, coordinate of all surface vertices (in Bohr) will be automatically exported to *surfptpos.txt* in current folder. Then you can use your favourite program to calculate mapped function value at these points, and write the values as the fourth column in this file (in free format, unit is in a.u.). For example

```
1324          // The first line is the total number of points
-1.6652369 -0.5480503 -0.2554867 -0.0196978306
-1.6835983 -0.5563165 -0.1342924 -0.0242275610
-1.6977125 -0.5530614 -0.0099311 -0.0287667191
-1.7013207 -0.5536310  0.1194197 -0.0330361826
-1.6954099 -0.5523031  0.2547866 -0.0371580132
....
```

Then input the path of this file (the name can still be *surfptpos.txt*), Multiwfn will directly read the values. An exemplificative application of this option is given in Section 4.12.4.

Hint: If you will analyze a system twice or more times, and want to avoid calculating mapped function values every time for saving time, at the first time you analyze the system, you can select option 7 at post-process interface to export coordinates and corresponding mapped function values of surface vertices to *vtx.txt* in current folder. At next time you analyze the system, if you choose this option, and input the path of the *vtx.txt* during the surface analysis, then mapped function values will be directly loaded rather than re-calculated (See Section 4.12.1 for example).

(2) Similar to 1, but specific for the case of using cubegen utility of Gaussian: A file named *cubegenpt.txt* will be generated in current folder after generation of molecular surface. This file is very similar to *surfptpos.txt*, the difference is that in this file the first line is not presented, and coordinate unit is in Angstrom. Based on this file, you can make use of cubegen utility of Gaussian to calculate mapped function at all surface vertices. After that, input the path of cubegen output file, the data will be loaded by Multiwfn.

(3) Interpolate mapped function from an external cube file: After generation of molecular surface, a template cube file named *template.cub* will be generated in current folder. Then you will be prompted to input the path of a cube file representing the mapped function you are interested in, the grid setting of this cube file must be exactly identical to *template.cub*. The mapped function values at surface vertices will be evaluated by interpolation from the cube file you provided.

3.15.4 Options at post-process stage

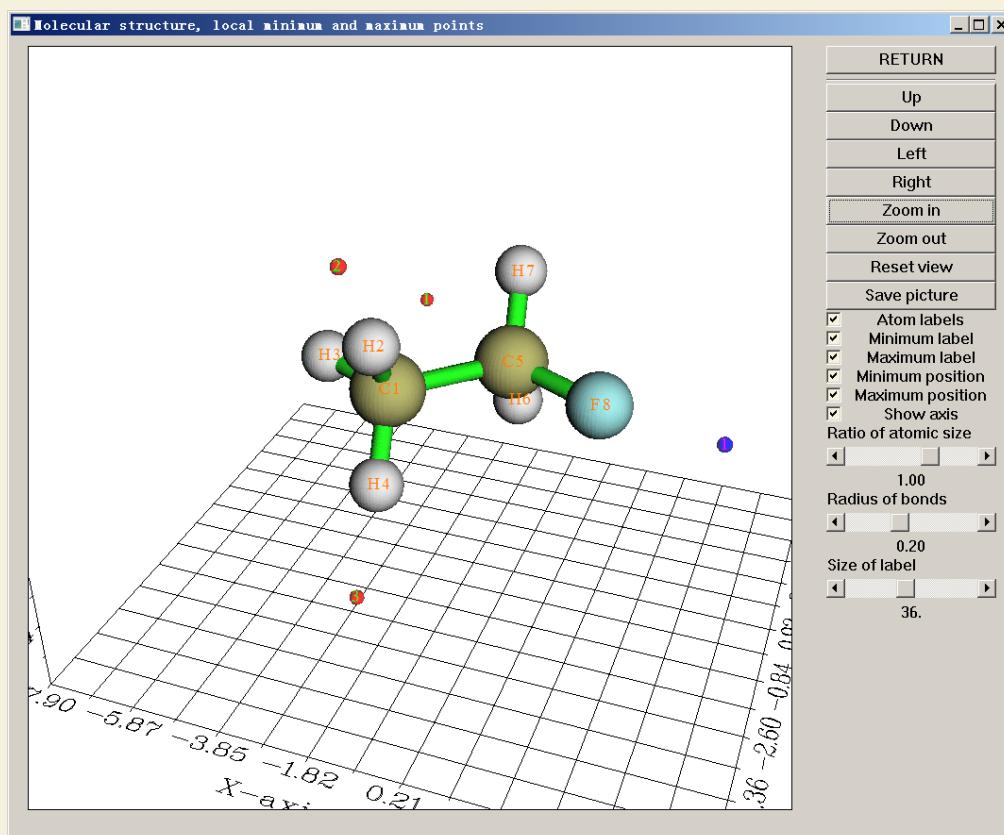
Once the all calculations of surface analysis are finished, a summary will be printed on screen. Meanwhile below options will appear on screen used to check, adjust and export results.

-3 Visualize the surface: By this option you can directly visualize the isosurface analyzed.

-2 Export the grid data to *surf.cub* in current folder: The grid data used to generated isosurface will be exported to the cube file *surf.cub* in current folder.

-1 Return to upper level menu

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 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 maximum 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 present molecule as pdb format file: This option output 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 all surface vertices to vtx.pdb in current folder: This option outputs the surface vertices to vtx.pdb file in current folder, mapped function values are written to B-factor field. This option is mainly used to check validity of isosurface polygonization and visualize distribution of mapped function on vdW surface.

There is a hidden option 66, which not only outputs surface vertices to vtx.pdb, but also outputs connectivity into CONECT field of this file. If you would like to visualize connectivity based on the vtx.pdb in VMD program, please consult my blog article "Setting connectivity of atoms according to CONECT field in VMD" (<http://sobereva.com/121>, in Chinese)

7 Export all surface vertices to vtx.txt in current folder: Namely output all reserved surface vertices to a plain text file named vtx.txt in current folder, including vertex X/Y/Z coordinates in Bohr, mapped function values in a.u., eV and kcal/mol.

9 Output surface area in specific value range of mapped function

By this option, one can gain the knowledge of the distribution of molecular surface area in various range of mapped function. First one needs to input the index range of the atoms in consideration, then input overall range, interval and the unit. For example, one sequentially inputs 2,6-9, -45,50 and 10 and 3, the statistic then is applied on the local molecular surface corresponding to atoms 2, 6, 7, 8 and 9, the output looks like below:

Begin	End	Center	Area	%
-50.0000	-40.0000	-45.0000	4.8764	1.5171
-40.0000	-30.0000	-35.0000	28.0413	8.7242
-30.0000	-20.0000	-25.0000	23.2699	7.2397
-20.0000	-10.0000	-15.0000	17.6022	5.4764
-10.0000	0.0000	-5.0000	61.4759	19.1263
0.0000	10.0000	5.0000	72.6197	22.5933
10.0000	20.0000	15.0000	55.2707	17.1957
20.0000	30.0000	25.0000	53.9060	16.7712
30.0000	40.0000	35.0000	2.8590	0.8895
40.0000	50.0000	45.0000	1.5000	0.4667
Sum:			321.4212	100.0000

where "begin" and "end" is the lower and upper limit of local value range, respectively. "Center" is their average value. Area is in Å², "%" denotes the proportion of the area in overall molecular surface

area.

10 Output the closest and farthest distance between the surface and a point

In this option, after defining a point (you can define a nuclear position or geometry center as the point, you can also directly input the coordinate of the point), the closest and farthest distance between the molecular surface and the point will be outputted. These two quantities have two main uses:

(1) In atoms in molecules (AIM) theory, for systems in gas phase, the vdW isosurface is defined as the $\rho=0.001$ a.u. isosurface. The closest distance between a nucleus and the surface can be regarded as non-bonded atomic radius. For a noncovalently interacting atom pair AB, The difference between the length of A-B and the sum of their non-bonded radii is termed as mutual penetration distance. In general, the larger the distance is, the stronger the interaction will be.

(2) The farthest distance between molecular surface and geometry center can be viewed as a definition of molecular radius. Of course, the concept of molecular radius is only meaningful for sphere-like molecules.

If you input f , Multiwfn will output the farthest distance between all surface points. This can be regarded as a definition of molecular diameter.

11 Output the surface properties of each atom

By default, the surface properties, such as average of ESP, are calculated on the whole molecular surface. By using this function, the whole surface will be first decomposed to local surfaces corresponding to individual atoms, and then the surface properties will be calculated on these atomic surfaces. This function is very helpful to study atomic properties.

After outputting the surface properties, the user can select if output the surface facets to locsurf.pdb in current folder. If choose "y", then in the outputted pdb file, each atom corresponds to a surface facet, and the B-factor field records the attribution of the surface facets, e.g. a facet having B-factor of 11.00 means the facet belongs to the local surface of atom 11. If you load the pdb file into VMD and set "Coloring Method" as "Beta", then you can directly visualize how the whole molecular surface is decomposed to atomic surfaces.

12 Output the surface properties of specific fragment

Similar to function 11, but user can define a fragment, the surface properties will only be calculated on the local surface corresponding to this fragment, so that one can study fragment properties according to the local surface descriptors. Also, you can choose to output the surface facets to locsurf.pdb in current folder, in which the atom having B-factor of 1 and 0 means corresponding surface facet belongs to and does not belong to the local surface of the fragment you defined, respectively.

13 Calculate grid data of mapped function and export it to mapfunc.cub

For example, if the mapped function you selected before the quantitative surface analysis is ALIE, then if you select this option in post-process menu, grid data of ALIE will be calculated and exported to *mapfunc.cub* in current folder, the grid setting is the same as the one employed in the quantitative surface analysis. Based on the *surf.cub* exported by option -2 and the *mapfunc.cub*, you can plot color-mapped isosurface map via VMD program. Section 4.12.6 illustrates value of this option.

Numerous examples of quantitative molecular surface analysis can be found in Section 4.12.

3.15.5 Special topic: Hirshfeld and Becke surface analyses

The quantitative molecular surface analysis module is also able to carry out Hirshfeld and Becke surface analysis, present section is devoted to introduce this point.

Theory of Hirshfeld and Becke surface analyses

Hirshfeld surface analysis was first proposed in *Chem. Phys. Lett.*, **267**, 215 (1997) and comprehensively reviewed in *CrystEngComm*, **11**, 19 (2009). This method focus on analyzing the so-called Hirshfeld surface to reveal weak interactions between monomers in complex or in molecular crystal.

Hirshfeld surface in fact is a kind of inter-fragment (or inter-monomer) surface, which is defined based on the concept of Hirshfeld weight. Probably Hirshfeld surface is the most reasonable way to define inter-fragment surface.

Atomic Hirshfeld weighting function of an atom is expressed as

$$w_A^{\text{Hirsh}}(\mathbf{r}) = \frac{\rho_A^0(\mathbf{r})}{\sum_B \rho_B^0(\mathbf{r})}$$

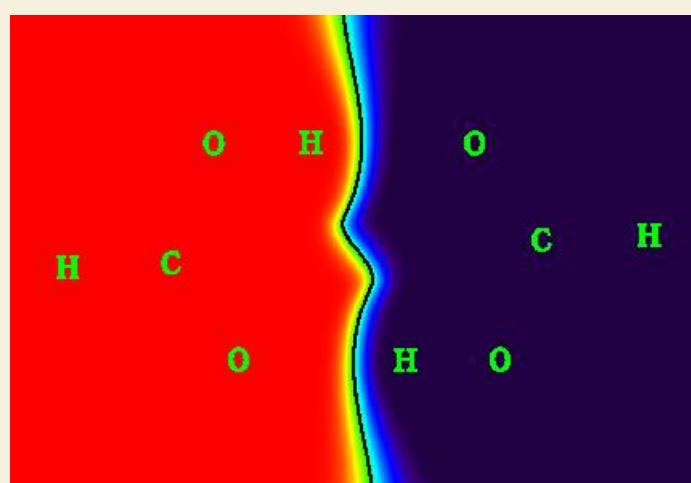
where ρ_A^0 denotes the density of atom A in free-state. Summing up weight of all atoms in a fragment yields Hirshfeld weight of this fragment

$$w_P^{\text{Hirsh}}(\mathbf{r}) = \sum_{A \in P} w_A^{\text{Hirsh}}(\mathbf{r})$$

Hirshfeld surface of fragment P is just the isosurface of $w_P^{\text{Hirsh}} = 0.5$.

Motivated by Hirshfeld surface, I proposed Becke surface, which replaces Hirshfeld weight with Becke weight (see Section 3.18.0 for introduction of Becke weight), only geometry and atomic radii are required to construct Becke surface. Generally the shape of Becke surface and Hirshfeld surface are comparable, but I prefer to use Becke surface, mostly because Becke surface does not rely on atomic density and thus somewhat easier to be constructed.

To intuitively illustrate Hirshfeld/Becke surface, acetic acid dimer in 2D case is taken as example here



Becke weighting function of the monomer at left side of the graph is represented by color, going

from red to purple corresponds to the weight varying from 1.0 to 0.0. The black line, which is the contour line of 0.5, is just its Becke surface. Evidently, Becke surface very elegantly partitioned the whole space into two monomer regions, the difference of atomic size is properly and automatically taken into account. The Becke surface in this case is an open surface, the surface extends to infinity; while if the monomer is completely buried, such as in molecular crystal or metal-organic framework environment, then its Becke surface will be a close surface and encloses all of its nuclei, just like common molecular surface.

If we map specific real space functions on Becke/Hirshfeld surface and study their distributions, just like quantitative analysis on molecular surface, we can gain many important information about intermolecular interaction. There are three real space functions very useful for this purpose

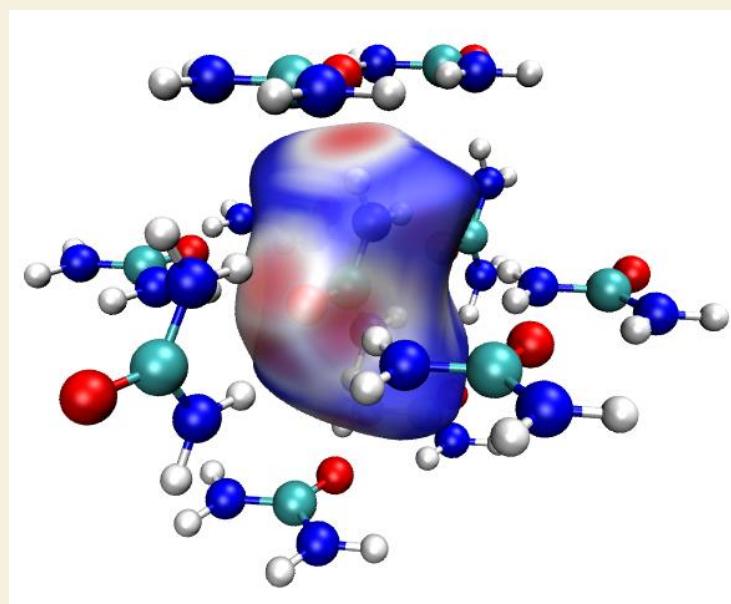
$$(1) \text{ Normalized contact distance } d_{\text{norm}} = \frac{d_i - r_i^{\text{vdW}}}{r_i^{\text{vdW}}} + \frac{d_e - r_e^{\text{vdW}}}{r_e^{\text{vdW}}}, \text{ where } d_i (d_e) \text{ is the}$$

distance from a point on the surface to the nearest nucleus inside (outside) the surface, r_i^{vdW} and r_e^{vdW} denote vdW radius of the corresponding two atoms. Small value of d_{norm} indicates close intermolecular contact and implies evident interaction.

(2) Electron density. If electron density is large in some local region of Becke/Hirshfeld surface, obviously the intermolecular interactions crossing these regions must be strong. The usefulness of electron density is similar to d_{norm} , but the former is more physically meaningful. While d_{norm} also has its own advantage, namely only geometry and vdW radii information are needed.

(3) $\text{Sign}[\lambda_2(\mathbf{r})]$, see corresponding part Section 2.6 for detailed explanation. This function can not only exhibit interaction strength but also reveal interaction type.

Below is urea crystal, the isosurface represents Hirshfeld surface of the central urea, and d_{norm} is the mapped function. Red parts correspond to small d_{norm} and thus exhibit close contact, which mainly originates from H-bond interaction.



Fingerprint plot

The so-called "fingerprint plot" defined in the framework of Hirshfeld surface analysis is very useful for investigating the noncovalent interactions in molecular crystals. X and Y axes in this plot correspond to d_i and d_e , respectively. The plot is created by binning (d_i, d_e) pairs in certain intervals and colouring each bin of the resulting 2D histogram as a function of the number of surface points in that bin, ranging from purple (few points) through green to red (many points). Black region means there are not surface points. If you are confused, see the fingerprint plot in Section 4.12.6 for example. The usefulness of fingerprint plot is demonstrated in page 24 and 25 of *CrystEngComm*, **11**, 19 (2009). Although in Multiwfn fingerprint plot can also be drawn in Becke surface analysis, the plot in this case is not as useful as that in Hirshfeld surface analysis.

The Hirshfeld surface in fact can be viewed as the contact surface between the atoms in the Hirshfeld fragment you defined and all of the other atoms. The remarkable flexibility of Multiwfn allows the overall contact surface be decomposed to many local contact surfaces and draw the corresponding fingerprint plots. For example, in Multiwfn one can draw fingerprint plot for the local contact surface between the nitrogen in the central urea and the hydrogens in specific peripheral ureas. In this case, all surface points in the contact surface satisfy two conditions: (1) In the Hirshfeld fragment (*viz.* the central urea), the atom closest to the surface point is the nitrogen (2) Among all of the atoms that do not belong to the Hirshfeld fragment, the atom closest to the surface point is one of hydrogen in specific peripheral ureas. Undoubtedly, fingerprint plot of local contact surface greatly facilitates one to study the noncovalent interaction in local region due to the contact between specific atoms.

Usage

The procedure to perform Becke/Hirshfeld surface analysis is similar to normal quantitative molecular surface analysis. After you entered main function 12, choose option 1 and select Becke or Hirshfeld surface, then input the index of the atoms in the fragment (referred to as *Hirshfeld fragment* below), and use option 2 to select mapped function. After that choose 0 to start calculation. Quantitative data on the surface such as average value and standard deviation will be outputted, and surface extrema will be located. Then via corresponding options you can visualize surface minima/maxima, export result and so on, all of the options in post-process menu (except one, see below) have already been introduced in Section 3.15.4 and thus are not described again here. If you use option 6 to export the surface vertices to pdb file, based on which you can also plot color-mapped Hirshfeld/Becke surface by means of VMD program, the drawing method should be set to "Points" and coloring method should be set to "Beta" (*viz.* B-factor in the pdb file, which records mapped function value).

Becke surface analysis and Hirshfeld surface analysis commonly work equally well, the computational cost of the latter is lower.

Current Multiwfn does not support .cif file. So if you would like to use Multiwfn to perform Hirshfeld/Becke surface analysis to study intermolecular interaction in molecular crystals, you may need to use some third-part softwares to open the corresponding .cif file and then properly dig out a cluster of molecules.

In order to draw fingerprint plot, the mapped function must be set to d_{norm} , which is default. Then after the Hirshfeld/Becke surface analysis is finished, in the post-process menu you will see an option titled "20 Fingerprint plot analysis", after enter it you will see a menu. If you select option

0, the fingerprint analysis will start. By default the fingerprint analysis is performed for the overall Hirshfeld surface. If you intend to study fingerprint plot in local contact region, you should use options 1 and 2 to properly define the "inside atoms" set and "outside atoms" set, only the contact surface between these two sets will be taken into account in the fingerprint plot analysis. The "inside atoms" set must be a subset of Hirshfeld fragment, while the atoms in "outside atoms" set must not belong to the Hirshfeld fragment. In options 1 and 2 you will be asked to input two filter conditions, their intersection defines the set. Condition 1 corresponds to the atom index range, the condition 2 corresponds to the type of element. For example, you inputted *I*,*3-6* and *Cl*, then among atoms 1, 3, 4, 5, 6 if there are Cl atoms they will be defined as the set. If you directly press ENTER button when inputting condition 1, for "inside atoms" set and "outside atoms" set the condition will be defaulted to all atoms in Hirshfeld fragment and all atoms external to Hirshfeld fragment, respectively. If directly pressing ENTER button for condition 2, the condition will be defaulted to all elements. For example, when you define "inside atoms" set, if you want to only select nitrogens in the Hirshfeld fragment, simply press ENTER button for condition 1 and input *N* for condition 2. Another example, if you want atoms 1~4, 7, 8 to be presented in the set regardless of their elements, you should input *I-4,7,8* for condition 1 and then directly press ENTER button for condition 2

After the fingerprint analysis is carried out, in post-process menu, by options 1~3 you can draw the fingerprint plot, save the plot, and export the original data of the plot (namely the number of surface points in each bin) to plain text file. By options 3 and 4 you can adjust color scale and the range of X/Y axes, respectively. By option 5 you can export the surface points corresponding to the current fingerprint plot to *finger.pdb* in current folder. In addition, by options 10 and 11, you can draw scatter map for the surface points between d_i and d_e , and export d_i and d_e values of all surface points to plain text file.

An example of Becke surface analysis is given in Section 4.12.5. An example of Hirshfeld analysis and drawing fingerprint plot is given in Section 4.12.6, this example also illustrates how to easily plot pretty color-mapped Hirshfeld/Becke surface via VMD program.

Information needed: Depending on the real space function used to define the surface and that mapped on the surface. At least atom coordinates must be provided. (For local electron affinity, virtual MOs must be presented, hence such as .fch, .molden and .gms should be used)

3.16 Process grid data (13)

If Gaussian-type cube file (.cub) or DMol3 grid file (.grd) was loaded when Multiwfn boots up, or grid data has been generated by main function 5 or other functions, a set of grid data will be presented in memory (which will be referred to as "present grid data" below), then this module will be available. If grid data has not been presented in memory but you choose this main function, you can also directly load grid data from an external file.

In this module, you can visualize present grid data, extract data in a specified plane, perform mathematical algorithm, set value in specified range by corresponding options and so on. These options will be described below.

3.16.0 Visualize isosurface of present grid data (-2)

Visualize isosurface of present grid data in a GUI window, this is useful to check validity of the grid data updated by some functions (e.g. function 11)

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 information will be outputted to a cube file.

3.16.2 Output all data points with value and 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 points 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 points in a plane defined 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 1 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 points 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

$k+abs(k)*0.03$ 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 -2 to visualize the updated grid data, use function 0 to output the updated grid data as cube file or extract data in a plane by function 2~9, etc.

Supported operations are shown below, where A means value of present grid data, B means value at corresponding point in the cube file that will be loaded. C means the updated value at corresponding point.

- | | |
|---|-----------------|
| • 1 Add a constant | e.g. A+0.1=C |
| • 2 Add 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. 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 |
| • 18 Add a grid file multiplied by a value | i.e. A+0.4*B=C |
| • 19 The same as 6, but multiplied by a weighting function at the same time. The weighting function is defined as $\min(A , B)/\max(A , B)$. So, at any point, the more the magnitude of the data in A deviates from the counterpart in B, the severely the result will be punished. | |
| • 20 Multiplied by a coordinate variable: This option multiplies all grid data by one of selected coordinate variables X, Y and Z. | |
| • 21 Get minimal value with another function | i.e. min(A,B) |
| • 22 Get $\min(A , B)$ | |

If the operation you selected involves a constant, you will be prompted to input its value; If involves another cube or Dmol3 .grd 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 memory.

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 to 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 has 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 the grid points that far away from / close to some atoms (13)

By this function, the value of grid points beyond or within scaled vdW region of a molecular fragment can be set to a specific value. This is very useful for screening uninteresting region when showing isosurface, namely setting value of this region to a very large value (very positive or very negative, according to the character of the grid data).

You need to input a scale factor for vdW radius, then input expected value. After that, you need to specify fragment, you can either directly input atomic indices (e.g. 3,5,1-15,20), or input filename of a plain text file, in which a molecular fragment is defined as atomic list, below is an example of the file:

```
3
1 3 4
```

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

Then all grid points that beyond the region occupied by scaled vdW spheres of the fragment atoms will be set to specific value.

If the scale factor of vdW sphere is set to a negative value, e.g. -1.3, then all grid points that within the scaled vdW surface of the fragment will be set to the specific value.

An example of using this function is given in Section 4.13.4.1.

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

This function is similar to function 13, but only the grid points outside superposition region of scaled vdW regions of two fragments will be set to a specified value. You can either directly input atomic indices of the two fragments, or prepare two files containing atom lists for the two fragments, the format is the same as function 13.

This function is very useful if you are only interested in studying isosurfaces between two fragments, all isosurfaces outside this region can be screened by setting grid data value to very large. An illustrative example is given in Section 4.13.4.2.

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 of the points in specific spatial and value range (17)

This function can output statistic data of the points in specific spatial and value range. If user do not want to impose any constraint (namely the statistic data is for all data points), input 1. If constraint is needed to be imposed, user should input 2, and then input lower and upper limit of X,Y,Z coordinates and value in turn.

The minimum and maximum value, average, root mean square, standard deviation, volume, sum, integral and barycenter position of the data points satisfied the constraints will be outputted. The positive, negative and total barycenter are computed respectively as

$$\begin{aligned}\mathbf{R}_+ &= \sum_i^+ \mathbf{r}_i f(\mathbf{r}_i) / \sum_i^+ f(\mathbf{r}_i) \\ \mathbf{R}_- &= \sum_j^- \mathbf{r}_j f(\mathbf{r}_j) / \sum_j^- f(\mathbf{r}_j) \\ \mathbf{R}_{tot} &= \sum_k^{all} \mathbf{r}_k f(\mathbf{r}_k) / \sum_k^{all} f(\mathbf{r}_k)\end{aligned}$$

where f is the data value, \mathbf{r} denotes coordinate vector, the indices i, j and k run over positive, negative and all grid points respectively.

3.16.14 Calculate and plot integral curve in X/Y/Z direction (18)

Integral curve is defined as below (e.g. in Z direction)

$$I(z') = \int_{z_{ini}}^{z'} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x, y, z) dx dy dz$$

Local integral curve may be defined as (e.g. in Z direction)

$$I_L(z) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x, y, z) dx dy$$

Evidently

$$I(z') = \int_{z_{ini}}^{z'} I_L(z) dz$$

In Multiwfn, I and I_L curves are evaluated based on cubic numerical integration of grid data. User first needs to choose which direction will be integrated, and then input the lower and upper limit of the coordinate in this direction. Then integral curve and local integral curve will be calculated within this range. After that, by corresponding options, the graph of the curves can be plotted or saved, the curve data can be exported to current folder as plain text file.

Assume that user has chosen Z direction, and the lower and upper limit were set to -5 and 10, respectively, then the spatial range of the curve generated by Multiwfn will be $z=[-5, 10]$, and the z_{ini} in above formula will be -5. If user input the letter a (denotes "all"), then the minimal and maximal value of Z coordinate of current grid data will regarded as lower and upper limit, respectively.

Worthnotingly, if the integrand is chosen as electron density difference, then the integral curve sometimes is known as "charge displacement curve" and useful in discussion of charge transfer, see *J. Am. Chem. Soc.*, **130**, 1048 for example. If you want to obtain such a curve, before entering present function, you should calculate grid data of electron density difference, or directly load the grid data from external file (e.g. cube file).

A practical example is given in Section 4.13.6.

Information needed: Grid data, atom coordinates

3.17 Adaptive natural density partitioning (AdNDP) analysis (14)

3.17.1 Theory

Famous NBO analysis developed by Weinhold and coworkers is able to recover up to 3-centers 2-electrons (3c-2e) orbitals from density matrix (e.g. by using "3cbond" keyword in NBO program). Adaptive natural density partitioning (AdNDP), which was proposed by Zubarev and Boldyrev in year 2008 (*Phys. Chem. Chem. Phys.*, **10**, 5207), may be viewed as a natural extension of NBO analysis aiming for locating $N > 3$ centers orbitals. AdNDP has been extensively used to study electronic structure characteristic of widespread of cluster systems, by googling "AdNDP" you can

find many related literatures.

Canonical molecular orbitals (CMOs) are generally highly delocalized, often lacking of chemical significances; While 2c or 3c NBOs are substantially localized, for highly conjugated system resonant description is often needed (otherwise large non-Lewis composition will occur, that means current system is inappropriate to be portrayed by single set of NBOs), this somewhat conflicts with modern quantum chemistry concepts and obscures delocalization natural of electrons in conjugated system. AdNDP orbitals seamlessly bridged CMOs and NBOs, AdNDP bonding patterns avoid resonant description and are always consistent with the point symmetry of the molecule.

The basic idea of AdNDP to generate multi-center orbitals is very similar to NBO analysis, that is constructing proper sub-block of density matrix in natural atomic orbital (NAO) basis and then diagonalize it, the eigenvalues and eigenvectors correspond to occupation number and orbital wavefunction respectively. For example, we want to generate all possible 4-centers orbitals for atom A,B,C,D, we first pick out corresponding sub-blocks and then combine them to together:

$$P^{(A,B,C,D)} = \begin{bmatrix} P_{A,A} & P_{A,B} & P_{A,C} & P_{A,D} \\ P_{B,A} & P_{B,B} & P_{B,C} & P_{B,D} \\ P_{C,A} & P_{C,B} & P_{C,C} & P_{C,D} \\ P_{D,A} & P_{D,B} & P_{D,C} & P_{D,D} \end{bmatrix}$$

After diagonalization of $P^{(A,B,C,D)}$, if one or more eigenvalues exceeded the predefined threshold, which is commonly set to close to 2.0 (e.g. 1.7), then corresponding orbitals will be regarded as candidate 4c-2e bonds. Completely identical strategy can be used to generate orbitals with higher number of centers.

Indeed, the orbital generating process of AdNDP is quite easy once atom combination is determined, however the searching process of final N c-2e orbitals in entire system is complicated, manual inspections and operations are necessary. AdNDP approach has large ambiguity, it is possible that the searching process carried out by different peoples finally results in different AdNDP pattern, I think this is the most serious limitation of current AdNDP approach. So, AdNDP is never a black box, before using it users must have preliminary understanding of the searching process of the AdNDP implemented in Multiwfn.

Before the search, densities from core-type NAOs are automatically eliminated from the density matrix, since they have no any contribution to bonding. After that, 1-center orbitals (lone pair), 2-centers orbitals, 3-centers orbitals, 4-centers orbitals ... will be searched in turn, until residual density (trace of density matrix) is close to zero. The search could be exhaustive, that means when searching N -centers orbitals, Multiwfn will construct and diagonalize $M!/(M-N)!/N!$ sub-blocks of density matrix, where M is the total number of atoms. All of the orbitals whose occupation numbers are larger than the threshold will be added to candidate orbital list. For large system, the searching process may be very time-consuming or even forbidden, for example, exhaustive search of 10-centers orbitals in the system with 30 atoms needs to construct and diagonalize 30045015 sub-blocks of density matrix! This is very difficult to be finished in personal computer, for such case, user-directed searching is necessary. In Multiwfn you can customly define a search list, then the exhaustive search will only apply to the atoms in the search list, so that the amount of computation would be greatly reduced. You can also directly let Multiwfn construct and diagonalize sub-block of density matrix for specified atom combination. Note that user-directed searching has relatively

high requirements of skill and experience on users.

Once the search of N -centers orbitals is finished, we will get a list containing candidate N -centers orbitals. We need to pick some of them out as final N -centers AdNDP orbitals. Commonly, one or more orbitals with the highest occupation numbers will be picked out. Notice that, since some densities are simultaneously shared by multiple candidate orbitals, if we directly pick out several candidate orbitals with the largest occupation at one time, the electrons may be overcounted. To avoid this problem, assume that K orbitals with the highest occupation numbers obviously overlap with some other candidate orbitals meanwhile there is no evident overlapping between the K orbitals, we should first pick out K orbitals as final AdNDP orbitals, then Multiwfn will automatically deplete their density from the density matrix and then reconstruct and diagonalize the corresponding sub-blocks of density matrix for remained candidate orbitals to update their shapes and occupation numbers. If there are still some candidate orbitals with occupation numbers close to 2.0, you may consider picking them out, then remained orbitals will be updated again. Such process may be repeated several times until there is no orbitals have high occupation numbers. After that, you can start to search $N+1$ centers orbitals.

The general requirements of AdNDP analysis are that: The final residual density (corresponds to non-Lewis composition in NBO analysis) should as low as possible; the occupation numbers of each AdNDP orbital should as close to 2.0 as possible; the number of centers of AdNDP orbitals should as less as possible; the resulting orbitals must be consistent with molecular symmetry.

However, there is no unique rule on how to search orbitals and pick out candidate orbitals as AdNDP orbitals. For example, one can first search 5-centers orbitals before completing search of 3-centers orbitals, and one can also directly search 6-centers orbitals after the search of 2-centers orbitals has finished. The sequence of picking out candidate orbitals is also not necessarily always in accordance to magnitude of occupation numbers. The final AdNDP pattern obtained by different operations may be different, how to do AdNDP analysis is largely dependent on users themselves. Actually some molecules may have two or even more equally reasonable AdNDP patterns, sometimes it is difficult to discriminate which pattern is the best. I have confidence to say that some AdNDP patterns presented in published papers are not the optimal ones. The experience of using AdNDP approach can be gradually accumulated in practices and during reading related literatures.

AdNDP is very insensitive to basis set quality as NBO analysis, 6-31G* is enough to produce accurate results for main group elements in the first few rows. Over enhancing basis set quality will not improve AdNDP analysis results but only lead to increase of the computational burden in diagonalization step, since the size of sub-block of density matrix is directly determined by size of basis set.

Multiwfn offers capacity of evaluating AdNDP orbital energies. You need to provide a file containing Fock (or Kohn-Sham) matrix in original basis functions. The Fock matrix can be obtained from output of Gaussian or other programs. The energy of AdNDP orbital is corresponding diagonal term of Fock matrix in AdNDP orbital representation. Specifically, Multiwfn performs below representation transform:

$$\mathbf{F}_{\text{AdNDP}} = \mathbf{C}^T \mathbf{F}_{\text{AO}} \mathbf{C} \quad \mathbf{C} = \mathbf{X}_{\text{AONAO}} \mathbf{c}$$

where \mathbf{F}_{AO} is the Fock matrix in original basis function that loaded from user-provided file, $C(r,i)$ corresponds to coefficient of basis function r in AdNDP orbital i . $c(s,i)$ corresponds to coefficient of NAO s in AdNDP orbital i . $\mathbf{X}_{\text{AONAO}}$ is transformation matrix between original basis function and

NAO, i.e. $X(t,s)$ is coefficient of basis function t in NAO s . Energy of AdNDP orbital j is simply $F_{\text{AdNDP}}(j,j)$, which is expectation value of Fock operator of AdNDP orbital wavefunction.

3.17.2 Input file

The output file of NBO program containing density matrix in NAO basis (DMNAO) can be used as input file for AdNDP analysis. If you also need to visualize AdNDP orbitals or export orbitals as cube files, .fch file must be provided, meanwhile transformation matrix between NAO and original basis functions (AONAO) must be presented in the NBO output file.

Assume that you are a Gaussian user, in order to obtain an Gaussian output file containing all information needed by Multiwfn to perform the AdNDP analysis and visualization, you should write a Gaussian input file of single point task with *pop=nboread* keyword in route section, and write **\$NBO AONAO DMNAO \$END** after molecular geometry section with a blank line as separator. Then run the input file by Gaussian and then convert .chk file to .fch format by formchk utility.

The Gaussian output file (not .fch file) should be used as the initial input file when Multiwfn boots up. Once you entered AdNDP module, Multiwfn will load NAO information and DMNAO matrix from this file. If then you choose corresponding options to visualize or export orbitals, AONAO matrix will be loaded and the program will prompt you to input the path of the .fch file (if .fch is in the same folder and has identical name as the Gaussian output file, then the .fch will be automatically loaded).

Multiwfn is also compatible with the output files of stand-alone NBO program (GENNBO), of course you have to add *DMNAO* keywords in \$NBO section in .47 file. In this case it is impossible to visualize AdNDP orbitals.

Formally, AdNDP approach is also applicable to open-shell systems; of course, the occupation threshold should be divided by 2. When you enter AdNDP module, Multiwfn will ask you which density matrix should be used, the so-called total density matrix is the sum of alpha and beta density matrix.

Notice that if after you entered AdNDP module Multiwfn suddenly crashes, and the basis set you used contains diffuse functions, you can try to use another basis set without diffuse functions. This problem is caused by the bug in NBO 3.1 module, namely in rarely cases the DMNAO output may be slightly problematic if diffuse functions present. Since AdNDP analysis is quite insensitive to diffuse functions, they can be safely removed without any loss of accuracy.

If you want to obtain AdNDP orbital energies, Fock matrix corresponding to present system at the same calculation level must be provided in lower-triangular sequence in a plain text file, namely: $F(1,1) F(2,1) F(2,2) F(3,1) F(3,2) F(3,3) \dots F(nbasis,nbasis)$, where $nbasis$ is the total number of basis functions. The format is free. If you are a Gaussian user, you can add *archive file=XXX* keyword between \$NBO ... \$END, then in the resulting XXX.47 file, search \$FOCK and copy all data between \$FOCK ... \$END to a plain text file, then this file can be directly used to provide Fock matrix to Multiwfn (In fact, Multiwfn is also able to automatically locate and read the \$FOCK field when the file name has .47 suffix).

3.17.3 Options

All of the options involved in AdNDP module are introduced below, some options are invisible

in certain cases. If current candidate orbital list is not empty, then all candidate orbitals will always be printed on screen in front of the menu (except when you select option 5 or 13), the candidate orbital indices are determined according to occupation numbers. The number of residual valence electrons of all atoms in the search list is always printed at the upper of the menu, this value decreases with gradually picking out candidate orbitals as final AdNDP orbitals. If this value is very low (e.g. lower than 1.4), it is suggested that new $Nc-2e$ AdNDP orbitals will be impossible to be found between the atoms in the search list.

AdNDP analysis is more complex than other functions in Multiwfn, if you are confused, please do the exercises in Section 4.14, you will learn the basic steps of AdNDP analysis.

-10 Return to main menu: Once you choose this option, you will return to main menu, meantime all results of AdNDP analysis will be lost. Hence the status of AdNDP module can be reseted by choosing this option and then re-entering the module.

-2 Switch if output detail of exhaustive searching process: If this option is switched to "Yes", then details of exhaustive searching process will be printed during searching

-1 Define exhaustive search list: In this option, one can customly define search list, exhaustive search (option 2) will only apply to the atoms in the search list. All commands in this defining interface are self-explained. Notice that default search list includes all atoms of the molecule.

0 Pick out some candidate orbitals and update occupations of others: Users need to input two numbers, e.g. i, j , then all candidate orbitals from i to j will be picked out and saved as final AdNDP orbitals. If user only input one number, e.g. k , then k candidate orbitals with the largest occupation numbers will be picked out. After that, the eigenvectors (orbital shape) and eigenvalues (occupation numbers) of remained candidate orbitals will be updated as mentioned earlier.

1 Perform orbitals search for a specific atom combination: Users need to input indices of some atoms, e.g. 3,4,5,8,9, then sub-block of density matrix for atom 3,4,5,8,9 will be constructed and diagonalized, all resultant eigenvectors will be added to candidate orbitals list, meantime all previous candidate orbitals will be removed. There is no limit on the number of inputted atoms.

2 Perform exhaustive search of N-centers orbitals within the search list: N atoms will be selected out from the search list in an exhaustive manner, assume that the search list contains M atoms, then totally $M!/(M-N)!/N!$ atom combinations will be formed. For each combination, corresponding sub-block of density matrix will be constructed and diagonalized, all eigenvectors with eigenvalues larger than user-defined threshold will be add to candidate orbital list. Old candidate orbital list will be cleaned.

3 Set the number of centers in the next exhaustive search: Namely set the value N in option 2. Once exhaustive search of N -centers orbitals is finished, N will be automatically increased by one.

4 Set occupation threshold in the next exhaustive search: Namely set the threshold used in option 2.

5 Show information of all AdNDP orbitals: Print occupation numbers and involved atoms of all saved AdNDP orbitals.

6 Delete some AdNDP orbitals: Input two numbers, e.g. i, j , then saved AdNDP orbitals from i to j will be removed.

7 Visualize AdNDP orbitals and molecular geometry: The path of corresponding .fch file will be prompted to be inputted, after loading some necessary information from the file, a GUI window will pop up and molecular geometry will be shown. Isosurfaces of AdNDP orbitals can be

plotted by clicking corresponding numbers in the right-bottom list

8 Visualize candidate orbitals and molecular geometry: Analogous to option 7, but used to visualize isosurfaces of candidate orbitals. It is useful to visualize the isosurfaces before picking out some candidate orbitals as final AdNDP orbitals.

9 Export some AdNDP orbitals to Gaussian-type cube files: User need to input two numbers, e.g. 2,6, then wavefunction value of AdNDP orbitals from 2 to 5 will be respectively exported to AdNDPorb0002.cub, AdNDPorb0003.cub, AdNDPorb0004.cub and AdNDPorb0005.cub in current folder, they are Gaussian-type cube files and can be visualized by many softwares such as VMD. The total number of grid points is controlled by "nprevorbgrid" in settings.ini. The length left in each direction is controlled by "Aug3D" in settings.ini.

10 Export some candidate orbitals to Gaussian-type cube files: Analogous to option 9, but used to export cube files for candidate orbitals.

11/12 Save/Load current density matrix and AdNDP orbital list: Option 11 is used to save current density matrix and AdNDP orbital list in memory temporarily, when density matrix and AdNDP orbital list is changed, you can choose option 12 to recover previous state.

13 Show residual density distributions on the atoms in the search list: After choosing this option, population number of each atom in the search list will be calculated according to present density matrix and then printed out. If some neighboring atoms have large population number, it is suggested that multi-center orbitals with high occupation number may appear on these atoms; while the atoms with low population number often can be ignored in the following searching process. Thus this option is very helpful for setting up user-directed searching.

14 Output all AdNDP orbitals as .molden file: The path of corresponding .fch file will be prompted to be inputted, after that all saved AdNDP orbitals will be outputted as AdNDP.molden in current folder (see Section 2.5 for introduction of .molden format). By using this file as input file, you can conveniently analyze composition of AdNDP orbitals by main function 8. Note that if there are N basis functions and M AdNDP orbitals have been picked out, then the first M orbitals in the AdNDP.molden will correspond to the M AdNDP orbitals, while the other $N-M$ orbitals in this file are meaningless and can be simply ignored.

16 Output energy of picked AdNDP orbital: This function is used to evaluate energy of AdNDP orbitals that have already been picked out. Multiwf will prompt you to input the path of the file containing Fock matrix in original basis functions, the elements of the matrix should be recorded in lower-triangular sequence, the NBO .47 file containing \$FOCK field can also be directly used as input file. Then after a simple transformation, orbital energies are immediately outputted.

Some examples of AdNDP analysis are given in Section 4.14.

Information needed: NBO output file (with AONAO DMNAO keywords), .fch file (only needed when visualizing and exporting cube file for AdNDP orbitals, or exporting AdNDP orbitals as .molden file), plain text file (containing Fock matrix. Only needed if you want to gain orbital energies)

3.18 Fuzzy atomic space analysis (15)

3.18.0 Basic concepts

Before introducing each individual function, here I first introduce some basic concepts of fuzzy atomic space.

Atomic space is the local space attributed to specific atom in the whole three-dimension molecular space. Below we will express atomic space as weighing function w . The methods used to partition the whole space into atomic spaces can be classified to two categories:

1 Discrete partition methods: The two representative methods are Bader's partition (also known as AIM partition) and Voronoi partition. They partition molecular space discretely, so any point can be attributed to only one atom, in other words,

$$\begin{cases} w_A(\mathbf{r}) = 1 & \text{if } \mathbf{r} \in \Omega_A \\ w_A(\mathbf{r}) = 0 & \text{if } \mathbf{r} \notin \Omega_A \end{cases}$$

where Ω_A is atomic space of atom A.

2 Fuzzy partition methods: The representative methods include Hirshfeld, Becke, Hirshfeld-I and ISA. They partition molecular space contiguously, atomic spaces overlap with each other, any point may simultaneously attributed to many atoms to different extent, and the weights are normalized to unity. In other words, below two conditions hold for all atoms and any point

$$\begin{aligned} 0 \leq w_A(\mathbf{r}) \leq 1 & \quad \forall A \\ \sum_B w_B(\mathbf{r}) = 1 & \end{aligned}$$

The most significant advantage of fuzzy partition may be that the integration of real space function in fuzzy atomic space is much easier than in discrete atomic space. By using Becke's numerical DFT integration scheme (*J. Chem. Phys.*, **88**, 2547), high accuracy of integration in fuzzy atomic space can be achieved for most real space functions at the expense of relatively low computation effort. In the fuzzy atomic space analysis module of Multiwfn, all integrations are realized by this scheme. The more integration points are used, the higher integration accuracy can be reached, one can adjust the number of points by "radpot" and "sphpot" parameter in settings.ini.

In fuzzy atomic space analysis module of Multiwfn, one can obtain many properties that based on fuzzy atomic spaces. Currently, the most widely used definitions of fuzzy atomic spaces, namely Hirshfeld, Hirshfeld-I and Becke are supported, they are introduced below. One can choose which fuzzy atomic spaces will be used by option -1.

Hirshfeld atomic space: In *Theor. Chim. Acta (Berl.)*, **44**, 129, Hirshfeld defined the atomic space as

$$w_A^{\text{Hirsh}}(\mathbf{r}) = \frac{\rho_A^{\text{free}}(\mathbf{r} - \mathbf{R}_A)}{\sum_B \rho_B^{\text{free}}(\mathbf{r} - \mathbf{R}_A)}$$

where \mathbf{R} is coordinate of nucleus, ρ^{free} denotes spherically averaged atomic electron density in free-state.

In option -1, you will find two options "Hirshfeld" and "Hirshfeld*". The former uses atomic .wfn files to calculate the weights, they must be provided by yourself or let Multiwfn automatically invoke Gaussian to generate them, see Section 3.7.3 for detail. The latter evaluates the weights directly based on built-in radial atomic densities and thus is more convenient, detail can be found in Appendix 3. I strongly suggest using "Hirshfeld*" instead of "Hirshfeld".

Hirshfeld-I (HI) atomic space: This is a well-known extension of Hirshfeld method, it was proposed in *J. Chem. Phys.*, **126**, 144111 (2007). Commonly the atomic space defined by HI is more physically meaningful than that of Hirshfeld, since it can respond actual molecular environment. Unfortunately, HI is much more expensive than Hirshfeld due to its iterative nature. Details of Hirshfeld-I and its implementation in Multiwfn have been introduced in Section 3.9.13 and thus will not be repeated here. When you choose HI in option -1, Multiwfn will first perform regular HI iterations (If you are confused by the operations, please consult the example of computing HI charges in Section 4.7.4). After HI atomic spaces have converged, you can do subsequent analyses.

Becke atomic space: First, consider a function p

$$p(d) = (3/2)d - (1/2)d^3$$

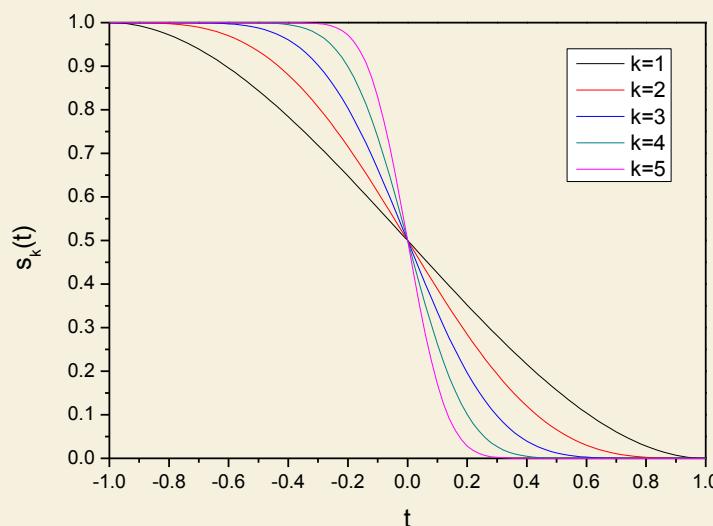
which can be iterated many times

$$\begin{aligned} f_1(d) &= p(d) \\ f_2(d) &= p[p(d)] \\ f_3(d) &= p\{p[p(d)]\} \\ &\dots \end{aligned}$$

Then define a function s

$$s_k(t) = (1/2)[1 - f_k(t)]$$

The plot of s_k versus t is



From above graph it can be seen that s_k gradually reduces from 1 to 0 with t varying from -1 to

1. The larger the k is, the sharper the curve becomes. The weighting function of Becke atomic space is based on simple transformation of s_k , for details please consult original paper *J. Chem. Phys.*, 88, 2547.

$$w_A^{\text{Becke}}(\mathbf{r}) = \frac{P_A(\mathbf{r})}{\sum_B P_B(\mathbf{r})}$$

$$P_A(\mathbf{r}) = \prod_{B \neq A} s_k(\nu_{AB}(\mathbf{r})) \quad \nu_{AB}(\mathbf{r}) = \mu_{AB}(\mathbf{r}) + a_{AB}(1 - \mu_{AB}(\mathbf{r})^2)$$

$$a_{AB} = \frac{u_{AB}}{u_{AB}^2 - 1} \quad u_{AB} = \frac{\chi_{AB} - 1}{\chi_{AB} + 1} \quad \chi_{AB} = \frac{R_A^{\text{cov}}}{R_B^{\text{cov}}}$$

$$\begin{cases} a_{AB} = -0.5 & \text{if } a_{AB} < -0.5 \\ a_{AB} = 0.5 & \text{if } a_{AB} > 0.5 \end{cases}$$

$$\mu_{AB}(\mathbf{r}) = \frac{r_A - r_B}{R_{AB}} \quad R_{AB} = |\mathbf{R}_A - \mathbf{R}_B| \quad r_A = |\mathbf{r} - \mathbf{R}_A| \quad r_B = |\mathbf{r} - \mathbf{R}_B|$$

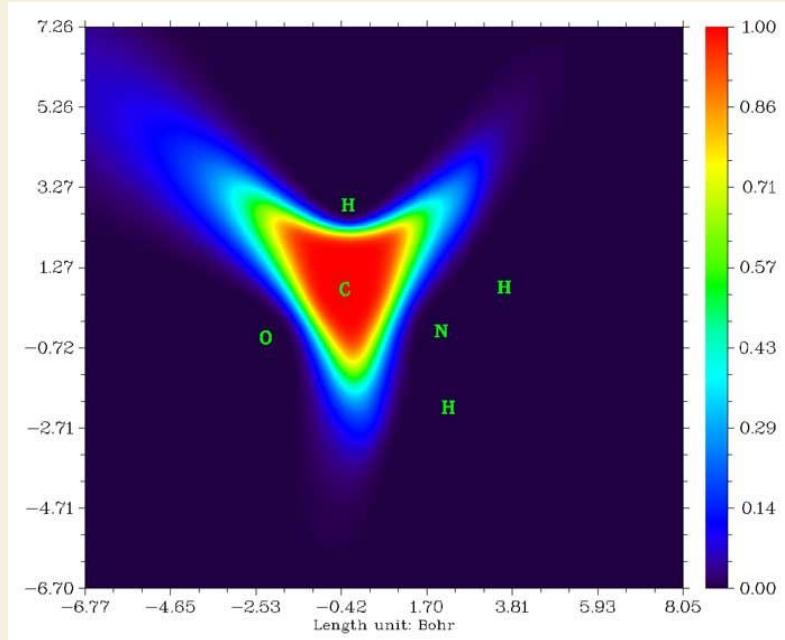
where \mathbf{R} stands for coordinates of nucleus. R^{cov} denotes covalent radius.

The number of iterations, namely k value, can be set by option -3. The default value (3) is appropriate for most cases. The definition of the covalent radius used to generate Becke atomic space can be chosen by option -2. Through corresponding suboptions, one can directly select a set of built-in radii (CSD radii, modified CSD radii, Pyykkö radii, Suresh radii, Hugo radii), load radii information from external plain text file (the format required is described in the program prompts), or modify current radii by manual input.

The origin paper of CSD radii is *Dalton Trans.*, **2008**, 2832, these radii were deduced from statistic of Cambridge Structural Database (CSD) for the elements with atomic numbers up to 96. Pyykkö radii was defined in *Chem. Eur. J.*, **15**, 186, which covers the entire periodic table, Groups 1–18, Z=1–118. Suresh radii was proposed in *J. Phys. Chem. A*, **105**, 5940, which is based on theoretically calculated geometries of H₃C-EH_n, the defined radii cover most of main group and transition elements in periodic table. Hugo radii was proposed in *Chem. Phys. Lett.*, **480**, 127, which has clear physical meaning and is based on atomic ionization energy. Notice that Hugo radii for hydrogen is rather large (even larger than Kr by 0.01 Bohr).

I found it is inappropriate to directly use any covalent radii definition shown above to define Becke's atomic space. The covalent radii of metal elements in IA and IIA groups are always large, e.g. CSD radius of lithium is 1.28 Angstrom. While covalent radii of elements in such as VIIA group are always small, e.g. CSD radius of fluorine is only 0.58 Angstrom. For main groups, the elements with small (large) covalent radius generally have large (small) electronegativity. So, in molecule environment, the atoms with small (large) covalent radius prefer to withdraw (donate) electrons to expand (shrink) their effective size, this behavior makes actual radii of main group elements in each row equalized. In order to faithfully reflect this behavior, I defined the so-called "modified CSD radii", namely the CSD radii of all main group elements (except for the first row) are replaced by CSD radii of the IVA group element in corresponding row, while transition elements still use their original CSD radii. The modified CSD radii are the default radii definition for Becke's atomic space.

The Becke atomic space of carbon in acetamide constructed by default parameters is illustrated below



3.18.1 Integration of a real space function in fuzzy atomic spaces (1)

This function is used to integrate real space function P in atomic spaces

$$I_A = \int_A w_A(\mathbf{r})P(\mathbf{r})d\mathbf{r}$$

For example, if P is chosen as electron density, then I_A will be the electron population number of atom A.

P may be also chosen as the real space functions involving coordinates of two electrons, such as exchange-correlation density and source function. For this case, the coordinate of reference point can be set by option -10 (this is equivalent to set "refxyz" in settings.ini). If you have carried out topology analysis, you can also use -11 to set a critical point as reference point, this is especially convenient for studying source function (for which bond critical point is usually set as reference point).

The "% of sum" and "% of sum abs" in output is defined as $\frac{I_A}{\sum_B I_B} \times 100\%$ and

$$\frac{I_A}{\sum_B |I_B|} \times 100\% \text{ respectively.}$$

By default, all atomic spaces will be integrated. If you only need integral value of certain atoms, you can use option -5 to define the atom list.

Special note: This option uses single-center integration grid to integrate each atom, commonly this is no problem. However, if you choose to use Hirshfeld or Hirshfeld-I partition, and the real space function to be integrated varies very fast around nuclei (e.g. Laplacian of electron density), then this option is unable to give accurate result. You should use option 101 (invisible in the interface) instead, which employs molecular integration grid to integrate every Hirshfeld atoms, the result is always very accurate.

3.18.2 Integration of a real space function in overlap spaces (8)

This function is used to integrate specified real space function P in overlap spaces between atom pairs

$$I_{AB} = \int_A w_A(\mathbf{r}) w_B(\mathbf{r}) P(\mathbf{r}) d\mathbf{r}$$

For example, if P is chosen as electron density, then I_{AB} will be the number of electrons shared by atom A and B. P may be also chosen as the real space functions involving coordinates of two electrons.

Integrations of positive and negative parts of P are outputted separately. Meanwhile, sum of diagonal elements $\sum_A I_{AA}$, sum of non-diagonal elements $\sum_A \sum_{B \neq A} I_{AB}$ and sum of all elements $\sum_A \sum_B I_{AB}$ for positive and negative parts are also outputted together.

Currently only the fuzzy atom space defined by Becke can be used together with this function.

3.18.3 Calculate atomic multipole moments (2)

This function is used to evaluate atomic monopole, dipole, quadrupole moments and octopole moments. All units in the output are in a.u.

In below formulae, superscript A means an atom named A. x, y and z are the components of electron coordinate \mathbf{r} relative to nuclear coordinate \mathbf{R} .

$$x = r_x - R_x^A \quad y = r_y - R_y^A \quad z = r_z - R_z^A$$

and $r^2 = x^2 + y^2 + z^2$.

Atomic monopole moment due to electrons is just negative of electron population number

$$p_A = -\int_A w_A(\mathbf{r}) \rho(\mathbf{r}) d\mathbf{r}$$

Atomic charges are outputted together, namely $q_A = p_A + Z_A$, where Z denotes nuclear charge.

Atomic dipole moment is useful to measure polarization of electron distribution around the atom, which is defined as

$$\mathbf{\mu}^A = \begin{bmatrix} \mu_x^A \\ \mu_y^A \\ \mu_z^A \end{bmatrix} = -\int_A \begin{bmatrix} x \\ y \\ z \end{bmatrix} w_A(\mathbf{r}) \rho(\mathbf{r}) d\mathbf{r}$$

Its magnitude, or say its norm is

$$|\mathbf{\mu}^A| = \sqrt{(\mu_x^A)^2 + (\mu_y^A)^2 + (\mu_z^A)^2}$$

Multiwfn also outputs the contribution of present atom to total molecular dipole moment,

which is evaluated as $q_A \mathbf{R} + \boldsymbol{\mu}^A$.

Cartesian form of atomic quadrupole moment tensor is defined as (see Section 1.8.7 of the book *The Quantum Theory of Atoms in Molecules-From Solid State to DNA and Drug Design*).

$$\boldsymbol{\Theta}^A = \begin{bmatrix} \Theta_{xx}^A & \Theta_{xy}^A & \Theta_{xz}^A \\ \Theta_{yx}^A & \Theta_{yy}^A & \Theta_{yz}^A \\ \Theta_{zx}^A & \Theta_{zy}^A & \Theta_{zz}^A \end{bmatrix} = -(1/2) \int_A \begin{bmatrix} 3x^2 - r^2 & 3xy & 3xz \\ 3yx & 3y^2 - r^2 & 3yz \\ 3zx & 3zy & 3z^2 - r^2 \end{bmatrix} w_A(\mathbf{r}) \rho(\mathbf{r}) d\mathbf{r}$$

The its magnitude can be calculated as

$$|\boldsymbol{\Theta}^A| = \sqrt{(2/3) [(\Theta_{xx}^A)^2 + (\Theta_{yy}^A)^2 + (\Theta_{zz}^A)^2]}$$

Atomic quadrupole moments in Cartesian form can be used to exhibit the deviation of the distribution from spherical symmetry. For example, $\Theta_{xx}^A < 0$ ($\Theta_{xx}^A > 0$) indicates the electron density of atom A elongated (contracted) along X axis. If the atomic electron density has spherical symmetry, then $\Theta_{xx} = \Theta_{yy} = \Theta_{zz}$. Noticeably, the Cartesian quadrupole moment tensor $\boldsymbol{\Theta}$ given here is in traceless form, that means the condition $\Theta_{xx} + \Theta_{yy} + \Theta_{zz} = 0$ always holds.

The atomic quadrupole and octopole moments in spherical harmonic form are also outputted. The general expression of multipole moments in spherical harmonic form is

$$Q_{l,m}^A = - \int_A R_{l,m}^A(\mathbf{r}) w_A(\mathbf{r}) \rho(\mathbf{r}) d\mathbf{r}$$

All of the five components of quadrupole moment in spherical harmonic form correspond to

$$\begin{aligned} R_{2,0} &= (1/2)(3z^2 - r^2) \\ R_{2,-1} &= \sqrt{3}yz \quad R_{2,1} = \sqrt{3}xz \\ R_{2,-2} &= \sqrt{3}xy \quad R_{2,2} = (\sqrt{3}/2)(x^2 - y^2) \end{aligned}$$

All of the seven components of octopole moment in spherical harmonic form correspond to

$$\begin{aligned} R_{3,0} &= (1/2)(5z^2 - 3r^2)z \\ R_{3,-1} &= \sqrt{3/8}(5z^2 - r^2)y \quad R_{3,1} = \sqrt{3/8}(5z^2 - r^2)x \\ R_{3,-2} &= \sqrt{15}xyz \quad R_{3,2} = (\sqrt{15}/2)(x^2 - y^2)z \\ R_{3,-3} &= \sqrt{5/8}(3x^2 - y^2)y \quad R_{3,3} = \sqrt{5/8}(x^2 - 3y^2)x \end{aligned}$$

The magnitude of multipole moments in spherical harmonic form is calculated as

$$|Q_l^A| = \sqrt{\sum_m (Q_{l,m}^A)^2}$$

At the end of the output, the total number of electrons, molecular dipole moment and its magnitude are outputted. Molecular dipole moment is calculated as the sum of the contribution from atomic dipole moments and atomic charges (i.e. the sum of all "Contribution to molecular dipole

moment" terms)

$$\mu^{\text{mol}} = \sum_A (q_A \mathbf{R}^A + \mu^A)$$

By default, atomic multipole moment for all atoms are evaluated. If you only need some of them, you can use option -5 to define the atom list.

3.18.4 Calculate atomic overlap matrix (3)

This function is used to calculate atomic overlap matrix (AOM) for molecular orbitals or natural orbitals in atomic spaces, the AOM will be outputted to AOM.txt in current folder. The element of AOM is defined as

$$S_{ij}(A) = \int_A \varphi_i(\mathbf{r}) \varphi_j(\mathbf{r}) d\mathbf{r}$$

where i and j are orbital indices. Notice that the highest virtual orbitals will not be taken into account. For example, present system has 10 orbitals in total, 7,8,9,10 are not occupied, and user has set the occupation number of orbital 3 to zero by option 26 in main function 6, then the dimension of each AOM outputted by Multiwfn will be (6,6), corresponding to the overlap integral between the first 6 orbitals in each atomic space.

Since orbitals are orthonormal in whole space, in principle, summing up AOMs for all atoms (corresponding to integrating in whole space) should yield an identity matrix

$$\mathbf{SUM} = \sum_A \mathbf{S}(A) = \mathbf{I}$$

Of course, this condition does not strictly hold, because the integration is performed numerically as mentioned earlier. The deviation of **SUM** to identity matrix is a good measure of numerical integration accuracy

$$\text{Error} = \frac{\sum_i \sum_j |\text{SUM}_{i,j} - \mathbf{I}_{i,j}|}{N_{Atom}}$$

Multiwfn automatically calculates and outputs the ERROR value. If ERROR is not small enough, e.g. >0.001, you may want to improve the integration accuracy by increasing "radpot" and "sphpot" in settings.ini.

For unrestricted HF/DFT or unrestricted post-HF wavefunction, AOM for α and β orbitals will be outputted respectively.

3.18.5 Calculate localization and delocalization index (4)

For open-shell systems, the LI (λ) and DI (δ) are calculated for each spin of electrons respectively. Below only the expression of LI and DI for α electrons is given. For β electrons, just replacing α with β , similarly hereinafter. The electrons in atom space A that can delocalize to atom space B is computed as

$$\delta^\alpha(A \rightarrow B) = - \int_A \int_B \Gamma_{\text{XC}}^{\alpha, \text{tot}}(\mathbf{r}_1, \mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2$$

where Γ_{XC} is exchange-correlation density; if you are not familiar with it, please consult the discussion in part 17 of Section 2.6. The electrons in atom space B that can delocalize to atom space A is

$$\delta^\alpha(B \rightarrow A) = - \int_B \int_A \Gamma_{\text{XC}}^{\alpha, \text{tot}}(\mathbf{r}_1, \mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2$$

Clearly, above two terms are identical in value, therefore we define DI between A and B as below, it measures the total number of α electrons shared by atom A and B

$$\delta^\alpha(A, B) = \delta^\alpha(A \rightarrow B) + \delta^\alpha(B \rightarrow A) = -2 \int_A \int_B \Gamma_{\text{XC}}^{\alpha, \text{tot}}(\mathbf{r}_1, \mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2$$

The LI $^\alpha$ measures the number of α electrons localized in an atom. Note that this quantity is not additive.

$$\lambda^\alpha(A) = - \int_A \int_A \Gamma_{\text{XC}}^{\alpha, \text{tot}}(\mathbf{r}_1, \mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 = \delta^\alpha(A, A) / 2$$

The relationship between LI, DI and the population number of electrons in atomic space is given below, the physical meaning is that the sum of α electrons of atom A that localized in atom A and that delocalized to other regions is the total number of α electrons of space A.

$$\begin{aligned} \lambda^\alpha(A) + \sum_{B \neq A} \delta^\alpha(A, B) / 2 &\equiv \lambda^\alpha(A) + \sum_{B \neq A} \delta^\alpha(A \rightarrow B) = \\ &= - \int_A \int \Gamma_{\text{XC}}^{\alpha, \text{tot}}(\mathbf{r}_1, \mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 = \int_A \rho^\alpha(\mathbf{r}) d\mathbf{r} = N_A^\alpha \end{aligned}$$

Using the approximate expression of Γ_{XC} , the DI and LI can be explicitly written as

$$\begin{aligned} \delta^\alpha(A, B) &= 2 \sum_{i \in \alpha} \sum_{j \in \alpha} \sqrt{\eta_i \eta_j} S_{ij}(A) S_{ij}(B) \\ \lambda^\alpha(A) &= \sum_{i \in \alpha} \sum_{j \in \alpha} \sqrt{\eta_i \eta_j} S_{ij}(A) S_{ij}(A) \end{aligned}$$

Total DI and LI are the summation of α part and β part

$$\begin{aligned} \delta(A, B) &= \delta^\alpha(A, B) + \delta^\beta(A, B) \\ \lambda(A) &= \lambda^\alpha(A) + \lambda^\beta(A) \end{aligned}$$

Since in closed-shell case $\delta^\alpha(A, B) = \delta^\beta(A, B)$, one can evaluate total DI as

$$\delta(A, B) = 2\delta^\alpha(A, B) = 2 \times 2 \sum_m \sum_n \sqrt{\frac{\eta_m}{2} \frac{\eta_n}{2}} S_{mn}(A) S_{mn}(B) = 2 \sum_m \sum_n \sqrt{\eta_m \eta_n} S_{mn}(A) S_{mn}(B)$$

where m and n denote closed-shell natural orbitals. Similarly, the total LI for closed-shell cases is

$$\lambda(A) = \sum_m \sum_n \sqrt{\eta_m \eta_n} S_{mn}(A) S_{mn}(A)$$

For closed-shell systems, it is argued that the value of total DI is a quantitative measure of the number of electron pairs shared between two atoms. For example, total $\delta(A, B) = 1.0$ implies a pair of electron (an α and a β electrons) is shared between atom A and B. (In fact, this is strictly true only for nonpolar bonds such as H-H bond in H₂. In polar bonds, the DI must be lower than formal

bond, because what total DI actually reflects is the *effective* number of electron pairs shared by two atoms and thus somewhat reflects covalency. Note that the value of DI is very sensitive the definition of atomic space employed)

For single-determinant wavefunction, because of integer occupation number of orbitals, DI and LI can be simplified as

$$\begin{aligned}\delta^\alpha(A, B) &= 2 \sum_{i \in \alpha}^{\text{occ}} \sum_{j \in \alpha}^{\text{occ}} S_{ij}(A) S_{ij}(B) \\ \delta(A, B) &= 4 \sum_m^{\text{occ}} \sum_n^{\text{occ}} S_{mn}(A) S_{mn}(B) \\ \lambda^\alpha(A) &= \sum_{i \in \alpha}^{\text{occ}} \sum_{j \in \alpha}^{\text{occ}} S_{ij}(A) S_{ij}(A) \\ \lambda(A) &= 2 \sum_m^{\text{occ}} \sum_n^{\text{occ}} S_{mn}(A) S_{mn}(A)\end{aligned}$$

Conventionally, LI and DI are calculated in AIM atomic space (also called as AIM basin). While in fuzzy atomic space analysis module of Multiwfn, they are calculated in fuzzy atomic space, the physical nature is the same. According to the discussion presented in *J. Phys. Chem. A*, **109**, 9904 (compare Eq. 13 and Eq. 18), the DI calculated in fuzzy atomic space is just the so-called fuzzy bond order, which was defined by Mayer in *Chem. Phys. Lett.*, **383**, 368.

For closed-shell system, atomic valence can be calculated as the sum of its fuzzy bond orders

$$V(A) = \sum_{B \neq A} \delta(A, B)$$

In Multiwfn, before calculating LI and DI, AOM is calculated first automatically, this is the most time-consuming step. For open-shell systems, the LI and DI for α and β electrons, as well as for all electrons are outputted respectively. Notice that the diagonal terms of DI matrix are calculated as the sum of corresponding off-diagonal row (or column) elements. For closed-shell system, as stated above, they correspond to atomic valence.

For strictly planar molecules, because overlap integral of σ orbital and π orbital is exactly zero in atomic space, the contributions from σ and π electrons to DI can be exactly decomposed as DI- σ and DI- π

$$\begin{aligned}\delta^\alpha(A, B) &= \delta_\sigma^\alpha(A, B) + \delta_\pi^\alpha(A, B) \\ \delta_\sigma^\alpha(A, B) &= 2 \sum_{i \in \alpha}^{\sigma} \sum_{j \in \alpha}^{\sigma} \sqrt{\eta_i \eta_j} S_{ij}(A) S_{ij}(B) \\ \delta_\pi^\alpha(A, B) &= 2 \sum_{i \in \alpha}^{\pi} \sum_{j \in \alpha}^{\pi} \sqrt{\eta_i \eta_j} S_{ij}(A) S_{ij}(B)\end{aligned}$$

Similarly, LI can be decomposed as LI- σ and LI- π . Summing up corresponding off-diagonal elements in DI- σ and DI- π matrix gives σ -atomic valence and π -atomic valence, respectively. If you want to compute DI/LI- σ (DI/LI- π), before the DI/LI calculation, you should set the occupation

numbers of all π orbitals (σ orbitals) to zero by subfunction 26 of main function 6.

By the way, in some literatures, especially the ones written by Bernard Silvi, the variance of electronic fluctuation in atomic space $\sigma^2(A)$ and the covariance of fluctuation of electron pair between two atomic spaces $\text{cov}(A,B)$ are discussed. They are not directly outputted by Multiwfn, because there is a very simple relationship correlates $\sigma^2(A)$, $\text{cov}(A,B)$ and $\text{DI}(A,B)$, thus you can calculate them quite easily, see *Chem. Rev.*, **105**, 3911 for derivation

$$\text{cov}(A,B) = -\delta(A,B)/2$$

$$\sigma^2(A) = N_A - \lambda(A) = -\sum_{B \neq A} \text{cov}(A,B) = \sum_{B \neq A} \delta(A,B)/2$$

where N_A is the electron population number in A. As mentioned above, the diagonal terms of the DI matrix outputted by Multiwfn are calculated as the sum of off-diagonal elements in the corresponding row (or column), hence you can simply obtain σ^2 by dividing corresponding diagonal term of DI matrix by two.

A quantity closely related to σ^2 is the relative fluctuation parameter introduced by Bader, which indicates the electronic fluctuations for a given atomic space relative to its electron population, you can calculate it manually if you want

$$\lambda_F(A) = \sigma^2(A) / N_A$$

Alternatively, you can calculate below value to measure the proportion of the electrons localized in the atomic space

$$l(A) = \lambda(A) / N_A$$

3.18.6 Calculate PDI (5)

Para-delocalization index (PDI) is a quantity used to measure aromaticity of six-membered rings. PDI was first proposed in *Chem. Eur. J.*, **9**, 400, also see *Chem. Rev.*, **105**, 3911 for more discussion. PDI is essentially the averaged para-delocalization index (para-DI) in six-membered rings.

$$\text{PDI} = \frac{\delta(1,4) + \delta(2,5) + \delta(3,6)}{3}$$

The basic idea behind PDI is that Bader and coworkers reported that DI in benzene is greater for para-related than for meta-related carbon atoms. Obviously, the larger the PDI, the larger the delocalization, and the stronger the aromaticity. The main limitation of the definition of PDI is that it can only be used to study aromaticity of six-membered rings, and it was shown that PDI is inappropriate for the cases when the ring plane has an out-plane distortion.

In Multiwfn, before calculating PDI, AOM and DI are first calculated automatically. Then you will be prompted to input the indices of the atoms in the ring that you are interested in, the input order must be consistent with atom connectivity.

PDI currently is only available for closed-shell systems, although theoretically it may be possible to be extended to open-shell cases.

Note that for completely planar systems, since DI can be decomposed to α and π parts, PDI can also be separated as PDI- α and PDI- π to individually study α aromaticity and π aromaticity. In

order to calculate PDI- α (PDI- π), before enter present module, you should first manually set occupation number of all MOs except for π (α) MOs to zero (or you can utilize option 22 in main function 100 to do this step, which will be much more convenient).

3.18.7 Calculate FLU and FLU- π (6,7)

Aromatic fluctuation index (FLU) was proposed in *J. Chem. Phys.*, **122**, 014109, also see *Chem. Rev.*, **105**, 3911 for more discussion. Like PDI, FLU is an aromaticity index based on DI, but can be used to study rings with any number of atoms. The FLU index was constructed by following the HOMA philosophy (see Section 3.100.13), i.e. measuring divergences (DI differences for each single pair bonded) from aromatic molecules chosen as a reference. FLU is defined as below

$$\text{FLU} = \frac{1}{n} \sum_{A-B}^{\text{RING}} \left[\left(\frac{V(B)}{V(A)} \right)^\alpha \left(\frac{\delta(A, B) - \delta_{\text{ref}}(A, B)}{\delta_{\text{ref}}(A, B)} \right) \right]^2$$

where the summation runs over all adjacent pairs of atoms around the ring, n is equal to the number of atoms in the ring, δ_{ref} is the reference DI value, which is precalculated parameter. α is used to ensure the ratio of atomic valences is greater than one

$$\alpha = \begin{cases} 1 & V(B) > V(A) \\ -1 & V(B) \leq V(A) \end{cases}$$

The first factor in the formula of FLU penalizes those with highly localized electrons, while the second factor measures the relative divergence with respect to a typical aromatic system. Obviously, lower FLU corresponds to stronger aromaticity.

The dependence on reference value is one of main weakness of FLU. The default δ_{ref} in Multiwfn for C-C, C-N, B-N are 1.468, 1.566 and 1.260 respectively, they are obtained from calculation of benzene, pyridine and borazine respectively under HF/6-31G* (geometry is optimized at the same level. Becke's atomic space with modified CSD radii and with sharpness parameter $k=3$ is used to derive δ_{ref}). Users can modify or add δ_{ref} through option -4.

The original paper of FLU also defined FLU- π , which is based on DI- π and π -atomic valence

$$\text{FLU}_\pi = \frac{1}{n} \sum_{A-B}^{\text{RING}} \left[\left(\frac{V_\pi(B)}{V_\pi(A)} \right)^\alpha \left(\frac{\delta_\pi(A, B) - \delta_{\text{Avg}}}{\delta_{\text{Avg}}} \right) \right]^2$$

where δ_π is the average value of the DI- π for the bonded atom pairs in the ring, and the other symbols denote the aforementioned quantities calculated using π -orbitals only. The advantage of FLU- π over FLU is that FLU- π does not rely on predefined reference DI value, while the disadvantage is that FLU- π can only be exactly calculated for planar molecules.

Akin to FLU, the lower the FLU- π , the stronger aromatic the ring. If FLU- π is equal to zero, that means DI- π is completely equalized in the ring. The reasonableness to measure aromaticity by FLU- π is that aromaticity for most aromatic molecules are almost purely contributed by π electrons, rather than σ electrons.

In fuzzy atomic space analysis module of Multiwfn, PDI, FLU and FLU- π are calculated in fuzzy atomic spaces. In *J. Phys. Chem. A*, **110**, 5108, the authors showed that the correlation between the PDI, FLU and FLU- π calculated in fuzzy atomic space and the ones calculated in AIM atomic space is excellent.

In Multiwfn, before calculating FLU and FLU- π , AOM will be calculated automatically. If you are calculating FLU- π , you will be prompted to input the indices of π orbitals, you can find out their indices by checking isosurface of all orbitals by main function 0. Then DI or DI- π matrix will be generated. Next, you should input the indices of the atoms in the ring, the input order must be consistent with atom connectivity. Besides FLU or FLU- π value, the contributions from each bonded atom pair are outputted too.

FLU and FLU- π are only available for closed-shell system in Multiwfn. It is not well known whether FLU and FLU- π are also applicable for open-shell systems.

3.18.8 Calculate condensed linear response kernel (9)

Linear response kernel (LRK) is an important concept defined in DFT framework, which can be written as

$$\chi(\mathbf{r}_1, \mathbf{r}_2) = \left(\frac{\delta^2 E}{\delta V(\mathbf{r}_1) \delta V(\mathbf{r}_2)} \right)_N = \left(\frac{\delta \rho(\mathbf{r}_1)}{\delta V(\mathbf{r}_2)} \right)_N$$

This quantity reflects the impact of the perturbation of external potential at \mathbf{r}_2 on the electron density at \mathbf{r}_1 , which may also be regarded as the magnitude coupling between electron at \mathbf{r}_1 and \mathbf{r}_2 .

In Multiwfn, LRK is evaluated by an approximation form based on second-order perturbation theory (see Eq.3 of *Phys. Chem. Chem. Phys.*, **14**, 3960)

$$\chi(\mathbf{r}_1, \mathbf{r}_2) \approx 4 \sum_{i \in \text{occ}} \sum_{j \in \text{vir}} \frac{\varphi_i^*(\mathbf{r}_1) \varphi_j(\mathbf{r}_1) \varphi_j^*(\mathbf{r}_2) \varphi_i(\mathbf{r}_2)}{\varepsilon_i - \varepsilon_j}$$

where φ is molecular orbital, ε stands for MO energy. Note that this approximation form is only applicable to HF/DFT closed-shell systems, therefore present function only works for HF/DFT closed-shell systems.

Condensed linear response kernel (CLRK) is calculated as

$$\chi_{A,B} = \int_A \int_B \chi(\mathbf{r}_1, \mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 = 4 \sum_{i \in \text{occ}} \sum_{j \in \text{vir}} \frac{S_{ij}(A) S_{ji}(B)}{\varepsilon_i - \varepsilon_j}$$

where A and $S(A)$ denote fuzzy atomic space and atomic overlap matrix for atom A, similar for atom B. In *Phys. Chem. Chem. Phys.*, **15**, 2882 (2013), it was shown that CLRK is useful for investigation of aromaticity and anti-aromaticity.

Present function is used to calculate CLRK between all atom pairs in current system, and the result will be outputted as a matrix. Due to evaluation of LRK requires virtual MO information, in current version .fch/.molden/.gms file must be used as input file.

Note that CLRK can be decomposed to orbital contribution, e.g. for MO i

$$\chi_{A,B}^{(i)} = 4 \sum_{j \in \text{vir}} \frac{S_{ij}(A) S_{ji}(B)}{\varepsilon_i - \varepsilon_j}$$

For instances, assume that you want to evaluate the contribution from MO 3,4,7, then before calculating CLRK, you should enter main function 6 and use option 26 to set occupation number of all MOs except for 3,4,7 to zero. (Note that the virtual MOs used to calculate LRK will automatically still be the original virtual MOs, rather than the ones after modification of MO occupation numbers.)

3.18.9 Calculate para linear response index (10)

The definition of para linear response index (PLR) has an analogy to PDI, the only difference is that DI is replaced by CLRK

$$PLR(A, B) = \frac{\chi_{1,4} + \chi_{2,5} + \chi_{3,6}}{3}$$

In *Phys. Chem. Chem. Phys.*, **14**, 3960 (2012), the authors argued that PLR is as useful as PDI in quantitatively measuring aromaticity, and it is found that the linear relationship between PLR and PDI is as high as $R^2=0.96$.

Present function is used to calculate PLR. Multiwfn will first calculate CLRK, and then you should input the indices of the atoms constituting the ring in question, e.g. 3,5,6,7,9,2. The input order must be consistent with atom connectivity. Then PLR will be immediately outputted on screen. PLR is only applicable to HF/DFT closed-shell systems, and currently .fch/.molden/.gms must be used as input file.

Note that for completely planar systems, PLR can be exactly separated as PLR- α and PLR- π to individually study α aromaticity and π aromaticity. In order to calculate PLR- α (PLR- π), before enter present module, you should first manually set occupation number of all MOs except for π (α) MOs to zero (or you can utilize option 22 in main function 100 to do this step, which will be much more convenient).

3.18.10 Calculate multi-center delocalization index (11)

n -center multi-center DI is calculated as

$$\delta(A, B, C \dots H) = 2^{n-1} \sum_i \sum_j \sum_k \dots \sum_q S_{ij}(A) S_{jk}(B) S_{kl}(C) \dots S_{qi}(H)$$

where $i, j, k \dots$ only cycle occupied orbitals. The normalized form of multi-center DI is defined as $\delta^{1/n}$, and may be compared between rings with different number of members.

Currently this function is only available for single-determinant closed-shell wavefunctions, and supports up to 10 centers. Note that for relatively large size of systems, calculating multi-center DI for more than 6 centers may be quite time-consuming.

3.18.11 Calculate information-theoretic aromaticity index (12)

In *ACS Omega*, **3**, 18370 (2018) it is shown that arithmetic mean of some information-theoretic quantities of the atoms constituting a ring has good linear relationship with other widely accepted aromaticity indices, such as HOMA and aromatic stabilization energy (ASE). It is thus clear that the arithmetic mean may be used as index for measuring aromaticity, although this point needs to be

further explored.

The information-theoretic aromaticity index, namely the above-mentioned arithmetic mean can be calculated via subfunction 12 of fuzzy analysis module. After entering this function, you should choose the way of defining atomic information-theoretic quantity, three choices are currently available:

$$\text{Atomic Shannon entropy: } s_S(A) = \int -\rho(\mathbf{r}) \ln \rho(\mathbf{r}) w_A(\mathbf{r}) d\mathbf{r}$$

$$\text{Atomic Fisher information: } i_F(A) = \int |\nabla \rho(\mathbf{r})|^2 / \rho(\mathbf{r}) w_A(\mathbf{r}) d\mathbf{r}$$

$$\text{Atomic GBP entropy: } s_{\text{GBP}}(A) = \int (3/2)\rho(\mathbf{r})\{\lambda + \ln[t(\mathbf{r})/t_{\text{TF}}(\mathbf{r})]\}w_A(\mathbf{r})d\mathbf{r}$$

Essentially, the three quantities correspond to the integral of user-defined functions 50, 51 and 54 in fuzzy atomic space. In this function, you also need to input the index of the atoms in the ring. Once calculation of the selected quantity for all atoms in the ring is finished, the average will be shown, and it can be regarded as an aromaticity index.

Before using this function, you can firstly select the way of defining atomic space. In the original paper, Hirshfeld partition was employed, while the default partition method of the fuzzy analysis module is Becke.

Information needed by fuzzy analysis module: GTFs, atom coordinates, basis functions (only for calculating condensed linear response kernel and PLR)

3.19 Charge decomposition analysis and plotting orbital interaction diagram (16)

3.19.1 Theory

The charge decomposition analysis (CDA) proposed by Dapprich and Frenking (*J. Phys. Chem.*, **99**, 9352 (1995)) is used to provide deep insight on how charges are transferred between fragments in a complex to achieve charge equilibrium. The idea of CDA is based on fragment orbital (FO), which denotes the molecular orbital (MO) of fragment in its isolated state. Besides, once the compositions of FOs in MOs of complex are obtained, the orbital interaction diagram can be directly plotted, which allows one visually and directly understand how orbitals of fragments are mixed to form orbitals of complex.

For simplicity, in this section we assume that the complex consists of only two fragments. However, the CDA can be straightforwardly used for more than two fragments cases.

Fragment orbital

Consider we are studying a complex AB, NA basis functions are located in the atoms of fragment A, NB basis functions in fragment B, then each MO of complex will be linearly expanded by $NA+NB$ basis functions, and meanwhile, the complex has $NA+NB$ MOs. By using the same basis set, and maintaining the same geometry as in complex, if we calculate the two fragments respectively,

we can obtain NA MOs of fragment A, and NB MOs of fragment B, they are collectively called as fragment orbital (FO). We can take these FOs as basis functions to linearly expand the MOs of complex. Since the dimension (the number of basis functions) is still $NA+NB$, the expansion is exact. In other words, we equivalently transformed the basis.

Charge decomposition analysis

In the original paper of CDA, the authors defined three terms:

$$\begin{aligned} d_i &= \sum_{m \in A}^{\text{occ}} \sum_{n \in B}^{\text{vir}} \eta_i C_{m,i} C_{n,i} S_{m,n} \\ b_i &= \sum_{m \in A}^{\text{vir}} \sum_{n \in B}^{\text{occ}} \eta_i C_{m,i} C_{n,i} S_{m,n} \\ r_i &= \sum_{m \in A}^{\text{occ}} \sum_{n \in B}^{\text{occ}} \eta_i C_{m,i} C_{n,i} S_{m,n} \end{aligned}$$

where i and η are index and occupation number of MO of complex, respectively. $S_{m,n} = \int \varphi_m(\mathbf{r}) \varphi_n(\mathbf{r}) d\mathbf{r}$ is overlap integral between FO m and FO n . Note that though the NA and NB FOs are respectively orthonormalization sets, the NA set are in common not normal to the NB set, so S is not an identity matrix. $C_{m,i}$ denotes the coefficient of FO m in MO i of complex. The superscript "vir" and "occ" mean virtual (viz. unoccupied) and occupied, respectively.

The term d_i denotes the amount of electron donated from fragment A to B via MO i of complex; similarly, the term b_i denotes the electron back donated from B to A. In fact, $\eta_i C_{m,i} C_{n,i} S_{m,n}$ can be regarded as the half of overlap population between FO m and n in MO i . Hence, the difference between term d and b is that which fragment provides its electrons from its occupied FOs to virtual FOs of another fragment. The term r reveals closed-shell interaction between two occupied FOs in different fragments; positive value of r_i means that owing to MO i , the electrons of the two fragments are accumulated in their overlap region and shows bonding character, while negative value indicates that the electrons are depleted from the overlap region and thus reflecting electron repulsive effect. The sum of all r_i terms is in general negative, because overall interaction between filled orbitals are generally repulsive. r is also known as "repulsive polarization" term

Beware that although the CDA formulae given in original paper are correct, by carefully inspecting the data, I found the d , b and r terms in the examples presented in the original paper are erroneous (the data should be divided by two).

Generalizing CDA

The original definition of CDA has two drawbacks. First, it is only applicable to closed-shell cases (namely, complex and each fragment must be closed-shell) and hence unable to be used when the two fragments are bound by covalent bonding. Second, in post-HF calculations, though the MOs of complex can be replaced by natural orbitals (NOs), the FOs can only be produced by HF or DFT calculation, because occupation numbers of FOs are not explicitly considered in the original CDA formulae.

To address the limitations of the original definition, in *J. Adv. Phys. Chem.*, **4**, 111-124 (2015) (<http://dx.doi.org/10.12677/JAPC.2015.44013>) I proposed a generalized form of CDA, which is the

form used in CDA module of Multiwfn:

$$t_i = \sum_{m \in A} \sum_{n \in B} \eta_i \frac{|\eta_m^{\text{FO}} - \eta_n^{\text{FO}}|}{\eta_{\text{ref}}} C_{m,i} C_{n,i} S_{m,n}$$

$$r_i = \sum_{m \in A} \sum_{n \in B} 2 \frac{\min(\eta_m^{\text{FO}}, \eta_n^{\text{FO}})}{\eta_{\text{ref}}} \eta_i C_{m,i} C_{n,i} S_{m,n}$$

In the generalized CDA, orbitals of complex and fragments can be produced either by HF/DFT or by post-HF method, corresponding to MOs and NOs, respectively. η_m^{FO} stands for occupation number of FO m . For open-shell cases, η_{ref} is 1.0, CDA will be performed for alpha spin and beta spin separately; for the former, i denotes alpha orbital of complex, m and n run over all alpha FOs; for the latter, i denotes beta orbital of complex, m and n run over all beta FOs. For closed-shell cases, η_{ref} is 2.0, m and n run over space orbitals. $\min()$ is the function used to extract minimum from two values. During calculation of t , if the values are only accumulated for the cases $\eta_m^{\text{FO}} > \eta_n^{\text{FO}}$, then the resulting t is d ; if only for the cases $\eta_m^{\text{FO}} < \eta_n^{\text{FO}}$, then t will be b .

For the situations when the original CDA is applicable, the b and d calculated by the generalized form are exactly identical to the ones obtained via original definition; while r will be exactly twice of the one produced via original definition. The reason why the factor 2 is introduced into the generalized form of r is because after doing so, r has more clear physical meaning, namely it equals to overlap population (also known as Mulliken bond order) between the occupied FOs in the two fragments.

Because the CDA has been generalized, below, FO will stand for MO or NO of fragment, "complex orbital" will denote MO or NO of complex. The orbitals can either be spin-space orbital (for open-shell) or space orbital (for closed-shell).

It is clear that the d , b and r terms can be decomposed into FO pair contributions, this kind of decomposition is supported by Multiwfn and greatly facilitates analysis of the nature of FO interaction.

Composition of FO in complex orbital and orbital interaction diagram

By using the methods discussed in Section 3.10, the composition of FO in complex orbital can be calculated. In common, Mulliken method is the best choice for this purpose, the composition of FO m in complex orbital i is calculated as

$$\Theta_{m,i} = \left(C_{m,i}^2 + \sum_{n \neq m} C_{m,i} C_{n,i} S_{m,n} \right) \times 100\%$$

Although using SCPA method to calculate composition is more convenient (since overlap matrix is not involved), if i is a high-lying virtual orbital, the calculated composition is not very reasonable. Note that due to a well-known drawback of Mulliken method, sometimes negative contributions occur, since the values are often not large, you can simply ignore them.

From Θ , we can clearly understand how each complex orbital is formed by mixing FOs of the two fragments. Furthermore, one can plot orbital interaction diagram to visually and intuitively study the relationship between complex orbitals and FOs, namely plot a bar for each complex orbital

and FO according to its energy, and then check each Θ to determine how to link the bars, e.g. if the value of $\Theta_{m,i}$ is larger than 5%, then the two bars corresponding to FO m and complex orbital i will be linked. Consequently, from the graph we will directly know that FO m has important contribution to complex orbital i .

Extended charge decomposition analysis

The difference between the total number of donation and back donation electrons, that is $d - b$, may be regarded as the net transferred electrons. However, in *J. Am. Chem. Soc.*, **128**, 278, the authors argued that this viewpoint is not correct, because b and d terms not only represent charge transfer effect (CT), but also electron polarization effect (PL); the latter describes the adjustment of electron distribution within the fragment, which is caused by mixing virtual and occupied FOs of the same fragment during formation of the complex, and should be excluded in the calculation of the amount of net transferred electrons. In this paper they proposed extended charge decomposition analysis (ECDA) method, by which they argued that the number of net transferred electrons can be calculated more reasonably.

In ECDA viewpoint, four terms can be defined as follows

1. $PL(A) + CT(A \rightarrow B)$ = The sum of compositions of occupied FOs of fragment A in all virtual orbitals of complex, multiplied by Occ
2. $PL(A) + CT(B \rightarrow A)$ = The sum of compositions of virtual FOs of fragment A in all occupied orbitals of complex, multiplied by Occ
3. $PL(B) + CT(B \rightarrow A)$ = The sum of compositions of occupied FOs of fragment B in all virtual orbitals of complex, multiplied by Occ
4. $PL(B) + CT(A \rightarrow B)$ = The sum of compositions of virtual FOs of fragment B in all occupied orbitals of complex, multiplied by Occ

where Occ is 1.0 and 2.0 for open-shell and closed-shell cases, respectively.

After the four terms are calculated, the number of net transferred electrons from A to B can be directly obtained as

$$CT(A \rightarrow B) - CT(B \rightarrow A) = [PL(A) + CT(A \rightarrow B)] - [PL(A) + CT(B \rightarrow A)]$$

The composition of complex orbitals can be calculated by various methods, leading to different ECDA result. The method used in Multiwfn is identical to the one in ECDA original paper, namely Mulliken method.

Note that ECDA can neither be applied to post-HF calculation nor the system consisting more than two fragments.

According to the name, ECDA is as if an extension of CDA, however in my personal opinion, ECDA is irrelevant to CDA, their basic ideas are quite different, and thus the amount of net transferred electron calculated by ECDA is not comparable with the $d - b$ produced by CDA at all. In addition, though ECDA is realized in Multiwfn, I do not think this is a useful method. The most remarkable feature of CDA is that the electron transfer can be decomposed to contribution of complex orbitals, however ECDA is incapable to do this; ECDA can only reveal how many electrons is transferred between two fragments, but this quantity actually can be obtained by a more straightforward approach, namely calculating the fragment charge by summing up all atomic charges in the fragment, and then subtracting it by the net charge of the fragment in its isolated state.

3.19.2 Input file

There are several points you should note:

In Multiwfns, infinite number of fragments can be defined; however, if more than two fragments are defined, ECDA analysis will be unavailable.

If the complex or any one fragment is an open-shell system, then CDA will be performed separately for alpha and beta electrons.

If in the calculation of the complex or any one fragment post-HF method is employed to produce natural orbitals, then ECDA analysis cannot be performed, and orbital interaction diagram cannot be drawn (since orbital energy of natural orbitals is not well-defined).

For Gaussian users

Below files are required as input files of CDA module:

(1) Gaussian output file of single point task of complex. The keywords *nosymm pop=full IOp(3/33=1)* must be specified in route section; in which *3/33=1* makes Gaussian output overlap matrix between basis functions.

(2) Gaussian output file of single point task of each fragment. The keywords *nosymm pop=full* must be specified in route section.

Notice that the atomic coordinates in fragment calculations must be exactly in accordance with that in complex calculation, therefore do not optimize geometry for each fragment, you only need to directly extract fragment coordinates from the optimized complex coordinate. The atomic sequence in fragments and complex must be identical, that means the atomic sequence in the complex can be retrieved by successively combining the atoms in fragment 1, 2, 3 ... *n*. Molecular geometry must be given in Cartesian coordinate. The basis set used in the complex and fragment calculations must be completely the same. Notice that if you intent to use mixed basis set, and in particular when Pople type basis set is involved (such as 6-31G*), you'd better specify "**5d 7f**" keywords in the fragment calculations to force Gaussian to use spherical-harmonic type basis functions.

By default Gaussian automatically eliminates linearly dependant basis functions, hence in some cases the number of basis functions will not be equal to the number of orbitals and thus CDA cannot be performed, in this case *IOp(3/32=2)* should be specified, which avoids Gaussian eliminating linearly dependant basis functions. The common reason of this problem comes from the use of diffuse basis functions. According to my experiences, when diffusion basis functions are presented, the CDA results are often unreasonable or even completely meaningless. So, **DO NOT use diffuse functions in CDA analysis!**

The theoretical method used in the calculations can be HF/DFT or post-HF. Closed-shell and unrestricted open-shell wavefunctions are supported (In fact ROHF/ROKS is also supported and closed-shell form of CDA will be employed, however the result is often not very useful). For post-HF calculation, in order to output coefficient and occupation number for natural orbitals, do not use "pop=full" but use "density pop=NO" and "density pop=NOAB" for closed-shell and open-shell cases, respectively.

It is also completely OK to use the corresponding .fch as input file of CDA module instead of Gaussian output file for HF/DFT, and in this case *IOp(3/33=1)* and *pop=full* are not needed. However for post-HF case, you need to follow the steps given at the beginning of Chapter 4 to save natural orbitals into .fch file, this is somewhat cumbersome and not as straightforward as using

Gaussian output file.

For other quantum chemistry program users

You can also use CDA module as long as your favourite *ab-initio* program is able to produce Molden input file (.molden), such as ORCA and Molpro. Note that not the .molden outputted by all of quantum chemistry codes can be properly recognized by Multiwfn, see part 4 of Section 2.3 for explanation. In addition, .gms file (GAMESS-US or Firefly output file) may also be used as input file.

Similar to the case of Gaussian, you need to use your *ab-initio* program generate .molden or .gms file for complex and all fragments. All of the points mentioned above must be fulfilled, namely the avoidance of elimination of linearly dependant basis functions, the consistency of the atomic coordinates and the correspondence of atomic sequence in complex and fragments .molden files. An example is given in "examples\CDA\COBH3_ORCA" folder, the .molden files were generated by ORCA 3.0.1.

3.19.3 Usage

After booting up Multiwfn, first you should input the path of the file (i.e. Gaussian output file or .fch or .molden or .gms) for complex, and then enter the CDA module. Next, you should set the number of fragments, and then input the path of the file for each fragment in turn, the orbital information recorded in which will be loaded into memory.

For the open-shell fragments, you will be prompted to select if flip its spin. If select "y", then orbital information of its alpha and beta orbitals will be exchanged. The reason for introduction of this step is obvious, for example, we want to use CDA to decompose electron transfer between fragment CH₃ and NH₂ in CH₃NH₂; CH₃NH₂ has 9 alpha and 9 beta electrons, while in the calculations of quantum chemistry codes, both CH₃ and NH₂ will be regarded as having 5 alpha and 4 beta electrons. In order to carry out CDA, we have to flip electron spin of either CH₃ or NH₂, otherwise the total numbers of alpha and beta electrons in the two fragments, namely 5+5 and 4+4 respectively, will be unequal to the ones of complex, namely 9 and 9, respectively.

After the loading is finished, Multiwfn will calculate some data. If only two fragments are defined, CDA and ECDA result will be directly shown. Then you will see a menu:

-2 Switch output destination (for options 0 and 1): By default the options 0 and 1 output results on screen; if you select this option once, then their results will be outputted to CDA.txt in current folder.

0 Print CDA result and ECDA result: Input the index of two fragments, then the CDA and ECDA analysis result between them will be outputted.

1 Print full CDA result: If you select this option, the CDA result for all complex orbitals will be outputted. By default, Multiwfn does not output CDA result for the complex orbitals lying higher than LUMO+5, because the number of such orbitals is too large, whereas their contributions to *d*, *b* and *r* terms are often completely negligible due to their occupation numbers are often quite small. (For HF/DFT wavefunctions, the occupation numbers of the orbitals lying higher than HOMO are exactly zero, and thus have no contribution to *d*, *b* and *r* terms at all. So for this case this option is meaningless)

2 Show fragment orbital contributions to specific complex orbital: If you input *x*, then the composition of complex orbital *x* will be outputted (for open-shell cases, the *x*th alpha and the *x*th beta complex orbital will be outputted respectively). By default only the FOs having contribution

$\geq 1\%$ will be shown, but this threshold can be altered by "comphresCDA" parameter in settings.ini.

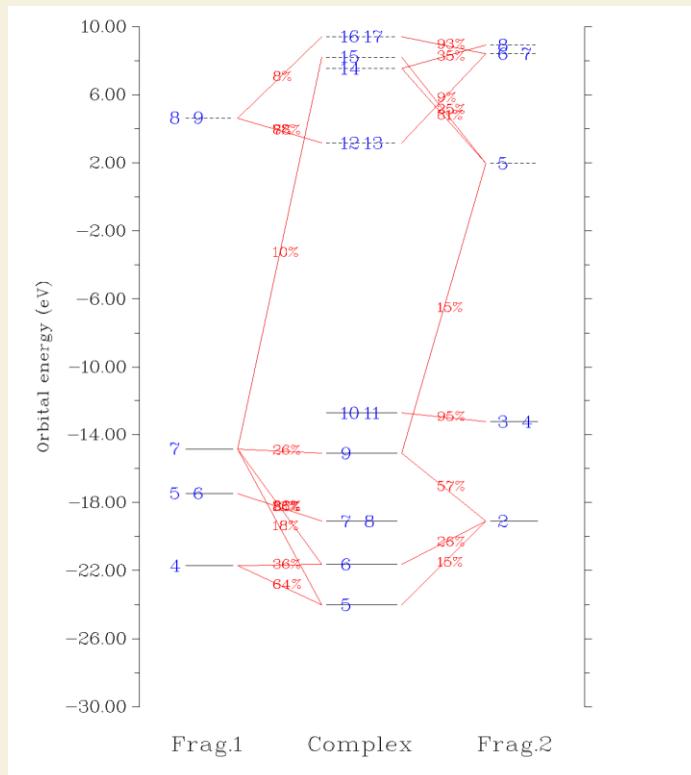
If you want to obtain composition of a fragment orbital in all complex orbitals, you can input for example 1,6, which means orbital 6 of fragment 1 is selected.

3 Export coefficient matrix of complex orbitals in fragment orbital basis: The coefficient matrix corresponding to all FOs in all complex orbitals will be outputted to "coFO.txt" in current folder.

4 Export overlap matrix between fragment orbitals: The overlap matrix between all FOs will be outputted to "ovlpint.txt" in current folder.

6 Decompose complex orbital contribution to CDA: You need to input index of a complex orbital and set threshold for printing, if contribution of a pair of fragmental orbitals to any of d, b and r term of this complex orbital is larger than the threshold then the contribution value will be shown. This greatly facilitates analysis of interaction between fragment orbitals.

5 Plot orbital interaction diagram: This option is only available for HF/DFT wavefunctions. If you select this option, you will enter a new menu, in which by corresponding options you can plot and save orbital interaction diagram and adjust plotting parameters, such as size of labels, energy range (namely Y-axis range) and the criterion for linking bars. The orbital interaction diagram plotted under default settings looks like this:



In above graph, occupied and virtual orbitals are represented as solid and dashed bars, respectively, the vertical positions are determined by their energies. The bars at left and right sides correspond to the FOs of the two fragments you selected; the bars in the middle correspond to complex orbitals. Orbital indices are labelled by blue texts. If two or more labels occur in the same bar, that means these orbitals are degenerate in energy. If composition of a FO in a complex orbital is larger than specific criterion, then the corresponding two bars will be connected by red line, so that simply by viewing the diagram one can directly understand the complex orbitals are constructed by mainly mixing which FOs. The compositions are labelled in the center of the lines.

By default, all FOs and complex MOs are plotted, and if the contribution of a FO of fragment A or B to a complex MO is larger than 10% then they will be connected. For large systems, usually there are too many bars and linking lines in the diagram, and it is hence difficult to identify the orbital interaction mode based on the diagram. In these cases, you should properly use the option "4 Set the rule for connecting and drawing orbital bars" to manually set up the rule for connecting and plotting the orbital bars. See the prompt shown on the screen on how to use this option.

Sometimes the difference between orbital energies of the two fragments in the orbital interaction diagram is too large and thus hinders one to analyze the diagram, so you may want to equalize their energies. In this case, you can use the option "12 Set orbital energy shifting value" to set the value used to shifting orbital energies of the complex or the two fragments presented at the left and right side of the diagram; their orbital energies will be augmented by the given value.

Some examples of CDA are given in Section 4.16.

Information needed: Gaussian output files of complex and fragments

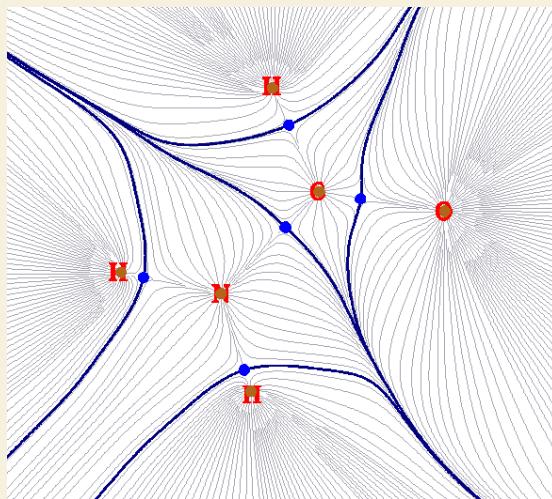
3.20 Basin analysis (17)

3.20.1 Theory

The concept of basin was first introduced by Bader in his atom in molecular (AIM) theory, after that, this concept was transplant to the analysis of ELF by Savin and Silvi. In fact, basin can be defined for any real space function, such as molecular orbital, electron density difference, electrostatic potential and even Fukui function.

A real space function in general has one or more maxima, which are referred to as attractors or (3,-3) critical points. Each basin is a subspace of the whole space, and uniquely contains an attractor. The basins are separated with each other by interbasin surfaces, which are essentially the zero-flux surface of the real space functions; mathmatically, such surfaces consist of all of the points \mathbf{r} satisfying $\nabla f(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) = 0$, where $\mathbf{n}(\mathbf{r})$ stands for the unit normal vector of the surface at position \mathbf{r} .

Below figure illustrates the basins of total electron density of NH_2COH , such figure is often involved in AIM analysis. Brown points are attractors; in this case they correspond to maxima of total electron density and thus are very close to nuclei. Blue bolded lines correspond to interbasin surfaces, they dissect the whole molecular space into basins, which in present case are also known as AIM atomic space or AIM basin. Grey lines are gradient paths of total electron density, they are emanated from attractors.



Multiwfn is able to generate basins for any supported real space functions, such as electron density, ELF, LOL, electrostatic potential. Alternatively, one can let Multiwfn load the grid file generated by third-part programs (e.g. cube file or .grd file) and then generate basins for the real space function recorded in the grid data.

After the basins were generated, some analyses based on the basins can be conducted to extract information of chemical interest. The three most useful types of basin analyses are supported by Multiwfn and are briefed below. All of them require performing integration in the basin.

(1) Study integral of a real space function in the basins. For example, one can first use electron density to define the basins, and then integrate electron density in the basins to acquire electron population numbers in the basins; this quantity is known as AIM atomic population number. Another example, one can first partition the space into basins according to ELF function, and then calculate integral of spin density in the basins.

(2) Study electric multipole moments in the basins. In Multiwfn electric monopole, dipole and quadrupole moments can be calculated in the basins. This is very useful to gain a deep and quantitative insight on how the electrons are distributed in different regions of a system. Based on these data, one can clearly understand such as how lone pair electrons contribute to molecular dipole moment (see *J. Comput. Chem.*, **29**, 1440 for example, in which ELF basins are used), how the electron density around an atom is polarized as another molecule approaching.

(3) Study localization index and delocalization index in the basins. The former one measures how many electrons are localized in a basin in average, while the latter one is a quantitative measure of the number of electrons delocalized (or say shared) between two basins. They have been very detailedly introduced in Section 3.18.5 and hence will not be reintroduced here. The only difference relative to Section 3.18.5 is that the range of integration in our current consideration is basins rather than fuzzy atom spaces.

3.20.2 Numerical aspects

This section I will talk about many numerical aspects involved in basin analysis, so that you can understand how the basin analysis module of Multiwfn works.

Algorithms for generating basins

The algorithms used to generate/integrate basins can be classified into three categories:

(1) Analytical methods. Due to the popularity and importance of AIM theory, and the complex shape of AIM basins, since long time ago numerous methods have been proposed to attempt to speed up integration efficiency and increase integration accuracy for AIM basins, e.g. *J. Comput. Chem.*, **21**, 1040 and *J. Phys. Chem. A*, **115**, 13169. All of these methods are analytical, and they are employed by almost all old or classical AIM programs, such as AIMPAC, AIM2000, Morphy and AIMAll. This class of methods suffers many serious disadvantages: 1. The algorithm is very complicated 2. High computational cost 3. Only applicable to the analysis of total electron density 4. Before generating basins, (3,-3) and (3,-1) critical points must be first located in some ways. The only advantage of these methods is that the integration accuracy is very satisfactory.

(2) Grid-based methods. This class of methods relies on cubic (or rectangle) grid data. Since the publication of TopMoD program, which aims to analyze ELF basin and for the first time uses a grid-based methods, grid-based methods continue to be proposed and incurred more and more attention. Grid-based methods solved all of the drawbacks in analytical methods; they are easy to be coded, the computational cost is relative low, and more important, they are suitable for any type of real space function. Unfortunately these methods are not free of shortcomings, the central one is that the integration accuracy is highly dependent on the quality of grid data; to obtain a high accuracy of integral value the grid spacing must be small enough.

(3) Mixed analytical and grid-based method. The only instance in this class of method is described in *J. Comput. Chem.*, **30**, 1082, in which Becke's multicenter integration scheme is used to integrate basins. This method is faster than analytical method, and more accurate than grid-based method, but it is only suitable to analyze total electron density, thus the application range is seriously limited. Since the basin analysis module of Multiwfn focuses on universality, this method is not currently implemented.

Currently the best grid-based method is near-grid method (*J. Phys.: Condens. Matter*, **21**, 084204), which is employed in Multiwfn as default method to generate basins. Multiwfn also supports on-grid method (*Comput. Mat. Sci.*, **36**, 354), which is the predecessor of near-grid method.

Basic steps of generating basins and locating attractors in Multiwfn

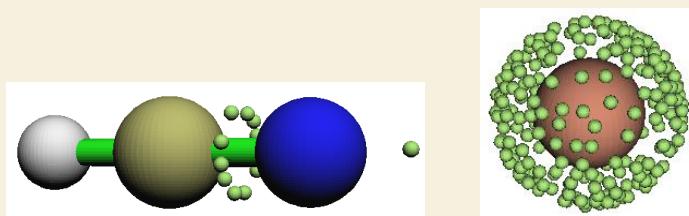
In Multiwfn, generating basin and locating attractor are carried out simultaneously; the basic process can be summarized as follows: Given a grid data, all grids (except for the grids at box boundary) are cycled in turn. From each grid, a trajectory is evolved continuously in the direction of maximal gradient. Each step of movement is restricted to its neighbour grid, and the gradient in each grid is evaluated via one direction finite difference by using the function values of it and its neighbour grids. There are four circumstances to terminate the evolvement of present trajectory and then cycle the next grid (a) The trajectory reached a grid, in which the gradients in all directions is equal or less than zero; such a grid will be regarded as an attractor, meanwhile all grids contained in the trajectory will be assigned to this attractor. (b) The trajectory reached a grid that has already been assigned; all of the grids contained in the trajectory will be assigned to the same attractor. This treatment greatly reduced computational expense. (c) The upper limit of step number is reached (d) The trajectory reached the grids at box boundary; all of the grids in the trajectory will be marked as "travelled to boundary" status.

After all grids have been cycled, we obtained position of all attractors covered by the spatial scope of the grid data. Each set of grids that assigned to a same attractor collectively constitutes the basin corresponding to the attractor. Interbasin grids will then be detected, according to the criterion that if any neighbour grid belongs to different attractor. Note that sometimes after finished above

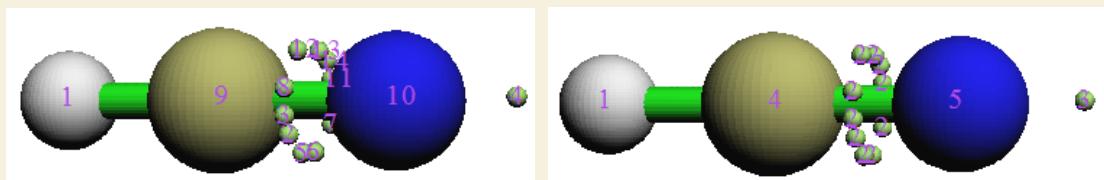
processes, some grids may remain unassigned, and some grids may have the status "travelled to boundary". Such grids are often far away from atoms and thus unimportant, in general you can safely ignore them.

The only difference between on-grid method and near-grid method is that in the latter one, the so-called "correction step" is introduced to continuously calibrate the evolvement direction of the trajectory, which greatly eliminates the artificiality problem of interbasin surfaces due to on-grid method, and in turn makes the basin integration more close to actual values. Near-grid method can be supplemented by a refinement step for interbasin surfaces at the final stage, so that the partition of basin can be even more accurate, this additional step is not very time-consuming.

For ELF (and may be other real space functions), if sphere- or ring-shape attractors exist in the system, in corresponding regions a large number of attractors having basically the same value will be found. Below two examples illustrated the existence of the ring-shape ELF attractor encircling N-C bond of HCN and the sphere-shape ELF attractor encompassing the Ar atom.



Based on nearest neighbour algorithm, Multiwfn automatically checks the distances and the relative value differences between all pairs of located attractors, and then clusters some attractors together. The resulting attractors thus have multiple member attractors, and will be referred to as "degenerate attractors" later. For example, the left graph and right graph given below displayed the attractor indices before and after clustering, respectively. As you can see, after clustering, all of the attractors constituting the ring-shape ELF attractor now have the same index, suggesting that the corresponding basins have been merged together as basin #2.



For certain cases, in the regions far beyond the systems, some artifical or physically meaningless attractors are located. These attractors have very low function value and thus are called as "insignificant" attractors; in contrast, other attractors can be called as "significant" attractors. The presence of insignificant attractors is because in corresponding regions the function values are rather small, hence the function behavior is not very definitive, even a very slight fluctuation of function value or trivial numerical perturbation is enough to result in new attractors. When insignificant attractors are detected, Multiwfn will prompt you to select how to deal with them, there are three options: (1) Do nothing, namely do not remove these attractors (2) Remove these attractors meanwhile set corresponding grids as unassigned grids (3) Remove these attractors meanwhile assign corresponding grids to the nearest significant attractors. The third option is generally recommended.

If the real space function simultaneously possesses positive and negative parts, e.g. electrostatic

potential, Multiwfn will automatically invert the sign of negative part of the grid data and then locate attractors and generate basins as usual, after that the sign will be retrieved. Therefore, for negative parts, the "attractors" located by Multiwfn actually are "repulsors".

Integration of basins

Once the basins have been generated by grid-based method, the integral of a real space function f in a basin P can be readily evaluated as $F = \sum_{i \in P} f(\mathbf{r}_i) \times dV$, where i is grid index, dV is volume differential element. Evidently, the finer the grid data (smaller grid spacing) is used, the more accurate the integral values will be, and correspondingly, the more computational effort must be paid.

Since the arrangement of grids in general is not in coincident with system symmetry, one should not expect that the integral values always satisfy system symmetry well, unless very fine grid is used.

Note that different functions have different requirement on the quality of grid data. For the same grid spacing, the faster the function varies, the lower the integration accuracy will be. Due to Laplacian of electron density, source function and kinetic density function etc. fluctuate violently near nuclei, it is impossible to directly integrate these functions in AIM basins solely by grid-based method at satisfactory accuracy (since only uniform grids are used, which is incapable to represent nuclear region well enough). In order to tackle this difficulty, an integration method based on mixed atomic-center and uniform grids is supported by Multiwfn, in which the regions close to nuclei are integrated by atomic-center grids, while the other regions are integrated by uniform grids. This method is able to integrate AIM basin for any real space function, including the ones having complicated behavior at generally acceptable accuracy without evident additional computational cost with respect to grid-based method. In order to further improve the integration accuracy for AIM basins, Multiwfn also enables one to exactly refine the assignment of the grids at basin boundary. An exact steepest ascent trajectory is emitted from each boundary grid, and terminates when reaches an attractor. Since there are often very large number of boundary grids and each steepest ascent step requires evaluating gradient of electron density once, this process is time-consuming. In Multiwfn these gradients can also be approximately evaluated by trilinear interpolation based on pre-calculated grid data of gradient, 1/3~1/2 of overall time cost of the AIM basin integration can be saved. Note that if the mixed type of atomic-center and uniform grids is employed, and especially when the exact refinement process of boundary basin is enabled at the same time, then the requirement on quality grid will be markedly lowered, usually accurate result can be obtained at "Medium-quality grid" level (grid spacing=0.1 Bohr).

It is worth noting that even though basin analysis module of Multiwfn can be used to acquire position of attractors, the accuracy is directly limited by the quality of grid data, since the attractors will be located on a grid point. Moreover, this module is unable to be used to search other types of critical points (attractor corresponds to (3,-3) type critical point); So if you want to obtain accurate coordinate and value for all types of critical points, you should make use of main function 2, namely topology analysis function.

3.20.3 Usage

The basin analysis module in Multiwfn is very powerful, fast and flexible. Basins can be

generated and visualized for any real space function, meanwhile any real space function can be integrated in the generated basins. Electric multipole moments and localization index in the basins and delocalization index between basins can be calculated.

Basic steps

To perform basins analysis in Multiwfn, there are five basic steps you need to do

(1) Load input file into Multiwfn and then enter main function 17.

(2) Select option 1 to generate basins and locate attractors. Before doing this, you can change the method used to generate basins via option -1, or adjust the parameters for clustering attractors via option -6.

(3) Visualize attractors and basins by option 0. This step is optional.

(4) Perform the analyses you want. Integral of a real space function in generated basins, electric multipole moments in the basins and localization/delocalization index of the basins can be calculated by option 2 (or 7), 3 (or 8) and 4, respectively.

You can export the generated basins as cube files by option -5, export attractors as .pdb file by option -4, check the coordinate of the located attractors by option -3, measure the distances, angles and dihedrals angles between attractors or atoms by option -2. Also you can use suboption 3 in option -6 to manually merge specified basins as a single one. If you want to regenerate basins and locate attractors for other real space functions or under new settings, you can select option 1 again.

About input file and grid data

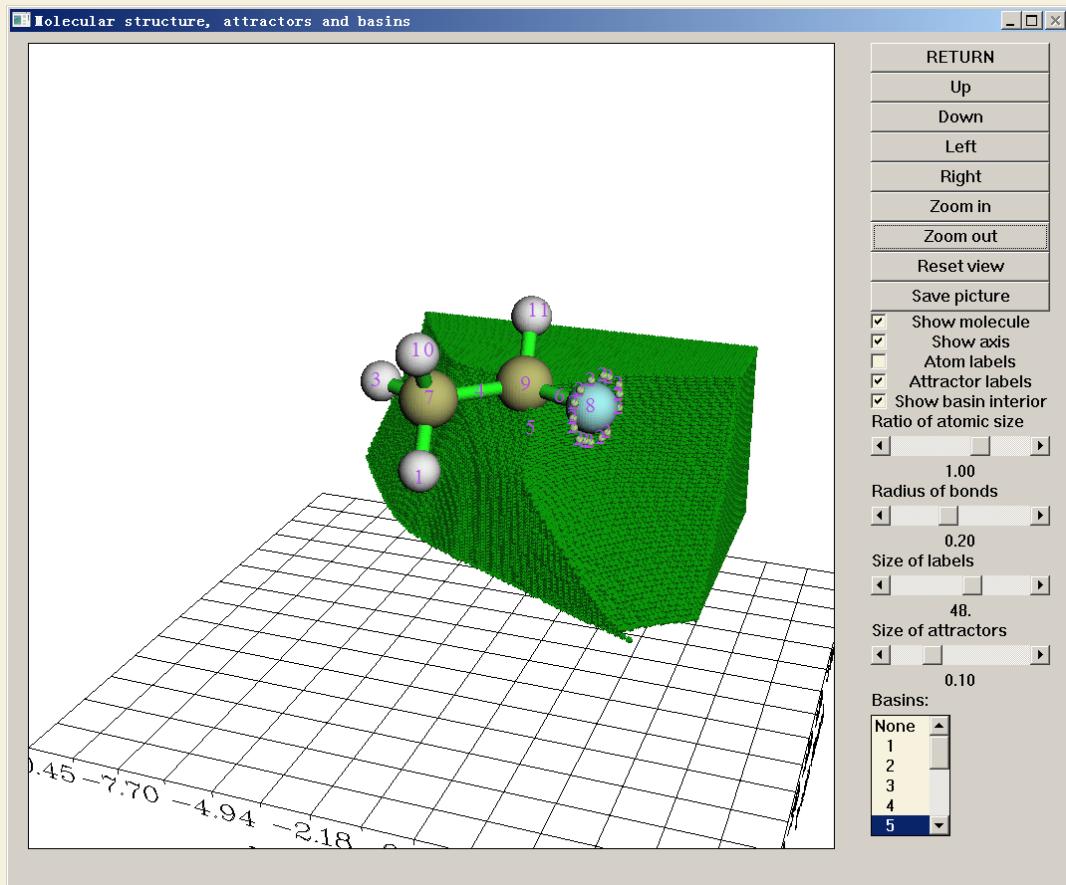
Grid data must be obtained first before generating basins. In option 1 of basin analysis module, you can let Multiwfn calculate the grid data. However, if a grid data has already been stored in memory, you can directly choose to use it. There are two circumstances in which grid data will be stored in memory: (1) The input file contains grid data, e.g. .cub and .grd file (2) You have used main function 5 or 13 to calculate grid data before entering main function 17. This design makes Multiwfn rather flexible: you can use Multiwfn to generate basins for a real space function recorded in a .cub or .grd file, which may be outputted by a third-part program; and you can also first use main function 5 or 17 to generate grid data for electron density difference, Fukui function, dual descriptor and so on, and then generate basins for them.

In the function of basin integration, namely option 2, the value of integrand at each grid must be available so that the integral can be evaluated. These values can come from three ways, you can choose any one: (1) Let Multiwfn directly calculate them (2) Load and use the grid data in a .cub/.grd file, note that this operation will not overwrite the grid data already stored in memory (3) Use the grid data already stored in memory. Beware that the grid setting of the grid data stored in memory or the grid data recorded in a .cub/.grd must be exactly in coincidence with that of the grid data used to generate basin, otherwise the integral is meaningless.

Note that if you have made Multiwfn calculate grid data for a real space function, then this grid data will be stored in memory and hence you can directly use it in the basin integration step.

Visualization

In option 0, you can visualize attractors and basins, for example



Most of buttons are self-explained, please feel free to play with them. You can plot basin by choosing corresponding term in the basin list at right-bottom corner. By default, only the grids at basin boundary are shown. If you want the entire basin to be portrayed, "Show basin interior" check box should be selected. At the end of the list, the "Unas" term means unassigned grid during basin generation, while "Boun" term means the grid travelled to box boundary, you can select them to plot the corresponding grids. Some basins are very small and very close to nucleus, such as core-type basin of ELF, they are often screened by molecular structure; if you want to inspect these basins you should deselect "Show molecule" check box.

Light green spheres denote attractors, and corresponding basins are colored as green. If the real space function simultaneously has positive and negative parts, the "attractors" in negative regions (in fact they are repulsors) will be shown as light blue spheres, and corresponding basin will be portrayed as blue color.

If a set of attractors have been clustered as a single one, although all of them will be displayed in the GUI, the index labels shown above them will be the same, which is the index of corresponding degenerate attractor.

After using option 1 to locate the attractors, if you visualize isosurface by the GUI of main function 5 or 13, you can also see the attractors and their labels. This design facilitates comparative study of attractors and isosurfaces.

Detail of all options

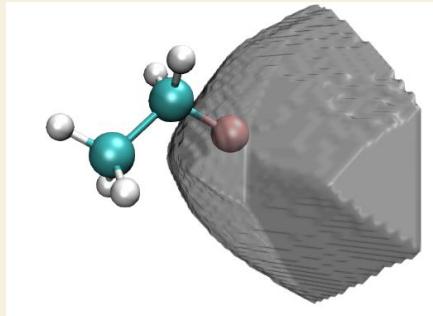
All of the options involved in basin analysis module are described below; some options have

been partially discussed above.

-6 This option controls the parameter used in attractor clustering. If the parameters in suboption 1 and 2 are set as A and B, respectively, that means for any two attractors, if between them the relative value difference is smaller than A, meanwhile their interval is less than $B * \sqrt{dx^2 + dy^2 + dz^2}$, where dx, dy, dz are grid spacings in X, Y, Z, then they will be clustered together. If you want to nullify the automatic clustering step after generating basins, simply set the parameter in suboption 2 to 0.

After basins have been generated, you can also use suboption 3 to manually cluster specified attractors.

-5 Export basins as cube file. The basins in the specified index range will be successively outputted to basinXXXX.cub in current folder, where XXXX is the index of the basin. In the cube file, the value 1 and 0 denote the corresponding grid belongs and not belongs to the basin, respectively. Therefore you can plot isosurface with isovalue=0.5 to visualize the basins in your favourite visualization software. For example, below graph is plotted by VMD, the grey isosurface corresponds to AIM basin of fluorine atom



Note that when Multiwfn asking you "If output internal region of the basin?", if you input "n" rather than "y", then in the exported cube file, only boundary grids of the basin will have value of 1, and the other grids will have value 0. Therefore only interbasin surface will be shown when you visualizing isosurface.

You can also choose 0,0, then basin.cub will be outputted, in which grid value corresponds to basin index. This is important if you would like to

-10 Return to main menu.

-4 Export attractors as pdb file. The coordinate of all attractors will be outputted to attractors.pdb in current folder. In this file, the atom indices correspond to the attractor indices before clustering, while the residue indices indicate the attractor indices after clustering.

-3 Show information of attractors. The coordinate and value of all attractors will be outputted on screen. For degenerate attractors, the shown coordinates and values are the average ones of their member attractors, the coordinate and the value of all of the member attractors will also be individually shown on screen.

-2 Measure distances, angles and dihedral angles between attractors or atoms. As the title says. Please follow the prompts shown on the screen.

-1 Select the method for generating basins. Three methods are available now: (1) On-grid method (2) near-grid method (3) near-grid method with boundary refinement step. Note that the near-grid method used in Multiwfn has been adapted by me, and thus became more robust. In general I recommend method 3, and this is also the default method.

0 Visualize attractors and basins. This option has already been introduced above.

1 Generate basins and locate attractors. Choose a real space function, and select a grid setting, then Multiwfn will calculate grid data, and then generate basins meanwhile locate attractors. After that, as mentioned earlier, if there are some attractors having similar value and closely placed then they will be automatically clustered. If a grid data has been stored in memory, you can directly choose to use this grid data to avoid calculating new grid data. You can use this option to regenerate basins again and again.

Multiwfn provided a lot of ways to define the grid setting. If you want to study the basins in the whole system, in commonly "medium quality grid" is recommended for qualitative analysis purposes; while if you want to obtain higher integration accuracy, "high quality grid" is in general recommended. If you only need to study the basins in a local region of the whole system, you can make the spatial scope of the grid data only cover this region; for this purpose, I suggest you to use the option "Input center coordinate, grid spacing and box length", by which you can easily define the spatial scope of the grid data.

2 Integrate real space functions in the basins. Select a real space function, or select the grid data stored in memory, or choose to use external .cub/.grd file, then the real space function you selected or the real space function recorded in the grid data will be integrated in the basins that have been generated by option 1, the integral in each basin and basin volumes will be outputted. If there are some unassigned grids or the grids travelled to box boundary, their integrals will also be shown.

If the basins you analyzed are AIM basins, it is highly recommended to use option 7 instead to gain much better integration accuracy.

3 Calculate electric multipole moments in the basins. Please consult Section 3.18.3. The differences relative to Section 3.18.3 are that the ranges of integration in our current consideration are basins rather than fuzzy atom spaces, and nuclear positions should be replaced by attractor positions. The multipole moments are outputted up to quadrupole, while octopole and higher orders are not supported, because based on grid integration they are difficult be evaluated accurately.

If the basins you analyzed are AIM basins, it is highly recommended to use option 8 instead to gain much better integration accuracy.

4 Calculate localization index and delocalization index for the basins. Please consult Section 3.18.5. The only difference relative to Section 3.18.5 is that the range of integration in our current consideration is basins rather than fuzzy atom spaces.

5 Calculate and output orbital overlap matrix in all basins to BOM.txt in current folder. All virtual orbitals are ignored.

Below options are available only when the real space function used to define the basins is chosen as electron density

7 Integrate real space functions in AIM basins with mixed type of grids. This option is specific for integrating AIM basins; there are three different ways to realize it:

Hint: Accuracy: $(2) \geq (3) \gg (1)$. Time spent: $(2) > (3) \gg (1)$. Memory requirement: $(3) > (2) = (1)$

(1) Integrate a specific function with atomic-center + uniform grids: Atomic-center grids are mainly used to integrate the real space function near nuclei, while uniform grids are mainly used for other regions. The accuracy is much better than solely using uniform grids (namely option 2 in basin analysis module).

(2) The same as (1), but with exact refinement of basin boundary: Compared to option (2), when integrating the grids at basin boundary, the assignment of these grids will be exactly refined by steepest ascent scheme, hence the result will be more accurate than using option (1), unfortunately the refinement step is time consuming.

(3) The same as (2), but with approximate refinement of basin boundary: This is an approximate version of option (2). The gradient of electron density used in refinement step will not be evaluated exactly, but obtained by trilinear interpolation of pre-calculated grid data of gradient. Although evaluating the grid data of gradient is also time consuming, the overall time cost is lower than option (2), and the loss of accuracy is trivial.

After you used option (2) or (3) once, the assignment of the boundary grids will be updated permanently, that means then if you use option (1), the result will be identical to (2) or (3).

Not only the basin volumes are outputted along with the integrals, the basin volumes with electron density > 0.001 are also outputted, which can be regarded as atomic sizes.

8 Calculate electric multipole moments in AIM basins with mixed type of grids. Similar to option 3, but use mixed atomic-center and uniform grids to calculate electric multipole moments to gain better accuracy without bringing evident additional computational cost. This option is only applicable to AIM basins.

9 Obtain atomic contribution to population of external basins. The external basins means the basin defined by a cube file named *basin.cub* in current folder, in which the grid value corresponds to basin index. This option aims to obtain atomic contribution to population of ELF bond basins (or other type of basins) based on AIM partition. Please check Section 4.17.7 for example.

Notice that if you use option 7 and 8, when you generating AIM basin, the spatial range of the grid data must cover the whole system. If effective core potential (ECP) is used for an atom and meantime you did not allow Multiwfn to supply EDF information (see "isupplyEDF" parameter in settings.ini file), then many attractors will occur at valence region of the atom; before entering option 7 and 8, you have to manually cluster these attractors as single one by suboption 3 in option -6, otherwise the program does not know which attractors correspond to this atom.

For option 3, 4, 5, 7 and 8, if the input file you used does not contain GTF information (e.g. .cub file), then Multiwfn will prompt you to input a new file, which should contain GTF information of present system, you can use for example .wfn/.wfx/.fch/.molden/.gms file as input.

Many examples of this module can be found in Section 4.18.

Information needed: GTFs or grid data, atom coordinates

3.21 Electron excitation analysis (18)

Basic information about electron excitation analysis module

1 Overview

Main function 18 contains a lot of subfunctions aiming for electron excitation analysis, namely characterizing the electron excitation in various ways. All functions in this category fully support single-reference methods (*i.e.* reference wavefunction for generating excited state wavefunction is single Slater-determinant), including TDDFT, TDA-DFT, CIS and TDHF, while ZINDO is also supported by transition density matrix plotting function. Other kinds of methods for excited state problems such as EOM-CCSD, LR-CC2/3, CASSCF, CASPT2 and MRCI are not formally

supported. Examples of some of these electron excitation analysis functions are provided in Section 4.18.

Both closed-shell and open-shell systems are fully supported by all kinds of electron excitation analyses of Multiwfn.

2 Basic knowledges about single-reference methods

Excited state wavefunction (Ψ^{exc}) of CIS and TDA-DFT methods can be represented as

$$\Psi^{\text{exc}} = \sum_{i \rightarrow a} w_i^a \Phi_i^a \equiv \sum_i^{\text{occ}} \sum_a^{\text{vir}} w_i^a \Phi_i^a$$

where i and a respectively run over all occupied and all virtual MOs, similarly hereafterin. Φ_i^a is the configuration state wavefunction corresponding to moving an electron from originally occupied MO i to virtual MO a . w is known as configuration coefficient. The electron excitation in CIS or TDA-DFT framework therefore can be represented as linear combination of orbital pair transitions. The weighting coefficients w satisfy this normalization condition:

$$\sum_{i \rightarrow a} (w_i^a)^2 = 1.0$$

Clearly, the $i \rightarrow a$ orbital pair transition has contribution of $(w_i^a)^2 \times 100\%$ to the electron excitation.

While for TDHF and TDDFT, excited state wavefunction also contains so-called de-excitation part:

$$\Psi^{\text{exc}} = \sum_{i \rightarrow a} w_i^a \Phi_i^a + \sum_{i \leftarrow a} w_i'^a \Phi_i^a$$

where w and w' correspond to configuration coefficient of excitation and de-excitation, respectively. In this cases, the normalization condition becomes:

$$\sum_{i \rightarrow a} (w_i^a)^2 - \sum_{i \leftarrow a} (w_i'^a)^2 = 1.0$$

The MOs used for CIS/TDHF and TDA-DFT/TDDFT are yielded by HF and DFT calculation for ground state of present system, respectively. The Slater determinant consisted of the occupied MOs, namely the ground state wavefunction, is known as reference state. If the reference state is closed-shell, then α and β MOs are exactly matched with each other, and thus $\beta \rightarrow \beta$ orbital transitions have one-to-one correspondence with $\alpha \rightarrow \alpha$ orbital transitions; in this situation, only one set of orbital transition is recorded, and correspondingly, the configuration coefficients are normalized to 0.5 instead of 1.0.

3 Input files

Input file of almost all electron excitation analysis functions are basically the same, except for subfunction 3 (Analyzing charge transfer based on density difference grid data). **Two kinds of input files as needed:**

(1) A file containing basis function and molecular orbital information. .fch/.fchk, .molden and .gms files produced by excited state calculation (*e.g.* TDDFT) can be directly used. This file should be loaded when Multiwfn boots up.

(2) A file containing configuration coefficients of excited states. The path of this kind of file should be inputted when you enter corresponding analysis function, Multiwf will load configuration coefficients from this file. There are three cases, as shown below:

- ◆ Gaussian users: Output file (.out or .log) of CIS, TDHF, TDDFT and TDA-DFT tasks can be used. Since by default Gaussian only outputs the configuration coefficients whose absolute value is larger than 0.1, In order to achieve acceptable accuracy, you must add *IOP(9/40=4)* keyword in the route section so that all configuration coefficients whose magnitude larger than 0.0001 will be printed (If the calculation is found to be too expensive, using *IOP(9/40=3)* instead is also generally acceptable)

Note: DO NOT use the Gaussian output file of optimization or frequency analysis task of excited state as input file! Multiwf cannot properly parse information from this kind of output file!

- ◆ ORCA users: Output file of CIS and TDA-DFT tasks can be used. Beware that *TPrint* keyword should be used within %cis or %tddft, otherwise only very small amount of configuration coefficients will be printed. *TPrint x* means outputting configurations whose contribution to excited state larger than $x * 100\%$. Typically, I suggest using *TPrint 1E-8*. Since contribution is calculated as square of configuration coefficient, *TPrint 1E-8* simply corresponds to outputting configurations who have absolute value of coefficients larger than 1E-4, the effect is identical to *IOP(9/40=4)* in Gaussian. Below is an example input:

```

! PBE0 def2-SVP nopen
%tddft
nroots 8
tprint 1E-8
end

```

Note: It is also possible to use ORCA TDDFT/TDHF output file, but the analysis result may be reliable!!! Because in TD task, ORCA only prints configuration contributions but does not print configuration coefficients, so Multiwf automatically generates the latter by computing square root of the former. This treatment looks reasonable, however when de-excitation is significant, the configuration coefficients yielded in this manner must be nonsense, because in TD task output file, de-excitation contribution and excitation contribution of an orbital pair are summed up and outputted as a single term.

Note: The sTDA and sTDDFT output file of ORCA is not supported (at least for ORCA 4.1.1), because the configuration coefficients of small magnitude cannot be printed out even *TPrint 1E-8* is used.

- ◆ General cases: You can also use plain text file as the input file. The format of transition information should be completely identical to Gaussian output, for instance:

```

Excited State 1 1 5.7945      // Label, index, multiplicity and excitation energy (ev)
    5 -> 6      0.70642      // MO pairs and configuration coefficients
                           // Use a blank line to separate each excited state

Excited State 2 1 7.8943
    5 -> 7      0.63860
    5 -> 8      0.30006

Excited State 3 1 7.8943
    5 -> 7      -0.30006
    5 -> 8      0.63860
    4 <- 8      0.01000

```

Evidently, the above mentioned two kinds of files must correspond to the same geometry and same calculation level. For example, if the MOs in the .fch were produced at B3LYP/6-31G* level

while the Gaussian output file corresponds to the TDDFT task carried out at PBE0/6-31G* level, the analysis results will be completely meaningless.

◆ Special case for GAMESS-US and Firefly users: If you are a user of Firefly or GAMESS-US program, you do not need to separately provide two kinds of files as mentioned above for electron excitation analysis. If the input file used for Multiwfn is TDDFT output file with .gms suffix, the Multiwfn will not only load basis function and molecular orbital information from this file when Multiwfn boots up, but also load configuration coefficients of excited states when performing electron excitation analysis. The *H2CO_TDDFT_Firefly.gms* and *H2CO_TDDFT_GAMESS.gms* in "examples\excit" folder are example file of TDDFT output file of Firefly and GAMESS-US, respectively.

For Firefly user, you should decrease the PRRTOL parameter in \$TDDFT so that more configuration coefficients could be printed.

For GAMESS-US, output file of CIS and TDA-DFT are not supported. In addition, there are no option used to control the printing threshold of configuration coefficients, therefore some analysis result may be not very accurate because some configurations, which have nonnegligible contributions, may be ignored.

3.21.0 Check, modify and export configuration coefficients of an excitation (-1)

This function allows one to check, modify and export coefficient of configuration coefficients. The input files needed by this function have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, then load a file containing configuration coefficients of excited states. The summary of all recognized excited states will be printed on screen, you should select one of them, the configuration coefficients of orbital pairs involved in this electron excitation will be loaded. After that, up to 10 orbital pairs that have largest absolute contribution to the excitation are automatically shown. Then in the newly appeared menu, you can find below options:

1 Set coefficient of an orbital pair: You can use this option to replace the loaded configuration coefficients of an orbital pair with inputted value.

2 Set coefficient for specific range of orbital pairs: This option is used to replace a batch of loaded configuration coefficients with inputted values. You should input index range of the occupied MOs and virtual MOs corresponding to the orbital transitions.

After manually modifying coefficients use above two options, in order to make the modification affect following electron excitation analysis, you should use option -3 to export the modified coefficients as plain text file, and then use this file as the second kind of input file for electron excitation analyses.

-1 Retrieve original coefficient of all orbital pairs: If configuration coefficients have been manually modified by above two options, you can select this option to retrieve the coefficients to the original loaded values.

-2 Print coefficient (and contribution to excitation) of some orbital pairs: You can use this option to print configuration coefficients whose absolute value are larger than specific value,

meantime the corresponding contributions to the electron excitation are shown together. Via this option you can easily find out which orbital pair transition has crucial contribution to the electron excitation.

-3 Export current excitation information to a plain text file: Basic information and configuration coefficients of currently selected excited state can be exported to a specific plain text file. This file can then be employed as the second kind of input file for various electron excitation analysis functions of Multiwfn, e.g. hole-electron analysis and NTO analysis, and then the analysis result will correspond to the modified configuration coefficients.

Obviously, if you set coefficient of some orbital pairs to zero, then their contributions to the quantities you studied will be completely ignored; while if you have cleaned all coefficients except for a specific orbital pair, then the resulting quantities will only reveal characters of this orbital transition.

The example given in Section 4.18.10 utilized this function.

3.21.1 Analyze and visualize hole-electron distribution, transition density, and transition electric/magnetic dipole moment density (1)

This very powerful module is used to analyze and visualize hole-electron distribution, transition density and transition electric/magnetic dipole moment density. Moreover, hole and electron can be decomposed to orbital pair contributions as well as atom and fragment contributions; furthermore, the atom/fragment contributions can be directly plotted as heat map for visual inspection.

3.21.1.1 Theory

There are many knowledge points involved in this module, they will be described below first.

Theory 1: Real space representation of hole and electron

Process of single-electron excitation can be described as "an electron leaves *hole* and goes to *electron*", the "hole" and "electron" can be defined in different ways. If an excitation can be perfectly described as HOMO→LUMO transition, then hole and electron could be simply represented by HOMO and LUMO, respectively. However, in most practical cases, the single orbital pair representation is not suitable, excitations have to be represented as transition of multiple MO pairs with corresponding weighting coefficients.

How to represent hole and electron distributions when there is no single dominant MO pair transition? One way is using natural transition orbital (NTO) analysis, as introduced in Section 3.21.6. Unfortunately, in many cases, even the MOs have been transformed to NTOs, there is still no single NTO pair that has dominating contribution. The best representation of hole and electron may be the one introduced in this Section, the idea was originally proposed by me and my collaborator Cheng Zhong in 2013. Although the method has not been published, you can cite Multiwfn manual if your study have utilized it.

It can be shown that density distribution of hole and electron can be perfectly defined as

$$\begin{aligned}\rho^{\text{hole}}(\mathbf{r}) &= \rho_{\text{(loc)}}^{\text{hole}}(\mathbf{r}) + \rho_{\text{(cross)}}^{\text{hole}}(\mathbf{r}) = \sum_{i \rightarrow a} (w_i^a)^2 \varphi_i(\mathbf{r}) \varphi_i(\mathbf{r}) + \sum_{i \rightarrow a} \sum_{j \neq i \rightarrow a} w_i^a w_j^a \varphi_i(\mathbf{r}) \varphi_j(\mathbf{r}) \\ \rho^{\text{ele}}(\mathbf{r}) &= \rho_{\text{(loc)}}^{\text{ele}}(\mathbf{r}) + \rho_{\text{(cross)}}^{\text{ele}}(\mathbf{r}) = \sum_{i \rightarrow a} (w_i^a)^2 \varphi_a(\mathbf{r}) \varphi_a(\mathbf{r}) + \sum_{i \rightarrow a} \sum_{j \neq i \rightarrow a} w_i^a w_j^b \varphi_a(\mathbf{r}) \varphi_b(\mathbf{r})\end{aligned}$$

note that the notions used here

$$\sum_{i \rightarrow a} \equiv \sum_i^{\text{occ}} \sum_a^{\text{vir}} \quad \sum_{i \rightarrow a} \sum_{j \neq i \rightarrow a} \equiv \sum_i^{\text{occ}} \sum_{j \neq i}^{\text{occ}} \sum_a^{\text{vir}}$$

where φ denotes MO wavefunction. "loc" and "cross" stand for the contribution of local term and cross term to the hole and electron distribution, respectively. Note that the definition of hole and electron given above is in density form rather than wavefunction form, hence the hole and electron do not have phase (If you really need phase information of hole and electron, you should resort on NTO analysis, see Section 3.21.6).

Due to the orthonormality of MOs and the fact that the sum of square of all configuration coefficients is 1.0, it is clear that

$$\int \rho^{\text{hole}}(\mathbf{r}) d\mathbf{r} = 1 \quad \int \rho^{\text{ele}}(\mathbf{r}) d\mathbf{r} = 1$$

This is an important property that any reasonable definition of hole and electron distribution should satisfy, it indicates that one electron is excited.

The overlap function between hole and electron distribution can be defined as

$$S_m(\mathbf{r}) = \min[\rho^{\text{hole}}(\mathbf{r}), \rho^{\text{ele}}(\mathbf{r})]$$

namely taking the minimal value of ρ^{hole} and ρ^{ele} everywhere. Another function for measuring the overlap is

$$S_r(\mathbf{r}) = \sqrt{\rho^{\text{hole}}(\mathbf{r}) \rho^{\text{ele}}(\mathbf{r})}$$

It is evident that S_r is always equal or larger than S_m . Both the two definitions are reasonable, but I prefer to use S_r , since its graphical effect is better and its mathematical meaning is more clear.

The charge density difference (CDD) between excited state and ground state can be easily evaluated as

$$\Delta\rho(\mathbf{r}) = \rho^{\text{ele}}(\mathbf{r}) - \rho^{\text{hole}}(\mathbf{r})$$

NOTE: Beware that if you are a Gaussian user, the $\Delta\rho$ calculated in this way is obviously different to the $\Delta\rho$ produced via subtracting excited state density by ground state density, unless you specified keyword *density=rhocci* when generating .wfn/wfx file of excited state. Because by default the excited state density exported to .wfn/wfx file by Gaussian is relaxed density rather than unrelaxed density (which is directly constructed by MOs and excited state configuration coefficients). In other words, unrelaxed excited state density can be simply written as

$\rho^{\text{excited}}(\mathbf{r}) = \rho^{\text{ground}}(\mathbf{r}) - \rho^{\text{hole}}(\mathbf{r}) + \rho^{\text{ele}}(\mathbf{r})$, while deriving relaxed excited state density requires employing the very complicated "Z-vector" method.

After generalization, above definitions of hole and electron can also be applied to TDHF and TDDFT cases, where de-excitations must be taken into account. The generalized local terms are

$$\begin{aligned}\rho_{(\text{loc})}^{\text{hole}} &= \sum_{i \rightarrow a} (w_i^a)^2 \rho_i - \sum_{i \leftarrow a} (w_i'^a)^2 \rho_i \\ \rho_{(\text{loc})}^{\text{ele}} &= \sum_{i \rightarrow a} (w_i^a)^2 \rho_a - \sum_{i \leftarrow a} (w_i'^a)^2 \rho_a\end{aligned}$$

where $\rho = |\varphi_i|^2$ stands for electron density of orbital i , w' denotes configuration coefficient of de-excitation. The generalized cross terms are

$$\begin{aligned}\rho_{(\text{cross})}^{\text{hole}} &= \sum_{i \rightarrow a} \sum_{j \neq i \rightarrow a} w_i^a w_j^a \varphi_i \varphi_j - \sum_{i \leftarrow a} \sum_{j \neq i \leftarrow a} w_i'^a w_j'^a \varphi_i \varphi_j \\ \rho_{(\text{cross})}^{\text{ele}} &= \sum_{i \rightarrow a} \sum_{i \rightarrow b \neq a} w_i^a w_i^b \varphi_a \varphi_b - \sum_{i \leftarrow a} \sum_{i \leftarrow b \neq a} w_i'^a w_i'^b \varphi_a \varphi_b\end{aligned}$$

If combining local and cross terms together, the expressions of hole and electrons could be simply expressed as

$$\begin{aligned}\rho^{\text{hole}} &= \sum_{i, j \rightarrow a} w_i^a w_j^a \varphi_i \varphi_j - \sum_{i, j \leftarrow a} w_i'^a w_j'^a \varphi_i \varphi_j \\ \rho^{\text{ele}} &= \sum_{i \rightarrow a, b} w_i^a w_i^b \varphi_a \varphi_b - \sum_{i \leftarrow a, b} w_i'^a w_i'^b \varphi_a \varphi_b\end{aligned}$$

Theory 2: Contribution of MOs, basis functions, atoms and fragments to hole and electron

In order to investigate which MOs have significant contributions to hole and electron, I defined the contribution of occupied MO to hole and contribution of virtual MO to electron as follows

$$\Theta_i^{\text{hole}} = \sum_a [(w_i^a)^2 - (w_i'^a)^2] \quad \Theta_a^{\text{ele}} = \sum_i [(w_i^a)^2 - (w_i'^a)^2]$$

Below normalization conditions are held evidently:

$$\sum_i \Theta_i^{\text{hole}} = 1 \quad \sum_a \Theta_a^{\text{ele}} = 1$$

Contribution to hole/electron by an atom can be derived as follows. Considering the normalization condition of the hole (de-excitation part is temporarily ignored for simplicity)

$$\begin{aligned}\int \left(\sum_{i, j \rightarrow a} w_i^a w_j^a \varphi_i \varphi_j \right) d\mathbf{r} &= 1 \\ \int \left(\sum_{i, j \rightarrow a} w_i^a w_j^a \sum_{\mu} \sum_{\nu} C_{\mu, i} C_{\nu, j} \chi_{\mu} \chi_{\nu} \right) d\mathbf{r} &= 1 \\ \sum_{i, j \rightarrow a} w_i^a w_j^a \sum_{\mu} \sum_{\nu} C_{\mu, i} C_{\nu, j} \int \chi_{\mu} \chi_{\nu} d\mathbf{r} &= 1 \\ \sum_{i, j \rightarrow a} w_i^a w_j^a \sum_{\mu} \sum_{\nu} C_{\mu, i} C_{\nu, j} S_{\mu, \nu} &= 1\end{aligned}$$

where χ denotes basis function, S and C are overlap matrix and coefficient matrix, respectively. If we employ Mulliken-like method to partition the $\sum_{\mu} \sum_{\nu} C_{\mu, i} C_{\nu, j} S_{\mu, \nu}$ term as atomic contributions,

then we can define contribution of atom A to hole in below form

$$\Theta_A^{\text{hole}} = \sum_{i, j \rightarrow a} w_i^a w_j^a \frac{1}{2} \left(\sum_{\mu \in A} \sum_{\nu} C_{\mu, i} C_{\nu, j} S_{\mu, \nu} + \sum_{\mu} \sum_{\nu \in A} C_{\mu, i} C_{\nu, j} S_{\mu, \nu} \right)$$

3 Functions

We can similarly apply above treatment on de-excitation part of hole as well as electron. The actual equations used to evaluate atomic contribution to hole and electron are

$$\Theta_A^{\text{hole}} = \sum_{i,j \rightarrow a} w_i^a w_j^a \frac{1}{2} \left(\sum_{\mu \in A} \sum_{\nu} T_{\mu,\nu}^{ij} + \sum_{\mu} \sum_{\nu \in A} T_{\mu,\nu}^{ij} \right) - \sum_{i,j \leftarrow a} w_i'^a w_j'^a \frac{1}{2} \left(\sum_{\mu \in A} \sum_{\nu} T_{\mu,\nu}^{ij} + \sum_{\mu} \sum_{\nu \in A} T_{\mu,\nu}^{ij} \right)$$

$$\Theta_A^{\text{ele}} = \sum_{i \rightarrow a,b} w_i^a w_i^b \frac{1}{2} \left(\sum_{\mu \in A} \sum_{\nu} T_{\mu,\nu}^{ab} + \sum_{\mu} \sum_{\nu \in A} T_{\mu,\nu}^{ab} \right) - \sum_{i \leftarrow a,b} w_i'^a w_i'^b \frac{1}{2} \left(\sum_{\mu \in A} \sum_{\nu} T_{\mu,\nu}^{ab} + \sum_{\mu} \sum_{\nu \in A} T_{\mu,\nu}^{ab} \right)$$

where \mathbf{T} is an intermediate matrix for facilitating calculation, it is defined as $T_{\mu,\nu}^{ij} = C_{\mu,i} C_{\nu,j} S_{\mu,\nu}$.

Fragment contributions to hole and electron can be simply evaluated by summing up atomic contributions

$$\Theta_{\text{frag}}^{\text{hole}} = \sum_{A \in \text{frag}} \Theta_A^{\text{hole}} \quad \Theta_{\text{frag}}^{\text{ele}} = \sum_{A \in \text{frag}} \Theta_A^{\text{ele}}$$

Contribution of a basis functions μ to hole and electron can be defined as

$$\Theta_{\mu}^{\text{hole}} = \sum_{i,j \rightarrow a} w_i^a w_j^a \frac{1}{2} \left(\sum_{\nu} T_{\mu,\nu}^{ij} + \sum_{\nu} T_{\nu,\mu}^{ij} \right) - \sum_{i,j \leftarrow a} w_i'^a w_j'^a \frac{1}{2} \left(\sum_{\nu} T_{\mu,\nu}^{ij} + \sum_{\nu} T_{\nu,\mu}^{ij} \right)$$

$$\Theta_{\mu}^{\text{ele}} = \sum_{i \rightarrow a,b} w_i^a w_i^b \frac{1}{2} \left(\sum_{\nu} T_{\mu,\nu}^{ab} + \sum_{\nu} T_{\nu,\mu}^{ab} \right) - \sum_{i \leftarrow a,b} w_i'^a w_i'^b \frac{1}{2} \left(\sum_{\nu} T_{\mu,\nu}^{ab} + \sum_{\nu} T_{\nu,\mu}^{ab} \right)$$

In order to significantly save computational time, Multiwfn ignores all terms if magnitude of product of corresponding two configuration coefficients is less than 0.001. The loss of accuracy due to this trick is negligible.

Furthermore, we can define contribution of atom and fragment to charge density difference as

$$\Theta_A^{\text{CDD}} = \Theta_A^{\text{ele}} - \Theta_A^{\text{hole}} \quad \Theta_{\text{frag}}^{\text{CDD}} = \Theta_{\text{frag}}^{\text{ele}} - \Theta_{\text{frag}}^{\text{hole}}$$

Overlap between hole and electron in atom and fragment spaces are defined as geometry average of their contributions:

$$\Theta_A^{\text{ovlp}} = \sqrt{\Theta_A^{\text{ele}} \Theta_A^{\text{hole}}} \quad \Theta_{\text{frag}}^{\text{ovlp}} = \sqrt{\Theta_{\text{frag}}^{\text{ele}} \Theta_{\text{frag}}^{\text{hole}}}$$

Notice that the overlap in this form is not additive, namely $\Theta_A^{\text{ovlp}} + \Theta_B^{\text{ovlp}} \neq \Theta_{AB}^{\text{ovlp}}$.

Some atomic contributions derived in above way may be small negative values in certain situations, this is due to the theoretical shortcoming of Mulliken type of partition, obviously in this case the overlap between hole and electron in corresponding atom spaces cannot be evaluated, so Multiwfn automatically sets the overlap values to zero. The Mulliken type of partition is well-known incompatible with diffuse functions, therefore basis set containing diffuse functions must not be employed in the excited state calculation if you intend to evaluate atomic contributions to hole and electron in above way.

Due to the extreme flexibility of Multiwfn, there are also other ways of determining atomic contributions to hole and electron, however you have to manually evaluate them. For example, if you want to employ fuzzy partition for hole and electron, you should first export cube file of hole or electron, then set "iuserfunc" in settings.ini to -1 (in this case the user-defined function will correspond to the interpolated function based on the grid data), then load hole or electron cube file

into Multiwfn, use subfunction 1 of main function 15 to integrate user-defined function in each atomic fuzzy space. Although this way is more time-consuming, it is quite robust and completely compatible with diffuse function, and no unphysical negative value will occur. If in the main function 15, you first select option -4 to define a fragment and then use subfunction 1 to integrate user-defined function, then the sum of results of all atoms will correspond to the fragment contribution.

Theory 3: Quantitative characterization of hole and electron distribution in the whole space

The overall distribution of hole and electron can be quantitatively characterized in following ways, they are quite useful for identifying type of electron excitations.

To characterize overlapping extent of hole and electron, S_m index and S_r index are defined as follows (S_r must be equal or larger than S_m index)

$$S_m \text{ index} = \int S_m(\mathbf{r}) d\mathbf{r} \equiv \int \min[\rho^{\text{hole}}(\mathbf{r}), \rho^{\text{ele}}(\mathbf{r})] d\mathbf{r}$$

$$S_r \text{ index} = \int S_r(\mathbf{r}) d\mathbf{r} \equiv \int \sqrt{\rho^{\text{hole}}(\mathbf{r}) \rho^{\text{ele}}(\mathbf{r})} d\mathbf{r}$$

Centroid can be calculated to reveal most representative position of hole and electron distribution. For example, X coordinate of centroid of electron is written as

$$X_{\text{ele}} = \int x \rho^{\text{ele}}(\mathbf{r}) d\mathbf{r}$$

where x is X component of position vector \mathbf{r} .

The charge transfer (CT) length in X/Y/Z can be measured by distance between centroid of hole and electron in corresponding directions:

$$D_x = |X_{\text{ele}} - X_{\text{hole}}| \quad D_y = |Y_{\text{ele}} - Y_{\text{hole}}| \quad D_z = |Z_{\text{ele}} - Z_{\text{hole}}|$$

The total magnitude of CT length is referred to as D index:

$$D \text{ index} = |\mathbf{D}| \equiv \sqrt{(D_x)^2 + (D_y)^2 + (D_z)^2}$$

It is noteworthy that the variation of dipole moment of excited state (corresponding to unrelaxed density) with respect to ground state in X, Y and Z can be simply calculated as

$$\Delta\mu_x = -(X_{\text{ele}} - X_{\text{hole}}) \quad \Delta\mu_y = -(Y_{\text{ele}} - Y_{\text{hole}}) \quad \Delta\mu_z = -(Z_{\text{ele}} - Z_{\text{hole}})$$

The RMSD of hole and electron can be used to characterize their extent of spatial distribution. For example, X component of RMSD of hole is expressed as

$$\sigma_{\text{hole},x} = \sqrt{\int (x - X_{\text{hole}})^2 \rho^{\text{hole}}(\mathbf{r}) d\mathbf{r}}$$

The $|\sigma_{\text{hole}}|$ and $|\sigma_{\text{ele}}|$ are referred to as σ_{hole} and σ_{ele} indices, they measure overall RMSD of hole and electron, respectively.

The difference between RMSD of electron and hole in X/Y/Z direction can be measured via $\Delta\sigma_\lambda$, while overall difference can be measured via $\Delta\sigma$ index

$$\Delta\sigma_\lambda = \sigma_{\text{ele},\lambda} - \sigma_{\text{hole},\lambda} \quad \lambda = \{x, y, z\}$$

$$\Delta\sigma \text{ index} = |\boldsymbol{\sigma}_{\text{ele}}| - |\boldsymbol{\sigma}_{\text{hole}}|$$

H_λ measures average degree of spatial extension of hole and electron distribution in X/Y/Z direction, H_{CT} is that in CT direction, and H index is an overall measure

$$H_\lambda = (\sigma_{\text{ele},\lambda} + \sigma_{\text{hole},\lambda})/2 \quad \lambda = \{x, y, z\}$$

$$H_{\text{CT}} = |\mathbf{H} \cdot \mathbf{u}_{\text{CT}}|$$

$$H \text{ index} = (|\boldsymbol{\sigma}_{\text{ele}}| + |\boldsymbol{\sigma}_{\text{hole}}|)/2$$

where \mathbf{u}_{CT} is unit vector in CT direction and can be straightforwardly derived using centroid of hole and electron.

t index is designed to measure separation degree of hole and electron in CT direction:

$$t \text{ index} = D \text{ index} - H_{\text{CT}}$$

If t index < 0, it implies that hole and electron is not substantially separated due to CT. Clear separation of hole and electron distributions must correspond to evidently positive t index.

There are often many nodes or complicated fluctuations in hole and electron distributions. In order to make visual study of hole and electron easier, C_{hole} and C_{ele} functions are defined as follows. The function behavior of C_{hole} and C_{ele} is similar to Gaussian function, they are highly smooth functions, the value asymptotically approaches zero from centroid of hole/electron.

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

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

The factor A is introduced so that C_{hole} and C_{ele} are normalized.

In fact, the definition of RMSD, C_{hole} , C_{ele} , H and t indices introduced above was motivated by *J. Chem. Theory Comput.*, 7, 2498 (2011), these quantities were originally used to analyze electron excitation based on density difference, but I found all of them work well under the framework of hole-electron analysis. Also note that many details of these indices have been modified when introduced to hole-electron analysis framework.

The above defined quantitative indices could be used for distinguishing type of electron excitation. My empirical rule is summarized as follows, it should be suitable for most cases.

Excitation type	Index			
	D	S_r	t	$\Delta\sigma$
LE	small	medium~large	<0	small
Single direction CT	large	?	?	?
Centrosymmetric CT	small	?	<0	large
Rydberg	small	Small	<0	large

In the table, three kinds of excitations are involved:

- Local excitation (LE): The hole and electron occupy similar spatial region.
- Charge-transfer excitation (CT): The spatial separation of hole and electron is large, leading to evident displacement of charge density. The CT may be single directional or multiple directional (centrosymmetric CT is a special case of the latter).
- Rydberg excitation: Electron mainly consists of very diffuse MOs, therefore the overlap between electron and hole must be small. This type of excitation in general does not lead to prominent long-range displacement of charge density.

Theory 4: Transition density matrix and transition density

(One-electron, spinless) transition density matrix between excited state and ground state of an N -electron system in real space representation is defined as follows (real type of wavefunctions is assumed, so complex conjugation sign is omitted)

$$T(\mathbf{r}; \mathbf{r}') \equiv T(\mathbf{r}_1; \mathbf{r}'_1) = \int \Phi^0(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) \Psi^{\text{exc}}(\mathbf{x}'_1, \mathbf{x}_2, \dots, \mathbf{x}_N) d\sigma_1 d\mathbf{x}_2 d\mathbf{x}_3 \cdots d\mathbf{x}_N$$

where Φ^0 is Slater-determinant of ground state wavefunction. \mathbf{x} is spin-space coordinate, σ stands for spin coordinate. The T is called as matrix because it has two continuous indices.

For excited state wavefunction generated by single-reference methods, after expanding Ψ^{exc} and applying Slater-Condon rule, it can be easily shown that T can be explicitly written as

$$T(\mathbf{r}; \mathbf{r}') = \sum_i \sum_a w_i^a \varphi_i(\mathbf{r}) \varphi_a(\mathbf{r}')$$

If we only take the diagonal terms of the transition density matrix, then we obtain transition density

$$T(\mathbf{r}) = \sum_i \sum_a w_i^a \varphi_i(\mathbf{r}) \varphi_a(\mathbf{r})$$

$T(\mathbf{r})$ can be studied as a common real space function, for example, visualized in terms of isosurface map. Assuming that there is only one dominant orbital transition, for example, HOMO→LUMO, then $T(\mathbf{r})$ is simply $\varphi_{\text{HOMO}}(\mathbf{r}) \varphi_{\text{LUMO}}(\mathbf{r})$. Therefore, it is easy to understand, if a region has large magnitude of transition density, the hole and electron must be strongly coupled in this region; while if a region has small distribution of $T(\mathbf{r})$, then overlap between hole and electron in this area should be insignificant. Clearly, $T(\mathbf{r})$ is a useful function for characterizing underlying nature of electron excitation, and its main distribution characteristics is closely related to the $S_r(\mathbf{r})$ function.

Note that due to the orthonormality of MOs, integral of $T(\mathbf{r})$ over the whole space is exactly zero. If the excited state and ground state correspond to different spin states, due to the orthonormality of spin coordinate, $T(\mathbf{r}; \mathbf{r}')$ must be a zero matrix, and $T(\mathbf{r})$ is correspondingly zero everywhere. However, notice that only spatial part of $T(\mathbf{r})$ is taken into account when Multiwfn evaluates it, therefore you are still able to study $T(\mathbf{r})$ for e.g. S0→T1 excitation.

Theory 5: Transition electric/magnetic dipole moment density

Note that there are many kinds of transition dipole moment, including transition electric dipole moment, transition magnetic dipole moment, transition velocity dipole moment and so on. The word "transition dipole moment" commonly refers to transition electric dipole moment.

X, Y and Z components of transition electric dipole moment density can be written as negative of product of X, Y and Z coordinate variables and transition density, respectively:

$$T_x(\mathbf{r}) = -xT(\mathbf{r}) \quad T_y(\mathbf{r}) = -yT(\mathbf{r}) \quad T_z(\mathbf{r}) = -zT(\mathbf{r})$$

Integrating transition electric dipole moment density over the whole space yields transition dipole moment **D**

$$D_x = \int T_x(\mathbf{r}) d\mathbf{r} \quad D_y = \int T_y(\mathbf{r}) d\mathbf{r} \quad D_z = \int T_z(\mathbf{r}) d\mathbf{r}$$

Obviously, one can conveniently study contribution to transition electric dipole moment of various molecular regions by plotting transition electric dipole moment density.

Next, we look at transition magnetic dipole moment. The operator for magnetic dipole moment due to movement of electrons is the angular momentum operator **L** (see e.g. *Theoret. Chim. Acta*, **6**, 341)

$$\begin{aligned} \mathbf{L} &= -i (\mathbf{r} \times \nabla) = \hat{\mathbf{i}} L_x + \hat{\mathbf{j}} L_y + \hat{\mathbf{k}} L_z \\ &= -i \left[\hat{\mathbf{i}} \left(y \frac{\partial}{\partial z} - z \frac{\partial}{\partial y} \right) + \hat{\mathbf{j}} \left(z \frac{\partial}{\partial x} - x \frac{\partial}{\partial z} \right) + \hat{\mathbf{k}} \left(x \frac{\partial}{\partial y} - y \frac{\partial}{\partial x} \right) \right] \end{aligned}$$

where **i**, **j**, **k** are unity vectors in X, Y and Z directions, respectively. Therefore, the X component of transition magnetic dipole moment can be explicitly defined as below. Noticed that in order to provide a real value I ignored the imaginary and negative signs simultaneously; the symbol " \leftarrow " denotes de-excitation MO pairs in TDHF/TDDFT formalism.

$$M_x = \langle \Phi^0 | y \frac{\partial}{\partial z} - z \frac{\partial}{\partial y} | \Psi^{\text{exc}} \rangle = \sum_{i \rightarrow a} w_i^a \langle \varphi_i | y \frac{\partial}{\partial z} - z \frac{\partial}{\partial y} | \varphi_a \rangle - \sum_{j \leftarrow b} w_j^b \langle \varphi_j | y \frac{\partial}{\partial z} - z \frac{\partial}{\partial y} | \varphi_b \rangle$$

M_y and M_z can be defined similarly.

We can define transition magnetic dipole moment density component $m_i(\mathbf{r})$ by considering the relationship $M_i = \int m_i(\mathbf{r}) d\mathbf{r}$ $i = x, y, z$, so that distribution of transition magnetic dipole moment can be visualized in terms of e.g. isosurface map. Explicit expression of $m_i(\mathbf{r})$ of X component is given below, Y and Z components can be defined similarly.

$$m_x(\mathbf{r}) = \sum_{i \rightarrow a} w_i^a \varphi_i(\mathbf{r}) \left[y \frac{\partial \varphi_a}{\partial z}(\mathbf{r}) - z \frac{\partial \varphi_a}{\partial y}(\mathbf{r}) \right] - \sum_{j \leftarrow b} w_j^b \varphi_j(\mathbf{r}) \left[y \frac{\partial \varphi_b}{\partial z}(\mathbf{r}) - z \frac{\partial \varphi_b}{\partial y}(\mathbf{r}) \right]$$

Theory 6: Coulomb attraction between hole and electron (exciton binding energy)

The "electron" of course carries negative charge, while "hole" can be regarded as carrying positive charge, therefore formally there is Coulomb attractive energy between them, it is also known as exciton binding energy. This term can be calculated via simple Coulomb formula (in atomic unit form):

$$E_C = \iint \frac{\rho^{\text{hole}}(\mathbf{r}_1) \rho^{\text{ele}}(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2$$

Some discussions about the exciton binding energy can be found in e.g. *J. Chem. Phys.*, **143**, 244905 (2015) and *J. Phys. Chem. C*, **121**, 17088 (2017). Note that the the exciton binding energy calculated in this form is different to the exciton binding energy defined in another form, namely $E_C = (\text{IP}-\text{EA}) - E_{\text{optical gap}}$ (see *Mater. Horiz.*, **1**, 17 (2014) for more details), because the practical electron ionization and electron affinity processes involve orbital relaxation effect; moreover, there is an exchange term

in E_C , while all of these factors are ignored in the evaluation of exciton binding energy in Multiwfn.

In Multiwfn, above integral is directly calculated based on evenly distributed grid data of hole and electron. Notice that although the code has been substantially optimized and parallelized, the computational cost is still high, therefore you need to wait patiently during calculation. The cost is formally proportionally to square of the number of grids; therefore, the cost of medium quality grid will be higher than low quality grid by one order of magnitude.

3.21.1.2 Usage and Functions

The input files needed by present module have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states. The summary of recognized excited states will be printed on screen, you should select the excited state that you want to carry out aforementioned analyses. Each time only one state can be analyzed, if you want to analyze another state, you should exit this function, then enter again and select another state.

Present module has many functions, they will be described below in turn.

Function 1: Visualize and analyze hole, electron and transition density and so on

After you enter this function, you are requested to set up grid data, then grid data will be calculated for hole distribution, electron distribution, overlap of hole and electron, transition density, transition electric/magnetic dipole moment density, charge density difference and $C_{\text{ele}}/C_{\text{hole}}$ functions.

After calculation of grid data is finished, various quantities introduced in Section 3.21.1.1 will be evaluated based on the evenly distributed grid data and then shown on screen, their meanings should be very easy to understand. The outputted transition electric/magnetic dipole moment is calculated by integrating grid data of transition dipole moment density, the value should be very close to the one directly outputted by quantum chemistry program. The ideal value of the integral of hole or electron over the whole space is 1.0, while for transition density the ideal value is 0. If the actual outputted values deviate too far from expected values, then the printed t index, H index, D index, S_m index and so on may be unreliable. There are three reasons may lead to this problem:

- (1) The grid quality is too poor. Higher number of grid points should be used
- (2) The spatial extent of the grid data is too narrow, you should enlarge extension distance so that the grid data could cover broader regions
- (3) You forgot to use the $IOp(9/40=x)$ option mentioned at the beginning of Section 3.21, as a result, only very small number of configuration coefficients are loaded

In post-process menu, grid data of hole, electron, transition density, S_m/S_r and so on can be directly visualized as isosurface map, or be exported as cube file in current folder by corresponding options. You can also choose corresponding option to calculate Coulomb attractive energy between hole and electron distribution, notice that this calculation is expensive even if you only choose low quality grid.

By default, the transition magnetic dipole moment density is not evaluated because it is less important than the transition electric dipole moment density. If you want to calculate it, select option -1 before entering this function.

For large systems, if computational cost for grid data is too high and you only need to qualitatively examine isosurface map of hole, electron, transition density and so on, in Gaussian you can safely use $IOp(9/40=3)$ instead of $IOp(9/40=4)$, so that smaller number of configurations will be taken into account.

Function 2: Show molecular orbital contribution to hole and electron distribution

You only need to input printing threshold, then contribution of MO to hole and electron distribution will be shown. This function is very useful to identify which MOs have significant contribution to hole and electron. Below is an output example:

```
MO    126, Occ:  2.00000   Hole:  0.29664   Electron:  0.00000
MO    127, Occ:  2.00000   Hole:  0.19783   Electron:  0.00000
MO    128, Occ:  2.00000   Hole:  0.38666   Electron:  0.00000
MO    130, Occ:  0.00000   Hole:  0.00000   Electron:  0.08058
MO    131, Occ:  0.00000   Hole:  0.00000   Electron:  0.46282
MO    132, Occ:  0.00000   Hole:  0.00000   Electron:  0.16703
MO    133, Occ:  0.00000   Hole:  0.00000   Electron:  0.22976
sum of hole: 1.00000   sum of electron: 1.00000
```

Function 3: Show atom or fragment contribution to hole and electron and plot the contributions as heat map

After you enter this function, many quantities mentioned in "Theory 2" of Section 3.21.1.1 will be printed on screen, below is an output example. Mulliken type of partition is used to derive the atomic contributions.

Contribution of each non-hydrogen atom to hole and electron:

```
1(C ) Hole: 1.37 % Electron: 8.96 % overlap: 3.50 % diff.: 7.59 %
2(C ) Hole: 11.86 % Electron: 0.74 % overlap: 2.97 % diff.: -11.11 %
3(C ) Hole: 8.96 % Electron: 11.00 % overlap: 9.93 % diff.: 2.04 %
...[ignored]
14(N ) Hole: 0.18 % Electron: 23.80 % overlap: 2.07 % diff.: 23.62 %
15(O ) Hole: 3.07 % Electron: 17.23 % overlap: 7.28 % diff.: 14.16 %
16(O ) Hole: 3.07 % Electron: 17.23 % overlap: 7.28 % diff.: 14.16 %
```

In the output, the "Overlap" is simply the geometry average of "Hole" and "Electron", while "Diff." is obtained by subtracting "Hole" from "Electron". Since hydrogens commonly do not participate in electronic excitations of interest, by default hydrogens are ignored, but you can choose "Toggle if taking hydrogens into account" option to switch status.

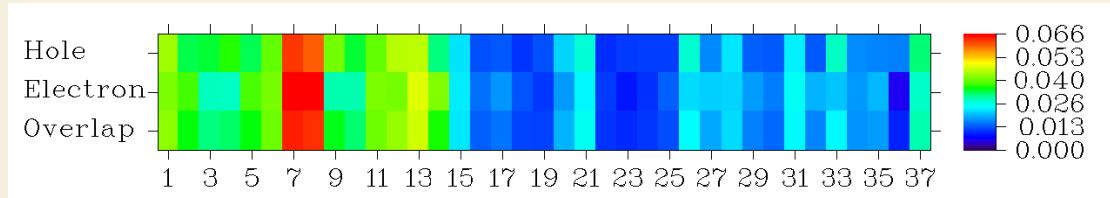
If you need contribution of molecular fragments to above mentioned quantities, you can select option "-1 Load fragment definition" and then input the number of fragments and atomic index of each fragment in turn. Fragment definition can also be loaded from an external plain text file, in which each fragment occupies a line, for example

```
1,3,6-10,12
2,4,5
11
13-15
```

This example totally defines four fragments, the first fragment consists of atom 1,3,6,7,8,9,10,12.

Once defining fragments is completed, contribution of the fragments to various quantities will be immediately printed on screen.

Composition of atom/fragment in hole and electron, as well as hole-electron overlap in various atom/fragment spaces can be plotted as heat map, so that their distribution character can be very vividly exhibited. Below is an example, the color correspond to function value, while abscissa corresponds to atom index.



From the graph, you can immediately recognize that this is a local excitation, since most part of both hole and electron are distributed on the fragment consisted of atom 1~14. In particular, atoms 7 and 8 are the atoms that contribute most to this electron excitation. If you load fragment definition before plotting, then the abscissa of the heat map will correspond to fragment index. In the menu, there are also options used to adjust color scale, ratio of the map and stepsize between labels in X axis.

An very detailed example of this hole-electron module is given in Section 4.18.1. Example of analyzing transition density and transition dipole moment density using this module is given in Section 4.18.2.1. More discussion and examples can be found from my blog article "Using Multiwfn to perform hole-electron analysis to fully investigate electron excitation character" (in Chinese, <http://sobereva.com/434>).

Information needed: See beginning of Section 3.21.

3.21.2 Plot atom/fragment transition matrix of various kinds as heat map (2)

This function is used to plot atom transition matrix (ATM) of various kinds as heat map (color-filled matrix map). The ATM refers to any kind of atom based matrix that represents electron transition information between two states. For example, it may correspond to the atom based transition density matrix (see below), the atom-atom charge transfer matrix, the atom transition dipole moment matrix and so on. In this function, the ATM can also be further transformed to fragment transition matrix (FTM) and then plotted as heat map.

Although this function can also plot heat map for other matrices, the major purpose of developing this function is plotting atom or fragment based transition density matrix, therefore I will first introduce theories related to transition density matrix.

Theories about transition density matrix (TDM)

Below, the word "TDM" refers to the transition density matrix in basis function representation. The TDM between ground state and an excited state can be calculated as (de-excitation transitions have been ignored for simplicity)

$$P_{\mu\nu}^{\text{tran}} = \sum_i^{\text{occ}} \sum_a^{\text{vir}} w_i^a C_{\mu i} C_{\nu a}$$

where $C_{\mu i}$ denotes the expansion coefficient of basis function μ in MO i . It is worth to note in passing that the TDM in real space representation, which is introduced in Section 3.21.1.1, can be constructed easily via TDM in basis function representation (χ stands for basis function):

$$T(\mathbf{r}; \mathbf{r}') = \sum_{\mu} \sum_{\nu} P_{\mu\nu}^{\text{tran}} \chi_{\mu}(\mathbf{r}) \chi_{\nu}(\mathbf{r}')$$

The off-diagonal elements of TDM essentially represent the coupling between various basis functions during electron excitation. Assume there are only two basis functions and meantime the excitation can be perfectly represented as $i \rightarrow a$ MO transition, then the TDM could be explicitly written as below form (notice that the index of the elements has been rearranged according to convention of TDM heat map)

$$\mathbf{P}^{\text{tran}} = \begin{bmatrix} 1,2 & 2,2 \\ 1,1 & 2,1 \end{bmatrix} = \begin{bmatrix} C_{1i}C_{2a} & C_{2i}C_{2a} \\ C_{1i}C_{1a} & C_{2i}C_{1a} \end{bmatrix}$$

If magnitude of off-diagonal element $P_{1,2}^{\text{tran}}$ is large, it implies that basis functions 1 and 2 significantly participate in occupied orbital i and virtual orbital a , respectively. More generally, we may say that basis functions 1 and 2 have large contribution to hole and electron, respectively, in this case the two basis functions are strongly coupled during the excitation. The diagonal terms are also meaningful, if element $P_{\mu,\mu}^{\text{tran}}$ has large magnitude, it implies that basis function μ must *simultaneously* have large contribution to both hole and electron.

Since TDM in general is not a symmetric matrix, in order to make certain discussions easier, some literatures employ below symmetrized form

$$\bar{P}_{\mu\nu}^{\text{tran}} = \frac{P_{\mu\nu}^{\text{tran}} + P_{\nu\mu}^{\text{tran}}}{\sqrt{2}}$$

The TDM can be contracted to atom based form according to correspondence between basis functions and atoms, it will be symbolized as \mathbf{p} . In Multiwfn, below construction ways are available:

$$\text{Way 1: } p_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} (P_{\mu\nu}^{\text{tran}})^2$$

$$\text{Way 2: } p_{AB} = \sqrt{\sum_{\mu \in A} \sum_{\nu \in B} (P_{\mu\nu}^{\text{tran}})^2}$$

$$\text{Way 3: } p_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} |P_{\mu\nu}^{\text{tran}}|$$

$$\text{Way 4: } p_{AA} = \sum_{\mu \in A} |P_{\mu\mu}^{\text{tran}}|, \quad p_{A \neq B}^{\text{tran}} = \sqrt{\sum_{\mu \in A} \sum_{\nu \in B} (P_{\mu\nu}^{\text{tran}})^2}$$

where μ and ν denote the basis functions centered at atom A and on B , respectively. Both original form and symmetrized form of TDM could be employed here.

If way 1 is employed, the \mathbf{p} will correspond to the matrix of so-called correlated electron-hole probability diagram (CEHPD), its (A,B) element was interpreted as the probability of simultaneously

finding a hole in atom A and an electron in atom B (this interpretation is not strictly true in general cases). See *J. Chem. Phys.*, **113**, 10002 (2000) and *J. Am. Chem. Soc.*, **129**, 14257 (2007) for example, in which the authors used $\bar{P}_{\mu\nu}^{\text{tran}}$ obtained at ZINDO level.

If the \mathbf{p} are constructed in way 2, 3 or 4, the resulting matrix may be referred to as atom transition density matrix. For example, the way 4 has been employed in *Chem. Rev.*, **102**, 3171 (2002). However, according my experiences, using way 2 or 3 is more preferred, since I found that the diagonal terms obtained in way 4 is often too large compared to the off-diagonal terms.

Assume that the TDM used to construct \mathbf{p} was not symmetrized, the general structure of the resulting \mathbf{p} could be expressed in below form

$$\mathbf{p} \equiv \begin{matrix} \text{electron} \\ \vdots \\ 1,2 \\ 1,1 \end{matrix} \begin{bmatrix} 1,N & 2,N & \cdots & N,N \\ \vdots & \vdots & \ddots & \vdots \\ 2,2 & \cdots & N,2 \\ 2,1 & \cdots & N,1 \end{bmatrix} \begin{matrix} \text{hole} \\ \vdots \\ N,2 \\ N,1 \end{matrix}$$

In complete analogy with the discussion about TDM, the physical meaning of the matrix elements of \mathbf{p} can be roughly understood as follows, irrespective of the choice of the specific way of constructing the \mathbf{p} :

- Diagonal terms: If (A,A) is large, it implies that atom A has large contribution to both hole and electron, therefore the electron excitation should result in evident charge reorganization within atom A
- Off-diagonal terms: If (A,B) is large, then atom A should have large contribution to hole and meantime atom B should have large contribution to electron, implying that electron excitation leads to CT from A to B

The "hole" and "electron" mentioned above are highly abstract concepts, although they have the same physical meaning as the those defined in the hole-electron analysis (Section 3.21.1), one cannot expect that the pattern of the \mathbf{p} defined in any one of above ways is always very close to the atom-atom charge transfer matrix, which is much more strictly defined and more meaningful.

If symmetrized form of TDM was used to build \mathbf{p} , then CT directional information will not be reflected by \mathbf{p} . In this case, if off-diagonal term $(A,B)=(B,A)$ is large, then we can simply say that coherence between atoms A and B is strong during the electron excitation, in other words, charge transfer occurs between atoms A and B .

The heat map of \mathbf{p} is particularly useful for analyzing large-size and highly conjugated molecules. Commonly hydrogens are omitted in the plot to make the map compact, since hydrogens rarely participate in electron excitation of chemical interest.

If fragments are defined, the \mathbf{p} (or other kinds of atom transition matrix) can further be contracted to fragment based form:

$$p_{RS} = \sum_{A \in \text{frag } R} \sum_{B \in \text{frag } S} p_{AB}$$

This form is very convenient when one wishes to study role of various fragments in electron excitation.

Input files

Since there are different types of atom transition matrix, and the matrix can be passed to

Multiwfn in different ways, there are several circumstances as shown below, you should use proper input files. The file that should be loaded when Multiwfn boots up is always the file containing basis function information, and it should corresponds to another file that needed to be loaded when you enter present function.

(1) Plotting heat map of **p** in usual way

You should load a file containing configuration coefficient information of excited states when you enter this function (see beginning of Section 3.21). Then Multiwfn will automatically generate TDM between ground state and you selected excited state, and at the same time you can choose if symmetrizing the resulting TDM in aforementioned way.

(2) Plotting heat map of **p** based on the TDM recorded in Gaussian output file

You should load Gaussian output file of electron excitation task when you enter this function. The keywords *density=transition=x IOp(6/8=3)* must be specified in Gaussian input file, so that TDM between ground state and excited state *x* can be printed in output file by Link 601 of Gaussian. Via this way, not only the TDM of CIS/TDHF/TDA-DFT/TDDFT can be plotted, but also the TDM generated by the EOM-CCSD and semi-empirical ZINDO method can be plotted.

Note 1: The TDM outputted by Gaussian is in aforementioned symmetrized form.

Note 2: If your ground state is singlet state while you used such as *TD=triplet* to request Gaussian to compute triplet excited states, then the outputted TDM will be exactly zero due to spin forbidden, and thus Multiwfn is unable to plot corresponding TDM map. However, it is possible to draw spatial part of the singlet-triplet TDM. To do this, you should let Multiwfn itself to generate TDM, see (1).

Note 3: If the basis set you used contains diffuse basis functions, in rare cases, the TDM outputted by Gaussian is incorrect, and thus the resulting heat map will be useless.

In summary, if the method you are using is not ZINDO, do not let Multiwfn to load TDM directly from Gaussian output file.

(3) Plotting heat map of **p** based on the TDM recorded in a plain text file

You should load a file named *tdmat.txt* when you enter this function, Multiwfn will read TDM from this file. Commonly, the *tdmat.txt* is generated by subfunction 9 of main function 18 (see Section 3.21.9 for detail), which can not only generate TDM between ground state and an excited state, but can also generate TDM between two excited states. An example file has been provided as *examples\excit\tdmat.txt*.

For above three cases, you can choose the way used to contract the TDM to the **p**.

(4) Plotting atom transition dipole moment matrix

You should load a file named one of *AAtrdip.txt*, *AAtrdipX.txt*, *AAtrdipY.txt*, *AAtrdipZ.txt* when you enter this function, Multiwfn will read atom transition dipole moment matrix from this file. Commonly, they are generated by subfunction 11 of main function 18 (see Section 3.21.11 for detail). By plotting heat map of these matrices, one can easily recognize which atoms and which interatomic couplings notably affect transition dipole moment.

(5) Plotting atom-atom charge transfer matrix

You should load a file named *atmCTmat.txt* when you enter this function, Multiwfn will read atom-atom charge transfer matrix from this file. Commonly, the *atmCTmat.txt* is generated by subfunction 8 of main function 18 (see Section 3.21.8 for detail). By plotting heat map of this kind of matrix, charge transfers between various atoms or fragments as well as charge reorganization sites can be intuitively recognized.

Hint: In fact, you can also make the *tdmat.txt* or *AAtrdip.txt/atmCTmat.txt* contain other kind of matrices so that they can be plotted as heat map via present module. For example, you can export bond order matrix as *bndmat.txt* using corresponding subfunction in main function 9, then rename it as *atmCTmat.txt* and delete the first line from it, then if you load this file into Multiwfn when entering present module, the plotted heat map will correspond to the bond order matrix.

Usage

After loading all needed files and generating all needed data as mentioned above, you will enter the interface for plotting heat map of the atom transition matrix (ATM). There are several options:

Option 0: Showing heat map of ATM on screen. By default, labels in abscissa and ordinate of this map correspond to indices of non-hydrogen atoms.

Option 1: The same as option 0, but save the heat map as graphical file in current folder.

Option 3: Exporting the ATM as *matrix.txt* in current folder, so that it can then be conveniently plotted by some third-party tools such as Origin and SigmaPlot.

Option 4: Switching the status that if hydrogens will be included in the heat map

Option 5: Changing upper and lower limits of color scale. By default they are automatically set to maximal and minimal elements of the ATM, respectively.

Option 6: Changing the number of interpolation steps between grid data. If you want to make the graph look smooth, it should be set to a large value (the default 10 is already quite large); if the value is set to 1, then interpolation will not be performed, in this case each square grid in the map exactly corresponds to a matrix element.

Option 7: Setting interval between labels in abscissa and ordinate.

Option 8: Determining if performing normalization. If the status is switched to "Yes", then normalization factor will be applied so that the sum of all elements of ATM is equal to unity.

If you select option "-1 Define fragments", fragment definition can be directly inputted or be loaded from a plain text file, which should look like below, each fragment occupies a line:

```
1,3,6-10,12  
2,4,5  
11  
13-15
```

Then Multiwfn will contract the atom transition matrix to fragment transition matrix (FTM). After that, the matrix to be plotted or exported in present module will be FTM instead of ATM

An example of plotting and studying **p** matrix as well as other kinds of atom transition matrices is given in Section 4.18.2.2.

3.21.3 Analyze charge-transfer based on density difference grid data (3)

Theory

In the paper *J. Chem. Theory Comput.*, 7, 2498 (2011), the authors proposed a method for analyzing charge-transfer (CT) during electron transition, present function fully implements this analysis method. It is also probable that this method can be used to study CT in other processes, such as formation of molecular complex. In the original paper, the author only discussed the cases when charge-transfer is in one-dimension, while in Multiwfn this scheme has been generalized to three-dimension case. In addition, some quantities introduced below are not proposed in the original paper but proposed by me, definition of some quantities in the original paper have also been modified by me to make the analysis more meaningful.

The electron density variation between excited state (EX) and ground state (GS) is

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

Notice that the geometry used in calculating ρ_{EX} and ρ_{GS} must be identical, otherwise the resulting $\Delta\rho$ will be meaningless. Therefore, present function can only be used to characterize "vertical" process. $\Delta\rho$ can be divided into positive and negative parts, namely ρ_+ and ρ_- . Of course, the integral of ρ_+ and $-\rho_-$ over the whole space should be equal. If evident inequality is observed, that means the error in numerical integral is unneglectable, and higher quality of grid (*i.e.* larger number of grid points) is required. Even though what you analyzed is single-electron excitation, the magnitude of ρ_+ and ρ_- as well as their integrals over the whole space in principle can also be larger than 1.0, this is because excitation of an electron must lead to reorganization of distribution of the rest of electrons, which also make contribution to $\Delta\rho$.

The transferred charge q_{CT} is the magnitude of the integral of ρ_+ and ρ_- over the whole space. It is important to correctly recognize the physical meaning of this quantity. q_{CT} only corresponds to the total amount of charge whose distribution is perturbed during electron excitation, it does not correspond to net charge transfer from one fragment to another fragment (*e.g.* from donor group to acceptor group)

The barycenter of positive and negative parts of $\Delta\rho$ 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 Cartesian component coordinates of \mathbf{R}_+ will be referred to as X_+, Y_+, Z_+ below, while that of \mathbf{R}_- will be referred to as X_-, Y_-, Z_- .

The distance between the two barycenters measures the CT length, its three Cartesian components:

$$D_x = |X_+ - X_-| \quad D_y = |Y_+ - Y_-| \quad D_z = |Z_+ - Z_-|$$

The D index is defined as $\sqrt{(D_x)^2 + (D_y)^2 + (D_z)^2} \equiv |\mathbf{R}_+ - \mathbf{R}_-|$, which characterizes total CT length.

The dipole moment variation caused by electron excitation can be evaluated as

$$\Delta\mu_x = (X_+ - X_-)q_{\text{CT}} \quad \Delta\mu_y = (Y_+ - Y_-)q_{\text{CT}} \quad \Delta\mu_z = (Z_+ - Z_-)q_{\text{CT}}$$

The RMSDs of distribution of ρ_+ and ρ_- 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 Cartesian components of position vector \mathbf{r} . For example, $\sigma_{+,y}$ can be explicitly written as

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

The difference between RMSD of ρ_+ and ρ_- in X/Y/Z direction can be measured via $\Delta\sigma_\lambda$, while overall difference can be measured via the $\Delta\sigma$ index

$$\begin{aligned}\Delta\sigma_\lambda &= \sigma_{+,\lambda} - \sigma_{-,\lambda} \quad \lambda = \{x, y, z\} \\ \Delta\sigma \text{ index} &= |\boldsymbol{\sigma}_+| - |\boldsymbol{\sigma}_-|\end{aligned}$$

It is noteworthy that D index is zero for exactly centrosymmetric systems, therefore, it is useless for discussing CT problem of such kind of system. However, $\Delta\sigma$ index is often useful to identify this type of excitation, since in this case $\Delta\sigma$ index must be large because diffuseness extent of ρ_+ is much higher than ρ_- .

C_+ and C_- functions are defined aiming for visualizing CT more intuitively than $\Delta\rho$. Their structures are similar to Gaussian function, the value asymptotically approaches zero from the centroid of the function.

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

The normalization factor A is introduced so that the integrals of C_+ and C_- over the whole space are equal to that of ρ_+ and ρ_- , respectively.

H_λ measures average degree of spatial extension of ρ_- and ρ_+ in X/Y/Z direction, H_{CT} is that in CT direction, and H index is an overall measure:

$$\begin{aligned}H_\lambda &= (\sigma_{+,\lambda} + \sigma_{-,\lambda})/2 \quad \lambda = \{x, y, z\} \\ H_{CT} &= |\mathbf{H} \cdot \mathbf{u}_{CT}| \\ H \text{ index} &= (|\boldsymbol{\sigma}_+| + |\boldsymbol{\sigma}_-|)/2\end{aligned}$$

where \mathbf{u}_{CT} is unit vector in CT direction and can be straightforwardly derived using centroid of ρ_- and ρ_+ .

t index measures separation degree of ρ_+ and ρ_- :

$$t \text{ index} = D \text{ index} - H_{CT}$$

If t index < 0 , it implies that ρ_- and ρ_+ are not substantially separated due to CT. Clear separation of ρ_- and ρ_+ distributions must correspond to evidently positive t index.

I defined another quantity to measure overlapping extent between C_+ and C_- :

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

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. This index is dimensionless.

Usage

Because all numerical integrals mentioned above are computed based on evenly distributed grid data, user needs to generate grid data of $\Delta\rho$ by using custom operation of main function 5, see Section 3.7.1, or load a file (e.g. cube file) containing grid data of density difference when Multiwfn

boots up. After that, enter subfunction 3 of main function 18, all aforementioned quantities will be shown on screen immediately. The "Overlap integral between C+ and C-" term is the S_{+-} introduced above. In the post-process menu, user can choose to visualize C_+ and C_- , or export grid data for the two functions to cube file in current folder.

An example is given in Section 4.18.3.

Information needed: Grid data of electron density difference

3.21.4 Calculate Δr index to measure charge-transfer length (4)

Theory

In the paper *J. Chem. Theory Comput.*, **9**, 3118 (2013), Δr index was proposed to measure CT length during electron excitation. The Δr can be expressed as

$$\Delta r = \sum_{i,a} \Delta r_i^a$$

where Δr_i^a is contribution of orbital transition between i and a to the Δr index:

$$\Delta r_i^a = \frac{(K_i^a)^2}{\sum_{i,a} (K_i^a)^2} |\langle \varphi_a | \mathbf{r} | \varphi_a \rangle - \langle \varphi_i | \mathbf{r} | \varphi_i \rangle|$$

The index i and a run over all occupied and virtual MOs, respectively. φ is orbital wavefunction. Assume that the method you used to calculate electron excitation is CIS or the TDDFT under Tamm-Danoff approximation, then K_i^a is simply the configuration coefficient corresponding to excitation of $i \rightarrow a$. While if the method you used is TDHF or TDDFT, then $K_i^a = w_i^a + w_i'^a$, where w_i^a and $w_i'^a$ denote the configuration coefficient corresponding to excitation of $i \rightarrow a$ and de-excitation of $i \leftarrow a$, respectively.

Δr is especially useful for diagnosing when certain classes of DFT functionals are failure for TDDFT purpose. When Δr is large, pure functionals such as BLYP and PBE, and the hybrid functionals with low Hartree-Fock exchange composition such as B3LYP and PBE0, will not work well. In this case, long-range corrected functionals should be employed; for instance, CAM-B3LYP, LC- ω PBE and ω B97X.

It is worth to mention that if an electron excitation can be perfectly represented by one pair of MO transition, then the Δr index and D index defined in hole-electron analysis framework will be exactly identical in principle:

$$\begin{aligned} \Delta r &= |\langle \varphi_a | \mathbf{r} | \varphi_a \rangle - \langle \varphi_i | \mathbf{r} | \varphi_i \rangle| \equiv \left| \int \mathbf{r} |\varphi_a(\mathbf{r})|^2 d\mathbf{r} - \int \mathbf{r} |\varphi_i(\mathbf{r})|^2 d\mathbf{r} \right| \\ D \text{ index} &= |\mathbf{D}| = \left| \int \mathbf{r} \rho^{\text{ele}}(\mathbf{r}) d\mathbf{r} - \int \mathbf{r} \rho^{\text{hole}}(\mathbf{r}) d\mathbf{r} \right| = \left| \int \mathbf{r} |\varphi_a(\mathbf{r})|^2 d\mathbf{r} - \int \mathbf{r} |\varphi_i(\mathbf{r})|^2 d\mathbf{r} \right| \end{aligned}$$

However, their values outputted by Multiwfn should be marginally different, since they are

evaluated based on different numerical integration algorithms.

Usage

The input files needed by present module have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states.

After entering present function (subfunction 4 of main function 18), you will be prompted to select the excited states for which the Δr will be calculated, then the results will be printed on screen immediately.

If you only selected one state, then Multiwfn will ask you to choose if decomposing the Δr into orbital pair contributions. If you inputted *e.g.* 0.01, then orbital pairs which have contribution to Δr larger than 0.01 will be printed. From the output, you can easily identify which orbital pairs have significant contribution to charge-transfer of electron excitation.

An example of present function is provided as Section 4.18.4.

Information needed: See beginning of Section 3.21.

3.21.5 Calculate transition electric dipole moments between all states and electric dipole moment of each state (5)

This function is used to calculate transition electric dipole moment between all states, including the ones between ground state and excited states, as well as the ones between excited states. This function is also able to print electric dipole moment for each state. For two states i and j , the transition electric dipole moment is defined as $\langle \psi_i | -\mathbf{r} | \psi_j \rangle$; when $i=j$, this quantity corresponds to electric dipole moment of this state that contributed by electrons.

The input files needed by present function have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states. If you need very accurate transition dipole moments, you should use *IOp(9/40=5)* keyword of Gaussian or *TPrint 1E-10* keyword of ORCA to make the program print as much configuration coefficients as possible.

After you enter present function, summary of all excitations will be printed. "Normalization" should be as close as possible to expected value (0.5 and 1.0 for closed- and open-shell reference state, respectively). If the deviation is large, then the resulting transition dipole moments must have large error, and you must make your quantum chemistry program output more configuration coefficients.

Then you will be prompted to select a task, there are four choices:

- (1) Output transition electric dipole moments between all states (including both ground state and excited states) to screen
- (2) The same as (1), but output to *transdipmom.txt* in current folder.
- (3) Generate input file of SOS module of Multiwfn as *SOS.txt* in current folder. Then if you use the *SOS.txt* as input file, you can use SOS module to evaluate (hyper)polarizability, see Section

3.200.8 for detail.

(4) Calculate and output electric dipole moment of each excited state to *ESdipmom.txt* in current folder. Note that both electronic and nuclear contribution to the value are taken into account.

The calculation process of all the four tasks consists of three stages:

Stage 1: Calculate dipole moment integrals between all basis functions

Stage 2: Calculate dipole moment integrals between all MOs

Stage 3: Calculate dipole moment integrals between all excited states.

If the output file of quantum chemistry program includes both singlet and triplet excited states, for example, you used *TD(50-50)* keyword in Gaussian and reference state is closed-shell, only aforementioned tasks (1) and (2) are available, and transition dipole moment of all singlet-singlet pairs (including ground state) and triplet-triplet pairs will be calculated by Multiwfn and outputted separately, while singlet-triplet pairs are ignored because due to spin-forbidden the result must be zero. In addition, excitation energies between S0 and all excited states are printed at the end of output. This function is of great importance if you want to use PySOC code to calculate spin-orbit coupling matrix element, see my blog article "Using Gaussian+PySOC to calculate spin-orbit coupling matrix element under TDDFT" (<http://sobereva.com/411>, in Chinese) for detail.

Below is a simple example of calculating dipole moment of all states. The corresponding Gaussian input file for generating the .fch and .out files is *examples\excit\4-nitroaniline.gjf*. Boot up Multiwfn and input

```
examples\excit\4-nitroaniline.fch
18 // Electronic excitation analysis
5 // Present function
examples\excit\4-nitroaniline.out
4 // Obtain dipole moment of each state
```

Then you will have *ESdipmom.txt* in current folder, the content is:

```
Ground state dipole moment in x,y,z: 0.326322 -2.792165 -0.000000 a.u.
```

Excited state dipole moments (a.u.):

State	X	Y	Z	exc.(eV)	exc.(nm)
1	0.334929	-1.219854	-0.000000	4.0557	305.92
2	0.251666	-7.797482	-0.000000	4.2762	290.14
3	0.334065	-1.439663	-0.000000	4.5846	270.62

Information needed: See beginning of Section 3.21.

3.21.6 Generate natural transition orbitals (NTOs) (6)

Theory

This function is used to generate natural transition orbitals (NTOs). NTO was proposed in *J. Chem. Phys.*, **118**, 4775 (2003), it has become a very popular and useful way to analyze character electron excitation obtained by single-reference methods.

Transition of electronic state is often not predominated by only one MO pair, in many cases

multiple MO pair transitions simultaneously have non-negligible contributions, which can be evaluated as square of corresponding configuration coefficient. This fact brings great hindrance of analyzing electron excitation character by simply visualizing related MOs. The NTO method aims to relieve this difficulty, it separately performs unitary transformation for occupied MOs and virtual MOs, so that only one or very few number of orbital pairs have dominant contributions.

The basic procedure of yielding NTOs is outlined below:

(1) Generating transition density matrix in MO basis (\mathbf{T}). Assume that the system has n_{occ} occupied MOs and n_{vir} virtual MOs, then \mathbf{T} has dimension of $(n_{\text{occ}}, n_{\text{vir}})$, its (i,l) element is simply constructed as

$$T_{i,l} = w_i^a$$

where $i < n_{\text{occ}}$, $l < n_{\text{vir}}$ and $a = l + n_{\text{occ}}$. w_i^a stands for configuration coefficient corresponding to $i \rightarrow a$ orbital transition. Note that for TD formalism, there may be some de-excitations, their configuration coefficients are simply ignored in constructing the \mathbf{T} .

(2) Generating temporary matrix for occupied and virtual orbitals, respectively

$$\mathbf{T}_{\text{occ}} = \mathbf{T}\mathbf{T}^T \quad \mathbf{T}_{\text{vir}} = \mathbf{T}^T\mathbf{T}$$

Evidently, both \mathbf{T}_{occ} and \mathbf{T}_{vir} are square matrices, their dimension are n_{occ} and n_{vir} , respectively.

(3) Diagonalizing \mathbf{T}_{occ} and \mathbf{T}_{vir} to obtain eigenvalues and eigenvectors

$$\mathbf{U}_{\text{occ}}^{-1} \mathbf{T}_{\text{occ}} \mathbf{U}_{\text{occ}} = \boldsymbol{\Lambda}_{\text{occ}} \quad \mathbf{U}_{\text{vir}}^{-1} \mathbf{T}_{\text{vir}} \mathbf{U}_{\text{vir}} = \boldsymbol{\Lambda}_{\text{vir}}$$

(4) The diagonal terms of $\boldsymbol{\Lambda}_{\text{occ}}$ and $\boldsymbol{\Lambda}_{\text{vir}}$ are eigenvalues of occupied and virtual NTOs, respectively. For the former, the eigenvalues are commonly sorted from low to high, while for the latter, the eigenvalues are commonly sorted from high to low. A NTO pair consists of an occupied NTO and a virtual NTO sharing the same eigenvalue. Eigenvalue of a NTO pair multiplied by 100 is just its percentage contribution to the electron excitation.

For CIS and TDA-DFT, the range of eigenvalue must be 0.0~1.0. However, in the TDHF and TDDFT cases, due to presence of de-excitations, which is not explicitly considered in the NTO analysis, it is possible that a NTO pair has eigenvalue slightly larger than 1.0, in this situation you can simply treat it as 1.0 (However, if the value is much larger than 1.0, the TDDFT result may be unreliable, and I suggest you use TDA-DFT instead).

(5) MOs are transformed to NTOs via unitary transformation matrix \mathbf{U}

$$\mathbf{C}_{\text{occ}}^{\text{NTO}} = \mathbf{C}_{\text{occ}}^{\text{MO}} \mathbf{U}_{\text{occ}} \quad \mathbf{C}_{\text{vir}}^{\text{NTO}} = \mathbf{C}_{\text{vir}}^{\text{MO}} \mathbf{U}_{\text{vir}}$$

where $\mathbf{C}_{\text{occ}}^{\text{MO}}$ and $\mathbf{C}_{\text{vir}}^{\text{MO}}$ are coefficient matrix of occupied MOs and virtual MOs in original basis functions, respectively; their columns correspond to different MOs. The counterpart matrices with NTO superscript denote coefficient matrix of occupied and virtual NTOs.

Implementation and Usage

The input files needed by present function have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states.

After you enter present function, you should input the index of the electron excitation to be studied, then Multiwfn will load corresponding configuration coefficients and generate NTOs according to the equations shown above, and then output eigenvalues of occupied and virtual NTOs. Next, you can choose whether exporting the NTOs to .fch or .molden file. If you choose to export, then you can use Multiwfn to load the newly generated .fch or .molden file to visualize the NTOs, analyze NTO orbital composition and so on (note that in this case the data in orbital energy field in fact is NTO eigenvalues).

Present function works for both restricted and unrestricted reference states; for the latter, the result of Alpha part and Beta part are calculated and printed separately, you only need to pay attention to the NTO pairs having largest eigenvalues (*e.g.* the largest eigenvalue of Alpha part is 0.03, while the largest eigenvalue of Beta part of 0.95, that means this electron excitation is dominated by transition of the Beta NTO pair)

According to my experiences, NTO analysis often works equally well as the hole-electron analysis introduced in Section 3.21.1, namely both of them are able to avoid necessity of inspecting many MOs when discussing electron excitation. An additional advantage of NTO analysis over hole-electron analysis is that the orbital phase information is retained; however, in some cases NTO analysis completely fails, namely even after transformation from MO to NTO representation, there is still no dominant orbital pair transition. Clearly in this case you have to resort to hole-electron analysis.

An example of generating and analyzing NTOs is provided in Section 4.18.6.

Information needed: See beginning of Section 3.21

3.21.7 Calculate ghost-hunter index (7)

In *J. Comput. Chem.*, **38**, 2151 (2017), Adamo *et al.* proposed an index used to diagnose if an excited state yielded by TDDFT calculation is a ghost state. The ghost states are artificial low-lying CT states resulting from evidently incorrect asymptotic behavior of exchange potential of pure DFT functionals or the hybrid functionals with low Hartree-Fock exchange composition. Since the ghost states are unreal states, they should be ignored when discussing electron excitation problems. If there are many ghost states which significantly polluted electronic spectrum, then DFT functionals with relatively high HF exchange composition (*e.g.* M06-2X and BH&HLYP) or long-range corrected functional (*e.g.* ωB97X and CAM-B3LYP) must be used to get rid of them. The ghost-hunter index (M_{AC}) aims to present a quantitative criterion to judge if an excited state should be regarded as a ghost state.

The ghost-hunter index in its original paper is defined as

$$M_{AC} = \sum_{i,a} \frac{w_i^a (-\varepsilon_i - \varepsilon_a)}{\sum_{i,a} w_i^a} - \frac{1}{D_{CT}}$$

where i and a denote occupied and virtual MOs, respectively. The summation loops all TDDFT configurations, the w is configuration coefficient. ε stands for MO energy. D_{CT} is charge-transfer index (see below). M_{AC} is essentially an approximate estimation of lower limit of CT excitation energy.

In my opinion, above definition of M_{AC} is somewhat peculiar, since it employs $w_i^a / \sum_{i,a} w_i^a$ as weight of $i \rightarrow a$ transition rather than $(w_i^a)^2 / \sum_{i,a} (w_i^a)^2$ as expected. In addition, it does not consider excitation and de-excitation separately (*i.e.* the summation simply loops all configurations irrespective of excitation and de-excitation). Taking these problems into account, I defined an another form of M_{AC} :

$$M_{AC} = \sum_{i \rightarrow a} \frac{(w_i^a)^2 (-\varepsilon_i - \varepsilon_a)}{\sum_{i \rightarrow a} (w_i^a)^2} - \frac{1}{D_{CT}}$$

In which all de-excitations are ignored in the summation.

In the original paper, it is argued that if calculated excitation energy is lower than M_{AC} , then the excited state should be regarded as ghost state, and the result should be dealt with caution. However, according to my personal experience, *this criterion is often too strong*.

In Multiwfn, the M_{AC} is not calculated in subfunction 7 of main function 18. To calculate it, you should perform usual hole-electron analysis (see introduction in Section 3.21.1 and example in Section 4.18.1). Once calculation of grid data of hole and electron is finished, Multiwfn automatically prints the M_{AC} index as well as its two terms (their difference corresponds to the M_{AC} index). The M_{AC} index is calculated and shown in two forms, the definition 1 corresponds to the formula in original paper, while the definition 2 corresponds to the formula defined by me, as shown above.

Beware that in Multiwfn, the D_{CT} used in the M_{AC} expression is evaluated as distance between centroid of electron and hole distributions, it is more or less different to the D_{CT} used in M_{AC} original paper (referred to as D_{CT}' below), which is calculated as centroid distance between positive and negative parts of density difference between relaxed excited state density and ground state density. The D_{CT} used by Multiwfn is not only reasonable enough, but also much cheaper than D_{CT}' (since evaluating TDDFT relaxed density for large system is fairly expensive). However, if you really want to calculate M_{AC} index based on D_{CT}' , you should get the first term of M_{AC} via electron-hole module, and then calculate the second term (*i.e.* D_{CT}') via subfunction 3 of main function 18 (see the example given in Section 4.18.3), and then manually get their difference.

It is worth to mention that the M_{AC} index is only applicable to one-dimension CT problem, if the CT takes place in multiple directions, then this index is incapable and must be properly generalized.

Calculation of M_{AC} is involved in the example in Section 4.18.1.

Information needed: See beginning of Section 3.21

3.21.8 Calculate interfragment charge transfer in electron excitation via IFCT method (8)

Theory

Interfragment charge transfer is a very important phenomenon in electron excitation process. I

devised an albeit simple but quite useful way of evaluating amount of interfragment charge transfer between any number of fragments, the method is described below (to be published publicly). This method will be referred to as IFCT (InterFragment Charge Transfer).

The IFCT method contains three steps:

- (1) Calculating atomic contribution to hole and electron (see introduction of the concept of hole and electron in Section 3.21.1.1)
- (2) Calculating fragment contributions to hole and electron by summing up atomic contributions
- (3) Constructing interfragment charge transfer matrix \mathbf{Q} . Its (R,S) element corresponds to the electron transfer from fragment R to fragment S during the excitation:

$$Q_{R,S} = \Theta_{R,\text{hole}} \Theta_{S,\text{ele}}$$

where $\Theta_{R,\text{hole}}$ and $\Theta_{S,\text{ele}}$ denote contribution of fragment R to hole and contribution of fragment S to electron, respectively. Above formula is very easy to comprehend, it essentially assumes that electron transfer from R to S is proportional to both composition of R in hole (where electron leaves) and composition of S in electron (where electron goes).

Then three additional useful quantities could be defined:

- Electron net transferred from fragments S to R : $p_{S \rightarrow R} = Q_{S,R} - Q_{R,S}$
- Variation of electron population of fragment R : $\Delta p_R = \sum_{S \neq R} p_{S \rightarrow R} = \sum_{S \neq R} (Q_{S,R} - Q_{R,S})$
- Intrafragment electron redistribution of fragment R : $Q_{R,R}$

By the way, it is easy to show that variation of electron population of a fragment evaluated in above way is quite reasonable:

$$\begin{aligned} \Delta p_R &= \sum_{S \neq R} (Q_{S,R} - Q_{R,S}) = \sum_{S \neq R} (\Theta_{S,\text{hole}} \Theta_{R,\text{ele}} - \Theta_{R,\text{hole}} \Theta_{S,\text{ele}}) \\ &= \Theta_{R,\text{ele}} \sum_{S \neq R} \Theta_{S,\text{hole}} - \Theta_{R,\text{hole}} \sum_{S \neq R} \Theta_{S,\text{ele}} \\ &= \Theta_{R,\text{ele}} (1 - \Theta_{R,\text{hole}}) - \Theta_{R,\text{hole}} (1 - \Theta_{R,\text{ele}}) \\ &= \Theta_{R,\text{ele}} - \Theta_{R,\text{hole}} \end{aligned}$$

It is easy to comprehend that this is a quite reasonable way of evaluating variation of electron population of fragment R , and thus well demonstrated reasonableness of the interfragment charge analysis formalism introduced above.

In addition, it is worth to note that sum of amount of interfragment transferred electrons and amount of intrafragment redistribution electrons exactly equals to unity, reflecting the fact that only one electron is excited:

$$\begin{aligned} & \sum_R \sum_{S \neq R} Q_{R,S} + \sum_S Q_{S,S} \\ &= \sum_R \sum_S Q_{R,S} \\ &= \sum_R \sum_S \Theta_{R,\text{hole}} \Theta_{S,\text{ele}} \\ &= \sum_R \Theta_{R,\text{hole}} \sum_S \Theta_{S,\text{ele}} \\ &= 1 \end{aligned}$$

Usage

The input files needed by the IFCT analysis have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states.

After entering this function, you need to choose the excited state to be studied, then input the total number of fragments, after that you should define each fragment in turn by inputting atomic indices. If you prefer to load fragment definition from a plain text file, you can input *0* when Multiwfn let you set the total number of fragments, then you can input the path of the file containing fragment defintion, the file format should look like below, definition of each fragment occupies a line:

```
1,3,6-10,12  
2,4,5  
11  
13-15
```

If in this step you input *-1*, Multiwfn will not carry out regular IFCT analysis but export a file named *atmCTmat.txt* in current folder, this file records atom-atom charge transfer matrix, whose element is defined as $Q_{A,B} = \Theta_{A,\text{hole}} \Theta_{B,\text{ele}}$. If you input path of this file after entering the function used to plot atom/fragment transition matrix (see Section 3.21.2), this matrix could be plotted as heat map so that you can visually study its matrix elements.

Once definition of fragments is completed, Multiwfn will calculate and print contribution of all defined fragments to hole and electron, as well as amount of electron transfer between fragments. In addition, net electron transfer as well as variation of electron population of each fragment are also printed. Below is an output instance:

```
Variation of population number of fragment 1: -0.25313  
Variation of population number of fragment 2: -0.23110  
Variation of population number of fragment 3: 0.48423  
  
Intrafragment electron redistribution of fragment 1: 0.00334  
Intrafragment electron redistribution of fragment 2: 0.31271  
Intrafragment electron redistribution of fragment 3: 0.02419  
  
Transferred electrons between fragments:  
1 -> 2: 0.11977 1 <- 2: 0.00874 Net 1 -> 2: 0.11103
```

1 -> 3: 0.14299	1 <- 3: 0.00089	Net 1 -> 3: 0.14210
2 -> 3: 0.36476	2 <- 3: 0.02263	Net 2 -> 3: 0.34213

By default, Mulliken-like method is automatically employed to calculate composition of hole and electron (see "Theory 2" of Section 3.21.1.1). This method is fast, however diffuse functions must not be used in this case, otherwise the result may be misleading. If you prefer to use the more robust Becke partition to evaluate composition of hole and electron, you should first use hole-analysis module to yield cube file of hole and electron, then if *hole.cub* and *electron.cub* can be found in current folder when entering IFCT module, Multiwfn will ask you if carrying out IFCT analysis for the hole and electron recorded in these two cube files based on Becke partition.

An example of IFCT analysis is given in Section 4.18.8.

Information needed: See beginning of Section 3.21

3.21.9 Generate and export transition density matrix (9)

According to MO expansion coefficients and configuration coefficients, transition density matrix (TDM) in basis function representation can be constructed by this function. Two kinds of TDMs can be generated (for simplicity, de-excitations in TD formalism are not shown but they are taken account into actual calculation):

(1) TDM between ground state and a selected excited state K :

$$P_{\mu\nu}^{\text{tran}} = \sum_i^{\text{occ}} \sum_a^{\text{vir}} w_K^{i \rightarrow a} C_{\mu i} C_{va}$$

where w corresponds to coefficient of the configurations involved in the excitation, $C_{\mu i}$ denotes the expansion coefficient of basis function μ in MO i .

(2) TDM between two selected excited states K and L :

$$P_{\mu\nu}^{KL} = \sum_i^{\text{occ}} \sum_a^{\text{vir}} w_K^{i \rightarrow a} \sum_j^{\text{occ}} \sum_b^{\text{vir}} w_L^{j \rightarrow b} V_{\mu\nu}^{iajb}$$

$$V_{\mu\nu}^{iajb} = \begin{cases} C_{\mu a} C_{vb} & (i = j, a \neq b) \\ -C_{\mu i} C_{vj} & (i \neq j, a = b) \\ P_{\mu\nu} - C_{\mu i} C_{vi} + C_{\mu a} C_{va} & (i = j, a = b) \\ 0 & (i \neq j, a \neq b) \end{cases}$$

where \mathbf{P} is density matrix of ground state (For TD case, the V between excitation and de-excitation configurations are simply ignored. In addition, when calculating V between de-excitation configurations, it is replaced with $-V$).

Once generation of TDM has been finished, you can choose if symmetrizing the TDM. There are two ways

- Way 1: $\bar{P}_{\mu\nu}^{\text{tran}} = (P_{\mu\nu}^{\text{tran}} + P_{\nu\mu}^{\text{tran}})/2$

- Way 2: $\bar{P}_{\mu\nu}^{\text{tran}} = (P_{\mu\nu}^{\text{tran}} + P_{\nu\mu}^{\text{tran}}) / \sqrt{2}$

The way 1 is reasonable and should be used in common case. However, it should be noted that the TDM generated by Gaussian program corresponds to the one symmetrized by way 2, therefore you should choose way 2 if you want the resulting TDM follows convention of Gaussian.

The generated matrix will be outputted to *tdmat.txt* in current folder. You can also choose to output *TDM.fch* in current folder, whose “Total SCF Density” field will correspond to TDM (this file is useful if you would like to calculate TrEsp type of atomic transition charges by making use of cubegen utility in Gaussian, see Section 4.A.9 for detail).

Note that when ground state and excited state have different spin multiplicities, due to the orthonormality of spin coordinate, although in principle the transition density matrix should be zero, the outputted matrix is not, because Multiwfn only takes spatial part of the MOs into account during constructing the matrix.

The input files needed by present function have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states.

The example in Section 4.18.2.3 utilized present function.

Information needed: See beginning of Section 3.21

3.21.10 Decompose transition dipole moment as molecular orbital pair contributions (10)

Oscillator strength (f) of an electron excitation directly relates to the integral area of the corresponding absorption peak. f has direct relationship with transition electric dipole moment \mathbf{D} (in atomic unit)

$$f = \frac{2}{3} \Delta E \times (D_x^2 + D_y^2 + D_z^2)$$

where ΔE denotes the transition energy between the two electronic states. Clearly, transition electric dipole moment is a crucial quantity of electron excitations.

The three Cartesian components of transition electric dipole moment between ground state and an excited state can be calculated as follows

$$D_x = -\sum_{i,a} w_i^a \langle \varphi_i | x | \varphi_a \rangle \quad D_y = -\sum_{i,a} w_i^a \langle \varphi_i | y | \varphi_a \rangle \quad D_z = -\sum_{i,a} w_i^a \langle \varphi_i | z | \varphi_a \rangle$$

where i and a loop over all occupied and virtual MOs, respectively. w is configuration coefficient (irrespective of excitation and de-excitation here), φ denotes molecular orbital, x/y/z stands for Cartesian components of position vector. It is clear that the transition dipole moment can be straightforwardly decomposed into contribution of various MO pairs. Via such a decomposition, one can easily study why some excitations have relatively large oscillator strength and thus have strong absorption, and why some excitations only have small oscillator strength and thus they are difficult to be observed in electronic spectrum.

The input files needed by present function have been detailedly described at the beginning of

Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states when you enter this function.

In this function, you will be prompted to choose the excited state for which the transition electric dipole moment will be decomposed as MO pairs, then a menu appears. You can select corresponding option to make Multiwfn output contribution of every MO pair to *transdip.txt* in current folder, or let Multiwfn sort the MO pairs according to their absolute contribution to certain component of transition electric dipole moment and then output the first few or dozens of terms, so that you can immediately capture the most important orbital pair transitions. In addition, you can ask the program to only output terms that contribute more than a given threshold.

Below is an output example, the terms are sorted according to absolute contribution to Y component of transition electric dipole moment and the first five terms are requested to be printed:

#Pair		Coefficient		Transition dipole x/Y/Z (a.u.)		
1082	36 ->	37	0.697700	-0.046386	-2.938793	-0.000000
1055	35 ->	38	0.096590	-0.001156	0.278865	-0.000000
2141	36 <-	37	-0.056630	0.003765	0.238532	0.000000
1083	36 ->	39	-0.047340	0.001138	0.065117	-0.000000
2118	35 <-	38	0.020870	-0.000250	0.060254	-0.000000

An example of this function is given in Section 4.18.10.

Information needed: See beginning of Section 3.21

3.21.11 Decompose transition dipole moment as basis function and atom contributions (11)

This function is used to decompose the total transition electric or magnetic dipole moment between ground state and a selected excited state into contributions from each basis function and atom. Therefore, you can infer which part of the system has significant impact on excitation properties such as oscillator strength. The result will be exported to *trdipcontri.txt*.

There are many possible ways to realize the decomposition. In this function Mulliken-like partition is employed due to its simplicity. For example, the contribution of basis function μ to Z component of transition electric dipole moment is evaluated as follows

$$D_z^{(\mu)} = P_{\mu\mu}^{\text{tran}} \langle \chi_\mu | -z | \chi_\mu \rangle + \sum_{\nu \neq \mu} [P_{\mu\nu}^{\text{tran}} \langle \chi_\mu | -z | \chi_\nu \rangle + P_{\nu\mu}^{\text{tran}} \langle \chi_\nu | -z | \chi_\mu \rangle] / 2$$

and the contribution of basis function μ to Z component of transition magnetic dipole moment

$$M_z^{(\mu)} = P_{\mu\mu}^{\text{tran}} \left\langle \chi_\mu \left| x \frac{\partial}{\partial y} - y \frac{\partial}{\partial x} \right| \chi_\mu \right\rangle + \sum_{\nu \neq \mu} \left(P_{\mu\nu}^{\text{tran}} \left\langle \chi_\mu \left| x \frac{\partial}{\partial y} - y \frac{\partial}{\partial x} \right| \chi_\nu \right\rangle + P_{\nu\mu}^{\text{tran}} \left\langle \chi_\nu \left| x \frac{\partial}{\partial y} - y \frac{\partial}{\partial x} \right| \chi_\mu \right\rangle \right) / 2$$

Then the contribution from an atom is simply the sum of the contribution from the basis functions belonging to it.

Since Mulliken partition is incompatible with diffuse functions, hence the decomposition result is unreliable if diffuse functions are presented in the basis set you used. In this case, the best way to study contribution from various atoms is visualizing the transition dipole moment density (see Section 3.21.1).

This function also asks you if outputting atom transition dipole moment matrix, if you choose y , then X, Y, Z components of the matrix will be exported to *AAtrdipX.txt*, *AAtrdipY.txt*, *AAtrdipZ.txt* in current folder, respectively, the matrix elements are defined as follows (I take atom-atom contribution matrix of transition electric dipole moment as example, the matrix for transition magnetic dipole moment is defined similarly and thus not explicitly shown here)

$$\begin{aligned} D_{A,B}^X &= \sum_{\mu \in A} \sum_{\nu \in B} P_{\mu\nu}^{\text{tran}} \langle \chi_\mu | -x | \chi_\nu \rangle \\ D_{A,B}^Y &= \sum_{\mu \in A} \sum_{\nu \in B} P_{\mu\nu}^{\text{tran}} \langle \chi_\mu | -y | \chi_\nu \rangle \\ D_{A,B}^Z &= \sum_{\mu \in A} \sum_{\nu \in B} P_{\mu\nu}^{\text{tran}} \langle \chi_\mu | -z | \chi_\nu \rangle \end{aligned}$$

For example, the term $D_{A,B}^X$ corresponds to joint contribution of A - B atom pair to X component of transition dipole moment, the sum of all elements of \mathbf{D}^X equals to X component of transition dipole moment of current system. Total transition dipole moment matrix (sum of square of X, Y, Z) is exported as *AAtrdip.txt* in current folder. All of these .txt files can be directly plotted as colored matrix map (heat map) by atom transition matrix plotting module (see Section 3.21.2 for detail).

The input files needed by present function have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states when you enter this function.

The example of Section 4.18.11 utilized this function.

Information needed: See beginning of Section 3.21

3.21.12 Calculate Mulliken atomic transition charges (12)

This function is used to calculate atomic transition charges, which is useful for studying Coulomb coupling between ground state and excited state (excitonic coupling) of two molecules, see e.g. *J. Phys. Chem. B*, **110**, 17268 (2006) and *Photosynth. Res.*, **111**, 47 (2012).

Transition population of a basis function μ derived by Mulliken method is

$$\Theta_\mu^{\text{tran}} = P_{\mu\mu}^{\text{tran}} + \sum_{\nu \neq \mu} S_{\mu\nu} (P_{\mu\nu}^{\text{tran}} + P_{\nu\mu}^{\text{tran}}) / 2$$

So, the Mulliken atomic transition charge of atom A should be $-\sum_{\mu \in A} \Theta_\mu^{\text{tran}}$. Sum of all atomic transition charges must be zero because the total number of electrons keeps unchanged during electron excitation.

The input files needed by present function have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn

boots up, and then load a file containing configuration coefficient information of excited states when you enter this function. After that, you should choose the excited state for which the Mulliken transition charges will be calculated. Then the result will be outputted to *atmtrchg.chg* file in current folder, the format of this kind of file has been introduced in Section 2.5, the last column of this file corresponds to the transition charges.

Below is an example of the calculation. Boot up Multiwfn and input

examples\excit\N-phenylpyrrole.fch

18 // Electron excitation analysis

12 // Calculate Mulliken transition charges

N-phenylpyrrole.out

3 // Study the transition from ground state to the third excited state

Then you will find *atmtrchg.chg* in current folder.

Note that in Multiwfn it is also possible to calculate the TrEsp (transition charge from electrostatic potential) introduced in *J. Phys. Chem. B*, **110**, 17268 (2006), which is derived by ESP fitting method based on transition density. See Section 4.A.9 on how to do this. For studying excitonic coupling purpose, TrEsp should work better than Mulliken atomic transition charge, but for large systems, cost of evaluating the former is significantly higher than the latter.

Information needed: See beginning of Section 3.21

3.21.13 Generate natural orbitals of specific excited states (13)

This function is used to generate natural orbitals (NOs) for a batch of selected excited states, and then export the NOs as .molden file. After that, if you want to perform wavefunction analysis for an excited state, you can simply load corresponding .molden file. Of course, you can also calculate *e.g.* density difference between two excited states using corresponding two .molden files via custom operation feature of main functions 3, 4 and 5. This function supports both closed-shell and open-shell reference state.

To use this function, you should load a file containing basis function information, and then load a file containing configuration coefficients when you enter this file, see beginning of Section 3.21 for detail. After that, you will be prompted to input the indices of the excited state for which NOs will be generated. For each selected excited state, the program will do below steps:

(1) Generating density matrix of excited state:

$$\mathbf{P}^{\text{ES}} = \mathbf{P}^{\text{GS}} + \sum_{i \rightarrow a} (w_i^a)^2 (-\mathbf{P}^i + \mathbf{P}^a) + \sum_{i \leftarrow a} (w_i^a)^2 (-\mathbf{P}^a + \mathbf{P}^i)$$

where \mathbf{P}^{GS} is density matrix of ground state wavefunction, The matrix like \mathbf{P}^i is density matrix constructed solely by orbital i , it can be evaluated as below, where \mathbf{C}_i is column vector of expansion coefficients of orbital i

$$\mathbf{P}^i = \mathbf{C}_i \mathbf{C}_i^T$$

Note that the density matrix constructed in this way corresponds to unrelaxed density.

(2) Diagonalizing the \mathbf{P}^{ES} to yield NOs. Each NO is an eigenvector of \mathbf{P}^{ES} , the accompanying

eigenvalue is occupation number of the NO.

(3) Exporting information of basis function and NOs to .molden file. If the excited state you selected is 2, then they will be exported as NO_00002.molden in current folder.

The example given in Section 4.18.13 fully utilized this function.

Information needed: See beginning of Section 3.21

3.21.14 Calculate Λ index to characterize electron excitation (14)

Theory

In the paper *J. Chem. Phys.*, **128**, 044118 (2008), Λ index was proposed to distinguish types of electron excitations. The form of the Λ index and the Δr index (see Section 3.21.4) is very similar. Λ index can be expressed as

$$\Lambda = \sum_{i,a} \Lambda_i^a$$

where Λ_i^a is contribution of MO transition between i and a to the Λ index:

$$\Lambda_i^a = \frac{(K_i^a)^2}{\sum_{i,a} (K_i^a)^2} \int |\varphi_i(\mathbf{r})| |\varphi_a(\mathbf{r})| d\mathbf{r}$$

All quantities involved in above expression are identical those in Δr index. The integral corresponds to overlap extent of the MO i and a , it is calculated numerically via Becke's multi-center grid-based integration approach. The default grid is a good compromise between cost and accuracy; if you want to change it, you can set "iautogrid" in settings.ini to 0 and then specify "radpot" and "sphpot" in settings.ini as your expected values. Note that calculation cost of the Λ index is by far higher than the Δr index, because of the above numerical integral step is expensive, especially for large systems.

The theoretical lower and upper limits of the Λ index are 0.0 and 1.0, respectively; the former (latter) corresponds to the case that hole and electron are completely separated (perfectly overlapped).

As the Δr index, the Λ index is useful for distinguishing type of electron excitations. Notice that their intrinsic characteristics are different, the Δr index is essentially an indicator of configuration weighted orbital separation distance, while Λ index reflects configuration weighted orbital overlapping extent. In some sense, the physical nature of Δr and Λ indices are similar to the D and S_r indices defined in hole-electron analysis framework, respectively (see Section 3.21.1.1), however I believe that the D and S_r indices are more reasonable, since their physical meanings are more clear and couplings between different configurations are fully taken into account. Therefore, without special reasons, using D and S_r indices is more recommended.

It is worth to mention that if an electron excitation can be perfectly represented by one pair of MO transition, then the Λ index and S_r index defined in hole-electron analysis framework will be exactly identical in principle:

$$\Lambda = \int |\varphi_i(\mathbf{r})| |\varphi_a(\mathbf{r})| d\mathbf{r}$$

$$S_r = \int \sqrt{\rho^{\text{hole}}(\mathbf{r}) \rho^{\text{ele}}(\mathbf{r})} d\mathbf{r} = \int \sqrt{|\varphi_i(\mathbf{r})|^2 |\varphi_a(\mathbf{r})|^2} d\mathbf{r} = \int |\varphi_i(\mathbf{r})| |\varphi_a(\mathbf{r})| d\mathbf{r}$$

However, their values outputted by Multiwfn should be marginally different, since they are evaluated based on different numerical integration algorithms.

Usage

The input files needed by present module have been detailedly described at the beginning of Section 3.21, namely you should load a file containing basis function information when Multiwfn boots up, and then load a file containing configuration coefficient information of excited states.

After entering present function (subfunction 14 of main function 18), the matrix containing overlap integral between norms of all occupied and unoccupied MOs will be evaluated first, then you will be prompted to select the excited states for which the Λ will be calculated, then the results will be printed on screen immediately.

If you only selected one state, then Multiwfn will ask you to choose if decomposing the Λ into orbital pair contributions. If you inputted *e.g.* 0.01, then orbital pairs which have contribution to Λ larger than 0.01 will be printed.

An example of present function is provided as Section 4.18.4.

Information needed: See beginning of Section 3.21.

3.22 Orbital localization analysis (19)

Theory and algorithm overview

Canonical molecular orbitals (CMOs) often show strong delocalization character and thus do not convey useful information about chemical bonding. There are many ways to localize MOs, the most popular ones are Foster-Boys (FB) localization, Edmiston–Ruedenberg (ER) localization and Pipek–Mezey (PM) localization. The NLMO method supported in NBO program is also a kind of orbital localization algorithm. The resulting orbitals from these methods are known as localized molecular orbitals (LMOs). Both the LMOs and CMOs are orthonormal sets and have identical dimension, they can be transformed to each other via unitary transformation.

The FB is the oldest orbital localization method, it was proposed in *Rev. Mod. Phys.*, **32**, 300 (1960). This method minimizes below quantity, so that the spatial distribution range of all orbitals become as small as possible

$$\langle \Omega \rangle_{\text{Boys}} = \sum_i \int \int \rho_i(\mathbf{r}_1)(\mathbf{r}_1 - \mathbf{r}_2)^2 \rho_i(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \quad \rho_i = |\varphi_i|^2$$

The FB method is very popular and widely used, so it is supported by Multiwfn.

The ER localization proposed in *Rev. Mod. Phys.*, **35**, 457 (1963) is also a well-known method, it localizes orbitals via maximizing below quantity (orbital self-repulsion integral)

$$\langle \Omega \rangle_{\text{ER}} = \sum_i \int \int \rho_i(\mathbf{r}_1) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \rho_i(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2$$

The ER method is highly deprecated, because it needs evaluation of two-electrons integral, which is very complicated; furthermore, transformation of the integral from AO basis to MO basis is very expensive. Although in a few papers some people argue that ER method has better physical meaning and the computational cost can be considerably reduced via introducing resolution-of-identity technique, I never think there is any convincing reason to employ ER method instead of FB, so ER method is not supported by Multiwfn.

The most popular orbital localization method is PM. The essence of PM localization is to maximize below quantity, so that distribution range of all orbitals can be shrunked as much as possible

$$P = \sum_i \sum_A (p_A^i)^2$$

In the original paper of PM method *J. Chem. Phys.*, **90**, 4916 (1989), p_A^i corresponds to Mulliken population of atom A in MO i . While in *J. Chem. Theory Comput.*, **10**, 642 (2014), it is shown that other population methods such as Löwdin, Hirshfeld, Becke, AIM can also be combinedly used with PM and obtaining reasonable result. Currently PM method based on Mulliken and Löwdin populations are supported in Multiwfn.

The maximization or minimization of above mentioned quantities can be done via Jacob sweep algorithm, which is employed in the original paper of the orbital localization methods and still be prevalently used until now. This method has lower efficiency than the later developed sophisticated methods such as unitary optimization and trust region; however, the Jacob sweep is very simple and works for most cases, in particular when only occupied orbitals are needed to be localized, therefore this algorithm is employed in Multiwfn.

The working equation of PM localization based on various population methods are largely identical, see Eq. 9 of *J. Comput. Chem.*, **14**, 736 (1993), they only differ in the definition of the term Q , which is needed to be computed in each iteration of Jacob sweep for every orbital pair. For Mulliken population, the term corresponding to orbitals i and j for atom A is

$$Q_A^{ij} = \frac{1}{2} \sum_{\mu \in A} \sum_{\nu} [C_{\nu i} C_{\mu j} + C_{\mu i} C_{\nu j}] S_{\mu \nu}$$

where μ and ν correspond to basis function index and the latter cycles all basis functions. While for Löwdin population, because the basis functions have been orthonormalized, the term Q is simplified as

$$Q_A^{ij} = \sum_{\mu \in A} C_{\mu i} C_{\mu j}$$

PM-Löwdin is seemingly much cheaper than PM-Mulliken; however, if properly programming, the cost of two methods are basically identical, because the Q for Mulliken case can be reformulated as

$$Q_A^{ij} = \frac{1}{2} \sum_{\mu \in A} [C_{\mu j} (\mathbf{SC})_{\mu i} + C_{\mu i} (\mathbf{SC})_{\mu j}]$$

If the **SC** matrix is calculated and stored in memory before Jacob sweep and updated frequently during iteration, the summation for index ν can be completely ignored. In fact, because symmetric

diagonalization step of Löwdin method is time-consuming for large system, the overall cost of PM-Mulliken is generally lower than PM-Löwdin, therefore PM-Mulliken is taken as the default localization method. Regarding the FB method, because it involves transformation of dipole moment integrals from AO to MO basis, which requires large amount of arithmetic operations, it can only be employed for much smaller system compared to PM-Mulliken and PM-Löwdin.

Usage

The input file must contain basis function information, thus you can use *e.g.* .fch, .molden and .gms as input file. This function only works for restricted and unrestricted SCF wavefunction.

Due to the robust and very low cost of PM-Mulliken method, it is chosen as the default orbital localization method in Multiwfn. If you want to change to PM-Mulliken or FB methods, use option -6.

In the interface, you can use option 1 to choose to localize occupied MOs only, or use option 2 to localize both occupied and unoccupied MOs (the two set of orbitals will be localized separately, *i.e.* no mixture between occupied and unoccupied orbitals is allowed). For unrestricted wavefunctions, the alpha and beta parts are treated separately.

Orbital localization is an iterative process, thus you should set criterion of convergence and maximum number of cycles. The default values are commonly appropriate and need not to be modified. The convergence status is printed during iteration, for all localizations methods, the

change of $P = \sum_i \sum_A (p_A^i)^2$ is used for judging convergence.

By default, once the localization is converged, orbital composition of all resulting LMOs will be calculated and major characters of the LMOs are printed. For occupied and virtual LMOs, the very cheap Mulliken and SCPA methods are employed, respectively (see Section 3.9 for detail). The reason that the two methods are used for different cases is that I found sometimes SCPA tends to overestimate delocalization character of lone pair type of LMO, while Mulliken method tends to result in extremely large atomic composition for virtual LMOs. Note that both the orbital composition analysis methods are unreliable when diffuse functions are presented. The default threshold for identifying can be changed by first choosing option -8 to disable

Finally, the LMOs are exported to *new.fch* in current folder, and then Multiwfn automatically loads it, after that you can analyze the localized orbitals in various ways; for example, plotting them as isosurfaces by main function 0 or performing orbital composition analysis by main function 8. If you do not want to let Multiwfn automatically load the newly generated *new.fch*, you can choose option -3 once to switch the status.

Hints on performing orbital localization analysis

In Multiwfn, the cost of PM-Löwdin and PM-Mulliken methods are proportional to $N_{\text{orb}}^2 N_{\text{bas}}$, while cost of FB is proportional to $N_{\text{orb}}^2 N_{\text{bas}}^2$, where N_{orb} and N_{bas} are the number of orbitals to be localized and the number of basis functions, respectively. Clearly, FB is much more expensive.

Convergence is usually more difficult for unoccupied orbitals than occupied orbitals, more difficult for large system than small system, and more difficult for FB than PM method.

If you do not have special reasons, **you only need to localize occupied orbitals**, since only occupied orbitals carry interesting information about electronic structure. Localization of unoccupied orbitals is much more time-consuming than localizing occupied orbitals, since the number of unoccupied orbitals is often very higher when extended basis set is used. Commonly, via

PM-Mulliken/Löwdin method, occupied orbitals can be easily localized for a system containing up to 200 atoms with medium-sized basis set. While for FB, in general this work can only be realized for a system containing up to 100 atoms.

If you want to decrease the cost of localization of occupied orbitals, you can choose "-5 If also localizing core orbitals" once to switch the status from the default "Yes" to "No". Commonly, mix between valence orbitals and core orbitals is rather weak, therefore ignoring inner-core orbitals during localization of valence orbitals is safe. However, in rare cases, ignoring the inner-core orbitals may cause difficulty in convergence.

PM method is not parallelized in Multiwfn, because I found that parallelization does not improve its speed evidently but sometimes make convergence more difficult. FB method is fully parallelized, thus using multi-cores CPU will reduce computational cost considerably.

Basis set of 2-Zeta with polarization functions quality (e.g. 6-31G*, def2-SVP) is completely adequate for orbital localization analysis, using larger quality basis set never leads to detectably improved result. It should be noticed that both **PM-Löwdin** and **PM-Mulliken methods can not be used when diffuse functions are presented**, because Löwdin and Mulliken populations are meaningless in this case. In contrast, FB localization is fully compatible with diffuse functions, though it is evidently more expensive. When the basis set does not contain diffuse functions, generally the resulting orbitals of all the above mentioned three methods are similar (except that FB orbitals do not preserve σ - π separation character as PM orbitals).

PS: It is noteworthy that if PM method is used in combination with such as Becke and Hirshfeld populations, the PM method will become more insensitive to basis set and can be used safely with diffuse functions. However, this combination is even much more expensive than FB method and thus does not show practical advantage, so this idea is not implemented into Multiwfn.

Special topic 1: Evaluating LMO energies

If you want to obtain energy of the localized orbitals, you need to choose option "-4 If calculating and print orbital energies", and input path of a file containing Fock (or Kohn-Sham, similarly hereinafter) matrix in original basis functions. The matrix elements should be provided in lower-triangular form, namely in this sequence: $F(1,1)$ $F(2,1)$ $F(2,2)$ $F(3,1)$ $F(3,2)$ $F(3,3)$... $F(nbasis,nbasis)$, where $nbasis$ is the total number of basis functions, the format is free. The Fock matrix may be obtained from output of some quantum chemistry codes (In fact, you can ask Gaussian to produce NBO .47 file, Multiwfn is able to automatically locate and read the \$FOCK field when the file name has .47 suffix). Once Fock matrix is loaded and transformed to localized orbital representation, the energy of localized orbitals will correspond to diagonal terms of Fock matrix. Specifically, Multiwfn performs below representation transform:

$$\mathbf{F}_{\text{LMO}} = \mathbf{C}^T \mathbf{F}_{\text{AO}} \mathbf{C}$$

where \mathbf{F}_{AO} is the Fock matrix in original basis functions that loaded from external file, $C(r,i)$ corresponds to coefficient of basis function r in localized orbital i . Energy of localized orbital j is simply $F_{\text{LMO}}(j,j)$.

Special topic 2: Revealing center of LMOs

To facilitate capturing basic distribution character of the generated LMOs, Multiwfn is able to calculate center position of LMOs and add them as Bq atom (ghost atom) into current system, so that you can use main function 0 to easily visualize them. The center of LMO is evaluated as follows

$$\mathbf{R}_i = \langle \varphi_i | \mathbf{r} | \varphi_i \rangle$$

where \mathbf{r} is coordinate vector.

To generate the LMO centers, you should choose “-8 If calculating center position and dipole moment of LMOs” once to switch its status to “Yes”. Then after generating LMOs, exporting .fch and reloading it, the center positions of the LMOs will be evaluated and added as Bq atoms. The coordinate of LMO centers as well as the correspondence between LMO indices and Bq indices will be outputted to *LMOcen.txt* in current folder, meantime the setting of main function 0 will be set to the best status for showing LMO centers (as illustrated in Section 4.19.1). Since the newly added Bq atoms do not have accompanying basis functions, the current wavefunction should not be subjected to wavefunction analyses, otherwise Multiwfn may crash or the result is completely meaningless.

Note that if there are multiple bonds and PM localization algorithm is used, the Bq atom corresponding to the center of σ -LMO and π -LMO of the same bond may overlay with each other. This can be avoided using FB algorithm instead, because FB represents multiple bond as multiple banana LMOs, whose center positions are evidently different with each other.

Special topic 3: Dipole moment analysis for occupied LMOs

Once “-8 If calculating center position and dipole moment of LMOs” has been switched to “Yes”, after performing orbital localization, you will be asked to choose if also performing dipole moment analysis for occupied LMOs. If you input *y*, then you will have *LModip.txt*, which contains dipole moment analysis result for all occupied LMOs. In order to make you correctly understand the output, below I describe the details.

The contribution of electron of an occupied LMO to dipole moment of the whole system is

$$\mathbf{D}_i = \langle \varphi_i | -\mathbf{r} | \varphi_i \rangle$$

This vector for all LMOs is outputted as “Contributions of all occupied LMOs to system dipole moment” in the *LModip.txt* file.

However, this quantity is unable to be directly used to measure polarity of a LMO. Given that $\mathbf{r} = (\mathbf{r} - \mathbf{r}_c) + \mathbf{r}_c$, where \mathbf{r}_c is a fixed point, the above quantity can be rewritten as follows

$$\mathbf{D}_i = \langle \varphi_i | -(\mathbf{r} - \mathbf{r}_c^i) | \varphi_i \rangle - \langle \varphi_i | \mathbf{r}_c^i | \varphi_i \rangle = -\langle \varphi_i | \mathbf{r} - \mathbf{r}_c^i | \varphi_i \rangle - \mathbf{r}_c^i \langle \varphi_i | \varphi_i \rangle = -\langle \varphi_i | \mathbf{r} - \mathbf{r}_c^i | \varphi_i \rangle - \mathbf{r}_c^i$$

we can defined a quantity \mathbf{d}_i , which measures dipole moment of the orbital with respect to the \mathbf{r}_c :

$$\mathbf{d}_i = -\langle \varphi_i | \mathbf{r} - \mathbf{r}_c^i | \varphi_i \rangle = \mathbf{D}_i + \mathbf{r}_c^i$$

If we properly choose the \mathbf{r}_c for an orbital, then the \mathbf{d}_i may be able to reflect polarity of the LMO.

For each LMO that identified as single-center one, the \mathbf{r}_c is automatically set to be the position of the atom having largest contribution to the LMO. Therefore, the \mathbf{d}_i represents deviation of centroid of the LMO electron distribution with respect to the nuclear position. These $\{\mathbf{d}\}$ are printed as “Single-center orbital dipole moments (a.u.)” in the *LModip.txt*.

For each LMO that identified as two-center one, assume that the two atoms with largest contributions are *A* and *B*, the \mathbf{r}_c is set to

$$\mathbf{r}_c = \mathbf{r}_A \frac{R_B}{R_A + R_B} + \mathbf{r}_B \frac{R_A}{R_A + R_B}$$

where \mathbf{r}_A and R_A are nuclear position and covalent radius of atom *A*, respectively. Similarly for atom *B*. The \mathbf{r}_c locates at center of the bonding region, therefore the \mathbf{d}_i , which exhibits deviation of

centroid of LMO electron distribution from \mathbf{r}_c , is capable of revealing the bond polarity. These $\{\mathbf{d}\}$ are printed as “Two-center orbital dipole moments (a.u.)” in the *LMOdip.txt*.

For closed-shell case, the printed \mathbf{d} are multiplied by a factor of 2, because the LMOs are doubly occupied. For open-shell case, the \mathbf{d} of alpha and beta LMOs are printed separately.

As byproducts, dipole moment of the whole system, as well as nuclear contribution and electronic contribution are also printed at the beginning of the *LMOdip.txt*. It is important to notice that even if there is no more delocalized LMOs, the sum of \mathbf{d} of all LMOs is generally unequal to the dipole moment of the whole system.

BTW: In fact, only when sum of the \mathbf{r}_c vectors in all \mathbf{d} equals to $-\sum_A Z_A \mathbf{r}_A$, namely exactly cancels the nuclear contribution to the system dipole moment, the sum of all \mathbf{d} will be equal to the system dipole moment. If one wants to satisfy this point, the $(\mathbf{r}_A + \mathbf{r}_B)/2$ should be employed as \mathbf{r}_c for all two-center LMOs, and all LMOs should just correspond to a Lewis structure of current system. Of course, these conditions are not met in present implementation of LMO analysis (but met in the “DIPOLe” analysis of NBO theory). Note that if $(\mathbf{r}_A + \mathbf{r}_B)/2$ is taken as \mathbf{r}_c and meantime you use the \mathbf{d} of two-center LMOs to measure bond polarity, ridiculous result will be obtained, for example you will find C-H is even much more polar than O-H!

Examples of orbital localization analysis are provided in Section 4.19. The examples of LOBA method (Section 4.8.4) and Section 4.100.22 also utilized present function.

Information needed: Atom coordinates, basis functions

3.23 Visual study of weak interaction (20)

Since propose of the NCI method in *J. Am. Soc. Chem.*, **132**, 6498 in 2010, visual study of weak interaction becomes increasingly popular, and numerous related analysis methods were put forward. Main function 22 of Multiwfn is a collection of these analyses methods.

3.23.1 Noncovalent interaction (NCI) anaylsis (1)

The noncovalent interaction (NCI) method, which is also known as reduced density gradient (RDG) method, is a very popular method for studying weak interaction, and realizing this analysis is a very important application of Multiwfn. The theory of NCI method has been detailed in *J. Am. Chem. Soc.*, **132**, 6498 (2010) and my blog article "Visual research of weak interaction by Multiwfn" (in Chinese, see <http://sobereva.com/68>). Some additional discussions and ways to improve plotting quality can be found in "Some key points and common problems of carrying out RDG analysis via Multiwfn+VMD" (in Chinese, see <http://sobereva.com/291>).

Use isosurface to represent weak interaction regions

How to visualize weak interaction? The first thing is to find a way to distinguish weak interaction region from other regions. From the table below we can know that if only the regions where the value of reduced density gradient (RDG) function is in the range of 0~medium are preserved, then “Around nuclei” and “Boundary of molecule” regions will be shielded.

	Around nuclei	Around chemical bond	Weak interaction region	Boundary 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

where the RDG function is defined as below, it is dimensionless form of electron density gradient norm

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

After that, for remaining regions ("Around chemical bond" and " Weak interaction region"), if we only pay attention to the region where $\rho(\mathbf{r})$ is small, then weak interaction region is revealed.

Now I use phenol dimer to exemplify this idea, we will calculate grid data of RDG function and visualize it as isosurface. Boot up Multiwfn and input following commands

examples\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 $40*40*40=64000$ grid is fine enough

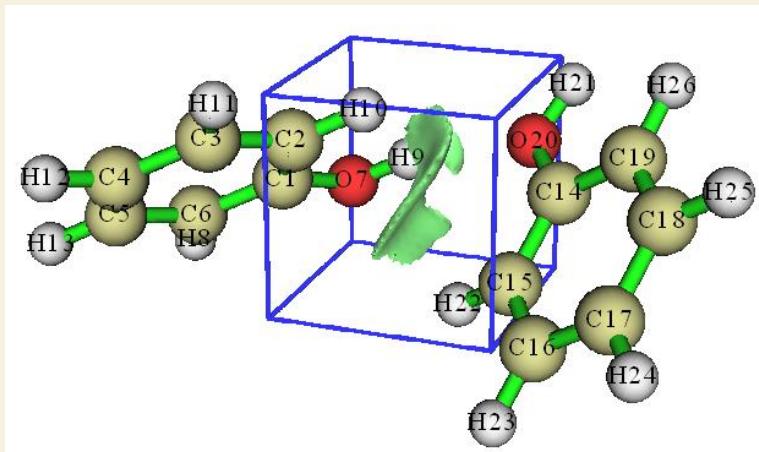
3,3,3 // Extension distance in all directions are 3 Bohr

-I // Show the isosurface of RDG

Make sure that the isovalue in the GUI window is set to 0.5, which is suitable for visualizing weak interaction regions (if the isovalue is too small, then isosurface will be too thin and thus ugly; if too large, then unwanted "Around nuclei" and "Around chemical bond" regions will appear)

-I // Show the isosurface

You will get below graph:

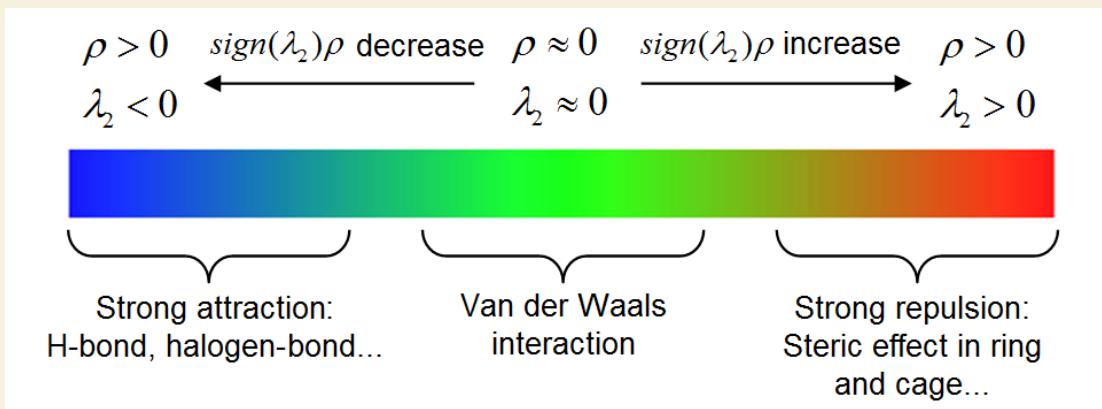


The green isosurface represents the weak interaction region between phenol dimer very clearly. Notice that by default, **RDG function is set to an arbitrarily large value (100.0) where electron density is larger than or equal to 0.05**, so that the isosurfaces in the regions “around chemical bond” can be shielded. The threshold is determined by “RDG_maxrho” parameter in settings.ini file. The default 0.05 is suitable for visualizing weak interaction regions for most cases. If you do not want to enable the density screening treatment due to special reason, you can set RDG_maxrho to zero.

The cubic blue frame in above graph shows spatial scope of the calculated grid data, it is only visible when “Show data range” is checked in the GUI window. Because the extension distance from center of the grid data was set to 3.0 Bohr, the side length is $2 \times 3 = 6$ Bohr.

Discriminate weak interaction types by filling color to RDG isosurfaces

In Bader’s AIM theory, the (3,-1) type of critical points indicate electron density is aggregated at that points, they appear in chemical bond path or between the atoms which have attractive interaction; while (3,+1) type critical point indicates electron density is depleted at that point and displays steric effect, it generally appears in the center of ring or cage systems. The criterion for distinguishing (3,-1) and (3,+1) is the second largest eigenvalue of Hessian matrix of electron density (referred to as λ_2 here); if λ_2 exceeds 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 corresponding to strong steric effect and hydrogen bonding always have relative large ρ . So we can define a real space function $\text{sign}[\lambda_2(\mathbf{r})]\rho(\mathbf{r})$, namely the product of sign of λ_2 and ρ . If we use different colors to represent value of this function according to below color bar and map it to RDG isosurfaces, we can not only know where weak interaction occurs, but also easily capture the interaction type.



Current Multiwfn does not support plotting color-filled isosurface graph, however, we can use Multiwfn to generate cube file for $\text{sign}(\lambda_2)\rho$ and RDG, and then use plotting script of VMD to draw such map. VMD is one of the best visualization tool and can be freely downloaded at <http://www.ks.uiuc.edu/Research/vmd>. Here I illustrate how to do this for phenol dimer by using subfunction 1 of main function 20. This time we do not only want to study the weak interaction region between the two monomers, but also want to examine the steric effect within in aromatic ring of phenol, therefore the spatial scope of grid data should be set to cover the entire dimer.

Boot up Multiwfn and input following commands

```
examples\PhenolDimer.wfn
```

```
20 // Visual study of weak interaction
```

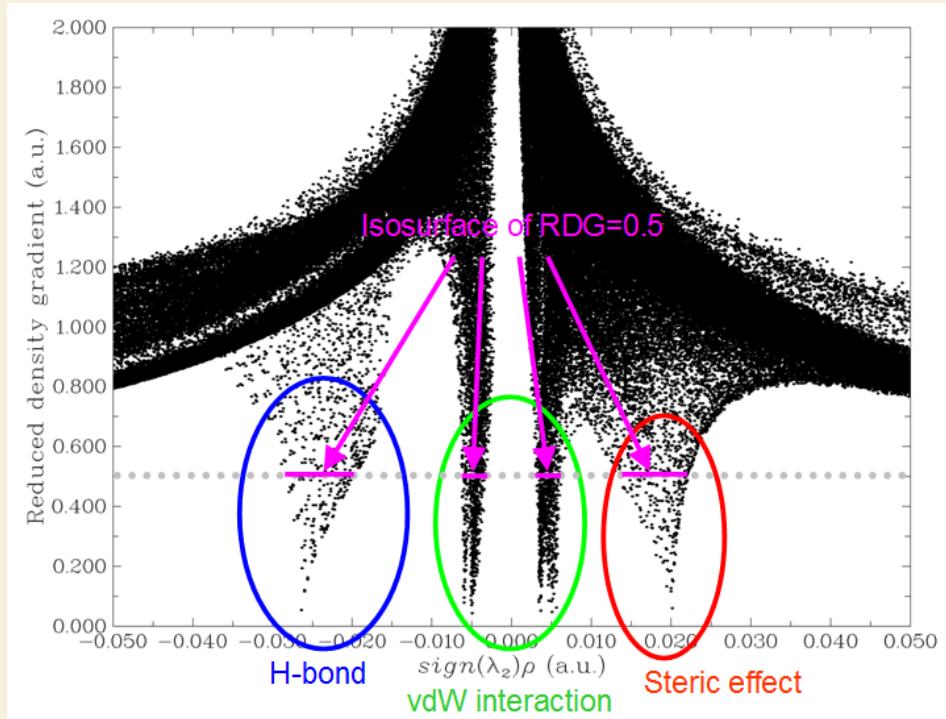
```
1 // NCI analysis
```

```
-10 // Set extension distance in all directions with respect to molecular boundary
```

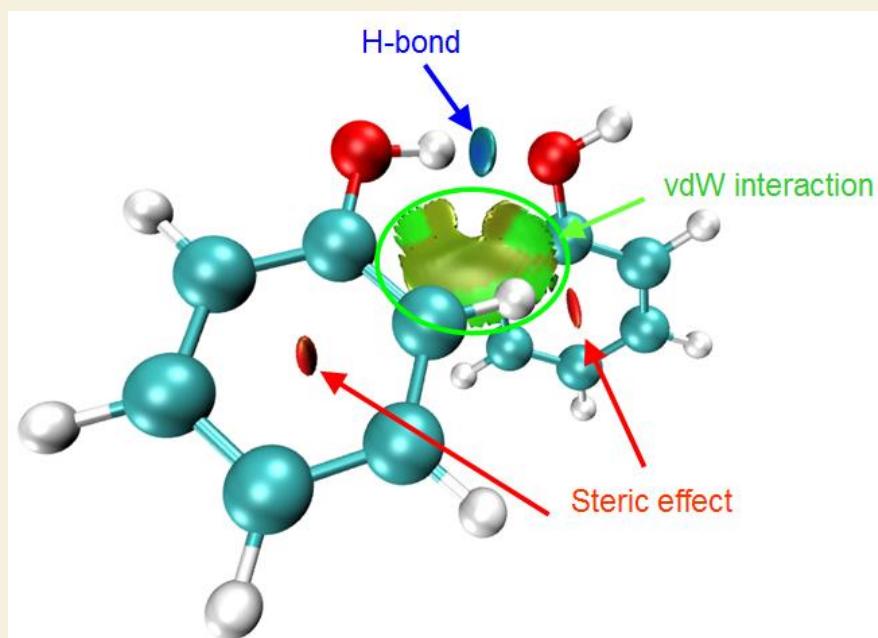
0 // Because weak interaction regions only appear in internal region of present system, we do not need to leave a buffer region at system boundary, so we set the extension distance to 0 Bohr

2 // Medium quality grid (totally about 512000 points). Because the spatial scope of grid data is evidently larger than last example, we need more grid points than last example, otherwise the the RDG isosurfaces will look discrete

I first discuss the characteristics of different kinds of regions via scatter graph, I think it will be helpful to understand the nature of the NCI method. Select option -1, a scatter graph immediately pops up (you can also select option 1 to export this graph as file). The X-axis and Y-axis correspond to $\text{sign}(\lambda_2)\rho$ and RDG functions, respectively; each point in the graph corresponds to a grid point in 3D space. There are four spikes, the points at their peaks are just the approximate positions of critical points in AIM theory. If you draw a horizontal line on the graph, then the segments crossing the spikes just correspond to the RDG isosurfaces. So, the NCI analysis method can be regarded as an extension of AIM theory for visual study. The spikes can be classified into three types, I marked them by blue, green and red circle, see below.

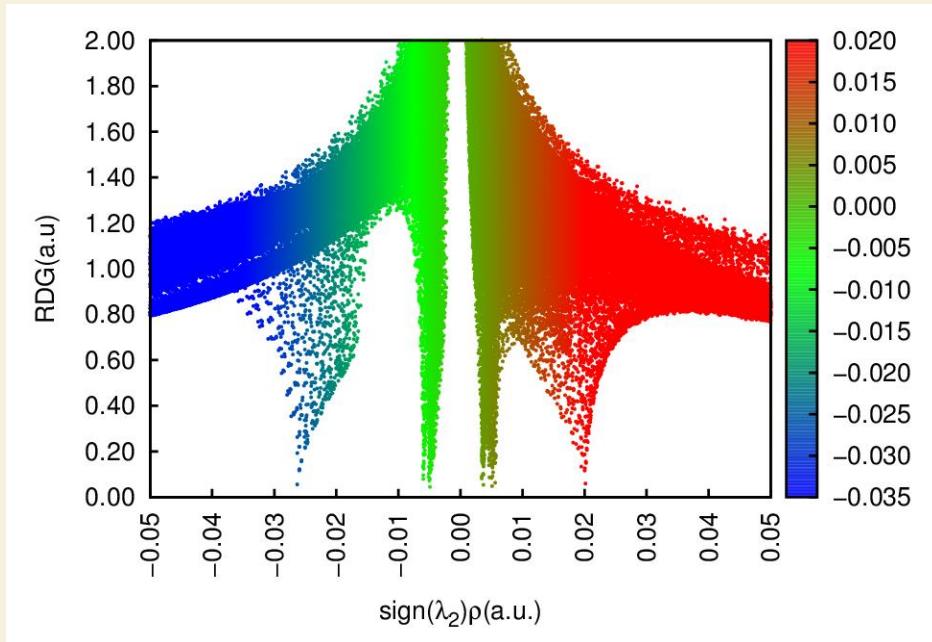


Then close the scatter map, select option 3 to export grid data of $\text{sign}(\lambda_2)\rho$ and RDG to func1.cub and func2.cub in current directory, respectively, then copy them and the RDGfill.vmd file in “examples” folder to VMD installation directory. The RDGfill.vmd is a plotting script of VMD written by me. Boot up VMD (here I use version 1.9.2), select “file”-“Load state”, choose “RDGfill.vmd” (alternatively, you can input *source RDGfill.vmd* in console window), you will see the graph below in OpenGL window. The default RDG isosurface is 0.5, and color range is -0.035 to 0.02. You can manually edit the RDGfill.vmd to change the default settings, the current values are suitable for general cases. (for present example, in VMD I suggest selecting “Display”-“Light 3” to enable light with index of 3, otherwise some parts of the isosurfaces will be in shadow and the color cannot be faithfully exhibited)



From the color-filled RDG isosurface, we can identify different type of regions by simply examining their colors. Recall the color scale bar I showed you, the more blue implies the stronger attractive interaction; it can be seen that the elliptical slab between oxygen and hydrogen shows light blue color, so we can conclude that there is a hydrogen bond, but not very strong. The interaction region marked by green circle can be identified as vdW interaction region, because the mapped color is green or light brown, which exhibits that 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.

It is also possible to map color to scatter map to facilitate identification of correspondence between spikes and RDG isosurfaces. A plotting script of gnuplot program (<http://www.gnuplot.info>) is provided as *examples\RDGmap.gnu*, which can realize this. First, select "2 Output scatter points to output.txt in current folder", then delete first three columns of data from the exported file (you can use text editor with column mode such as Ultraedit or notepad++ to do this, or simply use this command under Linux: *cut -c 34-66 output.txt > t.txt;mv t.txt output.txt -f*). Then move the output.txt and RDGmap.gnu into the folder containing gnuplot executable file, run this command: *gnuplot RDGmap.gnu*. After a while you will obtain RDGmap.ps in present folder, this is a graphic file of postscript format, you can open it using such as Acrobat or photoshop. The graph looks like this:



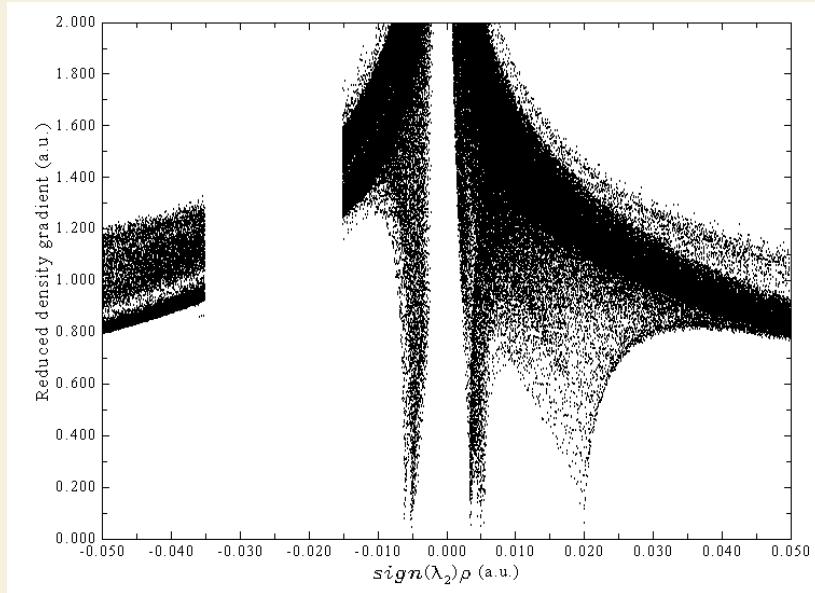
The default color range in this plotting script is from -0.035 to 0.02, if you intend to correlate this map with RDG color-filled map, you should ensure that the color scale setting in RDGmap.gnu and RDGfill.vmd are completely identical.

Interactively set RDG value where $\text{sign}(\lambda_2)\rho$ is in specific range

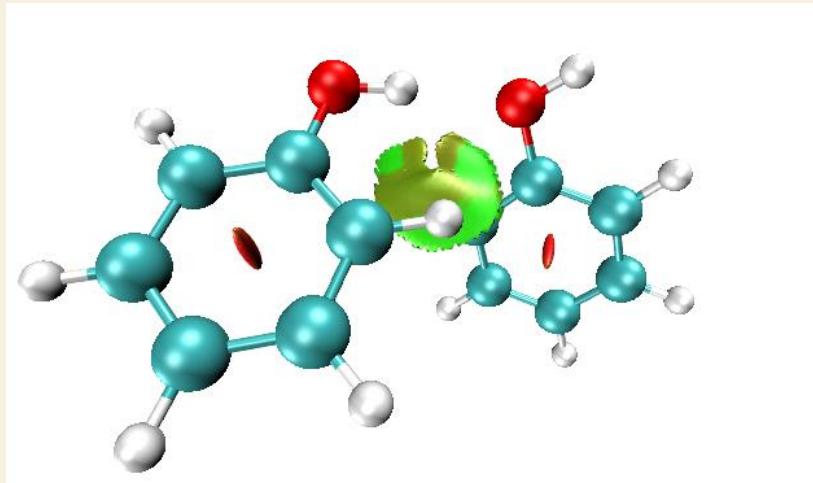
Multiwfn allows you to interactively set RDG value where $\text{sign}(\lambda_2)\rho$ is in specified range, using this feature you can easily screen regions of unwanted types. Here I continue the phenol dimer example and will illustrate how to screen H-bond RDG isosurface from the graph. From the scatter map, we know that the H-bond corresponds to range of $\text{sign}(\lambda_2)\rho$ of $-0.035 \sim -0.015$, therefore we input

```
-2 // Set RDG value where  $\text{sign}(\lambda_2)\rho$  in within given data range
```

```
-0.035,-0.015 // The lower and upper limit of sign( $\lambda_2$ ) $\rho$ 
100 // Set RDG value in these regions to a very large value to screen RDG isosurface
Then, if you select option -1 to plot the scatter map again, you will see
```



clearly the spike corresponding to H-bond no longer exists. We can also export the cube files and use VMD redraw the color-filled map, as shown below, the H-bond RDG isosurface indeed disappeared.



Notice that the grid data once modified as exemplified above cannot be retrieved.

Some worth mentioning points about NCI analysis

Here I talk more about grid setting for computing grid data of RDG and $\text{sign}(\lambda_2)\rho$, since this point significantly influences computational cost and quality of resulting RDG isosurface map. The total time spent in the calculation is linearly proportional to the total number of grid points, and the quality of RDG isosurface is highly dependent on grid spacing. The smaller the grid spacing, the more smooth the resulting isosurfaces. Too large grid spacing will finally result in severe jaggies at the edges of the isosurfaces. It is easy to understand, if the box size of grid data (*i.e.* spatial range to be calculated) is fixed, then the larger number the grid points you set, the smaller the grid spacing will be. So, the box should be properly defined. Its spatial range should not be too broad, otherwise

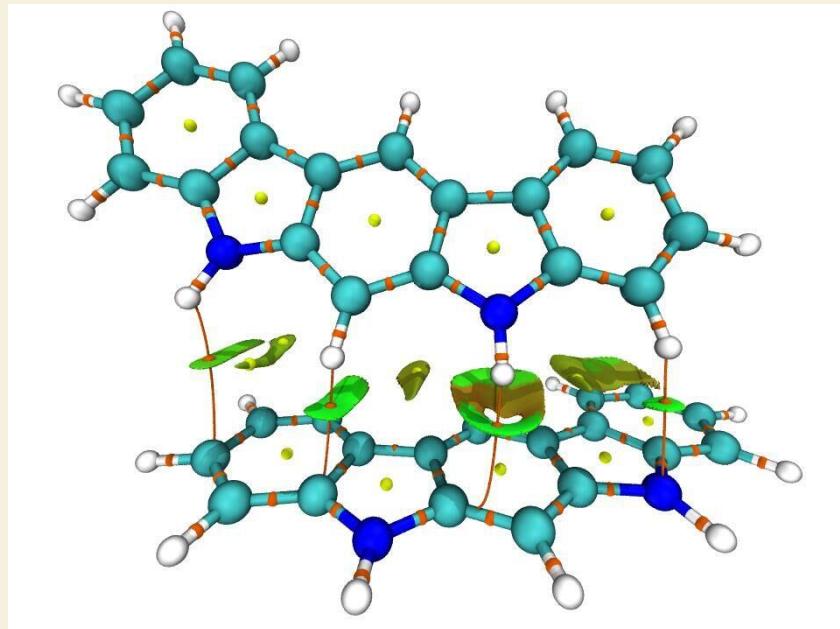
the grid spacing will be large and thus lower the graph quality; it should also not be too small, otherwise the interesting RDG isosurfaces may be truncated. The best practice is to make the box just enclose the interesting region. Then, if you can afford high computational cost, you may use as large number of points as possible to improve the final isosurface quality.

It is absolutely unnecessary to use large basis set to carry NCI analysis. Using moderate size of basis set such as def2-SVP or 6-31G* is completely adequate for NCI analysis, further enlarging the basis set is just waste of time. Regarding the choice of theoretical method, using popular functionals such as B3LYP or M06-2X to yield wavefunction is adequate. Although post-HF density is known to be more accurate than DFT density, the improvement in density quality is hardly reflected in the final NCI analysis result. You may know that the levels such as B3LYP/6-31G* perform quite bad for studying weak interactions, however, it never means that using electron density produced at this level is insufficient for NCI analysis, because electron density is never as sensitive as interaction energy to calculation level, and there is not strictly positive relationship between the quality of calculated interaction energy and electron density.

It is frequently encounter the problem that unexpected RDG isosurfaces occur around interesting regions and thus "pollute" NCI graph, this makes visual analysis of weak interaction at interesting regions difficult. For example, there is a system consisted of three molecules, we only want to study weak interaction between molecule 1 and 2; however, in the generated NCI graph, you may find unwanted isosurfaces corresponding to interactions between 1-3 and 2-3 as well as those corresponding to intramolecular interactions also occur. To screen the uninteresting isosurfaces, you can try to use the methods described in Section 4.13.4; alternatively, you can consider to use IGM method instead, which is completely free of this problem, see Section 3.23.5 for introduction.

Multiwfn is capable of integrating any real space function within isosurface defined by any real space function, therefore you can study such as volume and the number of electrons within isosurfaces of RDG (or other related functions such as DORI) to study weak interactions at quantitative level. Introduction about this kind of analysis is provided in Section 3.200.14 while illustrative examples are given in Section 4.200.14.

In VMD it is also possible to simultaneously plot AIM critical points and paths along with color-filled RDG map, so that more information about weak interaction may be revealed, below is an example (provided by yjmaxpayne@qq.com). The way of plotting this kind of map is exemplified in Section 4.20.1.



Information needed: Atom coordinates, GTFs

3.23.2 NCI analysis based on promolecular density (2)

Generating wavefunction and calculating grid data of RDG and $\text{sign}(\lambda_2)\rho$ for large system are very time-consuming, which greatly hinders application range of NCI analysis method. Fortunately, it is shown that the NCI analysis based on promolecular density is also reasonable in general. The so-called promolecular density is the electron density approximately constructed by superposing electron density of atoms in their free-state, this is known as “Promolecular approximation”. High quality free-state atomic electron density for almost all elements in periodic table are predetermined and built-in, hence NCI analysis based on promolecular density can be in principle used for any system in Multiwfn.

To carry out NCI analysis under promolecular approximation, just choose subfunction 2 in main function 20, all operation steps are completely identical to regular NCI analysis. Since only atom coordinate information is required for constructing promolecular density, any input file containing atomic coordinate information can be used as input file, such as the popular .pdb and .xyz formats.

The VMD plotting script for NCI analysis based on promolecular density is offered as *examples\RDGfill_pro.vmd*, which is slightly different to *examples\RDGfill.vmd* in the default setting of color scale and isovalue.

By default, RDG value is automatically set to 100.0 where ρ is larger than 0.1 when promolecular approximation is used, this threshold may not be suitable for certain circumstances. You can manually change the threshold by “RDGprodens_maxrho” in settings.ini.

Examples of performing NCI analysis based on promolecular density is given in Section 4.20.1

Information needed: Atom coordinates

3.23.3 Averaged NCI analysis (NCI analysis for multiple frames. 3)

Theory

In *J. Chem. Theory Comput.*, **9**, 2226 (2013), the NCI method described in last sections is extended to analyzing fluctuation environment (*e.g.* molecular dynamics trajectory), resulting in the averaged NCI (aNCF) method. Present function aims at realizing the aNCI analysis.

The only difference between aNCI and the original NCI method is that in the former, the electron density ρ and its gradient norm $|\nabla\rho|$ are not calculated for only one geometry, but for multiple frames in a trajectory file, then get average (namely $\bar{\rho}$ and $\overline{|\nabla\rho|}$). Therefore, the isosurface of averaged reduced density gradient

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

can be directly used to reveal the averaged weak interaction regions for a dynamics process.

Similarly, in order to exhibit averaged weak interaction type, in aNCI method, the λ_2 term in $\text{Sign}(\lambda_2)\rho$ function is obtained as the second largest eigenvalue of the averaged electron density Hessian matrix computed through out the dynamical trajectory.

aNCI method also defines a new quantity named thermal fluctuation index (TFI) to reveal the stability of weak interaction

$$\text{TFI}(\mathbf{r}) = \frac{\text{std}[\rho(\mathbf{r})]}{\rho(\mathbf{r})}$$

whose numerator is standard deviation of electron density in the dynamical trajectory, which can be calculated as

$$\text{std}[\rho(\mathbf{r})] = \sqrt{\frac{\sum_i [\rho_i(\mathbf{r}) - \overline{\rho(\mathbf{r})}]^2}{n}}$$

where n is the number of frames in consideration, ρ_i is the density calculated based on the geometry of frame i . After mapping TFI on the isosurface of aNCI, the stability of each weak interaction region can be clearly identified by visually examining the colors.

The quality of aNCI graph directly depends on the number of frames that taken into account. Small number of frames, for example 50 frames, can only leads to inaccurate and very unsmooth isosurface graph. In general, at least 500 frames are should to be used to generate aNCI graph.

Usage

Firstly, note that since calculating electron density based on wavefunction for large number of geometries is very expensive, promolecular approximation is forced to be used in the aNCI analysis function of Multiwfn. This approximation is highly reasonable and always works well.

The trajectory stored in .xyz file is allowed as input file. You can use such as VMD program to convert other format of trajectory files to .xyz trajectory file.

PS: The structure of a multiple frame .xyz file looks like below

[Number of atoms in frame 1]

[Element, x, y and z of atom 1 in frame 1]

[Element, x, y and z of atom 2 in frame 1]

```

...
[Element, x, y and z of atom n in frame 1]
[Number of atoms in frame 2]
[Element, x, y and z of atom 1 in frame 2]
[Element, x, y and z of atom 2 in frame 2]
...
[Element, x, y and z of atom n in frame 2]
[Number of atoms in frame 3]
...

```

In all of the frames, the coordinate of the interesting molecule should be fixed. For example, if you want to study the weak interaction between solvents and a benzene molecule, then the position of the benzene must be fixed through out the whole trajectory. Commonly, the geometry center of interesting molecule is fixed at box center.

After you enter present function, you will be prompted to input the frame range to be analyzed, for example inputting 140,450 means the frame from 140 to 450 will be used in the aNCI analysis. Then you need to set up grid, the spatial range of the box should properly enclose the molecule of interest. After that, averaged electron density, averaged density gradient and averaged density Hessian will be calculated for each frame, you should wait patiently. Once the calculation is finished, you can use corresponding options to draw scatter graph between averaged NCI and averaged $\text{sign}(\lambda_2)\rho$, output scatter points, export their cube files, etc. Thermal fluctuation index can also be calculated and export to cube file.

An example is given in Section 4.20.3.

Information needed: Multiple frames of atom coordinates

3.23.4 Density Overlap Regions Indicator (DORI) analysis (5)

Sometimes ELF and RDG are used in combination to simultaneously investigate covalent and noncovalent interactions, see *J. Chem. Theory Comput.*, **8**, 3993 (2012) for example. Is it possible to study both kinds of interactions by *single* real space function? The answer is yes. In *J. Chem. Theory Comput.*, **10**, 3745 (2014) the authors proposed a function named density overlap regions indicator (DORI), it was found that when properly choosing an isosurface, both covalent and noncovalent interaction regions can be well exhibited, and $\text{sign}(\lambda_2)\rho$ can also be mapped on to DORI isosurfaces to facilitate analysis of the nature of interactions.

The expression of DORI is

$$\text{DORI}(\mathbf{r}) = \theta(\mathbf{r}) / [1 + \theta(\mathbf{r})]$$

where $\theta(\mathbf{r}) = [\nabla(\nabla\rho(\mathbf{r}) / \rho(\mathbf{r}))^2]^2 / [\nabla\rho(\mathbf{r}) / \rho(\mathbf{r})]^6$. The value range of DORI is [0,1].

To plot the $\text{sign}(\lambda_2)\rho$ mapped DORI isosurface map, enter subfunction 5 of main function 20, then following operations are exactly identical to NCI analysis. After exporting the grid data of $\text{sign}(\lambda_2)\rho$ and DORI as cube files by option 3 in post-process menu, you can then copy them along with *examples\DORIfill.vmd* to VMD installation folder, then use the plotting script DORIfill.vmd to plot the color-filled isosurface map.

It is worth to mention that isosurface of RDG in fact is also able to reveal covalent and noncovalent interaction regions at the same time when "RDG_maxrho" parameter in settings.ini is

set to 0, but the graphical effect is not as good as DORI, because perfectly exhibiting these two kinds of regions requires somewhat different RDG isovales.

An example of performing DORI analysis is given in Section 4.20.1

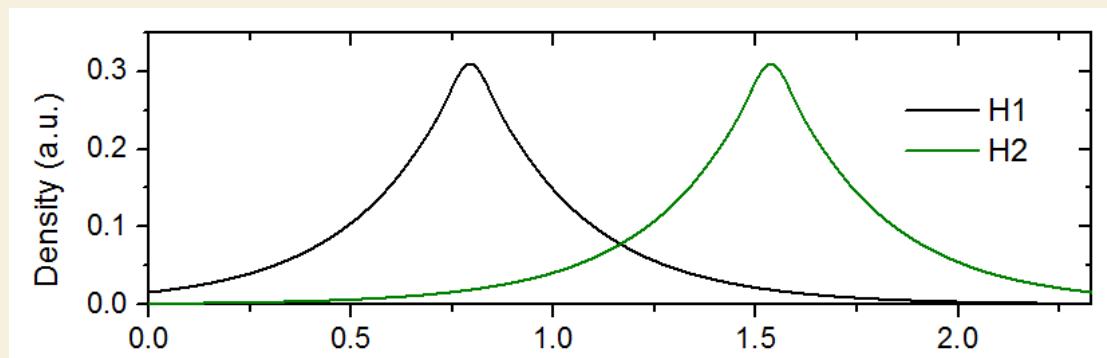
Information needed: Atom coordinates, GTFs

3.23.5 Independent Gradient Model (IGM) analysis based on promolecular density (10)

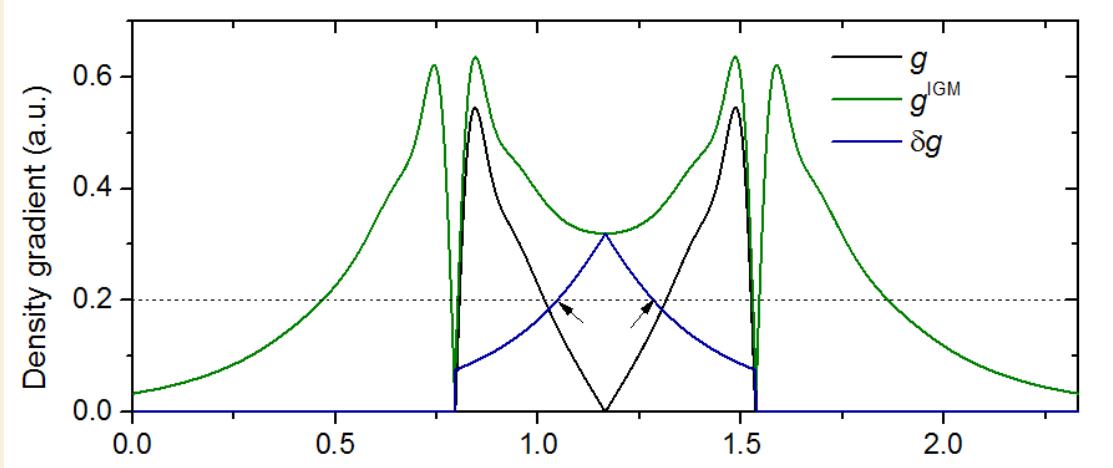
Theory

In *Phys. Chem. Chem. Phys.*, **19**, 17928 (2017), the authors proposed a very useful way of visually studying interfragment and intrafragment interactions, named Independent Gradient Model (IGM). Full introduction of IGM can be found in the original paper, below I outline the essential idea of this method from my personal point of view. Note that IGM has two versions, the one based on promolecular density and the one based on actual wavefunction (*ChemPhysChem*, **19**, 724 (2018)), what the Multiwfn supported and introduced below is the former. The atomic densities used in constructing promolecular density for IGM is the same as the one used in NCI method under promolecular approximation (Section 3.23.2).

Let us first look at a very simple system, the H₂ molecule. The atomic density in free-state of each atom along the molecular axis is shown as below



From above graph one notices that the gradient of atomic density of the two atoms in the interatomic region have opposite signs. For example, at the position of X=1.2, the density gradient of H1 is negative, while that of H2 is positive. Therefore, in the gradient of promolecular density (the curve g in below map), the contribution from the two atoms largely cancel with each other in the region between the two atoms. Note that at the midpoint of the two hydrogens, g is exactly zero, such point corresponds to bond critical point (BCP) in AIM theory under promolecular density.



In above map, the g^{IGM} is IGM type of gradient, it is calculated as sum of *absolute value* of density gradient of each atom in their free-states; in other words, phase is completely ignored and thus the density gradient originating from various atoms never cancel with each other. Due to this feature, g^{IGM} is upper limit of g .

The δg function is defined as the difference between g^{IGM} and g , it is plotted as deep blue curve in above map. It can be seen that δg is non-zero in the atom interaction region, and has maximum value at the BCP position. Clearly, δg could be used to reveal interaction regions like reduced density gradient (RDG). In addition, as will be illustrated in the examples in Section 4.20.10, magnitude of δg in interaction region has close relationship with interaction strength.

In general case, g^{IGM} and δg may be defined as follows

$$g(\mathbf{r}) = \left| \sum_i \nabla \rho_i(\mathbf{r}) \right| \quad g^{\text{IGM}}(\mathbf{r}) = \left| \sum_i \text{abs}[\nabla \rho_i(\mathbf{r})] \right|$$

$$\delta g(\mathbf{r}) = g^{\text{IGM}}(\mathbf{r}) - g(\mathbf{r})$$

where $\text{abs}()$ is the function to take absolute value for each component of a vector.

Based on the idea of g^{IGM} and δg , the IGM method also defines δg^{inter} and δg^{intra} aiming to study interfragment and intrafragment interactions, respectively

$$g^{\text{inter}}(\mathbf{r}) = \left| \sum_A \sum_{i \in A} \nabla \rho_i(\mathbf{r}) \right|$$

$$g^{\text{IGM,inter}}(\mathbf{r}) = \left| \sum_A \text{abs} \left[\sum_{i \in A} \nabla \rho_i(\mathbf{r}) \right] \right|$$

$$\delta g^{\text{inter}}(\mathbf{r}) = g^{\text{IGM,inter}}(\mathbf{r}) - g^{\text{inter}}(\mathbf{r})$$

$$\delta g^{\text{intra}}(\mathbf{r}) = \delta g(\mathbf{r}) - \delta g^{\text{inter}}(\mathbf{r})$$

where A and i are index of fragments and atoms, respectively. The fragment can be arbitrarily defined according to character of actual system and research purpose. (The above expression of δg^{inter} and δg^{intra} should be their most general forms and are employed in Multiwfns, they are not explicitly given in the IGM original paper)

The idea of δg^{inter} is easy to understand from above formula. One first calculate density gradient in usual way as g^{inter} , and then calculate the $g^{\text{IGM,inter}}$, which ignores cancellation effect of density gradient of various fragments due to possible different phases, then the difference between $g^{\text{IGM,inter}}$

and g^{inter} , namely δg^{inter} , must be able to reveal the interaction between the fragments. The δg reveals all kinds of interactions in present system, irrespective of the type is interfragment or intrafragment. Therefore, if δg^{inter} is subtracted from δg , the remaining part, namely δg^{intra} , must be capable of revealing intrafragment interactions.

In Section 3.23.1, it is shown that interaction region and interaction type can be simultaneously exhibited by plotting RDG isosurface map colored by $\text{sign}(\lambda_2)\rho$ function. Similarly, if $\text{sign}(\lambda_2)\rho$ function is mapped on δg^{inter} and δg^{intra} isosurfaces using various colors, the type and position of inter- and intra-fragment interactions could also be vividly revealed.

Motivated by the IGMplot program manual, in Multiwfn, I implemented atom pair δ index to quantify the contribution of atom pair to interaction between two fragments (A and B),

$$\begin{aligned}\delta_{i,j} &= \int [g_{i,j}^{\text{IGM}}(\mathbf{r}) - g_{i,j}(\mathbf{r})] d\mathbf{r} \quad i \in A, j \in B \\ g_{i,j}^{\text{IGM}}(\mathbf{r}) &= |abs[\mathbf{g}_i(\mathbf{r})] + abs[\mathbf{g}_j(\mathbf{r})]| \\ g_{i,j}(\mathbf{r}) &= |\mathbf{g}_i(\mathbf{r}) + \mathbf{g}_j(\mathbf{r})|\end{aligned}$$

and defined atom δ index to quantitatively measure importance of atom to interfragment interaction

$$\delta_i^{\text{atom}} = \sum_j \delta_{i,j} \quad i \in A, j \in B$$

When plotting molecule structure, if atoms are colored according to the atom δ indices, then the relative importance between various atoms to interfragment interaction can be immediately captured.

According to my viewpoint and experiences, the advantage of IGM method over the popular NCI method can be summarized as follows:

- The inter- and intra-fragment interactions can be studied individually and thus mutual interference is avoided
- Calculation of the functions defined by IGM method is rather fast and only dependent of geometry, thus the method can be applied to broad range of systems. (Note that NCI also has promolecular approximation version)
- As will be shown in the examples in Section 4.20.10, the isosurface graph given by the IGM method is more plump than the NCI map, and thus the IGM map has low requirement on grid spacing. In contrast, NCI graph is prone to ugly jaggies and holes when the grid points are sparse.
- Contribution of atoms and atomic pairs to interfragment interaction can be quantified, and the former can be vividly rendered on molecular structure, these features make identification of "hot" atoms easy.
- The value of δg function in interaction region directly reflects interaction strength. In particular, δg at bond critical point of AIM theory is a good quantitative indicator of strength of corresponding interaction.

Usage

Using Multiwfn to carry out IGM analysis is extremely easy and flexible. First, you should load a file containing atom coordinates. The most commonly used formats such as .xyz, .pdb and .mol are all supported by Multiwfn (Of course, any wavefunction file such as .wfn and .fch can also be used). Notice that the geometry must have been optimized using proper theoretical level,

otherwise the IGM result may be misleading.

IGM module is subfunction 10 of main function 20, after you enter this module, you should define fragments. The definition of fragment is rather flexible, you can define any number of fragments (at least one fragment). No atom should be simultaneously shared by two or more fragments. The union set of defined fragments is not forced to be equal to the whole system, only the atoms in the defined fragments will be finally taken into calculation.

Next, you need to setting up grid, it is better to make the box just enclose the region where the interesting interactions may occur. Some advices about setting grid are given in Section 3.23.1.

After that, Multiwfn starts to calculate grid data of $\text{sign}(\lambda_2)\rho$, δg , δg^{inter} and δg^{intra} . Assume that you have defined n fragments, then the calculated δg^{inter} will correspond to interaction between all the n fragments, while δg^{intra} will correspond to intrafragment interactions within all the n fragments. The larger number of atoms in the fragments, the higher the overall cost.

The options in post-process menu are self-explained, I briefly describe them here:

-1: Suboptions 1, 2 and 3 of this option are used to draw scatter map of δg , δg^{inter} , δg^{intra} vs. $\text{sign}(\lambda_2)\rho$, respectively. While suboption 4 is used to draw δg^{inter} and δg^{intra} vs. $\text{sign}(\lambda_2)\rho$ simultaneously with different colors. As shown in the original paper of IGM, this kind of map is useful for discussing details about interactions (Recall that RDG vs. $\text{sign}(\lambda_2)\rho$ scatter map is frequently involved in NCI analysis). If you want to directly save the scatter map in current folder as graphic file, use option 1. If the default axis range is not appropriate, use option -2 or -3 to adjust.

2: If you want to draw scatter map using third-part softwares such as Origin and gnuplot, use this option to export data of δg , δg^{inter} , δg^{intra} and $\text{sign}(\lambda_2)\rho$ to plain text in current folder. Meaning of each column of this file is shown on screen.

3: Output grid data of $\text{sign}(\lambda_2)\rho$, δg , δg^{inter} and δg^{intra} to cube file in current folder. After exporting the cube files, you can use IGM_inter.vmd and IGM_intra.vmd scripts in "examples" folder to draw color-filled δg^{inter} and δg^{intra} isosurfaces map in VMD, respectively. See examples of Section 4.20.10.

4: This option is used to directly visualize isosurface of $\text{sign}(\lambda_2)\rho$, δg , δg^{inter} or δg^{intra} in Multiwfn.

5: This option is used to set δg^{intra} to zero where $\text{sign}(\lambda_2)\rho$ is not within specified value range. By this option uninteresting regions could be screened from δg^{intra} scatter and isosurface maps. For example, we merely want to study weak intrafragment interactions, then we can input the range corresponding to relatively small value of $\text{sign}(\lambda_2)\rho$. (The aim of this option resembles *RDG_maxrho* parameter used in NCI analysis)

6: This option is used to evaluate atom and atom pair δ indices. If you have defined more than two fragments, here you need to choose two fragments for which the δ indices will be calculated. Multiwfn will compute δg grid data of every atom pair between the two fragments and calculate integral of δg to derive the δ indices. The computational cost will be relatively high if there are large number of atoms in the interfragment contact region. After the calculation is finished, atmrdg.txt will be outputted to current folder, which records all atom and atom pair δ indices, the values are sorted from high to low. Then the program asks you if also outputting atmrdg.pdb in current folder, which contains all atoms in present system. The Beta field of this pdb file corresponds to ten times of atom δ index. If you load this file into VMD visualization program and color the atoms according to Beta field, then relative importance of various atoms to interfragment interactions can be intuitively

identified.

7&8 These two options are used to set value of δg and δg^{inter} respectively if the sign($\lambda_2\rho$) at corresponding grid is out of specific range. Clearly, these options are useful when you want to screen unwanted region from IGM isosurface map. For example, you only want to visualize isosurface of δg^{inter} where sign($\lambda_2\rho$) is within -0.04~-0.025, then you can enter option 8, input -0.04,-0.025 and then input 0 to set δg^{inter} of these grids to zero; next, you can plot the updated scatter map or export cube files to visualize IGM map in VMD.

If your input file contains GTF or basis function information (e.g. .wfn, .fch, .molden), when carrying out IGM analysis, Multiwfn will let you choose the kind of the sign($\lambda_2\rho$) to be used, the first one is that based on actual electron density, while the second one is that based on promolecular density. Using the former one should give more meaningful result, however, calculation cost for the former is evidently higher than the latter (if your input file only contains atomic coordinate information, the latter one is always used).

Several examples of IGM analysis are given in Section 4.22.10. More discussions and instances about IGM method can be found in my blog article "Investigating intermolecular weak interactions via Independent Gradient Model (IGM)" (in Chinese, <http://sobereva.com/407>).

Information needed: Atom coordinates

3.24 Energy decomposition analysis (21)

3.24.1 Energy decomposition analysis based on molecular forcefield (EDA-FF)

Energy decomposition analysis (EDA) is an important method to reveal the nature of interaction. Most EDA analysis methods are based on wavefunction, they are accurate, rigorous and the results are meaningful, unfortunately they are often too expensive for large systems. Present module is designed for analyzing intramolecular and intermolecular weak interactions based on classical molecular force field (FF), this method could be referred to as EDA-FF, the computational cost is negligible for systems consisted of hundreds of atoms, and it can even be applied to systems with thousands of atoms. Due to limitation of FF, this module evidently cannot be used to discuss the natural of chemical bond interactions. The word "weak interaction" in this section refers to the interatomic interaction separated by more than three bonds.

Theory

There are many popular FFs for molecular systems. The major ingredients of weak interactions are van der Waals (vdW) interaction and electrostatic interaction, most FFs represent them by mean of pairwise potential, as shown below.

- Electrostatic interation energy between atoms A and B (atomic unit is used):

$$E_{AB}^{\text{ele}} = \frac{q_A q_B}{r_{AB}}$$

where q is atomic charge and r_{AB} is distance between A and B.

- vdW interaction energy between atoms A and B :

$$E_{AB}^{\text{vdW}} = E_{AB}^{\text{rep}} + E_{AB}^{\text{disp}}$$

$$E_{AB}^{\text{rep}} = \epsilon_{AB} \left(\frac{R_{AB}^0}{r_{AB}} \right)^{12} \quad E_{AB}^{\text{disp}} = -2\epsilon_{AB} \left(\frac{R_{AB}^0}{r_{AB}} \right)^6$$

where E^{rep} represents repulsive interaction due to Pauli repulsion effect (also known as exchange-repulsion), while E^{disp} is attractive dispersion interaction. The ϵ_{AB} is well depth of interatomic vdW interaction potential, while R_{AB}^0 is vdW nonbond distance. When $r_{AB}=R_{AB}^0$, the interaction energy just corresponds to well depth.

The parameters ϵ and R^0 are provided by FFs, and the values are commonly defined for each atom type. The interatomic parameters used in practical calculation are commonly evaluated as geometric average or arithmetic average of atomic parameters. For example, in UFF forcefield, below mixing rule is used

$$\epsilon_{AB} = \sqrt{\epsilon_A \epsilon_B} \quad R_{AB}^0 = \sqrt{R_A^0 R_B^0}$$

While for some other FFs such as AMBER and GAFF, atomic ϵ and R^* are defined, the employed mixing rule is

$$\epsilon_{AB} = \sqrt{\epsilon_A \epsilon_B} \quad R_{AB}^0 = R_A^* + R_B^*$$

where R^* is known as atomic nonbond radius or atomic vdW radius.

Given the interatomic interaction terms, evaluating various physical components of interfragment interaction energy is straightforward:

$$E_{IJ}^{\text{ele}} = \sum_{A \in I} \sum_{B \in J} E_{AB}^{\text{ele}} \quad E_{IJ}^{\text{rep}} = \sum_{A \in I} \sum_{B \in J} E_{AB}^{\text{rep}} \quad E_{IJ}^{\text{disp}} = \sum_{A \in I} \sum_{B \in J} E_{AB}^{\text{disp}}$$

The present module is mainly used to evaluate above three terms between two or more user-defined fragments, many useful quantities can be obtained at the meantime.

Usage

The basic steps of using this module are:

- (1) Prepare a "molecule list" file, which contains paths of "molecule type" files. Detailed description is given later.
- (2) Load a file containing geometry information of the whole system. Evidently, many formats supported by Multiwfn could be used, for example, .xyz, .mol, .pdb, .fch and so on.
- (3) Enter subfunction 1 of main function 21.
- (4) Use option 3 to load molecule list file. This step is used to assign atomic charge and type for each atom.
- (5) Use option 2 to define fragments. Infinite number of fragments may be defined, any atom should not simultaneously appear in two or more fragments. If you want to study interaction between two fragments in the same molecule, any atom pair between the two fragments should be separated by at least three bonds (otherwise the interaction will no longer belong to the scope of weak

interaction). Note that the order of the steps (4) and (5) could be exchanged.

(6) Choose option 1 to start analysis, then interaction energy components between each fragment pair, as well as atomic contributions will be printed. Before the analysis, if you want to check whether the charges and types have been properly assigned, you can choose option 4.

There are other options you can select before calculation:

- Option -1: This is used to choose the FF employed in the calculation. Currently, AMBER99 & GAFF (default) and UFF are supports, their difference is that the built-in atomic vdW parameters and the mixing rule used in the calculation are different.

- Option -2: This option can choose the operator in calculating electrostatic interactions. By default $1/r$ operator is employed, while using this option you can change it to $1/r^2$. Obviously, the attenuation of E^{ele} calculated based on $1/r^2$ with respect to interaction distance r is much faster than the default case, this is why some studies employ this simple strategy to effectively exhibit water environment, since it is well-known that polar solvents such as water can significantly shield electrostatic interaction strength due to its large dielectric constant.

- Option -3: If you choose this option once to switch the status to "Yes", then after calculation, all interatomic interaction energy terms including their physical components will be exported to *interatm.txt* in current folder.

- Option -4: In the standard .pqr format, the last two columns are specific for storing atomic vdW radii and atomic charges. If you choose this option once to switch the status to "Yes", then during calculation, *atmint_tot.pqr*, *atmint_ele.pqr*, *atmint_rep.pqr* and *atmint_disp.pqr* will be outputted in current folder, their last columns record atomic contribution to total, electrostatic, repulsion and dispersion interaction energies, respectively. In the popular VMD visualization program, you can load one of these .pqr file and color the atoms according to "atomic charges" data, then the atom colors will vividly exhibit contribution of each atom to corresponding kind of interaction energy.

Next, I introduce the rule of writing the molecular list file. The content of the file should look like this:

```
C:\mol1\phenol.txt 1  
C:\mol2\H2O.txt 4  
C:\HCl.txt 2
```

This example file implies that, in the geometry information provided by the file loaded when Multiwfn boots up, the recording sequence is: one phenol molecule, four H₂O molecules and two HCl molecules. The three .txt files contain atom types (case sensitive) and charges of respective molecule. For example, below is the content of the *C:\mol2\H2O.txt*, which records information of the atoms in H₂O molecule, the OW and HW are atom types of AMBER force field.

```
OW -0.728713  
HW 0.364427  
HW 0.364286
```

The first column is atom type corresponding to the FF you currently choose, the second column corresponds to atomic charge. Notice that the atom sequence in this file must be exactly identical to the geometry information loaded when Multiwfn boots up. If the forcefield employed in analysis is UFF, then this file should only contain atomic charge, thus there should be only one column (because

for each element, all relevant atom types share the same UFF vdW parameters, therefore users do not need to define atom types).

- About atom types: Detailed description of atom types can be found in original paper of corresponding forcefields. For AMBER, see Table 1 of *J. Am. Chem. Soc.*, **117**, 5179 (1995). For GAFF, see Table 1 of *J. Comput. Chem.*, **25**, 1157 (2004); you can also consult the *AMBER99.txt* and *GAFF.txt* in "examples\EDA\EDA_FF" folder for description of atom types. In general, you can manually find appropriate atom type for each atom in current system according to its actual chemical environment. However, if you feel this process is troublesome, you can use third-part programs to help you to identify atom types and construct the molecular files. For example, GaussView can automatically assign AMBER atom types (enter "Atom List", click the icon with large organge "M" symbol, double click head of "AMBER Type" column, select "File"- "Export Data", then extract the data corresponding to "AMBER Type" column), while Antechamber utility in AmberTools package is able to assign GAFF atom types. Note that AMBER and GAFF atom types can be mixed together in the same molecule file, since these two forcefields are completely compatible with each other, the AMBER and GAFF atom types use upper and lower case, respectively.

Note: The atom types assigned by GaussView are always in upper case, however, some atom types of AMBER99 are lower case, e.g. Br. Clearly, you should manually make modification before loading the file into Multiwfn. If you are confused, take a look at *examples\EDA\EDA_FF\AMBER99.txt*.

- On the choice of forcefield: For organic type of systems, commonly I suggest using AMBER/GAFF forcefield to conduct the analysis, the result should be reasonable and chemically meaningful. In fact, since vdW parameters of GAFF are directly inherited from AMBER, commonly there should be no different between using GAFF and AMBER atom types. Evidently, the geometry used in the analysis should be firstly optimized under reasonable level. Using UFF is generally deprecated, since I found that when UFF is employed, the total interfragment interaction energy is usually positive due to overestimation of E^{rep} , even if the geometry has already been substantially optimized with appropriate quantum chemistry method. A way to solve this problem is using the geometry optimized by UFF itself (many programs can do this, such as Gaussian and OpenBabel), however the resulting geometry for weakly interacting molecular dimer or multimer is often not quite good. The unique advantage of UFF is that it covers almost entire periodic table. Considering this, I designed a trick in Multiwfn: If you are using AMBER/GAFF, when atom type is written as *UF*, then UFF vdW parameter will be employed. This treatment greatly extends the application scope of AMBER and GAFF.

- On the choice of atomic charges: The atomic charges used for energy decomposition analysis should be able to reproduce electrostatic potential (ESP) around molecular vdW surface well. Commonly I suggest to use CHELPG atomic charge, which is obtained via ESP fitting process and can be directly calculated via Multiwfn, see Section 3.9.10 for introduction and Section 4.7.1 for example. Note that if a type of molecule appears more than once in current system with significantly different conformations, given that ESP fitting charges of each monomer may be very different to others, it is suggested to treat these replicas as different types of molecules, so that atomic charges can be individually assigned. If some monomers of the system is to large to calculate their ESP fitting atomic charges, you may change to EEM atomic charges using the parameters fitted for reproducing ESP fitting charges, see Section 3.9.15 for introduction. The computational cost of EEM charges is negligible for a system even composed of hundreds of atoms, since the calculation is purely based on molecular geometry information and empirical parameters.

Examples of this module are provided in Section 4.21.1.

Information needed: Atom coordinates and special files containing atomic charges/types

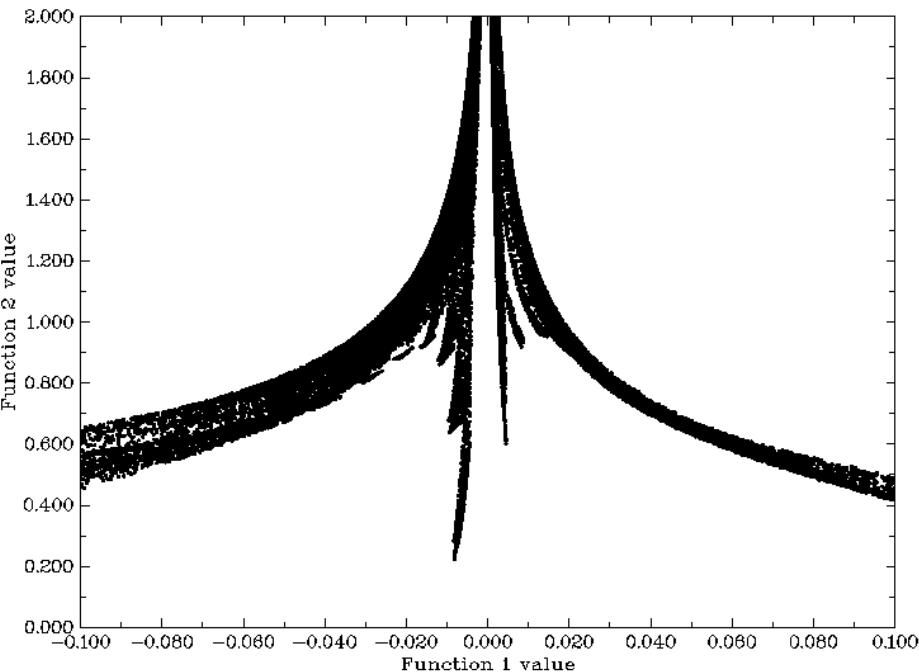
3.100 Other functions, part 1 (100)

Since Multiwfn has too many functions, some of the functions are relatively "small" compared to main functions, and some functions are not closely related to wavefunctions analysis, these functions are classified as "other functions". Because the number of subfunctions in "other functions" are huge, "other functions" are splitted as part1 (main function 100) and part 2 (main function 200). Part 1 will be described below, and part 2 will be introduced in Section 3.200.

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

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. In fact, the NCI/aNCI analysis and DORI analysis introduced in Section 3.20 are special instance of this function.

After you entered this function, select two real space functions that 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 referred to as functions 1 and 2 respectively below). Then select a mode to set up grid points. After that Multiwfn starts the calculation of grid data for them. 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.

Special usage: If you already have cube files of the two functions to be studied (referred to as func1.cub and func2.cub, respectively) and you want to directly use the functions in post-process menu (e.g. plotting scatter map, modifying values), you should follow these steps: Input the path of func1.cub after booting up Multiwfn, then enter main function 100 and select subfunction 1, then input 0,0 when selecting real space functions, then input the path of func2.cub, after that the grid data of the two cube files will be directly taken as function 1 and function 2. Notice that, of course, the grid setting of func1.cub and func2.cub must be exactly identical.

3.100.2 Export .pdb, .pqr, .xyz, .wfn, .wfx, .molden, .fch, NBO .47 or input file of mainstream quantum chemistry codes

This function can be used to output current structure to .pdb and .xyz files. Wavefunction information can be exported as .wfn or .wfx file when GTF information is available, and can be exported as .molden, .fch or GENNBO input file (.47) when basis function information is presented. By this function Multiwfn can be used as file format converter, e.g. .wfn/.wfx/.fch/.molden/.gms ...

→ .pdb/.xyz/.gjf/.wfn/.wfx/.molden/.fch/.47 ...

For convenience, this function is also able to yield basic input file for a batch of known quantum chemistry codes based on current structure, net charge and multiplicity, including Gaussian, GAMESS-US, ORCA, MOPAC, Dalton, MRCC, Molpro, NWChem, PSI, CFOUR, Molcas. The task type is single point except for MOPAC. For GAMESS-US input file, if basis function information is presented, you can select if writing the orbital coefficients as \$VEC field so that they can be used as initial guess.

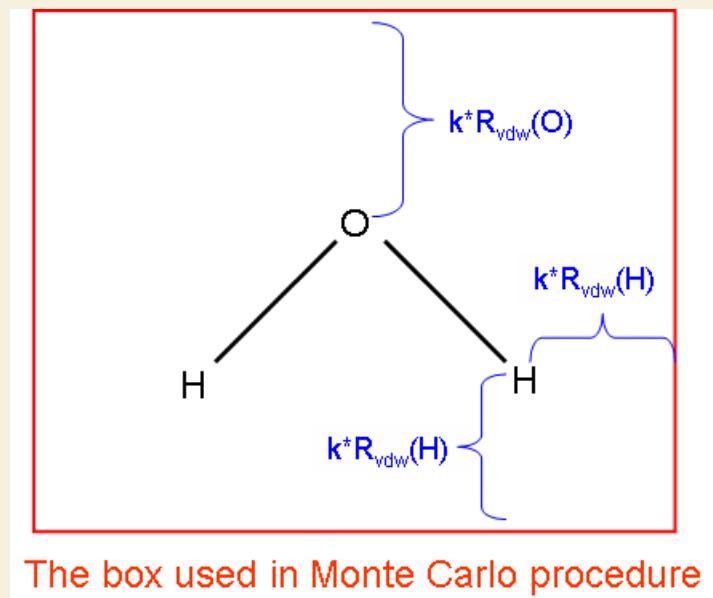
If current input file is .chg format (see Section 2.5 for details), you can use option 2 to convert it to .pqr file. The .pqr and .pdb formats are very similar, the major differences is that the former has additional two columns to record atomic charges and atomic radii. In the resulting .pqr file, the atomic charges are identical to that in .chg file, while the atomic radii column corresponds to Bondi vdW radii. The .pqr can be directly loaded into VMD program, the atoms can be colored according to atomic charges (if the fourth column of .chg file records other atomic information such as atomic spin populations, in VMD the atoms can also be colored according to spin populations). This is very useful for intuitively exhibiting atomic properties, see Section 4.A.10 for illustrations.

Information needed: Atom coordinates, GTFs (for exporting .wfn/.wfx), basis functions (for exporting .fch/.molden/.47)

3.100.3 Calculate molecular van der Waals volume

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 "MCvommethod" 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.



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

3.100.4 Integrate a function in whole space

This is a very useful and powerful function for integrating selected real space function in the whole space. The numerical integration method used here is based on the one proposed by Becke in the paper *J. Chem. Phys.*, **88**, 2547 for integrating DFT functional, which is also suitable for any real space function, but notice that the function must be smooth and converges to zero at infinite asymptotically. The accuracy is determined by the number of integration points, you can adjust the number of radial and angular integration points by “radpot” and “sphpot” parameters in settings.ini, respectively. The integrand can be selected from built-in functions, and you can also write new function by yourself as user-defined function, see Sections 2.6 and 2.7.

Exmination of difference of a real space function between two wavefunction files

By the way, if you would like to examine difference between two wavefunction files for a real space function, you can select function -4 or -5 (hidden functions) in main function 100, the function to be integrated will be $[f(file1)-f(file2)]^2$ or $|f(file1)-f(file2)|$, respectively, where f is the real space function you will select, file1 is the wavefunction file loaded when Multiwfn boots up, and file2 is the wavefunction file you will choose in this function. In this function you can also set criterion of electron density and thus let Multiwfn only evaluate the difference for low density region (*i.e.* ignoring core region). In addition, after calculation, Multiwfn automatically exports grid data of file1 as a plain text file in current file, the file name directly reflects current calculation condition. For example, the file name *a.wfn_003_0075_0434* implies that file1 is *a.wfn*, the 3rd real space function was selected, the radial and angular integration points are 75 and 434, respectively. In later studies, if all calculation conditions match with the file name (the file must be placed in current folder), then Multiwfn will directly load data of file1 from this file rather than recalculate them to reduce computational time.

Evaluation of spherically symmetric average ELF / LOL

In *J. Comput. Chem.*, **38**, 2258 (2017), the authors proposed that the optimal ω parameter of range-separated DFT functionals can be determined by means of below quantity:

$$r_{\text{ELF}} = \sqrt{\frac{\int \text{ELF}(\mathbf{r}) \mathbf{r}^2 \text{ELF}(\mathbf{r}) d\mathbf{r}}{\int \text{ELF}(\mathbf{r}) \text{ELF}(\mathbf{r}) d\mathbf{r}}}$$

In Multiwfn, this quantity will be automatically calculated and outputted if you select ELF as the integrand in present function. The r_{ELF} is outputted as “spherically symmetric average ELF”, the numerator and denominator in the root sign are also outputted together.

In *J. Phys. Chem. C*, **123**, 4407 (2019), the LOL-tuning is proposed, in which the “spherically symmetric average LOL” is involved:

$$r_{\text{LOL}} = \sqrt{\frac{\int \text{LOL}(\mathbf{r}) \mathbf{r}^2 \text{LOL}(\mathbf{r}) d\mathbf{r}}{\int \text{LOL}(\mathbf{r}) \text{LOL}(\mathbf{r}) d\mathbf{r}}}$$

The r_{LOL} along with its numerator and denominator in the root sign will be printed if LOL is selected in present function.

Information needed: Atom coordinates, GTFs.

3.100.5 Show overlap integral between alpha and beta orbitals

For unrestricted wavefunctions, orthonormalization condition does not in general hold 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 indicates that spin polarization is remarkable.

In present function, there are two options, option 1 calculates the full overlap matrix, the diagonal elements will be printed on screen and the whole overlap matrix can be selected to output to *ovlpmat.txt* in current folder; in addition, the maximum pairing between Alpha and Beta orbitals are shown. This calculation may be time-consuming for large system. Option 2 only calculates and prints the diagonal elements, this is always fast.

Since the expectation of S^2 operator for single determinant (SD) wavefunction can be easily derived from the overlap matrix, if option 1 is selected, Multiwfn also outputs this quantity:

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

where N^α and N^β are the number of alpha and beta electrons. $\langle S^2 \rangle_{\text{Exact}}$ is the exact value of square of total spin angular momentum

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

Information needed: GTFs

3.100.6 Monitor SCF convergence process of Gaussian

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 information of SCF process will be recorded in output file.

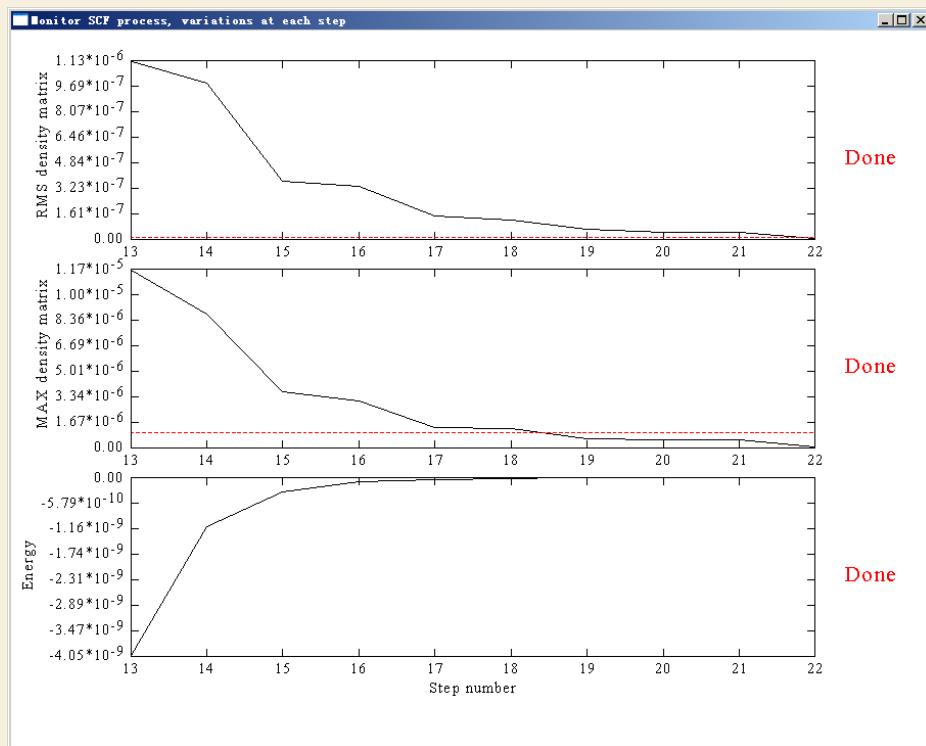
When you entered this function, all information of previous steps and the thresholds of convergence are printed on screen, such as

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

In the graph, gray dashed line shows the zero position of Y-axis, the red dashed 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

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

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

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 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

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

where E_{els} is electrostatic interaction term, normally negative if the two fragments are neutral; E_{ex} is exchange repulsion term, which comes from the Pauli repulsion effect and is invariably positive. For convenience, it is customary to combine these two terms as steric term (E_{steric}).

E_{orb} in above formula is orbital interaction term, and sometimes also known as induction term or polarization term. E_{orb} arises from the mix of occupied MOs and virtual MOs. If the combined wavefunction is used as initial guess for complex, then E_{orb} can be evaluated by subtracting the first SCF iteration energy from the last SCF iteration energy.

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

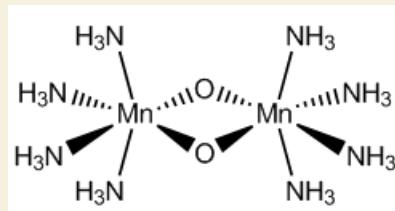
Note that $E_{\text{SCF, last}} = E^{\text{complex}}$, obviously we can write out below relationship

$$\Delta E_{\text{steric}} = \Delta E_{\text{els}} + \Delta E_{\text{ex}} = \Delta E_{\text{tot}} - \Delta E_{\text{orb}} = E_{\text{SCF, 1st}} - \sum_i E_i^{\text{frag}}$$

By the way, if the complex you studied concerns evident dispersion interaction (vdW interaction), there are two possible ways to evaluate the dispersion energy component:

- (1) Use Hartree-Fock to calculate interaction energy first ($\Delta E_{\text{tot}}^{\text{HF}}$), then use MP2 (or better post-HF method) to calculate interaction energy again ($\Delta E_{\text{tot}}^{\text{MP2}}$), then $\Delta E_{\text{disp}} = \Delta E_{\text{tot}}^{\text{MP2}} - \Delta E_{\text{tot}}^{\text{HF}}$. This relationship comes from the fact that dispersion energy is completely missing in HF energy.
- (2) Use HF or the DFT functionals that completely failed to represent dispersion energy to calculate interaction energy (e.g. B3LYP and BLYP), then use Grimme's DFT-D3 program (<http://www.thch.uni-bonn.de/tc/index.php?section=downloads&>) with corresponding parameter to evaluate DFT-D3 dispersion correlation to interaction energy, which can be simply regarded as the dispersion component in total interaction energy. If you do not know how to do this and you can read Chinese, you may consult the post in my blog: <http://sobereva.com/210>

3 Modelling antiferromagnetic coupling system: I exemplify this concept and show you how to use the function by a representative antiferromagnetic coupling system -- $\text{Mn}_2\text{O}_2(\text{NH}_3)_8$,



The ground state is singlet, while the two Mn atoms have opposite spin and each Mn atom has high spin. Obviously, restricted closed-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 closed-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(NH₃)₄ at left side. Charge=+2, sextet.
- Fragment 2: Mn(NH₃)₄ at 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 calculation 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 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

```
! Alpha 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).

If you used diffuse functions and encounter problem at Link401 when running .new by Gaussian, you can add IOp(3/32=2) keyword and retry.

3.100.9 Evaluate interatomic connectivity and atomic coordination number

In the original paper of DFT-D3 (*J. Chem. Phys.*, **132**, 154104 (2010)), the authors argued that the coordination number (*CN*) of an atom *A* can be approximately expressed as

$$\text{CN}_A = \sum_{B \neq A} \frac{1}{1 + \exp \left\{ -16 \times [(4/3)(R_A + R_B)/r_{AB} - 1] \right\}}$$

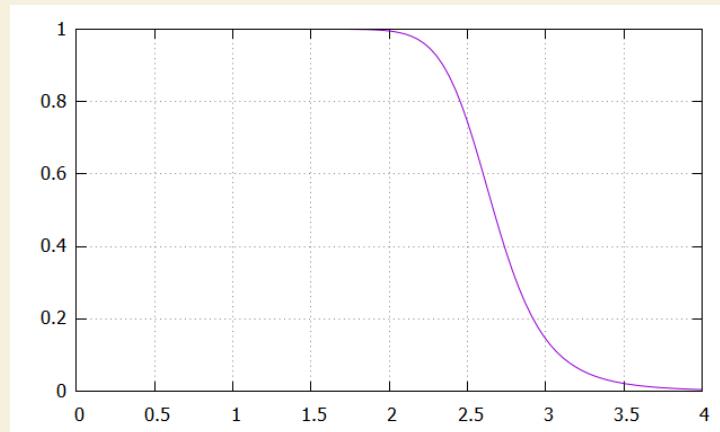
where *R* is Pyykkö covalent radius from *Chem. Eur. J.*, **15**, 186 (2009), and *r_{AB}* is distance between *A* and *B*.

3 Functions

According this idea, in present module the interatomic connectivity index (I) between A and B is determined as follows:

$$I_{AB}(r_{AB}) = \frac{1}{1 + \exp\{-16 \times [(4/3)(R_A + R_B)/r_{AB} - 1]\}}$$

Assume that $R_A+R_B=2.0$, then the function could be plotted as follows. The I value equals to 0.995 when $r_{AB}=R_A+R_B$



Note that the I should not be utilized as an indicator of bond order, it does not have capability of discriminating bonding type and strength.

Present module outputs I between each pair of atoms, the printing threshold can be inputted by user. Commonly, when I is close to 1.0, it implies that the two atoms are bonded, while if it is close to 0.0, then they may be regarded as not binded by chemical bond. The nearest integer of I , namely $\text{nint}(I)$, is also outputted for facilitating examination of the result.

For each atom (e.g. atom A), the $\sum_{B \neq A} I_{AB}$ is printed as "Sum of connectivity", while $\sum_{B \neq A} \text{nint}(I_{AB})$ is printed as "Sum of integer connectivity". The former and the latter may be regarded as raw and actual coordination number, respectively.

Finally, you can choose if exporting all I values as matrix to *connmat.txt* in current folder.

Below is output example of ethyne at equilibrium geometry:

```

1C --- 2C : 0.99951 Nearest integer: 1
1C --- 3H : 0.99974 Nearest integer: 1
2C --- 4H : 0.99974 Nearest integer: 1

1 C Sum of connectivity: 2.0016 Sum of integer connectivity: 2
2 C Sum of connectivity: 2.0016 Sum of integer connectivity: 2
3 H Sum of connectivity: 1.0021 Sum of integer connectivity: 1
4 H Sum of connectivity: 1.0021 Sum of integer connectivity: 1

```

Information needed: atom coordinates

3.100.11 Calculate overlap and centroid distance between two orbitals

This function is used to calculate overlap and centroid distance between two orbitals, this is useful for many purposes, e.g. analyzing charge transfer during electron excitation. You need to input index of two orbitals, then X, Y, Z of centroid of the orbitals will be calculated as follows

$$X_i = \int |\phi_i(\mathbf{r})|^2 x d\mathbf{r} \quad Y_i = \int |\phi_i(\mathbf{r})|^2 y d\mathbf{r} \quad Z_i = \int |\phi_i(\mathbf{r})|^2 z d\mathbf{r}$$

then the centroid distance between orbital i and j is calculated as

$$D_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2}$$

Present function also calculates overlap degree of the two orbitals, below two quantities are calculated respectively (while directly calculating overlap integral of two orbital wavefunctions is clearly meaningless, since it must be zero due to orthonormalization condition):

$$\int |\phi_i(\mathbf{r})| |\phi_j(\mathbf{r})| d\mathbf{r} \quad \int |\phi_i(\mathbf{r})|^2 |\phi_j(\mathbf{r})|^2 d\mathbf{r}$$

The integrals shown above are not calculated analytically but numerically via Becke's grid-based integration approach. The integration grid can be set by "radpot" and "sphpot" in settings.ini, the default values are high enough, and you may want to somewhat decrease them to reduced computational cost for large system, especially when you want to study many orbital pairs.

When the calculation is finished, Multiwfn will ask you whether or not add the two centroids as two additional dummy atoms (the symbol is Bq). If you choose y, then you can go to main function 0 to visualize corresponding orbital isosurfaces by transparent or mesh style to examine correspondence between centroid position and orbital shape.

Information needed: GTFs, atom coordinates

3.100.12 Perform biorthogonalization between alpha and beta orbitals

Introduction

It is well known that for wavefunctions generated by unrestricted open-shell calculations (UHF or UKS), the alpha and beta orbitals are often evidently mismatch with each other, this phenomenon makes analysis of orbitals difficult, because one must simultaneously consider two set of orbitals. Although restricted open-shell (ROHF or ROKS) calculation does not have this problem, the total electronic energy, orbital energy and electron distribution is not as accurate as unrestricted open-shell calculation.

Present function is used to perform biorthogonalization between alpha and beta orbitals for unrestricted open-shell wavefunction with spin multiplicity higher than 1. Original alpha and beta molecular orbitals will be respectively transformed to a set of new orbitals. Although finally there are still two sets of orbitals, their wavefunctions have matched with each other almost perfectly, therefore then you only need to discuss one set of orbitals.

The so-called biorthogonalization mentioned here specifically refers to simultaneously satisfying two conditions: (1) For each set of spin orbitals, they are orthonormal with themselves (2) Alpha orbitals are orthonormal with beta orbitals. Without applying the biorthogonalization, the UHF/UKS orbitals only satisfy the first condition.

Algorithm details

The biorthogonalization is realized via singular value decomposition (SVD) technique. The overlap integral matrix \mathbf{O} between alpha and beta orbitals is first constructed, and then SVD is applied to decompose it as $\mathbf{O}=\mathbf{U}\Sigma\mathbf{V}^\dagger$, where Σ is a diagonal matrix, the diagonal elements are referred to as "singular values", which essentially correspond to the overlap integrals between the orbitals after the biorthogonalization transformation, under normal situations they should be very close to 1.0. The column matrix \mathbf{U} (\mathbf{V}) corresponds to the transformation matrix between the original orbitals and the new orbitals of alpha (beta) spin. The coefficient matrix of the newly generated biorthogonalized orbitals can be obtained as

$$\begin{aligned} C_{\text{biortho}}^\alpha &= \mathbf{U}\mathbf{C}_{\text{original}}^\alpha \\ C_{\text{biortho}}^\beta &= \mathbf{V}\mathbf{C}_{\text{original}}^\beta \end{aligned}$$

Since \mathbf{U} and \mathbf{V} are unitary matrices, such a transformation does not affect observable quantities of current system.

Notice that the biorthogonalization transformation in fact is not done for all orbitals at once, because this will lead to mix between occupied and virtual orbitals and thus result in change of observable properties. In Multiwfn, the transformation is successively carried out via below three steps. The total number of orbitals of each spin will be denoted as n_{tot} , the numbers of alpha and beta electrons will be denoted as n_α and n_β , respectively. $n_\alpha > n_\beta$ is assumed.

- (1) Biorthogonalization between all occupied alpha orbitals ($1 \sim n_\alpha$) and all occupied beta orbitals ($1 \sim n_\beta$). This step makes each resulting occupied beta orbital paired with a resulting alpha orbital.
- (2) Biorthogonalization between alpha orbitals ($n_\beta+1 \sim n_\alpha$) and all virtual beta orbitals ($n_\beta+1 \sim n_{\text{tot}}$). This step makes each resulting "singly occupied" alpha orbital paired with a resulting beta virtual orbital
- (3) Biorthogonalization between all alpha virtual orbitals ($n_\alpha+1 \sim n_{\text{tot}}$) and the virtual beta orbitals that have not been paired ($n_\alpha+1 \sim n_{\text{tot}}$).

After these three steps of tranformation, one-to-one pairing between all alpha and beta orbitals should be nearly perfectly satisfied, difference of orbital wavefunction distribution of alpha orbital i and the beta orbital with the same index should be negligible. Note that in biorthogonalization steps 2 and 3, the utilized overlap integral matrix \mathbf{O} should be reconstructed based on the coefficient matrices updated at the last step.

The biorthogonalized orbitals are not eigenfunctions of Fock operator (or Kohn-Sham operator, similarly hereinafter) as molecular orbitals, however their energies can be evaluated as expectation value of Fock operator. Specifically, if you request Multiwfn to evaluate the orbital energies, Multiwfn performs below representation transform:

$$\mathbf{F}_{\text{biortho}} = \mathbf{C}^T \mathbf{F}_{\text{AO}} \mathbf{C}$$

where \mathbf{F}_{AO} is the Fock matrix in original basis functions that loaded from external file, $C(\mu, i)$ corresponds to coefficient of basis function μ in biorthogonalized orbital i . Energy of biorthogonalized orbital j is simply $F_{\text{biortho}}(j, j)$. Once generation of orbital energies has done, Multiwfn is able to order the orbitals according to their energies. Notice that the energy used in the ordering process is average of energy of alpha orbital and its beta counterpart, and the three batch of orbitals ($1 \sim n_\alpha$), ($n_\beta+1 \sim n_\alpha$) and ($n_\alpha+1 \sim n_{\text{tot}}$) are ordered individually (hence for example, index of

an orbital originally in the second batch must still be higher than any orbital in the first batch after ordering).

Since the number of unoccupied MOs is generally much higher than the number of occupied MOs, while one often only has interest in occupied biorthogonalized orbitals, therefore Multiwfn provides an option to skip the biorthogonalization between unoccupied MOs, namely skipping the step (3) shown above. In this case the alpha and beta orbitals in the range of ($n_\alpha+1 \sim n_{\text{tot}}$) will be meaningless and you should not then study them.

Usage

After booting up Multiwfn, simply loading a file containing basis function information (e.g. .fch, .molden, .gms) that generated by UHF or UKS calculation, then go to subfunction 12 of main function 100, Multiwfn will biorthogonalize the alpha and beta orbitals.

Next, you can choose whether or not performing biorthogonalization between unoccupied MOs, as well as if generating energies of the biorthogonalized orbitals. For the last option, if you choose *y*, you then need to input path of a file containing Fock (or Kohn-Sham) matrix in original basis functions. The matrix elements should be provided in lower-triangular form, namely in this sequence: $F(1,1) F(2,1) F(2,2) F(3,1) F(3,2) F(3,3) \dots F(nbasis,nbasis)$, where *nbasis* is the total number of basis functions, the format is free. The Fock matrix may be obtained from output of some quantum chemistry codes (In fact, you can ask Gaussian to produce NBO .47 file, Multiwfn is able to automatically read the \$FOCK field when the file name has .47 suffix). After that, you can choose if ordering the biorthogonalized orbitals in the way mentioned earlier.

After that, two files are automatically exported to current folder

- *biortho.txt*: This file contains singular values and occupancy of the biorthogonalized orbitals. If energies of the orbitals have been generated, they will also be written into this file.

- *biortho.fch*: This file contains biorthogonalized alpha and beta orbitals. If you did not request Multiwfn to generate their energies, then in this file the “orbital energies” information will correspond to their singular values; while if their energies have been generated, then these energies will be written into this .fch file as “orbital energies” information.

Finally, if you input *y*, then the *biortho.fch* will be loaded directly, so that then you can directly visualize the newly generated orbitals via main function 0 or analyze them by various functions; if you input *n*, then the file that loaded when Multiwfn boots up will be reloaded to recover to the initial state.

Application of the biorthogonalization transformation for a practical system is illustrated in Section 4.100.12.

Information needed: Basis functions, plain text file containing Fock/KS matrix (optional)

3.100.13 Calculate HOMA and Bird aromaticity index

HOMA is one of the most popular indexes for measuring aromaticity. This quantity was originally proposed in *Tetrahedron Lett.*, **13**, 3839, and then the generalized form was given in *J. Chem. Inf. Comput. Sci.*, **33**, 70. The generalized HOMA can be written as (notice that the HOMA formula has been incorrectly cited by numerous literatures)

$$\text{HOMA} = 1 - \sum_i \frac{\alpha_{i,j}}{N} (\mathbf{R}_{\text{Ref}} - \mathbf{R}_{i,j})^2$$

where N is the total number of the atoms considered, j denotes the atom next to atom i , α and \mathbf{R}_{Ref} are pre-calculated constants given in original paper for each type of atom pair. If HOMA equals to 1, that means length of each bond is identical to optimal value \mathbf{R}_{Ref} and thus the ring is fully aromatic. While if HOMA equals to 0, that means the ring is completely nonaromatic. If HOMA is significant negative value, then the ring shows anti-aromaticity characteristic.

HOMA can be calculated by subfunction 13 in main function 100. The built-in α and \mathbf{R}_{Ref} parameters are directly taken from *Chem. Inf. Comput. Sci.*, **33**, 70, they can be adjusted customly by option 1. When you choose option 0, Multiwfn will prompt you to input the indices of the atoms in the local system, for example, 2,3,4,5,6,7 (assume that there are six atoms in the ring. The input order must be consistent with atom connectivity), then HOMA value and contributions from each atom pair will be immediately outputted on the screen. For example, thiophene optimized under MP2/6-311+G**, the output is

Atom pair	Contribution	Bond length(Angstrom)
1(C) -- 2(C):	-0.001852	1.382006
2(C) -- 3(C):	-0.056708	1.421170
3(C) -- 4(C):	-0.001852	1.382006
4(C) -- 5(S):	-0.023886	1.712627
5(S) -- 1(C):	-0.023886	1.712627

HOMA value is 0.891817

Since 0.891817 is close to 1, the HOMA analysis suggests that thiophene has relatively strong aromaticity.

Bird index (*Tetrahedron*, **41**, 1409) is another geometry-based quantity aimed at measuring aromaticity, and can be calculated by option 2 of subfunction 13 in main function 100. The formula is

$$I = 100[1 - (V / V_K)]$$

where

$$V = \frac{100}{\bar{N}} \sqrt{\frac{\sum_i (N_{i,j} - \bar{N})^2}{n}} \quad N_{i,j} = \frac{a}{R_{i,j}} - b$$

In the formula, i cycles all of the bonds in the ring, j denotes the atom next to atom i . n is the total number of the bonds considered. N denotes Gordy bond order, \bar{N} is the average value of the N values. $R_{i,j}$ is bond length. a and b are predefined parameters respectively for each type of bonds. V_K is pre-determined reference V , for five and six-membered rings the value is 35 and 33.2, respectively. The more the Bird index close to 100, the stronger the aromaticity is.

Available a and b include C-C, C-N, C-O, C-S, N-O and N-N; for other type of bonds user should set corresponding parameter by option 3. By option 4 user can adjust or add V_K parameter.

Corresponding example of this function is provided in Section 4.100.13.

Information needed: Atom coordinates

3.100.14 Calculate LOLIPOP (LOL Integrated Pi Over Plane)

In the paper *Chem. Commun.*, **48**, 9239 (2012), the authors proposed a quantity named LOLIPOP (Localized Orbital Locator Integrated Pi Over Plane) to measure π -stacking ability of aromatic systems, they argued that a ring with smaller LOLIPOP value has stronger π -depletion (namely lower π -delocalization), and hence shows stronger π -stacking ability.

LOLIPOP is defined as definite integral of LOL- π (the LOL purely contributed by π -orbitals) from a distance of 0.5Å away from the molecular plane. Only points with $\text{LOL-}\pi>0.55$ are taken into account. The integration is made in a cylindrical region perpendicular to the molecular plane, with a radius of 1.94Å corresponding to the average between the C and H ring radii in benzene.

In Multiwfn, function 14 in main function 100 is designed for calculating LOLIPOP, the default value 0.5Å and 1.94Å mentioned above can be changed by option 4 and 3 respectively. Before starting the calculation, you have to choose which orbitals are π orbitals by option 3, you can find out π orbitals by visualizing orbital isosurfaces via main function 0. The calculation can be triggered by option 0, you need to input the indices of the atoms in the ring in accordance with the atom connectivity, then grid data of LOL around the ring will be evaluated, then the LOLIPOP value is computed by numerical integration. Of course, smaller grid spacing gives rise to higher integration accuracy, but brings severer computational requirement on evaluating LOL. The default spacing of grid data is 0.08 Bohr, this value is fine enough in general.

Corresponding example of this function is provided in Section 4.100.14.

Information needed: GTFs, atom coordinates

3.100.15 Calculate intermolecular orbital overlap

This function is used to calculate orbital overlap integral between two molecules, namely

$$S_{i,l}^{\text{intmol}} = \int \varphi_i^{\text{monomer1}}(\mathbf{r}) \varphi_l^{\text{monomer2}}(\mathbf{r}) d\mathbf{r}$$

where i and l are molecular orbital indices of monomer 1 and monomer 2, respectively. This integral is useful in discussions of intermolecular charge transfer, *e.g.* *J. Phys. Chem. B*, **106**, 2093.

Three files are required for evaluating the integral:

(1) Gaussian output file of dimer, *IOp(3/33=1) nosymm guess=only* should be specified in route section. Multiwfn will read overlap matrix from this file.

(2) Gaussian output file of monomer 1, *nosymm pop=full* should be specified in route section. Multiwfn will read orbital coefficients of monomer 1 from this file.

(3) Gaussian output file of monomer 2, *nosymm pop=full* should be specified in route section. Multiwfn will read orbital coefficients of monomer 2 from this file.

Notice that, the atomic coordinates in file (2) and (3) must be in accordance with the one in file

(1). The basis set used should be the same for the three calculations. The atomic sequence should be identical. Molecular geometry must be given in Cartesian coordinate.

To calculate the intermolecular orbital overlap integral, after booting up Multiwfn, the file (1) should be loaded first. After entering present module, the paths of file (2) and (3) should be inputted in turn. Then one can input such as 8,15 to obtain $S_{8,15}^{\text{Intmol}}$. If user input the letter *o*, then the entire S^{Intmol} matrix will be outputted to "ovlpint.txt" in current folder.

Corresponding example of this function is provided in Section 4.100.15.

Information needed: Gaussian output files

3.100.18 Yoshizawa's electron transport route analysis

Theory

This function is used to analyze electron transport route and is mainly based on Yoshizawa's formula (*Acc. Chem. Res.*, **45**, 1612 (2012)). At the Fermi energy, the matrix elements of the zeroth Green's function, $G_{rs}^{(0)\text{R/A}}$, which describes the propagation of a tunneling electron from site *r* to site *s* through the orbitals in a molecular part, can be written as follows:

$$G_{rs}^{(0)\text{R/A}}(E_F) = \sum_k \frac{C_{rk} C_{sk}^*}{E_F - \varepsilon_k \pm i\eta}$$

where C_{rk} is the *k*th MO coefficient at site *r*, asterisk on the MO coefficient indicates a complex conjugate, ε_k is the *k*th MO energy, and η is an infinitesimal number determined by a relationship between the local density of states and the imaginary part of Green's function. By this formula, if two electrodes are connected to site *r* and *s*, by this formula we can readily predict the transmission probability between the two sites. From the expression of the denominator, it can be seen that HOMO and LUMO have the largest contribution to *G*. If only HOMO and LUMO are taken into account, the formula can be explicitly written as

$$G_{rs}^{(0)\text{R/A}}(E_F) = \frac{C_{r\text{HOMO}} C_{s\text{HOMO}}^*}{E_F - \varepsilon_{\text{HOMO}} \pm i\eta} + \frac{C_{r\text{LUMO}} C_{s\text{LUMO}}^*}{E_F - \varepsilon_{\text{LUMO}} \pm i\eta}$$

This formula allows one to visually examine the possibility of electron transmission between site *r* and *s* by means of observing molecular orbital diagram. If both site *r* and *s* have large *C* in magnitude, and their relative phase in HOMO and in LUMO is different, then the magnitude of *G* will be large, indicating that transmission between *r* and *s* will be favourable. For examples, see *Acc. Chem. Res.*, **45**, 1612 (2012). Note that *G* may be positive and negative, but only absolute value of *G* is meaningful for discussing transmission.

In Multiwfn, $i\eta$ term in *G* is always ignored, since this is an infinitesimal number. The coefficient *C* comes from the output of "NAOMO" keyword in NBO program. In almost all organic conductors, σ orbitals have little contribution to conductance, therefore present function only takes π orbitals into account, and the molecular plane must be parallel in XY or YZ or XZ plane.

Input file

To run this kind of analysis, there are two choices on the input file

- (1) Only using Gaussian output file

In the route section, `pop=nboread` must be specified, and `$NBO NAOMO $END` must be written at the last line, for example:

```
#P b3lyp/6-31g* pop=nboread

b3lyp/6-31g* opted

0 1
[Molecular geometry field]

$NBO NAOMO $END
```

- (2) Load a file containing basis function information when Multiwfn boots up, then after entering present function, load a NBO output file containing NAOMO information (namely the `NAOMO` keyword has been passed to NBO program. If you are using GENNBO, you can add `NAOMO` into `$NBO...$END` field of .47 file).

Usage

After you enter this function, you need to select which plane is the one your molecular plane parallel to. Then program will load file and find out the atoms having expected π atomic orbital (restrictly speaking, the "Val"-type p natural atomic orbitals that perpendicular to the chosen plane). The coefficient of these atomic orbitals will be used to compute G .

Then you will see a menu, the options are explained below:

-4 Set distance criterion: Input lower and upper limits. Then in option 2 and 3, the route whose distance exceeds this criterion will not be shown.

-3 Set value criterion: In option 1, the MO whose contribution to G is less than this criterion will not be shown. In option 2 and 3, the route whose $|G|$ is smaller than this criterion will not be shown.

-2 Set Fermi energy level: Namely set E_F in the Yoshizawa's formula.

-1 Select the range of MOs to be considered: Namely set the MO range of the summation in the Yoshizawa's formula.

0 View molecular structure: As the title says.

1 Output detail of electron transport probability between two atoms: You need to input index for two atoms, they will be regarded as site r and s , then G_{rs} (the value behind "Total value") will be outputted; meanwhile program also outputs the contribution from each MO, the distance between the two atoms, and the calculated G_{rs} for the case when only HOMO and LUMO are taken into the summation.

2 Output and rank all electron transport routes in the system: All routes in current system will be tested, if the route simultaneously fulfills the G and distance criteria set by -3 and -4, then the involved atoms, G and distance of the route will be printed. The routes are ranked by absolute value of G .

3 Output and rank all electron transport routes for an atom: Similar to option 2, but only

consider the routes involving specific atom.

Information needed: As mentioned above

3.100.19 Generate promolecular .wfn file from fragment wavefunctions

This function is used to generate promolecular wavefunction in .wfn format based on fragment wavefunctions. In the promolecular state, the electron distribution simply comes from the superposition of the electron distribution of the related fragments; in other words, in this state the electron transfer and polarization between the fragments have not started. Each promolecular orbital directly corresponds to one of fragment orbitals.

The input file can be in any wavefunction format, such as .wfn, .wfx, .fch, .molden and .gms, any kind of wavefunction is supported. Infinite number of fragments are supported.

Assume that the promolecular wavefunction consists of N fragments, after boot up Multiwfn, you should make Multiwfn load the first fragment wavefunction file, and then go to this function, input the total number of fragments (N), then input the path of the wavefunction files corresponding to the other fragments in turn. The promolecule .wfn file will be generated in current folder as "promol.wfn".

For SCF wavefunctions, the MOs are sorted according to orbital energies from low to high, virtual MOs are not recorded. For post-HF wavefunctions, the natural orbitals are sorted according to orbital occupation from high to low.

If any fragment is unrestricted open-shell, then the promolecular wavefunction will be regarded as unrestricted open-shell type, namely the alpha and beta orbitals will separately occur in the resultant .wfn file. In this case, for each unrestricted open-shell fragment Multiwfn will ask you if flipping the spin of its orbitals; if you choose yes, then the energies and occupations of all alpha and all beta orbitals will be exchanged; this treatment is particularly important if you want to get the promolecular wavefunction combined from free-radical fragments.

Please always carefully check the resultant .wfn file to examine its reasonableness before using it.

Some examples are given in Section 4.100.19.

Information needed: GTFs, atom coordinates

3.100.20 Calculate Hellmann-Feynman forces

The Hellmann-Feynman (H-F) force is the actual force acting on nuclei in a quantum system, which is exerted by electron density and other nuclei:

$$\mathbf{F}_A = -\left\langle \Psi \left| \frac{\hat{H}}{\partial \mathbf{R}_A} \right| \Psi \right\rangle = \mathbf{F}_A^{\text{ele}} + \mathbf{F}_A^{\text{nuc}} = -Z_A \int \frac{\rho(\mathbf{r})(\mathbf{R}_A - \mathbf{r})}{|\mathbf{R}_A - \mathbf{r}|^3} d\mathbf{r} + Z_A \sum_{B \neq A} \frac{Z_B (\mathbf{R}_A - \mathbf{R}_B)}{|\mathbf{R}_A - \mathbf{R}_B|^3}$$

Present function calculates and prints total H-F force as well as the contribution from electron density and other nuclei respectively.

In practice, notice that the H-F forces calculated as above based on the electronic wavefunction

and nuclear information recorded in the wavefunction file are generally not equivalent to the forces computed by quantum chemistry program at current calculation level (\mathbf{F}^{QC}), which may be expressed as:

$$\mathbf{F}_A^{\text{QC}} = -\frac{\partial E}{\partial \mathbf{R}_A} = -\langle \Psi | \frac{\hat{H}}{\partial \mathbf{R}_A} | \Psi \rangle - 2 \langle \partial \Psi / \partial \mathbf{R}_A | \hat{H} | \Psi \rangle$$

The partial derivative of wavefunction with respect to nuclear coordinate involves partial derivative of orbital coefficients, configuration state coefficients and basis functions with respect to \mathbf{R}_A . Therefore, only for the fully variational wavefunctions such as HF, DFT and MCSCF with basis functions independent of nuclear coordinates (e.g. plane wave), the second term on the r.h.s. of above equation is vanishing, and then the H-F forces just equals to the force at current calculation level.

Information needed: GTFs, atom coordinates

3.100.21 Calculate properties based on geometry information for specific atoms

Input index range of some atoms (or input *all* to select the whole system), then many properties based on geometry information will be calculated for them, including

- (1) Mass
- (2) Geometry center, center of mass, center of nuclear charges
- (3) Radius of gyration
- (4) Sum of nuclear charges and dipole moments from nuclear charges
- (5) The atom having minimum/maximum coordinate in X/Y/Z
- (6) Minimum and maximum distance
- (7) Moments of inertia tensor, principal axes and principal moments of inertia
- (8) Rotational constant
- (9) Electrostatic interaction energy between nuclear charges

Note: If the input file is .chg, then "nuclear charges" mentioned above will stand for atomic charges.

Radius of gyration is computed as $\sqrt{\frac{\sum_i m_i (\mathbf{r}_i - \mathbf{r}_c)^2}{\sum_j m_j}}$, where \mathbf{r} is the coordinate of nucleus,

\mathbf{r}_c is mass center, m is atomic mass.

The moments of inertia tensor, which is a symmetry matrix, is calculated as

$$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} \sum_i m_i (y_i^2 + z_i^2) & -\sum_i m_i x_i y_i & -\sum_i m_i x_i z_i \\ -\sum_i m_i y_i x_i & \sum_i m_i (x_i^2 + z_i^2) & -\sum_i m_i y_i z_i \\ -\sum_i m_i z_i x_i & -\sum_i m_i z_i y_i & \sum_i m_i (x_i^2 + y_i^2) \end{bmatrix}$$

In the output, "The moments of inertia relative to X,Y,Z axes" correspond to the three diagonal terms of \mathbf{I} .

The three eigenvectors of \mathbf{I} are known as principal axes, they are outputted as the three columns of the matrix with the title "***** Principal axes (each column vector) *****". The corresponding three eigenvalues of \mathbf{I} are the moments of inertia relative to principal axes.

If the moments of inertia are given in amu $\cdot\text{\AA}^2$, then the rotational constants in GHz can be directly evaluated; for example, the rotational constant relative to Z axis is $\frac{h}{8\pi^2 I_{ZZ}\zeta} 10^{11}$, where h is Planck constant, and $\zeta=1.66053878 \cdot 10^{-27}$ is a factor used to convert amu to kg.

If you input `size` in this function, the system will be properly rotated to make its three principal axes respectively parallel to the three Cartesian axes. Then, according to position of boundary atoms and vdW radii proposed by Bondi (*J. Phys. Chem.*, **68**, 441 (1964)), the radius, diameter, length, width and height of present system are calculated and printed. The length, width and height can be visualized as a box enclosing the system in a GUI window, and the system can be exported as new.pdb file in current folder, which can be further visualized in VMD program. An example is provided in Section 4.100.21.

Information needed: Atom coordinates

3.100.22 Detect π orbitals and set occupation numbers

Sometimes one wants to analyze π properties of a conjugated system, *e.g.* calculating ELF- π , hence one must first find out which ones are π orbitals by visualizing orbital isosurfaces, and then set occupation number of all the other orbitals to zero. Unfortunately, for large system this work is very tedious and time-consuming. To make your life easier, this function is developed to automatically identify π orbitals, and then set their occupation numbers. There are two cases:

(1) The input file contains non-localized orbitals (*e.g.* canonical orbital, natural orbital, etc.): In this case, present function is only applicable to **exactly** planar system, namely in the input file all atoms are exactly in the same Cartesian plane XY, YZ or XZ. You can use options 1, 2 and 3 to choose Cartesian plane according to actual situation, or you can directly choose 0 to let program automatically determine the plane. Then π orbitals will be detected automatically.

(2) The input file contains localized molecular orbitals (LMOs): In this case, π orbitals are identified using following method (to be published): First, Multiwfn calculates orbital composition of present orbital (Mulliken method is used in present function), and finds out the atom having the largest contribution and the one having the second largest contribution (they will be referred to as atoms *A* and *B*, respectively). If contribution of *A* is larger than a given threshold (default is 85%), then the orbital will be regarded as single-center orbital and thus be skipped. If the orbital is not singly centered, occupation of this orbital will be temporarily set to 2.0 and electron density due to this orbital at $0.7\mathbf{R}_A+0.3\mathbf{R}_B$ and $0.3\mathbf{R}_A+0.7\mathbf{R}_B$ will be calculated, where \mathbf{R}_A and \mathbf{R}_B are coordinate vectors of *A* and *B*, respectively. If both the density are smaller than a given density threshold (default is 0.02 a.u.), the orbital will be finally regarded as π orbital. The reason of introduction of the density threshold is easy to understand: if a standard π orbital forms between two atoms, since

π orbital have a nodal plane along the bond, the electron density along two points in the linking line between the two atoms should be exactly zero. However, when the two atoms are not in a exactly planar local region, since σ and π orbitals are not strictly separable due to unavoidable mixing, the density at midpoint must not be exactly zero; therefore, a threshold should be employed to tolerate this circumstance.

The case (2) corresponding to option -1, after you select it, you will enter an interface, in which you can manually set the two above-mentioned thresholds. Also, in the interface you can set constraint of atom range by inputting atom indices. For example, if you inputted 2,4-7,9, then only the orbitals in which the two atoms with largest contributions are both in the range of 2,4,5,6,7,9 may be finally identified as π orbitals. Clearly, you can use this feature to identify all π type of LMOs lying at an interesting region, *e.g.* conjugated ring.

It is worth to note that only the LMOs yielded by Pipek-Mezey orbital localization method show separation character of σ and π and thus could be used in combination with present function. Although Multiwfn also supports Foster-Boys localization, the σ and π characters are mixed together and result in banana type of bonds, and thus the resulting LMOs cannot be used in conjunction with present function.

During identification process, indices, occupation numbers and energies of identified π orbitals are shown on screen. Then you can choose option 1 to set the occupation numbers of these π orbitals to zero, or choose option 2 to set the occupation number of all other orbitals to zero. Option 3 is akin to option 1, but only valence π orbitals are taken into account. Option 4 is used to clean occupation number for all orbitals except for valence π orbitals.

The use of this function is illustrated in Section 4.100.22. Section 4.5.3 also involves this function.

Information needed: GTFs (case 1), basis functions (case 2), atom coordinates

3.100.23 Fit function distribution to atomic value

This function is very similar to the function used to fit ESP charge (see Section 3.9.10 and 3.9.11), but the real space function to be fitted is not limited to ESP, for example you can fit average local ionization energy or even Fukui function distributed on molecular surface to atomic values.

After select option 1, and then select a real space function, the function value will be calculated on the fitting points and then fit to atomic value by least-squares method, namely minimizing below error function

$$F(p_1, p_2 \dots p_N) = \sum_i [V(\mathbf{r}_i) - V'(\mathbf{r}_i)]^2$$

where i denotes the index of fitting points, p is atomic value, V is the value of real space function, while V' is the function value evaluated by atomic value, which is defined as

$$V'(\mathbf{r}_i) = \sum_A \frac{p_A}{r_{A,i}}$$

where A denotes atom index, $r_{A,i}$ corresponds to the distance between nucleus of atom A and the fitting point i .

The error of fitting are measured by RMSE and RRMSE

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

where N denotes the number of fitting points.

By default all fitting centers are placed at nuclei, you can also load additional fitting centers from external file by option -2, see Section 3.9.10 for the format.

The default fitting points are exactly the ones used by Merz-Kollman method (see Section 3.9.11). You can use option 3 to customize the number of layers, use option 2 to adjust the density of the fitting points distributed on the surface, and use option 4 to set the scale factor of the vdW radii used for constructing each layer. (e.g. If you set two layers, and their scale factors are set to 1.0 and 1.2 respectively, then the first and the second layer will be constructed by superposing the vdW radii of all atoms multiplied by 1.0 and 1.2, respectively)

No constraint on the total value (viz. $\sum_i p_i$) is applied by default. However, you can use

option 5 to set a constraint on the total value. Evidently, if the constraint is set to the net charge of your system, and you choose ESP as fitting function, then the result will be identical to MK charges.

The fitting points can be directly defined via external plain text file by option -1 (and thus will not be constructed by superposing vdW spheres), see Section 3.9.10 for the file format. If the file contains calculated function values of all fitting points and meanwhile the first line of this file (the number of points) is set as negative value, then the real space function at each fitting point will not be evaluated by Multiwfn but loaded from this file directly.

Information needed: GTFs, atom coordinates

3.100.24 Obtain NICS_ZZ value for non-planar or tilted system

NICS is a very popular index used to measure aromaticity. In many literatures, such as *Org. Lett.*, **8**, 863, It was shown that NICS(0)_{ZZ} or NICS(1)_{ZZ} is a better index than the original definition of NICS, which is current known as NICS(0).

For exactly planar systems, if the system plane is parallel to XY plane, then NICS(0)_{ZZ} means the ZZ component of magnetic shielding tensor at ring center. The only different from NICS(1)_{ZZ} to NICS(0)_{ZZ} is that the calculated point is not ring center, but the point above (or below) 1 Å of the plane from ring center. Note that the definition of ring center is highly arbitrary, the original definition uses geometry center, while some people use center of mass, and some researchers recommend using ring critical point (RCP) of AIM theory as ring center, for example *WIREs Comput. Mol. Sci.*, **3**, 105. (Personally, I think using RCP is the best choice)

If the ring of interest is skewed, not exactly planar or tilted, calculation of NICS_{ZZ} is somewhat cumbersome, because one cannot directly acquire the component of magnetic shielding tensor perpendicular to the plane from output file of quantum chemistry programs. Moreover, for NICS(1)_{ZZ}, it is difficult to properly set the position to be calculated. Present function is designed to solve these difficulties.

In this function, the component of magnetic shielding tensor perpendicular to a given ring is

calculated as $\sigma_{\perp} = \mathbf{u}^T \boldsymbol{\sigma} \mathbf{u}$, where $\boldsymbol{\sigma}$ is magnetic shielding tensor, \mathbf{u} is column unit vector perpendicular to the ring, and \mathbf{u}^T is transpose of \mathbf{u} .

If you want to calculate NICS(1)_{ZZ} for a non-planar system, you can follow below steps:

(1) Use Multiwfn to open a file containing atomic coordinates of your system (e.g. .xyz/.pdb/.mol/.wfn/.wfx/.fch/.molden/.gms...)

(2) Find out ring center. You can use topology analysis module (main function 2) to locate RCP, or use subfunction 21 in main function 100 to obtain geometry center or center of mass. Using which method to define the ring center depends on yourself.

(3) Enter subfunction 24 of main function 100 (namely the function I am describing), input the ring center you just obtained, and input index of three atoms to define the ring plane. Then the coordinate of the points above and below 1 Å of the ring plane from the ring center will be outputted. You can use any one of these two points in quantum chemistry program to calculate magnetic shielding tensor at this position.

Hint: The unit normal vector perpendicular to the ring plane is also outputted by Multiwfn, by which you can easily derive the position used to calculate such as NICS(2), NICS(3.5)...

(4) Input all components of the magnetic shielding tensor in Multiwfn according to the output of your quantum chemistry program. Then the negative value of "The shielding value normal to the plane" outputted by Multiwfn is just NICS(1)_{ZZ}.

For calculating NICS(0)_{ZZ}, the process is more simple:

(1) Identical to the step 1 shown above

(2) Identical to the step 2 shown above

(3) Use the ring center you just obtained in your quantum chemistry program to calculate the magnetic shielding tensor at this position.

(4) Enter subfunction 24 of main function 100, input ring center, and input index of three atoms to define the ring plane. Then input all components of the magnetic shielding tensor according to the output of your quantum chemistry program. Then the negative value of "The shielding value normal to the plane" outputted by Multiwfn is just NICS(0)_{ZZ}.

Information needed: Atom coordinates

3.200 Other functions, part 2 (200)

3.200.1 Calculate core-valence bifurcation (CVB) index and related quantities

(1) Theory of CVB index

The idea of the so-called core-valence bifurcation (CVB) index was firstly proposed in *Theor. Chem. Acc.*, **104**, 13 (2000), this index was defined based on electron localization function (ELF) and mainly used to distinguish strength of various kind of hydrogen bonds (H-bonds). For a H-bond of typical form (D-H \cdots A, where D=donor, H=hydrogen, A=acceptor), this index is expressed as:

$$\text{CVB index} = \text{ELF(C-V)} - \text{ELF(DH-A)}$$

where ELF(C-V) corresponds to the ELF bifurcation value between ELF core domain and valence domain, while the ELF(DH-A) stands for the ELF value at bifurcation point between V(D,H) and V(A).

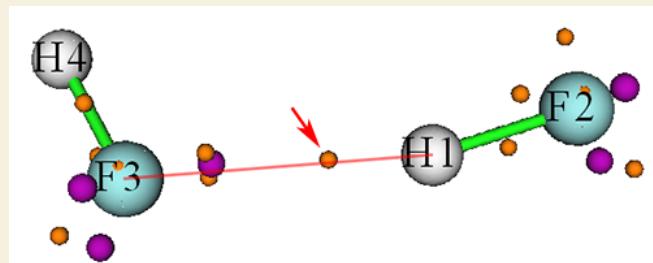
In the above-mentioned *Theor. Chem. Acc.* paper, the authors examined many H-bond dimers composing of HF and various kinds of monomers, it was found that the CVB index has good linear relationship with H-bond binding energy. In some succeeding papers, such as *Struct. Chem.*, **16**, 203 (2005) and *J. Phys. Chem. A*, **115**, 10078 (2011), this point has been further confirmed, and in the former it was pointed out at CVB index “is positive in the case of weak complexes and negative in stronger ones”. In addition, in *Chem. Rev.*, **111**, 2597 (2011) the author stated that CVB index “is positive for weak hydrogen bond, and it decreases if the strength of this interaction increases; usually this index is negative for strong hydrogen bonds”.

I found the ELF(C-V) and ELF(DH-A) themselves sometimes have better linear relationship with H-bond binding energy than CVB index, therefore I suggest you also examine this point in your practical studies when you intend to use CVB index.

(2) Manual evaluation of CVB index

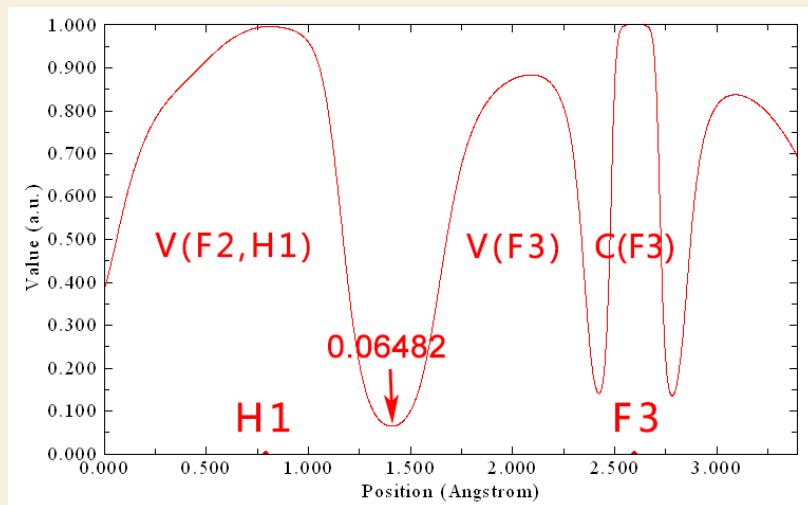
Below I will show how to manually evaluate the two terms involved in the CVB index. HF...HF dimer is taken as example, the wavefunction file is provided as *examples\HF_HF.wfn*, it was generated at B3LYP-D3(BJ)/def2-TZVP level, the optimization was also conducted at this level. The D, H, A atoms in this system correspond to F2, H1, F3, respectively

The ELF(DH-A) term is defined unambiguously in the original paper. The topology analysis module of Multiwfn is able to locate bifurcation points of ELF, namely (3,-1) critical points of ELF. Then you can check ELF value of the bifurcation point lying between the hydrogen and acceptor atom (Section 4.2.2 illustrated how to perform topology analysis for LOL. ELF can be analyzed in similar way). However, topology analysis of ELF is time-consuming for large system. Considering the fact that the actual ELF bifurcation point between V(D,H) and V(A) is almost exactly lying on the straight line linking H and A, it is better to use main function 3 of Multiwfn to plot a ELF curve map from the H to A and then directly read the value of corresponding minimum. Below is a screenshot of ELF topology analysis result for the HF...HF dimer



The purple and orange spheres are (3,-3) and (3,-1) type of ELF critical points (CPs), respectively. The (3,-1) CP pointed by the arrow corresponds to the aforementioned ELF bifurcation point between V(D,H) and V(A), its ELF value was found to be 0.06487, which is just the ELF(DH-A) of present system.

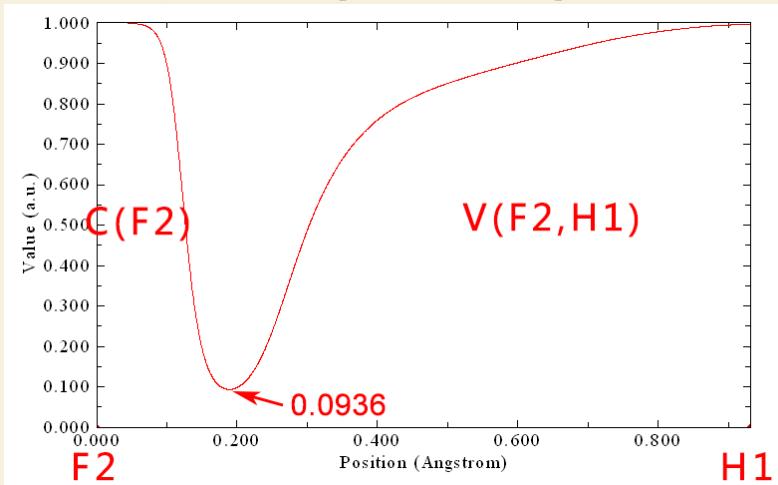
As can be seen from the above graph, the red linking line basically crosses the center of the orange sphere, this is why the ELF(DH-A) can also be approximately evaluated based on the ELF curve map between H1 and F3. The curve map plotted using main function 3 is shown below



The minimum highlighted by the arrow is 0.06482, which is very close to the value 0.06487 obtained based on the expensive ELF topology analysis. This observation well demonstrates the reasonableness of employing ELF curve map between H and A to estimate the ELF(DH-A).

As regards ELF(C-V), its definition is fairly ambiguous. Since in the original paper of CVB index the authors did not explicitly and clearly explain how this quantity should be evaluated, different papers often employ different rule to calculate it, leading to serious confusion in existing literatures. For example, in the CVB original paper, namely *Theor. Chem. Acc.*, **104**, 13 (2000), it seems that the ELF(C-V) was determined as maximal value at all ELF minima dissecting core and valence shell on the ELF curve between D and A atoms. However, in the subsequent paper *Struct. Chem.*, **16**, 203 (2005) written by the same author, I found the ELF(C-V) is seemingly calculated as the ELF value at one of exactly located ELF bifurcation points connecting core and valence basin of donor atom (while acceptor atom is seemingly ignored).

In my viewpoint, the best definition of ELF(C-V) should be the ELF value at the minimum dissecting core and valence shell of donor atom on the ELF curve between D and H. Again taking the HF...HF (H4-F3...H1-F2) dimer as example, the ELF curve plotted between F2 and H1 is:



Namely $\text{ELF}(\text{C-V}) = 0.0936$. Hence, the CVB index for the HF...HF system should be $0.0936 - 0.0648 = 0.0288$. This value is very different to the counterpart (-0.006) in Table 2 of *Theor. Chem. Acc.*, **104**, 13 (2000), because the calculation levels are different, the ways of obtaining ELF(C-V) are different, and the sign of the data in this paper was erroneously reversed.

(3) Calculating CVB index in full automatic way

In order to simplifying the calculation of CVB index in above mentioned way as much as possible, Multiwfn provides a function used to calculate this index in full automatic way. Still taking the HF...HF dimer as example, boot up Multiwfn and input below commands

examples\HF_HF.wfn

200 // Other function, part 2

1 // Calculate CVB index and related quantities

2,1,3 // Index of donor atom, hydrogen and acceptor atom of the H-bond, respectively

The result is

Core-valence bifurcation value at donor, ELF(C-V,D): 0.0936

Distance between corresponding minimum and the hydrogen: 0.743 Angstrom

Core-valence bifurcation value at acceptor, ELF(C-V,A): 0.1408

Distance between corresponding minimum and the hydrogen: 1.628 Angstrom

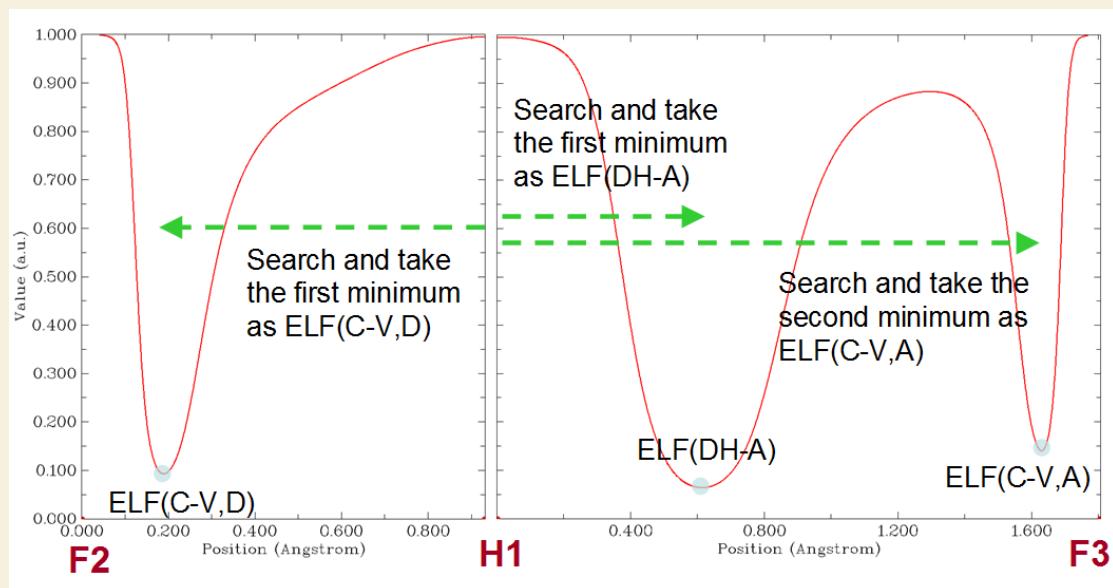
Bifurcation value at H-bond, ELF(DH-A): 0.0648

Distance between corresponding minimum and the hydrogen: 0.614 Angstrom

The CVB index, namely ELF(C-V,D) - ELF(DH-A): 0.028768

The result is completely identical to that we calculated manually. The outputted ELF(CV,A) is useless in current context, but some users may be interested in it.

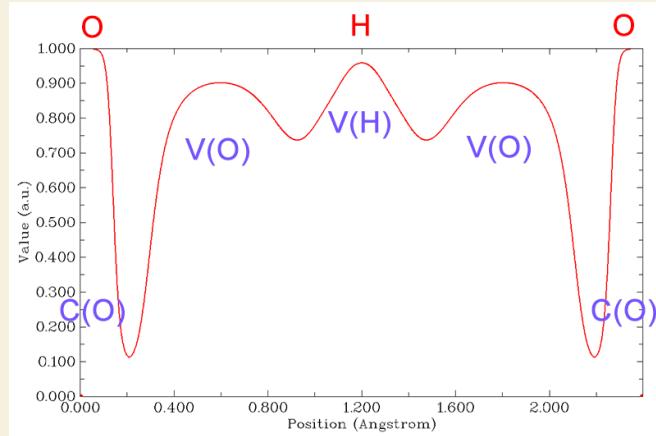
In order to make you better understand how the CVB index is automatically calculated in Multiwfn, here I explain the implementation detail. After the user inputted index of D, H and A atoms, the ELF curves corresponding to D-H and H-A are calculated in turn, and then the ELF(CV,D), ELF(DH-A) and ELF(CV,A) are automatically identified from the curve data, as illustrated below



(4) Special case: Calculating CVB index for some very strong H-bonds

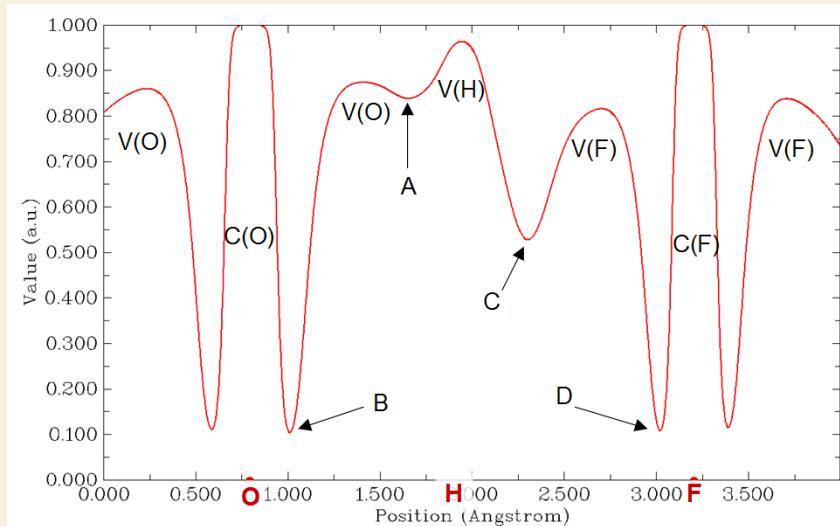
In principle, the CVB index calculation protocol described above works for most kinds of

systems that have typical H-bond, both intermolecular and intramolecular H-bonds can be analyzed in the same way. However, for some very strong H-bonds, whose hydrogen is lying at midpoint between two heavy atoms, such as $\text{H}_2\text{O}\cdots\text{H}^+\cdots\text{OH}_2$, this protocol is no longer valid because its ELF curve does not show typical feature, as shown below (since the O-H-O angle in this system is close to 180° , only one plot is needed):



It can be seen that the $V(D,H)$ has bifurcated as $V(O)$ and $V(H)$. In this case you should evaluate CVB index manually by plotting ELF curve maps, and the $\text{ELF}(DH-A)$ in the standard CVB index expression should be replaced with the ELF value at the local minimum between the $V(H)$ and $V(O)$ in the curve map.

Below is a more complicated case, $\text{F}\cdots\text{H}\cdots\text{O-H}$, you also need to manually evaluate the CVB index by plotting ELF curve map. The ELF curve map shown below was plotted between the F and O (the $\text{F}\cdots\text{H}\cdots\text{O}$ is almost linear, therefore only one plot is needed), as can be seen the ELF is unsymmetric with respect to the central hydrogen:

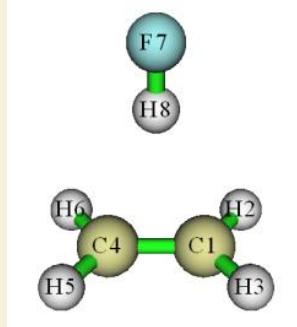


This system can be regarded as having two H-bonds, the H-bond binding energy of $\text{O-H}\cdots\text{F}$ and $\text{O}\cdots\text{H-F}$ must be very different. For the former, $\text{CVB index} = \text{ELF}(B) - \text{ELF}(C)$, while for the latter, $\text{CVB index} = \text{ELF}(D) - \text{ELF}(A)$. This is because when discussing the system for example as $\text{O-H}\cdots\text{F}$, the O and F behave as donor and acceptor atoms, respectively. Therefore the minimum of point C between H and F should be regarded as the $\text{ELF}(DH-A)$, while the minimum of point B should be viewed as $\text{ELF}(C-V,D)$.

(5) Special case: H-bond acceptor is not a single atom

Acceptor of some H-bonds is not a single atom. For example, the acceptor of HF...ethylene is the π region of ethylene. This kind of H-bond is known as π -hydrogen bond. In this case, you also have to manually calculate the CVB index.

The wavefunction file of the HF...ethylene has been provided as *examples\C2H4_HF.wfn*, its geometry is shown below.

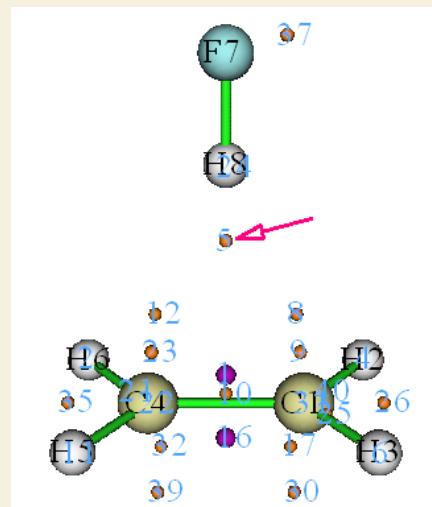


For this system, you can obtain ELF(C-V,D) by plotting ELF curve map between F7 and H8 and read the ELF value at minimum, the value will found to be 0.0944.

ELF(DH-A) of this system can be obtained via ELF topology analysis. To do this, we input below commands in Multiwfn:

```
2 // Topology analysis
-11 // Select the real space function to be analyzed
9 // ELF
6 // Search critical points by randomly distribute initial guesses within a sphere
4 // Set the sphere center as geometry center of three atoms
1,4,8 // Center of C1, C4 and H8 will be set as the sphere center
0 // Start searching (the sphere radius, the number of guessing points can be set by
corresponding options in the interface)
-9 // Return
0 // Visualize topology analysis result
```

Now you can see below graph. Clearly, the critical point 5 corresponds to the bifurcation point between V(D,H) and the basin of π electron.



Close the GUI, select option 7, and then input 5 to check properties of the critical point 5, you will find its ELF value is 0.1241, which is just the ELF(DH-A) of this H-bond. Hence, the CVB index

of this system is $0.0944 - 0.1241 = -0.0297$.

In fact, since this system has high symmetry, you can also obtain ELF value of the critical point 5 by simply plotting ELF curve map between H8 and midpoint of C1-C4.

Information needed: Atom coordinates, GTFs

3.200.2 Calculate atomic and bond dipole moments in Hilbert space

This function is used to calculate atomic and bond dipole moments directly based on basis functions (*viz.* in Hilbert space). You can also consult Section 12.3.2 of the book *Ideas of Quantum Chemistry* (L. Piela, 2007).

Theory

In the formalism of basis functions, the system dipole moment can be expressed as follows

$$\mu = \mu^{\text{nuc}} + \mu^{\text{ele}} = \sum_A Z_A \mathbf{R}_A - \sum_i \sum_j P_{i,j} \langle \chi_i | \mathbf{r} | \chi_j \rangle$$

where Z and \mathbf{R} are charge and coordinate of nuclei. P is density matrix, $\langle \chi_i | \mathbf{r} | \chi_j \rangle$ is dipole moment integral between basis function i and j .

The system dipole moment can be decomposed as the sum of single-atom terms and atom pair terms

$$\mu = \sum_A \mu_A^{\text{tot}} + \sum_{A>B} \sum_B \mu_{AB}^{\text{tot}} = \sum_A (\mu_A^{\text{nuc}} + \mu_A^{\text{pop}} + \mu_A^{\text{dip}}) + \sum_{A>B} \sum_B (\mu_{AB}^{\text{pop}} + \mu_{AB}^{\text{dip}})$$

The expression and physical meaning of the five terms are

μ_A^{nuc} : Dipole moment due to nuclear charge

$$\mu_A^{\text{nuc}} = Z_A \mathbf{R}_A$$

μ_A^{pop} : Dipole moment due to the electron population number localized on single atom (Notice that this is different to the electron population number calculated by Mulliken or similar methods, because the overlap population numbers have not been absorbed into respective atoms)

$$\mu_A^{\text{pop}} = -p_A^{\text{loc}} \mathbf{R}_A \quad p_A^{\text{loc}} = \sum_{i \in A} \sum_{j \in A} P_{i,j} \langle \chi_i | \chi_j \rangle$$

μ_A^{dip} : Atomic dipole moment, which reflects the electron dipole moment around an atom. \mathbf{r}_A is the coordinate variable with respect to nucleus A

$$\begin{aligned} \mu_A^{\text{dip}} &= -\sum_{i \in A} \sum_{j \in A} P_{i,j} \langle \chi_i | \mathbf{r}_A | \chi_j \rangle \quad \text{where } \mathbf{r}_A = \mathbf{r} - \mathbf{R}_A \\ &= -\sum_{i \in A} \sum_{j \in A} P_{i,j} \langle \chi_i | \mathbf{r} | \chi_j \rangle - \mu_A^{\text{pop}} \end{aligned}$$

μ_{AB}^{pop} : Dipole moment due to the overlap population between atom A and B

$$\mu_{AB}^{\text{pop}} = -p_{AB} \mathbf{R}_{AB} \quad p_{AB} = 2 \sum_{i \in A} \sum_{j \in B} P_{i,j} \langle \chi_i | \chi_j \rangle \quad \mathbf{R}_{AB} = (\mathbf{R}_A + \mathbf{R}_B) / 2$$

μ_{AB}^{dip} : Bond dipole moment, which somewhat reflects the electron dipole moment around geometry center of corresponding two atoms. Of course, if A and B are not close to each other, then this term will be very small, and thus inappropriate to be called as bond dipole moment.

$$\begin{aligned}\mu_{AB}^{\text{dip}} &= -2 \sum_{i \in A} \sum_{j \in B} P_{i,j} \langle \chi_i | \mathbf{r}_{AB} | \chi_j \rangle \quad \text{where } \mathbf{r}_{AB} = \mathbf{r} - \mathbf{R}_{AB} \\ &= -2 \sum_{i \in A} \sum_{j \in B} P_{i,j} \langle \chi_i | \mathbf{r} | \chi_j \rangle - \mu_A'^{\text{pop}}\end{aligned}$$

By means of Mulliken-type partition, the bond dipole moments can be incorporated into atomic dipole moments, so that the system dipole moment can be written as the sum of single center terms

$$\mu = \sum_A (\mu_A^{\text{nuc}} + \mu_A'^{\text{pop}} + \mu_A'^{\text{dip}})$$

where $\mu_A'^{\text{pop}}$ is the dipole moment due to the Mulliken population number of atom A

$$\mu_A'^{\text{pop}} = -p_A^{\text{Mul}} \mathbf{R}_A \quad p_A^{\text{Mul}} = \sum_B \sum_{i \in B} \sum_{j \in B} P_{i,j} \langle \chi_i | \chi_j \rangle$$

and $\mu_A'^{\text{dip}}$ is the atomic overall dipole moment of atom A

$$\mu_A'^{\text{dip}} = -\sum_B \sum_{i \in B} \sum_{j \in B} P_{i,j} \langle \chi_i | \mathbf{r}_A | \chi_j \rangle = -\sum_B \sum_{i \in B} \sum_{j \in B} P_{i,j} \langle \chi_i | \mathbf{r} | \chi_j \rangle - \mu_A^{\text{Mul}}$$

Note that the B index in above formulae runs over all atoms.

Usage

The input file must contain basis function information (e.g. .fch, .molden and .gms).

After you enter present function, you can choose option 1 to output information of a specific atom, including: Atomic local population number, p_A^{loc} ; atomic dipole moment, μ_A^{dip} ; contribution to system dipole moment due to nuclear charge, μ_A^{nuc} ; contribution to system dipole moment due to electron, $\mu_A^{\text{dip}} + \mu_A^{\text{pop}}$; contribution to system dipole moment, $\mu_A^{\text{nuc}} + \mu_A^{\text{dip}} + \mu_A^{\text{pop}}$.

You can also choose option 2 to output information between specific atom pair, including: bond population number, μ_{AB}^{pop} ; bond dipole moment, μ_{AB}^{dip} ; contribution to system dipole moment, $\mu_{AB}^{\text{pop}} + \mu_{AB}^{\text{dip}}$.

If choose 3, atomic overall dipole moment and related information of selected atoms will be outputted, including: Atomic Mulliken population number, p_A^{Mul} ; atomic overall dipole moment, $\mu_A'^{\text{dip}}$; contribution to system dipole moment due to nuclear charge, μ_A^{nuc} ; contribution to system dipole moment due to electron, $\mu_A'^{\text{dip}} + \mu_A'^{\text{pop}}$; contribution to system dipole moment, $\mu_A^{\text{nuc}} + \mu_A'^{\text{dip}} +$

μ_A^{pop} .

If you choose option 10, then X/Y/Z components of electron dipole moment matrix will be outputted to dipmatx.txt, dipmaty.txt and dipmatz.txt in current folder, respectively. For example, the (i,j) element of Z component of electron dipole moment matrix corresponds to

$$-\sum_i \sum_j P_{i,j} \langle \chi_i | z | \chi_j \rangle$$

Information needed: Atom coordinates, basis functions

3.200.3 Generate cube file for multiple orbital wavefunctions

By this function, grid data of multiple orbital wavefunctions can be calculated and then exported to a single cube file or separate cube files at the same time.

After you entered this function, you need to first define the orbitals you are interested in (e.g. 3,5,9-17), and then define grid setting, after that choose the scheme to export the grid data. If you select scheme 1, then grid data of all orbitals you selected will be outputted to orbital.cub in current folder. Lots of visualization programs, including VMD and Multiwfn, support the cube file containing multiple sets of grid data. If you select scheme 2, then the grid data will be exported as separate files, for example *orb000003.cub*, *orb000005.cub*, *orb000009.cub*, etc. The number in the filename corresponds to orbital index.

Information needed: Atom coordinates, GTFs

3.200.4 Generate iso-chemical shielding surfaces (ICSS) and related quantities

Theory

Nuclear independent chemical shielding (NICS) is commonly studied at some special points (e.g. ring center), and in some papers NICS is investigated by scanning its value in a line (1D) or in a plane (2D). The so-called iso-chemical shielding surface (ICSS) actually is the isosurface of NICS, which clearly exhibits the distribution of NICS in 3D space, and thus presents a very intuitive picture on aromaticity.

Present function is used to generate grid data and visualize isotropic ICSS, anisotropic ICSS, ICSS_{XX}, ICSS_{YY} and ICSS_{ZZ}, they correspond to the isosurface of NICS, NICS_{ani}, NICS_{XX}, NICS_{YY} and NICS_{ZZ}, respectively. At a given point, NICS_{ani} is defined as $\varepsilon_3 - (\varepsilon_1 + \varepsilon_2)/2$, where ε denotes the eigenvalue of magnetic shielding tensor ranked from small to large (*viz.* ε_3 is the largest one).

The original paper of ICSS is *J. Chem. Soc. Perkin Trans. 2*, **2001**, 1893. While ICSS_{ani}, ICSS_{XX}, ICSS_{YY} and ICSS_{ZZ} are proposed by me. I believe for planar systems, the component form of ICSS must be more meaningful and useful than ICSS, just like NICS_{ZZ} has conspicuous advantage over NICS. ICSS_{ani} is useful to reveal the anisotropic character of NICS in different regions.

Usage

Multiwfn itself is incapable of calculating magnetic shielding tensor and thus requires Gaussian to do that. The general steps is shown below

(1) Prepare an input file of Gaussian single point task for the system under study, %chk have to be explicitly specified. For example, see "examples\ICSS\anthracene.gjf". This file will be taken as template to automatically build Gaussian input file of NMR task

(2) After boot up Multiwfn, load the file containing the atomic coordinate of your system (e.g. .pdb/.xyz/.wfn/.fch ...), then enter subfunction 4 of main function 200. Beware that the atomic coordinates loaded must exactly accord with the ones in the file prepared in step 1.

(3) Set up grid by following the prompt. Beware that even using medium quality grid may be fairly time-consuming. Hence low quality grid is in general recommended for medium-size system.

(4) Input n , namely do not skip step 5 and 6.

(5) Input the path of the Gaussian input file you prepared in step 1.

(6) Many input files of Gaussian NMR task are generated in current folder, they are named as NICS0001.gjf, NICS0002.gjf ...

In these files, each Bq atom corresponds to a grid point. In the NMR task Gaussian will output magnetic shielding tensor at each Bq along with that at each nuclei. By default 8000 atoms (the real ones + Bq) are presented in each input file, but this can be altered via "NICSnptlim" parameter in settings.ini. The reason why separate files rather than a single file are generated is because Gaussian cannot run properly if the number of Bq atoms is too large due to over-consume of memory, also there is upper limit on the total number of atoms in each Gaussian run. You can try to set NICSnptlim to a larger value if you have large physical memory, this will reduce overall cost of ICSS analysis.

Note that for G09 D.01 and E.01, due to a bug in memory allocation when using the default Harris initial guess, you should always add "guess=huckel" to route section of template .gjf file, otherwise the NICSnptlim has to be set to a very small value (e.g. 1000) to make Gaussian work; in this case the overall cost of ICSS calculations is often quite high. For other Gaussian versions, this keyword should not be added. If error occurs in Link 401 module when "guess=huckel" is specified, try to use "guess=core" instead. For G16, the guess keyword is not needed.

(7) Feed all of the input files generated in last step to Gaussian to yield output files. If possible, please keep Multiwfn running (If you have terminated it, reboot Multiwfn and repeat step 2, 3 with the same setting and input y at step 4).

Hint: You can make use of the script "examples\runall.sh" (for Linux) or "examples\runall.bat" (for Windows), which invokes Gaussian to run all .gjf files in current folder to yield output files with the same name but with .out suffix.

(8) Input the path of the folder containing Gaussian output files yielded in the last step. Then Multiwfn will load the magnetic shielding tensors from the NICS0001.out, NICS0002.out ... in this folder.

(9) Select the property you are interested in.

(10) Visualize isosurface or export the grid data to cube file by corresponding option. For example, in step 9 you selected "ZZ component", then the isosurface and the grid data will correspond to ICSS_{ZZ}. You can also select "-1 Load other property" to load other property.

Notice that if this is not the first time you analyze your system and you already have Gaussian output files of NMR task of present system in hand, you can start from step 2 and select y in step 4 to bypass the step 5, 6 and 7. Of course, the grid setting in step 3 must be exactly accord with the one used in generating the Gaussian output files of NMR task.

An example is given in Section 4.200.4.

Information needed: Atom coordinates, input file of Gaussian single point task, output file of Gaussian NMR task.

3.200.5 Plot radial distribution function for a real space function

This function is used to plot radial distribution function (RDF) for a real space function

$$RDF(r) = \int f(r, \Omega) r^2 d\Omega$$

where r is radial distance from sphere center, and Ω denotes angular coordinate in a sphere layer.

The integration curve of RDF can also be plotted

$$I(r') = \int_{r_{\text{low}}}^{r'} RDF(r) dr = \int_{r_{\text{low}}}^{r'} \int f(r, \Omega) r^2 d\Omega dr$$

Clearly, if r_{low} is set to 0 (viz. sphere center), then $I(\infty)$ will be the integral of f over the whole space.

In present function, one can choose the real space function to be studied, set the position of sphere center, set the lower and upper limit to be calculated and plotted, set the number of points in radial and angular parts. The larger the number of points, the more accurate the integration curve.

After the parameters are set, select option 0 to start the calculation, and then you will see a new menu, in which you can plot RDF and its integration curve, save the graph or export the corresponding original data.

An example is given in Section 4.200.5.

Information needed: Atom coordinates, GTFs

3.200.6 Analyze correspondence between orbitals in two wavefunctions

Theory

This function is used to analyze correspondence between the orbitals in two wavefunctions. The two sets of orbitals can be produced under different basis sets, by different theoretical methods, at different external environments, in different electronic states, or at slightly different geometries. The two sets of orbitals can also be different types, for example the first set of orbitals are canonical MOs produced by Hartree-Fock calculation, while the second set of orbitals are natural orbitals produced by post-HF calculation.

The orbitals $\{i\}$ in present wavefunction (the wavefunction loaded when Multiwfn boots up) can be represented as linear combination of the orbitals $\{j\}$ in another wavefunction (the wavefunction you specified after entering present module), i.e.

$$|i\rangle = \sum_j C_{i,j} |j\rangle \quad \text{where } C_{i,j} = \langle i | j \rangle$$

Once we have the overlap integral, we immediately know how j is associated to i . The contribution from orbital j to orbital i is simply the square of overlap integral, namely $\langle i | j \rangle^2 \times 100\%$.

Usage

After you enter this function, first you need to input the orbital range to be considered for present wavefunction ($i\text{start1}\sim i\text{end1}$), and then input the path of the second wavefunction and the orbital range to be considered ($i\text{start2}\sim i\text{end2}$). After that the overlap matrix between $i\text{start1}\sim i\text{end1}$ and $i\text{start2}\sim i\text{end2}$ will be calculated by Becke's multi-center numerical integration scheme. Then

you will see the five largest contributions from istart2~iend2 to each orbital in istart1~iend1. If you want to obtain all coefficients (as well as the corresponding contributions) of istart2~iend2 in a given orbital among istart1~iend1, you can then directly input the index of the orbital. If you need all overlap integrals between istart1~iend1 and istart2~iend2, input -1, then they will be outputted to convmat.txt in current folder.

If an orbital (*i*) in present wavefunction can be exactly expanded as linear combination of istart2~iend2, then the normalization condition must be satisfied:

$$\sum_{j=\text{istart2}}^{\text{iend2}} \langle i | j \rangle^2 \times 100\% = 100\%$$

From the Multiwfn output you can find the maximum deviation to normalization condition. If the value is zero, that means all orbitals in istart1~iend1 can be exactly represented by the orbitals in istart2~iend2.

Note that the atomic coordinate of present wavefunction and that of the second wavefunction are not necessarily identical, the two wavefunctions can even correspond to different molecules. However if the difference of the distribution scope of the atomic coordinates in the two wavefunctions is large, the integration accuracy must be low and the result is not reliable.

Commonly the default integration grid is fine enough, i.e. 50 radial points and 230 angular points. If you wish to improve the accuracy, you should set "iautointgrid" in settings.ini file to 0, then the number of radial and angular integration grids will be directly determined by "radpot" and "sphpot" in settings.ini; set them to a large value will result in good integration accuracy.

The computational cost of this function directly depends on the number of orbitals in consideration; so if your system contains very large number of orbitals, do not choose all orbitals at once.

Example

As an example, we use present function to study the conversion relationship between HF/6-31+G* MOs and MP2/6-31+G* natural orbitals (NO) for CH₃NH₂.

After boot up Multiwfn we input

C:\CH3NH2_MP2.wfn // MP2/6-31+G* wavefunction file, there are 48 NOs

200

6

[Press ENTER button directly to choose all orbitals]

C:\CH3NH2_HF.wfn // HF/6-31+G* wavefunction file, there are 9 MOs

[Press ENTER button directly to choose all orbitals]

Then you will see

1:	2(94.63%)	1(5.36%)	4(0.01%)	3(0.00%)	6(0.00%)
2:	1(94.64%)	2(5.36%)	3(0.00%)	9(0.00%)	6(0.00%)
3:	3(85.21%)	4(9.87%)	7(3.24%)	9(0.88%)	6(0.79%)
4:	4(86.88%)	3(10.66%)	6(1.94%)	9(0.48%)	7(0.02%)
5:	6(54.88%)	7(41.11%)	9(3.20%)	4(0.45%)	3(0.34%)
6:	8(57.50%)	5(42.47%)	7(0.00%)	6(0.00%)	4(0.00%)
7:	5(57.50%)	8(42.48%)	7(0.00%)	4(0.00%)	6(0.00%)
8:	9(86.62%)	6(10.30%)	7(1.63%)	4(0.89%)	3(0.50%)
9:	7(53.98%)	6(32.07%)	9(8.76%)	3(3.28%)	4(1.89%)

```

10:      9( 0.00%)   7( 0.00%)   6( 0.00%)   4( 0.00%)   2( 0.00%)
11:      5( 0.00%)   8( 0.00%)   3( 0.00%)   7( 0.00%)   4( 0.00%)
... (ignored)
47:      6( 0.01%)   7( 0.00%)   9( 0.00%)   3( 0.00%)   4( 0.00%)
48:      9( 0.04%)   7( 0.00%)   6( 0.00%)   4( 0.00%)   3( 0.00%)

```

The first column denotes the index of the MP2 NOs, the largest five contributions from the Hartree-Fock MOs to them are shown at right side. As can be seen, the first (second) NO is nearly equivalent to the second (first) MO. While the 9th NO cannot be solely represented by any MO, it mainly arises from the severe mix of the 7th (53.98%) and 6th MOs (32.07%), the 9th MO also has non-neglectable contribution (8.76%).

If you want to get all coefficients and compositions of an orbital of present wavefunction (e.g. the 5th NO), then simply input 5, you will see

```

1 Contribution: 0.000 % Coefficient: 0.001509
2 Contribution: 0.000 % Coefficient: -0.000042
3 Contribution: 0.339 % Coefficient: -0.058201
4 Contribution: 0.447 % Coefficient: 0.066869
5 Contribution: 0.000 % Coefficient: 0.000000
6 Contribution: 54.882 % Coefficient: 0.740826
7 Contribution: 41.113 % Coefficient: -0.641194
8 Contribution: 0.000 % Coefficient: 0.000000
9 Contribution: 3.199 % Coefficient: -0.178856

Total: 99.980 %

```

this output suggests $|\text{NO } 5\rangle = 0.7408 |\text{MO } 6\rangle - 0.6412 |\text{MO } 7\rangle - 0.1788 |\text{MO } 9\rangle \dots$. The 99.98% at the last line indicates that the 5th NO can be perfectly represented by linear combination of these nine MOs.

Information needed: Atom coordinates, GTFs

3.200.7 Parse output of (hyper)polarizability task of Gaussian

The output of (hyper)polarizability task of Gaussian (*polar* keyword) is difficult to understand, at least for beginners. This function is used to parse these outputs and then print them in a more readable format, and at the same time some quantities relating to (hyper)polarizability analysis will be outputted. Currently this function is formally compatible with Gaussian 09 and 16.

Basic concepts

Energy of a system can be written as Taylor expansion with respect to uniform external electric field \mathbf{F}

$$\begin{aligned}
 E(\mathbf{F}) &= E(\mathbf{0}) + \left. \frac{\partial E}{\partial \mathbf{F}} \right|_{\mathbf{F}=0} \mathbf{F} + \frac{1}{2} \left. \frac{\partial^2 E}{\partial \mathbf{F}^2} \right|_{\mathbf{F}=0} \mathbf{F}^2 + \frac{1}{6} \left. \frac{\partial^3 E}{\partial \mathbf{F}^3} \right|_{\mathbf{F}=0} \mathbf{F}^3 + \frac{1}{24} \left. \frac{\partial^4 E}{\partial \mathbf{F}^4} \right|_{\mathbf{F}=0} \mathbf{F}^4 + \dots \\
 &\equiv E(\mathbf{0}) - \boldsymbol{\mu}_0 \mathbf{F} - \frac{1}{2} \boldsymbol{\alpha} \mathbf{F}^2 - \frac{1}{6} \boldsymbol{\beta} \mathbf{F}^3 - \frac{1}{24} \boldsymbol{\gamma} \mathbf{F}^4 - \frac{1}{120} \boldsymbol{\delta} \mathbf{F}^5 - \frac{1}{720} \boldsymbol{\epsilon} \mathbf{F}^6 \dots
 \end{aligned}$$

$$\boldsymbol{\mu}_0 = -\left. \frac{\partial E}{\partial \mathbf{F}} \right|_{\mathbf{F}=0} \quad \boldsymbol{\alpha} = -\left. \frac{\partial^2 E}{\partial \mathbf{F}^2} \right|_{\mathbf{F}=0} \quad \boldsymbol{\beta} = -\left. \frac{\partial^3 E}{\partial \mathbf{F}^3} \right|_{\mathbf{F}=0} \quad \boldsymbol{\gamma} = -\left. \frac{\partial^4 E}{\partial \mathbf{F}^4} \right|_{\mathbf{F}=0}$$

where $\boldsymbol{\mu}_0$ is permanent dipole moment, which is a vector; $\boldsymbol{\alpha}$ is polarizability, which is a matrix (second rank tensor); $\boldsymbol{\beta}$ is first hyperpolarizability, which is a third rank tensor and known as second-order nonlinear optical response (NLO) coefficient; $\boldsymbol{\gamma}$ is second hyperpolarizability, which is a fourth rank tensor and known as third-order NLO coefficient. The higher terms such as $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$ are very unimportant and thus rarely discussed. The (hyper)polarizability tensors are directly correlated to the frequency of external field \mathbf{F} . If \mathbf{F} has zero-frequency (static electric field), then the (hyper)polarizabilities are known as static or frequency-independent ones. The dynamic or frequency-dependent (hyper)polarizabilities correspond to those at external electromagnetic fields with non-zero frequency.

Dipole moment of a system in uniform electric field can be written as

$$\boldsymbol{\mu} = -\frac{\partial E}{\partial \mathbf{F}} = \boldsymbol{\mu}_0 + \underbrace{\boldsymbol{\alpha} \mathbf{F}}_{\boldsymbol{\mu}_1} + \underbrace{(1/2) \boldsymbol{\beta} \mathbf{F}^2}_{\boldsymbol{\mu}_2} + \underbrace{(1/6) \boldsymbol{\gamma} \mathbf{F}^3}_{\boldsymbol{\mu}_3} + \dots$$

The linear response of dipole moment with respect to \mathbf{F} , namely the $\boldsymbol{\mu}_1$ term, can be explicitly written as below

$$\boldsymbol{\mu}_1 = \boldsymbol{\alpha} \cdot \mathbf{F} \Rightarrow \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} = \begin{bmatrix} \alpha_{xx} & \alpha_{xy} & \alpha_{xz} \\ \alpha_{yx} & \alpha_{yy} & \alpha_{yz} \\ \alpha_{zx} & \alpha_{zy} & \alpha_{zz} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$$

The polarizability $\boldsymbol{\alpha}$ is a symmetric matrix rather than a scalar, implying the difference of polarizability in different directions. In order to facilitate comparison of overall polarizability between various systems, it is convenient to define the isotropic average polarizability

$$\langle \boldsymbol{\alpha} \rangle = \text{Tr}(\boldsymbol{\alpha}) / 3 = (\alpha_{xx} + \alpha_{yy} + \alpha_{zz}) / 3$$

Anisotropy of polarizability can be defined in various ways:

Definition 1 see *Chem. Phys.*, **410**, 90 for example

$$\Delta\alpha = \sqrt{[(\alpha_{xx} - \alpha_{yy})^2 + (\alpha_{xx} - \alpha_{zz})^2 + (\alpha_{yy} - \alpha_{zz})^2 + 6(\alpha_{xy}^2 + \alpha_{xz}^2 + \alpha_{yz}^2)] / 2}$$

Definition 2 This definition is the most commonly use one, see *J. Chem. Phys.*, **98**, 3022 (1993) for example

$$\Delta\alpha = \sqrt{[(\alpha_{xx} - \alpha_{yy})^2 + (\alpha_{xx} - \alpha_{zz})^2 + (\alpha_{yy} - \alpha_{zz})^2] / 2}$$

Definition 3 $\{\varepsilon\}$ stand for eigenvalues of $\boldsymbol{\alpha}$ ranking from small to large

$$\Delta\alpha = \varepsilon_3 - (\varepsilon_1 + \varepsilon_2) / 2$$

Hyperpolarizability $\boldsymbol{\beta}$ is a third rank tensor that can be described by a 3*3*3 matrix. For dynamic $\boldsymbol{\beta}$, Gaussian is capable of calculating its dc-Pockels form $\beta_{i,j,k}(-\omega;\omega,0)$ and SHG form $\beta_{i,j,k}(-2\omega;\omega,\omega)$. For the former, the i and j index can be freely exchanged, while for the later the j and k index can be freely exchanged, therefore the 3*3*3=27 components can be reduced to 18 unique

ones. For static case, viz. $\beta_{i,j,k}(0;0,0)$, since all of the three indices are exchangeable (known as Kleinman's symmetry, e.g. $\beta_{xxy}=\beta_{yxy}=\beta_{yyx}$, $\beta_{zyy}=\beta_{yzy}=\beta_{yyz} \dots$), only 10 components are unique. The Kleinman's symmetry is only approximately applied to dyanmic β at low-frequency external field.

The β value in one of the three Cartesian axes can be calculated by the general equation

$$\beta_i = (1/3) \sum_j (\beta_{ijj} + \beta_{jji} + \beta_{iji}) \quad i, j = \{x, y, z\}$$

The magnitude of β is defined as

$$\beta_{\text{tot}} = \sqrt{\beta_x^2 + \beta_y^2 + \beta_z^2}$$

β_{proj} (also known as β_{vec}), which is the projection of β on dipole moment vector μ , can be sampled experimentally (electric field induced second harmonic generation (EFISH) experiment). β_{\parallel} is the β component in the direction of μ

$$\beta_{\text{proj}} = \sum_i \frac{\mu_i \beta_i}{|\mu|} \quad \beta_{\parallel} = (3/5)\beta_{\text{proj}}$$

Some people prefer to discuss the perpendicular and parallel components of β with respect to Z axis, they are defined respectively as

$$\beta_{\perp(Z)} = (1/5) \sum_j (2\beta_{zjj} - 3\beta_{jzj} + 2\beta_{jjz}) \quad \beta_{\parallel(Z)} = (3/5)\beta_z$$

For static case, we can explicitly write out β in x, y and z directions as

$$\begin{aligned} \beta_x &= \beta_{xxx} + \beta_{xxy} + \beta_{xz} \\ \beta_y &= \beta_{yxx} + \beta_{yyy} + \beta_{yz} \\ \beta_z &= \beta_{zxz} + \beta_{zyy} + \beta_{zzz} \end{aligned}$$

and $\beta_{\perp(Z)} = (1/5)\beta_z$.

Usage

The *polar* keyword in Gaussian is specific for calculating α and β based on analytic derivatives (by means of coupled-perturbed SCF equation) or numerical derivatives (by means of finite field treatment).

After you enter this function of Multiwfn, you should select which calculation combination is used, so that Multiwfn can properly parse the output. In Gaussian 09/16 there are several circumstances:

- (1) *polar* keyword + the methods supporting analytic 3-order derivatives (HF/DFT/Semi-empirical methods)
- (2) *polar* keyword + the methods supporting analytic 2-order derivatives (e.g. MP2)
- (3) *polar=Cubic* keyword + the methods supporting analytic 2-order derivatives
- (4) *polar* keyword + the methods supporting analytic 1-order derivatives (CISD, QCISD, CCSD, MP3, MP4(SDQ), etc.)
- (5) *polar=DoubleNumer* (equivalent to *Polar=EnOnly*) keyword + the methods supporting analytic 1-order derivatives
- (6) *polar* keyword + the methods only supporting energy calculation (CCSD(T), QCISD(T), MP4(SDTQ), MP5, etc.)

Only (1), (3) and (5) produce hyperpolarizability, other combinations only produce

polarizability. Notice that in the Gaussian input file you must specify **#P!**

Only for (1), if *CPHF=RdFreq* is also specified along with *polar*, or you used *polar=DCSHG*, and meantime the external field frequencies (e.g. 0.05 0.07 0.1) are written after molecular geometry with a blank line in front of it, Gaussian will calculate and output frequency-dependent (hyper)polarizabilities along with static (hyper)polarizability. *CPHF=RdFreq polar* cases only evaluate $\beta(-\omega; \omega, 0)$ values, while *polar=DCSHG* cases evaluate both $\beta(-\omega; \omega, 0)$ and $\beta(-2\omega; \omega, \omega)$ values. By default Multiwfn only parses the static (hyper)polarizability. If you wish to parse the frequency-dependent ones instead of the static one, before selecting option 1 you should select “-1 Toggle if load frequency-dependent result for option 1” first. The user then can choose the result at which frequency will be parsed. Also, one can choose parsing the $\beta(-\omega; \omega, 0)$ one or the $\beta(-2\omega; \omega, \omega)$ one, the latter can be chosen only for the case of *polar=DCSHG*.

It is noteworthy that, it is well known that the sign of all hyperpolarizability components outputted by Gaussian are wrong and should be multiplied by -1, Multiwfn automatically accounts for this problem.

In this function, Multiwfn outputs all unique components of dipole moment, polarizability and hyperpolarizability (if available) with explicit labels, as well as isotropic polarizability, polarizability anisotropy (in definitions 1 and 2), hyperpolarizability in three axes (β_x , β_y , β_z), magnitude of hyperpolarizability (β_{tot}) and so on. All of them have been introduced above.

By using option -3, you can choose the unit in the output. Atomic unit, SI unit and esu unit can be chosen. The conversion factors are

		SI	esu
μ	1 a.u.	8.47835×10^{-30} C m	2.54175×10^{-18} esu
α	1 a.u.	1.6488×10^{-41} C ² m ² J ⁻¹	1.4819×10^{-25} esu
β	1 a.u.	3.20636×10^{-53} C ³ m ³ J ⁻²	8.63922×10^{-33} esu
γ	1 a.u.	6.23538×10^{-65} C ⁴ m ⁴ J ⁻³	5.03670×10^{-40} esu

The polarizability α is often expressed in terms of "polarizability volume" (α'), which has volume unit. $\alpha(1 \text{ a.u.}) = \alpha'(0.14818470 \text{ \AA}^3)$.

An example is given in Section 4.200.7. More discussion and examples about this function can be found in my blog article "Using Multiwfn to analyze polarizability and first hyperpolarizability outputted by Gaussian" (<http://sobereva.com/231>, in Chinese)

Information needed: Output file of "polar" task of Gaussian

3.200.8 Calculate (hyper)polarizability by sum-over-states (SOS) method

This function is used to calculate polarizability, first, second, and third hyperpolarizabilities based on sum-over-states method.

A brief survey of the theories for evaluating (hyper)polarizability

Some basic concepts of (hyper)polarizability are introduced in Section 3.200.7. There are a few different ways to calculate (hyper)polarizability, including derivative method, sum-over-states (SOS) and response method

(1) Derivative method: This is the most straightforward and commonly used one. The derivatives needed by static (hyper)polarizability can be evaluated analytically by means of coupled-perturbed SCF (CPSCF) equation; specifically, CPHF for HF and CPKS for KS-DFT. These derivatives can also be evaluated numerically by means of finite difference technique, which is also known as finite field (FF) method. Evidently FF is much slower and not as accurate as CPSCF, however it is still useful, because high-order of analytic derivatives, especially the ones at sophisticated post-HF levels, are not widely supported by many quantum chemistry programs due to the difficulties in coding. When all requested derivatives are available analytically, derivative method will be very efficient. The frequency-dependent variant of CPSCF equation enables the derivative method to evaluate dynamic (hyper)polarizability, but there is no way to evaluate dynamic (hyper)polarizability in terms of FF treatment. The *polar* keyword in Gaussian, as discussed carefully in Section 3.200.7, corresponds to this derivative method.

(2) SOS method: This method for evaluating static and dynamic (hyper)polarizability is relatively inefficient, because in principle it involves a sum over all excited states (in practical applications, taking 60~120 lowest states into account is often enough), while determination of a large number of excited states is usually quite time consuming in *ab initio* cases (e.g. CIS and TDDFT), especially for large system (e.g. >40 atoms). Due to the high computational cost, SOS is generally not recommended for evaluation of (hyper)polarizability when derivative method can be carried out analytically. The only advantages of SOS may be that the contribution from different states can be separated and discussed respectively, and when transition dipole moments between different excited states are available in hand, the (hyper)polarizability at different frequencies can be evaluated rather rapidly. Worthnoting, the SOS based on the cheap semi-empirical ZINDO calculation (SOS/ZINDO) is very popular for evaluating (hyper)polarizability of large system.

(3) Response method: This method is specific for dynamic (hyper)polarizability and also known as propagator method. TDHF and TDDFT are its two practical realizations. This method is not prevalently supported by mainstream quantum chemistry codes.

Working equations of SOS method

The explicit SOS equations for evaluating polarizability and 1st/2nd/3rd hyperpolarizability can be found in *J. Chem. Phys.*, **99**, 3738 (1993), the idea was originally proposed by Orr and Ward in *Mol. Phys.*, **20**, 512 (1971).

The equations for polarizability α and first hyperpolarizability β are (all units are in a.u.)

$$\alpha_{AB}(-\omega; \omega) = \sum_{i \neq 0} \left[\frac{\mu_{0i}^A \mu_{i0}^B}{\Delta_i - \omega} + \frac{\mu_{0i}^B \mu_{i0}^A}{\Delta_i + \omega} \right] = \hat{P}[A(-\omega), B(\omega)] \sum_{i \neq 0} \frac{\mu_{0i}^A \mu_{i0}^B}{\Delta_i - \omega}$$

$$\beta_{ABC}(-\omega_\sigma; \omega_1, \omega_2) = \hat{P}[A(-\omega_\sigma), B(\omega_1), C(\omega_2)] \sum_{i \neq 0} \sum_{j \neq 0} \frac{\mu_{0i}^A \overline{\mu_{ij}^B} \mu_{j0}^C}{(\Delta_i - \omega_\sigma)(\Delta_j - \omega_2)}$$

where

$$\mu_{ij}^A = \langle i | \hat{\mu}^A | j \rangle \quad \overline{\mu_{ij}^A} = \mu_{ij}^A - \mu_{00}^A \delta_{ij} \quad \omega_\sigma = \sum_i \omega_i$$

$A, B, C \dots$ denote one of directions {x,y,z}; ω is energy of external fields, $\omega=0$ corresponds to static electric field; Δ_i stands for excitation energy of state i with respect to ground state 0. \hat{P} is

permutation operator, for α and β evidently there are $2!=2$ and $3!=6$ permutations, respectively. μ_{ij}^A is A component of transition dipole moment between state i and j ; when $i=j$ the term simply corresponds to electric dipole moment of state i . $\hat{\mu}$ is dipole moment operator, e.g. $\hat{\mu}^x \equiv -x$.

The SOS equation for second hyperpolarizability γ is

$$\begin{aligned}\gamma_{ABCD}(-\omega_\sigma; \omega_1, \omega_2, \omega_3) &= \hat{P}[A(-\omega_\sigma), B(\omega_1), C(\omega_2), D(\omega_3)](\gamma^I - \gamma^{II}) \\ \gamma^I &= \sum_{i \neq 0} \sum_{j \neq 0} \sum_{k \neq 0} \frac{\mu_{0i}^A \overline{\mu_{ij}^B} \overline{\mu_{jk}^C} \mu_{k0}^D}{(\Delta_i - \omega_\sigma)(\Delta_j - \omega_2 - \omega_3)(\Delta_k - \omega_3)} \\ \gamma^{II} &= \sum_{i \neq 0} \sum_{j \neq 0} \frac{\mu_{0i}^A \mu_{i0}^B \mu_{0j}^C \mu_{j0}^D}{(\Delta_i - \omega_\sigma)(\Delta_i - \omega_1)(\Delta_j - \omega_3)}\end{aligned}$$

The Z components of γ is defined as

$$\gamma_z = (1/15) \sum_A (\gamma_{zAAz} + \gamma_{zAzA} + \gamma_{zzAA})$$

Similarly for X and Y components. The total magnitude of γ may be measured as $\gamma_{\text{tot}} = \sqrt{\gamma_x^2 + \gamma_y^2 + \gamma_z^2}$. There are two common definitions of average value of γ :

def 1: $\langle \gamma \rangle = \gamma_x + \gamma_y + \gamma_z$

def 2: $\langle \gamma \rangle = (1/5)[\gamma_{xxxx} + \gamma_{yyyy} + \gamma_{zzzz} + \gamma_{xxyy} + \gamma_{xxzz} + \gamma_{yyzz} + \gamma_{yyxx} + \gamma_{zzxx} + \gamma_{zzyy}]$

The SOS equation for third hyperpolarizability δ is

$$\begin{aligned}\delta_{ABCDE}(-\omega_\sigma; \omega_1, \omega_2, \omega_3, \omega_4) &= \hat{P}[A(-\omega_\sigma), B(\omega_1), C(\omega_2), D(\omega_3), E(\omega_4)](\delta^I - \delta^{II} - \delta^{III}) \\ \delta^I &= \sum_{\substack{i,j,k,l \\ (\neq 0)}} \frac{\mu_{0i}^A \overline{\mu_{ij}^B} \overline{\mu_{jk}^C} \overline{\mu_{kl}^D} \mu_{l0}^E}{(\Delta_i - \omega_\sigma)(\Delta_j - \omega_\sigma + \omega_1)(\Delta_k - \omega_3 - \omega_4)(\Delta_l - \omega_4)} \\ \delta^{II} &= (1/2) \sum_{\substack{i,j,k \\ (\neq 0)}} \frac{\mu_{0i}^A \mu_{i0}^B \mu_{0j}^C \overline{\mu_{jk}^D} \mu_{k0}^E}{(\Delta_j + \omega_2)(\Delta_k - \omega_4)} \left(\frac{1}{\Delta_i - \omega_\sigma} + \frac{1}{\Delta_i - \omega_1} \right) \left(\frac{1}{\Delta_j - \omega_3 - \omega_4} + \frac{1}{\Delta_k + \omega_2 + \omega_3} \right) \\ \delta^{III} &= (1/2) \sum_{\substack{i,j,k \\ (\neq 0)}} \frac{\mu_{0i}^A \mu_{i0}^B \mu_{0j}^C \overline{\mu_{jk}^D} \mu_{k0}^E}{(\Delta_i - \omega_\sigma)(\Delta_i - \omega_1)} \left[\frac{1}{(\Delta_j - \omega_3 - \omega_4)(\Delta_k - \omega_4)} + \frac{1}{(\Delta_j + \omega_2)(\Delta_k + \omega_2 + \omega_3)} \right]\end{aligned}$$

Input file

Two kinds of input files may be used:

(1) Plain text file containing excitation energies and transition dipole moments for all involved states. Polarizability, first, second and third hyperpolarizabilities can be calculated in this case. Below format should be satisfied (assume a very simple case, only 2 excited states).

```
2 // The number of excited states
1 1.1 // Excited state 1, its index and excitation energy (eV)
2 3.2
```

```
0 0 0.845 0.2 0.4 // Electric dipole moment of ground in x,y,z (a.u.)
0 1 0.231 0.3 0.7 // Transition dipole moment between ground state and excited state 1
0 2 0.112 0.564 0.21
1 1 0.021 0.465 0.0 // Electric dipole moment of excited state 1
1 2 0.001 0.3 0.11 // Transition dipole moment between excited states 1 and 2
2 2 0.432 0.14 0.42
```

Important hint: You can directly utilize the function introduced in Section 3.21.5 to generate such a plain text file based on the Gaussian or ORCA output file of electron excitation task.

If merely polarizability is interesting, only the content before the line "1 1" is needed to be provided, all other contents can be omitted; in this case, the number of excited states should be written as a negative number (-2 in above case) to tell Multiwfn do not to load them.

(2) Gaussian output file of common CIS, TDHF, TDDFT or ZINDO task. Since Gaussian does not output all transition dipole moments needed by SOS hyperpolarizability calculation, *only polarizability will be calculated by Multiwfn in this case*. In order to obtain accurate polarizability, the number of calculated states should be large enough. If *nstates* keyword is specified to a very large value, e.g. 1000000, then all states will be calculated. #P is suggested to be used, since the excitation energy will then be printed in a higher precision format.

Usage

After you entered this function you will see a menu, there are three kinds of functions:

(1) Option 1~4: Used to calculate α , β , γ and δ at a given frequency, respectively. User needs to input frequency of each external field. Since calculation of γ and δ are often time-consuming, in these cases users will be prompted to input the number of states in consideration, small number is evidently cheaper, but may give rise to lower accurate result.

(2) Option 5~7: Used to study the variation of α , β and γ with respect to the number of states in consideration. User needs to input frequency of each external field. For α and β , the number of states taken into account ranges from 1 to all states loaded, the stepsize is 1. While for γ , since the computational cost may be quite high, users are allowed to customly defined the ending value and stepsize. The result will be outputted to plain text file "alpha_n.txt" or "beta_n.txt" or "gamma_n.txt" in current folder, the meaning of each column can be seen in command-line window. One can directly drag these plain text files into window of Origin program to plot the variations as curve maps.

(3) Option 15~17: Used to study the variation of the α , β and γ with respect to frequency of external fields. For α , users need to input initial value, ending value and stepsize of external field frequencies. For β and γ , users should write a plain text file, each row corresponds to a pair of frequency (in a.u.) to be calculated. Multiwfn will prompt users to input the path of the file. Below is an example file used to study how $\gamma(-0,0,\omega,-\omega)$ varies as ω goes from 0 to 0.2 a.u. with stepsize of 0.02

```
0.0 0.0 0.0
0.0 0.02 -0.02
0.0 0.04 -0.04
...[ignored]
0.0 0.2 -0.2
```

Since the computational cost for evaluating γ may be quite high, in this case users are allowed to set the number of states in consideration. The result will be outputted to plain text file "alpha_w.txt" or "beta_w.txt" or "gamma_w.txt" in current folder, the meaning of each column can

be seen in command-line window. One can directly drag these plain text files into window of Origin program to plot the variations as curve maps.

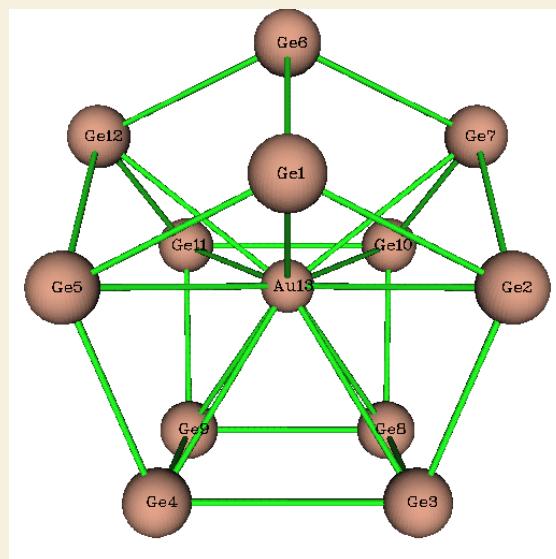
Multiwfn not only outputs the tensor of (hyper)polarizability, but also outputs many related quantities, such as anisotropy, magnitude and the component along Z axis. The quantities involving α and β are discussed in Section 3.200.7, the ones involving γ have been introduced above.

An example is given in Section 4.200.8.

More discussion and examples about this function can be found in my blog article "Using Multiwfn to calculate polarizability and hyperpolarizability based on sum-over-states (SOS) method" (<http://sobereva.com/232>, in Chinese)

3.200.9 Calculate average bond length and average coordinate number

This function is used to calculate average bond length between two elements and average coordinate number. This function is particularly useful for analyzing structure character of atom clusters, for example the Ge₁₂Au cluster shown below (the structure file is provided as *examples\Ge12Au.pdb*). By using this function, we can immediately obtain the average Ge-Ge bond length and average Au-Ge bond length, as well as average coordinate number of Ge due to Ge-Au or Ge-Ge bonds, or of Au due to Ge-Au bonds. A nice application of this kind of analysis on Al clusters can be found in *J. Chem. Phys.*, **111**, 1890 (1999).



The average bond length is defined as follows

$$\langle R \rangle = \frac{1}{n_b} \sum_{i>j} R_{ij}$$

where R_{ij} is the distance between atom i and j , only the terms smaller than or equal to a given distance cutoff (e.g. 2.2Å) will be regarded as bonds and thus be taken into the summation. n_b is the total number of bonds.

The average coordinate number is calculated as follows

$$CN = \frac{1}{n} \sum_i N_i$$

where N_i is the number of bonds surrounding the atom i , n is the total number of atoms.

After you entered this function, you need to input two elements, for example *Ge,Au*, and input a distance cutoff, for example 3.2, then the Ge-Au contacts $\leq 3.2\text{\AA}$ will be regarded as Ge-Au bonds and the average bond length will be calculated, the minimum and maximum bond lengths will also be outputted. After that, if you select *y*, the average coordinate number of Ge due to Ge-Au bonds will be shown.

Information needed: Atom coordinates

3.200.10 Output various kinds of integral between orbitals

This function is used to calculate electric/magnetic dipole moment integral, velocity integral, kinetic energy integral and overlap integral between orbitals, advanced users may recognize the significance of these data. In the case of a range of orbitals, the results are exported to orbint.txt in current folder, the first and second columns correspond to the index of the two orbitals; In the case of a pair of orbitals, the result is directly printed on screen.

The electric dipole moment integral vector between two orbitals is defined as

$$\mathbf{p}_{ij} = \mathbf{p}_{ji} = \langle \varphi_i | -\mathbf{r} | \varphi_j \rangle$$

The magnetic dipole moment integral vector between two orbitals is calculated as (more detail can be found in Section 3.21.1.1. The negative sign is ignored)

$$\mathbf{M}_{ij} = i \langle \varphi_i | \mathbf{r} \times \nabla | \varphi_j \rangle$$

The velocity integral vector between two orbitals is evaluated as (the negative sign is ignored)

$$\mathbf{v}_{ij} = i \langle \varphi_i | \nabla | \varphi_j \rangle$$

Worthnoting, due to the Hermitian of the operators, we have

$$\begin{aligned} \mathbf{M}_{ii} &= 0 & \mathbf{M}_{ij} &= \mathbf{M}_{ji}^* = -\mathbf{M}_{ji} \\ \mathbf{v}_{ii} &= 0 & \mathbf{v}_{ij} &= \mathbf{v}_{ji}^* = -\mathbf{v}_{ji} \end{aligned}$$

Note that the imaginary sign is not explicitly shown in the output.

The kinetic energy and overlap integrals between two orbitals are respectively evaluated as

$$K_{ij} = -(1/2) \langle \varphi_i | \nabla^2 | \varphi_j \rangle \quad S_{ij} = \langle \varphi_i | \varphi_j \rangle$$

Information needed: Atom coordinates, GTFs

3.200.11 Calculate center, the first and second moments of a real space function

This function is used to calculate center, integral value, the first and second moments of a specific real space function for present system.

Theory

The center of a real space function f is defined as

$$\mathbf{r}_c = \frac{\int \mathbf{r} \times f(\mathbf{r}) d\mathbf{r}}{\int f(\mathbf{r}) d\mathbf{r}}$$

where the denominator is the integral of the function over the whole space.

The first moment is a vector and is evaluated as

$$\mathbf{\mu} = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} = \int \begin{bmatrix} x \\ y \\ z \end{bmatrix} f(\mathbf{r}) d\mathbf{r}$$

The second moment is a matrix and defined as

$$\mathbf{\Theta} = \begin{bmatrix} \Theta_{xx} & \Theta_{xy} & \Theta_{xz} \\ \Theta_{yx} & \Theta_{yy} & \Theta_{yz} \\ \Theta_{zx} & \Theta_{zy} & \Theta_{zz} \end{bmatrix} = \int \begin{bmatrix} x^2 & xy & xz \\ yx & y^2 & yz \\ zx & zy & z^2 \end{bmatrix} f(\mathbf{r}) d\mathbf{r}$$

where x, y, z are the coordinate components relative to \mathbf{r}_c .

If its eigenvalues $\{\varepsilon\}$ are sorted from low to high, then the anisotropy of $\mathbf{\Theta}$ can be calculated

as $\varepsilon_3 - (\varepsilon_1 + \varepsilon_2) / 2$. The radius of gyration can be calculated as $\sqrt{\frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3}{\int f(\mathbf{r}) d\mathbf{r}}}.$

Usage

This function employs Becke's multicenter integration method for evaluating above mentioned quantities. The accuracy is determined by radial points and angular points, which can be set by "radpot" and "sphpot" in settings.ini, respectively.

The option 1 calculates and outputs the integral, the first and second moments (relative to \mathbf{r}_c), and the corresponding anisotropy and radius of gyration of the real space function selected by option 3. The center (\mathbf{r}_c) should be properly first set by option 4. Of course, commonly \mathbf{r}_c is initially unknown, the default value is (0,0,0), you may use option 2 to evaluate the actual center of a real space function and take it as the center for the succeeding calculations triggered by option 1.

For example, to evaluate the first and second moments of spin density (relative to the center of spin density), after you entered this function you should input

```
3 // Select real space function
5 // Spin density
2 // Calculate center of spin density
y // Take the resultant center as the one for evaluating the first and second moments
I // Evaluate the first and second moments
```

When using option 1, if the function to be studied is chosen as electron density, then the nuclear quadrupole and molecular quadrupole moment tensors will also be outputted, the latter can be straightforwardly obtained by subtracting the former by the second moment of electron density.

Information needed: Atom coordinates, GTFs

3.200.12 Calculate energy index (EI) or bond polarity index (BPI)

This function is used to calculate energy index (EI) or bond polarity index (BPI), which were defined in *J. Phys. Chem.*, **94**, 5602 (1990).

The EI for atom *A* in a molecule is defined as follows

$$EI_A = \frac{\sum_i^{\text{val}} \varepsilon_i \eta_i \Theta_{i,A}}{\sum_i^{\text{val}} \eta_i \Theta_{i,A}}$$

where $\Theta_{i,A}$ denotes composition of atom *A* in MO *i*. η_i and ε_i are occupation and energy of MO *i*, respectively. The summation runs over valence MOs. In fact, the denominator is simply the number of valence electrons of atom *A*, and the numerator corresponds to total energy of its valence electrons. Therefore, EI_A can be regarded as average energy per valence electron of atom *A*. In the original paper of EI, Mulliken method was used to compute the atomic contribution to MOs, thus this method is also employed in present implementation of EI, though other methods such as Hirshfeld partition should work equally well or even better. (Note that since Mulliken method is used, which is incompatible with diffuse functions, the use of diffuse basis functions must be avoided!)

The BPI between atoms *A* and *B* in a molecule is defined as

$$BPI_{AB} = (EI_A - EI_A^{\text{ref}}) - (EI_B - EI_B^{\text{ref}})$$

where EI^{ref} is reference EI value derived from calculation of homonuclear species. For example, you study BPI_{CN} for H_3C-NH_2 , then EI_C^{ref} is computed as EI_C in ethane, and EI_N^{ref} is computed as EI_N in H_2N-NH_2 . The larger magnitude of BPI_{AB} implies higher bond polarity of the A-B bond.

Group electronegativity is evaluated as negative of EI_X for corresponding radical, X is the attaching atom. For example, to obtain group electronegativity for $-CH_3$ group, you should calculate $-EI_C$ for $\cdot CH_3$ radical.

This function of Multiwfn is used to calculate EI for specific atom in present system, all the R, RO and U types of HF/DFT wavefunction are supported. Multiwfn automatically detects the number of inner-core electrons and determines which MOs are the valence ones and thus should be taken into account.

An example is given in Section 4.200.12.

Information needed: Atom coordinates, basis functions

3.200.14 Domain analysis (Obtaining properties within isosurfaces of a function)

This function is used to integrate specific real space functions in domains. The domain refers to individual spatial regions enclosed by isosurfaces of a given real space function. For example, you can integrate electron density within various domains defined by isosurfaces of reduced density gradient (RDG) to study strength of weak interaction (In fact, if this module is flexibly used, much more things can be realized. For example, visualizing and obtaining volume of molecular cavity).

Below is basic procedure of using this module:

- (1) Use option 2 and 3 to set the way to define the domains. For example, you selected RDG by option 2 and input <0.5 in option 3, then the region where RDG is less than 0.5 will be integrated.
- (2) Choose option 1 and properly define grid, then Multiwfn starts to calculate the grid data for the real space function you selected and identifies domains that satisfied the criterion you set. (If you already have grid data in memory, you may also choose option -1 to directly use it rather than calculate new grid data)
- (3) Once calculation in last step is finished, Multiwfn prints total number of grids in each domain. In very simple case, from these information you may directly infer which domain is the one you want to study, while for common case, you need to use option "3 Visualize domains" to visualize domains in a GUI window, in which you can select domain at the right-bottom list and check its profile, each green point on the graph corresponds to a grid in the domain. Once you found the domain you are interested in, close GUI and choose "1 Perform integration for a domain", then select the real space function to be integrated in the domain, after that the integral value, domain volume and average/maximum/minimum value of the integrand in the domain will be outputted. In addition, minimum and maximum X/Y/Z of grids belonging to the domain, as well as span distance in X/Y/Z will also be outputted. You can also select "2 Perform integration for all domains" to obtain integral values for all domains. (Hint: If what you need is just domain volume, you should choose the user-defined function as integrand, which by default is 1.0 everywhere and thus integrating this function does not take any computational time).

In addition, some regions you are interested in may be identified as separate domains, to study the property of the regions more conveniently, you can choose "-1 Merge specific domains" to merge selected domains are single domain, so that you do not need to manually sum up their integral values.

If you wish to visualize domains in third-part program such as VMD, there are two ways:

- Select "10 Export a domain as domain.cub file in current folder" and input index of the domain of interest, then Multiwfn will export the domain as domain.cub, in which the grid point belonging and not belonging to the domain have value of 1 and 0, respectively (value of boundary grids are also outputted as 0 to guarantee that the isosurfaces always look closed). After that, you can load the cube file into visualization program to visualize isosurface using isovalue of 0.5.
- Select "11 Export boundary grids of a domain to domain.pdb file in current folder" and input index of the domain of interest, then the resulting .pdb file will contain particles, each one corresponds to a boundary grid. You can directly drag this file into VMD and render the particles as spheres to visualize domain.

By the way, in the post-process menu there is an option "5 Calculate q_bind index for a domain", this is used to calculate the q_{bind} index defined in *J. Phys. Chem. A*, **115**, 12983 (2011), in which it was demonstrated that for hydrogen-bond dimer, the scan curve of q_{bind} index well mimics to actual potential energy curve. This index for a domain is defined as:

$$\begin{aligned} q_{\text{att}} &= \int_{\lambda_2(\mathbf{r})<0} \rho^n(\mathbf{r}) d\mathbf{r} && \text{attractive effect} \\ q_{\text{rep}} &= \int_{\lambda_2(\mathbf{r})>0} \rho^n(\mathbf{r}) d\mathbf{r} && \text{repulsive effect} \\ q_{\text{bind}} &= -(q_{\text{att}} - q_{\text{rep}}) \end{aligned}$$

where $\lambda_2(\mathbf{r})$ is the second largest eigenvalue of electron density Hessian matrix at \mathbf{r} , its sign can be utilized to discriminate interaction type. Since the paper showed that $n=4/3$ gives best correlation between q_{bind} and actual potential curve, in Multiwfn the n parameter is chosen as $4/3$. Note that the paper used isosurface of RDG=0.6 when calculating this index. More negative of q_{bind} may imply more stabilized interaction.

The method used to integrate domains in present module is even-grid integration method. In other words, the integration value of a real space function for a domain is simply the sum of the real space function value of the grids constituting the domain multiplied by grid volume. Therefore, the accuracy of integration result is directly affected by the quality of grid you set.

Illustrative application examples of present module is given in Section 4.200.14.

Information needed: Atom coordinates, GTFs (depends on the choice of real space function)

3.200.15 Calculate electron correlation index

The total, dynamic and nondynamic electron correlation indices proposed by Matito et al. in *Phys. Chem. Chem. Phys.*, **18**, 24015 (2016) are useful indicator of measuring magnitude of electron correlation in present system.

Dynamic and nondynamic electron correlation indices (I_D and I_{ND}) are defined as

$$\begin{aligned} I_D &= \frac{1}{4} \sum_i \{ [\eta_i(1-\eta_i)]^{1/2} - 2\eta_i(1-\eta_i) \} \\ I_{\text{ND}} &= \frac{1}{2} \sum_i \eta_i(1-\eta_i) \end{aligned}$$

Where i denotes index of natural spin orbital, η is corresponding occupation number. Note that in some cases, η may be marginally larger than 1.0 or negative, Multiwfn automatically set it to 1.0 and 0.0 respectively to make the calculation feasible.

Total electron correlation index defined is

$$I_T = I_D + I_{\text{ND}} = \frac{1}{4} \sum_i [\eta_i(1-\eta_i)]^{1/2} |\varphi_i(\mathbf{r})|^2$$

Present function is used to calculate all the three electron correlation indices. Any wavefunction file carrying occupation number of natural orbitals may be used as input file, e.g. .wfn, .wfx and .molden files. An example is given in Section 4.A.6.

Note that Matito et al. also proposed local version of the three functions to characterize electron

correlation in local regions, Multiwfn is also able to study them, see Section 4.A.6 for example.

3.200.16 Generate natural orbitals, natural spin orbitals and spin natural orbitals based on the density matrix in .fch/.fchk file

In .fch (or .fchk) file, density matrix is always recorded. For example, if you carried out a MP2 task for an open-shell system with Gaussian keywords "# MP2/cc-pVTZ density", then the resulting .fch file will have below four fields recording corresponding type of density matrix:

```
Total SCF Density, Spin SCF Density, Total MP2 Density, Spin MP2 Density
```

While for a closed-shell system, if the keyword used is "# TD PBE1PBE/6-311G* density", then the resulting .fch file will contain below type of density matrix:

```
Total SCF Density, Total SCF Density, Total CI Rho(1) Density, Total CI Density
```

If you do not know which kinds of density matrix are recorded in the .fch file, simply search "Density" in the file.

Various kinds of natural orbitals can be obtained via diagonalization of proper type of density matrix:

Natural orbitals (NOs): Diagonalizing total density matrix. The occupation is from 0.0 to 2.0. This type of NOs is also known as spatial NOs, and specifically, unrestricted natural orbital (UNO) for unrestricted wavefunctions

Alpha and beta natural orbitals (collectively known as natural spin orbitals, NSOs): Diagonalizing alpha and beta density matrix, respectively. The occupation is from 0.0 to 1.0.

Spin natural orbitals (SNOs): Diagonalizing spin density matrix (i.e. Difference between alpha and beta density matrix). The occupation is from -1.0 to 1.0. SNO with positive (negative) occupation represent distribution of unpaired alpha (beta) electrons.

Using present function, you can obtain any set of above mentioned types of NOs. For example, you want to obtain SNOs of triplet water at CCSD/cc-pVDZ level, you can run below Gaussian input file:

```
%chk=C:\CCSD_water_m3.chk  
#p CCSD/cc-pVDZ density
```

```
test
```

```
0 3  
O 0.00000000 0.00000000 0.11930801  
H 0.00000000 0.75895306 -0.47723204  
H 0.00000000 -0.75895306 -0.47723204
```

Convert the .chk file to .fch, then boot up Multiwfn and input

```
C:\CCSD_water_m3.fch
```

```
200
```

```
16
```

```
CC // Meaning we want to analyze coupled-cluster density matrix. You can also input SCF
```

here to analyze Hartree-Fock density matrix

3 // Generate SNOs (if the system is closed-shell, this selection will not occur, since only NOs can be generated in this case)

Now the basis function information in memory has been updated to SNOs. If then you want to visualize SNOs, or to perform real space function analysis (*e.g.* analyzing orbital composition of SNOs via Hirshfeld partition), you should choose *y* to export wavefunction information to *new.molden* file in current folder, and then program will automatically load it. After that, all following analysis will correspond to SNOs.

One of my blog articles detailedly discussed and presented analysis example of SNOs: "The way of generating natural orbitals based on fch file in Multiwfn and analysis instances about excited state wavefunctions and spin natural orbitals" (in Chinese) <http://sobereva.com/403>.

Note: Once .molden file containing SNOs is loaded into Multiwfn, the system will be regarded as open-shell and there will be the same number of alpha and beta orbitals, only the former correspond to SNOs, while the latter are completely meaningless and you should simply ignore them.

This function works well for .fch/.fchk files produced by Gaussian and PSI4, but may or may not be compatible with other programs.

If this function is used in combination with PSI4, you can analyze wavefunction as high as CCSD(T) level, please check Section 4.A.8 for detail.

Information needed: .fch/.fchk file

4 Tutorials and Examples

Prologue and generation of input files

Welcome to use Multiwfn! If you have not read "ALL USERS MUST READ" at page 2 of this manual, please read it first. If you encountered any problem in using Multiwfn, please free feel to post topic on Multiwfn forum.

Before getting start, I first show you how to generate various kind of input files.

Notice that different functions in Multiwfn require different type of input file, see Section 2.5 for explanation. briefly speaking, for any analysis that solely based on real space function, you can use .wfn or .wfx as input file. However, many functions require basis function information, in these cases you have to use .fch/fchk, .molden or .gms file as input file. Since these files contain richer information than .wfn/wfx file (*i.e.* basis functions and virtual orbitals), in principle for any function that requires .wfn/wfx file as input file, you can also use .fch/molden/gms instead. A few functions in Multiwfn (*e.g.* AdNDP and ICSS analysis) rely on some special files, requirements on the input files for these situations are clearly indicated in corresponding section in Chapter 3.

Generating .wfn and .wfx files

- 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* (you can consult *H2O.gif* in “examples” folder), then run this file. If the task terminates normally, H2O.wfn will appear in *C:\otoboku* folder.

If you would like to generate .wfx file in Gaussian (supported since G09 B.01), simply write *out=wfx* instead of *out=wfn* in route section.

If you use MCSCF in Gaussian, in order to generate and export natural orbitals to .wfn, you should also use *pop=no* keyword. If you are using Gaussian older than G09 C.01, please carefully read below information:

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 will still be HF orbitals. If you are using TDDFT or CIS and you want to export natural orbitals corresponding to excited state wavefunction, you also need to specify “density” keyword.

For CCD/CCSD, QCISD or MP2/3, MP4SDQ tasks based on unrestricted HF reference state, only when “pop=NOAB” keyword is also specified then natural spin orbitals rather than spatial natural orbitals will be saved to the .wfn file. TD, CI and MCSCF tasks of Gaussian can not produce natural spin orbitals.

If the Gaussian you are using is older than G09 B.01, be aware that there is a serious bug, if your task is restricted open-shell (ROHF and RODFT), the occupation numbers of singly occupied orbitals in .wfn file will erroneously be 2.0, you have to open the file by text editor, locate the last entry “OCC NO =”, and then manually change the value behind it to 1. 0000000.

- GAMESS-US: Add *AIMPAC=.TRUE.* in \$CONTRL section. After the task is finished, the generated .dat file in the folder defined by \$SCR environment variable (see *rungms* script) will contain wavefunction information with the same format as .wfn file, extract the content between “---- TOP OF INPUT FILE FOR BADER'S AIMPAC PROGRAM ----” and “---- END OF INPUT FILE FOR BADER'S AIMPAC PROGRAM ----” and save them to a new file with “.wfn” suffix.

- ORCA: .wfn and .wfx files can be generated simply using *aim* keyword in the input file, or using the command *orca_2aim XXX* to convert *XXX.gbw* to *XXX.wfn* and *XXX.wfx*. At least for ORCA 4.1, .wfn and .wfx files cannot be generated when ECP is used. Using .molden file as input

file of Multiwfn is always more recommended.

As regards the method of outputting .wfn files in other quantum chemistry packages, please consult corresponding manuals.

Generating .fch file

- Gaussian: First run a Gaussian task with e.g. %chk=test.chk to yield binary checkpoint file *test.chk* file, then run command *formchk test.chk* to convert *test.chk* to *test.fch*.

Note: There is no any difference between .fch and .fchk formats. The former and the latter are the default extensions of formatted checkpoint file of Windows and Linux version of Gaussian, respectively. You can use either of them as input file of Multiwfn.

When post-HF task is performed, the orbitals and occupations recorded in Gaussian .fch file by default are the HF ones, hence the Multiwfn analysis results are identical to HF. Similarly, under default case, analysis results based on .fch file produced by TDDFT task are identical to ground state DFT wavefunction. To analyze wavefunction for post-HF wavefunction or TDDFT excited state wavefunction, analysis should be done using natural orbitals (NOs) at corresponding level, there are two ways to yield them:

(1) Make Multiwfn generate natural orbitals (or spin natural orbitals, natural spin orbitals) by using subfunction 16 of main function 200. See Section 3.200.16 for detail. This way is recommended since it is very convenient.

(2) Make Gaussian write natural orbitals into .fch file. You should first perform post-HF or TDDFT task with “density” keyword, and then rerun the task only with “guess (save,only,naturalorbitals) chkbasis” in route section. Note that Gaussian fills orbital occupation numbers into orbital energy field in .fch file, hence you should write “saveNO” in the first line of .fch file to let Multiwfn know this behavior. Beware that if what you performed is open-shell post-HF calculation, even above process is unable to correctly store natural spin orbitals into .fch file. Generally, I strongly recommend using .wfn/.wfx file to view natural orbitals and analyze real space functions for post-HF wavefunctions.

For MCSCF calculation, you should load the resulting .fch file and use subfunction 16 of main function 200 to generate .molden file containing NOs at MCSCF level, and then use this .molden file as input file. Since for MCSCF the alpha and beta orbitals cannot be generated separately, for systems with spin multiplicity larger than 1, you must manually open the .fch file, set “Number of beta electrons” to the same value as “Number of alpha electrons” to make Multiwfn recognize that there is only one set of orbitals in the input file.

• Q-Chem: Write *GUI 2* in \$rem field, after task has finished, you will find the resulting .fchk file in current folder. Beware that before loading the .fchk files produced by Q-Chem into Multiwfn, you must set “ifchprog” in settings.ini to 2.

• PSI4: The .fchk file produced by currently latest version (not older versions) of PSI4 is compatible with Multiwfn. The *examples\psi4_fch.inp* is an example file of generating .fchk file at B3LYP/6-31G** level. In Section 4.A.8, I also show how to analyze post-HF wavefunction based on .fchk file of PSI4.

Generating .molden file

- ORCA: Using the command *orca_2mkl XX -molden* to convert *XX.gbw* to Molden input file *XX.molden.input*. You do not need to then manually change the suffix from .molden.input to .molden, since the former can also be recognized by Multiwfn.
- Molpro: Adding such as *put,molden,ltd.molden* at the last line of your input file, the Molden input file *ltd.molden* will be produced after finishing the calculation.

- Dalton: The program automatically outputs .molden file when calculation is finished. The file is *molden.inp* in .tar.gz package. This file can be directly loaded without changing suffix.
- NWChem: An example input file is provided as *examples\NWChem_molden.nw*. After running it, the .molden file will be generated in current folder. Notice that spherical harmonic basis functions must be employed (*i.e.* the "spherical" keyword) and "noautosym" keyword must be employed when the system has symmetry of point group.
- Dalton: Under default setting, once calculation is finished, the automatically generated .tar.gz package will contain *molden.inp* file, which is molden input file and can be directly loaded into Multiwfn.
- MRCC: Once the calculation is normally finished, a file named *MOLDEN* will be generated in current folder. Then rename it to make it possess .molden suffix. An example is given in Section 4.A.8.
- xtb: Run xtb with --molden option, then *molden.input* will be generated in current folder.
- Other programs: Please consult corresponding manuals.

When pseudo-potential is used and you need to do some analyses relating to nuclear charges, do not forget to manually change atomic indices in the .molden file as nuclear charges, see Section 2.5 for details.

Note: Currently, only the Molden input file generated by Molpro, ORCA, xtb, Dalton, NWChem, MRCC, deMon2k, BDF programs are formally supported. If the file is generated by other programs, the result may or may not be correct, because the files produced by numerous programs are non-standard or problematic. Fortunately, molden2aim utility is able to deal with the Molden input files produced by wider scope of programs and can output standardized Molden input files, which is then able to be used as Multiwfn input file. See Section 5.1 for detail.

Generating .gms file

GAMESS-US and Firefly (old name is PC-GAMESS) output file can also be used as Multiwfn input file, you need to change its suffix as .gms so that Multiwfn can recognize it. Currently, I can only guarantee that output file of HF/DFT calculation with default NPRINT option can be normally loaded by Multiwfn. A sample of input and output files of GAMESS-US are provided as *GAMESS_US.inp* and *UKS_cc-pVDZ.gms* in examples folder, respectively.

Now let us start! Note that the examples in 4.*x* section are relevant to main function *x*, therefore you can quickly find the examples you needed. Section 4.A includes special topics and advanced tutorials, in which more than one functions and some advanced skills may be involved, such as studying aromaticity and weak interactions. You can find almost all of the files involved in these examples in "examples" folder. All the texts behind // are comments and should not be inputted as command. Tutorials in this chapter only cover basic applications of Multiwfn, if you want to learn the usage of more functions and more options, please read corresponding sections in Chapter 3 and play with the options that not mentioned in the examples.

For most examples, I take .fch or .wfn as format of input file, however **it never means these functions can only accept these two formats!** If you have read Section 2.5, you must know how to properly choose format of input file for different functions.

PS1: If you would like to analyze wavefunction higher than CCSD level, reading Section 4.A.8 is suggested, you will need PSI4 or MRCC program.

PS2: In chapter 4, many my blog articles written in Chinese are involved, they often contain extended discussion and more examples. If you cannot read Chinese, you can try to use Google translator (For example, you can install Mozilla Firefox add-on called "Google translator for Firefox". After a successful installation, you will find an icon "T" in the firefox toolbar. Now open

the desired weblink, click the "T" icon and you will find the entire text in the desired language).

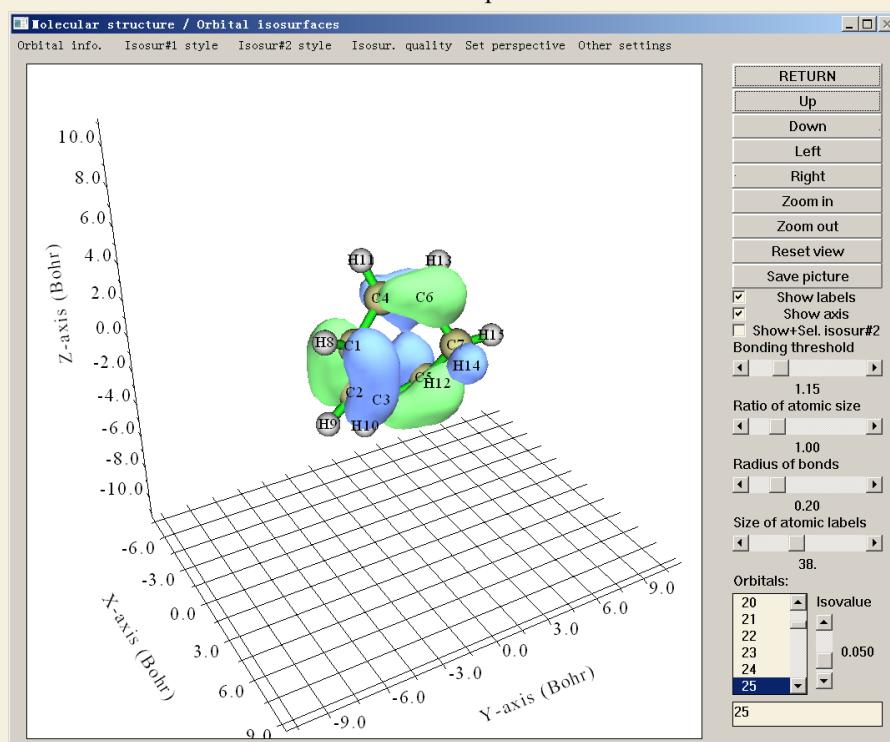
PS3: If you can read Chinese, reading these three articles will be highly helpful: "Tips for getting start with Multiwfn" (<http://sobereva.com/167>), "Multiwfn FAQ" (<http://sobereva.com/452>) and "The significance, functions and uses of multifunctional wavefunction analysis program Multiwfn" (<http://sobereva.com/184>).

4.0 View orbitals and structure

In Multiwfn, you can use main function 5 to generate cube file of an orbital wavefunction, or you can use subfunction 3 in main function 200 to generate single or multiple cube files for a batch of orbitals, then you can load them to your favourite visualization programs (*e.g.* VMD) to visualize corresponding orbitals. However, the simplest way to view orbitals is using main function 0 of Multiwfn, as described below.

4.0.1 Viewing molecular orbitals of cycloheptatriene

First, run a single point task for cycloheptatriene to get its .fch file, which has already been provided as *examples\cycloheptatriene.fch*. Boot up Multiwfn, input this path, then select main function 0, a GUI window will pop up, meanwhile information of all atom coordinates along with basic information of featured molecular orbitals are printed on Multiwfn console window.



You can rotate molecule, zoom in/out, adjust bonding threshold, save graph and so on via corresponding widgets. The numbers in right-bottom list are orbital indices, you can view orbital isosurface by selecting corresponding number, green and blue isosurfaces correspond to positive and negative regions, respectively. The iso value can be adjusted by dragging slide bar. Isosurface style and colors and be altered by corresponding suboptions in "Isosur#1 style". The quality of the

isosurface can be set by the option in "Isosur. quality", the isosurface will be finer/coarser if you input a larger/smaller value than default. In Section 4.0.2, you will learn how to plot two orbitals simultaneously in this window.

By selecting "Orbital info." - "Show all", energy, occupation number and type of all orbitals will be shown in Multiwfn console window. If you do not want too many high-lying virtual MOs are shown, you can choose "Show up to LUMO+10" or "Show occupied orbitals". If irreducible representations are recorded in the loaded .molden/.gms file, then they will be shown as the last column.

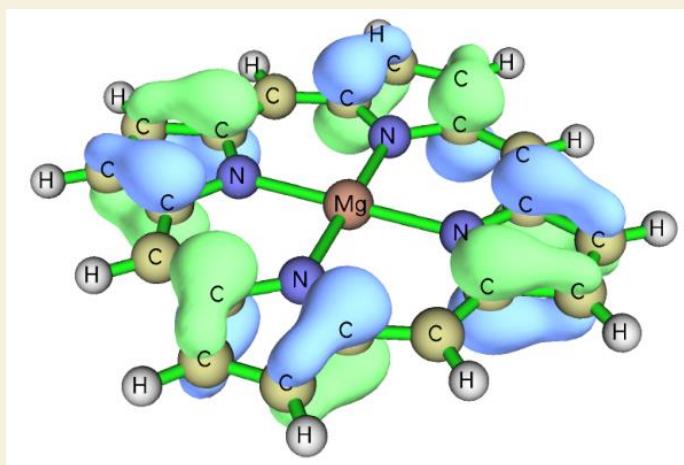
To close the window, click "RETURN" button. More detailed explanation about this interface can be found in See Section 3.2.

Note: Visualizing isosurface of Rydberg orbitals by main function 0 of Multiwfn is also possible, however, since they show very diffuse character, in order to avoid truncating of isosurfaces, you should select "Other settings"- "Set extension distance" in the menu, and then input a relatively large value, for example, 12 (the unit is Bohr), then select the orbital to visualize it. The default value of extension distance is controlled by "Aug3D" in settings.ini. An example of visualizing Rydberg orbitals is given in Part 2 of Section 4.200.5.

Tip: Recommendation of the steps for obtaining pretty orbital isosurface graph

- Enter main function 0, select the orbital to be visualized, properly set isovalue
- Click "Show Labels" to disable axis
- Properly adjust viewpoint
- Properly change the size of atomic labels. Note that type of the labels can be changed via "Set atomic label type" in "Other settings" at menu bar
 - If the rendering effect of the isosurface is not quite good, use "Set lighting" in "Other settings" to adjust lightings.
 - Select "Isosur. quality" in the menu bar, set the number of grid points to five times of the default value (For large systems, set to 10 times).
 - Click "Save picture". Use such as Irfanview or Photoshop program to open it, shrink the size of the image file to 50% (in this process resample will be automatically done, making anti-aliasing effect effectively realized), then properly crop the graph.

Below is an example obtained via above steps, the quality is pretty good



4.0.2 Viewing natural bond orbitals (NBO) of ethanol

There are two ways to view NBOs, if you are a Gaussian user, way 2 may be more convenient, however if you also need to view natural hybrid orbital (NHO) or natural atomic orbital (NAO) or some other types of orbitals generated by NBO program, you have to use way 1.

Way 1: Using NBO plot files

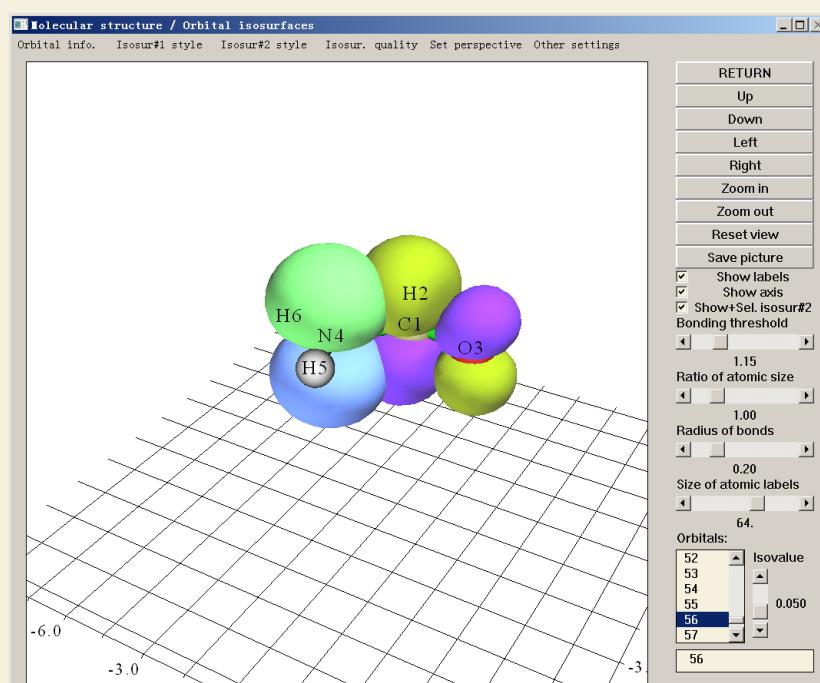
The common way is to generate NBO plot files (.31~.40) and load them into Multiwfn. To generate these files by Gaussian, you should add `pop=nboread` in route section, that means the keywords of NBO at the end of input file will be passed to NBO module (Link 607 in Gaussian), then add e.g. `$NBO plotfile=C:\NH2COH $END` at the end of the input file with a blank line before it, you can refer to the `NH2COH_NBO.gjf` in “example” directory. Run the input file by Gaussian, you will find that `NH2COH.31`, `NH2COH.32` ... `NH2COH.41` have been generated in `C:\` folder. The `NH2COH.31` and `NH2COH.37` have already been provided in “example” folder. Now boot up Multiwfn and input following commands

`examples\NH2COH.31` // .31 file contains necessary basis function information for plotting

`examples\NH2COH.37` // .37 file contains NBO information. .32~.40 files correspond to PNAO/NAO/PNHO/NHO/PNBO/NBO/PNLMO/NLMO/MO respectively. **Hint:** You can only input .37, because in present example the .37 and the .31 file share the same name

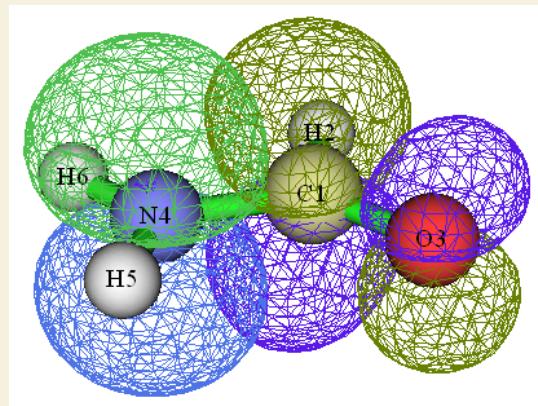
`0` // Enter the GUI

You can choose corresponding NBO orbital from right-bottom list to view the isosurface. Multiwfn is also capable to plot two orbitals simultaneously, for instance, here we will plot NBO 12 and NBO 56, which correspond to occupied lone pair of nitrogen atom and unoccupied anti- π bond between carbon and oxygen atoms respectively. Firstly, we choose 12 from the orbital list to plot NBO 12, and then click "Show+Sel. isosur#2", after that we click 56 in the list, you will see both of NBO 12 and NBO 56 are shown. The yellow-green and purple parts of NBO 56 (isosurface#2) correspond to positive and negative parts, respectively.



It is somewhat difficult to study overlapping extent between the two orbitals from the solid

face graph, so we choose "Use mesh" in "Isosur#1 style" and the counterpart in "Isosur#2 style" to make the two isosurfaces represented as mesh, see below. (Please also try "transparent face" style) Now the overlapping extent become distinct, it is quite clear that NBO 12 substantially overlapped with NBO 56, the resulting strong delocalization is one of the main reasons why the second-order perturbation energy between them is very large (~60 kcal/mol). In Section 4.4.5, you will learn how to obtain contour map for the two orbitals.



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 consulting other plot files such as .34, it is very easy.

Regarding the ways to pass the keywords for generating NBO 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 NBO plot files, an input file (.47) is needed to be prepared first. To generate it, you should load a file containing basis function information into Multiwfn, then enter main function 100, select subfunction 2, then choose corresponding option to export .47 file. After that, manually add *plot* keyword between "\$NBO" and "\$END" in the .47 file. Then if you use GENNBO program to run the .47 file, you will get NBO plot files.

Way 2: Using .fch file as NBO information carrier

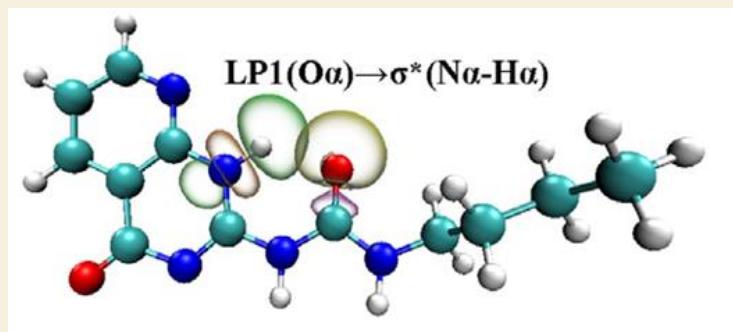
Gaussian provides a keyword *pop=saveNBO*, if you add it in your Gaussian input file, NBOs will be saved to checkpoint file instead of MOs. You can use corresponding .fch file as Multiwfn input file to view NBOs. If theoretical level of the task is HF or DFT, you should add *saveNBOene* in the first line of the .fch file; if post-HF method is used and *density* keyword has also been specified, you should add *saveNBOocc* in the first line of the .fch file, in this case Multiwfn will do some special treatments internally. However, if your aim is just viewing NBOs in main function 0, you can ignore this step.

Beware that when Gaussian storing the NBOs to checkpoint file, they may be automatically reordered. For example, you may see the information like below in the Gaussian output file:

```
Reordering of NBOs for storage:    7   8   3   1   2   4   6   5   9   38 ...
```

That means the 1st, 2nd, 3rd, 4th ... orbitals in the .chk/.fch file in fact correspond to the 7th, 8th, 3rd, 1st ... NBOs generated by the NBO module, respectively.

It is worth to note that if you use Multiwfn in combination with VMD, you can plot very pretty NBO isosurface maps, see my blog article "Using Multiwfn to plot NBO and related orbitals" (in Chinese, <http://sobereva.com/134>) for detail. Below is a map plotted by a Multiwfn user in his work *J. Mol. Graph. Model.*, **59**, 31 (2015).



4.0.3 Using Multiwfn+VMD to rapidly plot high-quality orbital isosurface map

Note: Chinese version of this tutorial is <http://sobereva.com/447> and <http://sobereva.com/449>.

Prologue

If one employs Multiwfn to export cube file for the orbitals of interest, and then render them as isosurface map in VMD (<http://www.ks.uiuc.edu/Research/vmd/>), very ideal orbital isosurface map can be obtained, the procedure has been detailedly described in my blog article "Using Multiwfn to visualize molecular orbitals" (in Chinese, <http://sobereva.com/269>). However, the procedure introduced in this article is somewhat lengthy, many manual operations are needed. In order to simplify the procedure as much as possible, here I show how to use scripts to very easily and quickly draw high-quality orbital isosurface map by combiningly using Multiwfn and VMD. In this section I only illustrate how to plot MOs, but the same procedure can also be applied for plotting other kinds of orbitals, however you need to properly modify the inputstream file (see below). If you do not know how to run Multiwfn in silent mode, I suggest you read Section 5.2 first so that you can better understand this section. Here I assume you are using Windows system, for Linux platform you should manually write corresponding script. The VMD program I used here is version 1.9.3.

Preparation work

Copying *showorb.bat* and *showorb.txt* from "examples\scripts" to the folder containing Multiwfn executable file.

The *showorb.bat* is a Windows batch process file, it is used to invoke Multiwfn to calculate grid data of wavefunction for selected orbitals and then move the exported cube files to VMD folder. You should manually edit this file to make the input file path corresponds to the actual path of input file, and then replace the VMD folder in this file with actual VMD folder in your machine.

The *showorb.txt* is inputstream file, each line corresponds a command needed to be inputted in the Multiwfn interactive interface. You should manually set the third line as indices of the orbitals you want to plot, for example, 10,20-23,28-30.

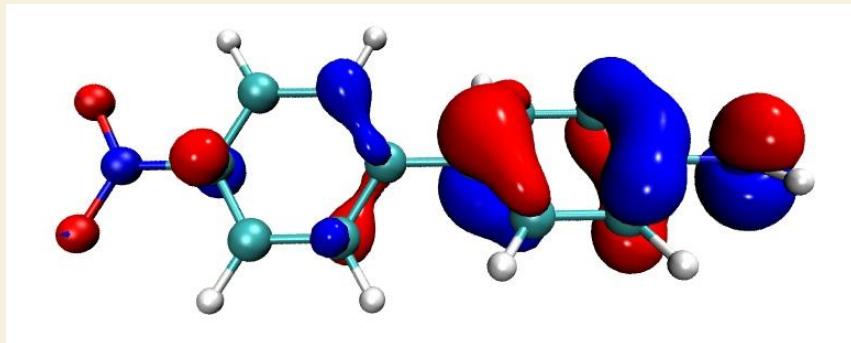
The *showorb.vmd* in "examples\scripts" is a VMD plotting script, you should copy it to VMD folder, and then add *source showorb.vmd* to the end of the *vmd.rc* file in VMD folder, so that the script will be automatically executed when VMD boots up. This script defines three customized commands:

- *orb i*: Used to load cube file of orbital *i* and show it as isosurfaces. The default isovalue is 0.05, you can change it by editing *showorb.vmd*
- *orbiso x*: Used to change the isovalue to *x*.

- `orbclean`: Used to delete all orbital cube files in VMD folder.

Example

Here we plot MOs for `examples\excit\D-pi-A.fchk`. Make sure that all preparation work has done, then edit the `showorb.bat`, replace the default input file `1.fch` with `examples\excit\D-pi-A.fchk`, and ensure that the actual VMD folder has been properly specified in this file. Then open `showorb.txt`, set the third line as `54-59`, so that we can visualize MOs from 54 to 59. Then double-click the `showorb.bat`, Multiwfn will be invoked to load input file, calculate and export grid data of wavefunction for the selected orbital. For e.g. orbital 54, the exported file will be named `orb000054.cub`. All the orbital cube files are then automatically moved to the VMD folder. After that, boot up VMD, input `orb 56` in VMD console window, then `orb000056.cub` will be loaded into VMD and drawn as isosurfaces:



In above map, positive and negative phases are represented as red and blue colors, respectively. "Glossy" material is used by default. If you want to change the color or material, you should enter "Graphics" - "Representation" and modify corresponding options. You can also change default color and material by modifying the `showorb.vmd` script.

If then you want to visualize another orbital, for example MO54, then simply input `orb 54` in the VMD console window.

If you want to change the isovalue to e.g. 0.02, simply input `orbiso 0.02`.

If after visualization, you want to clean all orbital cube files in the VMD folder, just input `orbclean`, then all `orb?????.cub` files will be deleted.

By default, "medium quality grid" (about 512000 points) is used to calculate orbital wavefunction, this is adequate for small and medium sized systems. However, for large systems, such as those consisted of one hundred of atoms or more, you must employ higher number of grid points. If you want to change the default grid to "high quality grid", you should set the fourth line in `showorb.txt` to 3. In addition, as mentioned in Section 4.0.1, for visualizing Rydberg orbitals you must increase extension distance of grid data. To do so, you should add

-10

12

between the third and fourth lines of `showorb.txt`, then extension distance will be increased from default value to 12 Bohr.

Plot state-of-the-art orbital isosurface map

If you want to obtain even better quality of orbital isosurface graph, just follow below procedure:

- (1) Plot an orbital in VMD as described above

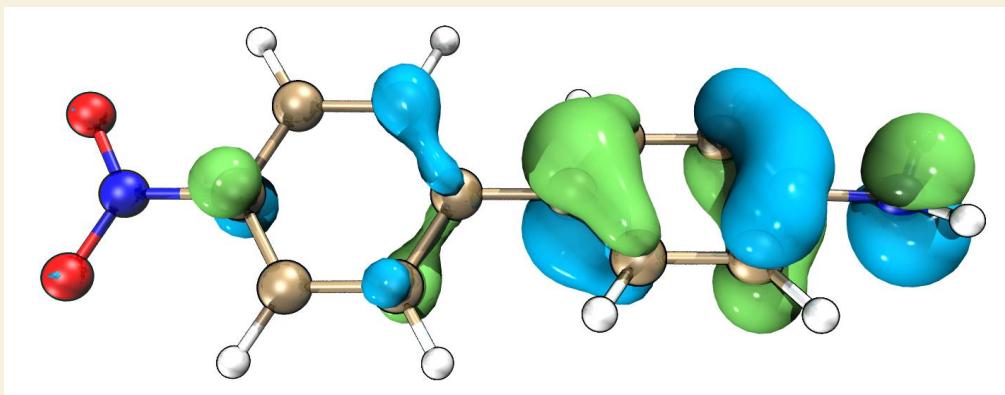
(2) Copy all content in *examples\VMRender.txt* to VMD console window to modify the plotting setting

(3) In VMD, select "File" - "Render" - "Tachyon", click "Start Rendering". Then *vmdscene.dat* will appear in VMD folder, it is input file of the Tachyon render.

(4) Copy the *examples\VMRender_full.bat* to VMD folder

(5) Double click the *VMRender_full.bat*, then the Tachyon render (*tachyon_WIN32.exe*) in VMD folder will be invoked to carry out render. After a while, *full.bmp* appears in the VMD folder, it is the produced image file.

The rendered image of MO56 of *examples\excit\D-pi-A.fchk* is shown below, the graph looks extremely good!



The rendering time is fairly long for large systems. For saving time, you can use the *VMRender_noshadow.bat* instead of the *VMRender_full.bat*, in this case no shadow effect will be observed in the resulting graph, while the rendering cost is correspondingly reduced.

Sometimes, especially for large system, the shadows casted by the transparent orbital isosurfaces make the graph look too dark, you can manually add *-shadow_filter_off* argument in the .bat file to disable this kind of shadow during rendering.

4.1 Calculate properties at a point

4.1.1 Show all properties of triplet water at a given point

In this example I illustrate how to calculate a wide variety of real space functions at a given point for triplet water. Boot up Multiwfn and input below commands

examples\H2O_m3ub3lyp.wfn

I // Main function function 1, show properties at a point

0,2,2,1,2 // X, Y, Z coordinate of the point

I // 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. If you are unable to fully understand the output, please read Sections 2.6 and 2.7 carefully, all terms in the output are very detailed described.

Density of all electrons: 0.4598301528E-02

Density of Alpha electrons: 0.2861566387E-02

4 Tutorials and Examples

```
Density of Beta electrons: 0.1736735141E-02
Spin density of electrons: 0.1124831246E-02
Lagrangian kinetic energy G(r): 0.3365319167E-02
Hamiltonian kinetic energy K(r): 0.1088761528E-03
Potential energy density V(r): -0.3474195320E-02
Energy density: -0.1088761528E-03
Laplacian of electron density: 0.1302577206E-01
Electron localization function (ELF): 0.1998328717E+00
Localized orbital locator (LOL): 0.1008002781E+00
Local information entropy: 0.3533635333E-02
Reduced density gradient (RDG): 0.2033111359E+01
Reduced density gradient with promolecular approximation: 0.2294831921E+01
Sign(lambda2)*rho: -0.4598301528E-02
Sign(lambda2)*rho with promolecular approximation: -0.3918852312E-02
Corr. hole for alpha, ref.: 0.00000 0.00000 0.00000 : -0.1251859403E-03
Source function, ref.: 0.00000 0.00000 0.00000 : -0.3565867942E-03
Wavefunction value for orbital 1 : 0.1536978161E-03
Average local ionization energy: 0.4664637535E+00
User defined real space function: 0.1000000000E+01
ESP from nuclear charges: 0.3453377860E+01
ESP from electrons: -0.3439063818E+01
Total ESP: 0.1431404144E-01 a.u. ( 0.3895049E+00 ev, 0.8982204E+01 kcal/mol)
```

Note: Below information are for electron density

Components of gradient in x/y/z are:

-0.7919856828E-03 -0.6903543769E-02 -0.6651181972E-02

Norm of gradient is: 0.9618959378E-02

Components of Laplacian in x/y/z are:

-0.3809549089E-02 0.1052804857E-01 0.6307272576E-02

Total: 0.1302577206E-01

Hessian matrix:

-0.3809549089E-02 0.1394394193E-02 0.1197923973E-02

0.1394394193E-02 0.1052804857E-01 0.1008387143E-01

0.1197923973E-02 0.1008387143E-01 0.6307272576E-02

Eigenvalues of Hessian: -0.3959672207E-02 0.1886890004E-01 -0.1883455778E-02

Eigenvectors(columns) of Hessian:

0.9964397434E+00 0.8076452238E-01 0.2418531975E-01

-0.4735415689E-01 0.7734993430E+00 -0.6320255930E+00

-0.6975257409E-01 0.6286301443E+00 0.7745700228E+00

Determinant of Hessian: 0.1407217564D-06

Ellipticity of electron density: 1.102344

All data are expressed in scientific notation, the value behind E is exponent, e.g. 0.6307272576E-02 corresponds to 0.006307272576.

In the line of "Corr. hole (correlation hole)" and "Source function", the so-called "ref" is the position of reference point, which is determined by *refxyz* parameter in *settings.ini*.

By default, the outputted wavefunction value corresponds to orbital 1, you can input for example *o6* to choose orbital 6.

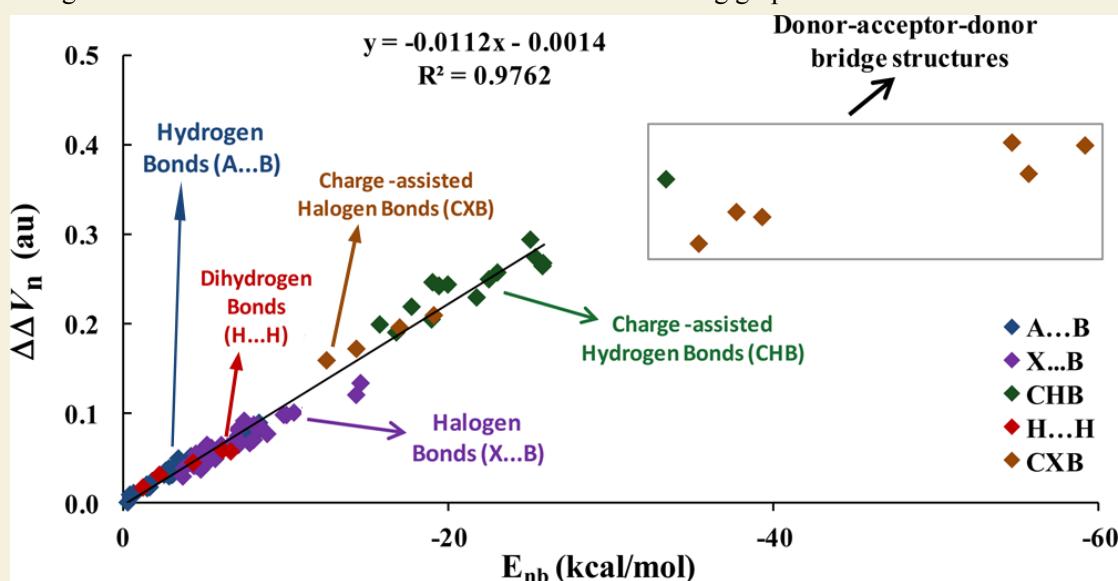
By default, the components of gradient and Laplacian as well as Hessian and its eigenvalue/eigenvectors are for electron density. You can input such as *f10* to choose the real space function with index of 10 (namely ELF), after that all of these quantities will be for ELF. If you want to inquire indices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

You can continue to input other coordinates, when you want to return to upper level menu, input *q*; If you want to exit the program, press “CTRL+C” button or directly close command-line window.

4.1.2 Calculate ESP at nuclear positions to evaluate interaction strength of $\text{H}_2\text{O}\cdots\text{HF}$

This is an advanced example, if you are not interested in weak interactions you may skip this section.

In *J. Phys. Chem. A*, **118**, 1697 (2014), Mohan and Suresh studied a batch of electrostatic dominated interacting systems, including hydrogen, halogen and dihydrogen bonds, all of them belong to electron donor-acceptor interactions, where donor stands for electron-rich moiety (Lewis base), while acceptor is electron-deficient moiety (Lewis acid). They fitted a surprisingly good linear equation to correlate $\Delta\Delta V_n$ index with interaction energy (E_{nb}) for all kinds of interactions, the R^2 is as high as 0.9762. Their results can be summarized as following graph



For an electrostatic dominated complex, assume that we can obtain $\Delta\Delta V_n$, then according to the equation shown in above graph, we can easily predict the interaction energies as

$$E_{\text{nb}} = -89.2857 \times \Delta\Delta V_n - 0.125$$

The $\Delta\Delta V_n$ is defined based on ESP at nuclear positions

$$\Delta\Delta V_n = \Delta V_{n-D} - \Delta V_{n-A} = (V_{n-D'} - V_{n-D}) - (V_{n-A'} - V_{n-A})$$

where $V_{n-D'}$ is the ESP at nuclear position of donor atom in complex environment, but the contribution due to nucleus of this donor atom is ignored. The only difference between V_{n-D} and $V_{n-D'}$ is that the former is calculated in monomer state, therefore $\Delta V_{n-D} = V_{n-D'} - V_{n-D}$ can be regarded as the change in ESP at nuclear position of donor atom due to presence of another molecule, which directly reflects strength of intermolecular interaction. The definition of $V_{n-A'}$ and V_{n-A} are identical to $V_{n-D'}$ and V_{n-D} , respectively, but they are calculated for acceptor atom.

In this example, we calculate $\Delta\Delta V_n$ for $\text{H}_2\text{O}\cdots\text{HF}$ and check if the interaction energy predicted based on $\Delta\Delta V_n$ is really closed to the accurately calculated interaction energy. In this complex the oxygen of H_2O is electron donor atom and hydrogen of HF is electron acceptor atom. Because the equation presented by Mohan and Suresh was fitted for specific calculation level, in order to properly use their equation, the calculation level we employed here is identical to them. The .wfn files used below were produced at MP4(SDQ)/aug-cc-pVTZ level at MP2/6-311++G** optimized geometries, these .wfn files and the corresponding Gaussian input files can be found in "examples\Vn" folder.

Note that if you are using relatively old revision of G09 and post-HF method is employed, "density" keyword is indispensable, otherwise the density in the resultant .wfn file will correspond to Hartree-Fock density. Besides, in G09 and G16, density cannot be produced at MP4 level, so we use MP4(SDQ) keyword instead (MP4 keyword is default to MP4(SDTQ), which is more accurate and but much expensive than MP4(SDQ)).

First we calculate $V_{n-A'}$ and $V_{n-D'}$. Boot up Multiwfn and input

examples\Vn\H2O-HF.wfn

I // Calculate properties at a point

a1 // Nuclear position of atom 1

From the output you can see

Total ESP without contribution from nuclear charge of atom 1:

-0.2228775074E+02 a.u. (-0.6064805E+03 ev, -0.1398579E+05 kcal/mol)

That means $V_{n-D'}$ is **-22.2877 a.u.** Then input *a5*, you will find $V_{n-A'}$ is **-0.9608 a.u.**

Next we calculate V_{n-D} . Reboot up Multiwfn and input below commands

?H2O.wfn // The symbol ? means the folder of the file we last time loaded

I

a1 // In H2O.wfn oxygen is atom 1

We find V_{n-D} is **-22.3339 a.u.** Then we calculate V_{n-A} . Reboot Multiwfn and input

?HF.wfn

I

a2 // In HF.wfn hydrogen is atom 2

The V_{n-A} is found to be **-0.9136 a.u.**

The $\Delta\Delta V_n$ is thus $-22.2877 - (-22.3339) - [-0.9608 - (-0.9136)] = 0.0462 + 0.0472 = 0.0933$ a.u.

Using the equation mentioned earlier, the interaction energy can be approximately predicted as $-89.2857 \times 0.0933 - 0.125 = -8.45$ kcal/mol, this value is quite close to the accurate interaction energies (-8.31 kcal/mol) obtained by Mohan and Suresh at MP4/aug-cc-pVTZ level with Counterpoise correction.

Generating wavefunction at MP4(SDQ)/aug-cc-pVTZ is quite time consuming even for small complex such as the system we studied here, thus it is important to find a calculation level that significantly saves computational time but without too much sacrifice in accuracy. For present system, based on the MP2/6-311++G** geometry, I tried using several levels to evaluate the $\Delta\Delta V_n$:

B3LYP/6-311+G**: 0.1021 a.u.

MP2/cc-pVTZ: 0.1052 a.u.

MP2/aug-cc-pVTZ: 0.0955 a.u.

B3LYP/aug-cc-pVTZ: 0.0985 a.u.

MP2/aug-cc-pVDZ: 0.0939 a.u.

B3LYP/aug-cc-pVDZ: 0.0980 a.u.

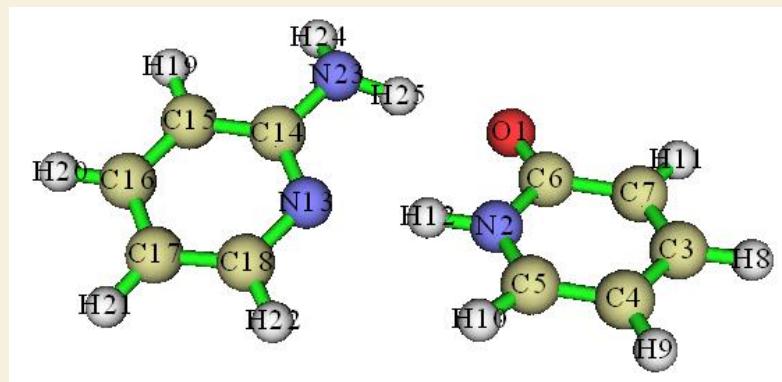
The $\Delta\Delta V_n$ produced at MP2/aug-cc-pVDZ (0.0939) is very close to the value we obtained above at MP4(SDQ)/aug-cc-pVTZ (0.0933), while the computational cost is reduced by factors of two. So, in practical studies, using MP2/aug-cc-pVDZ level to evaluate $\Delta\Delta V_n$ is a very ideal choice.

4.2 Topology analysis

Multiwfn is able to perform topology analysis for various kinds of real space functions, such as electron density, its Laplacian function, ELF, LOL, orbital wavefunctions, as well as user-defined functions. Four kinds of critical points (CPs) can be located and real space function values at these points can be easily obtained; topology paths linking CPs and interbasin surfaces can be generated. There are also many additional capacities, see Section 3.14 for details. Below I will present some practical applications to illustrate how to use this powerful module.

4.2.1 Atoms in molecules (AIM) and aromaticity analysis for 2-pyridoxine 2-aminopyridine

Topology analysis of electron density is a main ingredient of Bader's atoms in molecules (AIM) theory. In this example we will perform this kind of analysis for 2-pyridoxine 2-aminopyridine complex.



Boot up Multiwfn and input following commands

examples\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 entire absolute path

2 // Topology analysis

Then we search all critical points (CPs) by inputting below commands

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

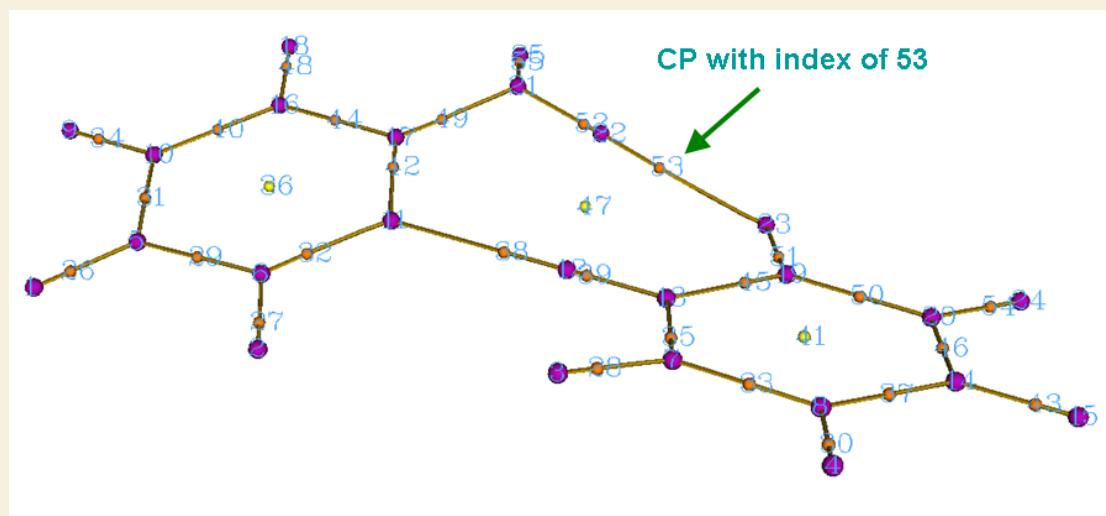
3 // Use midpoint of each atom pair in turn as initial guesses. Generally all (3,-1) CPs could be found, some (3,+1) or (3,+3) may also be found at the same time

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 show that Poincaré-Hopf relationship has been satisfied, that means all CPs may have been found. If this relationship is unsatisfied, then some CPs must be missing. From the GUI that popped up, we can see all expected CPs are presented, hence we can confirm that all CPs have been found.

Click "RETURN" button in 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 (Click "CP labels" at the right side of the GUI if index of the CPs are not shown):



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 53 and 38 corresponds to the bond critical point (BCP) of N-H.....O and N-H.....N hydrogen bond, respectively. It is worth to note that the label color of the CPs can be changed via “Set label color” option in the menu bar.

Now close the GUI window.

The topology analysis module provides many analysis options, now let us measure the distance between CP30 and the nucleus of H25. Select option -9, and input *c30 a25*, the result is 1.268379 Bohr. Then we measure the angle between C14-N13-H12, namely input *a14 a13 a12*, the result is 120.297432 degree. Now, we input *q* to return.

Evaluate H-bond binding energy

In *Chem. Phys. Lett.*, **285**, 170 (1998), the authors stated that for the hydrogen bond [X-H----O (X=C,N,O)], the relationship between bond energy E_{HB} and potential energy density $V(\mathbf{r})$ at corresponding BCP can be approximately described as

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

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 53, you will see value of all real space functions at this point

```
CP Position: 0.44887255865472 3.56434324597741 -0.10652884364257
```

```
CP type: (3,-1)
Density of all electrons: 0.3129478049E-01
Density of Alpha electrons: 0.1564739024E-01
Density of Beta electrons: 0.1564739024E-01
Spin density of electrons: 0.0000000000E+00
Lagrangian kinetic energy G(r): 0.2530207716E-01
Hamiltonian kinetic energy K(r): 0.8463666362E-03
Potential energy density V(r): -0.2614844379E-01
Energy density: -0.8463666362E-03
Laplacian of electron density: 0.9782284209E-01
Electron localization function (ELF): 0.1105388527E+00
... (Ignored)
```

The output indicates that the $V(\mathbf{r})$ at this BCP is -0.026148, therefore the hydrogen bond energy could be evaluated as $E_{\text{HB}} = -0.026148/2 \times 2625.5 = -34.3 \text{ KJ/mol}$.

Evaluating aromaticity based on CP properties

Here we use information entropy method to examine if the aminopyridine (the monomer at the left side of above graph) in the dimer can be regarded as an aromatic molecule. This method was proposed in *Phys. Chem. Chem. Phys.*, **12**, 4742 and is based on electron density at BCPs of a ring, see Section 3.14.6 for detail. First, we select option 20, and then input indices of the BCPs in the ring, namely 44,42,32,29,31,40, the outputted Shannon aromaticity index (SA) is 0.000812. The smaller the SA index, the more aromatic is the ring. In origin paper, $0.003 < \text{SA} < 0.005$ is chosen as the boundary of aromaticity/antiaromaticity. Since our result is much smaller than 0.003, we can conclude that aminopyridine is an aromatic molecule. The SA for 2-pyridoxine (the monomer at the right side of above graph) is 0.000865, hence shows slightly weaker aromaticity than aminopyridine.

Next, we calculate the curvature of electron density perpendicular to ring plane at RCP. In *Can. J. Chem.*, **75**, 1174, it was shown that more negative curvature implies stronger aromaticity. We first calculate the curvature for aminopyridine. Select option 21, input the index of the RCP (namely 36), and then input three atoms to define the ring plane, here we input 15,13,17 (also, you can input such as 14,16,18 and 13,15,18, because all of them correspond to the same plane). From the output we find the curvature is -0.0187. Then we calculate the curvature for 2-pyridoxine, namely input 41 and 2,7,4, the result is -0.0164. Comparison between the two curvatures again shows that aminopyridine has stronger aromaticity. In Multiwfn, aromaticity can also be measured in many other schemes, such as HOMA, FLU, PDI, ELF- π and multicenter bond order, they are collectively discussed in Section 4.A.3.

Generating interbasin surfaces

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, 38 and 37. Choose function 10, and input

```
53 // 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
```

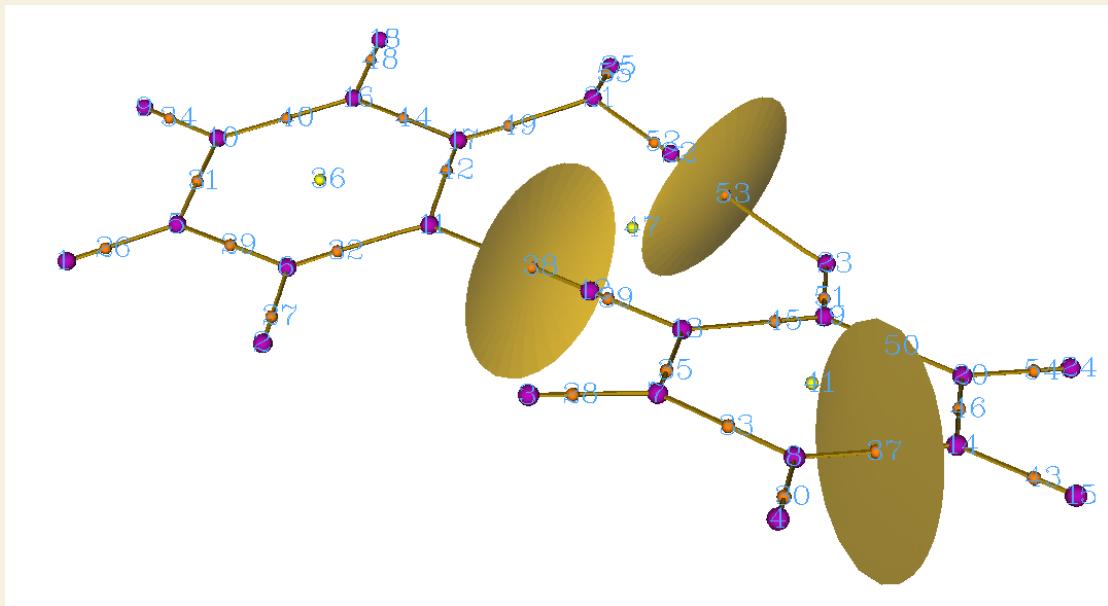
```
38
```

```
37
```

```
q // Return
```

Visualize the results by choose function 0, the graph will be shown as below. The three surfaces

are IBS.



In Section 4.20.1, we will use another important weak interaction analysis method NCI to further study this system.

You may feel that the current Multiwf GUI for showing CPs and topology paths is somewhat difficult to use for large system, since the system cannot be rotated completely smoothly, and sometimes index of interesting CP is difficult to be observed. [In Section 4.2.5 I will introduce how to use the powerful VMD program based on Multiwf outputs to very easily plot CPs and topology paths](#), in this case the graph is very pretty, the perspective is completely controllable, and index of interesting CPs can be easily found out.

There are two important points regarding AIM topology analysis I would like to mention here, though they are not related to present example.

Tip 1: What should I do if some CPs of electron density are missing?

For small systems, commonly we can check whether all CPs have been located by simply enter the GUI and visualize the distributions of the CPs. There is also an useful equation named Poincaré-Hopf relationship. For isolate system, the relationship is

$$n_{\text{NCP}} - n_{\text{BCP}} + n_{\text{RCP}} - n_{\text{CCP}} = 1$$

If all CPs have been found, this relationship must be satisfied, but the satisfaction of this relationship does not necessarily mean all CPs have been found. If the Poincaré-Hopf relationship is unsatisfied, then some CPs must be missing.

Sometimes, you may find some expected CPs are not successfully located after searches. There are two reasons may cause this problem: (1) The position of initial guesses are not close enough to the CPs (2) The default CP searching parameters are not well-suited for present case. There are some commonly used ways to solve this problem. More detailed descriptions may be found in Section 3.14.2.

a) If you have tried options 2~5 and some CPs are not located, try to use suboption -1 of option 6. This searching mode is powerful but expensive, which by default places 1000 guessing points

within in a spherical region around every atom. If after repeating this mode several times the missing CPs are still unable to be located, it is highly possible that the reason is due to the inappropriate searching parameters rather than the positions of guessing points.

b) If some BCPs are unable to be located, you can enter option -1, set the scale factor of stepsize to 0.5, and then try again

c) NCPs of very heavy atoms are difficult to be located, because the peak of electron density at nucleus in these cases are very sharp, thus under default parameters the searching algorithm is difficult to capture the NCPs. In order to locate them, you can enter option -1, loose the criteria for gradient-norm and displacement convergences by several factors, and then try to use option to 2 search the NCPs again. If the NCPs are then successfully found, do not forget to recover the convergence criteria. In fact, since NCP of heavy atoms are almost exactly located at nucleus position, you can directly enter option -4 and choose suboption 3 to add NCPs artificially at the corresponding nuclear positions.

d) If some missing CPs are far away from atoms, for example, the CCP at the center of a very large cage or tube system, try to enter option -1 and decrease the criterion for determining singularity of Hessian matrix via option 8 by several factor, and then search the CPs again.

Tip 2: Describing electron density of very heavy atoms

Very heavy atoms (heavier than Kr) bring much more computational burden than light atoms, and relativistic effect is non-neglectable. There are two different ways to describe them.

(1) Using pseudo-potential (PS): As mentioned in Section 2.5, if PS is employed but the .wfx file produced by Gaussian is used as input file, the EDF (electron density functions) field in the .wfx file by default will be loaded into Multiwfn, which represents the inner-core electron density. For other type of input files, such as .wfn, .fch, .molden and .gms, by default Multiwfn automatically loads proper EDF information from built-in EDF library.

When EDF information is provided, all CPs of electron density can be properly located and artificial CPs will never occur, the bond paths emitting from BCPs can connect to NCPs normally, all CP properties that solely based on electron density will be reasonable. Although large core PS can be employed without problems in this situation, I still recommend using small core PS, because the accuracy of resultant CP positions and properties must be better than using large core PS.

	Lanl1	Lanl2, Lan2TZ/08	SDD	cc-pVnZ-PP, def2- series	SBKJC
Main groups	Large	Large	L/S (optional)	Small	Large
Transition metals	Large	Small	L/S (optional)	Small	Small

If you decide not to utilize EDF information (see "readEDF" and "isupplyEDF" in settings.ini for detail), evidently it is impossible to find out (3,-3) CP at nuclear position, and accordingly, the bond paths emitted from BCP will be unable to connect to the nucleus. Instead, you may find (3,+3) at nucleus position due to the vacancy of inner-core density, and a lot of CPs in different types will appear around the nucleus, this is because the electron density no longer decreases exponentially from nucleus, so the topology structure of electron density becomes quite complicated. However, you can simply ignore those irrelevant CPs but only focus on the BCPs that you are really interested in.

(2) Using all-electron basis set with relativistic Hamiltonian: This is the most expensive but most accurate solution for representing electron structure of heavy atoms. Only considering scalar

relativistic effect is totally enough for AIM analysis. DKH2 Hamiltonian is a very good choice (in Gaussian, simply using *int=DKH2* keyword to employ it, beware that basis set optimized for DK calculation must be used, e.g. cc-pVDZ-DK).

For more discussions, please consult my post "Some explanation of performing wavefunction analysis under pseudo-potentials" (in Chineses, <http://sobereva.com/156>).

4.2.2 LOL topology analysis for acetic acid

A brief of localized orbital locator (LOL) has been given in Section 2.6, and its graphical interpretation will be illustrated in later examples. In this example, we will locate CPs and generate topology paths of LOL for acetic acid. With completely identical procedure shown in this example, you can also study topology character for electron localization function (ELF) as well as Laplacian of electron density.

Boot up Multiwfn and input following commands

```
examples\acetic_acid.wfn
2 // Enter topology analysis module
-11 // Select a real space function
10 // Localized orbital locator (LOL)
```

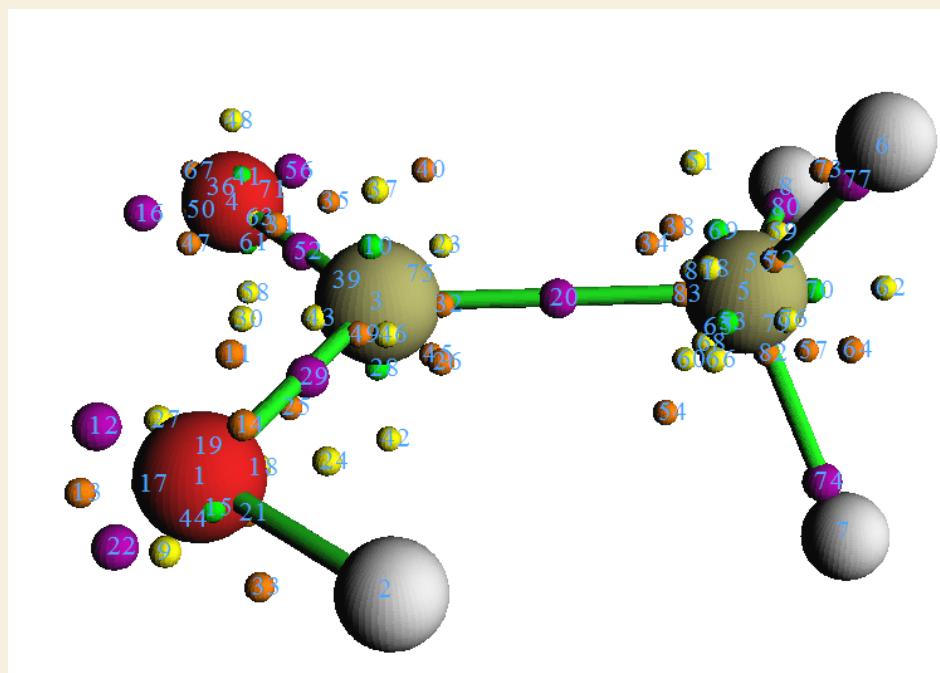
Notice that the distribution feature of LOL is much more complex than electron density, so it is very difficult to locate all of its CPs. Fortunately, in general only a small subset of CPs is what we are interested in, the search of CPs can be terminated once all expected CPs have been found.

We first use option "2 Search CPs from nuclear positions" to locate the CPs that very close to nuclei. However the positions of CPs in other regions are somewhat unpredictable, hence a lot of initial guess points have to be randomly scattered around each atom to try to locate those CPs.

```
6 // In this searching mode, initial guessing points will randomly scattered in a sphere, the
sphere center, radius, number of points and so on can be defined by users, this time we leave the
default value unchanged
```

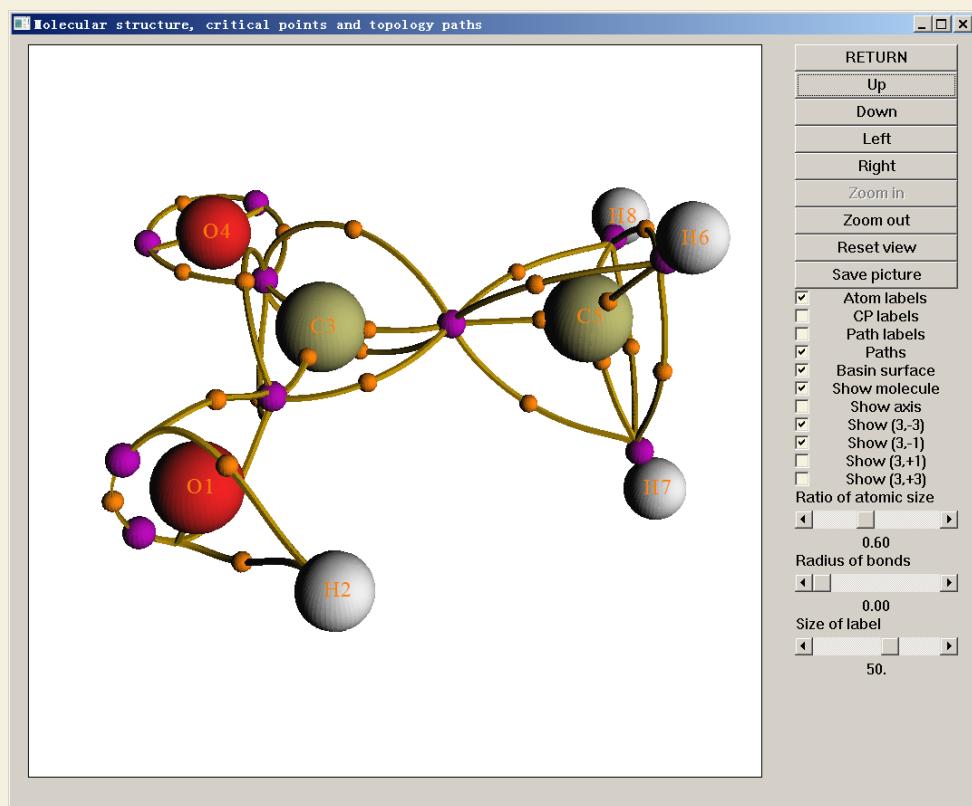
```
-1 // Use each nucleus as sphere center in turn to search CPs. Since there are 8 atoms, and
guessing points in each sphere is 1000, Multiwfn will try to search CPs from 8*1000 guessing points.
Of course, the more the guessing points you set, the larger probability all CPs could be found in this
search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

```
-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 unneccessary. 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.



4.2.3 Plot real space function along bond path

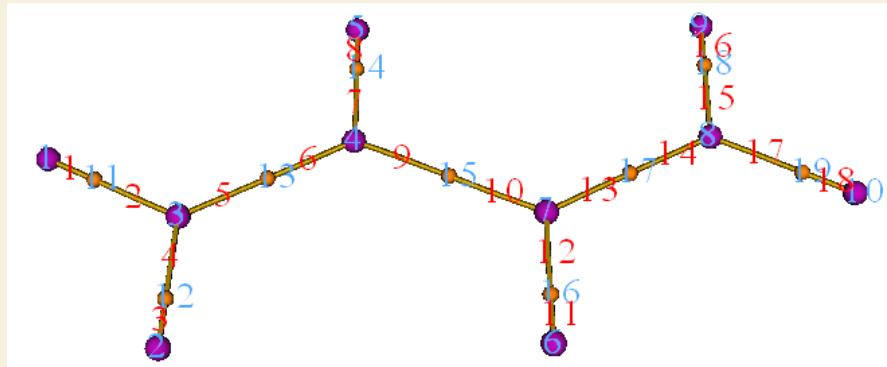
All real space functions that supported by Multiwfn could be easily plotted along topology paths. In this example we plot ellipticity of electron density along bond path of boundary C-C bond of butadiene.

First open settings.ini file and change "iuserfunc" parameter to 30, because the 30th user-defined function corresponds to electron density ellipticity, see Section 2.7 for detail.

Then boot up Multiwfn and input below commands:

```
examples\butadiene.fch
2 // Topology analysis
2 // Search nuclear critical points from nuclear positions
3 // Search bond critical points from midpoint of atom pairs
8 // Generate bond path
0 // Enter GUI window to visualize result
```

Clicking "Atom labels" and "Path labels" buttons at right side of the GUI window, then we can find paths 5 and 6 collectively constitute the bond path of a boundary C-C bond:



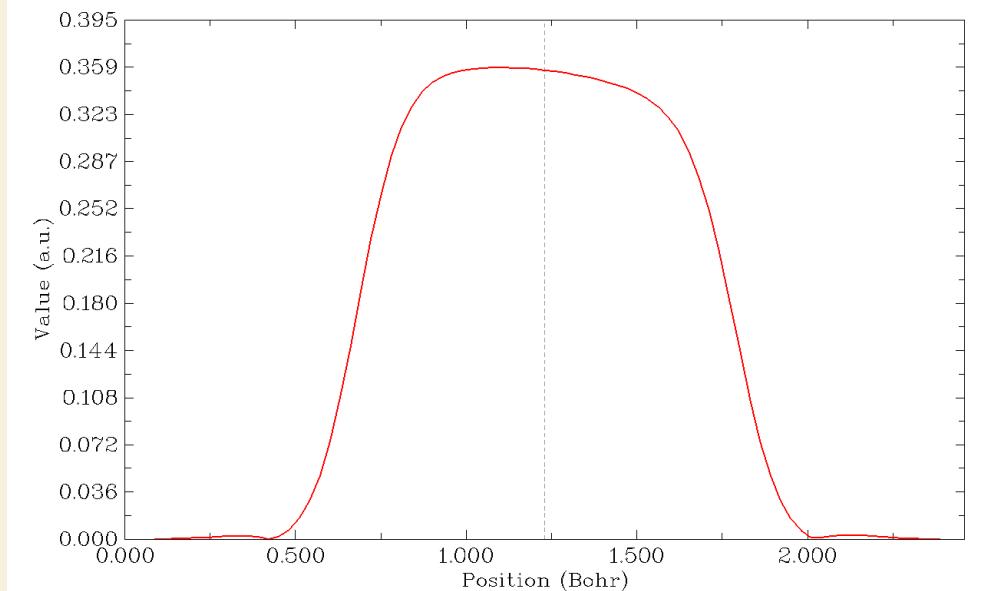
Clicking "RETURN" to close the window and then input

-5 // Various operations on paths

7 // Calculate and plot specific real space function along a path

5,6 // The index of the paths (in fact, you can also equivalently input c13 here)

100 // User-defined function, which corresponds to ellipticity of electron density currently



The curve of electron density ellipticity along the boundary C-C bond path immediately shows on the screen, the dashed line denotes the position of bond critical point. In the plot, the left and right corner correspond to CP3 and CP4, respectively. At the same time, the raw data of the curve are shown on the command-line window and you can copy them out, so that the map can be further analyzed or replotted in third-part plotting tools such as Origin.

From the graph it is clear that the electron density ellipticity is positive in the middle region of the bond path, exhibiting the double-bond character of the C-C bond.

You can also plot other real space functions such as ELF and kinetic energy density along bond paths, please have a try.

4.2.4 Decompose properties at a critical point as orbital contributions

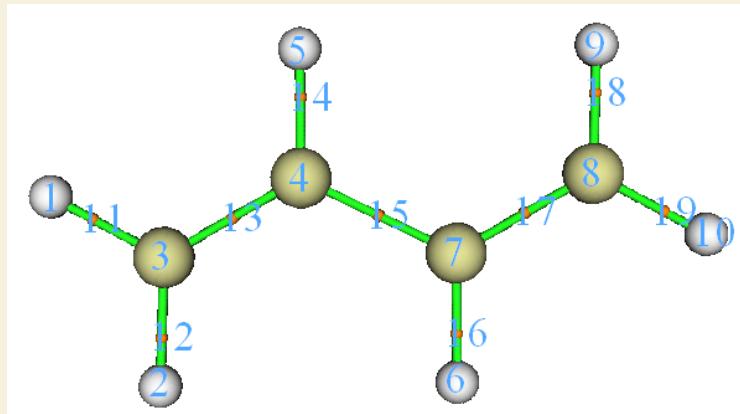
Many real space functions can be exactly or approximately decomposed as orbital contributions. If occupation number of all orbitals except for an orbital i is set to zero, then the calculated function value just corresponds to contribution of orbital i .

Multiwfn is able to decompose any real space function as orbital contributions at any point, this feature is supported by both main function 1 and main function 2; in the former the position of the point can be directly inputted by user, while in the latter the point can be selected as one of found CPs. In this section I will use 1,3-butadiene as example to illustrate this feature.

We will first check which MOs have evident contribution to BCP corresponding to the boundary C-C bonds. Boot up Multiwfn and input

examples\butadiene.fch

```
2 // Topology analysis
2 // Search nuclear critical points
3 // Search BCPs
0 // Visualize CPs
```



As can be seen, CP13 and CP17 are the BCPs of boundary C-C bonds. Close the GUI and then input

```
7 // Show properties at a CP
13d // Decompose properties of CP13
I // The real space function to be decomposed is electron density
[Press ENTER button] // Take all occupied orbitals into account, but only print ten orbitals
having largest contributions
```

You will see below output

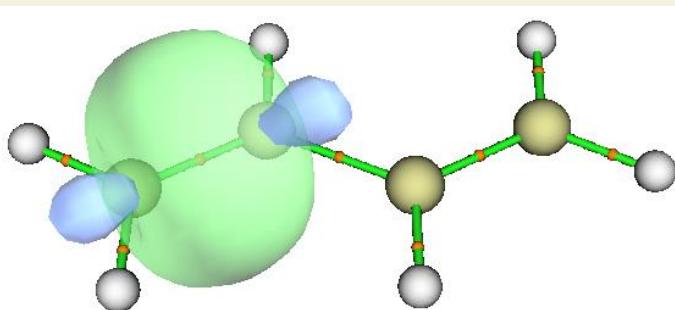
Contribution from orbital	11 (occ= 2.000000):	0.107469 a.u. (31.34%)
Contribution from orbital	6 (occ= 2.000000):	0.085220 a.u. (24.85%)
Contribution from orbital	9 (occ= 2.000000):	0.061997 a.u. (18.08%)
Contribution from orbital	5 (occ= 2.000000):	0.059119 a.u. (17.24%)
Contribution from orbital	12 (occ= 2.000000):	0.011205 a.u. (3.27%)
Contribution from orbital	10 (occ= 2.000000):	0.008898 a.u. (2.60%)
Contribution from orbital	7 (occ= 2.000000):	0.004864 a.u. (1.42%)
Contribution from orbital	8 (occ= 2.000000):	0.003807 a.u. (1.11%)
Contribution from orbital	2 (occ= 2.000000):	0.000070 a.u. (0.02%)
Contribution from orbital	1 (occ= 2.000000):	0.000070 a.u. (0.02%)
Sum of above values:		0.34286855 a.u. (100.00%)
Exact value:		0.34286855 a.u.

Clearly, electron density at this BCP is simultaneously contributed by many MOs, the largest contribution is 31.3%. What will happen if we transform the MOs to localized MOs (LMO)? (see Section 3.21 for introduction about orbital localization and LMO). To examine this, return to main

function menu, then input

```
19 // Orbital localization
1 // Only localize occupied orbitals
2 // Enter topology analysis function again. We do not need to redo topology analysis, since
all topology information are retained when you exit topology analysis module
7 // Show properties at a CP
13d // Decompose properties of CP13
1 // The real space function to be decomposed is electron density
[Press ENTER button]
You will see
Contribution from orbital 11 (occ= 2.000000): 0.339266 a.u. ( 98.95% )
Contribution from orbital 6 (occ= 2.000000): 0.001072 a.u. ( 0.31% )
Contribution from orbital 7 (occ= 2.000000): 0.000630 a.u. ( 0.18% )
...[ignored]
Sum of above values: 0.34286855 a.u. ( 100.00% )
Exact value: 0.34286855 a.u.
```

As can be seen, currently only one orbital, namely LMO11 has remarkable contribution to the BCP. This is what we expected, since in the LMO framework, each chemical bond is commonly mainly represented by only one or very few number of LMOs. You can visualize the LMO11 using main function 0:



It is clear that LMO11 fully corresponds to the boundary C-C σ -bond, this is why properties such as electron density of the corresponding BCP is solely dominated by LMO11.

Next, let us check which LMOs have nonnegligible contribution to the point above 1 Bohr of the BCP corresponding to the boundary C-C bond. The index of the BCP is 13, when you choose option 0, you can find its coordinate in console window:

Index	XYZ Coordinate (Bohr)	Type
...[ignored]		
11 -2.227945085	-3.985434919	0.000000000 (3,-1)
12 -0.041512516	-3.979859852	0.000000000 (3,-1)
13 -1.131478160	-2.047305989	0.000000000 (3,-1)
...[ignored]		

Evidently, the point above 1 Bohr of CP13 should be (-1.131,-2.047,1.0). Input below command

```
-10 // Return to main menu
1 // Print various properties at a given point
d // Decompose to orbital contributions
-1.131,-2.047,1.0
```

I // The unit of inputted coordinate is Bohr

I // Decompose electron density

[Press ENTER button]

Then you will see below information

Contribution from orbital 11 (occ= 2.000000): 0.110266 a.u. (66.33%)

Contribution from orbital 15 (occ= 2.000000): 0.055354 a.u. (33.30%)

Contribution from orbital 1 (occ= 2.000000): 0.000215 a.u. (0.13%)

...[ignored]

Exact value: 0.16623908 a.u.

It can be seen that, not only the LMO11, but also LMO15 has evident contribution. If you inspect this orbital in main function 0, you will find it corresponds to π bond of the boundary C-C bond.

Via similar way, we then decompose electron density of the point above 1 Bohr of the BCP corresponding to the middle C-C bond, the result is

Contribution from orbital 13 (occ= 2.000000): 0.097450 a.u. (78.20%)

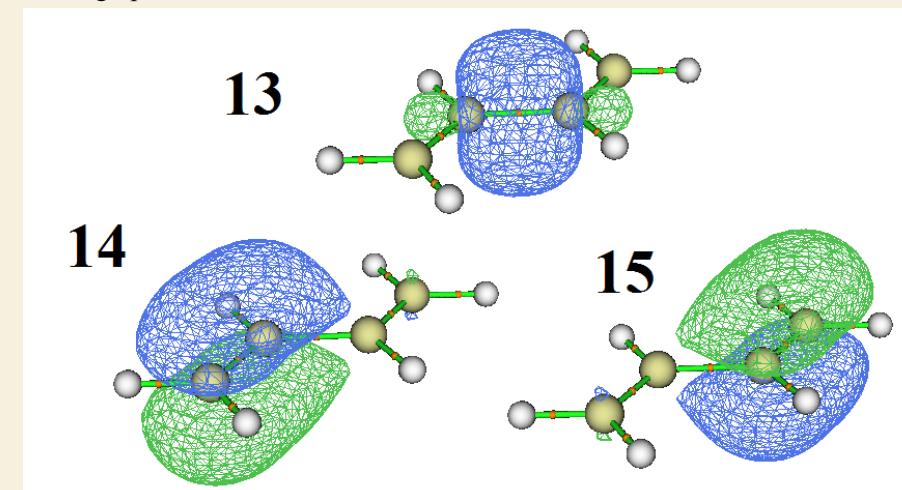
Contribution from orbital 14 (occ= 2.000000): 0.013348 a.u. (10.71%)

Contribution from orbital 15 (occ= 2.000000): 0.013348 a.u. (10.71%)

...[others are not shown due to negligible contribution]

Exact value: 0.12461415 a.u.

Isosurface graph of the three LMOs are shown below.



Although both the LMO 14 and LMO 15 mainly correspond to π bond of the marginal C-C bonds, they have 10.7% contribution to the position above 1 Å of the middle C-C bond, implying that the middle C-C bond must also have π character, though much weaker than the boundary C-C bonds.

4.2.5 Easily plot high quality AIM topology map in VMD visualization program based on Multiwfn outputs

Note: Chinese version of this tutorial is <http://sobereva.com/445>.

In Section 4.2.1 I have shown how to carry out AIM topology analysis for 2-pyridoxine 2-aminopyridine complex, in present section I will show how to very easily render the located CPs and generated topology paths in the very powerful and freely available VMD program

(<http://www.ks.uiuc.edu/Research/vmd/>). The map is very pretty, and you can easily find index of interesting CPs. The whole process is highly automatic, because it is based on pre-provided Windows batch process file and VMD plotting script. If you do not know how to run Multiwfn in silent mode, I strongly suggest you read Section 5.2 first, so that you can fully understand how the batch process file works. If you manually write a similar shell script, this method can also be realized under Linux platform.

We need to do a few preparation works first. Copy *AIM.bat* and *AIM.txt* from "examples\scripts" to the folder containing Multiwfn executable file. Edit the *AIM.bat*, modify the default VMD folder to actual VMD folder on your machine. Then copy the VMD plotting script *AIM.vmd* to VMD folder, and add *proc aim {} {source AIM.vmd}* to the end of the *vmd.rc* file in VMD folder so that then you can activate this script by simply inputting command *aim*.

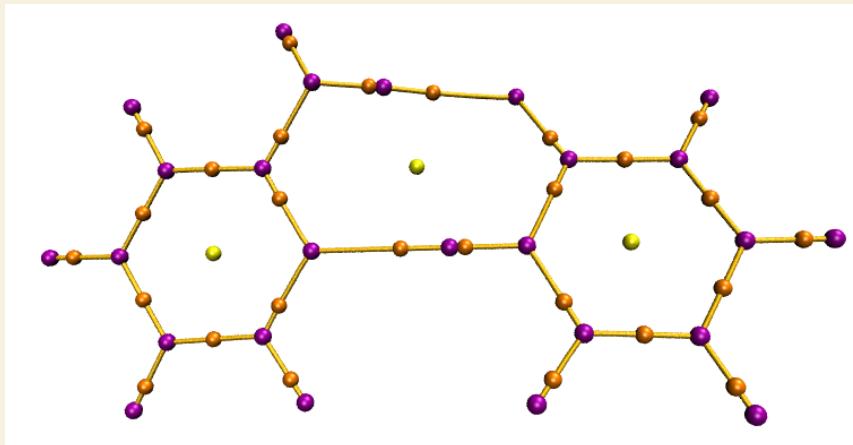
The *AIM.txt* is an input stream file for running Multiwfn in silent mode, it does below things:

- (1) Carrying out standard AIM analysis (locating CPs via options 2,3,4,5 in succession and generating topology paths)
- (2) Exporting the CPs and paths as *CPs.pdb* and *paths.pdb* in current folder
- (3) Calculate all properties except for ESP at all located CPs and export the result to *CPprop.txt* in current folder
- (4) Exporting the structure of present system as *mol.pdb* in current folder

In this example, we copy the *examples\2-pyridoxine_2-aminopyridine.wfn* to the folder containing Multiwfn executable file, modify the input file name in the *VMD.bat* as *2-pyridoxine_2-aminopyridine.wfn*, then double-click the *VMD.bat*, this batch process file will invoke Multiwfn to carry out analysis according to the commands in *VMD.txt* for this .wfn file. After a while, *CPprop.txt* is generated in this folder, the meantime generated *mol.pdb*, *CPs.pdb* and *paths.pdb* are automatically moved to the VMD folder.

Now boot up VMD and input command *aim* in VMD console window, you will immediately see below graph.

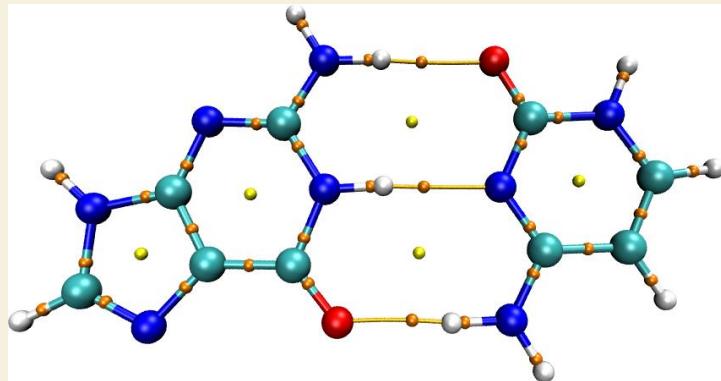
Notice that in order to gain slightly better effect, I used the built-in Tachyon render to obtain below graph, namely selecting "File" - "Render", change to "Tachyon (internal, in-memory rendering)" and click "Start Rendering" button (The resulting file is in .tga format, you need to use advanced image viewer to view it, such as IrfanView, which is freely available at <https://www.irfanview.com>)



It is also possible to simultaneously show molecular structure on the map. Double click the "D" shown in below screenshot to make it become black:

ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	A	D	F		CPs.pdb	55	1	0
1	A	D	F		paths.pdb	2164	1	0
2	T	A	D	F	mol.pdb	25	1	0

Then the molecular structure will be visible:



The CP indices can be labelled on the graph easily. The *AIM.vmd* script defined a command named *labcp* to do this, the usage is:

labcp [type] [label size] [offset in X] [offset in Y]

The “type” could be “all”, “no”, “3n3”, “3n1”, “3p1”, “3p3”. The “offset in X” and “offset in Y” are used to define position offset of the labels, they are default to -0.1 and 0.0, respectively.

Examples:

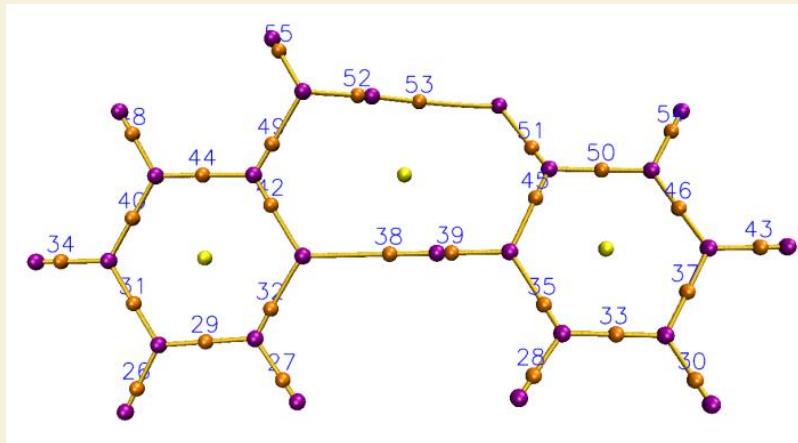
labcp all: Labelling index of all CPs

labcp no: Removing all CP labels

labcp 3n1 1.3: Labelling all (3,-1) CPs with size of 1.3

labcp 3p3 1.8 -0.05 0.1: Labelling all (3,+3) CPs with size of 1.8, the position offset in X and Y are set to -0.05 and 0.1, respectively

For present system, we input *labcp 3n1 1.5 -0.1 0.1* in VMD console window, you will see

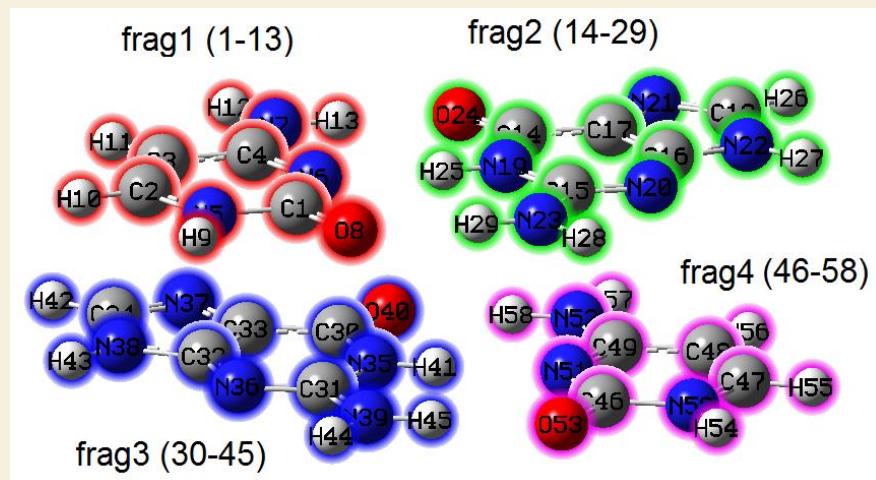


According to the labels, you can easily check properties of all (3,-1) CPs by checking corresponding entry in the *CPprop.txt*.

Hint: If you want to reduce calculation cost of *VMD.bat*, you can remove the fourth and fifth lines of *VMD.txt*, in this case Multiwfnn will only try to locate CPs starting from nuclear positions and midpoints of atom pairs. However, this treatment may cause missing of some CPs.

4.2.6 Topology analysis in special regions: G-C...G-C base pair as an example

Owing to the extreme flexibility of Multiwfn, it is possible to perform topology analysis only in some special regions that you are really interested in. In this section, the below G-C...G-C base pair will be used as instance. The system can be regarded as consisting of four fragments. Its .wfn file can be downloaded at <http://sobereva.com/multiwfn/extrafiles/GCGC.zip>.

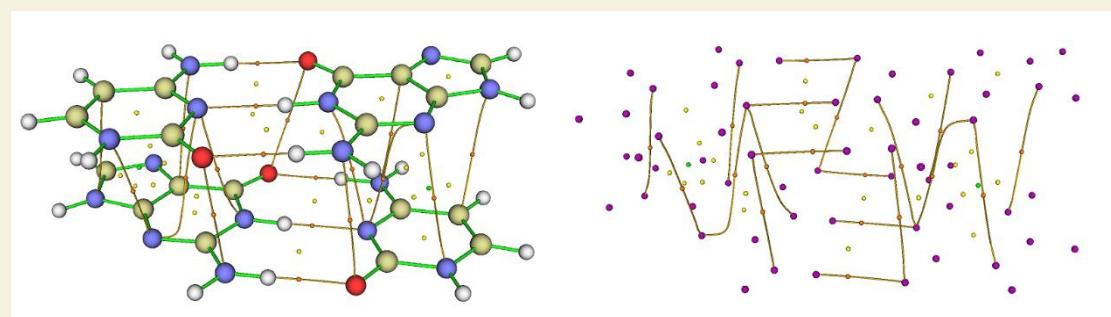


(1) Perform AIM analysis only in weakly interacting regions

First I show how to only locate CPs and generate bond paths in weakly interacting regions, where electron density must be relatively low. Boot up Multiwfn and input

```
GCGC.wfn
2 // Topology analysis
2 // Search CPs from nuclear positions
-1 // Set CP searching parameters
9 // Set value range for reserving CPs:
0,0.1 // Only CPs with density within 0~0.1 a.u. (i.e. relatively low density) will be reserved
during searching
0 // Return
3 // Search CPs from midpoint of atom pairs
8 // Generating the paths connecting (3,-3) and (3,-1) CPs
```

Now choose option 0 to visualize the result, see below. The left and right graphs are actually the same, but molecule structure is hidden in the right graph. It is clear that only BCPs as well as accompanying bond paths corresponding to weak interactions have been generated, while BCPs and bond paths corresponding to chemical bonds are not obtained.



If you are not interested in RCPs (ring CPs, yellow spheres) and CCPs (cage CPs, green spheres), you can easily delete them. Input below commands:

```
-4 // Modify or export CPs
2 // Delete some CPs
5 // Delete all (3,+1) CPs
6 // Delete all (3,+3) CPs
0 // Return
0 // Return
```

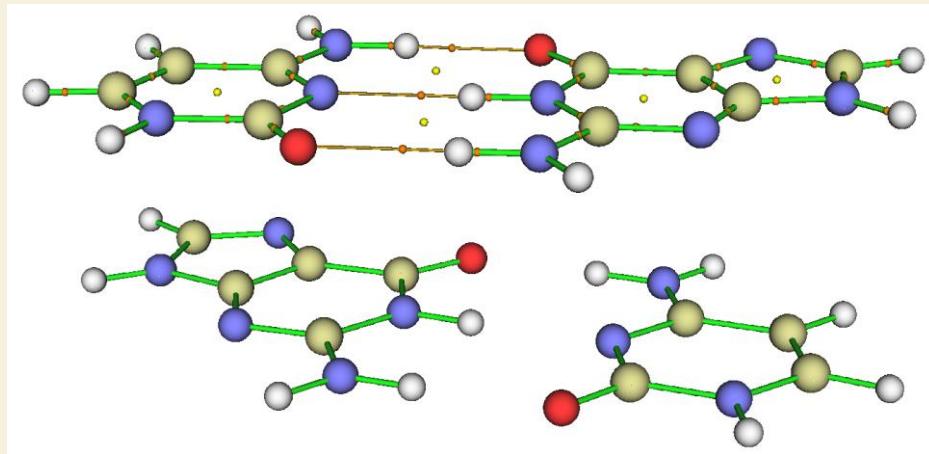
Then if you choose option 0 again to visualize the result, you will find RCPs and CCPs have disappeared.

(2) Perform AIM analysis only in local spatial region

Now I illustrate how to perform AIM topology analysis only for fragments 1 and 2. Boot up Multiwfn and input

```
GCGC.wfn
2 // Topology analysis
-1 // Set CP searching parameters
10 // Set the range of atoms considered for searching modes 2, 3, 4, 5
1-29 // The atomic index range of fragments 1 and 2
2 // Set scale factor of stepsize. We modify this because under default setting some BCPs
corresponding to C-H and N-H of present system may be missing
0.5 // Stepsize will scaled by 0.5
0 // Return
2 // Search CPs from nuclear positions
3 // Search CPs from midpoint of atom pairs
8 // Generating the paths connecting (3,-3) and (3,-1) CPs
```

Choose option 0 to visualize result, you will see below graph. Clearly, only CPs and bond paths related to fragments 1 and 2 are generated, this is what we expected.



(3) Only retaining bond paths and corresponding BCPs connecting two specific fragments

Sometimes we only want to study interfragment interaction between two specific fragments and hope that all irrelevant bond paths and BCPs could be removed to make the graph clearer. Although you can manually delete undesired BCPs and bond paths manually via corresponding suboptions in options -1 and -2, respectively, the process is usually tedious. Fortunately, in Multiwfn there is a special option aiming for realizing this purpose. Below I will illustrate how to only retain

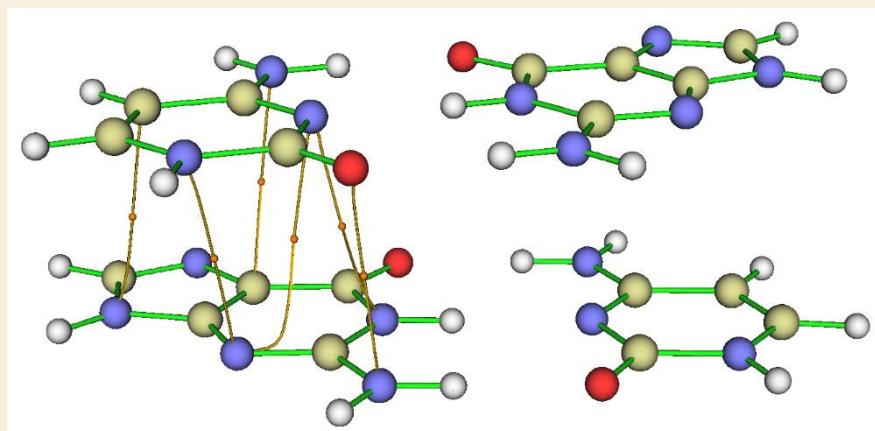
bond paths and corresponding BCPs connecting fragments 1 and 3 while removing all other BCPs and bond paths.

Boot up Multiwfn and input

GCGC.wfn

```
2 // Topology analysis
2 // Search CPs from nuclear positions
3 // Search CPs from midpoint of atom pairs
8 // Generating the paths connecting (3,-3) and (3,-1) CPs
-5 // Manipulate paths
8 // Only retain bond paths (and corresponding BCPs) connecting two specific fragments
while delete all other bond paths and BCPs
I-13 // Atomic indices in fragment 1
30-45 // Atomic indices in fragment 3
y // Also delete BCPs in other bond paths
```

Then we manually delete all (3,+1) and (3,+3) CPs as described above, and then visualize result, you will see



Evidently, only the BCPs and bond paths characterizing the stacking interaction between fragments 1 and 3 are retained, the graph looks very clear, this is what we desired.

4.3 Output and plot various properties in a line

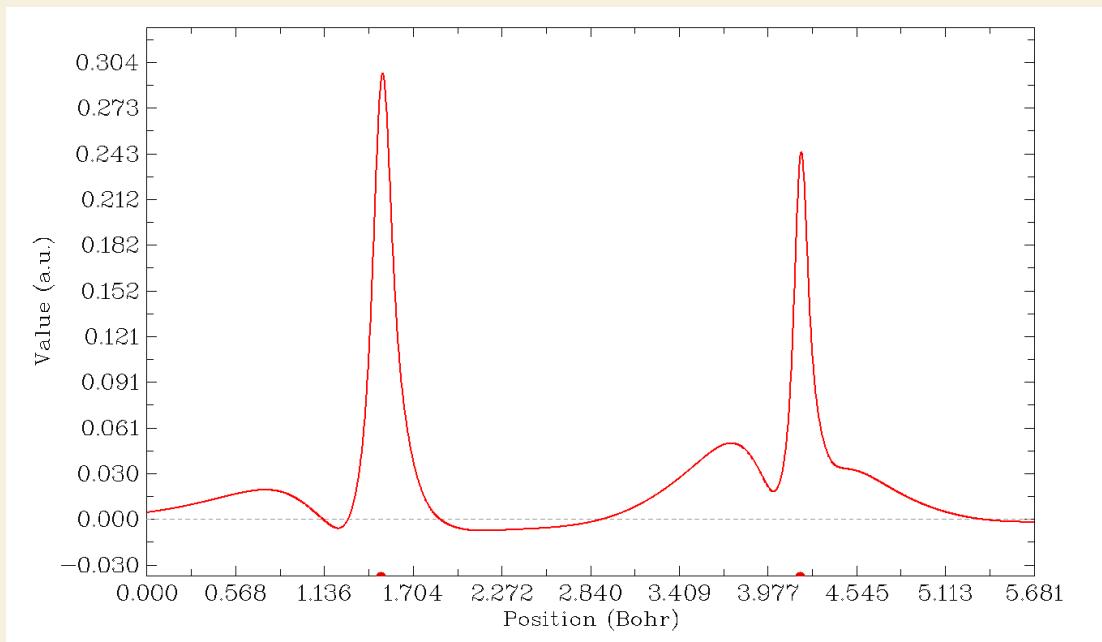
4.3.1 Plot the spin density curve of triplet formamide along carbon and oxygen atoms

Boot up Multiwfn and input following commands

examples\formamide-m3.wfn

```
3 // Main function 3, plot real space function along a line
5 // Spin density
I // Defining the line by nuclear coordinate of two atoms
1,6 // Indices of the two atoms, carbon and oxygen atoms correspond to 1 and 6 in present
example, respectively
```

The graph shows up immediately:



The X-axis corresponds to the position in the line you defined, the dashed line corresponds to the position of $Y=0$, the left and right red circles denote the position of the first and the second atoms you selected. After clicking right mouse button on the graph to close it, a new menu appears on command-line window, you can save graphic file, adjust the range of Y-axis, export X-Y data points 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
Totally found 3 local minimum, 4 local maximum

```

Using the same procedure illustrated above, you can plot curve map for any real space function supported by Multiwfn, please have a try.

4.3.2 Study Fermi hole and Coulomb hole of H₂

This is a relatively advanced example, you can skip this section if you are a newbie of quantum chemistry.

In this example we will plot correlation hole (Fermi hole and Coulomb hole) along the axis of H₂. This is an advanced topic, if you are not familiar with the concept of correlation hole, please consult the discussion in part 17 of Section 2.6.

Hartree-Fock wavefunction is capable to exhibit Fermi correlation, but Coulomb correlation is completely omitted. In this case exact Fermi hole can be calculated and plotted by Multiwfn. If Coulomb hole is needed to be analyzed, then post-HF wavefunction must be employed. In current

version, Multiwfn is able to evaluate and plot approximate Fermi hole and Coulomb hole for post-HF wavefunctions by Müller approximation, which makes use of natural orbitals to mimic exact pair density. Although the approximation is introduced, the result is generally at least qualitative correct.

First, we optimize and generate .wfn file for H₂ under CCSD/cc-pVTZ level. CCSD is the highest level of wavefunction that can be generated by Gaussian program (If higher level of wavefunction is needed, you must resort on other programs, see Section 4.A.8 for details). The file has already been provided as *examples\H2_CCSD.wfn*. The coordinate of H1 is (0.0, 0.0, 0.7016) Bohr, the coordinate of H2 is (0.0, 0.0, -0.7016) Bohr.

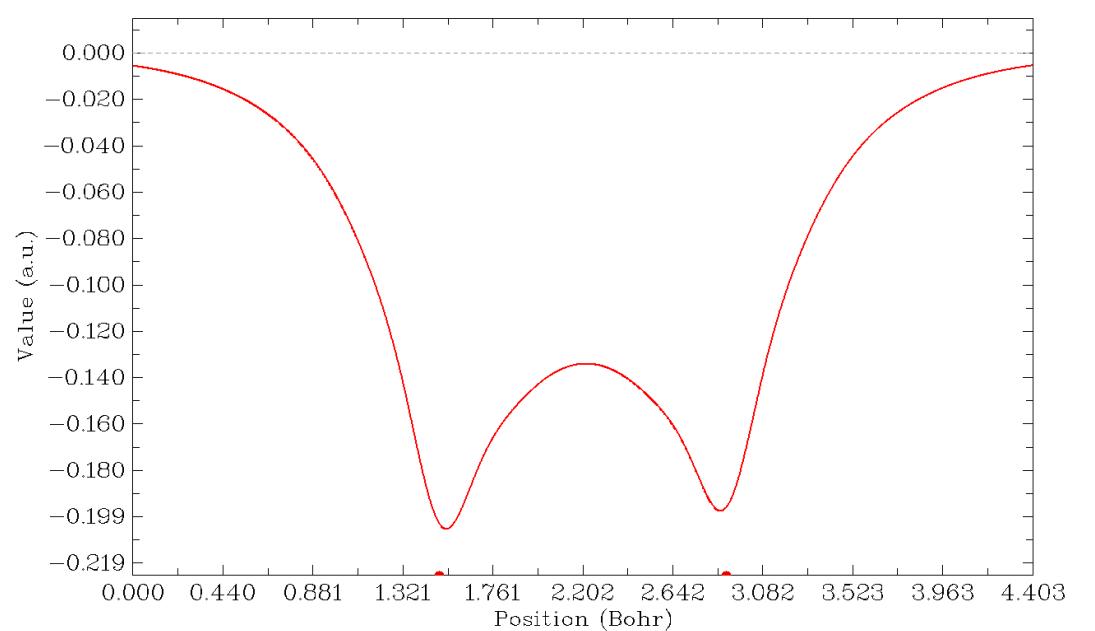
Correlation hole involves coordinates of two electrons, to visually study it we must determine position of a reference electron. In this example, we set reference point at (0.0,0.0,-0.3) Bohr. Therefore, we open settings.ini, change *refxyz* parameter to 0.0,0.0,-0.3.

The correlation hole we first analyzed is Fermi hole (also known as exchange hole), so we change *paircorrtype* in settings.ini to 1. Since this is closed-shell system, the results for α or β electron are exactly the same, while for open-shell system, you should use "pairfunctype" in settings.ini to select which type of spin electrons will be studied, you can also choose to study exchange-correlation density or correlation factor by adjusting this parameter.

Now boot up Multiwfn, input following commands

```
examples\H2_CCSD.wfn
3 // Draw curve map
17 // Correlation hole
1 // Defining the line by nuclear coordinate of two atoms
2,1 // Draw curve graph along H2 and H1
```

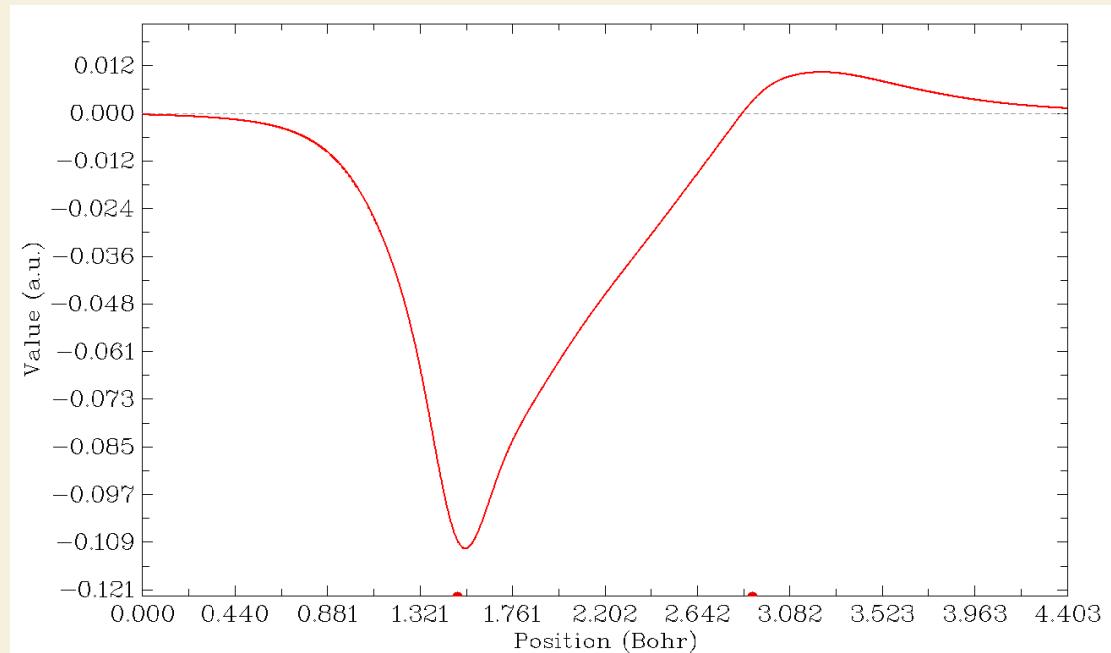
Then you will see



This graph suggests that if we place an α electron at (0.0,0.0,-0.3), then the probability of finding another α electron around the two nuclei will be significantly decreased by almost identical extent due to Pauli repulsion between like-spin electrons. In the H-H bonding region, the probability

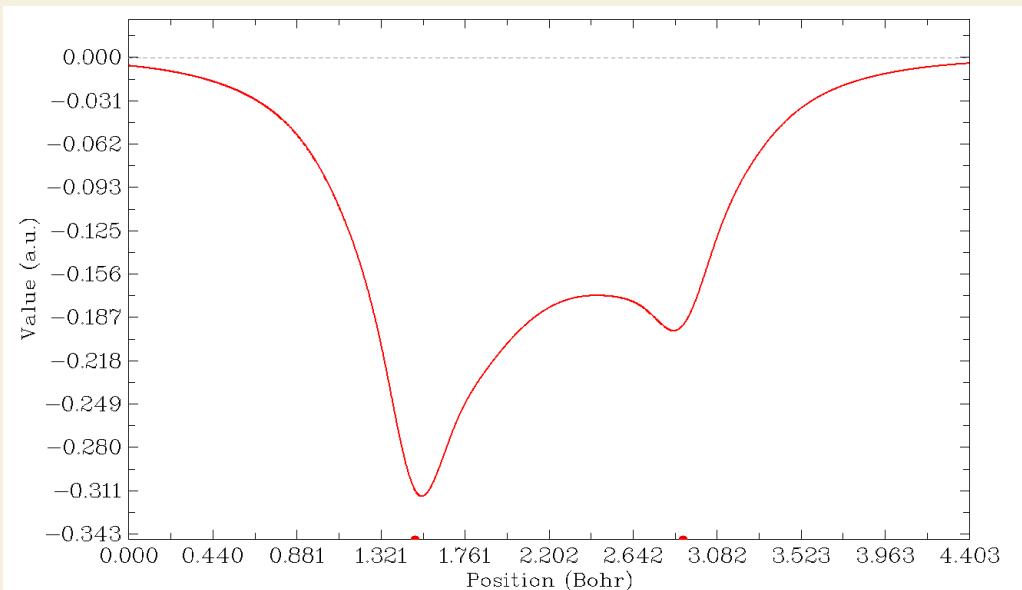
also obviously decreases. According to Bader's statement "An electron can go where its hole goes and, if the Fermi hole is localized, then so is the electron" (p251 in *Atoms in molecules - A quantum theory*), we can say that if an α electron presents at (0.0,0.0,-0.3), then it must be able to easily delocalize over the whole H₂ molecular space.

Now we study Coulomb hole. Close Multiwfn, change "paircorrtype" to 2, reboot Multiwfn and then replot the curve graph by using the same procedure, you will see



The red dot at left and right sides highlight position of H₂ and H₁, respectively. How can I infer the position of the reference point on the graph? From the prompt "Set extension distance for mode 1, current: 1.500000 Bohr" we know that the plotting range is extended by 1.5 Bohr in both sides relative to Z coordinates of H₂ and H₁, and we know that Z coordinate of reference point is larger than H₂ by about 0.4016 Bohr, so the reference point on above graph is 1.5+0.4016=1.9016 Bohr (If you want to highlight the position of reference point on the graph, you can select "4 Draw a vertical line at specific X" and input 1.9016, then redraw the graph via option -1). Above graph reveals that owing to Coulomb repulsion, the probability of finding another electron (α or β) around the hydrogen closest to the reference electron (namely around H₂) is largely decreased. As a compensation, the probability is increased at backside of H₁.

Finally, we study the collective effect of Fermi correlation and Coulomb correlation. Change "paircorrtype" to 3, and plot the line graph again, you will see



This graph is actually the sum of the first two graphs. From this graph it is clear that the probability decrease of finding an electron (α or β) around the hydrogen closest to reference electron is more severe than that around another hydrogen.

4.3.3 Study interatomic interaction via PAEM-MO method

This is a relatively advanced example, you can skip this section if you are a newbie of quantum chemistry.

In *J. Comput. Chem.*, **35**, 965 (2014), the authors proposed a method named PAEM-MO to reveal the nature of interaction between two atoms (covalently or noncovalently bounded). In this example I will briefly introduce this method and show how to realize PAEM-MO analysis in Multiwfn.

PAEM (potential acting on one electron in a molecule) refers to the total potential acting on an electron at point \mathbf{r} , and can be written as $V_{\text{PAEM}}(\mathbf{r}) = -V_{\text{ESP}}(\mathbf{r}) + V_{\text{XC}}(\mathbf{r})$; where V_{ESP} is molecular electrostatic potential and has been introduced in part 12 of Section 2.6. $-V_{\text{ESP}}$ can be regarded as the classical potential acting on an electron in the system, while the exchange-correlation (XC) potential V_{XC} represents the important correction to the classical potential due to quantum effect. V_{XC} has two components, namely correlation potential (V_C) and exchange potential (V_{XC}); in fact only the latter is important, that means even the potential obtained at Hartree-Fock level is in general a good approximation to exact V_{XC} .

In wavefunction theory, the exchange-correlation potential can be explicitly written as

$$V_{\text{XC}}(\mathbf{r}) = \frac{1}{\rho(\mathbf{r})} \int \frac{\Gamma_{\text{XC}}(\mathbf{r}, \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}', \text{ where } \Gamma \text{ is known as exchange-correlation density, see part 17}$$

of Section 2.6 for detail. In DFT theory, the XC potential directly comes from the variation of exchange-correlation functional with respect to electron density, i.e. $V_{\text{XC}}(\mathbf{r}) = \frac{\delta E_{\text{XC}}[\rho(\mathbf{r})]}{\delta \rho(\mathbf{r})}$. The

V_{XC} can be used in Multiwfn in terms of user-defined function. If parameter "iuserfunc" is set to **33**, then V_{XC} will be calculated based on Γ ; if it is set to **34**, V_{XC} will be evaluated in terms of DFT XC potential, in this case the computational cost is several times lower than **33**. For more information please check corresponding description in Section 2.7.

The essence of HF and KS-DFT theories is the one-electron eigenvalue equation

$$\hat{h}\varphi(\mathbf{r}) = \epsilon\varphi(\mathbf{r}) \quad \text{where } \hat{h} = -(1/2)\nabla^2 + V_{PAEM}(\mathbf{r})$$

Solving the equation results in a set of MOs $\{\varphi\}$, their eigenvalues correspond to the energy of the electrons running in the orbitals.

According to the viewpoint of PAEM-MO, if the energy of an occupied MO is higher than the barrier of V_{PAEM} between a pair of atoms, then the electrons in this MO will be able to freely delocalize over the two atoms, and thus have direct contribution to covalent bonding.

Below I present two very simple examples of using PAEM-MO method to judge the type of interatomic interaction. More examples and discussions can be found in *J. Comput. Chem.*, **35**, 965 (2014).

H-H interaction in hydrogen molecule

First set *iuserfunc* in settings.ini to 33, then user-defined function will be equivalent to the V_{XC} evaluated based on Γ . Boot up Multiwfn and input below commands

examples\H2\fch // Produced at HF/def2-TZVP level

3 // Plot real space function along a line

100 // User-defined function

0 // Adjust extension size at both sides

3 // 3 Bohr, which is larger than the default value

1 // Use two nuclei to define the line

1,2

Close the graph, then adjust some plotting parameters to make the graph better

11 // Change length unit of the graph to Angstrom

3 // Change range of Y axis

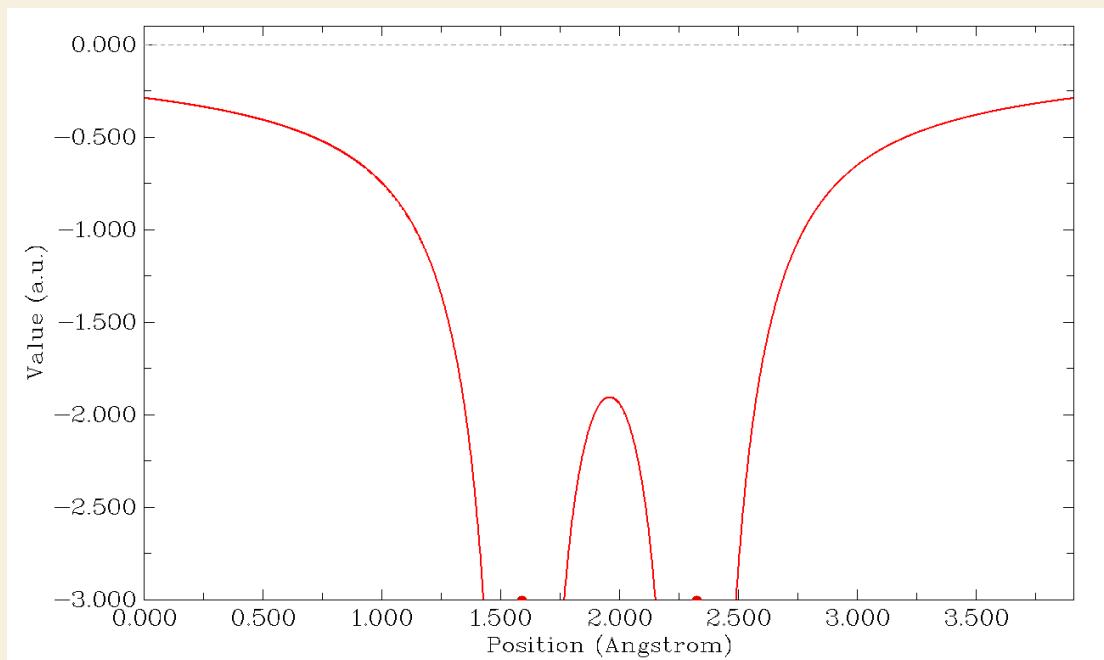
-3,0.1 // From -3.0 a.u. to 0.1 a.u.

10 // Set stepsize of X and Y axes

0.5,0.5

-1 // Replot

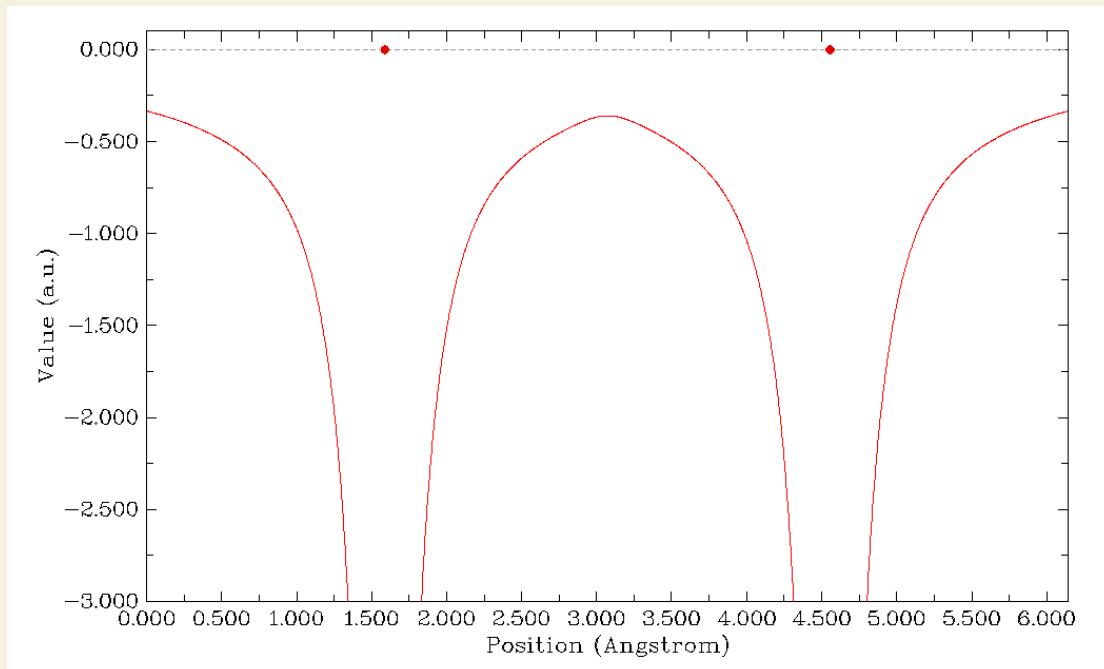
You will see below graph, which exhibits the PAEM curve along the H₂ axis



As you can see, there is a barrier of PAEM around the midpoint of H-H bond. In order to locate the exact value of the barrier, we close the graph and select 6. From the output we find the maximum of the curve is -1.903 a.u. (-51.78 eV). Present system only has one doubly occupied MOs, whose energy (-16.23eV) significantly exceeded the PAEM barrier, implying that the electron is not confined in any hydrogen atom but substantially contributes to the H-H binding, so the H-H bond must be covalent.

He-He interaction in He₂

Please plot PAEM curve along the He-He axis in He₂ with the same steps as above. The wavefunction file used here is *examples\He2.fch*, which was produced at HF/def2-TZVP level at separation of 2.97 Å (a reasonable distance). The resultant graph should look like below



Obviously, the PAEM barrier (-9.78 eV) is higher than that in the case of H₂. The energy of the two doubly occupied MOs are -25.03 and -24.91 eV respectively, both of them are lower than the barrier, therefore the electrons in each He atom are difficult (but not completely impossible due to tunnel effect) to overcome the barrier to freely delocalize to another He. The He-He interaction thus should be regarded as noncovalent interaction.

Note that in the original paper of PAEM-MO the PAEM is evaluated based on the expensive CISD wavefunction, while we merely use HF wavefunction. However, our results are in good agreement with that at CISD level, showing that correlation potential may be safely neglected in the study of PAEM.

The interested users are suggested to replot the PAEM with iuserfunc=34 to employ DFT XC potential in PAEM, you will find the results are very similar to those we obtained earlier.

Worthnotingly, although the PAEM-MO analysis method has clear physical meaning, its many limitations severely hinders it to be a universal method to distinguish covalent and noncovalent interactions like ELF or LOL: (1) PAEM-MO analysis does not always present reasonable conclusion in all cases. For example, PAEM-MO erroneously indicates that the H-bond in water dimer is covalent interaction. (2) PAEM-MO is difficult to be applied to polyatomic molecules, since there are often too many occupied orbitals with complex shape. (3) If the two atoms are placed too close to each other, then PAEM-MO almost always indicates that the interaction is covalent.

4.4 Output and plot various properties in a plane

Main function 4 of Multiwfn is used to plot various kind of plane maps for real space functions. This module is extremely flexible, it is obviously impossible to demonstrate all usage of this function with limited examples, however, if you have carefully read these examples and tried to reproduce graphs, you will gain basic knowledge about plotting plane map using Multiwfn. Section 3.5 is highly suggested to read, in which many important points about plotting plane map are introduced.

4.4.1 Color-filled map and contour line map of electron density of hydrogen cyanide

In this example we plot electron density for hydrogen cyanide as color-filled map and contour line map, which are the most employed types of plane map.

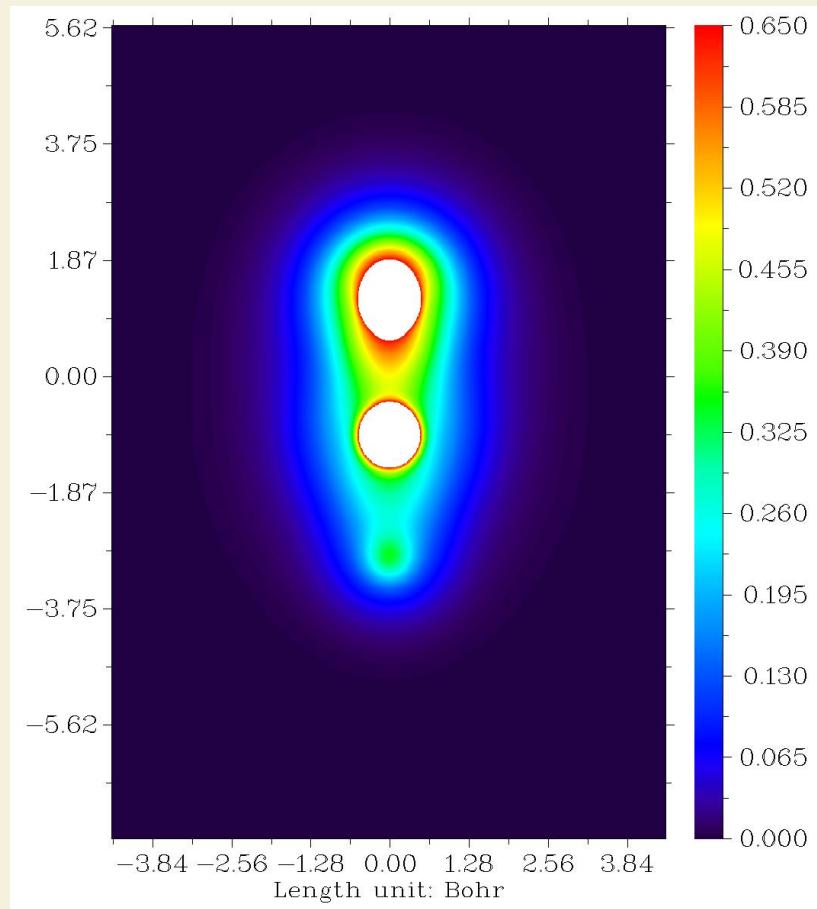
Boot up Multiwfn and input following commands

```
examples\HCN.wfn  
4 // Plot graph in a plane  
1 // Electron density  
1 // Color-filled map
```

Press [ENTER] button // Use the recommended grid setting, namely 200,200. If you increase the number of grid points, graph will become finer and smoother, but you have to wait longer time for calculating data and plotting the graph

```
2 // XZ plane (Z-axis is the molecular axis of present system)  
0 // The XY plane with Y=0 Bohr
```

After a few seconds the graph pops up

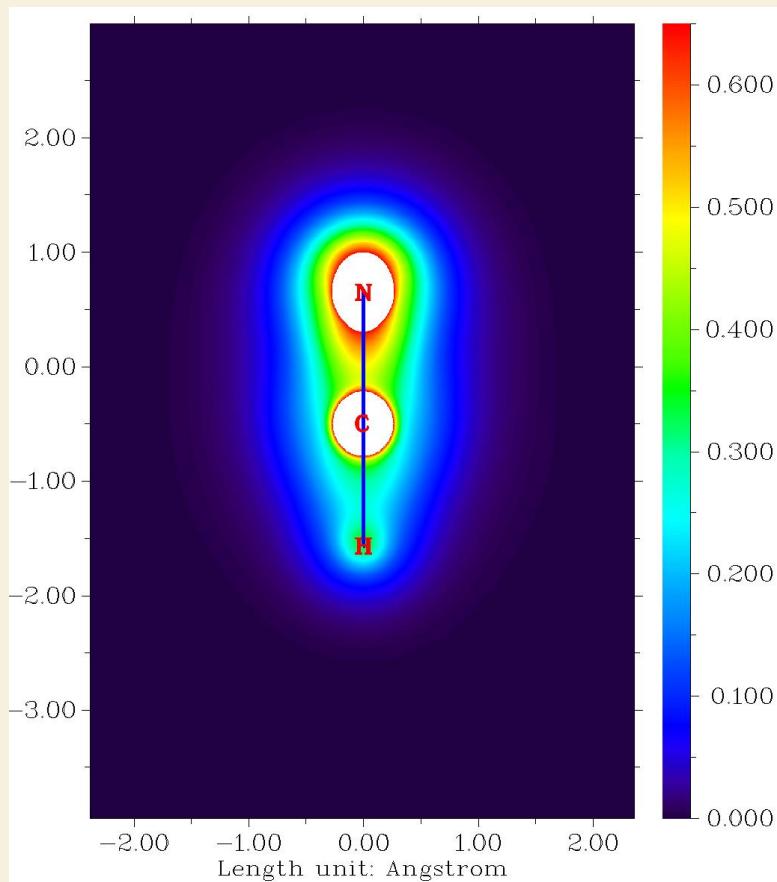


The central regions of carbon and nitrogen are white, suggesting that electron density exceeds the upper limit of color scale (0.65). Close the graph, then a post-process menu appears, there are many options and their meanings are very easy to understand. For example, you can choose corresponding options to adjust plotting parameters and then use -1 to replot again, or export X-Y data set to a plain text file for replotting the graph by external data visualization software (sigmaplot, Origin, matlab, etc.), or save image file in current directory (the graphical format is controlled by *graphformat* in *settings.ini*).

Now we slightly improve the graph. Input below commands:

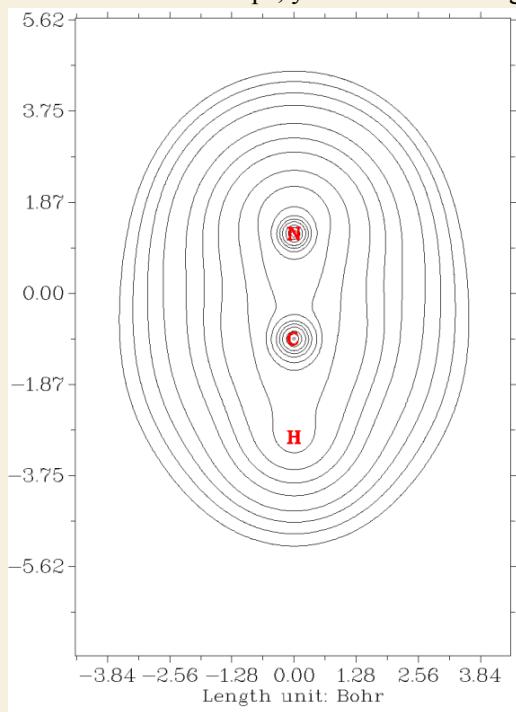
```
-8 // Change length unit of the graph to Angstrom
-2 // Set stepsize in X, Y, Z axes
1,1,0.1
4 // Enable showing atom labels
1 // Red labels
8 // Enable showing bonds
3 // Blue color for bonds
-1 // Redraw the graph
```

Now you can see below map on screen



In Section 4.6.2 we will plot valence electron density for HCN, you will find valence electron density conveys much more information than total electron density.

Next, we plot the electron density as contour line map. Repeat above example, but select "2 Contour line map" instead of "1 Color-filled map", you will see below graph



There are numerous options in post-process menu. 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, various parameters such as color, thickness, contour values can be easily customized, please try to play with them, and if you are confused, please consult Section 3.5.4 for more details.

It is worth to note that the .pdf format is more suitable than the default .png format for contour line map (or other maps mainly consisted of lines), because .pdf is a vector format, the graph can be scaled loselessly, and the lines look smoother. In order to change to .pdf format, you should change "graphformat" in settings.ini to *pdf*.

In addition, the color-filled map and contour lines can be shown together, namely plotting color-filled map as usual first, and then choose "2 Enable showing contour lines" in post-process menu.

4.4.2 Shaded surface map with projection effect of electron localization function (ELF) of monofluoroethane

Boot up Multiwfn and input following commands

*examples*C2H5F.wfn

0 // Let us view the molecular structure first to find the plane we are interested in. Suppose that the C-C-F plane is what we want, find their atomic indices from the graph (1, 5, 8), and then click RETURN button

4 // Plot graph in a plane

9 // Electron Localization Function (ELF)

5 // Shaded surface map with projection effect

[Press ENTER button to use recommended grid setting, namely 100,100]

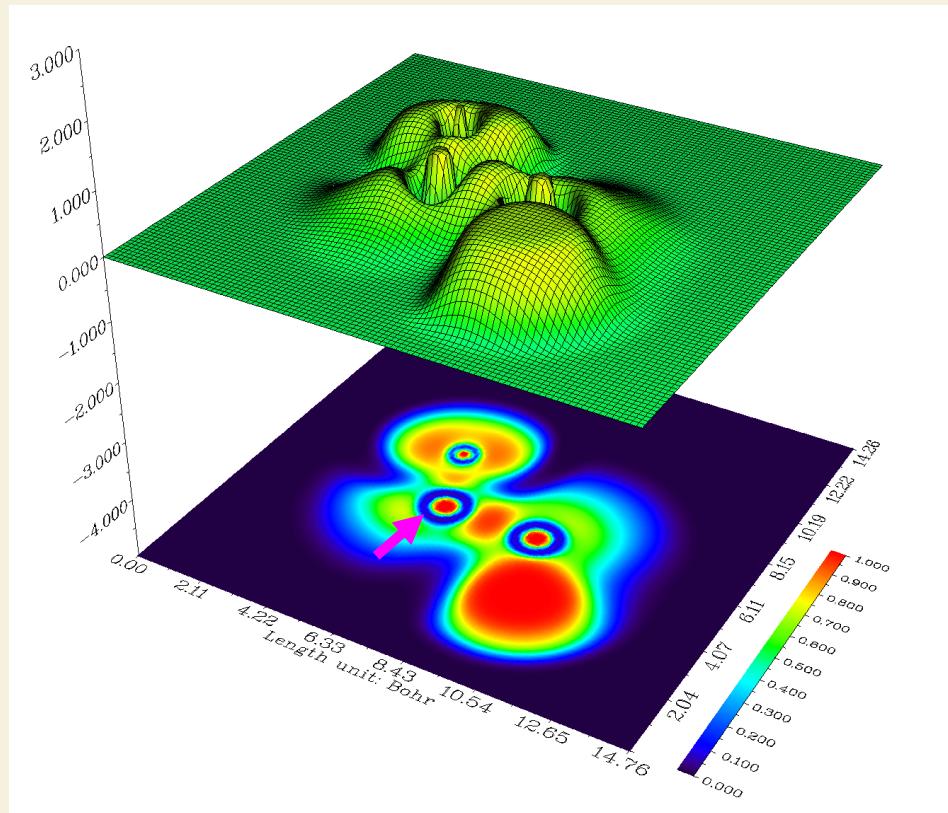
0 // Manually set extension distance. If you do not do this, you will found the resulting graph is somewhat truncated at boundary because the default extension distance is too small for present case

6 // Set extension distance to 6 Bohr, which is slightly larger than the default one

4 // Define the plane by three atoms

1,5,8 // Indices of the three atoms defining plotting plane

Below graph immediately pops up:



We can see that C-C and C-F covalent bond regions have high LOL value, the electron depletion regions between valence shell and inner shell are shown by the blue circles around nuclei. A lone pair of fluorine atom is pointed out by purple arrow.

The graph you have seen in the GUI can be saved to graphic file by option 0 at post-process interface. If you found the exported picture is truncated at edge, select option -1 to re-enter the GUI window, zoom out the graph and then export the picture again.

If you want to translate or rotate the content in the graph, in the interface of defining plotting plane, before choosing option 4 (or 5), you should first choose option -1 to set translation value and rotation angle, see Section 3.5.2 for detail.

4.4.3 Plotting plane map without contributions from some atoms

The main purpose of this section is illustrating how to plot plane map of a real space function without contribution of some atoms. This aim can be realized in two different ways in Multiwfn, as respectively exemplified in below two examples.

Example 1: Contour map of electron density Laplacian of uracil without contributions of two atoms

In main function 6, one can use subfunctions -3 and -4 to delete Gauss type functions (GTFs) centered at some atoms to remove their contributions to various kinds of analyses that based on real space function. This feature will be utilized in present example. Since this treatment reduces total number of GTFs, the computational cost in the subsequent analyses will be lowered.

Boot up Multiwfn and input following content

```
examples\uracil.wfn
```

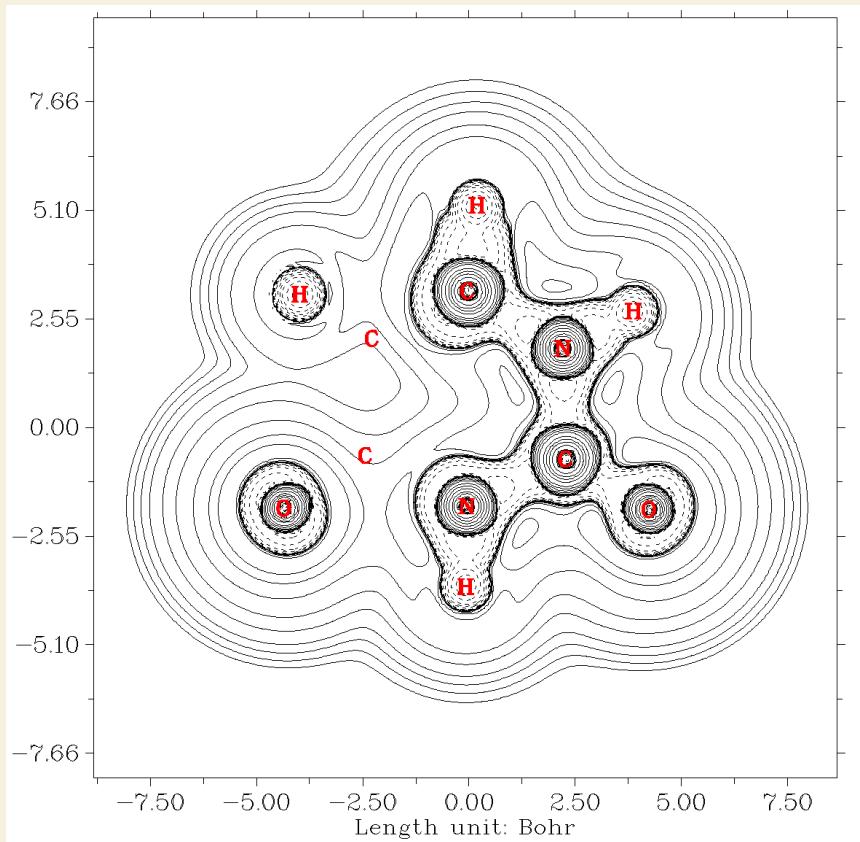
```
6 // Modify wavefunction
```

```
-4 // Discard contribution of some atoms
```

3,4 // All GTFs centered on atoms 3 and 4 will be discarded. In other words, contribution of atoms 3 and 4 will be removed from current wavefunction

```
-I // Return to main menu
4 // Plot graph in a plane
3 // Laplacian function
2 // Contour line map
[Press ENTER button to use default grid setting]
1 // XY plane will be plotted
0 // The Z-position of the XY plane is zero, that is molecular plane
```

Below is the resulting graph, solid and dashed line correspond to positive and negative regions, respectively.



From the graph you can see that the contribution from the two carbons have been discarded as we expected.

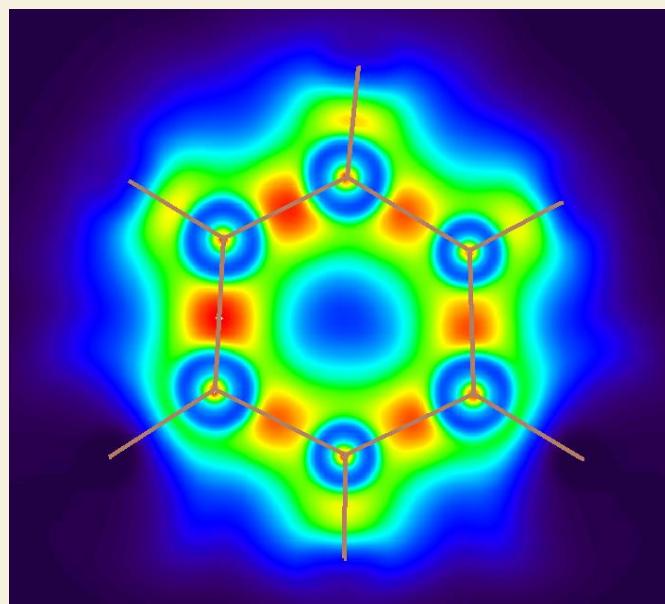
By the way, if you are studying a large system but only a local region is of interest, you can remove GTFs at atoms far from this region to save computational time of real space function analyses (e.g. topology analysis, basin analysis, calculating grid data...).

Example 2: Plane map of LOL contributed by atoms in the uracil ring

After plotting plane map as usual, one can request the program to plot the map only contributed by certain fragment. Specifically, Hirshfeld weighting function of the user defined fragment will be generated and multiplied to the plane data. This treatment only affects the currently plotted map while do not influence any further analysis, since this treatment does not modify wavefunction.

Here we use this feature to plot localized orbital locator (LOL) map of uracil that only contributed by the six atoms in the ring. Boot up Multiwfn and input below commands:

```
examples\uracil.wfn
4 // Plane map
10 // LOL
1 // Color-filled map
[Press ENTER button to use default grid setting]
1 // XY plane will be plotted
0 // Z=0
Close the graph, and then input
-9 // Only plot the data around certain atoms
1-6 // The index of the six atoms in the uracil ring
8 // Enable showing bonds
14 // Brown
-1 // Replot
Then you will see
```



Clearly, the value of LOL at the grids far from the ring atoms have been significantly screened. Then if you want to restore the original map, you can choose "-9 Recovery original plane data" in the post-process menu and then replot.

4.4.4 Contour map of electrostatic potential of chlorine trifluoride

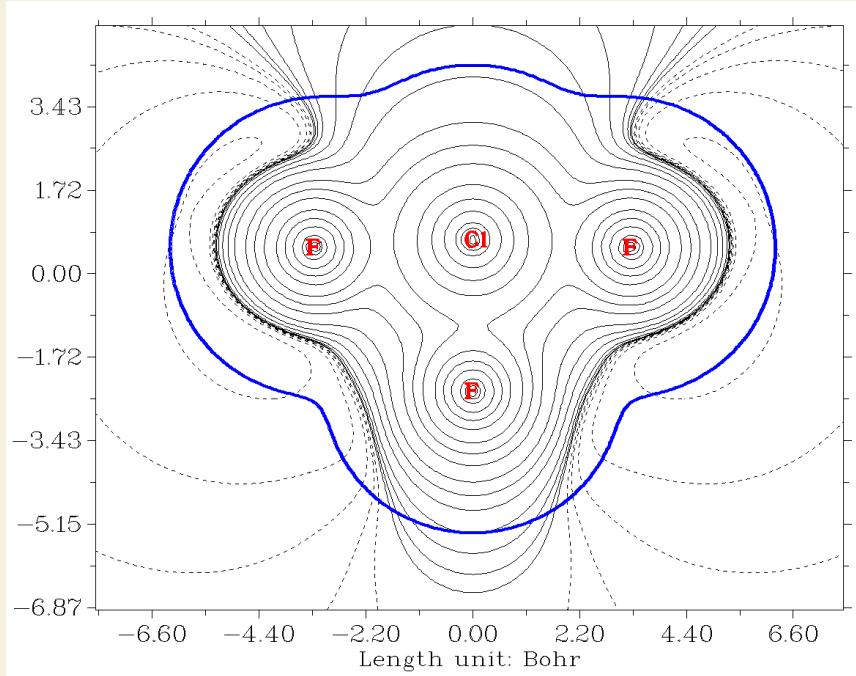
In this example we plot electrostatic potential (ESP) for chlorine trifluoride as contour map. Boot up Multiwfn and input following commands

```
examples\ClF3.wfn // Generated at B3LYP/6-31G* level
4 // Plot graph in a plane
12 // Total electrostatic potential
2 // Draw contour line map
120,120 // Number of grids in each direction
3 // YZ plane
0 // Set X coordinate of the YZ plane to 0
```

Because evaluation of ESP is evidently more time-consuming than other real space functions, you need to wait for a while.

IMPORTANT NOTE: If you make use of the cubegen utility in Gaussian to evaluate ESP value at the plotting points, the overall computational cost could be significantly reduced, since cubegen calculates ESP much faster than internal code of Multiwfn, especially for large systems and high-quality basis sets. See Section 5.7 on how to make use of cubegen in Multiwfn analyses.

After the calculation is finished, ESP map pops up. This map is inconvenient to be visually analyzed, since what we are interested in is often the ESP value on molecular vdW surface, hence 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 in post-process menu, and then choose option -1 to replot the graph, you will see such a picture. The solid and dashed lines represented the region having positive and negative value of ESP, respectively.



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 chlorine atom is overall positively charged, because the vdW surface close to the chlorine atom largely intersects solid contour lines. For the same reason, we can see that the equatorial fluorine atom possesses less electrons than the two axial fluorine atoms, this point can be further verified when we calculate atomic charges for this molecule in section 4.7.1.

Plane map of the ESP derived from atomic charges can be directly plotted by Multiwfn too. First, you need to prepare a plain text file with .chg extension, the first column corresponds to element name, the 2th, 3th and 4th columns correspond to X,Y and Z cooredinates in Angstrom respectively, the last column is atomic charge. For example:

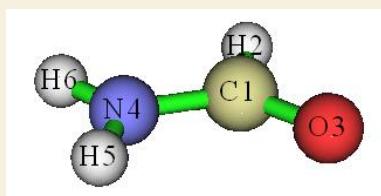
C1	0.000000	0.000000	0.359408	0.529971
F	0.000000	1.726507	0.294501	-0.228394
F	0.000000	0.000000	-1.267884	-0.073185
F	0.000000	-1.726507	0.294501	-0.228394

Boot up Multiwfn as usual, and then use the .chg file as input. The plotting procedure is

completely identical to the one given above, except that when Multiwfn prompts you to select real space function, you should select 8 (ESP from atomic charges) instead of 12.

4.4.5 Contour map of two orbital wavefunctions

Multiwfn is capable of plotting contour map for two orbitals simultaneously. In this section, we will draw contour map for simultaneously portraying NBO 12 and NBO 56 of NH₂COH (recall Section 4.0.2). The plane we selected is the one perpendicular to molecular plane and passed through both carbon and nitrogen atoms. As you will see, we need to use a special manner to define such a plotting plane. The molecule geometry and atomic index are shown as follows.



Boot up Multiwfn and input:

examples\NH2COH.31

37 // Load NH2COH.37

4 // Plot plane graph

4 // Orbital wavefunction

12,56 // The two orbital indices. If you only input one index, then only one orbital will be plotted

Press ENTER button to use default grid setting

7 // This mode is used to define a plotting plane parallel to a bond and meantime normal to a plane defined by three atoms

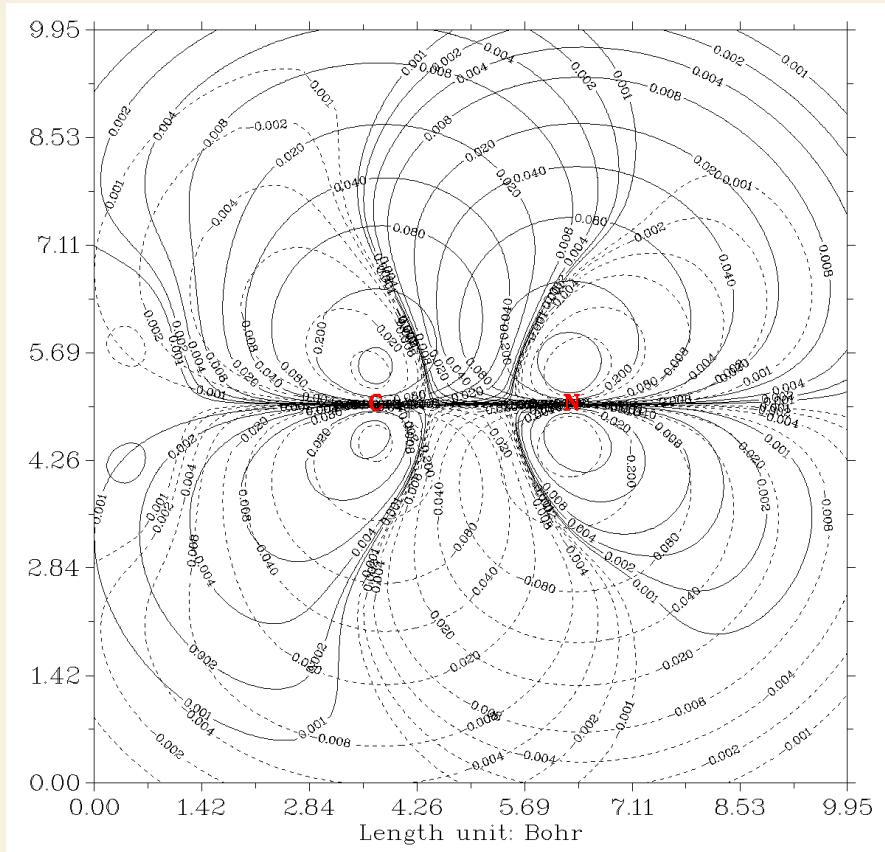
1,4 // The plotting plane is parallel to C1-N4

3,1,4 // The plotting plane is vertical to the plane defined by O3-C1-N4

10 // The length of X-axis of the resulting map is 10 Bohr

10 // The length of Y-axis of the resulting map is 10 Bohr

Immediately a graph pops up. We close it by clicking right mouse button, choose option 2 and input 25 to enable showing isovalue on contour lines, then choose -1 to redraw the graph, we will see:



This contour map is not quite ideal, there are too many contour lines intertwined together and thus confused our vision. The culprits are the contour lines with too small isovalue (magnitude smaller than 0.01). Since these contour lines are unimportant, we can delete them to make the graph clearer. Therefore, we close the graph and input

3 // Change contour line setting

4 // Delete some contour lines

1-4 // Delete contour lines 1~4, they respectively correspond to 0.001, 0.002, 0.004, 0.008

4 // Delete some contour lines

28-31 // Delete the four contour lines corresponding to -0.001, -0.002, -0.004, -0.008. For convenience, you can choose option 6 to export current contour line setting to an external file, when you use Multiwfn next time you can load present setting directly by choose option 7 in current interface

15 // Set the drawing style suitable for publication, namely positive and negative parts are portrayed as red solid lines and blue dashed lines, respectively

1 // Save setting and return to the upper menu

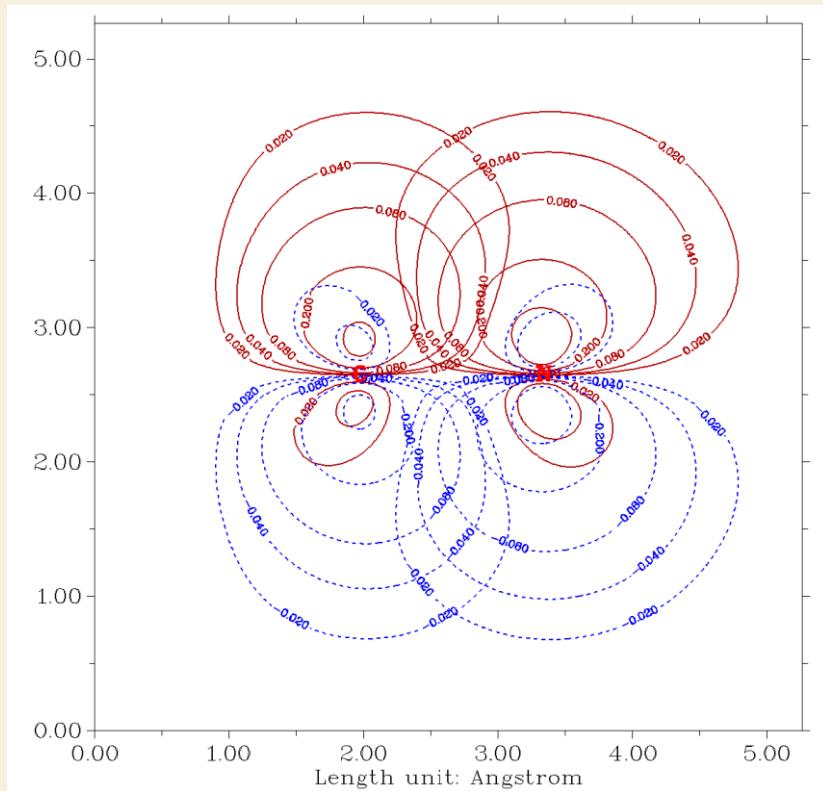
-8 // Change length unit of the graph to Angstrom

-2 // Set stepsize in X and Y axes

1,1 // Stepsize in both X and Y axes are 1.0 Å

-1 // Replot the contour map

Now the graph become clear and informative, the same phase overlap areas are very obvious. Notice that only the labels of the atoms whose vertical distance from the plane are smaller than *disshowlabel* in settings.ini are displayed in the graph, this parameter can also be adjusted by option "17 Set distance threshold for showing atom labels" in post-process menu.



4.4.6 Gradient+contour map with topology paths of electron density of hydrogen peroxide

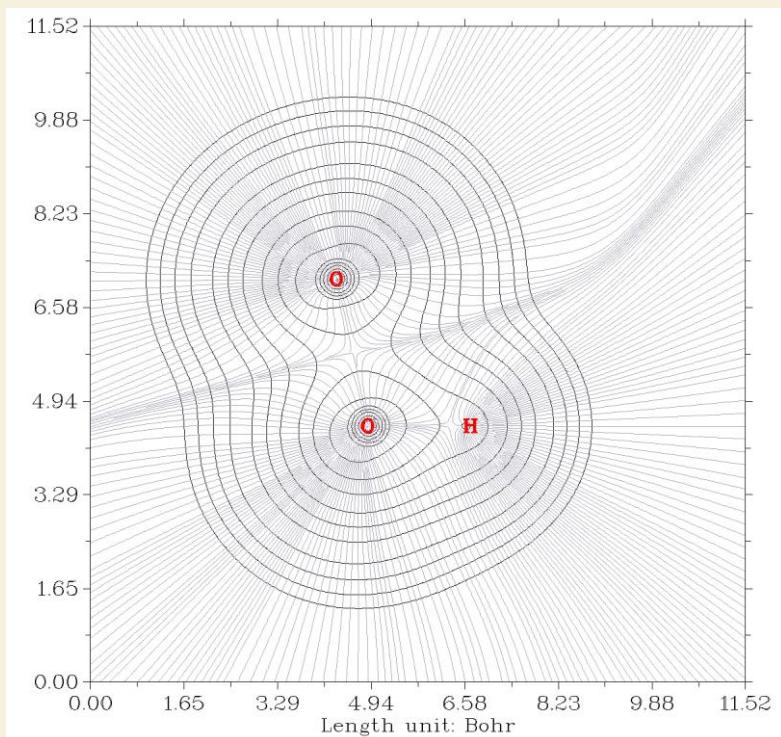
As introduced in Section 3.5.5, critical points, bond paths and interbasin surfaces can also be plotted on plane map, here I give a very simple example.

Boot up Multiwfn and input following commands

examples\H2O2.fch // Of course, you can also use other type of file as input, as long as the file contains GTF information

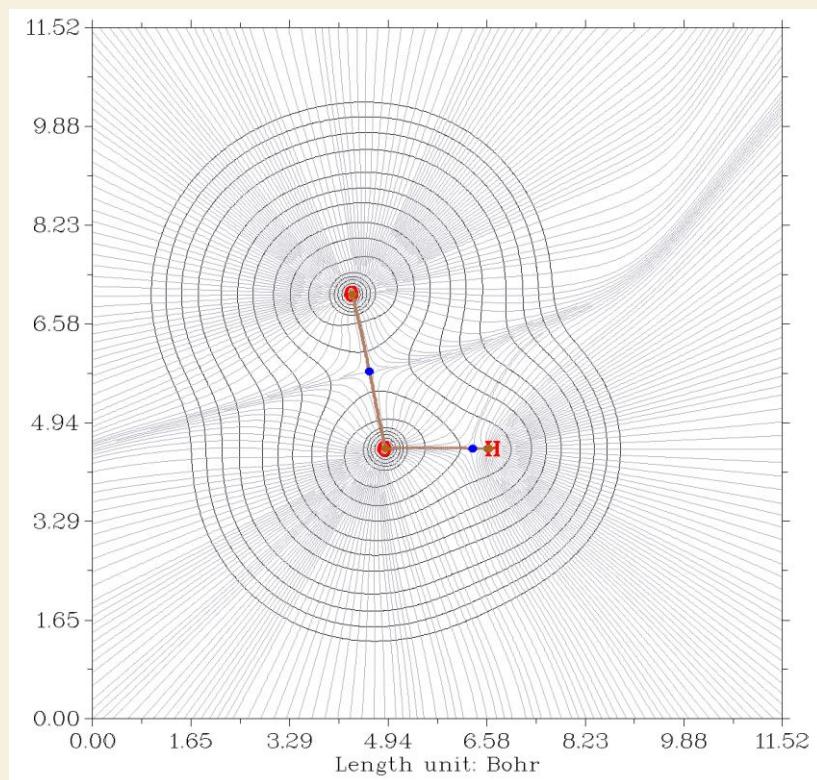
```
4 // Plotting plane map
1 // Electron density
6 // Gradient line with/without contour line map
[Press ENTER button to use default grid setting]
4
2,1,3 // Define the plane by nuclear coordinates of atom 1, 2 and 3
```

Generating data and plotting gradient map take more computational time relative to other graph type, however since present system is small and the basis set is only 6-31G*, resulting graph shows up immediately:



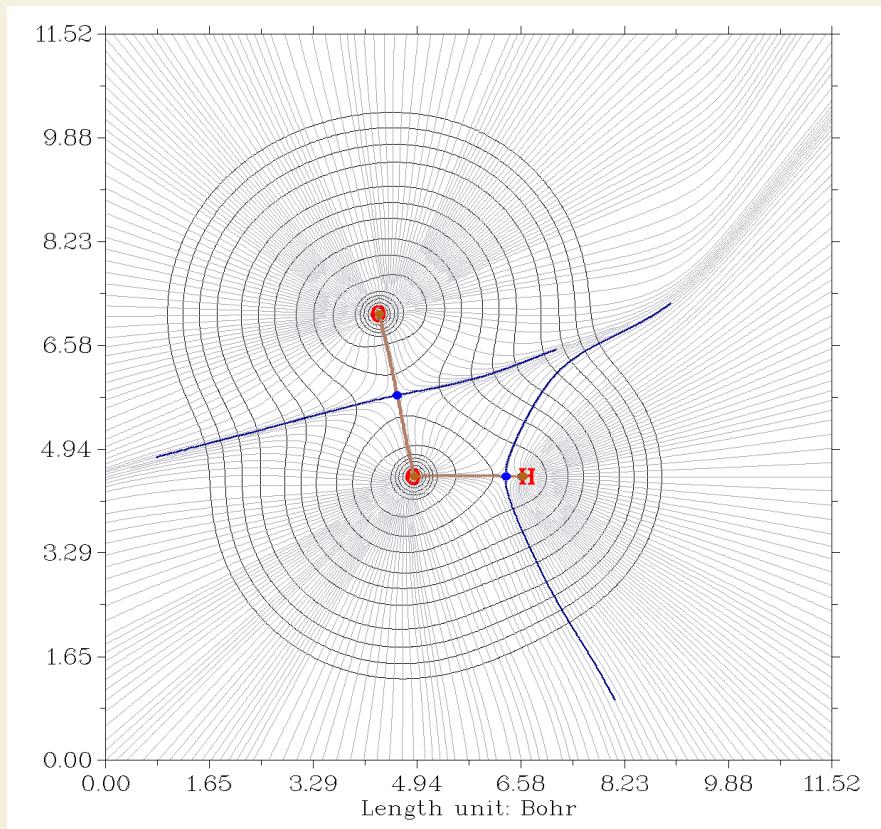
This type of graph is very useful in Bader's AIM analysis. You can also plot gradient+contour map for any other real space functions supported by Multiwfn. In the post-process menu, you can use options 11, 12, 13 and 14 to adjust plotting effect of the gradient lines, they can control the smoothness, density, color and density of the gradient lines.

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 resulting graph looks like below. Brown, blue, and orange circles denote (3,-3), (3,-1) and (3,+1) critical points, respectively, bold dark brown lines depict bond paths.



In the post-process menu, you can enter “4 Set details of plotting critical points and paths” to adjust settings for showing the critical points and 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 replot the graph, the interbasin paths will be drawn on the graph by bold deep blue lines:



Note that before generating the interbasin paths, related parameter (stepsize and the number of iteration) can be set by "7 Set stepsize and maximal iteration for interbasin path generation" option in post-process menu. Larger number of iteration may result in longer interbasin paths.

4.4.7 Deformation map of electron density of acetyl chloride

Deformation map of electron density clearly shows variation of electron density distribution during formation of a molecule, it is defined as subtracting actual molecular electron density by electron density of all of its constituent atoms in their free-states. Illustrative example of deformation density analysis can be found from my paper [Acta Phys. -Chim. Sin., 34, 503 \(2018\)](#).

It is a labor work to draw such a graph via custom operation feature since there are so many atoms in practical chemical systems. Fortunately, Multiwfn provides a special option to realize this in a highly automatic way. Boot up Multiwfn and input following commands

examples\CH3COCl.wfn

4

-2 // Tell Multiwfn you want to draw deformation map, then Multiwfn prepares free-state atom wavefunctions

B3LYP/6-31G* // The level used to generate atomic wavefunction files by Gaussian, it is the same as the level used for generating CH3COCl.wfn

D:\study\g09w\g09.exe // The path of executable file of Gaussian (you can also use other Gaussian version). 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 remaining commands.

I // Electron density function

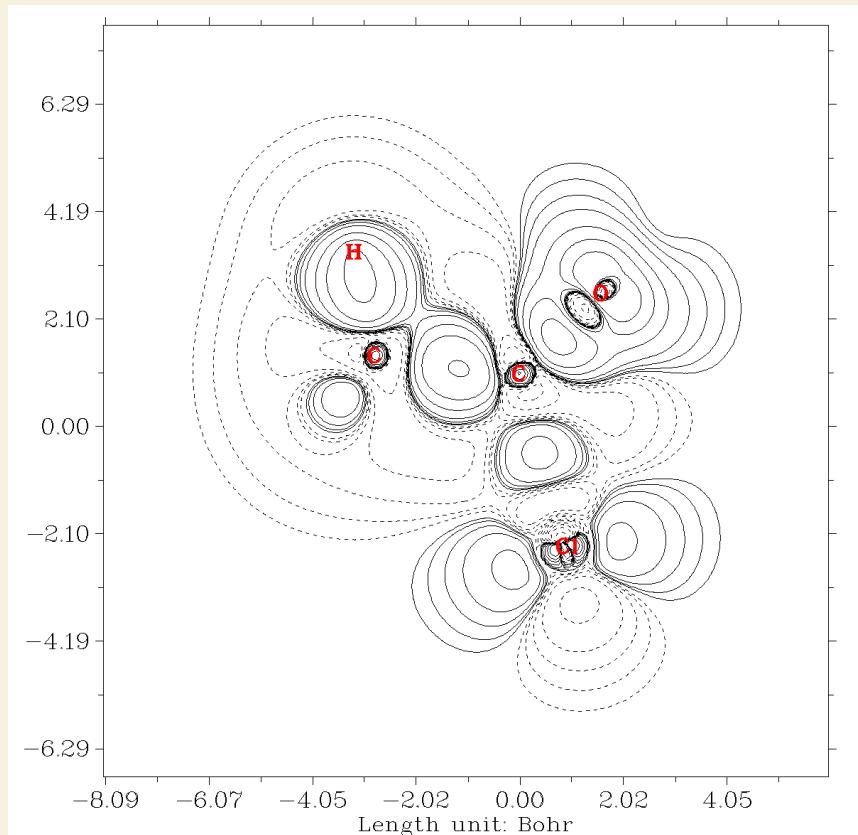
2 // Contour map

Press ENTER button to use default grid setting

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 that the density distribution around chlorine atom is largely desphericalized, this observation is in line with hybrid orbital theory, chlorine atom forms somewhat sp^3 hybrid state.

You can also plot deformation maps for other functions by choosing corresponding real space function, though not all of them are meaningful.

If you want to avoid recalculating atomic wavefunction files next time, you can copy the .wfn files without number suffix (such as “C.wfn”) from “wfntmp” folder to “atomwfn” folder in current directory, if Multiwfn finds that all needed atom wavefunctions have already existed in “atomwfn” folder, then Multiwfn will not invoke Gaussian to calculate them again.

Hint: You can also use “genatmwfnpdb” in “examples” directory to generate all atom wavefunctions under specific basis set in a single run, please consult Section 3.7.3.

The “atomwfn” folder in “examples” directory contains atom wavefunctions (by 6-31G*) for all first-four row elements, you can directly copy this folder to current directory, after that you will not need Gaussian again during plotting deformation map.

If your system involves some elements heavier than Kr, you have to manually calculate the corresponding atomic .wfn files and put them into "atomwfn" folder". More detailed information about preparing atomic wavefunction files can be found in Section 3.7.3.

4.4.8 Draw difference map of electron density and ELF for water tetramer with respect to its constituent monomers

In this example I will illustrate how to plot difference map between a system and its constituent fragments for a given real space function. Electron density and ELF will be employed as the function to be studied.

examples\water_tetramer\wfn\complex.wfn is wavefunction file of optimized water tetramer, while the *water1/2/3/4.wfn* in this folder is wavefunction file of each water monomer. The corresponding Gaussian input files are also provided in the folder. Notice that the monomer coordinates were directly extracted from the complex coordinate, and *nosymm* keyword was used for all files to avoid Gaussian automatically reorientating the molecular geometry during the calculations. (Bear in mind, density difference map is meaningful only when coordinates of all fragments are completely consistent with that of the whole complex)

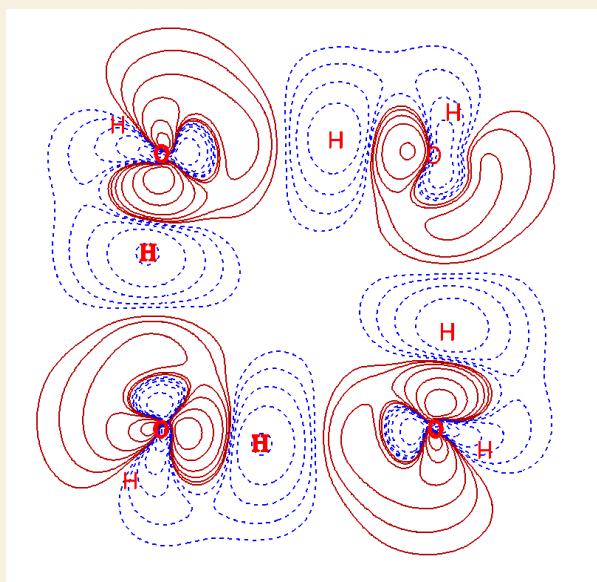
First we plot plane map of electron density difference for the complex with respect to all the four monomers. Boot up Multiwfn and input

```
examples\water_tetramer\wfn\complex.wfn
4 // Plot plane map
0 // Custom operation
4 // Four files will be operated on the firstly loaded system
-,examples\water_tetramer\wfn\water1.wfn
-,examples\water_tetramer\wfn\water2.wfn
-,examples\water_tetramer\wfn\water3.wfn
-,examples\water_tetramer\wfn\water4.wfn
1 // Electron density
2 // Contour map
[Press ENTER to use default grid setting]
4 // Define plane by three atoms
7,10,1
```

The graph pops up immediately. We can further improve the plotting effect. Close the graph and input

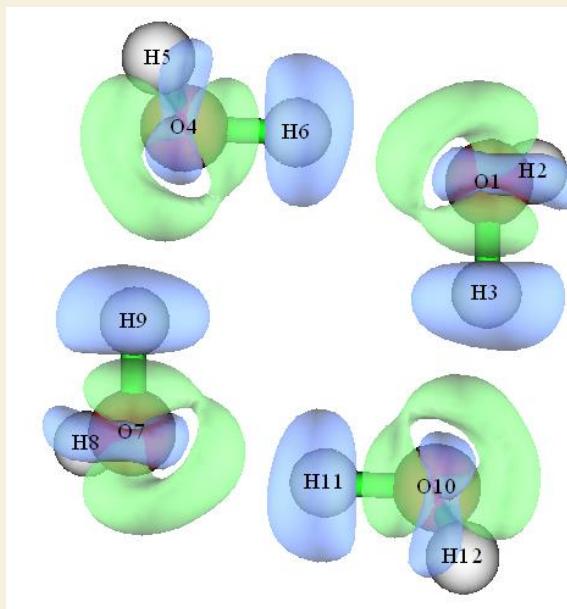
```
3 // Change contour line settings
15 // Set line style and width suitable for publication
1 // Save and return
17 // Set distance threshold for showing atom labels
0.2 // 0.2 Bohr
y // If the distance between an atom and the plotting plane is larger than the specified 0.2
Bohr, then the label will be drawn as thin style
0 // Save the plot as graphic file in current folder
```

The graphic file should look like below



In the graph, red solid lines and blue dashed lines correspond to the regions having increased electron density and decreased electron density during formation of the tetramer, respectively.

In Multiwfn you can also easily plot the density difference in the form of isosurface map by main function 5, the resulting graph is shown below. If you do not know how to do, please consult Section 4.5.5.



Next, we plot color-filled difference map of ELF. Input below commands

```
-5 // Return to main menu
4 // Plot plane map
0 // Custom operation
4 // Four files will be operated on the firstly loaded system
-examples\water_tetramer\wfn\water1.wfn
-examples\water_tetramer\wfn\water2.wfn
-examples\water_tetramer\wfn\water3.wfn
-examples\water_tetramer\wfn\water4.wfn
```

```

9 // ELF
1 // Color-filled map
[Press ENTER button]
4 // Define plane by three atoms
7,10,1

```

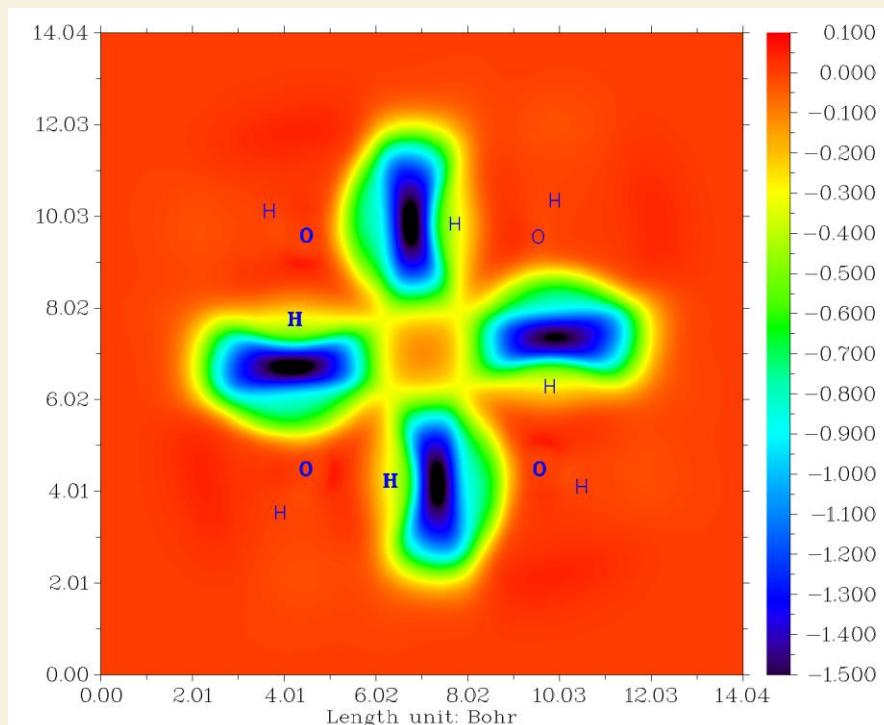
The graph shown on screen is ugly currently, because the default color scale is inappropriate for present case. Close the graph and input

```

1 // Set lower&upper limit of color scale
-1.5,0.1
4 // Disable showing atomic labels
4 // Enable showing atomic labels again, now you can select label color
3 // Blue labels
-1 // Show the graph again

```

You will see



The blue and especially dark blue regions exhibit decrease of ELF in corresponding regions. This plot shows that during formation of the complex, the electron localization is reduced in the intermolecular interaction regions, it may be attributed to the consequence of the Pauli repulsion effect.

4.4.9 Draw LOL- π map for porphyrin to reveal favorable electron delocalization path

The well-known ELF- π is the ELF solely contributed by π electrons. Similarly, LOL- π can be defined as a variant of localized orbital locator (LOL). The features of LOL- π are highly analogous to ELF- π , but sometimes LOL- π performs better. In this Section I will illustrate how to plot color-

filled LOL- π plane map above 1.2 Bohr of porphyrin, you will find this map is quite useful for understanding preferential electron delocalization path, which is closely related to molecular aromaticity.

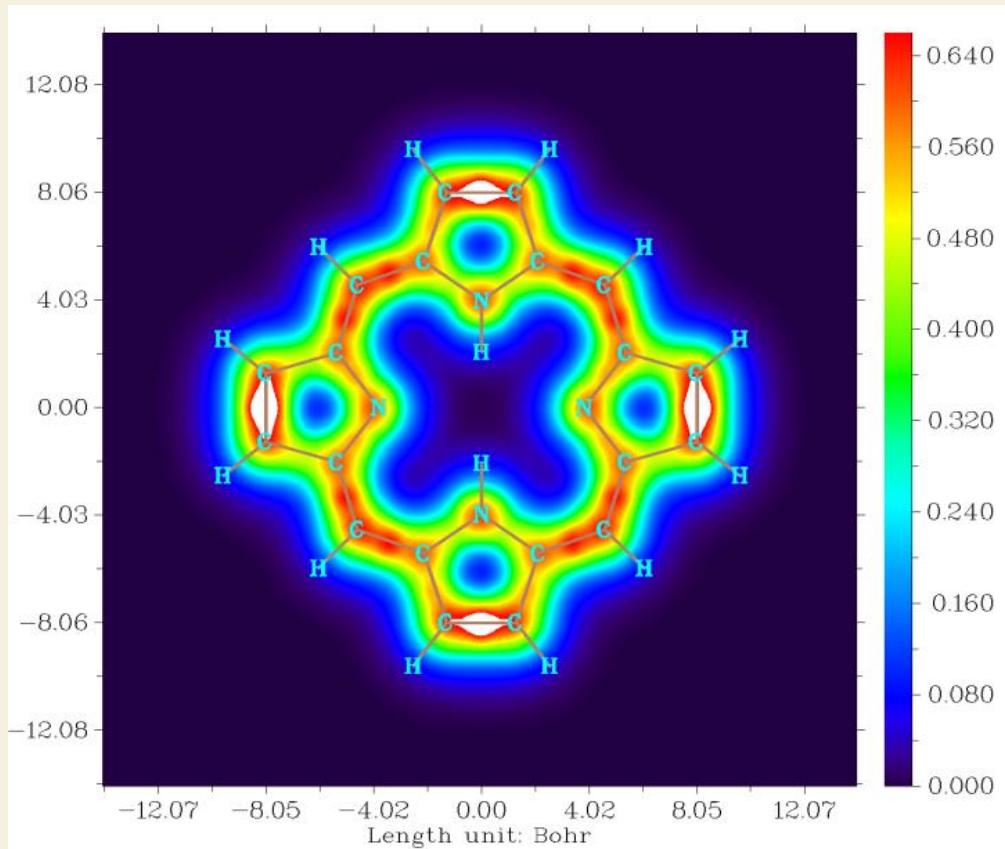
The only difference of plotting and analyzing LOL and LOL- π is that for the latter case, you should first set occupation number of all orbitals to zero except for π orbitals. As illustrated below, this can be automatically done via Multiwfn.

The .fch file of porphyrin calculated at B3LYP/6-31G* level can be downloaded from <http://sobereva.com/multiwfn/extrafiles/porphyrin.rar>. Boot up Multiwfn and load the .fch file, then input below commands:

```
100 // Other functions
22 // Detect  $\pi$  orbitals
0 // Automatically identify molecular plane. Note that this mode is applicable when all atoms
are in the same Cartesian plane, such as present case (all atoms are in YZ plane). For other cases,
automatic detection of  $\pi$  orbitals must be realized based on localized molecular orbitals, see Section
4.100.22
2 // Set occupation number of all other orbitals to zero
0 // Return to main menu
4 // Plot plane map
10 // LOL
1 // Color-filled map
Press ENTER button to use default grids
3 // Plot YZ plane
1.2 // X=1.2 Bohr
Close the graph, then input
1 // Set lower&upper limit of color scale
0,0.66
4 // Enable showing atom labels
7 // Cyan color
17 // Set distance threshold for showing atom labels
2 // Since the distance between the plotting plane and molecular is 1.2 Bohr, to make all
atomic labels shown on the graph, this threshold must be set to a value larger than 1.2 Bohr. Here
we set it to 2.0 Bohr
```

```
y
8 // Enable showing bonds
14 // Brown color
-1 // Replot
```

Now you will see below graph



The high LOL- π regions (red or orange color regions) clearly reveal the favorable delocalization path. If you plot current map induced by an external magnetic field vertical to the molecular plane (using for example AICD or GIMIC methods, see my slideshow for details: <http://sobereva.com/148>), you will find the unidirection contiguous induced current is mainly formed on the favorable delocalization path highlighted by the LOL- π function.

4.5 Generate grid data and view isosurface map

This section contains examples of main function 5 of Multiwfn, all of them need to calculate grid data. Once the grid data is generated, it can be visualized as isosurface map, or be exported to .cub file so that it can be rendered by third-part tool such as VMD or be further utilized by other analysis codes.

4.5.1 Electron localization function of chlorine trifluoride

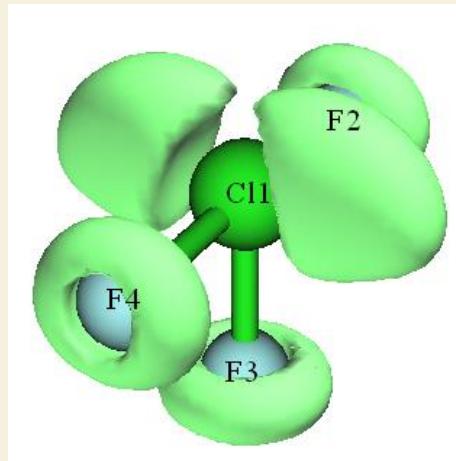
Boot up Multiwfn and input following commands

```
examples\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 information.
```

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 setting the *nthreads* parameter in settings.ini to actual number of physical 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 information, in the new menu you can draw the isosurface by option -1, then a GUI window pops up, input isovalue of 0.85 in the text box and press ENTER button on your keyboard, the isosurface would looks like this



The ELF isosurfaces clearly point 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 picture" in GUI), or use option 2 to export the grid data as Gaussian cube file in current directory.

If you want to use different colors to shade different isosurfaces, you can follow the procedure described in a mini tutorial named "Drawing ELF isosurfaces with different colors for different domains" (http://sobereva.com/multiwfn/res/ELF_Chimera.pdf).

4.5.2 Laplacian of electron density of 1,3-butadiene

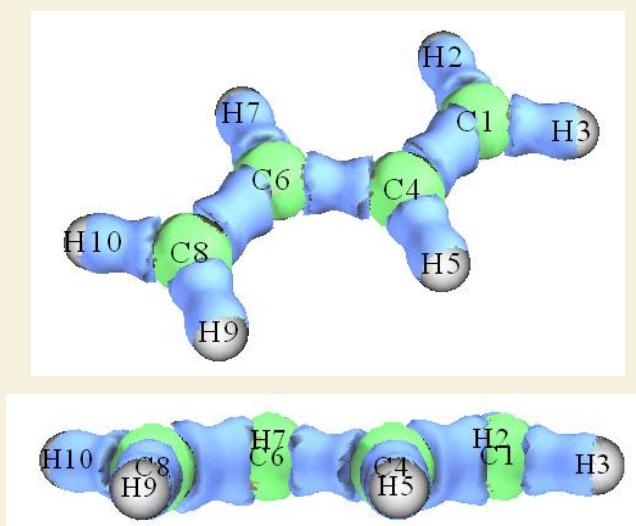
Electron density Laplacian is another useful real space function like ELF and LOL to reveal electronic structure. Laplacian is not as good as ELF and LOL for highlighting localization region due to its poor distinguishability. For example, the shell structures of atoms heavier than krypton cannot be 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 are difficult to be identified. Moreover, the value range of Laplacian is too large, which brings difficulties on visual analysis. However, for many systems Laplacian of electron density is still useful. In this example we will plot isosurface map of this function for 1,3-butadiene.

Boot up Multiwfn and input following commands

```
examples\butadiene.fch // Yielded at B3LYP/6-31G** level
5 // Generate grid data and view isosurface
3 // Electron density Laplacian
2 // Medium quality grid
-1 // View isosurface
```

In the newly occurred window, change isosurface value from default value to 0.3, then the

green and blue isosurfaces will correspond to isovalue of 0.3 and -0.3 respectively. The current graph will look like below



The presence of blue isosurfaces between C-C and C-H suggests that valence-shell electrons are strongly concentrated on these regions, this is typical pattern of covalent bonding. If you inspect the isosurfaces between carbon atoms carefully, you will find that the isosurface between C6-C8 or C1-C4 is wider than the isosurface between C4-C6, this phenomenon reflects that the two boundary C-C covalent bonds are stronger than the central one. This conclusion can also be verified by other wavefunction analysis schemes, for example, Mayer bond order (bond order between C6-C8 is 1.863, while the bond order between C4-C6 is only 1.136).

The isosurface map we obtained in this section is not quite smooth, this is because we only used medium quality grid. If you choose high quality grid instead, much better isosurface will be viewed.

4.5.3 Calculate ELF- α and ELF- π to study aromaticity of benzene

ELF- α and ELF- π indices are commonly used α and π aromaticity indices, they are defined as the ELF value at bifurcation point (*i.e.* (3,-1) type of CP) of ELF domains that solely contributed from α orbitals and π orbitals, respectively, see *J. Chem. Phys.*, **120**, 1670 (2004) and *J. Chem. Theory Comput.*, **1**, 83 (2005) 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. Strong multi-center delocalization is commonly recognized as 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.

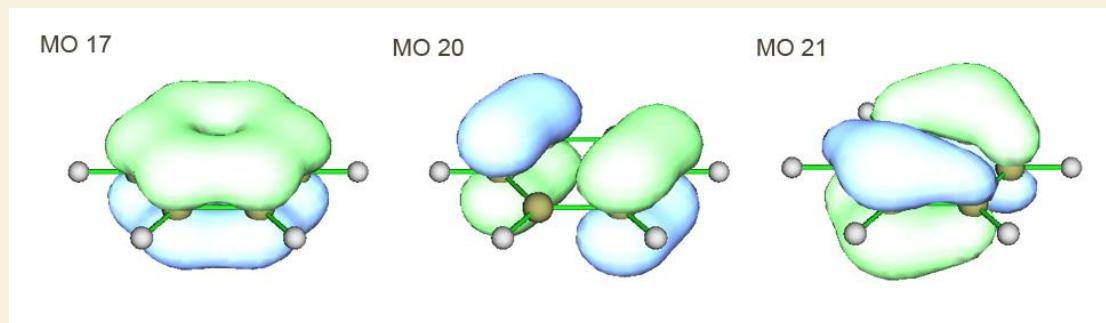
In order to separate α and π orbitals, we need to know which orbitals are π orbitals first. Boot up Multiwfn and input following commands

```
examples\benzene.wfn // Optimized at B3LYP/6-311G* level, the same level used in Chem. Phys. Lett., 443, 439
```

```
0 // View molecular orbitals (MOs)
```

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 σ orbitals to zero**.

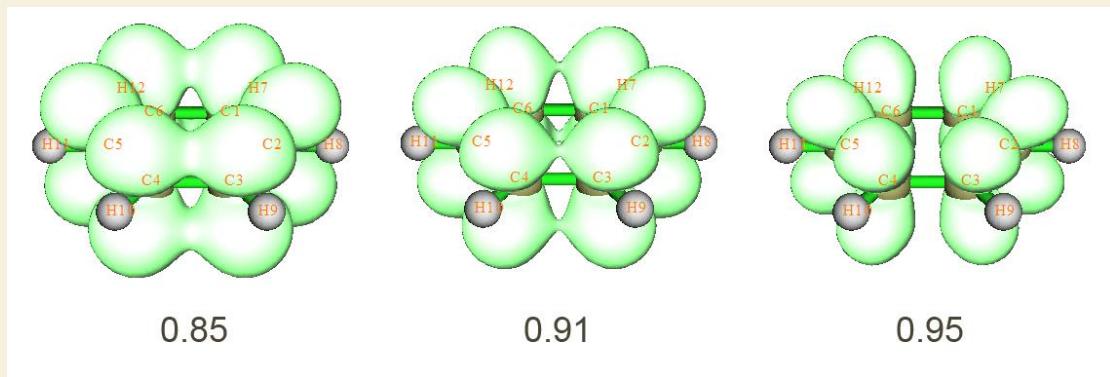
```

6 // Enter "Modify & Check wavefunction" interface
26 // Set occupation number for some orbitals
0 // Selecting all orbitals
0 // Set occupation number of all orbitals to zero
17,20,21 // Select MO 17, 20 and 21, namely all  $\pi$  orbitals
2 // Set occupation numbers of MO 17, 20 and 21 to 2.0 (doubly occupied). If you want to
check if occupation numbers have been correctly set, choose option 3. You will find occupation
number of all  $\alpha$  orbitals have become zero, namely they will have no contribution to all results
yielded in following calculations
q // Return to last menu
-1 // Return to main menu

```

Note: For this system, in fact there is a much more convenient way to set occupation number of all orbitals except for the π ones to zero. The procedure is: Enter subfunction 22 of main function 100, select 1 to choose actual molecular plane (in this example the plane is XY), then all π orbitals will be automatically identified, then choose 1 to set occupation number of all other orbitals to zero (or choose option 3, if heavier elements such as silicon are involved in present system). Finally, choose 0 to return to main menu. More details about automatic identification of π orbitals can be found in Section 3.100.22.

There are two ways to study ELF- π , the way 1 is to examine ELF isosurface directly, while the way 2 is performing topology analysis. Way 1 is more intuitive but less accurate than way 2. Here I illustrate way 1 first. Generate and view isosurface for ELF by main function 5 as usual (recall Section 4.5.1. Using High quality grid is recommended). This time the ELF isosurface only reflects π -electron localization character. By gradually increasing isovalue, you will find that the two circle-shape ELF domains are bifurcated to twelve spherical-like domains at about the isovalue of 0.91 (see below graph), implying that ELF- π index of benzene is about 0.91.

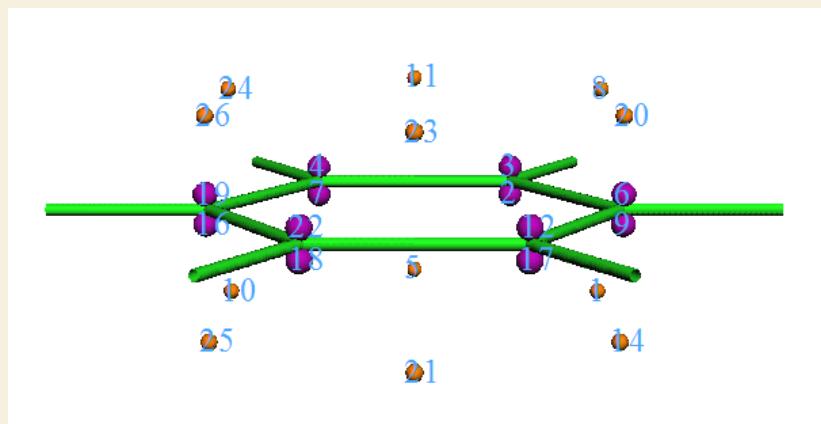


Next, let us use way 2 to evaluate ELF- π index again, this way is more rigorous than way 1. Choose 0 to return to main menu.

```

2 // Topology analysis
-11 // Select real space function
9 // ELF
6 // The guessing points will be scattered around each atom in turn. This searching mode is
the most appropriate one for locating ELF CPs
-1 // Start the CP search
-9 // Return to upper menu
0 // Visualize results. The resulting graph is shown below, two (3,+1) CPs are not shown

```



By comparing this graph with ELF isosurface map, it clear that the (3,-1) CPs (orange) are bifurcation positions of ELF domains, while (3,-3) CPs (purple) correspond to the maximum points of the twelves ELF domains. Now we check ELF value at a (3,-1) CPs, we can choose any one, since they are all equivalent.

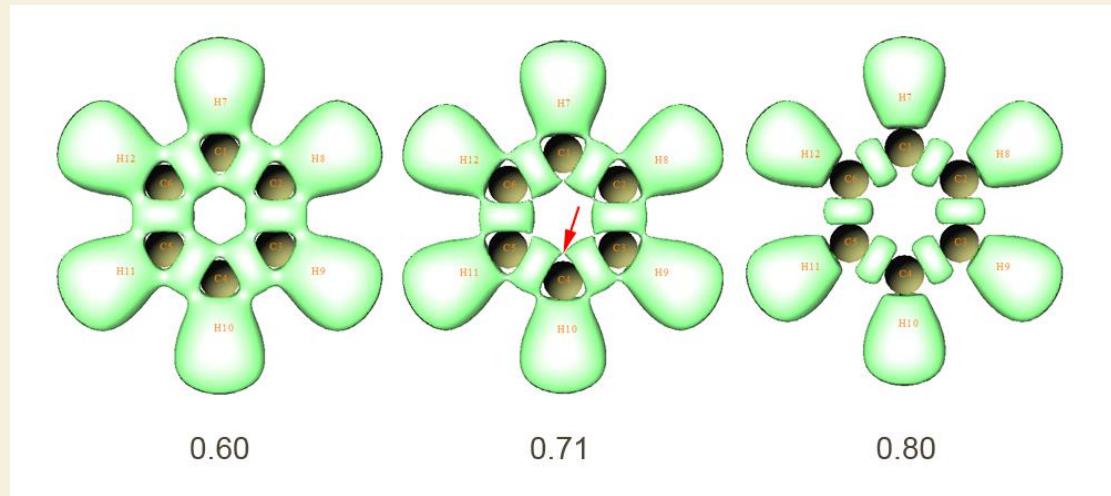
```
7 // Show all properties at a CP
```

```
23 // CP23
```

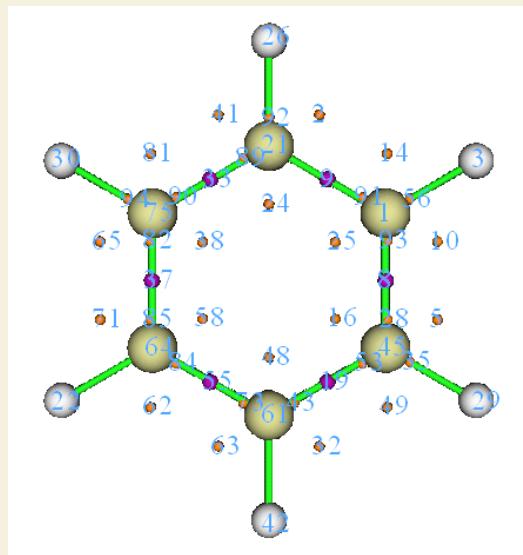
From the output, we find the ELF value at CP 23, namely ELF- π index of benzene is 0.91247, this result is in very good agreement with the value 0.913 given in *Chem. Phys. Lett.*, **443**, 439. Evidently, this value exceeds the criteria (0.70) of π aromaticity, suggesting that benzene has strong π aromaticity.

Now we calculate ELF- α for benzene. Reboot Multiwfn and load benzene.wfn, set occupation number of MO 17, 20 and 21 to zero (it is more convenient to use subfunction 22 in main function

100 to do this). Then generate isosurface for ELF as usual, gradually adjust isovalue, try to find out at which isovalue the domains corresponding to the α bonds between carbon atoms are bifurcated. One can finally find that at the isovalue equals to 0.71 the domain are bifurcated, suggesting that ELF- α index is about 0.71. The bifurcation point is pointed by red arrow in below graphs:



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.



Comparing positions of CPs with ELF isosurface map, it is clear that the (3,-1) CPs such as CP 48 and 58 correspond to bifurcation points of the α bond domains between the carbon atoms. Check ELF value at CP 48, we get 0.70907, which is the accurate ELF- α value. Our result agrees well 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 Use Fukui function and dual descriptor to study favorable site for electrophilic attack of phenol

(1) Fukui function

Fukui function is an important concept in conceptual density functional theory, it has been widely used in prediction of reactive site. Fukui function is defined as (see *J. Am. Chem. Soc.*, **106**, 4049 for original paper, see my paper [Acta Phys. -Chim. Sin., 30, 628 \(2014\)](#) for related discussions and comparison)

$$f(\mathbf{r}) = \left[\frac{\partial \rho(\mathbf{r})}{\partial N} \right]_{\nu}$$

where N is number of electrons in present system, the constant term ν in the partial derivative is external potential. Generally the external potential only comes from nuclear charges, so ν can be simply regarded as nuclear coordinates for isolated chemical system. It is argued that reactive sites have larger value of Fukui function than other 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:

$$\text{Nucleophilic attack: } f^+(\mathbf{r}) = \rho_{N+1}(\mathbf{r}) - \rho_N(\mathbf{r}) \approx \rho^{\text{LUMO}}(\mathbf{r})$$

$$\text{Electrophilic attack: } f^-(\mathbf{r}) = \rho_N(\mathbf{r}) - \rho_{N-1}(\mathbf{r}) \approx \rho^{\text{HOMO}}(\mathbf{r})$$

$$\text{Radical attack: } f^0(\mathbf{r}) = \frac{f^+(\mathbf{r}) + f^-(\mathbf{r})}{2} = \frac{\rho_{N+1}(\mathbf{r}) - \rho_{N-1}(\mathbf{r})}{2} \approx \frac{\rho^{\text{HOMO}}(\mathbf{r}) + \rho^{\text{LUMO}}(\mathbf{r})}{2}$$

In this example, we first attempt to find out the reactive sites for electrophilic attack of phenol by means of the Fukui function f^- shown above. The approximate form of Fukui function based on frontier orbitals will not be used here.

First, use quantum chemistry program to optimize structure and meantime generate wavefunction file for neutral state of phenol. Next, based on this geometry, create a input file of single point task, set charge and multiplicity to 1 and 2, respectively, then run this task to generate wavefunction file for +1 charged state of phenol. Notice that for the +1 charged state you **should not reoptimize molecule structure**, because ν (nuclear coordinates in this context) is defined as constant in the partial derivative of Fukui function. I assume that the two resulting .wfn files are named as *phenol.wfn* and *phenol+1.wfn*, you can find them in “examples” directory. The *phenol+1.wfn* corresponds to phenol with +1 net charge ($N-1$ electrons state)

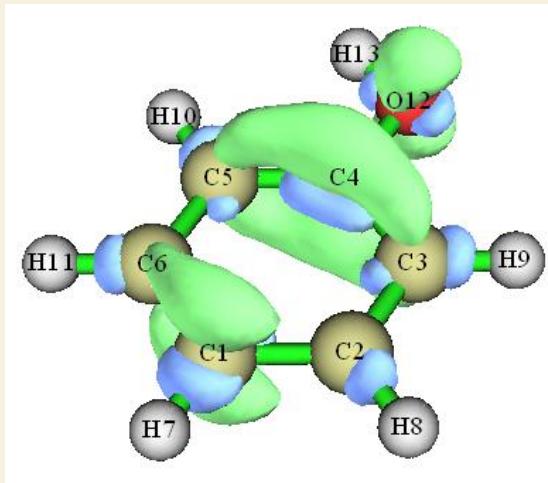
To study the isosurface of f^- , we need to properly use the “custom operation” feature of Multiwfn (see Section 3.7.1 for detail). Boot up Multiwfn and input following commands:

```
examples\phenol.wfn // Phenol at neutral state
5 // Calculate grid data
0 // Set custom operation
1 // Only one file will be operated with the file that has been loaded (namely phenol.wfn)
-,examples\phenol+1.wfn // “-“ is subtraction sign. Property of the firstly loaded file (namely phenol.wfn) will be subtracted by phenol+1.wfn.
```

1 // Electron density

2 // Medium quality grid

Now Multiwfn starts to calculate electron density grid data for *phenol.wfn*, then calculate that for *phenol+1.wfn*, and finally get their difference to yield grid data of f^- . We choose option -1 to check the isosurface, after adjusting the isovalue to a proper value (0.007), the graph will be



Clearly, most positive part of f^- function is localized on O12, C1, C3, C4 and C5, that means *para* and *ortho* positions of hydroxyl are favourable reactive sites for electrophilic attack, this conclusion is in agreement with common knowledge, namely hydroxyl group is an *ortho-para*-director.

Next, I illustrate how to plot Fukui function $f^0 = (\rho_{N+1} - \rho_{N-1})/2$ for propylene. Boot up Multiwfn and input:

examples\propylene\propylene-1.fch // -1 charged state, namely $N+1$ electrons state

5 // Calculate grid data

0 // Set custom operation

1 // One file will be operated with propylene-1.fch

-, *examples\propylene\propylene+1.fch* // +1 charged state, namely $N-1$ electrons state

1 // Electron density

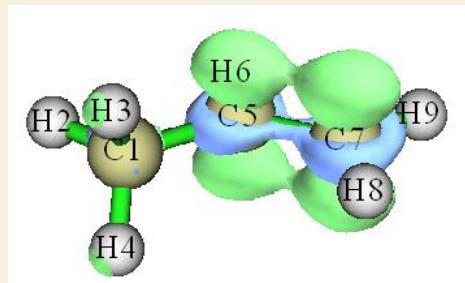
2 // Medium quality grid

6 // Divide all grid data by a factor

2 // Divided by 2

-1 // Visualize isosurface map

The isosurface map of $f^0 = 0.01$ is shown below



(2) Dual descriptor

Dual descriptor is another useful function used to reveal reactive sites, see *J. Phys. Chem. A*, **109**, 205 (2005) for detail. Formally, the definition of the dual descriptor Δf has close relationship with Fukui function:

$$\begin{aligned}\Delta f(\mathbf{r}) &= f^+(\mathbf{r}) - f^-(\mathbf{r}) \\ &= [\rho_{N+1}(\mathbf{r}) - \rho_N(\mathbf{r})] - [\rho_N(\mathbf{r}) - \rho_{N-1}(\mathbf{r})] = \rho_{N+1}(\mathbf{r}) - 2\rho_N(\mathbf{r}) + \rho_{N-1}(\mathbf{r})\end{aligned}$$

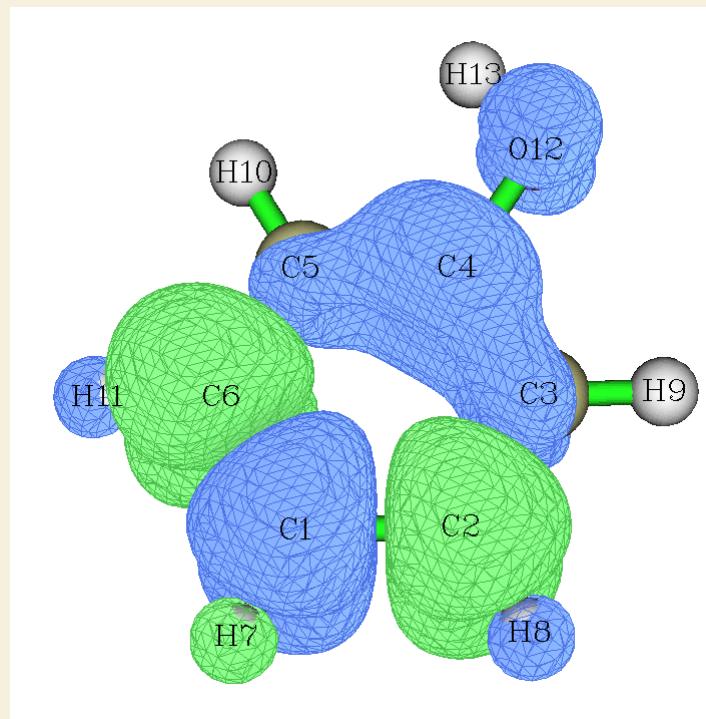
Worthnotingly, dual descriptor can also be evaluated in terms of spin density ρ^s . Since $\rho_{N+1} - \rho_N$ and $\rho_N - \rho_{N-1}$ can be approximated as ρ_{N+1}^s and ρ_{N-1}^s respectively, it is clear that $\Delta f(\mathbf{r}) \approx \rho_{N+1}^s(\mathbf{r}) - \rho_{N-1}^s(\mathbf{r})$. Commonly, there is no evident qualitative difference between the dual descriptor evaluated based on electron density of three states ($N+1, N, N-1$) and the one based on spin density of two states ($N+1, N-1$).

Unlike Fukui function, via Δf both types of reactive sites can be revealed simultaneously. It is argued that if $\Delta f > 0$, then the site is favorable for a nucleophilic attack, whereas if $\Delta f < 0$, then the site is favorable for an electrophilic attack. However, according to my experience, if your aim is to figure out which ones are more favorable among many potential sites, you do not need to concern the sign of Δf , you only need to study which sites have more positive or more negative of Δf . If the distribution of Δf around a site A is more positive than another site B , then one can say A is a more favorable site for nucleophilic attack than B , and meantime B is a more preferential site for electrophilic attack than A .

Here we calculate dual descriptor for phenol based on spin density of $N+1$ and $N-1$ states. Since we have calculated phenol+1.wfn earlier, now we only need to calculate phenol-1.wfn (this file has been provided in "example" folder). After that, boot up Multiwfn and input:

```
examples\phenol-1.wfn // Phenol anion, corresponding to N+1 electron system
5 // Calculate grid data
0 // Set custom operation
1 // Only one file will be operated with the file that has been loaded
-examples\phenol+1.wfn // Corresponding to N-1 electron system
5 // Electron spin density
2 // Medium quality grid
4 // Change the isovalue
0.005
-1 // Visualize isosurface of dual descriptor
```

Change isosurface style to "solid face+mesh" from menu of the GUI, then you will see below graph. The region covered by negative isosurfaces (blue) suggests that only *para*- and *ortho*-carbons were activated for electrophilic attack by hydroxyl group, this conclusion is in agreement with Fukui function.



If you would like to evaluate Δf in its exact form (based on ρ of three states), you can follow below steps:

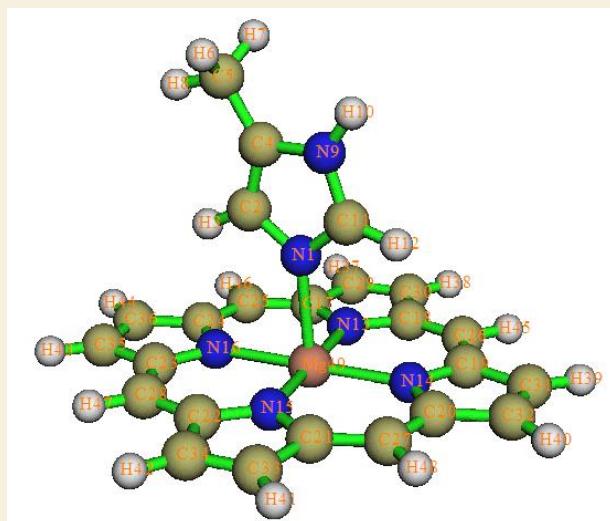
```
examples\phenol-1.wfn // N+1 electron system
5 // Calculate grid data
0 // Set custom operation
3 // Three files will be operated with the file that has been loaded
-,examples\phenol.wfn // N electron system
-,examples\phenol.wfn // N electron system
+,examples\phenol+1.wfn // N-1 electron system
1 // Electron density
```

The remaining steps are identical as above. If you visualize the isosurface, you will find this time the ortho-carbons do not have negative value of Δf , they are only very slightly more negative than the para-carbons, that means this time Δf does not have good ability to discriminate preferential sites. So, using exact form to evaluate Δf does not necessarily give rise to better result than using spin density to approximately evaluate Δf !

Above we used visualization manner to examine Fukui function and dual descriptor and obtained the conclusion what we expected. However, visual analysis is somewhat ambiguous and the discussion is inconvenient to be given in literature. Therefore sometimes we hope that the discussions of Fukui function and dual descriptor can be quantified, namely using a value for each atom to indicate the probability that it can be acted as reactive site. To do so, one should calculate "condensed" version of Fukui function and dual descriptor based on population analysis techniques. Since population analysis is exemplified in Section 4.7, the method for calculating condensed Fukui function and condensed dual descriptor will be deferred to be introduced as Section 4.7.3. Another scheme to study Fukui function and dual descriptor is to first partition the whole molecular surface to local surface corresponding to each atom, and then examine the their average values on these local surfaces. Because this scheme relies on quantitative molecular surface analysis technique, illustration is deferred to Section 4.12.4.

4.5.5 Plot difference map of electron density to study electron transfer of imidazole coordinated magnesium porphyrin

In this example, I will show you how to plot fragment electron density difference in Multiwfn. During coordination between imidazole and magnesium porphyrin, electron transfer and polarization occur, the variation of electron density can be clearly revealed by subtracting electron density of imidazole (referred to as NN below) and magnesium porphyrin (referred to as MN below) in their isolated states from the whole system (referred to as MN-NN below). The geometry of MN-NN is shown below:



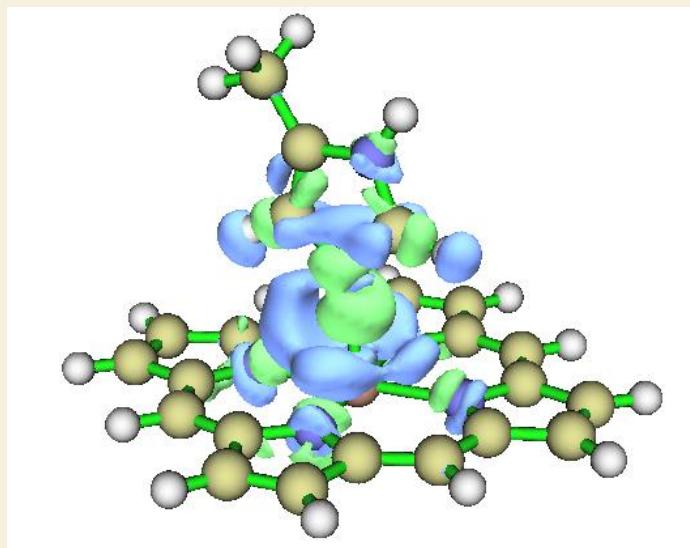
examples\MN-NN.gjf is Gaussian input file of the MN-NN system (geometry has been optimized), run it by Gaussian after modifying the .wfn output path at the last line, then MN-NN.wfn will be yielded. Next, respectively delete MN and NN parts from the MN-NN.gjf and properly modify .wfn output path and then save MN.wfn and NN.wfn (which have already been provided in "example" folder). Then run them by Gaussian to obtain MN.wfn and NN.wfn. It should be paid attention that by default, Gaussian always puts the system to standard orientation, which makes the coordinates in MN.wfn and NN.wfn inconsistent with MN-NN.wfn, and thus the density difference will be meaningless. Therefore, *nosymm* keyword must be specified in route section to avoid the automatic adjustment of coordinates. (The MN.wfn, NN.wfn and MN-NN.wfn can also be directly loaded from here: <http://sobereva.com/multiwfn/extrafiles/MN-NN.zip>)

Now we generate grid data of electron density difference by Multiwfn. Boot up Multiwfn and input following commands

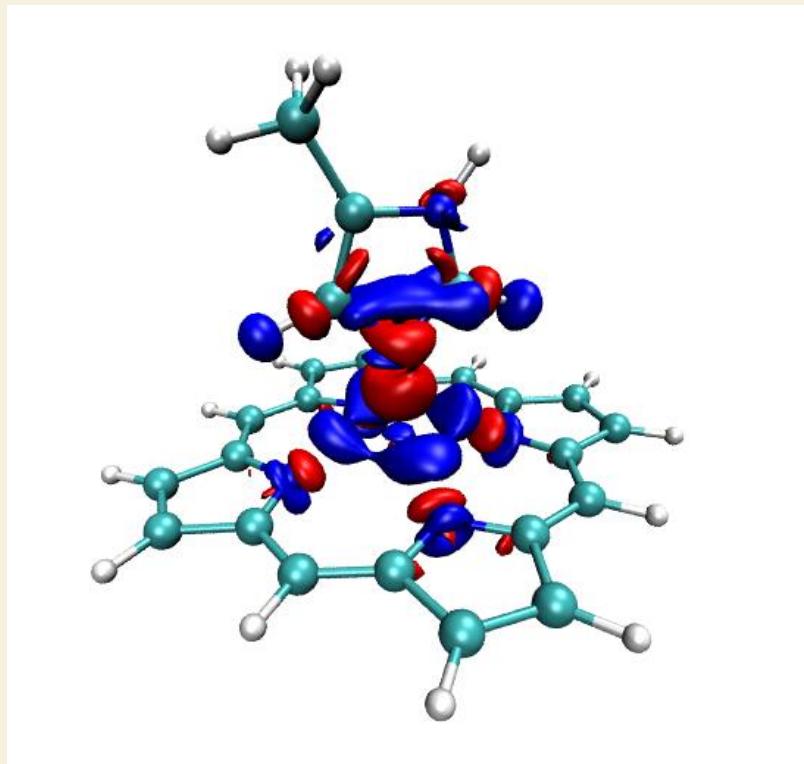
```
MN-NN.wfn
5 // Calculate grid data
0 // Set custom operation
2 // Two files will be operated with MN-NN.wfn
-,MN.wfn // Will subtract property of MN.wfn from that of MN-NN.wfn
-,NN.wfn // Will subtract property of NN.wfn from that of MN-NN.wfn
1 // The property is selected as electron density
3 // Since present system is relative huge, we need more grid points than normal cases, so we
choose high quality grid
```

After the calculation is finished, you can choose option -1 and then set isovalue to about 0.001

to visualize the isosurface of the grid data, as shown below.



Multiwfn is not very professional at visualization of grid data, for large size of grid data the visualization speed is relatively slow. You can choose option 2 to export the grid data to cube file and then visualize it by external tools, such as VMD, ChemCraft, Molekel and GaussView. Below is the graph generated by VMD (freely available at <http://www.ks.uiuc.edu/Research/vmd/>) based on the cube file generated by Multiwfn.



The red and blue isosurfaces (+0.0012 and -0.0012 a.u., respectively) represent the region in which electron density is increased and decreased after NN coordinated to MN, respectively. It is obvious that electron density is shifted from backside of nitrogen in NN toward magnesium atom to strengthen the coordination bond. Besides, it can be seen that the appearance of NN does not

perturbe electron density distribution of porphyrin ring remarkably, only slight polarization occurs on the four coordination nitrogens in MN.

Detailed steps of drawing above graph in VMD: First, drag the cube file density.cub into VMD main window. Select "Graphics" - "Representations", create a new representation by clicking "Create Rep" button, change the "drawing method" to "isosurface", set "Draw" to "solid surface", set "Show" to "Isosurface", change the isovalue to 0.0012, set "coloring method" to "ColorID" and choose red. Now the isosurface of positive part of density difference has been displayed. Then click "Create Rep" button again to create another representation, select blue in "ColorID" and change the isovalue to -0.0012. If you would like to use white background instead of the default black background, select Graphics - Colors - Display - Background - 8 white.

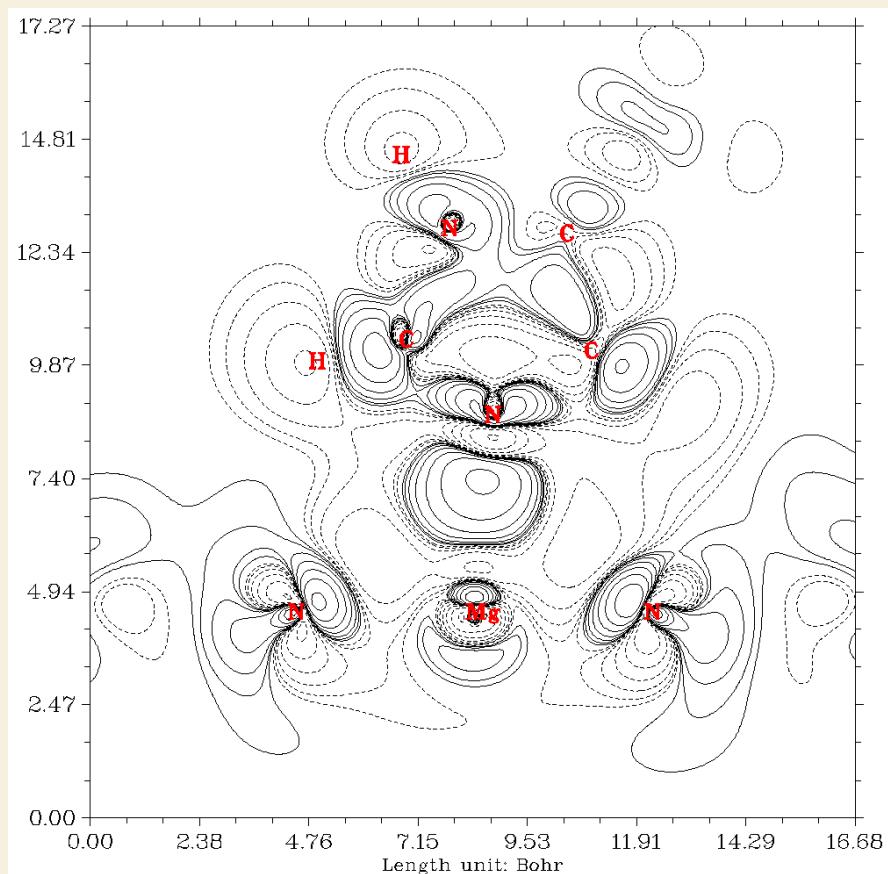
Next, let us generate contour map of electron density difference in the plane defined by atoms 16, 14, 9. Input following commands:

```
0 // Return to main menu
4 // Draw plane graph
0
2
-,MN.wfn
-,NN.wfn
1
2 // Contour line map
[Press ENTER button to use default grid setting]
4 // Define the plane by three atoms
16,14,9
```

Immediately the contour map pops up. The solid and dashed contour lines exhibit where electron density is increased and decreased, respectively. The contour lines in the graph are a bit sparse, so we adjust contour line setting to make the graph looks denser and thus more informative. Close the graph and then input

```
3 // Change contour line setting
9 // Generate contour value by geometric series
0.0001,2,30 // Start value, step size and the number of steps, respectively
y // Clean existing contour lines
9
-0.0001,2,30 // Set negative contour lines
n // Append the newly generated contour lines to existing ones
I // Save setting and return
-I // Redraw the graph
```

Below is the final graph, looks nice!



Hint: After completing the definition the contour lines, you can choose option 6 to save the setting to external plain text file. Next time you can directly load the setting by option 7.

Note that in Multiwfn, plotting difference map for electron density or other real space functions can be easily extended to more than two fragments cases, see Section 4.4.8 for example.

4.5.6 Study electron delocalization range function EDR($\mathbf{r};d$) of anionic water dimer

This section was contributed by Arshad Mehmood and slightly adapted by Tian Lu.

This example shows how to calculate electron delocalization range function $\text{EDR}(\mathbf{r};d)$ at user-defined length scale d for anionic water cluster $(\text{H}_2\text{O})_2^-$ (cf. *Phys. Chem. Chem. Phys.*, **17**, 18305 (2015)). Using $\text{EDR}(\mathbf{r};d)$ we can vividly inspect distribution of solvated electron.

Boot up Multiwfn and input following commands:

`examples\solvatedelectron.wfn // Anionic water dimer optimized at B3LYP/6-311++G(2d,2p)
level`

`5 // Calculate grid data`

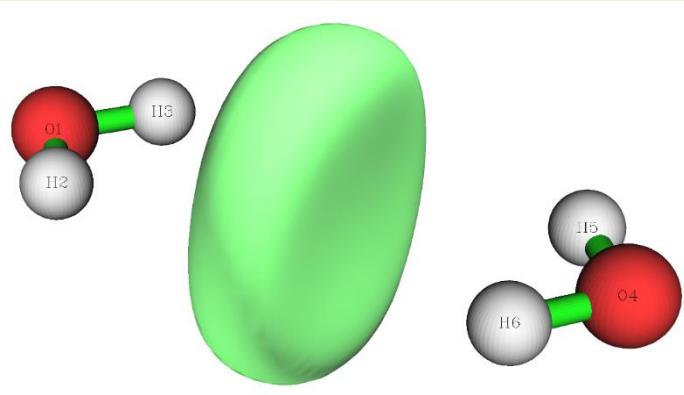
`20 // EDR($\mathbf{r};d$)`

`11.22 // Input length scale d (Bohr). Here we consider the relatively delocalized solvated electron at $d=11.22$ Bohr. Further details are given in J. Chem. Phys., 141, 144104 (2014).`

`2 // Medium quality grid`

-1 // Show isosurface graph

Now a GUI window pop up. Input isovalue of 0.74 in the “Isosurface value” box and press ENTER button. The following isosurface will appear.



This figure shows that at length scale of $d=11.22$ Bohr the solvated electron is between the two H_2O molecules.

4.5.7 Study orbital overlap distance function $D(\mathbf{r})$ of thioformic acid

This section was contributed by Arshad Mehmood and slightly adapted by Tian Lu.

This example will show the calculation procedure of orbital overlap distance function $D(\mathbf{r})$ of thioformic acid and map it on molecular electron density surface. If you are not familiar with $D(\mathbf{r})$, you can check entry 21 of Section 2.6 or *J. Chem. Theory Comput.*, **12**, 3185 (2016).

Boot up Multiwfn and input following commands:

examples\ThioformicAcid.wfn // Thioformic acid optimized at B3LYP/6-311++G(2d,2p)

5 // Calculate grid data

21 // Orbital overlap length function $D(\mathbf{r})$, which maximizes EDR($\mathbf{r}; d$) with respect to d

Now we need to set input total number, start and increment of EDR exponents $\alpha_i = 1/d_i^2$, since the overlap distance is fit using an even-tempered grid of exponents. The start value is the largest exponent (α_1), subsequent exponents are yielded by $\alpha_{i+1}/\alpha_i = 1/\alpha_{inc}$, where α_{inc} is increment. The default setting (i.e. $n=20$, $\alpha_1=2.50$, $\alpha_{inc}=1.50$) suffices for common systems. After selecting the manual input (option 1) or default setting (option 2), a list of exponents will be appeared, which will be used in evaluation of $D(\mathbf{r})$

2 // Medium quality grid

At this stage Multiwfn starts calculation. Wait until calculation is finished, then choose option 2 to export grid data of $D(\mathbf{r})$ as EDRDmax.cub in current folder. The next step is to generate molecular density isosurface.

0 // Return to main menu

5 // Calculate grid data

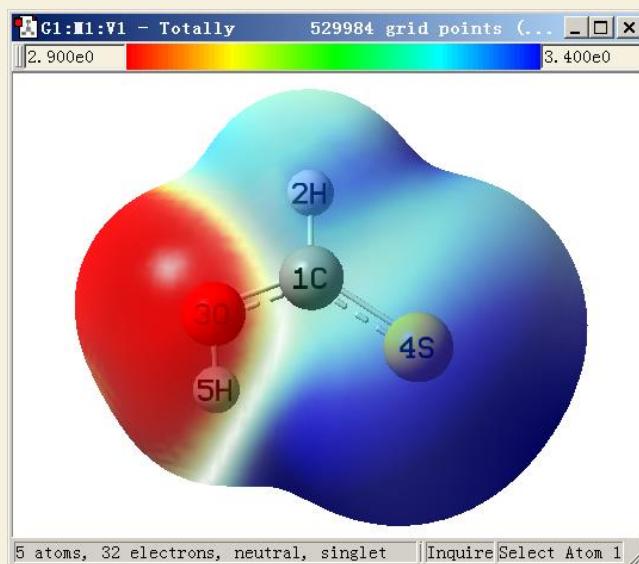
1 // Electron density

2 // Medium quality grid (The grid setting must be the same as for $D(\mathbf{r})$ calculation)

Then export grid data of electron density in current folder as density.cub by selecting option 2.

Based on EDRDmax.cub and density.cub, then the $D(\mathbf{r})$ grid data can be mapped on molecular density isosurface by many visualization programs, such as VMD and GaussView. Below is the $D(\mathbf{r})$

mapped electron density isosurface ($\rho=0.001$ a.u.) plotted by GaussView (If you do not know how to map a real space function using various colors on isosurface of another real space function based on two cube files via GaussView and VMD, you can consult my blog article <http://sobereva.com/402>, in Chinese).



As can be seen from the graph, $D(\mathbf{r})$ plotted between 2.9 Bohr (red) to 3.4 Bohr (blue) clearly distinguishes the chemically hard oxygen lone pair (red) from the softer sulfur lone pair (blue).

This system will be further studied in Section 4.12.8 by means of quantitative molecular surface analysis module.

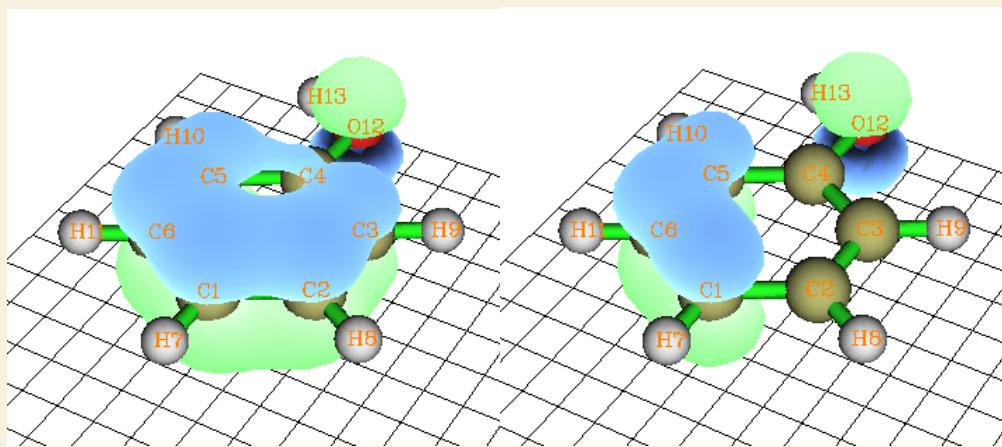
4.6 Modify and check wavefunction

4.6.1 Delete certain Gaussian functions

In subfunction 25 of main function 6, you can set orbital expansion coefficients of Gaussian type functions (GTFs) which satisfied certain conditions. If the coefficients are set to zero, that means the information 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. Boot up Multiwfn and input following commands

```
examples\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 only one orbital is selected
0 // Set coefficients of selected GTFs in orbital 23 to zero, that is delete their information
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 consisting of inner-core atomic orbitals to electron density for hydrogen cyanide, so that we can obtain valence electron density map. This is known as valence electron density analysis, which was demonstrated to be quite powerful for analyzing molecular electronic structure. Lots of examples of this kind of analysis can be found from my paper "Revealing Molecular Electronic Structure via Analysis of Valence Electron Density" [Acta Phys. -Chim. Sin., 34, 503 \(2018\)](#).

The realization of the valence density analysis is very simple. Inner-core atomic orbitals always have very low energies, so they can only contribute to the MOs with lowest energies. Hydrogen has no inner-core atomic orbital, while both carbon and nitrogen have an inner-core atomic orbital, hence what we need to do is to set occupation numbers of the first two MOs to zero (note that this is closed-shell wavefunction). Now, boot up Multiwfn, and input following commands

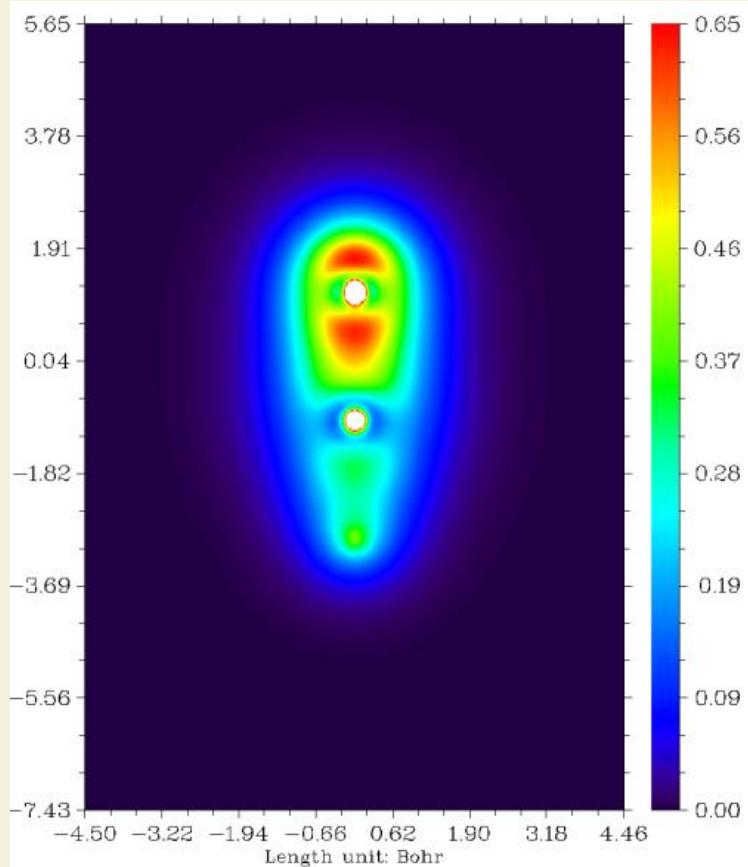
```
examples\HCN.wfn
6 //Modifying wavefunction
26 // Set occupation numbers
```

```

I,2 // Select MO 1 and 2
0 // Set their occupation numbers to zero
q // Return to last menu
-I // Return to main menu

```

Then, if we plot color-filled map of electron density as usual by main function 4, we will get below graph



From the picture, the bonding region of C-N and C-H and be easily identified, and the lone pair region of nitrogen can be clearly visualized. Note that the small white circles in carbon and nitrogen centers do not correspond to 1s electrons, but result from the fact that valence atomic orbitals have penetration effect into core region.

For convenience consideration, Multiwfn provides subfunction 34 in main function 6, one can directly choose it to set occupation numbers of all the MOs that consist of inner-core atomic orbitals to zero.

It is worth to mention that analysis of valence electron density is never limited to plotting analysis. For example, using main function 2 and 17, topology analysis and basin analysis can also be straightforwardly applied to valence electron density, respectively.

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 do not show any periodic character, of course you can enlarge your inputted system to multiple cells before PBC calculation, but you have to spend much more time for computation. Multiwfn provides a way to convert primitive cell wavefunction to a large supercell wavefunction, so that in the region you are interested the real space functions show periodic character.

Note: Beware that constructing supercell wavefunction in this manner is only a crude approximation! Since the orbital mix between neighbouring primitive cells is ignored, therefore the orbital wavefunctions of supercell cannot be faithfully reproduced.

In this section I use graphene as example. First, generate .fch (or .wfn) file of graphene primitive cell. The content of “molecular specification” field in the Gaussian input file is

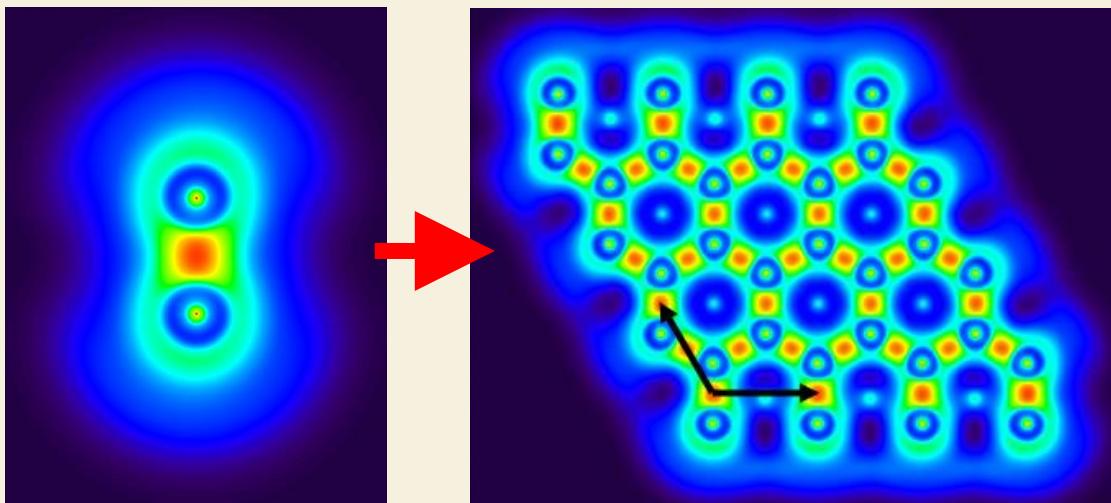
C	0.000000	0.000000	0.000000
C	0.000000	1.429118	0.000000
TV	2.475315	0.000000	0.000000
TV	-1.219952	2.133447	0.000000

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:

```
examples\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 should not 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 Population analysis and atomic charge calculation

4.7.0 Mulliken population analysis on triplet ethanol

In this section I will illustrate how to use Multiwfn to carry out Mulliken analysis, triplet ethanol is taken as instance. It is worth to note that Mulliken analysis is incompatible with diffuse functions, if diffuse functions are employed, the analysis result will be meaningless.

Boot up Multiwfn and input

```
examples\ethanol_triplet.fch // Calculated at UB3LYP/6-31G** level based on optimized singlet structure
```

```
7 // Population analysis and atomic charges
```

```
5 // Mulliken population analysis
```

1 // Output Mulliken analysis result. By default the result is outputted on screen, you can also select "-1 Choose output destination for option 1" to change the output destination to a specified plain text file

From the output, first you can find population of each basis function:

Population of basis functions:

Basis	Type	Atom	Shell	Alpha pop.	Beta pop.	Total pop.	Spin pop.
1	S	1(C)	1	0.99597	0.99597	1.99193	-0.00000
2	S	1(C)	2	0.34180	0.34330	0.68510	-0.00150
3	X	1(C)	3	0.34747	0.35290	0.70037	-0.00543
4	Y	1(C)	3	0.34869	0.35220	0.70088	-0.00351
...							
60	Z	8(O)	30	0.64840	0.09113	0.73953	0.55726
61	S	8(O)	31	0.30323	0.47550	0.77873	-0.17228
...							

Since present system is an open-shell system, not only the total population (*i.e.* alpha+beta), but also alpha and beta populations are outputted individually. The spin populations, which equal to difference between alpha and beta populations, are also printed. The output content is easy to understand, for example, from the output we can see that there are nearly two electrons located on

4 Tutorials and Examples

the first S basis function of C1 atom, and one of PZ basis functions of O8 atom has large amount of unpaired electrons (0.557).

Next, we can find population of each basis function shell of each atom:

Population of shells:

Shell	Type	Atom	Alpha pop.	Beta pop.	Total pop.	Spin pop.
1	S	1(C)	0.99597	0.99597	1.99193	-0.00000
2	S	1(C)	0.34180	0.34330	0.68510	-0.00150
3	P	1(C)	1.05533	1.06504	2.12037	-0.00971
4	S	1(C)	0.30449	0.30842	0.61291	-0.00393
...						
28	S	8(O)	0.99668	0.99637	1.99305	0.00032
29	S	8(O)	0.51929	0.45809	0.97739	0.06120
30	P	8(O)	1.71305	1.06458	2.77763	0.64846
...						

As you can see, for example, the basis function shell 30, which corresponds to one of P shells of O8, has unpaired electrons of 0.648 and total population of 2.777.

Next, you can find population of each angular moment atomic orbitals of each atom:

Population of each type of angular moment atomic orbitals:

Atom	Type	Alpha pop.	Beta pop.	Total pop.	Spin pop.
...					
8(O)	s	1.81920	1.92996	3.74916	-0.11076
	p	2.49431	1.65216	4.14647	0.84216
	d	0.00748	0.00607	0.01355	0.00142
...					
Total	s	8.35987	7.42761	15.78748	0.93226
	p	5.61927	4.56478	10.18405	1.05448
	d	0.02087	0.00761	0.02847	0.01326

The output shows that *d* type of atomic orbitals only have marginal contribution to total population (0.02847) and spin population (0.01326) of the whole system, since D type of basis functions only behave as polarization functions for present system. In O8, most unpaired alpha electrons are located on its *p* atomic orbitals, while slight unpaired beta electrons are distributed on its *s* atomic orbitals (positive and negative value indicate that the unpaired electrons are alpha and beta, respectively).

Finally, we can find atomic populations and atomic charges:

Population of atoms:

Atom	Alpha pop.	Beta pop.	Spin pop.	Atomic charge
1(C)	3.15460	3.16849	-0.01388	-0.32309
2(H)	0.43857	0.43159	0.00698	0.12984
3(H)	0.43857	0.43159	0.00698	0.12984
4(H)	0.46846	0.43410	0.03437	0.09744
5(C)	3.13963	2.95977	0.17986	-0.09940
6(H)	0.49613	0.36169	0.13444	0.14218

7(H)	0.49613	0.36169	0.13444	0.14218	
8(O)	4.32100	3.58819	0.73281	0.09082	
9(H)	1.04690	0.26289	0.78401	-0.30979	
Total net charge:		-0.00000	Total spin electrons:		2.00000

Triplet system has two unpaired electrons, one can see that in the triplet ethanol, most unpaired electrons (more than 1.5) are located on the hydroxyl group. It is well known that in ground state ethanol, oxygen atom should have significant negative charge due to very large electronegativity of oxygen. However, in present system, the oxygen even carries marginal positive charge. This observation reflects the fact that electronic structure of different electronic states may differ from each other remarkably.

Mulliken population analysis does not show population information of each atomic orbital, however, if you first identify correspondence between basis functions and atomic orbitals (see Section 4.7.6 for details), you can easily obtain population of each atomic orbital by simply summing up population of corresponding basis functions.

There are several other options in the Mulliken analysis interface, they can help you to gain deeper insight into electronic population, please play with them by consulting corresponding explanation in Section 3.9.3.

4.7.1 Calculate Hirshfeld and CHELPG atomic charges as well as fragment charge 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

```
examples\ClF3.wfn
7 // Population analysis and atomic charges
1 // Hirshfeld population
```

Hirshfeld population analysis requires electron density of atoms in their free-states, you need to choose a method to calculate atomic densities. Selecting 1 to use built-in atomic densities is very convenient, see Appendix 3 for detail; alternatively, you can select 2 to evaluate atomic densities based on atomic .wfn files, see Section 3.7.3 for detail. Here we choose option 1.

From the result we find the charges of the three fluorine atoms are unequal, the equatorial one is -0.075 while the axial ones have more electrons, are -0.224.

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 select “Electrostatic potential from atomic charges” in main function 3, 4 or 5 to study the electrostatic potential distribution from Hirshfeld charges.

Next, we calculate CHELPG charge. CHELPG charge has been introduced in Section 3.9.10. First select option 12 in the population analysis module, you will see a new menu, in general you do not need to modify the default options, and you can directly select option 1 to start the calculation. Since calculation of ESP is time-consuming, for large system you may need to wait for a while. The result is 0.5772 for Cl, -0.2496 for axial F and -0.0779 for equatorial F. The conclusion of CHELPG charge is the same as Hirshfeld charge, namely axial F are more negatively charged than the

equatorial one.

Fragment charge is defined as sum of charge of atoms constituting a fragment. You can manually sum up atomic charges to derive fragment charge; however, for large systems this process must be laborious. In Multiwfn it is possible to directly calculate charge for a fragment. For example, here we calculate CHELPG charge for the fragment composed by the two axial F atoms. Boot up Multiwfn and input

```
examples\ClF3.wfn
7 // Population analysis
-1 // Define fragment
2,4 // Index of the two axial F atoms
12 // CHELPG charge
1 // Start calculation
```

Since the fragment has been defined, Multiwfn not only prints atomic charges, but also prints fragment charge at the end of all output:

```
Fragment charge: -0.499331
```

Hint: Calculation of ESP fitting charges, including CHELPG, MK and so on, needs electrostatic potential (ESP) at fitting points. The calculation speed of ESP of cubegen utility in Gaussian package is much faster than the internal code of Multiwfn. If you have Gaussian installed on your system and you want to significantly reduce time cost in the calculation of ESP fitting charges, it is **strongly** recommended to set *cubegenpath* parameter in settings.ini file to actual path of cubegen, so that cubegen could be automatically invoked by Multiwfn at the right time to evaluate the ESP at fitting points. More information about this point can be found in Section 5.7.

4.7.2 Calculate and compare ADCH atomic charges with Hirshfeld atomic charges for acetamide

The ADCH (atomic dipole moment corrected Hirshfeld population) charge proposed by me is an improved version of Hirshfeld charge, it resolved many inherent drawbacks of Hirshfeld charge, such as poor dipole moment reproducibility, see Section 3.9.9 for brief introduction and [J. Theor. Comput. Chem., 11, 163 \(2012\)](#) for discussion and comparison. I highly recommend using ADCH charge to characterize charge distribution. The calculation process of ADCH charges is exactly identical to the one described in last section, the only difference is that you should select option 11 instead of option 1 in population analysis interface. For example, here we calculate ADCH charges for CH₃CONH₂. Boot up Multiwfn and input

```
examples\CH3CONH2.fch
7 // Population analysis and atomic charges
11 // Calculate ADCH charges
1 // Use built-in atomic densities in free-state
```

Multiwfn will calculate Hirshfeld charges first, and then perform atomic dipole moment correction for them to yield ADCH charges. The result is shown below

```
===== Summary of atomic dipole moment corrected (ADCh) charges =====
```

```

Atom: 1C Corrected charge: -0.265840 Before: -0.090370
Atom: 2H Corrected charge: 0.096194 Before: 0.037254
Atom: 3H Corrected charge: 0.105929 Before: 0.043058
Atom: 4H Corrected charge: 0.117894 Before: 0.048339
Atom: 5C Corrected charge: 0.272281 Before: 0.170596
Atom: 6O Corrected charge: -0.364414 Before: -0.308866
Atom: 7N Corrected charge: -0.677574 Before: -0.159120
Atom: 8H Corrected charge: 0.355620 Before: 0.131920
Atom: 9F Corrected charge: 0.359828 Before: 0.127108
Summing up all corrected charges: -0.0000816

```

Note: The values shown after "Corrected charge" are ADCH charges, the ones after "Before" are Hirshfeld charges

```

Total dipole from ADC charges (a.u.) 1.4368131 Error: 0.0001385
x/y/z of dipole moment from the charge (a.u.) 0.0432390 -1.4253486 0.1759079

```

It is obvious that for all atoms, the magnitude of ADCH charges are evidently larger than Hirshfeld charges, the former are in agreement with common chemical senses, while the latter turns out to be too small.

A remarkable feature of ADCH charges is that the molecular dipole moment can be exactly reproduced. The dipole moment derived from ADCH charges is 1.4368 a.u. (as shown above), which is exactly identical to the actual dipole moment, namely the one derived based on present electron density distribution. (The error 0.0001385 comes from trivial numerical aspects and is totally negligible).

If you scroll up the command-line window, you will find the following information

```
Total dipole from atomic charges: 1.073849 a.u.
```

This is the dipole moment derived from Hirshfeld charges, which deviates to actual dipole moment (1.4368 a.u.) apparently. In fact, for almost all small molecules, Hirshfeld charges always severely underestimate molecular dipole moments.

4.7.3 Calculate condensed Fukui function and condensed dual descriptor

In Section 4.5.4, I have introduced how to calculate and visualize Fukui function and dual descriptor. In this section, we will calculate "condensed" version of these two functions, so that the discussion of the possibility that an atom could act as a reactive site can be upgraded to quantitative level. Phenol will still be used as example case.

Before calculating phenol, we first derive the expression of condensed Fukui function and condensed dual descriptor. In the condensed version, atomic population number is used to represent the amount of electron density distribution around an atom. Recall the definition of Fukui function f^+ :

$$f^+(\mathbf{r}) = \rho_{N+1}(\mathbf{r}) - \rho_N(\mathbf{r})$$

The definition of condensed Fukui function for an atom, say A, can be written as

$$f_A^+ = p_{N+1}^A - p_N^A$$

where p^A is the electron population number of atom A.

Since atomic charge is defined as $q^A = Z^A - p^A$, where Z is the charge of atomic nuclear, the f^+ can be expressed as the difference of atomic charges in two states (note that the two Z terms are cancelled)

$$f_A^+ = q_N^A - q_{N+1}^A$$

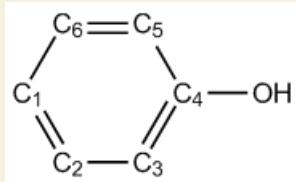
By analogous treatment, one can easily formulate other types of condensed Fukui function

Nucleophilic attack: $f_A^+ = q_N^A - q_{N+1}^A$
Electrophilic attack: $f_A^- = q_{N-1}^A - q_N^A$
Radical attack: $f_A^0 = (q_{N-1}^A - q_{N+1}^A)/2$

Similarly, condensed dual descriptor can be written as

$$\Delta f_A = f_A^+ - f_A^- = (q_N^A - q_{N+1}^A) - (q_{N-1}^A - q_N^A) = 2q_N^A - q_{N+1}^A - q_{N-1}^A$$

There are numerous ways to calculate atomic charge, although currently there is no consensus that which method is the most suitable to calculate condensed Fukui function and dual descriptor, in my opinion, Hirshfeld charge and NPA charge are the ones most worth to be recommended for this purpose. For example, *J. Phys. Chem. A*, **106**, 3885 illustrated how successfully is Hirshfeld charge used to study reactive site by means of condensed Fukui function.



According to the method introduced in Section 4.5.4, we calculate Hirshfeld charges for all carbons in phenol in its N , $N+1$ and $N-1$ electrons states, respectively. The atomic charges are listed in below table as column 2, 3 and 4. The molecular wavefunction files used here are the same with the ones used in Section 4.5.4. By the formulae shown above, the condensed f^- and f^+ and dual descriptor can be readily calculated, and they are listed below as the last three columns.

	N	$N-1$	$N+1$	f^-	f^+	Δf
C1	-0.0596	0.0725	-0.0992	0.1321	0.0396	-0.0925
C2	-0.0249	0.0297	-0.1620	0.0546	0.1371	0.0825
C3	-0.0597	0.0375	-0.2043	0.0973	0.1446	0.0473
C4	0.1017	0.2526	0.0619	0.1510	0.0397	-0.1112
C5	-0.0740	0.0153	-0.2112	0.0893	0.1372	0.0479
C6	-0.0276	0.0416	-0.1649	0.0691	0.1373	0.0682

Note: The Hirshfeld charges reported above were estimated based on build-in sphericalized atomic densities.

For f^- , the smallest two values occur at C2 and C6, therefore *meta* atoms are unfavorable sites for electrophilic attack.

For dual descriptor, the most positive values occur at C2 and C6, suggesting that they are the most unfavorable sites for electrophilic attack. C4 has a large negative value and hence favored by electrophilic reactant. Although the two *ortho* carbons (C3 and C5) have positive value, its magnitude is not as large as *meta* carbons, so dual descriptor indicates that *ortho* carbons are more possible than *meta* carbons to be reactive site for electrophilic attack. Our conclusion is completely in line with that of Section 4.5.4, in which we obtained the conclusion by visual inspecting isosurface of Fukui function and dual descriptor.

PS: In Section 4.5.4, when we calculate dual descriptor, we used spin density as approximation for electron density difference between two states. However, it seems that this approximation is not necessary for calculating condensed dual descriptor.

4.7.4 Illustration of computing Hirshfeld-I atomic charges

Hirshfeld-I (HI) is a more advanced technique to define atomic spaces than its predecessor (Hirshfeld). In this section I will illustrate how to compute HI charges using two systems as examples. Before reading this section please read Section 3.9.13 to gain basic knowledge of HI method and its implementation in Multiwfn.

(1) CH₃COCl

Here we calculate HI charges for CH₃COCl. Before calculation, you should properly set "gaupath" in settings.ini file to actual Gaussian executable file. Boot up Multiwfn and input

```
examples\CH3COCl.wfn // Generated at B3LYP/6-31G* level
7 // Population analysis and atomic charges
15 // Hirshfeld-I method
1 // Start calculation with default settings
B3LYP/6-31G* // The keyword of Gaussian used to calculate atomic .wfn files
```

From the prompts shown on screen, you can find that Multiwfn invokes Gaussian to calculate atomic .wfn files for all elements involved in present molecule at various charge states. Then Multiwfn converts atomic .wfn files to .rad files, which record spherically averaged atomic radial density. The automatically generated Gaussian input file (.gjf), the resulting Gaussian output file (.out or .log) and the .rad files are all produced in "atmrad" subdirectory of current folder, you can manually inspect them if you have interesting.

Once all .rad files have been generated, Multiwfn initializes some data, and then starts to carry out HI iteration, the convergence status can be seen on the screen:

```
Cycle 1
Cycle 2 Maximum change: 0.215441
Cycle 3 Maximum change: 0.154451
Cycle 4 Maximum change: 0.105326
Cycle 5 Maximum change: 0.080011
...

```

The "maximum change" denotes the maximum change of HI atomic charges, the iteration continues until "maximum change" is smaller than the given criterion. After convergence, Multiwfn prints final HI atomic charges:

```
Atom 1(C ): -0.667920
Atom 2(H ): 0.194766
```

```
Atom 3(H ): 0.188545
Atom 4(H ): 0.194766
Atom 5(C ): 0.755439
Atom 6(O ): -0.414041
Atom 7(Cl): -0.251555
```

Then you can choose if outputting these charges to .chg file in current folder. I suggest you compare above result with Hirshfeld charges, you will find the magnitude of HI charges is much higher than Hirshfeld charges. This phenomenon is expected, because HI atomic spaces properly contract or expand with respect to that of neutral state according to actual chemical environment, hence the difference of diffuse extent of atomic space between various atoms is greatly increased.

If you do not delete the "atmrad" folder or clean it up, then when you recalculate HI charges for CH₃COCl, or calculate a molecule only consisting of C, H, O and Cl elements (or some of them), Multiwfn will directly perform HI calculation based on existing .rad files in "atmrad" folder rather than invokes Gaussian to calculate any new file.

(2) Pt(NH₃)₂Cl₂

The Pt(NH₃)₂Cl₂.wfn in "examples" folder was calculated by B3LYP functional, Lanl2DZ pseudo-potential basis set was used to Pt, while 6-31G* was applied to other atoms. To calculate HI charges for this system, two steps are needed since we are unable to specify calculation level using a single set of keywords. In the first step, we arbitrarily use a file solely containing Pt atom as input file, and specify B3LYP/Lanl2DZ keyword during HI charge calculation, so that .rad files of Pt can be generated at this level. In the second step, we use Pt(NH₃)₂Cl₂.wfn as input file to calculate HI charges as usual, in the meantime we specify B3LYP/6-31G* keyword to generate .rad files for all remaining elements. The .rad files of Pt will not be calculated in this step at B3LYP/6-31G* level because they have existed in "atmrad" folder.

Notice that "atmrad" folder should be deleted if it has existed in current folder, otherwise Multiwfn may employs old files from it. Now let's start.

Firstly, we build a .xyz file containing only one Pt atom, namely editing a plain text file named "Pt.xyz" and fill below content into it:

```
1
test
Pt 0. 0. 0.

Now boot up Multiwfn and input
Pt.xyz
7
15
I
B3LYP/lanl2DZ // The level used to generate .wfn files for Pt
```

After generation of .rad files is finished, Multiwfn suddenly crashes, because the input file does not contain wavefunction information, and thus the HI charge cannot be computed. Then boot up Multiwfn again and input below commands:

```
examples\Pt(NH3)2Cl2.wfn
7
15
```

*I**B3LYP/6-31G**

From the prompts on screen you can find the files corresponding to Pt are not calculated since they have already existed. The final HI charges are

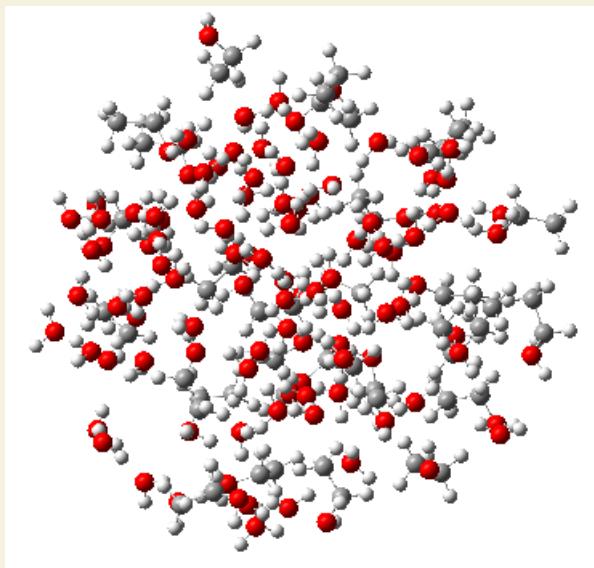
```
Atom 1(Pt): 0.410827
Atom 2(Cl): -0.414569
Atom 3(Cl): -0.414549
Atom 4(N ): -0.928921
Atom 5(H ): 0.382603
Atom 6(H ): 0.382600
Atom 7(N ): -0.928945
Atom 8(H ): 0.382606
Atom 9(H ): 0.382603
Atom 10(H ): 0.372874
Atom 11(H ): 0.372872
```

The result is in good agreement with our common chemical intuition.

Worthnotingly, the support of Hirshfeld-I in Multiwfn is never limited to population analysis, this partition manner can also be used to calculate orbital composition (main function 8), and be applied in fuzzy analysis module (main function 15).

4.7.5 Calculating EEM atomic charges for ethanol-water cluster

Please first read Section 3.9.15 to understand basic features of Electronegativity Equalization Method (EEM) charges before following this example. Here we calculate EEM charges for ethanol-water cluster, which contains as many as 492 atoms:



Evidently, calculating atomic charges by quantum chemistry method for such a large system is too expensive; however, as you will see, evaluation of EEM charges even for a system composed by hundreds of atoms is rather easy.

Note that in order to calculate EEM charges in Multiwfn, currently you have to use MDL molfile (.mol) as input file, because only this file provides atomic connectivity information, which

is needed in the calculation of EEM charges.

Boot up Multiwfn and input below commands

```
examples\ethanol_water.mol // This is a snapshot of molecular dynamics simulation  
7 // Population analysis and atomic charges  
17 // EEM charge  
0 // Start calculation
```

You will immediately see

```
EEM charge of atom 1(O ): -0.658886  
EEM charge of atom 2(H ): 0.322520  
EEM charge of atom 3(H ): 0.270187  
...  
EEM charge of atom 488(C ): -0.071564  
EEM charge of atom 489(H ): 0.145191  
EEM charge of atom 490(H ): 0.125555  
EEM charge of atom 491(O ): -0.616013  
EEM charge of atom 492(H ): 0.310970  
Electronegativity: 2.454144
```

The default EEM parameters were fitted by some researchers for reproducing B3LYP/6-31G* CHELPG charges, therefore, the above EEM charges should be close to CHELPG charges evaluated at B3LYP/6-31G* level (In fact, for present system, even if calculation of CHELPG charges is feasible, the result should be much worse than the EEM charges we just obtained. Because it is well-known that the quality of electrostatic fitting charges is very low for the atoms far from van der Waals surface, while in present system there are numerous heavily buried atoms).

Note that there are also many other built-in EEM parameters, you can choose them via option 1 before calculation.

4.7.6 Determining correspondence between basis functions and atomic orbitals via population analysis

Determine correspondence between basis functions and atomic orbitals is important if one want to plot PDOS of some atomic orbitals via main function 10, or to evaluate contribution to molecular orbitals from specific atomic orbitals using main function 8. The correspondence is easy to be identified if Pople basis set is used. For example, 6-31G* implies using one basis function with contraction degree of 6 to represent each inner atomic orbital, while each valence atomic orbital is represented by a basis function with contraction degree of 3 and an uncontracted basis function. However, for most of other type of basis sets, the correspondence is often difficult to be determined. Fortunately, as will be illustrated in this section, if one studies total and spin population of basis function shells via Mulliken population analysis, the correspondence can be unambiguously identified.

Two typical examples will be given below, more examples and discussions can be found from my blog article (<http://sobereva.com/418>, in Chinese). In below text atomic orbitals will be denoted as lower case (e.g. s, p, d...), while basis functions will be written as upper case (e.g. S, P, D...).

Example 1: cc-pVTZ for sulfur

Sulfur atom has configuration of $1s^2 2s^2 2p^6 3s^2 3p^4$, the ground state is triplet. The *example\sulfur_cc-pVTZ.fch* is the .fch file calculated at B3LYP/cc-pVTZ level by Gaussian16 for a single sulfur atom at its triplet state. Load this file into Multiwfn, then input

```
7 // Population analysis and atomic charges
5 // Mulliken analysis
1 // Output Mulliken analysis result
```

You will immediately see

Shell	Type	Atom	Alpha_pop.	Beta_pop.	Total_pop.	spin_pop.
1	S	1(s)	0.99997	0.99997	1.99994	-0.00000
2	S	1(s)	0.94424	0.94387	1.88812	0.00037
3	S	1(s)	0.60678	0.56547	1.17225	0.04131
4	S	1(s)	0.12371	0.13260	0.25631	-0.00889
5	S	1(s)	0.32408	0.35782	0.68189	-0.03374
6	P	1(s)	2.93141	2.90842	5.83983	0.02299
7	P	1(s)	1.58628	0.47788	2.06416	1.10840
8	P	1(s)	0.59303	0.26385	0.85688	0.32918
9	P	1(s)	0.88863	0.34985	1.23848	0.53879
10	D	1(s)	0.00064	0.00017	0.00081	0.00048
11	D	1(s)	0.00058	0.00011	0.00069	0.00046
12	F	1(s)	0.00065	0.00000	0.00065	0.00064

We want to identify which S basis functions respectively correspond to 1s, 2s and 3s atomic orbitals, and which P basis function shells respectively correspond to 2p and 3p atomic orbital shells.

All the two unpaired electrons of triplet sulfur atom are distributed on 3p shell, since the sum of spin population of 7P, 8P and 9P is $1.10840+0.32918+0.53879=1.976$, which is nearly equal to two, we can say that these three P shells correspond to 3p shell. The remainder 6P shell clearly corresponds to 2p shell, this can also be confirmed that its population number is 5.840, which is close to expected occupation number of 2p shell (6.0).

Then we check the case of S shells. The sum of population number of 3S, 4S and 5S is $1.17225+0.25631+0.68189=2.110$, which is close to actual occupation number of 3s atomic orbital (2.0); considering that occupation number of both 1S and 2S are close to 2.0, it can be concluded that 1S, 2S and (3S,4S,5S) mainly represent 1s, 2s and 3s atomic orbitals, respectively.

Example 2: def2-TZVP for Au

For Au atom, def2-TZVP is a pseudopotential basis set with Stuttgart small core pseudopotential, 60 inner electrons are replaced with pseudopotential, therefore only the valence electrons $5s^2 5p^6 5d^{10} 6s^1$ are explicitly represented by the def2-TZVP basis set. The *example\Au_def2-TZVP.fch* is the .fch file calculated at B3LYP/def2-TZVP level by Gaussian16 for a single Au atom at its ground state (doublet state). Load this file into Multiwfn and carry out population analysis as the last example, you will see

Shell	Type	Atom	Alpha_pop.	Beta_pop.	Total_pop.	spin_pop.
1	S	1(Au)	0.01764	0.01588	0.03352	0.00175
2	S	1(Au)	-0.25037	-0.22638	-0.47675	-0.02399
3	S	1(Au)	0.90648	0.84814	1.75462	0.05834
4	S	1(Au)	0.33047	0.35849	0.68895	-0.02802
5	S	1(Au)	0.60949	0.00436	0.61385	0.60513

6	S	1(Au)	0.38630	-0.00049	0.38581	0.38679
7	P	1(Au)	1.32302	1.32562	2.64864	-0.00260
8	P	1(Au)	1.43496	1.44146	2.87642	-0.00650
9	P	1(Au)	0.24145	0.23250	0.47395	0.00896
10	P	1(Au)	0.00057	0.00042	0.00099	0.00015
11	D	1(Au)	3.11664	3.17522	6.29186	-0.05859
12	D	1(Au)	1.48961	1.45237	2.94198	0.03724
13	D	1(Au)	0.39375	0.37241	0.76616	0.02135
14	F	1(Au)	0.00000	0.00000	0.00000	0.00000

Undoubtedly, all P shells (7P, 8P, 9P, 10P) represent the only p shell (5p), while all D shells (11D, 12D, 13D) represents the only d shell (5d). Since the sum of population number of 5S and 6S (*i.e.* 0.61385+0.38581) is exactly equal to 1.0, and meantime the sum of spin population number of 5S and 6S is also equal to 1.0, it is clearly that the 5S and 6S shells collectively represent the 6s atomic orbital, which has a single unpaired electron. The total electrons in the other four S shells (1S, 2S, 3S, 4S) is almost exactly 2.0, evidently the doubly occupied 5s is mainly represented by them.

4.7.7 Illustration of deriving RESP charges and normal ESP fitting charges with extra constraints

In this section I will take four examples to substantially illustrate the use of the extremely powerful and flexible RESP module of Multiwfn, which can very conveniently calculate standard RESP atomic charges and normal ESP fitting charges with equivalence constraints and/or charge constraints. If you have not read Section 3.9.16, reading it is strongly recommended to guarantee that you have enough knowledges about the RESP module as well as adequate understanding on the idea of ESP fitting method.

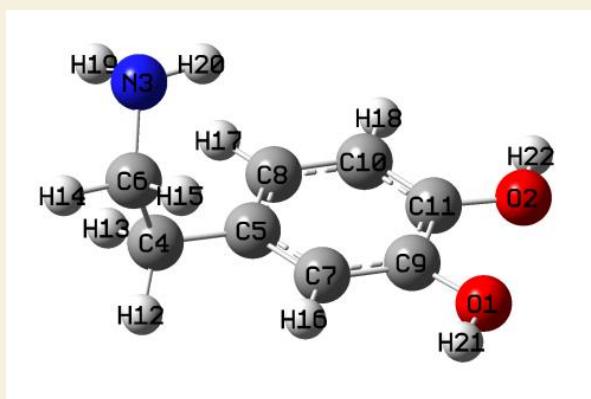
More detailed descriptions and discussions can be found from my blog "Principle of RESP charge and its calculation in Multiwfn" (in Chinese, <http://sobereva.com/441>).

For saving space, only the most important files involved in below examples are provided in "example\RESP" folder, while other files, including Gaussian output files and .fch files, can be downloaded at <http://sobereva.com/multiwfn/extrafiles/RESP.zip>.

As mentioned in Section 3.9.16, if you have Gaussian on your machine, it is highly suggested to enable utilizing cubegen utility in Gaussian package for evaluating ESP at fitting points, the computational cost of the ESP fitting charges could thus be significantly reduced.

Example 1: Deriving RESP charges for dopamine in ethanol environment

In this section I introduce the procedure of calculating standard RESP atomic charges for dopamine. Ethanol solvent environment is assumed and it will be represented using IEFPCM implicit solvation model. The structure of dopamine is shown as below.



Commonly, the geometry used for deriving RESP charges should be optimized at reasonable level. Above geometry was optimized at B3LYP-D3(BJ)/6-311G** level with IEFPCM implicit solvation model, and it was found to be the most stable geometry of present molecule.

Now, use Gaussian to run *examples\RESP\dopamine-single\dopamine.gjf* to generate corresponding .fch file for this geometry. As can be seen in the .gjf file, the keywords are *b3lyp/6-311g(d,p) scrf=solvent=ethanol*, this combination is not expensive while the resulting wavefunction is completely adequate to yield reliable RESP charges.

Boot up Multiwfn and input

```
dopmaine.fch // The .fch file just yielded
7 // Population analysis
18 // RESP module
```

1 // Calculate standard RESP charges using two-stage fitting procedure

During the calculation, Multiwfn first sets up atomic radii and determines position of fitting points, and then calculates ESP values at the fitting points. After that, the first stage of standard RESP calculation starts, the parameters and conditions employed in this stage can be found from outputted information:

```
**** Stage 1: RESP fitting under weak hyperbolic penalty
No charge constraint and equivalence constraint are imposed in this stage
Convergence criterion: 0.00000100
Hyperbolic restraint strength (a): 0.000500  Tightness (b): 0.100000
Iter: 1 Maximum charge variation: 1.0067226888
Iter: 2 Maximum charge variation: 0.0503421833
Iter: 3 Maximum charge variation: 0.0040157938
Iter: 4 Maximum charge variation: 0.0003426287
Iter: 5 Maximum charge variation: 0.0000329953
Iter: 6 Maximum charge variation: 0.0000032389
Iter: 7 Maximum charge variation: 0.0000003207
Successfully converged!
```

As you can see, variation of atomic charges converges after 7 cycles in this stage. Then the second stage starts:

```
**** Stage 2: RESP fitting under strong hyperbolic penalty
Atoms equivalence constraint imposed in this fitting stage:
Constraint 1: 12(H ) 13(H )
Constraint 2: 14(H ) 15(H )
```

Fitting objects: sp3 carbons, methyl carbons and hydrogens attached to them

Indices of these atoms:

4C 12H 13H 6C 14H 15H

Convergence criterion: 0.00000100

Hyperbolic restraint strength (a): 0.001000 Tightness (b): 0.100000

Iter: 1 Maximum charge variation: 1.0237455608

Iter: 2 Maximum charge variation: 0.0068736797

Iter: 3 Maximum charge variation: 0.0000294321

Iter: 4 Maximum charge variation: 0.0000001351

Successfully converged!

As indicated in the output, in the second fitting stage, the two hydrogens at each of the two $-\text{CH}_2-$ groups are required to be equivalent during the fitting. In addition, charges of only six atoms are fitted in the stage 2, they are carbons and hydrogens in the two $-\text{CH}_2-$ groups, while charges of other atoms keep unchanged at the values yielded in fitting stage 1.

The resulting RESP charges are

Center	Charge
1(O)	-0.540662
2(O)	-0.536015
... [ignored]	
12(H)	0.072719
13(H)	0.072719
14(H)	-0.064083
15(H)	-0.064083
... [ignored]	
Sum of charges: 0.000000	
RMSE: 0.002097 RRMSE: 0.110457	

If you examine the charges carefully, you will find all charges are chemically meaningful. The RMSE and RRMSE are not large, implying that quality of ESP fitting is nice. One can see that equivalence constraints indeed work, the H12 and H13 share the same charge 0.0727, while both H14 and H15 have charge of -0.064.

As mentioned in Section 3.9.16, during calculation of ESP fitting charges in the RESP module, it is possible to make Multiwfn directly load fitting points and ESP values from Gaussian output file of pop=MK or pop=CHELPG task. As an illustration, the Gaussian input file of dopamine for this purpose is provided as *examples\RESP\dopamine-single\dopamine_pop_MK.gjf*, use Gaussian to run it, then boots up Multiwfn and input

dopmaine.fch // In present situation this file in fact is only used to provide geometry information so that Multiwfn can determine atomic connectivity, therefore you can also use other formats such as .xyz, .pdb and .wfn instead

```

7 // Population analysis
18 // RESP module
8 // Let Multiwfn directly load fitting points information from Gaussian output file
1 // Calculate standard RESP charges using two-stage procedure
dopamine_pop_MK.out // The Gaussian output file with IOp(6/33=2,6/42=6) pop=MK

```

keywords

Then the calculation of atomic charges will be completed very quickly, because calculation of ESP values are avoided. Since the number and positions of fitting points generated by Multiwfn and those generated by Gaussian pop=MK task are different, current result is slightly different to that we obtained earlier.

Example 2: Taking multiple conformations into account during RESP charge calculation of dopamine

In this example we still calculate standard RESP charges for dopamine, but multiple conformations are explicitly considered in the ESP fitting procedure. It was found that there are four dominating conformations of dopamine in gas phase, the corresponding Gaussian input files of optimization task at B3LYP-D3(BJ)/6-311G** level have been provided in *examples\RESP\dopamine_4conf* folder, run them by Gaussian and then convert the resulting .chk files to .fch files.

My earlier Gibbs free energy calculations showed that at room temperature, according to Boltzmann distribution, the population of the four conformers are 8.48%, 2.66%, 48.45% and 40.42%, respectively. Therefore we should write a plain text file named *conf.txt* (other filenames are also acceptable) with below content, assuming that all the .fch files have been put into current folders.

```
dopamine1.fch 0.0848  
dopamine2.fch 0.0265  
dopamine3.fch 0.4844  
dopamine4.fch 0.4041
```

The first column is file path of each conformer, while the second column is corresponding weight. Evidently, the sum of all weights must be exactly equal or approximately equal to unity.

Boot up Multiwfn and input

dopamine1.fch // In present case, the file loaded at this stage is only used to provide geometry information that used to determine atomic connectivity, thus you can also use .fch of other conformers, the result will not be affected

```
7 // Population analysis  
18 // RESP module  
-I // Load conformation list file  
conf.txt // Input actual path of this file  
I // Calculate standard RESP charges using the two-stage procedure
```

The result is

```
Center      Charge  
1(O )    -0.512712  
2(O )    -0.494615  
...[ignored]  
21(H )    0.404082  
22(H )    0.388544  
Sum of charges:  0.000000  
Conformer:   1  RMSE:  0.002885  RRMSE:  0.176042  
Conformer:   2  RMSE:  0.002727  RRMSE:  0.163666
```

```
Conformer: 3 RMSE: 0.002234 RRMSE: 0.146771
Conformer: 4 RMSE: 0.002102 RRMSE: 0.134091
weighted RMSE: 0.002249 weighted RRMSE 0.144547
```

As can be seen, when considering multiple conformations, Multiwfn gives RMSE and RRMSE for each conformer as well as weighted RMSE and RRMSE. The data shows that current atomic charges have better ESP reproducibility for conformations 3 and 4 than conformations 1 and 2. The reason is not difficult to interpret, because the weights of conformations 3 and 4 in *conf.txt* are significantly higher than 1 and 2, therefore the fitted charges prone to faithfully represent charge distribution of conformers 3 and 4.

It is worth to note that if in the *conf.txt* you set weight of conformer 1 to unity while set that of other ones to zero, then the outputted statistical error will be

```
Conformer: 1 RMSE: 0.002047 RRMSE: 0.124925
Conformer: 2 RMSE: 0.002862 RRMSE: 0.171810
Conformer: 3 RMSE: 0.003391 RRMSE: 0.222734
Conformer: 4 RMSE: 0.004172 RRMSE: 0.266133
```

It can be seen that the atomic charges obtained at this time represent ESP of conformer 1 very well, because the RMSE and RRMSE are small, while ESP reproducibility of conformers 3 and 4, which have highest probability of occurrence, is no longer quite good. Therefore, current RESP charges is not ideal for molecular dynamics modeling of dopamine. This observation reflects the importance of considering multiple conformations for flexible molecules. Indeed, explicit consideration of multiple conformations in ESP fitting is somewhat troublesome and time-consuming, if you decide to obtain ESP fitting charges only by single structure, you should at least use the structure with the lowest free energy as much as possible.

When considering multiple conformations, the coordinates of fitting points as well as ESP values can also be directly loaded from Gaussian output files, here I present an example. For present molecule, the Gaussian input files of pop=MK task corresponding to the four conformers have been provided in "examples\RESP\dopamine_4conf\ESP" folder, run them by Gaussian to obtain .out files, then write a plain text file named e.g. confESP.txt with below content, with assumption that the four .out files have been placed to C:\ directory.

```
C:\dopamine1_ESP.out 0.0848
C:\dopamine2_ESP.out 0.0265
C:\dopamine3_ESP.out 0.4844
C:\dopamine4_ESP.out 0.4041
```

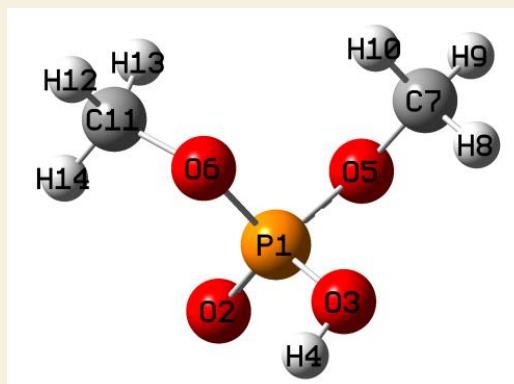
After that, load .fch (or other kinds of files) of any conformer into Multiwfn and enter interface of RESP module, then select

```
-1 // Load conformation list file
confESP.txt // Input actual path of this file
8 // Make Multiwfn directly load fitting point information from Gaussian output file
1 // Calculate standard RESP charges using the two-stage procedure
```

Then standard RESP charges will be immediately shown.

Example 3: Imposing equivalence constraint in ESP fitting of Dimethyl phosphate

Calculation of standard ESP charges has been illustrated in above two examples, next I exemplify how to calculate normal ESP fitting (*i.e.* one-stage fitting) with equivalence constraint. Dimethyl phosphate is taken as instance, its structure is shown below



The two methoxy groups of this system are chemically equivalent, and are easily rotated during molecular dynamic simulation. Therefore, the charges of O5 and O6 should be the same, the charges of C7 and C11 should be the same, and a total of six hydrogens on the two methyl groups (H8, H9, H10, H12, H13, H14) should also be the same. However, when only one structure is taken into account, it is clear that the such expectation in charge distribution cannot be achieved. Present example uses this system to demonstrate how to calculate the ESP fitting charges that meet the above equivalence requirements.

We first create a plain text file called e.g. *eqvcons.txt*, where each row contains indices of the atoms whose charges will be constrained to be the same. Therefore, the file content corresponding to current situation should be (in random order)

```
5,6
7,11
8-10,12-14
```

Run the Gaussian input file of optimization task at B3LYP-D3(BJ)/6-311G** level for present molecule (*examples\RESP\C2H7O4P\C2H7O4P.gjf*), then convert the resulting .chk file to .fch. Next, Boot up Multiwfn and input

```
C2H7O4P.fch
7 // Population analysis
18 // RESP module
5 // Modify the equivalence constraint used in one-stage ESP fitting (namely option 1). By default hydrogens in each CH2 and CH3 group are constrained to be equivalent
```

```
1 // Load equivalence constraint setting from external plain text file
eqvcons.txt // The file we just created
2 // Start one-stage ESP fitting calculation
```

The result is

center	charge
1(P)	1.120525
2(O)	-0.622980
3(O)	-0.588773

```

4(H ) 0.410987
5(O ) -0.403464
6(O ) -0.403464
7(C ) 0.035298
8(H ) 0.069429
9(H ) 0.069429
10(H ) 0.069429
11(C ) 0.035298
12(H ) 0.069429
13(H ) 0.069429
14(H ) 0.069429

Sum of charges: 0.000000
RMSE: 0.002541 RRMSE: 0.136047

```

Obviously, the result fully satisfies the equivalence constraint we have made, and the atomic charge values are also very reasonable and chemically significant. If we do not make the customized constraint but employ the default equivalency setting, the RRMSE will be 0.113024. Although the equivalence constraint we have made increases the RRMSE, indicating that the ESP reproducibility is lowered, since the RRMSE does not increase too much, so the constraint we currently employed is within reasonable range.

Example 4: Evaluation of atomic charges of aspartic acid residue with equivalence and charge constraints

This example is more complicated than the previous three ones, because multiple conformations, equivalence constraints and charge constraints are all involved. After carefully reading this section, I believe you will deeply feel that the RESP module of Multiwfn is amazingly flexible.

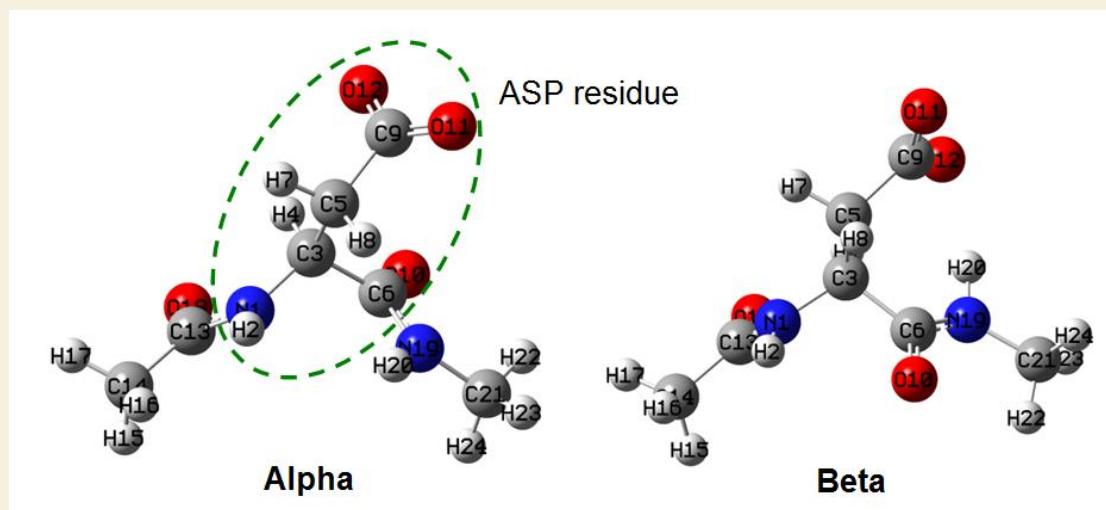
In this section we will calculate ESP fitting charges for aspartic acid (ASP) residue. The ASP is one of the most important amino acids in proteins. In general, in order to make electronic structure of a given residue in quantum chemistry calculation close to that in actual protein environment, the nitrogen terminal of the residue should be capped by acetyl group (ACE) while carbon terminal should be capped by N-methyl amide (NME). For present case, this treatment results in a model system ACE-ASP-NME.

The two most typical secondary structures of proteins are alpha helix and beta-sheet. From the point of view of the residues that make up them, the difference comes from the phi and psi dihedrals of the residue backbone. It has been suggested that residue conformations corresponding to both the secondary structures should be taken into account in the ESP fitting procedure. Also note that the net charge of the residue segment in the ACE-ASP-NME system must be an integer. Assume that the proton of the carboxyl group of the ASP side chain has dissociated, the net charge of the ASP residue should be constrained to be -1.0. In addition, given that the two oxygens of the carboxylate are chemically equivalent, it is preferable to apply an equivalence constraint to the two oxygens. The two hydrogens in the CH₂ group of the ASP side chain should also be constrained to be equivalent.

The Gaussian input files of optimization task for the ACE-ASP-NME models corresponding to alpha helix and beta-sheet have been provided as *alpha.gjf* and *beta.gjf* in "examples\RESP\ACE-

"ASP-NME" folder. As can be seen in the files, the keywords correspond to B3LYP-D3/6-311G** level with IEFPCM solvation model to represent water environment. In the optimization, the phi and psi dihedrals are fixed to their initial values (the dihedrals will vary remarkably during optimization if they are not frozen). In *alpha.gjf*, the phi and psi are -90° and -60°, respectively, corresponding to typical case of alpha helix. While in *beta.gjf*, the two dihedrals are set to -100° and 130°, reflecting typical situation of beta-sheet.

Run the two .gjf files by Gaussian, and convert resulting .chk files to .fch format. The two optimized structures are shown below. The region surrounded by green dashed ellipse is the ASP residue, the charges of these atoms are what we are interested in. The phi and psi dihedrals mentioned above correspond to 6-3-1-13 and 1-3-6-19, respectively.



We create a plain text file named for example *chgcons.txt*, in this file each line defines a charge constraint term. Since we require that the ASP residue has total charge of -1, we should write below content in this file

```
1-12 -1
```

Note that in the RESP module, there is no upper limit on the number of charge constraint terms.

Then we create a plain text file named for example *eqvcons.txt*, in this file each line defines a equivalence constraint term. As mentioned earlier, O11 and O12 should be equivalent, H7 and H8 should be equivalent, therefore for present case the content should be

```
11,12
```

```
7,8
```

Although the hydrogens in the methyl groups at the two ends of the model system are chemically equivalent, since they are not of our interest, the equivalence constraint setting is ignored.

Next, we write a file named for example *conflist.txt*, which contains list of .fch files of all conformers. In present circumstance we hope that the resulting atomic charges can equally well represent the actual charge distribution of ASP residue in both alpha helix and beta-sheet secondary structures, therefore weight of both the conformers should be 0.5. Assuming that .fch files have been placed in D:\ folder, the file content should be

```
D:\alpha.fch 0.5
```

```
D:\beta.fch 0.5
```

Finally, boot up Multiwfn, load either *alpha.fch* or *beta.fch*, then enter RESP module and input below commands

```
5 // Modify the equivalence constraint used in one-stage ESP fitting
1 // Load equivalence constraint setting from external plain text file
eqvcons.txt // The equivalence constraint file we created
6 // Set charge constraint in one-stage fitting
1 // Load charge constraint setting from external plain text file
chgcons.txt // The charge constraint file we created
-1 // Load list of conformer and weights from external file
conflist.txt // The conformation list file we created
2 // Start one-stage ESP fitting calculation
```

The result is

```
1(N ) -0.568030
2(H ) 0.298690
3(C ) 0.232066
4(H ) 0.003952
5(C ) -0.187247
6(C ) 0.580605
7(H ) 0.030942
8(H ) 0.030942
9(C ) 0.773254
10(O ) -0.602338
11(O ) -0.796419
12(O ) -0.796419
...[ignored]
Sum of charges: -1.000000
Conformer: 1 RMSE: 0.002175 RRMSE: 0.017514
Conformer: 2 RMSE: 0.002087 RRMSE: 0.017379
Weighted RMSE: 0.002131 Weighted RRMSE 0.017446
```

The above calculation result is very reasonable, and it can be seen that both the charge constraint and equivalence constraint work perfectly. Moreover, since the weights of the two conformations are set to be the same, the RMSE or RRMSE corresponding to the two conformers have comparable magnitude. Given that the RRMSE is very small, the current fitted charges should be able to describe the state of ASP residue in various proteins well.

Example 5: Using two times of one-stage fitting to equivalently realize standard RESP two-stage fitting

In example 1 of this Section, I have illustrated how to derive RESP charges using standard RESP two-stage fitting procedure. Thanks to the flexibility of RESP module of Multiwfn, this “composite procedure” can also be manually realized via two separated one-stage fittings, as illustrated in this section. After reading this section, I believe you will better understand how to customize the RESP calculation procedure. Below we will use a very simple molecule methanol as example, whose .fch file can be found in <http://sobereva.com/multiwfn/extrafiles/RESP.zip>.

Boot up Multiwfn and input
methanol.fch

7 // Population analysis
18 // RESP charge calculation
5 // Set equivalence constraint
0 // Remove default equivalence constraint
2 // Using one-stage fitting to derive charges

The result is

center	charge
1(C)	0.238915
2(H)	0.045904
3(H)	-0.018089
4(H)	-0.018089
5(O)	-0.664522
6(H)	0.415880

They are identical to the charges obtained at the first stage of standard RESP two-stage fitting.

According to definition of standard RESP charge calculation procedure, the charge of the atoms in hydroxyl group of methanol should keep fixed during the second fitting stage, therefore we create a file *chgcons.txt* with below content.

5 -0.664522
6 0.415880

Then input below commands in Multiwfn interface

n // Do not export .chg file
4 // Set hyperbolic penalty parameters
2 // Set restraint strength (a)
0.001 // This value is the one used in the second stage of standard RESP fitting procedure
0 // Return to the upper menu
5 // Set equivalence constraint
2 // Constraint hydrogens in CH₂ and CH₃ groups to be equivalent, as required by the second stage of standard RESP fitting
6 // Set charge constraint
1 // Load charge constraint setting file
chgcons.txt

2 // Calculate charges by one-stage fitting

The final result is

1(C)	0.235334
2(H)	0.004436
3(H)	0.004436
4(H)	0.004436
5(O)	-0.664522
6(H)	0.415880

Which are completely identical to the charges derived by standard two-stage RESP fitting.

Hint: When the molecule is large, manually editting the *chgcons.txt* is often cumbersome. In fact, you can create an empty file named *chgcons_stage2.txt* in current folder and carry out standard

two-stage RESP fitting, then before performing the second stage fitting, Multiwfn will automatically export the indices and charges of the atoms whose charges will be kept fixed in the second stage to this file, so that you will not need to manually write the *chgcons.txt* file.

4.7.8 Examine electrostatic potential reproducibility of atomic charges

Electrostatic potential (ESP) reproducibility is a crucial property of atomic charges, only atomic charges having good ESP reproducibility could be employed to reveal intramolecular and intermolecular electrostatic interactions. It is possible to examine ESP reproducibility of given atomic charges using the MK and CHELPG charge calculation modules, which have been introduced in Sections 3.9.10 and 3.9.11, respectively. Here we compare the ability of Hirshfeld and ADCH charges for reproducing ESP values at Merz-Kollmann ESP fitting points (which are distributed around molecular van der Waals surface) for CH₃CONH₂. We first calculate Hirshfeld charges as usual using *examples\CH3CONH2.fch* (see Section 4.7.1), then select "y" to export the atomic charges to *CH3CONH2.chg*. Then we enter the MK charge calculation module (subfunction 13 of main function 7) and input

```
-3 // Using atomic charges from a .chg file
CH3CONH2.chg // Atomic charges (i.e. Hirshfeld charges) will be directly loaded from this
file
1 // Start calculation. In current case MK charges will not be yielded
The data shown on the screen is
  center      charge
  1(C )   -0.090370
  2(H )    0.037254
  3(H )    0.043058
  4(H )    0.048339
  5(C )    0.170596
  6(O )   -0.308866
  7(N )   -0.159120
  8(H )    0.131920
  9(H )    0.127108
Sum of charges: -0.000081
RMSE: 0.006214 RRMSE: 0.310394
```

These charges are just the Hirshfeld charges loaded from *CH3CONH2.chg*, the RMSE and RRMSE measure the ESP reproducibility of the Hirshfeld charges. If you compute MK charges as usual, you will find the RRMSE will be about 0.05, since as shown above the RRMSE of Hirshfeld charges is as high as 0.31, it is evident that the ESP reproducibility of Hirshfeld charges is much worse than MK charges. If you redo the analysis based on the .chg file containing ADCH charges, you will find the RRMSE is 0.21. Clearly, ADCH charges have evidently lower error in reproducing ESP compared to Hirshfeld charges.

It is also possible to measure ESP reproducibility on the fitting points corresponding to specific atom or fragment. By default, the MK points are generated around all atoms in turn and then the points lying inside the intermost layer are pruned. If only specific atoms are taken into account, the

constructed MK fitting points will only correspond to those atoms. Let us compare ESP reproducibility of Hirshfeld and ADCH charges around the amino group, only two MK layers with scale factor of 1.4 and 1.6 will be considered (no special reason, just give an example). Enter the MK module and input

```
-3 // Using atomic charges from a .chg file
CH3CONH2.chg // Assume that this file contains Hirshfeld charges
3 // Set number and scale factors of layers of MK fitting points
1.4 // Set scale factor of layer 1
1.6 // Set scale factor of layer 2
q // Setting has finished, now quit
4 // Choose the atoms involved in the construction of fitting points
7-9 // Atomic indices of amino group
I // Start calculation
```

You will find below information from the output

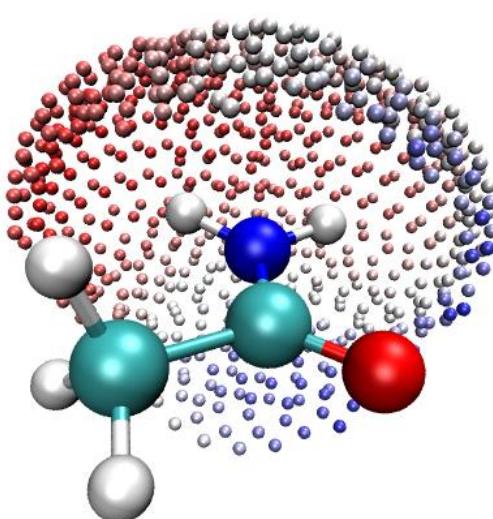
```
RMSE: 0.008745 RRMSE: 0.374584
```

If we repeat the calculation based on .chg file containing ADCH charge, the output will be

```
RMSE: 0.003817 RRMSE: 0.163478
```

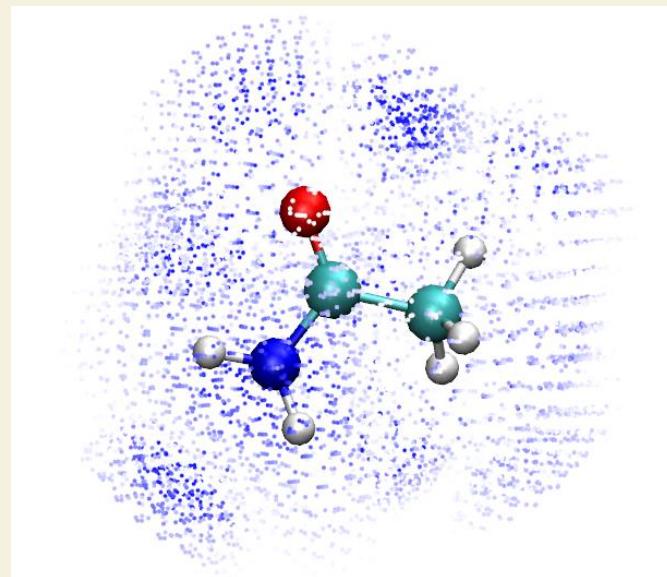
Since the RRMSE of ADCH charge (0.163) is by far less than that of Hirshfeld charge (0.374), the ADCH charges have much better ESP reproducibility around the amino group.

If you want to visualize the fitting points corresponding to the amino group, you can select "6 Toggle if exporting fitting points with ESP after the task" once to change the status to "Yes" and then use option 1 to start calculation. Once calculation is finished, choose 2 to export the fitting points to *ESPfitpt.pqr* in current folder. This file can be directly loaded into the famous visualization tool VMD. If you set the drawing method to "VDW" and change the "Sphere Scale" to 0.8, set "Coloring Method" to "Charge", then set the color transition mode to "BWR" (Graphics - "Colors" - "Color Scale"), you will see below graph (molecular structure file is also loaded).



Clearly, the fitting points well correspond to the amino group. The more red (blue), the more positive (negative) the ESP on the points.

Finally, I would like to mention that the reproduction error of ESP can also be visualized by combinely using Multiwfn and VMD. Here we check this for MK charges. Load the *examples\CH3CONH2.sch* into Multiwfn, enter MK module, choose option 6 once, and then choose option 1 to start calculation. Once calculation is completed, choose 3 to export ESP fitting points with ESP reproduction error to *ESPerro.pqr* in current folder. In this file, the "Charge" column corresponds to absolute value of difference (in kcal/mol) between the exact ESP and the ESP evaluated based on current atomic charges (namely MK charges). If you render this file by VMD, you will see below graph. The color scale has been set to -1.5 to 1.5 (can be set in "Graphics" - "Representation" - "Trajectory" page), the default color transition "Red-White-Blue" is used, perspective has been set to orthographic ("Display" - "Orthographic").



In this graph, more blue region corresponds to higher ESP reproduction error, while ESP at white points can be well reproduced by the MK charges (*i.e.* the absolute error is close to zero). You can also use this method to visualize ESP reproducibility of other atomic charges (need to use .chg file, as illustrated earlier).

4.8 Molecular orbital composition analysis

In this section, I will show how to use Mulliken, NAO and Hirshfeld/Becke methods to analyze molecular orbital composition. These methods are also applicable to other types of orbitals, *e.g.* natural orbitals, natural transition orbitals and localized MO. Details about orbital composition analysis can be found in Section 3.10. The pros and cons of different methods are very detailedly discussed in my paper *Acta Chim. Sinica*, **69**, 2393 (2011, http://sioc-journal.cn/Jwk_hxxb/CN/abstract/abstract340458.shtml), citation is highly welcomed. Also you can consult my blog article "On the calculation methods of orbital composition" (in Chinese, <http://sobereva.com/131>).

Simply speaking, if your aim is merely obtaining atom compositions in orbitals, Hirshfeld/Becke method may be the most robust and convenient way, see Section 4.8.3; if you also would like to obtain atomic orbital composition, then the NAO method exemplified in Section 4.8.2 may be the best choice. The Mulliken method illustrated in Section 4.8.1 also generally works well

but diffuse functions must not be used.

4.8.1 Analyze acetamide by Mulliken method

In this example we employ Mulliken method to first analyze the composition of the 6th molecular orbital of acetamide, and then analyze which orbitals have main contribution to the bonding between formamide part and methyl group. Beware that Mulliken method is incompatible with diffuse functions, if they are involved, you should either choose other orbital composition methods (*e.g.* NAO, Hirshfeld...) or remove them from your basis set.

Boot up Multiwfn and input following commands

examples\CH3CONH2.fch // You have to use .fch/.molden/.gms file as input for this type of analysis

```
8 // Orbital composition analysis
1 // 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 "comptres"
parameter in settings.ini.
```

```
orbital: 6 Energy(a.u.): -0.90528995 occ: 2.00000000 Type: Alpha&Beta
```

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%
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: > 0.500000%
```

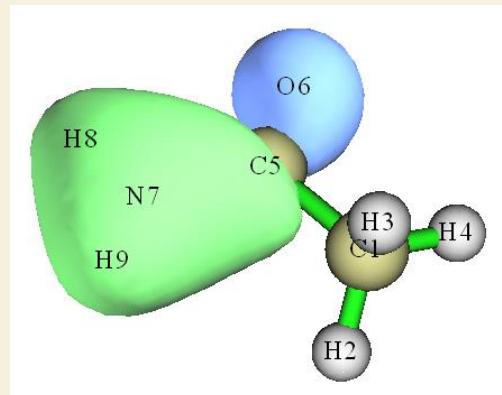
Shell	14	Type: S	in atom	5(C) :	1.12409%
Shell	15	Type: P	in atom	5(C) :	10.95268%
Shell	17	Type: P	in atom	5(C) :	-0.97156%
Shell	20	Type: S	in atom	6(O) :	6.15544%
Shell	22	Type: S	in atom	6(O) :	5.00803%
Shell	26	Type: S	in atom	7(N) :	30.53098%
Shell	28	Type: S	in atom	7(N) :	34.29046%
Shell	31	Type: S	in atom	8(H) :	4.30946%

```
Shell    33 Type: S    in atom   9(H ) :      5.13880%
```

Composition of each atom:

```
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 consists of two S-shells (30.5% and 34.3%). P-shells of neighbour carbon and S-shells of oxygen have slight contribution too (both are about 12%). We can check if the result is reasonable by viewing isosurface (iso value 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 us find which molecular orbitals have main contribution to the bonding between formamide part and methyl group.

```
examples\CH3CONH2.fch
8 // Orbital composition analysis
-I // Define fragment 1
a 1-4 // Add all basis functions in atom 1, 2, 3, 4 (methyl group) into fragment1
q // Save fragment and return to upper menu
-2 // Define fragment 2
a 5-9 // Add all basis functions in atom 5, 6, 7, 8, 9 (formamide moiety) into fragment 2
```

q

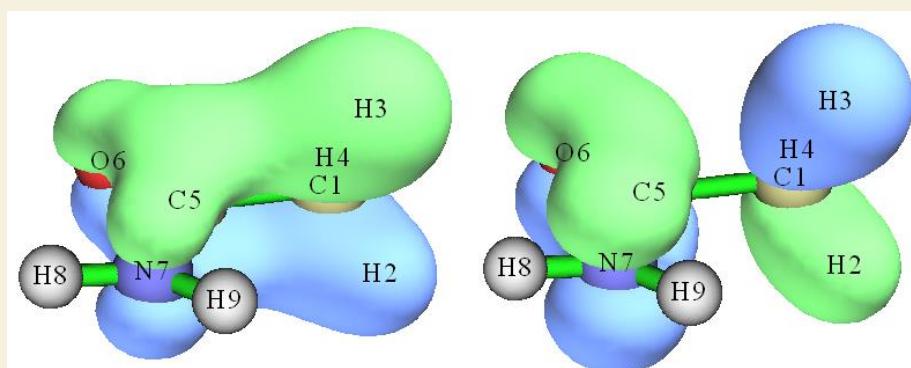
4 // Print composition of fragment 1 and the cross term between fragment 1 and 2 in all orbitals by Mulliken analysis. If you only defined fragment 1, then only composition of fragment 1 will be printed

Amount of the printed information 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 Mulliken bond order between them contributed by orbital *i*. From above information we can see MO 7 and 11 are beneficial to 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.



As you have seen, using Mulliken method to analyze orbital composition is very convenient. However, the result of Mulliken method is sensitive to basis set, and is not as robust as NAO method and Hirshfeld method illustrated below. Especially, do not use Mulliken method when diffuse functions are involved in your calculation, otherwise the result will be meaningless!

4.8.2 Analyze water by natural atomic orbital method

In this example we analyze molecular orbital composition of water by the natural atomic orbital (NAO) method discussed in Section 3.10.4. NAO method has much better basis set stability (*i.e.* insensitive to the choice of basis set) and stronger theoretical basis than Mulliken or Mulliken-like methods (such as SCPA and Stout-Politzer).

Note: NAO is never the only way of obtaining composition of atomic orbitals in MOs, you can also use Mulliken or similar methods (e.g. SCPA) to do that. The correspondence between basis function and atomic orbital can be identified according to basis set definition or by mean of population analysis, see Section 4.7.6.

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. Notice that the Gaussian task should be single point task, do not perform geometry optimization together!

```
# 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 Gaussian output file is named as H2O_NAOMO.out (can be found in "example" folder), we start Multiwfn and input:

```
examples/H2O_NAOMO.out // Note that DO NOT use .fch as input file!
8 // Enter orbital composition analysis module
7 // 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%

4 Tutorials and Examples

```

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.

Note that the sum of non-Rydberg compositions, as shown above, is not 100% rather than 99.777%. To make the physical meaning more clear, I personally recommend to manually perform renormalization process for the result, for example, the composition of 2pz should be $84.089\%/0.99777=84.277\%$. Since before and after the renormalization the difference is only 0.188%, the renormalization is not necessary for current case. Only when the non-Rydberg composition is unnegligible (e.g. larger than 2%), the renormalization is indispensable.

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	O	1	dx2y2 Ryd(3d)	0.00073	2.47969
... [ignored]					
15	O	1	dz2 Ryd(3d)	0.00254	2.02361
16	H	2	s Val(1s)	0.52321	0.33308
17	H	2	s Ryd(2s)	0.00086	0.70497
18	H	3	s Val(1s)	0.52321	0.33308
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 *I*,*8*, the contribution from the four NAOs to MO 1~8 will be shown as below

orb.#	core	valence	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.

The physical meaning of Rydberg NAOs is difficult to be interpreted, and these NAOs do not directly reflect atomic orbital characteristics. It is questionable to say the two hydrogens contributed either 32.43% or 62.00% to MO 8. Although seemingly one can employ renormalization process to "annihilate" the Rydberg composition, however when Rydberg composition is too large, for example larger than 10%, this treatment will make the result lack of reliability. So it is not generally recommended to use NAO method to analyze atomic contributions to virtual MOs; for this case, the Hirshfeld or Becke method introduced in Section 3.10.5 and exemplified in next section may be the best choice.

4.8.3 Analyze acetamide by Hirshfeld and Becke method

In this section, we will first use Hirshfeld method and then Becke method to analyze the MO composition of acetamide and compare the result with the one obtained by Mulliken method in Section 4.8.1. Note that Hirshfeld and Becke methods are only capable of analyzing composition of atom or fragment in orbitals, while the composition of atomic orbitals are impossible to be obtained by these approaches.

Boot up Multiwfn and input

examples\CH3CONH2.fch // You can also use such as .wfn and .wfx file as input. But .wfn and .wfx files do not contain virtual orbital information!

8 // Orbital composition analysis

8 // Use Hirshfeld partition

Hirshfeld analysis requires electron density of atoms in their free-states, you need to choose a method to calculate atomic densities. Selecting 1 to use built-in atomic densities is very convenient, see Appendix 3 for detail; alternatively, you can select 2 to evaluate atomic densities based on atomic .wfn files, see Section 3.7.3 for detail. Here we choose option 1.

Then Multiwfn initializes the data, for large system you may need to wait for a while. Assume that you want to analyze MO 6, then simply input 6, the result will be printed on screen, as shown below. (Because the integrals are evaluated numerically, the sum of all terms will be slightly deviated to 100%, so Multiwfn automatically normalizes the result.)

Atom 1(C) : 1.553970%

```

Atom 2(H ) : 0.248499%
Atom 3(H ) : 0.154527%
Atom 4(H ) : 0.037207%
Atom 5(C ) : 14.924962%
Atom 6(O ) : 12.109412%
Atom 7(N ) : 56.338603%
Atom 8(H ) : 6.687609%
Atom 9(H ) : 7.945211%

```

The composition of C5, O6 and N7 are 14.92%, 12.11% and 56.34%, respectively. This result is close to the one obtained by Mulliken method (Section 4.8.1), namely 11.81%, 11.63% and 65.50%, respectively. In fact, for occupied MOs, if diffuse basis functions are not employed, in general Mulliken, NAO and Hirshfeld methods give similar results.

Now let us check the composition of 7N in MO from 14 to 19. We input

-2

7 // Atomic index

14,19 // Orbital range

You will see:

Orb#	Type	Ene(a.u.)	occ	Composition	Population
14	Alpha&Beta	-0.3674	2.000	16.075%	0.321499
15	Alpha&Beta	-0.2661	2.000	48.508%	0.970170
16	Alpha&Beta	-0.2438	2.000	6.794%	0.135870
17	Alpha&Beta	0.0410	0.000	12.390%	0.000000
18	Alpha&Beta	0.0762	0.000	21.757%	0.000000
19	Alpha&Beta	0.1252	0.000	14.516%	0.000000

Population of this atom in these orbitals: 1.427539

where 1.427539 (namely 0.321499+0.970170+0.135870) is the total population number of N7 in MO 14~19.

PS: If the orbital range you specified is 1~16, namely all occupied MO, then the outputted value 7.1586 will be the atomic population number of N7, and its Hirshfeld atomic charge is therefore $7.0 - 7.1586 = -0.1586$.

The steps of analyzing orbital composition by Becke method are completely identical to that of Hirshfeld method. Here we calculate the composition of MO 6. Boot up Multiwfn and input

examples\CH3CONH2.fch

8 // Orbital composition analysis

9 // Use Becke partition

6 // The 6th orbital

The result is

```

Atom 1(C ) : 1.229610%
Atom 2(H ) : 0.085144%
Atom 3(H ) : 0.048572%
Atom 4(H ) : 0.002414%
Atom 5(C ) : 14.901723%
Atom 6(O ) : 12.069731%
Atom 7(N ) : 60.741513%
Atom 8(H ) : 4.929374%

```

```
Atom    9(H ) : 5.991918%
```

As you can see, the result is highly close to that produced by Hirshfeld method.

In Multiwfn it is also possible to use Hirshfeld-I partition to calculate orbital composition, however this is not commonly employed, because generating Hirshfeld-I atomic space requires additional computational cost, while the result is not greatly improved (the orbital composition computed by Hirshfeld method is already reliable and meaningful enough).

4.8.4 Calculate oxidation state by LOBA method

Please read Section 3.10.7 first to comprehend basic idea of LOBA method. This is a simple and useful method to evaluate oxidation state (OS). In this section I will use two examples to illustrate this function. Notice that the implementation of LOBA in Multiwfn is incompatible with diffuse functions, if basis set containing diffuse functions is used, the result may be meaningful.

(1) Fe(CN)₆³⁻

First, we use Gaussian to perform regular calculation of this system, the input file is *example\Fe(CN)6_3-.gjf*, please run it by yourself to get *Fe(CN)6_3-.fch* file. LOBA analysis needs localized MO (LMO), thus we use Multiwfn to carry out orbital localization. Boot up Multiwfn and input following commands:

```
Fe(CN)6_3-.fch
19 // Orbital localization
1 // Only localize occupied orbitals, this is enough for LOBA analysis
Now the orbitals recorded in memory has been updated to localized orbitals
0 // Return to main interface
8 // Orbital composition analysis
100 // LOBA analysis
50 // Percentage threshold for performing LOBA
oxidation state of atom 1(Fe) : 3
oxidation state of atom 2(C ) : 2
oxidation state of atom 3(C ) : 2
oxidation state of atom 4(C ) : 2
oxidation state of atom 5(C ) : 2
oxidation state of atom 6(C ) : 2
oxidation state of atom 7(C ) : 2
oxidation state of atom 8(N ) : -3
oxidation state of atom 9(N ) : -3
oxidation state of atom 10(N ) : -3
oxidation state of atom 11(N ) : -3
oxidation state of atom 12(N ) : -3
oxidation state of atom 13(N ) : -3
The sum of oxidation states: -3
```

The result looks reasonable and in good agreement with chemical intuition. The sum of all oxidation states just corresponds to the total net charge of -3. However, the LOBA is not free of ambiguity, the choice of the threshold is highly arbitrary. If we input 60 instead of 50, we will see oxidation state of carbon and oxygen become 4 and -1, respectively, and the sum of oxidation states

become 21. Fortunately, the OS of transition metal is never so sensitive to the choice of threshold, the iron always keeps +3 oxidation state in present case as long as the threshold is not set to very small or very large. (In my viewpoint, the most appropriate threshold for evaluating OS of transition metal is 50~60%)

(2) Ferrocene

For this system, we will not only check OS of iron, but also check OS of C₅H₅ fragment. The corresponding regular Gaussian input file is *example\Ferrocene.gjf*, run it by yourself to obtain corresponding .fch file, then load it into Multiwfn and perform orbital localization first as shown above, after that enter LOBA analysis interface and input below commands:

```
-I // Define fragment
1-5,7-11 // Index of the atoms constituting the C5H5 fragment
50 // Percentage threshold for performing LOBA
```

The result is

```
oxidation state of atom 1(C ) : 2
oxidation state of atom 2(C ) : 2
oxidation state of atom 3(C ) : 2
oxidation state of atom 4(H ) : 1
oxidation state of atom 5(H ) : 1
oxidation state of atom 6(Fe) : 2
oxidation state of atom 7(C ) : 2
oxidation state of atom 8(C ) : 2
...[ignored]
```

The sum of oxidation state: 32

Oxidation state of the fragment: -1

In this system the OS of Fe is +2, which is again reasonable. From the output it is seen that the OS of individual carbons are not useful, however, the OS of the whole C₅H₅ fragment is a meaningful value -1.

Above two examples exhibit usefulness of LOBA method. We should always focus on OS of transition metal or OS of whole ligands, while the OS of individual atom in ligand often does not make sense.

4.9 Bond order analysis

4.9.1 Mayer bond order and fuzzy bond order analysis on acetamide

In this instance we will calculate Mayer bond order and fuzzy bond order for acetamide. Related theories have been introduced in Section 3.11.1 and 3.11.6, respectively.

We first calculate Mayer bond order. Note that calculating Mayer bond order requires basis function information, thus currently .fch/.molden/.gms file must be used as input file.

Boot up Multiwfn and input:

```
examples\CH3CONH2.fch
9 // Bond order analysis
I // Calculate Mayer bond order
```

Immediately you get below output:

```
The total bond order > 0.100000
# 1:    1(C )   2(H )   0.93802674
# 2:    1(C )   3(H )   0.93473972
# 3:    1(C )   4(H )   0.94494566
# 4:    1(C )   5(C )   0.96585484
# 5:    5(C )   6(O )   1.90392771
# 6:    5(C )   7(N )   1.11849509
# 7:    7(N )   8(H )   0.83250273
# 8:    7(N )   9(H )   0.83869874
```

Total valences and free valences defined by Mayer:

```
Atom 1(C ) : 3.77555991 0.00000000
Atom 2(H ) : 0.93147308 0.00000000
Atom 3(H ) : 0.92778456 0.00000000
Atom 4(H ) : 0.93657474 0.00000000
Atom 5(C ) : 3.97788022 0.00000000
Atom 6(O ) : 2.05925868 0.00000000
Atom 7(N ) : 2.85041375 0.00000000
Atom 8(H ) : 0.86522064 0.00000000
Atom 9(H ) : 0.85875080 0.00000000
```

By default, only the bond order terms larger than specific criteria will be outputted, the criteria can be adjusted in "bndordthres" in settings.ini. Mayer bond order often coincide with empirical bond order well. In this example, bond order between C5 and O6 is 1.9, which is very close to ideal value 2.0 (double bonds).

Total valence of an atom is the sum of Mayer bond orders that it formed. Free valence of a atom measures its remained capacity to form new bonds by sharing electron pairs, for closed-shell this quantity is always zero.

Then if you choose "y", entire bond order matrix will be outputted to bndmat.txt in current folder.

Next, we want to try to find which orbitals have main contributions to Mayer bond order between C5 and O6, calculating the so-called "Orbital occupancy-perturbed Mayer bond order" is useful for realizing this goal. Hence, we select option 6 in bond order analysis module, and then input 5,6. Below information will be outputted:

```
Mayer bond order before orbital occupancy-perturbation: 1.903928
```

orbital	occ	Energy	Bond order	Variance
1	2.00000	-19.10356	1.906089	0.002162
2	2.00000	-14.34920	1.903934	0.000006
3	2.00000	-10.28192	1.905514	0.001586
4	2.00000	-10.18447	1.903939	0.000011
5	2.00000	-1.03758	1.606912	-0.297016

6	2.00000	-0.90529	1.825661	-0.078267
7	2.00000	-0.73868	1.872011	-0.031917
8	2.00000	-0.58838	1.896333	-0.007594
9	2.00000	-0.54103	1.868127	-0.035801
10	2.00000	-0.46630	1.799351	-0.104576
11	2.00000	-0.44722	1.512533	-0.391394
12	2.00000	-0.40158	1.830827	-0.073101
13	2.00000	-0.39610	1.659075	-0.244853
14	2.00000	-0.36742	1.393122	-0.510805
15	2.00000	-0.26611	1.743383	-0.160545
16	2.00000	-0.24383	1.796157	-0.107770

Summing up occupancy perturbation from all orbitals: -2.03987

From the output we can know that, for example, if the two electrons are removed from orbital 15, then Mayer bond order between C5 and O6 will decreased from 1.903928 to 1.743383, we can also say that the contribution from orbital 15 is 0.160545. The sum of contributions from all occupied MOs is 2.03987, the reason that this value is not equal to 1.903928 is that Mayer bond order is not a linear function of density matrix, we do not need to concern this.

Orbital 14 has the largest negative value of orbital occupancy-perturbed Mayer bond order, therefore this orbital must be greatly beneficial to the bonding. This conclusion can be further testified by visual inspection of the orbital isosurface, see the graph given at the end of Section 4.8.1. As expected, this orbital shows strong character of π -bonding between C5 and O6.

Now we calculate fuzzy bond order. Unlike Mayer bond order, fuzzy bond order does not rely on basis function, therefore you can also use such as .wfn/.wfx as input file. Calculating fuzzy bond order is more time consuming than Mayer bond order, its advantage over Mayer bond order is that the basis set sensitivity is greatly reduced, and using diffuse basis functions will never deteriorate result.

In the bond order analysis menu, we select "7 Fuzzy bond order analysis", the result will be printed, as shown below

The total bond order >= 0.050000			
# 1:	1(C)	2(H)	0.89666213
# 2:	1(C)	3(H)	0.89075146
# 3:	1(C)	4(H)	0.88888051
# 4:	1(C)	5(C)	1.08050669
# 5:	1(C)	6(O)	0.13600686
# 6:	1(C)	7(N)	0.11096284
# 7:	3(H)	5(C)	0.05357850
# 8:	5(C)	6(O)	2.00411901
# 9:	5(C)	7(N)	1.40339002
# 10:	6(O)	7(N)	0.24523041
# 11:	7(N)	8(H)	0.87233783
# 12:	7(N)	9(H)	0.88243512

Comparing the result with that of Mayer bond order, you will find the results of both types of bond orders are very similar, in fact this is the common case. However, for highly polar bonds their

results may deviate with each other relatively evidently.

4.9.2 Three-center bond order analysis on Li₆ cluster

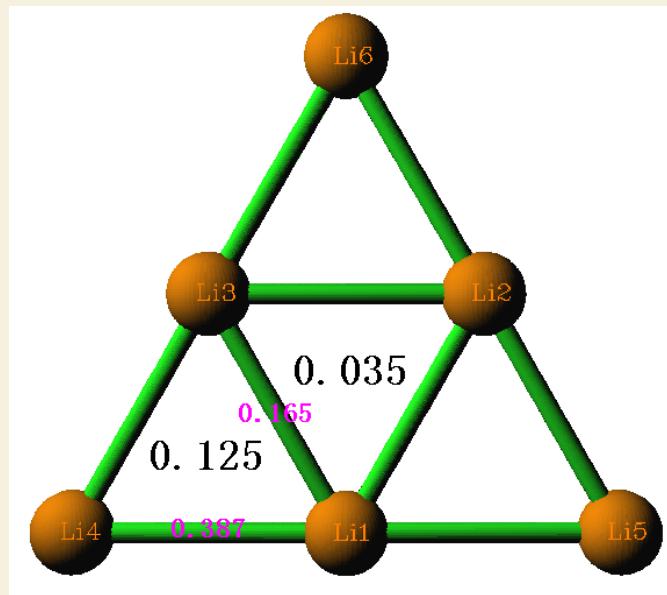
The electron structure character of complex system, such as cluster is hard to predict by simple empirical rules, we have to resort to wavefunction analysis methods. In this section we use three-center bond order to study which type of three-membered ring in Li₆ cluster is the most stable. We will calculate the three-member ring consisting of Li1, Li4 and Li3 first.

Notice that multi-center bond order analysis requires basis function information, therefore in current Multiwfn version, you have to use .fch/.molden/.gms file as input file.

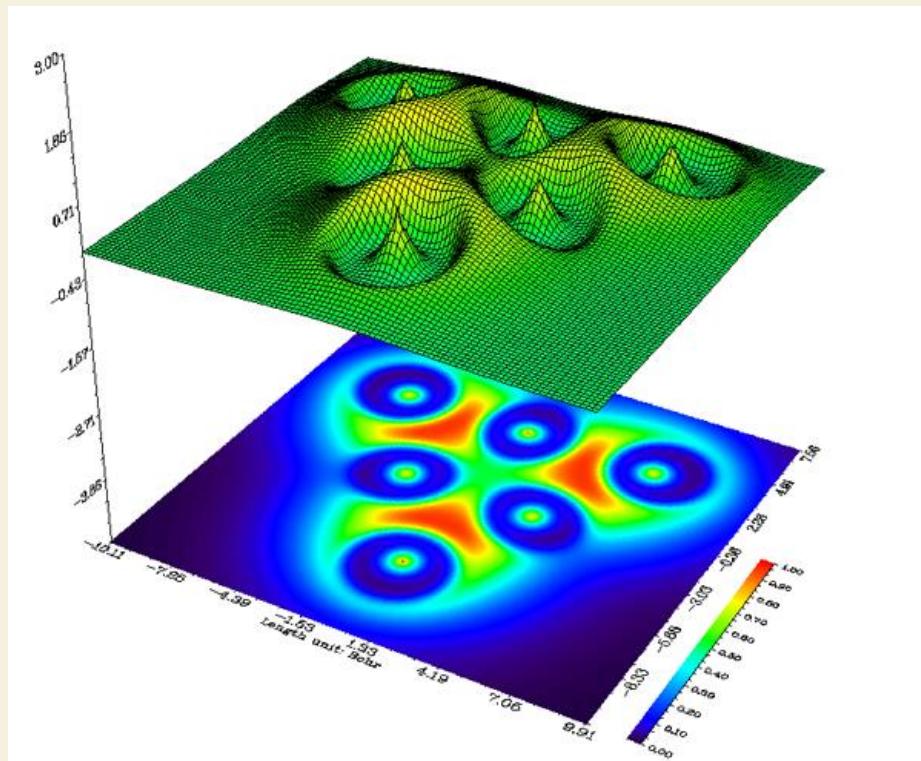
Boot up Multiwfn and input following commands

```
examples\Li6.fch
9 // Bond order analysis
2 // Multi-center bond order analysis
1,4,3 // Input the indices of the atoms constituted the three-member ring
```

The 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 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 LOL 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 will draw the same conclusion.

The *Li6.fch* used in this example was produced at B3LYP/6-31G* level. Sometimes diffuse functions must be employed for properly describing a system (in particular, anions), in this case you should evaluate the multi-center bond order based on natural atomic orbitals (NAO) rather than based on original basis functions as illustrated above, otherwise the result may be even completely useless, see Section 3.11.2 for detail.

4.9.3 Calculate Laplacian bond order (LBO)

The Laplacian bond order (LBO) was proposed by me in *J. Phys. Chem. A*, **117**, 3100, see Section 3.11.7 for detail. LBO is very suitable for organic system and has close correlation with bonding strength. Let us calculate LBO for C-C bond of ethane, ethene and acetylene.

Boot up Multiwfn and input following commands

```
examples\ethane.wfn // Optimized and produced at B3LYP/6-31G**
9 // Bond order analysis
8 // Laplacian bond order
```

You will see the result:

```
The bond order >= 0.050000
# 1: 1(C ) 2(H ): 0.887111
# 2: 1(C ) 3(H ): 0.889492
# 3: 1(C ) 4(H ): 0.889492
# 4: 1(C ) 5(C ): 1.059879
```

```
# 5: 5(C ) 6(H ): 0.887111
# 6: 5(C ) 7(H ): 0.889492
# 7: 5(C ) 8(H ): 0.889492
```

As you can see, LBO is very close to formal bond order (1.0) for C-C and C-H. LBO only reflects covalent bonding character, due to C-H is a weakly polar bond, the value is slightly smaller than 1.0.

Then use *examples\ethene.wfn* to calculate LBO for ethene

```
# 1: 1(C ) 2(H ): 0.919443
# 2: 1(C ) 3(H ): 0.919443
# 3: 1(C ) 4(C ): 2.022583
# 4: 4(C ) 5(H ): 0.919443
# 5: 4(C ) 6(H ): 0.919443
```

Then calculate LBO for acetylene by using *examples\C2H2.wfn*

```
# 1: 1(C ) 2(H ): 0.958393
# 2: 1(C ) 3(C ): 2.767449
# 3: 3(C ) 4(H ): 0.958393
```

The LBO of the C-C bonds in the three systems are 1.060, 2.022 and 2.767, the ratio is 1:1.907:2.61. It is known that the ratio of the bond dissociation energy (BDE) of the three bonds is 1:1.85:2.61. Clearly, LBO has surprisingly good correlation with BDE, in other words, LBO exhibits bonding strength fairly well (no other bond order definitions have so close relationship with BDE in comparison with LBO)

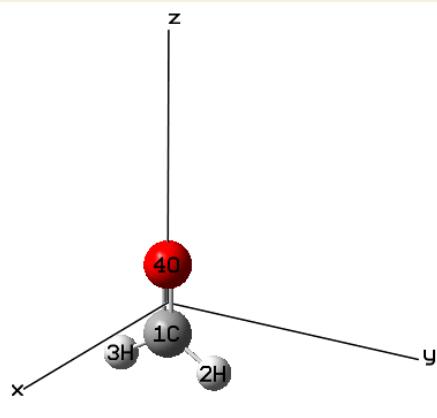
Moreover, LBO predicts that the sequence of the C-H bonding strength in the three systems is acetylene (0.958) > ethene (0.919) > ethane (0.889), this is completely in agreement with the experimental BDE sequence! (Other bond order definitions, such as Mayer bond order, fail to reproduce this sequence)

Finally, calculate LBO for O-H bond in water by using "examples\H2O.fch", the result is 0.585. This value is significantly smaller than the C-H bond order, reflecting that O-H bond is much more polar than C-H bond.

4.9.4 Decomposition analysis of Wiberg bond order in NAO basis for formaldehyde

This example briefly illustrates a unique feature of bond order analysis module of Multiwfn, namely decomposing Wiberg bond order to atomic orbital pair and atomic shell pair contributions. A very simple molecule formaldehyde will be used as example, of course you can extend the analysis to much more complicated systems. Please read Section 3.11.8 first to understand basic idea of this analysis method.

This analysis requires natural atomic orbital (NAO) information and density matrix in NAO basis outputted by Weinhold's NBO program. For Gaussian user, you can run *examples\H2CO_DMNAO.gjf* and use the output file (*examples\H2CO_DMNAO.out*) as input file for this analysis. The orientation of the H₂CO molecule in Cartesian system is shown in below graph.



Boot up Multiwfn and input below command:

```
examples\H2CO_DMNAO.out
```

```
9 // Bond order analysis
```

```
9 // Decompose Wiberg bond order in NAO basis
```

Then you can input two atom indices to obtain their Wiberg bond order calculated under NAO basis, and meantime obtain major components (the threshold for printing components is controlled by "bndordthres" parameter in settings.ini). For example, we input 1,4, below result is immediately shown on screen:

Contribution from NAO pairs that larger than printing threshold:

Contri.	NAO	Center	NAO type	NAO	Center	NAO type	
0.0823	2	1(C)	val(2s) s	---	21	4(O)	val(2s) s
0.1907	2	1(C)	val(2s) s	---	28	4(O)	val(2p) pz
0.9145	5	1(C)	val(2p) px	---	24	4(O)	val(2p) px
0.0658	7	1(C)	val(2p) py	---	26	4(O)	val(2p) py
0.2482	9	1(C)	val(2p) pz	---	21	4(O)	val(2s) s
0.3700	9	1(C)	val(2p) pz	---	28	4(O)	val(2p) pz

Contribution from NAO shell pairs that larger than printing threshold:

Contri.	Shell	Center	Type	Shell	Center	Type	
0.0823	2	1(C)	2s	---	2	4(O)	2s
0.1907	2	1(C)	2s	---	5	4(O)	2p
0.2482	5	1(C)	2p	---	2	4(O)	2s
1.3504	5	1(C)	2p	---	5	4(O)	2p

Total Wiberg bond order: 1.9161

From above information, the detail of total Wiberg bond order of 1.9161 becomes quite clear. According to the molecular graph shown earlier, the p_x type of NAO corresponds to the $2p$ atomic orbital perpendicular to molecular plane, thus the p_x-p_x mixing results in π bond, its contribution to the total bond order (0.9145) is close to unity, which is in line with chemical intuition. The $2s-2s$ interaction only has weak contribution to the C=O bond, since the value 0.0823 is almost negligible; the reason should be attributed to the fact that the orbital overlap is insufficient. In addition, the $2p_y-2p_y$ interaction also plays insignificant role, the contribution is merely 0.0658. The interaction between $2s(C)-2p_z(O)$, $2p_z(C)-2p_z(O)$ and $2p_z(C)-s(O)$ have remarkable contribution to total bond

order, which are 0.1907, 0.3700 and 0.2482, respectively, and the sum reaches as high as 0.8089. The large contributions must mainly stem from good orbital overlapping.

In order to facilitate discussion, the program also outputs contribution to Wiberg bond order from various atomic shell pairs. For example, as you can see from above information, interaction between all $2p$ orbitals of carbon and all $2p$ orbitals of oxygen totally contributes 1.3504 of bond order.

4.9.5 Study orbital contributions to Mulliken bond order for C-C bond of CH₃CONH₂

The Mulliken bond order has been introduced in Section 3.11.4, it is also known as Mulliken overlap population. This kind of bond order is not particularly useful, since it neither correlates well with bonding strength nor closely related to bond multiplicity. However, a unique advantage is that it can be exactly decomposed into orbital contributions, and positive and negative value correspond to bonding and anti-bonding effect, respectively, this feature is useful for unraveling characteristics of orbitals. In this section I will use CH₃CONH₂ as example to illustrate this point.

Boot up Multiwfn and input

examples\CH3CONH2.fch

9 // Bond order analysis

5 // Decompose Mulliken bond order between two atoms to orbital contributions

1,5 // Decompose C1-C5 bond

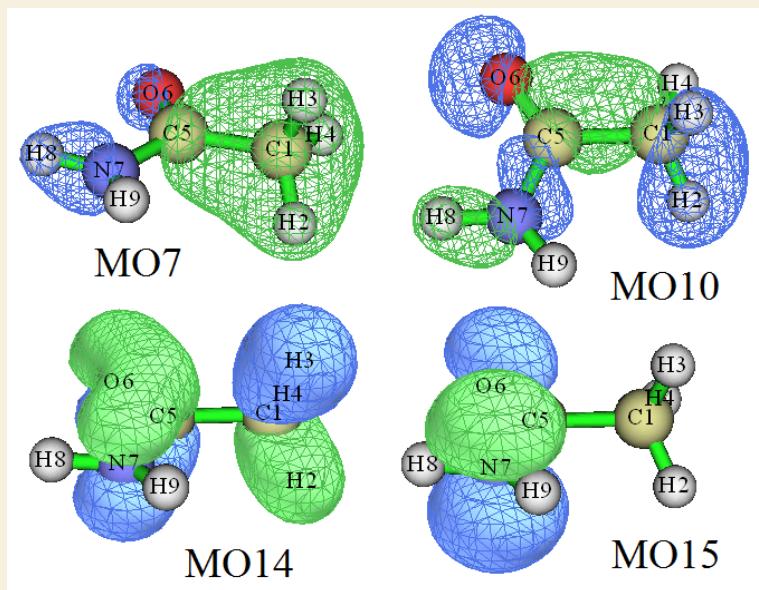
The result is

...[ignored]

orbital	7 occ:	2.000000 Energy:	-0.738679 contributes	0.30341308
orbital	8 occ:	2.000000 Energy:	-0.588379 contributes	-0.03865545
orbital	9 occ:	2.000000 Energy:	-0.541034 contributes	-0.00559743
orbital	10 occ:	2.000000 Energy:	-0.466302 contributes	0.20938473
orbital	11 occ:	2.000000 Energy:	-0.447220 contributes	0.12977388
orbital	12 occ:	2.000000 Energy:	-0.401575 contributes	0.14101027
orbital	13 occ:	2.000000 Energy:	-0.396099 contributes	-0.02147123
orbital	14 occ:	2.000000 Energy:	-0.367424 contributes	-0.11159089
orbital	15 occ:	2.000000 Energy:	-0.266106 contributes	0.00503022
orbital	16 occ:	2.000000 Energy:	-0.243833 contributes	0.02829203
Total Mulliken bond order: 0.66486037				

It can be seen that many MOs have evident positive contributions, such as MO7 (0.303), and MO10 (0.209); a few MOs have negative contributions, especially MO14 (-0.111). There are also some MOs have almost vanished contributions, such as MO15 (0.005). Therefore, occupation of MO7 and MO10 should enhance the strength of the C1-C5 bond, while occupation of MO14 must be harmful for formation of the C1-C5 bond.

The value of the MO contributions to Mulliken bond order can also be understood in terms of orbital isosurface map:



For MO7 and MO10, from above graph it can be seen that there is no nodal plane between C1 and C5, the isosurface substantially encloses the C1-C5 bonding region, thus MO7 and MO10 act as bonding orbital for C1-C5 and have positive contribution to its Mulliken bond order. For MO14, an evident nodal plane perpendicular to the C1-C5 bond can be clearly seen at the midpoint of C1-C5, clearly MO14 behaves as an anti-bonding orbital for C1-C5 bond and thus should have negative contribution to its Mulliken bond order. For MO15, the C1-C5 bonding region is not covered by the orbital isosurface, this is why contribution from MO15 to C1-C5 Mulliken bond order is negligible.

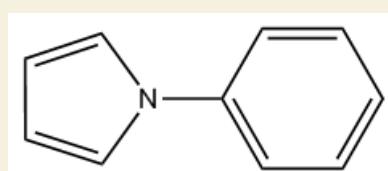
Beware that in rare cases the contribution values to Mulliken bond order cannot be well explained by isosurface map, showing deficiency of definition of Mulliken bond order. In this case you may try to use the orbital occupancy-perturbed Mayer bond order instead (as illustrated in Section 4.9.1), which is more robust.

It is noteworthy that decomposition of Mulliken and Mayer bond order can be carried out not only based on molecular orbitals, but also based on localized molecular orbitals (LMOs), in the latter case the discussion is usually more meaningful. To do so, in general you should use main function 19 to yield LMOs and then carry out the decomposition analyses as usual.

4.10 Plot density of states (DOS)

4.10.1 Plot total, partial and overlap DOS for N-phenylpyrrole

In this example, we will plot total, partial and overlap density-of-states (TDOS, PDOS and OPDOS) for N-phenylpyrrole, whose structure is shown below. This example consists of five parts. Please read Section 3.12.1 first if you are not familiar with DOS.



Because basis function information is required in plotting PDOS and OPDOS, we use .fch as input file in this example; .molden and .gms files are also OK, but .wfn/.wfx are not supported, since they do not contain information of basis function and virtual MOs. If you only need to obtain TDOS, you can also simply use a plain text file recording MO energy levels or the Gaussian output file with pop=full keyword as input file, see Section 3.12.1 for file format.

It is noteworthy that if you intend to plot PDOS and OPDOS, employing diffuse basis functions in the basis set is highly deprecated, because they lack of chemical meaning and thus always worsen the reliability of the results. The wavefunction of present system was generated at B3LYP/6-31G* level.

Hint: It is suggested using .pdf or .wmf graphical format instead of the default .png format for saving DOS maps via option 2 in post-process menu, since in this case the DOS map could be scaled losslessly and the curves look very smooth. The graphic format could be set via "graphformat" in settings.ini.

Part 1: Plot total DOS (TDOS)

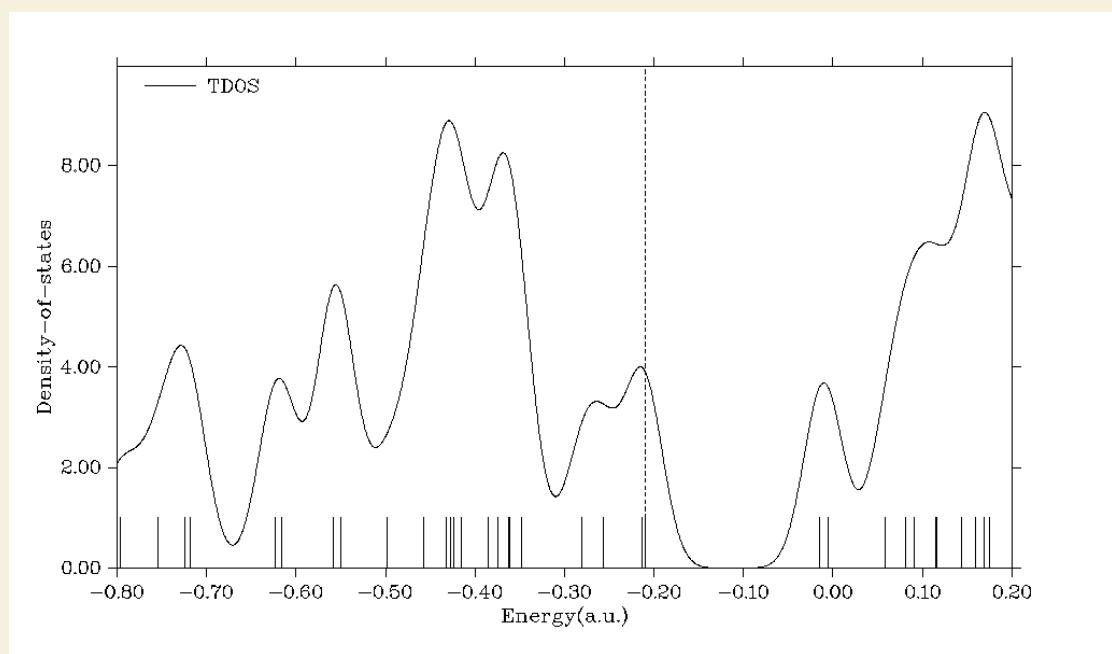
Boot up Multiwfn and input below commands

examples\N-phenylpyrrole.fch

10 // Plot various kind of DOS maps

0 // Plot map

Since currently no fragment is defined, only TDOS is plotted. The TDOS map pops up immediately, see below



In the TDOS map, each discrete vertical line correspond to a molecular orbital (MO), the dashed line highlights the position of HOMO. The curve is the TDOS simulated based on the distribution of MO energy levels. In the negative part, the region around -0.40 a.u. has obviously larger state density than other regions.

Clicking mouse right button on the graph to close it, and select option 0 to return to last menu.

Part 2: Plot PDOS and OPDOS for fragments

Next, we define the heavy atoms of pyrrole moiety as fragment 1 and that of the phenyl moiety as fragment 2 to check their PDOS and OPDOS. In addition, we define all hydrogens as fragment 3.

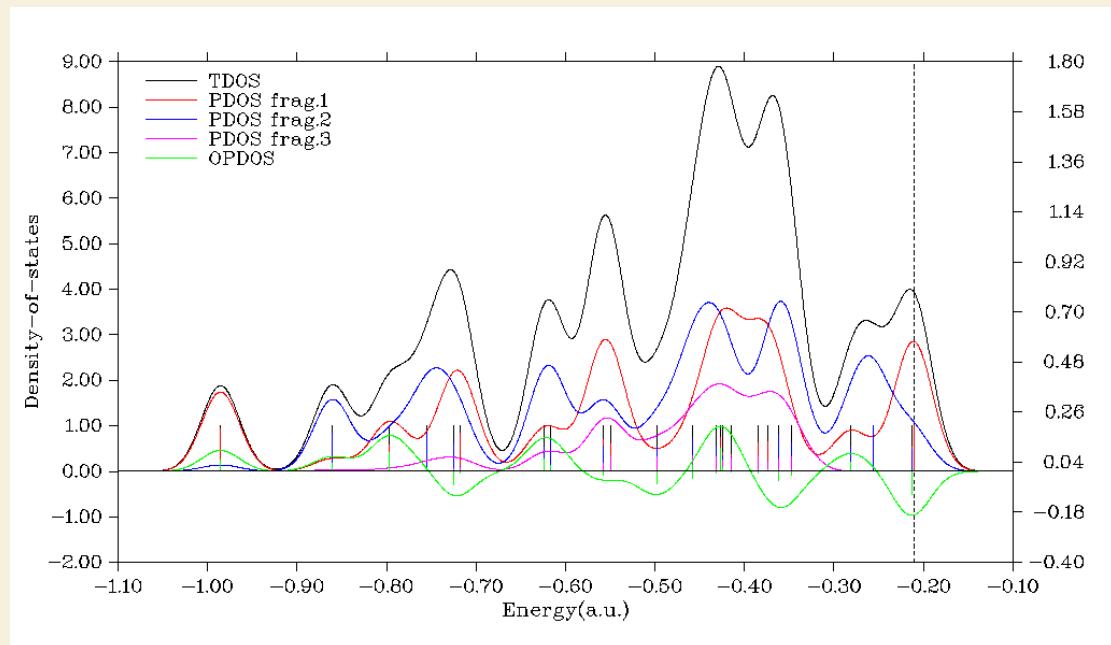
-1 // Enter the interface for defining fragments. You can define up to 10 fragments. PDOS will be plotted for all of them but OPDOS will only be drawn between fragment 1 and 2

```

1 // Define fragment 1
a 1-5 // Add carbons and nitrogen of pyrrole moiety (atoms 1~5) to the fragment
q // Save fragment 1
2 // Define fragment 2
a 10-13,15,17 // Add phenyl moiety (atoms 10~13, 15 and 17) to the fragment
q // Save fragment 2
3 // Define fragment 3
a 6-9,14,16,18-20 // Add all hydrogens to the fragment
q // Save fragment 3
0 // Return to last menu
2 // Set X-axis
-1.1,-0.1,0.1 // Set the range of X-axis to -1.2 ~ -0.1 a.u., so that all valence MOs can be
shown in the graph. The step between labels is set to 0.1 a.u..
0 // Draw TDOS+PDOS+OPDOS

```

Current graph is not very ideal. Close the graph, you can see many options used to customize the graph, such as setting curve colors, setting legend texts. Try to play with them and if you are confused you can consult Section 3.12.3. Here we select option 4 and input -2,9,1 to set lower, upper limits and stepsize of left Y-axis (corresponding to TDOS and PDOS) to -2.0, 9.0 and 1.0, respectively. Select 14 and input the scale factor 0.2, then range of the right Y-axis (corresponding to OPDOS) will be set to -0.4, 1.8 (because $-2.0 \times 0.2 = -0.4$ and $9.0 \times 0.2 = 1.8$). Shrinking the range of the Y-axis at right side is equivalent to enhancing the amplitude of OPDOS curve, which makes the variation of OPDOS in the map clearer. Then select 1 to replot the DOS map, you will see



The axis at left side corresponds to TDOS and PDOS, while the one at right side corresponds to OPDOS. The red, blue and magenta curves and discrete lines represent PDOS of fragment 1, 2 and 3, respectively. It can be seen that in most valence MOs, the fragments 1 and 2 have comparable amount of contribution. The fragment 3 (hydrogen) mainly contributes to the MOs between -0.60~-0.35 a.u. Green curve is the OPDOS between fragments 1 and 2, its positive part implies that the MOs in corresponding energy range show bonding character between the two fragments (e.g. the

one at -0.8 a.u., which corresponds to MO14); there are also regions where OPDOS is negative, *e.g.* the HOMO-1 (-0.213 a.u.) behave as antibonding orbitals between the two fragments.

Part 3: Plot PDOS of a specific atom orbital

Current molecule is in YZ plane, as an example, let us check the PDOS of *px* atomic orbital of the nitrogen atom, it represents π electron on this site. Select 0 to return to last menu and then input

```
-1 // Define fragments  
-2 // Fragment 2 is not needed, so we input corresponding negative value to unset it  
-3 // Also unset fragment 3  
I // Redefine fragment 1  
clean // Clean existing content of the fragment  
all // Print out information of all basis functions
```

The information corresponding to nitrogen atom is extracted and shown below

```
Basis: 61 shell: 25 Center: 5(N ) Type: S  
Basis: 62 shell: 26 Center: 5(N ) Type: S  
Basis: 63 shell: 27 Center: 5(N ) Type: X  
Basis: 64 shell: 27 Center: 5(N ) Type: Y  
Basis: 65 shell: 27 Center: 5(N ) Type: Z  
Basis: 66 shell: 28 Center: 5(N ) Type: S  
Basis: 67 shell: 29 Center: 5(N ) Type: X  
Basis: 68 shell: 29 Center: 5(N ) Type: Y  
Basis: 69 shell: 29 Center: 5(N ) Type: Z  
Basis: 70 shell: 30 Center: 5(N ) Type: XX  
Basis: 71 shell: 30 Center: 5(N ) Type: YY  
Basis: 72 shell: 30 Center: 5(N ) Type: ZZ  
Basis: 73 shell: 30 Center: 5(N ) Type: XY  
Basis: 74 shell: 30 Center: 5(N ) Type: XZ  
Basis: 75 shell: 30 Center: 5(N ) Type: YZ
```

Current system is calculated under 6-31G* basis set, according to the basis set definition, each valence atomic orbital is represented by two basis functions of corresponding type. Therefore, what we should do is to put basis functions 63 and 67 into the fragment, they collectively represent the *px* orbital of nitrogen (For other kinds of basis set, you can consult Section 4.7.6 on how to identify correspondence between basis functions and atomic orbitals). Input below commands

```
b 63,67 // Then you can input command all again, the basis functions added to present fragment are marked by asterisks
```

```
q // Save fragment  
0 // Return  
0 // Plot TDOS and PDOS
```

Please analyze the resulting graph by yourself.

Part 4: Plot PDOS of all π molecular orbitals

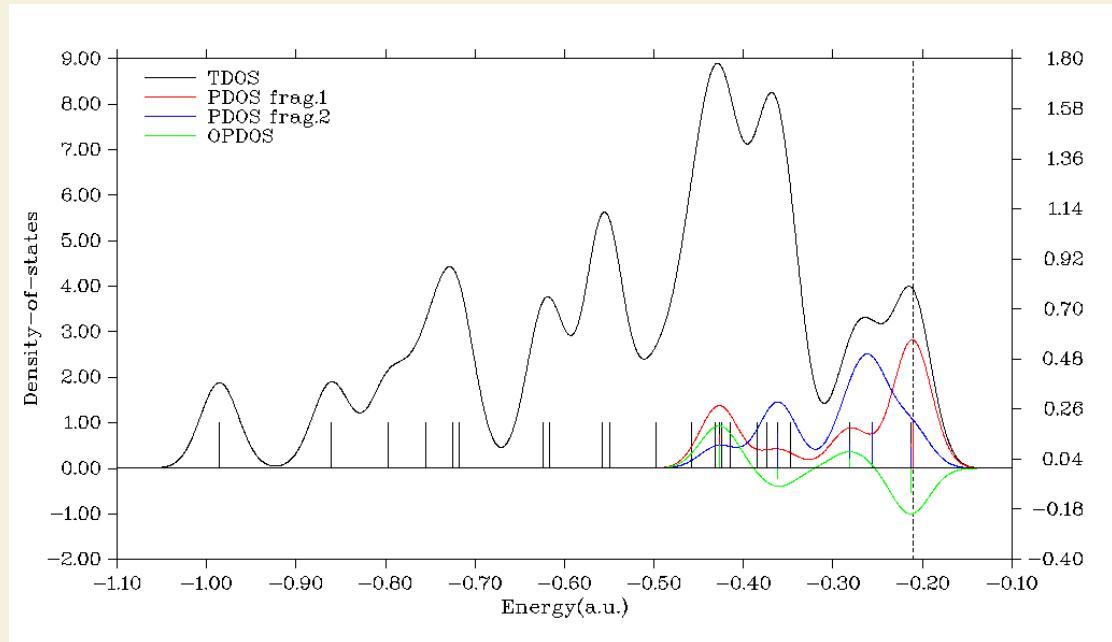
This molecule is in YZ plane, assume that we only intend to study PDOS/OPDOS of π orbitals of pyrrole and phenyl moieties and want to get rid of effect of all other MOs, although in the fragment definition interface we can choose each PX basis function in turn, because there are too

many atoms, this process will take you a lot of time and thus is very boring. A much better way is to use the condition selection command. Select 0 to return to last menu and then input

```

-I // Define fragments
I // Redefine fragment 1
clean // Clean existing content of the fragment
cond // Use conditions to select basis functions. You will be prompted to input three
conditions, the basis functions simultaneously satisfying the three conditions will be added to
current fragment
I-5 // The first condition is that the basis functions must belong to the heavy atoms in pyrrole
moiety (atoms 1~5)
a // The second condition is the index range of basis functions. Inputting a means any basis
function is OK (in other words, do not employ this condition)
X // The third condition is that the type of basis function should be PX
q // Save fragment 1
2 // Define fragment 2
cond
10-13,15,17 // Atom index of the carbons in phenyl moiety
a
X
q // Save fragment 2
0 // Return to last menu
0 // Draw TDOS+PDOS+OPDOS

```



This time the PDOS curves only cover high-energy regions, implying that most π MOs in present system have higher energy than σ MOs. Please use main function 0 of Multiwfn to visualize corresponding MO isosurfaces.

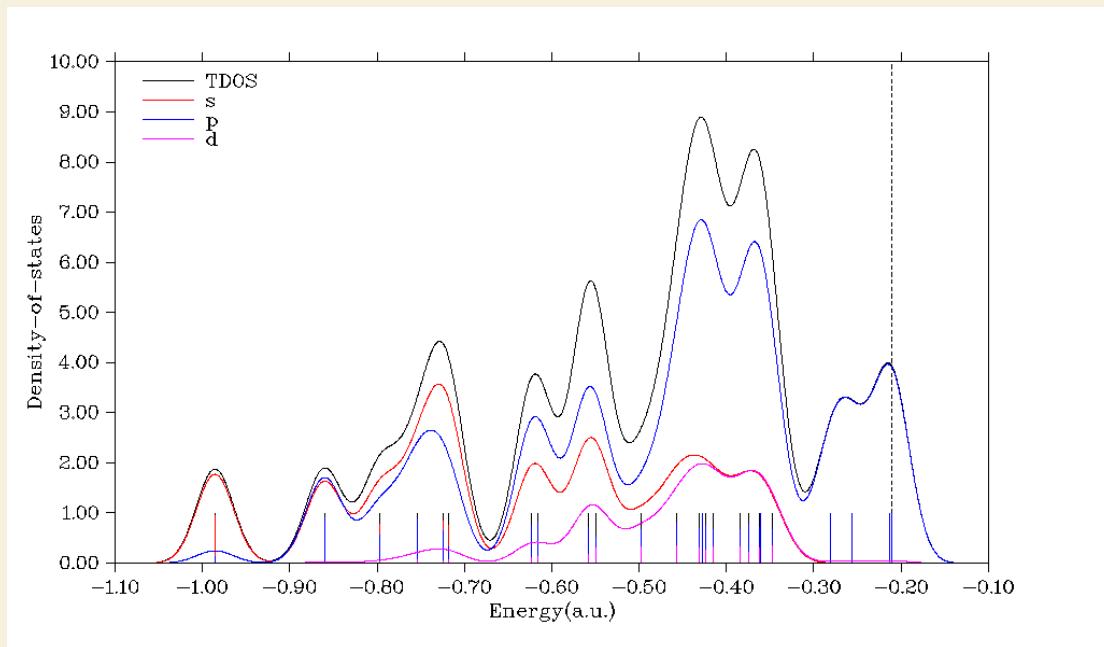
Part 5: Plot PDOS for s, p, d atomic orbitals individually

As final part of present example, we plot PDOS for s , p , d atomic orbitals individually. Reboot

Multiwfn and then input

```
examples\N-phenylpyrrole.fch
10 // Plot various kind of DOS maps
-1 // Define fragments
1 // Define fragment 1
ls // Add basis functions with angular moment of s to the fragment
q // Save fragment
2 // Define fragment 2
lp // Add basis functions with angular moment of p to the fragment
q // Save fragment
3 // Define fragment 3
ld // Add basis functions with angular moment of d to the fragment
q // Save fragment
0 // Return to last menu
0 // Draw TDOS+PDOS+OPDOS
Then close the graph and input
9 // Disable showing OPDOS curves
10 // Disable showing OPDOS lines
4 // Set range of Y axis
0,10,1 // Lower and upper limits are set to 0 and 10 with stepsize of 1.0
16 // Set legends
1 // Set legend of PDOS corresponding to fragment 1
s
2 // Set legend of PDOS corresponding to fragment 2
p
3 // Set legend of PDOS corresponding to fragment 3
d
0 // Exit the interface for setting legends
1 // Replot the map
```

Now you can see below map



From this map, it is clear that occupied frontier MOs are solely contributed by p orbitals.

4.10.2 Plot local DOS for 1,3-butadiene

If you do not know what is local DOS (LDOS), please check Section 3.12.4 first. Briefly speaking, TDOS represents DOS curve for the whole system, PDOS describes DOS curve for an atom (or fragment), while LDOS exhibits DOS curve for a point (*i.e.* space-resolved). In addition, we can plot LDOS for a set of point constituting a line as color-filled map, the X-axis corresponds to energy while the Y-axis shows position in the line. LDOS is useful when interpreting the data from scanning tunneling microscope (STM), you can find relating experimental data in *e.g. J. Phys. Chem. Lett.*, **5**, 3701 (2014).

In the current example, we plot LDOS for butadiene at selected points. First, we plot LDOS for the point over 1.5 Bohr of terminal carbon of butadiene. Boot up Multiwfn and input following commands:

examples\butadiene.fch

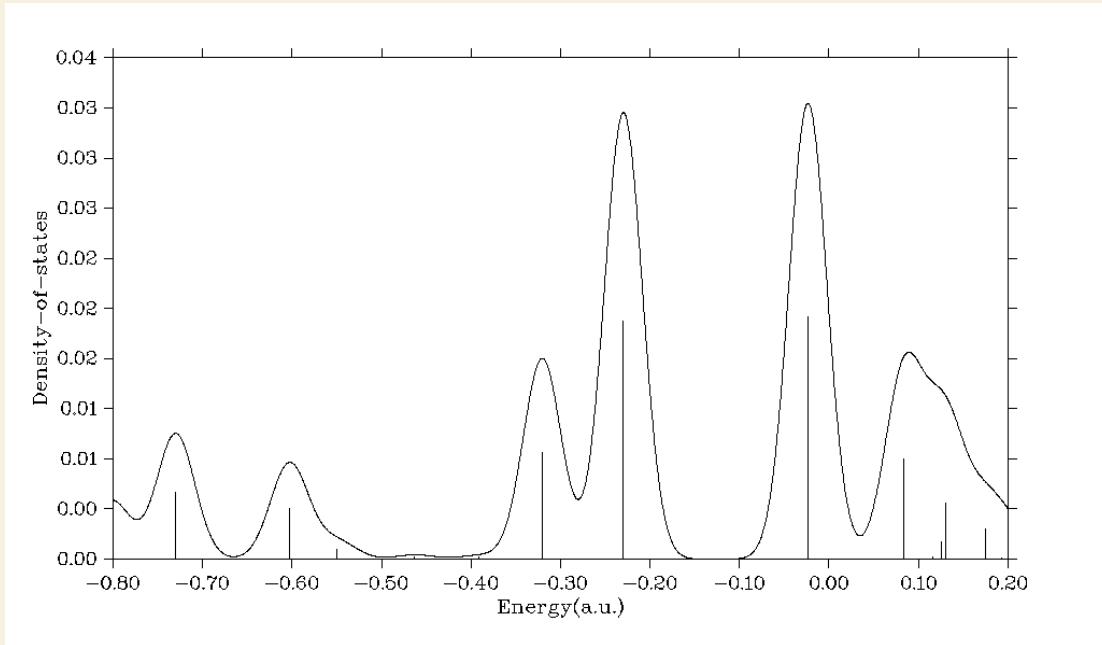
0 // From output in command-line we can find the expected point should be 1.137 3.308 1.5
(1.5 Bohr above C1)

10 // DOS plotting module

10 // Draw local DOS for a point

1.137 3.308 1.5

Then you will see (you can compare it with TDOS map)



Close the graph and select 0 to return to the last menu.

Next, we plot color-filled map along the line connecting the two points above 1.5 Bohr of the two terminal carbons (C1 and C8):

1.137 3.308 1.5

-1.137 -3.308 1.5

200 // Evenly taking 200 points along line

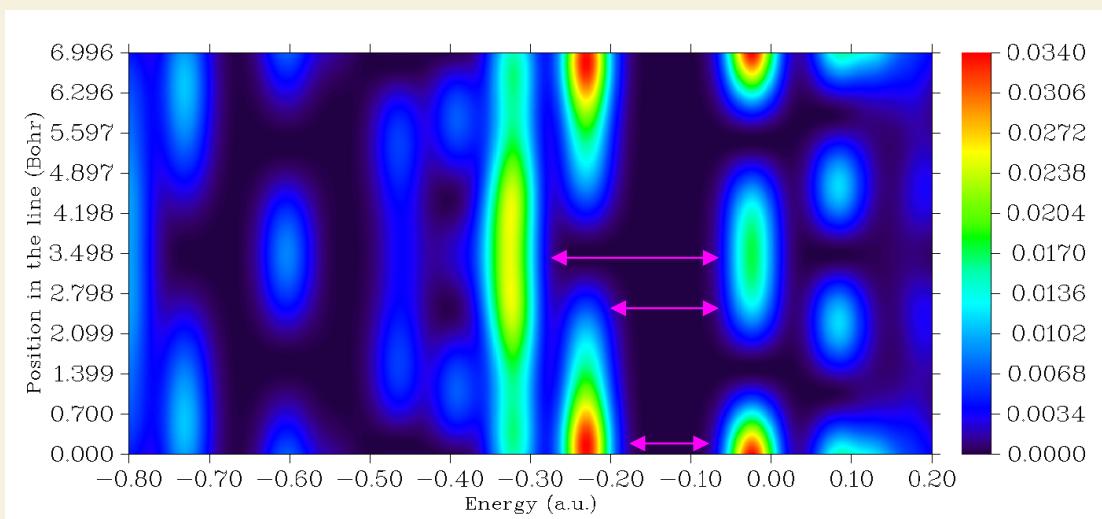
Then close the graph that pops up and input

4 // Modify the ratio between Y and X axes

0.5 // The length of Y-axis will be half of X-axis

1 // Replot

Then you can see



The lowermost part of the graph (viz. Y=0, standing for the point 1.5 Bohr above C1) corresponds to the LDOS curve map we have obtained. The pink arrows highlight the gaps at three different positions in the line.

4.11 Plot various kinds of spectra

Multiwfn has a very powerful spectrum plotting module. The basic principles, supported input files and all options of this module have been detailedly introduced in Section 3.13, below sections will briefly exemplify the usage of this module.

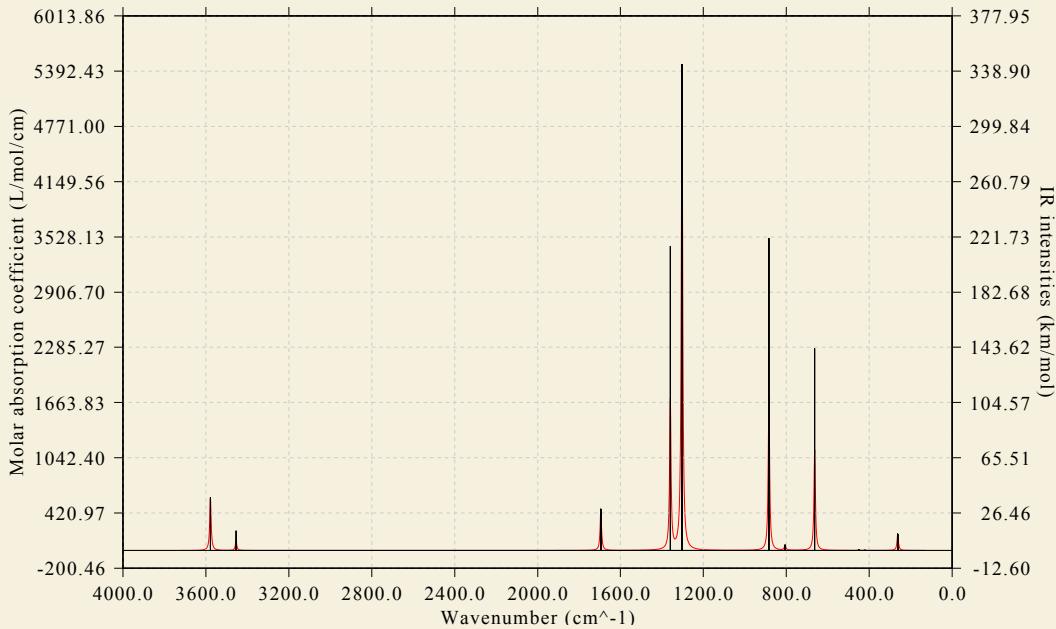
4.11.1 Plot infrared (IR) spectrum for NH₃BF₃

This example plots infrared (IR) spectrum for NH₃BF₃. Multiwfn can read in frequencies and intensities from output file of Gaussian or ORCA vibration analysis task (“freq” keywords). Boot up Multiwfn and input following commands

```
examples\spectra\NH3BF3_freq.out // The output file of optimization and vibrational analysis task of Gaussian at B3LYP/6-31G* level
```

```
11 // Plot spectrum
1 // The type of the spectrum is IR
0 // Plot spectrum right now
```

You will get below graph



The left axis corresponds to curve (broadened data), the right axis corresponds to discrete lines (original transition data). Plotting parameters such as full width at half maximum (FWHM), broadening function, unit and range of the axes can be adjusted by corresponding options in the interface. The graph and X-Y data set of discrete lines/curve can be exported by option 1 and 2, respectively.

It is well known that the frequencies produced under harmonic approximation deviate to experimental vibrational frequencies systematically. In order to correct this problem, fundamental frequency scale factor should be applied, this can be done easily in Multiwfn. We close the spectrum and then choose "14 Multiply the vibrational frequencies by a factor", then press ENTER button directly to choose all vibrational modes, then press ENTER button directly again to employ the scale factor fitted for B3LYP/6-31G* level, namely 0.9614, which can be found in Table 1 of *J. Phys.*

Chem., **100**, 16502 (1996). After that, if you replot the spectrum, the resulting spectrum will correspond the scaled one.

Many experimental spectra use different convention to the spectrum we obtained above, namely the Y-axis corresponds to transmission rather than absorption, therefore the peaks point downwards. To plot such a spectrum, you can select "4 Set left Y-axis" and then input e.g. 6000,-200,-400 to set lower limit, upper limit and stepsize to 6000, 200 and -400, respectively. Since currently lower limit is larger than upper limit, the Y-axis is inverted.

The procedure of plotting Raman, UV-Vis, electronic/vibrational circular dichroism (ECD/VCD) spectra are very similar with plotting IR spectrum. If the quantum chemistry program you used for spectrum calculation is not Gaussian, ORCA or sTDA, you can manually extract data from corresponding output file and then write them into a plain text file according to the format shown in Section 3.13.2, then the file can be used as input file of Multiwfn for plotting spectrum.

4.11.2 Plot UV-Vis spectrum and contributions from individual transitions for acetic acid

The spectrum plotting module of Multiwfn is quite flexible, not only the total spectrum but also the contribution from individual transitions can be exported. This feature is particularly useful when you want to identify nature of spectrum. In this section I will show how to realize this analysis, UV-Vis spectrum of acetic acid is taken as example.

Boot up Multiwfn and input

```
examples\spectra\acetic_acid_TDDFT.out // Calculated at TD-B3LYP/cc-pVDZ level by Gaussian
```

```
11 // Plot spectrum
```

```
3 // The type of the spectrum is UV-Vis
```

```
15 // Output the spectrum including the contributions from certain individual transitions
```

```
0.01 // The criterion of selecting transitions is oscillator strength > 0.01
```

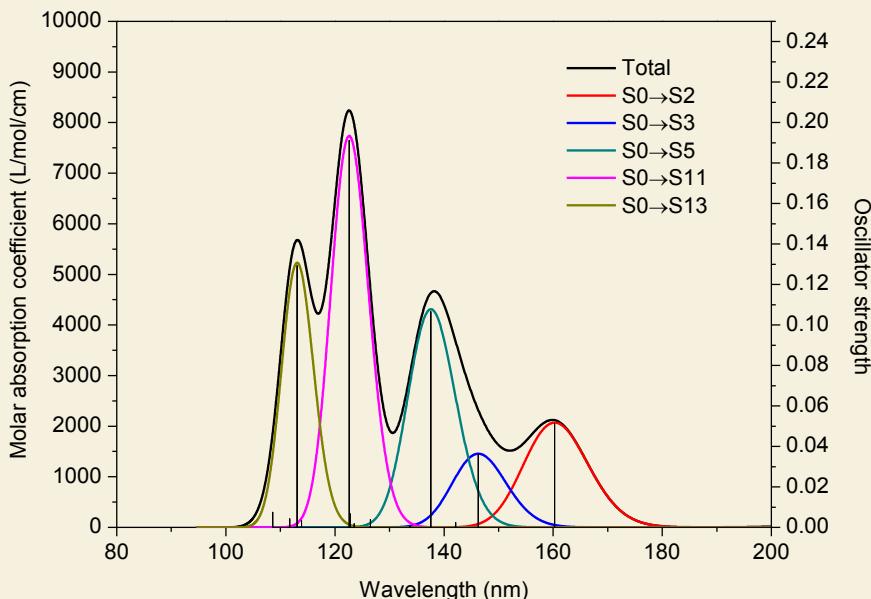
The curve of the UV-Vis spectrum together with the contributions from the transitions whose absolute value of strength are larger than 0.01 have been outputted to "spectrum_curve.txt" in current folder. The first two columns correspond to energies and molar absorption coefficients, the correspondence between the other columns and transition modes are clearly indicated on screen:

Column#	Transition#	
3	2	//i.e. transition S0→S2
4	3	//i.e. transition S0→S3
5	5	//i.e. transition S0→S5
6	11	//i.e. transition S0→S11
7	13	//i.e. transition S0→S13

The discrete line data are outputted to "spectrum_line.txt" in current folder.

Now you can plot the data in the two files as curves in a single graph by your favourite program (if you use Origin to plot, you can directly drag these two files into Origin window to import them). In the two files, the first column should be taken as X-axis data, while the other columns should be taken as Y-axis data. The spectrum plotted by Origin is shown below, if you are confused about the procedure, you can consult "acetic_acid_TDDFT.opj" provided in "examples\spectra" folder, which

is the corresponding .opj file of Origin 8.



From the graph the underlying character of the total UV-Vis spectrum (black curve) is now very clear. Although the $S_0 \rightarrow S_3$ transition (146.28nm) does not have very small oscillator strength (0.036), no absorption peak directly corresponds to this transition, since its absorption curve (blue curve) has been completely merged into the neighboring large absorption peak due to $S_0 \rightarrow S_5$ transition (cyan curve).

Multiwfn is able to locate minima and maxima of spectrum curve. Select option 16, then you will see their positions and values. This feature is quite useful for pointing out the exact position of the peaks

```

Local maximum x:    113.1515      value:    5680.8733
Local maximum x:    122.6050      value:    8239.3520
Local maximum x:    138.1754      value:    4667.8074
Local maximum x:    159.8627      value:    2123.3503
Local maximum x:    213.9266      value:    48.5312

Local minimum x:    116.9205      value:    4222.4896
Local minimum x:    130.5755      value:    1866.1290
Local minimum x:    152.0775      value:    1515.6418
Local minimum x:    187.9759      value:    1.5107

```

Totally found 4 local minimum, 5 local maximum

Based on above outputs, we can calculate the contributions from different transitions to a peak. For example, we want to study the composition of the peak at 138.1754 nm. In spectrum_line.txt, move to the line corresponding to 138.17536 nm, you can find the total value is 4667.80738, while the values in column 4 and 5 are 310.33658 and 4272.69585, respectively. Therefore, the contribution from $S_0 \rightarrow S_3$ and $S_0 \rightarrow S_5$ can be respectively calculated as $310.33658/4667.80738*100\% = 6.65\%$ and $4272.69585/4667.80738*100\% = 91.53\%$.

4.11.3 Plot electronic circular dichroism (ECD) spectrum for asparagine

In this example we plot electronic circular dichroism (ECD) spectrum for asparagine. Boot up Multiwfn and input

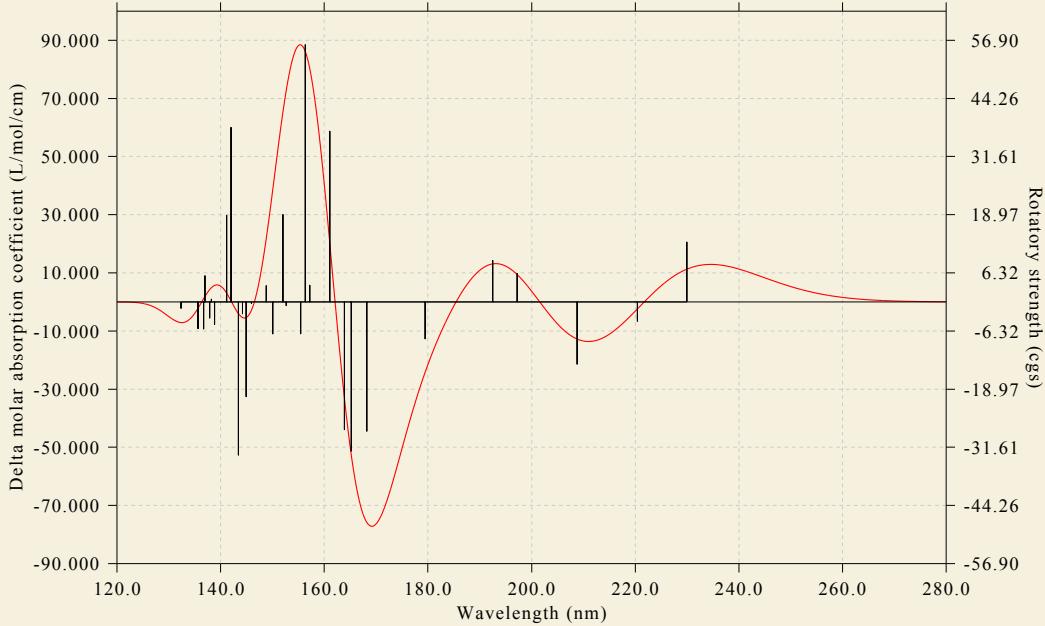
```
examples\spectra\Asn_TDDFT.out // Gaussian TDDFT task at PBE0/6-311G* level, 30 lowest excited states were calculated
```

```
11
4 // ECD
2 // Use the rotatory oscillator in velocity representation
0 // Plot spectrum
```

From the resulting spectrum, you will find the labels of X-axis and Y-axis are decimal. In order to make the graph more beautiful, it is suggested to modify the scale so that label of each tick is integer. Therefore, we close current graph and input below commands:

```
3 // Set X-axis
120,280,20 // Lower and upper limits, as well as spacing between ticks of X-axis
4 // Set left Y-axis
-90,100,20 // Lower and upper limits, as well as spacing between ticks of left Y-axis
y // Let program properly adjust right Y-axis to guarantee that zero point of left and right axes are in the same horizontal line
0 // Plot spectrum
```

Then you will see below graph



Note that you can use exactly the same way as that illustrated in Section 4.11.2 to decompose the total ECD spectrum to individual contribution from each transition.

4.11.4 Plot conformational weighted UV-Vis and ECD spectra for plumericin

For a flexible system with many thermally accessible conformation (or configurations), when plotting its spectrum, it is crucial to take weighting average of various conformations into account, otherwise the resulting spectrum is impossible to be compared well with experimental spectrum. Fortunately, weighted spectrum can be very conveniently plotted by Multiwfn, I will show how to do this in present section. Plotting conformationally weighted UV-Vis and ECD spectra of plumericin are taken as instances.

(1) Preparation

Before calculating conformationally weighted spectrum, we need to evaluate population of these conformations, commonly Boltzmann's weight is used. We assume that plumericin has four accessible conformations, and properly construct their initial geometries, then optimize them and perform frequency analysis at B3LYP/6-31G* level with zero-point energy scale factor of 0.9806. Based on optimized geometries, high-accuracy single point energies at M06-2X/def2-TZVP level were calculated. Finally, we add the Gibbs thermal corrections produced by frequency analysis to the single point energies to yield relatively accurate Gibbs energy of various conformations. After that, according to Boltzmann's formula and relative Gibbs energy among these conformations, we calculate their weights at 298.15K. Then, using TD-PBE0/TZVP level we calculate the lowest 20 excited states for these conformations. The output file of the TDDFT tasks have been provided in "examples\spectra\weighted" folder as a.out, b.out, c.out and d.out (the names are arbitrary, you can also use other file names).

Now we write a plain text file named "multiple.txt" with below content (the name "multiple.txt" should not be changed, otherwise Multiwfn will unable to properly recognize it):

```
examples\spectra\weighted\ a.out 0.6046  
examples\spectra\weighted\ b.out 0.1950  
examples\spectra\weighted\ c.out 0.1686  
examples\spectra\weighted\ d.out 0.0317
```

The first column records path of Multiwfn input file for various conformations, the second column is the Boltzmann weights we calculated above. Below the four conformations will be called as *a*, *b*, *c* and *d*, respectively.

PS: If you are using Linux system, and there are / symbols or space in the path, do not forget to add double quotation marks at the two ends of the path, for example:

```
"./sob/weight/a.out" 0.8  
"./sob/weight/b.out" 0.2
```

(2) Plot conformationally weighted UV-Vis spectrum

Boot up Multiwfn and input

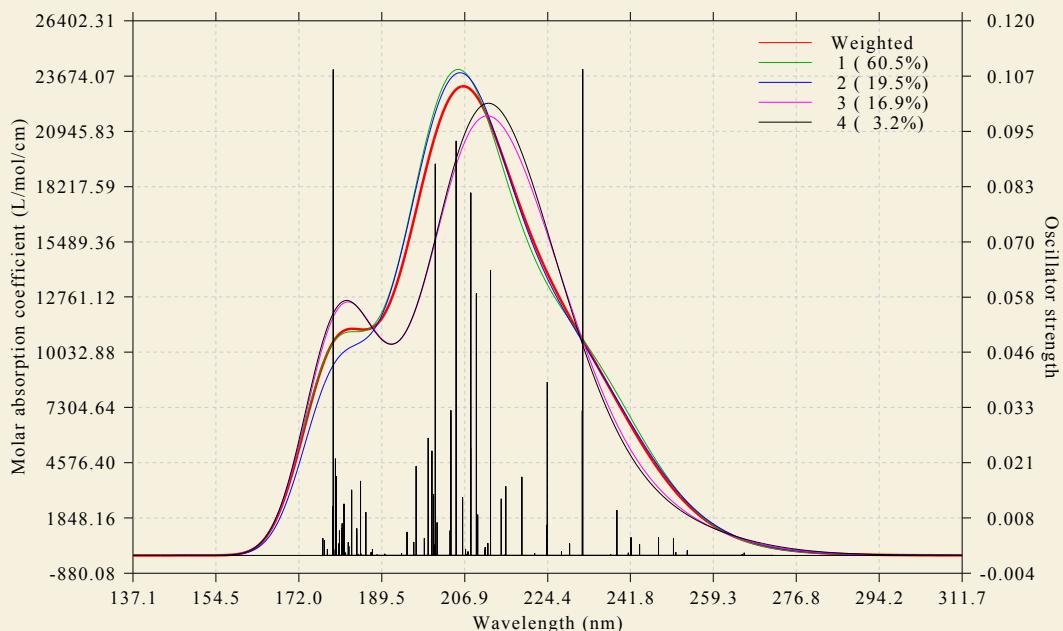
```
examples\spectra\weighted\multiple.txt // The file mentioned in last section
```

```
11
```

```
3 // UV-Vis
```

```
0 // Plot the spectrum
```

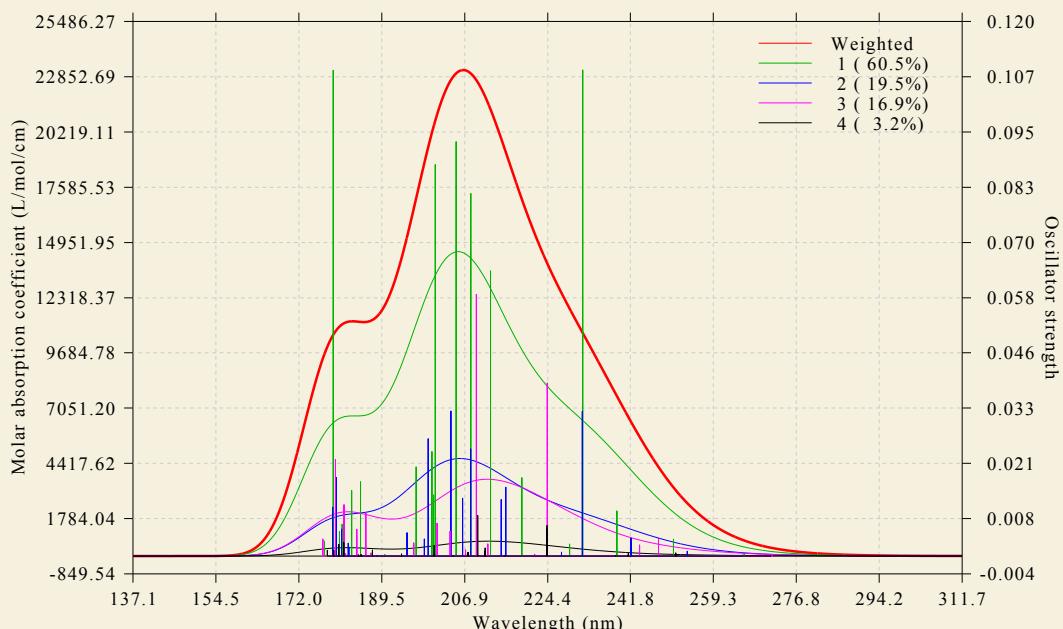
The resulting graph is shown below



The thick red curve corresponds to conformationally weighted UV-Vis spectrum, while the green, blue, purple and black curves correspond to UV-Vis spectrum of conformation *a*, *b*, *c* and *d*, respectively. The weight of each conformation is also shown in the legend. From this graph we can very conveniently compare the character of weighted spectrum and spectra of individual conformations.

The black discrete lines on the graph represent all transition data of the four conformations, their heights have already been scaled by conformational weight. Hence, the thick red curve can be regarded as broadened by all discrete lines shown on the graph.

Multiwfn provides another mode to plot spectrum of individual conformations. We close above graph and choose "18 Toggle weighting spectrum of each system" once, then choose option 0 to view the spectrum again, we will see



The spectrum curve of each conformation shown on this graph has already been multiplied by

corresponding weight. Obviously, what this graph represents is contribution of each conformation to the conformationally weighted spectrum. In other words, the height of thick red curve is simply the sum of height of all other curves. It can be seen that conformation *a* (green line) has major contribution to the conformationally weighted spectrum, their profiles are rather similar, this is because *a* has as high as 60.5% population.

The discrete lines in above graph now have different colors, the color correspond to legend shown at right-top side. For each conformation, since both discrete lines and curve currently have identical color, we can say for example, green curve can be directly yielded by broadening the green discrete lines.

(3) Plot conformationally weighted ECD spectrum

Using the same procedure illustrated in the last section, we plot conformationally weighted ECD spectrum and ECD spectrum for all the four conformations.

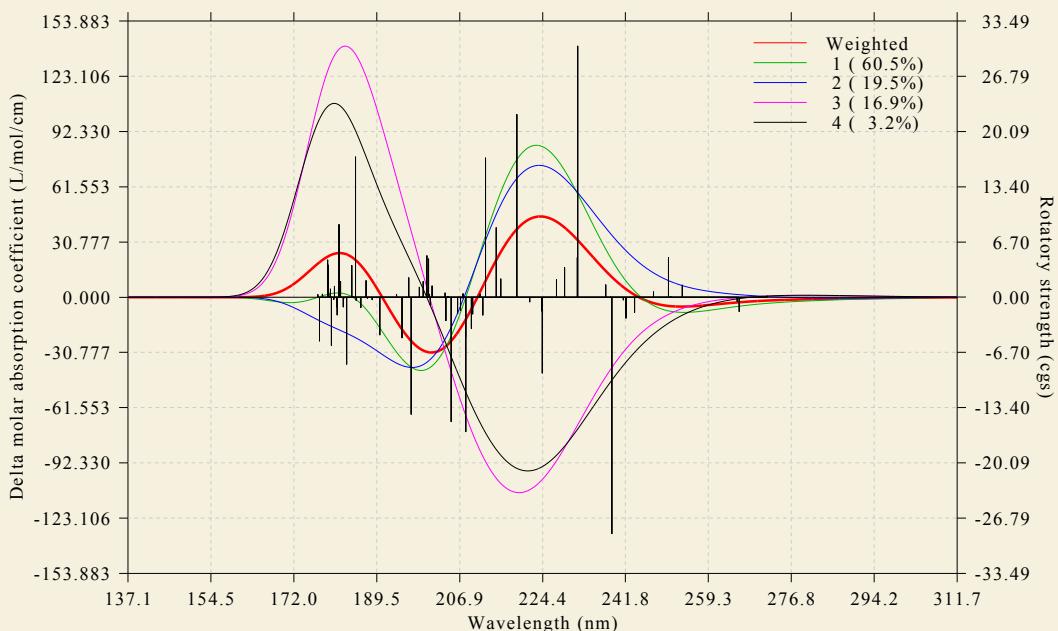
Boot up Multiwfn and input

examples\spectra\weighted\multiple.txt

II

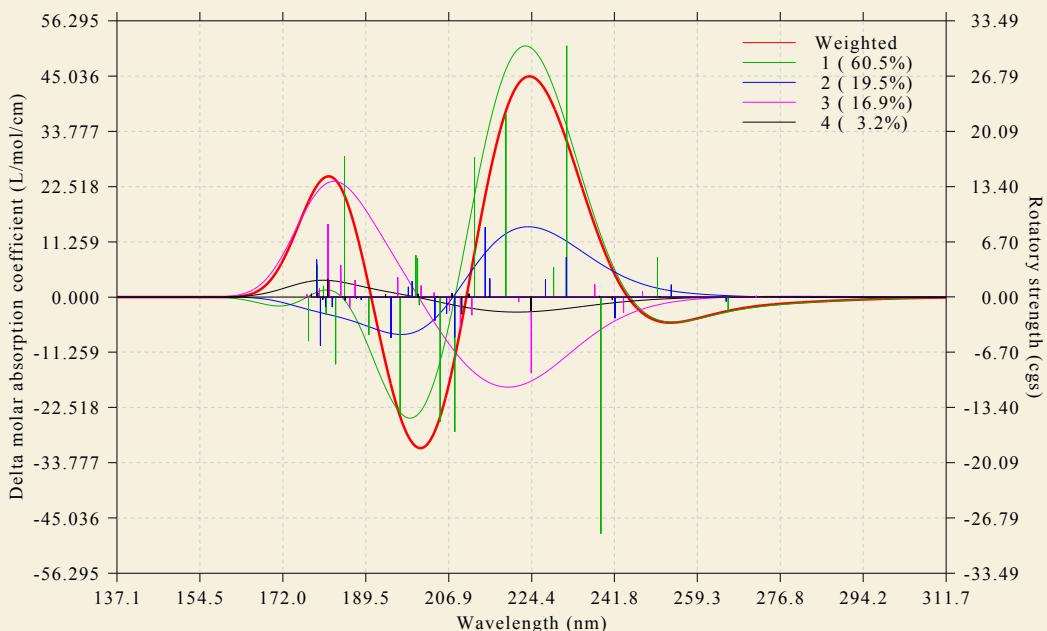
```
4 // Plot ECD
2 // Read rotatory strength in velocity representation from the 1st file in multiple.txt
2 // ibid, but for the 2nd file
2 // ibid, but for the 3rd file
2 // ibid, but for the 4th file
0 // Plot spectrum
```

You will see



From this graph you can see conformationally weighted ECD spectrum (thick red curve) as well as ECD spectrum of individual conformations (other curves).

Then choose "18 Toggle weighting spectrum of each system" option once and plot spectrum again, you will see



This graph decomposes the final weighted ECD spectrum to contribution of individual conformation. Again, since conformation *a* (green) has very high population and thus dominates the final weighted curve, most characters of these two curves are similar. However, influence from other conformations cannot be simply ignored. From the graph it is easy to find that if conformation *c* (purple) is missing, then there will not be an evident ECD peak at approximately 180 nm, since only ECD of *c* at this wavelength has significant signal.

4.11.5 Plot Raman spectrum for 2-methyloxirane based on Raman intensity

The procedure of plotting Raman spectrum is very similar with plotting IR spectrum, the only additional step you would better do is to convert the Raman activities directly outputted by Raman task of quantum chemistry codes to Raman intensities before plotting the spectrum, so that the resulting spectrum can be comparable with the experimental one. The Raman intensities are dependent of wavelength of incident light source and ambient temperature, while Raman activities are not. This point has been emphasized in Section 3.13.1. In this section, I illustrate how to properly plot Raman spectrum using (2S)-2-methyloxirane as example.

Boot up Multiwfn and input

```
examples\spectra\2-methyloxirane_Raman.out // Output file of Raman task calculated at
B3LYP/6-31G* level by Gaussian09
```

11

2 // Raman spectrum

14 // Apply frequency scale factor

[Press ENTER button] // Select all frequencies

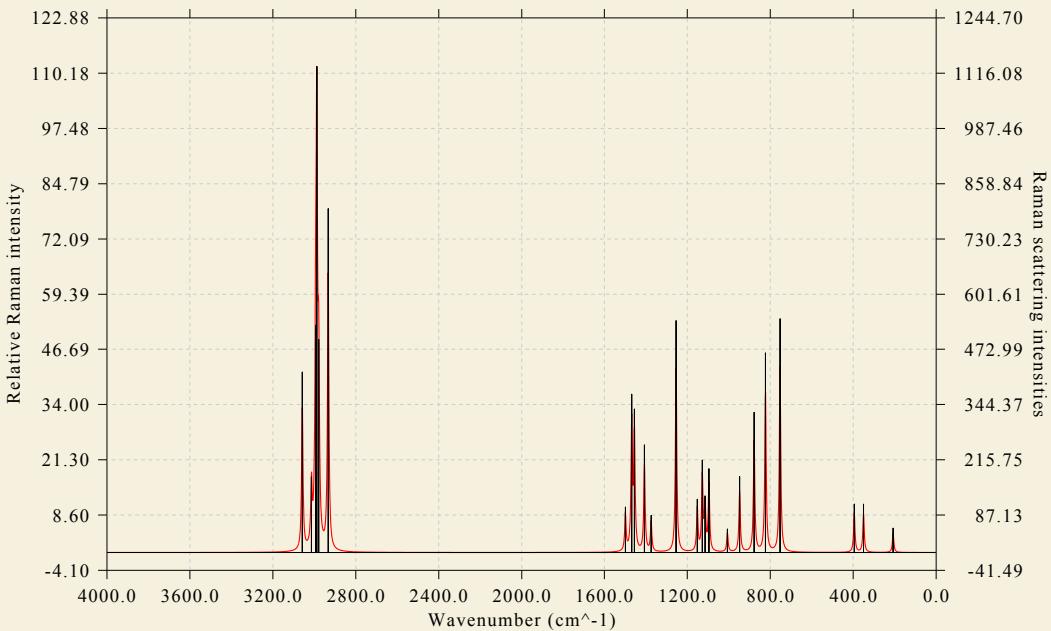
[Press ENTER button] // Employ the fundamental scale factor 0.9614, which is suitable for B3LYP/6-31G* level

19 // Convert Raman activities to intensities

15000 // Wavenumber (cm^{-1}) of incident light. This value should be consistent with actual experimental condition, the value we inputted here is arbitrarily chosen

298.15 // Assume that experimental temperature is 298.15K (You can also press ENTER button directly, 298.15K will be used as default)

Then input *0* to plot spectrum, you will see below Raman spectrum



Multiwfn can also plot pre-resonance Raman spectrum. An example output file of pre-resonance Raman task of Gaussian is *examples\spectra\2-methyloxirane_Raman.out*, the corresponding input file (.gjf) is also given. This task calculates Raman activity at incident wavelength of 150nm and 140nm, which are close to S0→S1 and S0→S2 TDDFT excitation energies at the same calculation level (147.36nm and 138.81nm at TD-B3LYP/6-31G* level, respectively). You can load this output file into Multiwfn and plot Raman spectrum as usual. The only difference is that, before entering spectrum plotting interface, Multiwfn asks you to choose the incident frequency for which the Raman activities will be loaded. If you choose 2 or 3, the spectrum you finally obtained will be pre-resonance Raman at corresponding frequency; if you choose "1: 0.00000000", namely the static limit case, the resulting spectrum will be exactly identical to the one we obtained earlier.

4.11.6 Simultaneously plot multiple systems

In Multiwfn, it is very easy to plot spectrum for multiple systems simultaneously, these systems may correspond to different conformations, different configurations, different molecules or different calculation conditions. In this section two examples are provided.

Compare spectra yielded by different theoretical methods and basis sets

In "examples\spectra\indigo" folder, you can find Gaussian output file of electronic excited state task carried out at different levels. In this example, we plot them together so that their results can be conveniently compared.

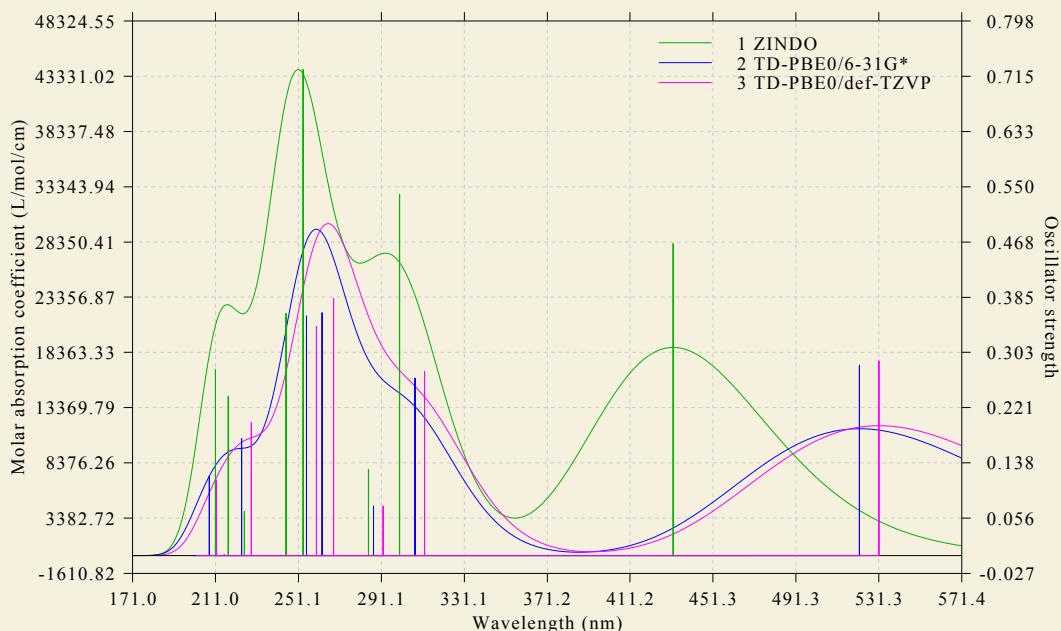
What we need to do first is to prepare a file named *multiple.txt* including path of various

systems with their legends, this file has been provided as *examples\spectra\indigo\multiple.txt*, its content is:

```
examples\spectra\indigo\ZINDO.out ZINDO
examples\spectra\indigo\TD-PBE0.out TD-PBE0/6-31G*
examples\spectra\indigo\TD-PBE0_TZVP.out TD-PBE0/def-TZVP
```

Note that the legends must not simply be a digital, otherwise it will be interpreted as weight of corresponding system (see Section 4.11.4). In addition, in Linux system, if the file path contains / symbol, do not forget to add double quotation marks at the two ends of the path.

Boot up Multiwfn, load *multiple.txt* and then plot UV-Vis spectrum as usual, you will see below graph



From the graph it is clear that basis set only has small influence on the resulting spectrum, while the spectrum profile of ZINDO differs from that of TD-PBE0 significantly. The curve of all systems may be exported via option 2 as *curveall.txt* and then replotted via third-part program such as Origin.

In "examples\spectra\indigo" folder you can also find *ZINDO_30.out*, it corresponds to a ZINDO calculation with 30 excited states produced. You can also include it into *multiple.txt*, however, notice that since it contains more states than other files, it must be added as the first entry in *multiple.txt*. (**It is compulsory that the file with maximum number of transitions must present at the beginning of the list**)

Compare spectra with and without spin-orbit coupling effect

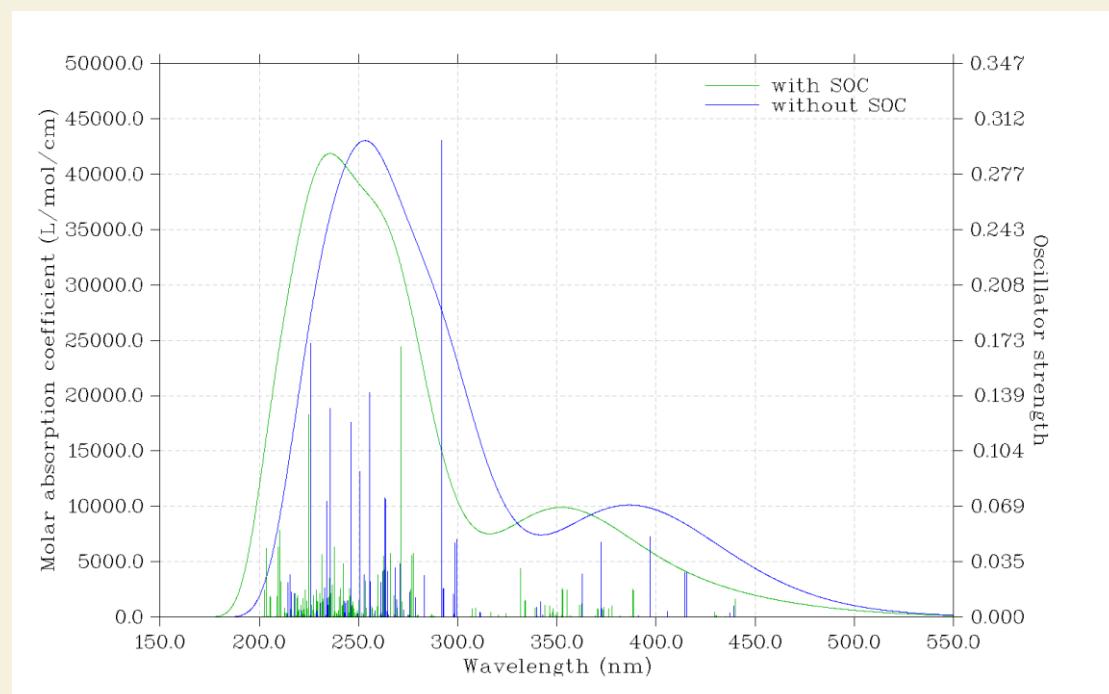
ORCA program is able to take spin-orbit coupling (SOC) effect into account during TDDFT calculation, here we plot and compare the UV-Vis spectrum with and without the SOC consideration. Please download http://sobereva.com/multiwfn/extrafiles/SOC-TDDFT_ORCA.zip, it is an ORCA output file of TDDFT task for Ir(ppy)₃ coordinate, SOC treatment had been enabled in the calculation via *dosoc true* keyword in the %tddft field. In the output information, there are excitation energies and oscillator strengths before and after SOC correction.

Extract the .zip package, put the .out file in current folder, then create a *multiple.txt* with below content

```

Ir_ppy3.out with SOC
Ir_ppy3.out without SOC
Boot up Multiwfn and input
multiple.txt
11 // Plot spectrum
3 // Plot UV-Vis
y // For the first spectrum, let Multiwfn use the data with SOC consideration
n // For the second spectrum, let Multiwfn use the data without SOC consideration
Then you will enter the interface for setting up the spectrum. After slight adjustment of settings,
you will obtain below graph. Clearly, SOC effect has non-negligible influence on the spectrum for
present systems.

```



It is worth to note that the number of states with SOC is by far larger than that without SOC. Because SOC effect splits each originally degenerate triplet state to three sublevels. For example, assume that the TDDFT calculates 50 singlets and 50 triplets, then after taking SOC correction into account, there will be $50+3*50=200$ states. Since the number of states corresponding to SOC-TDDFT is higher than that corresponding to regular TDDFT, when two (or more) set of data are simulated as theoretical spectra, the first legend in *multiple.txt* must correspond to SOC-TDDFT case, and when loading data for the first spectrum, you must choose *y* to let Multiwfn load SOC corrected TDDFT data, as I illustrated above.

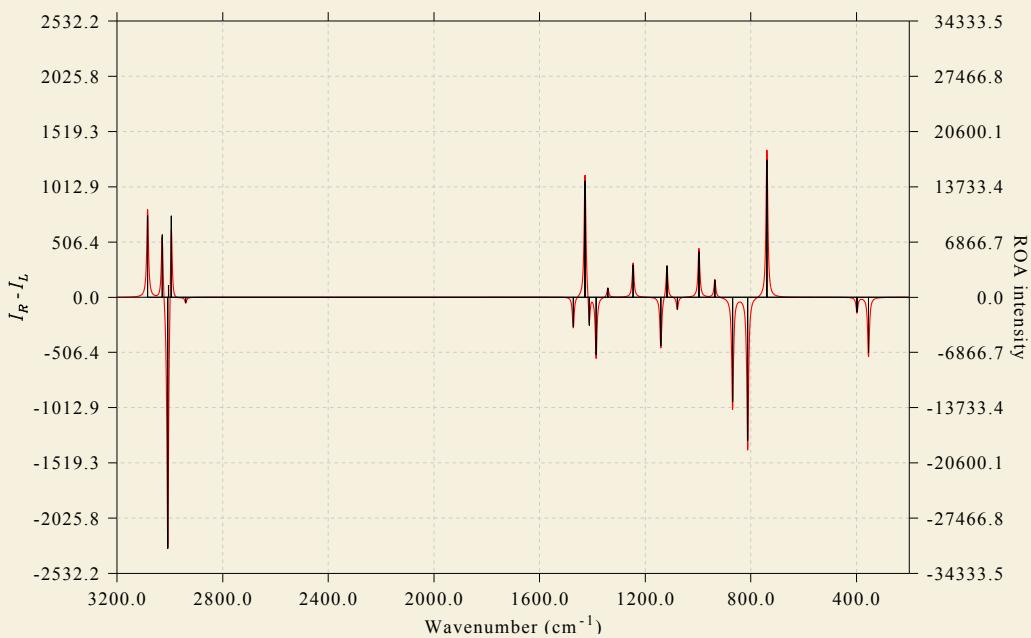
4.11.7 Plot Raman optical activity (ROA) spectrum for chiral molecule S-methyloxirane

ROA is an important type of vibrational spectrum for chiral molecule, only chiral molecule has ROA signal, see Section 3.21 for detail. In this example, I will illustrate how to plot ROA spectrum for a typical chiral molecule S-methyloxirane based on output file of Gaussian *freq=ROA* task. The

Gaussian input and output files are *S-methyloxirane_ROA.gif* and *S-methyloxirane_ROA.out* in *examples\spectra* folder, respectively. As can be seen from the input file, this calculation takes three incident light frequencies (500, 532 and 600 nm) into account. It is well-known that diffuse functions are important for obtaining accurate ROA data, so aug-cc-pVDZ is used here.

```

Boot up Multiwfn and input
examples\spectra\S-methyloxirane_ROA.out
11 // Plot spectra
6 // ROA
2 // Three incident light frequencies are detected, here we select the 532nm case
2 // There are totally six kinds of data can be selected, here we select the commonly studied
"ROA SCP(180)", namely backscattered circular polarization ROA spectrum
14 // Scale frequencies by a scale factor
[Press ENTER button] // Select all frequencies
0.97 // Employ fundamental scale factor of 0.97, which is suitable for B3LYP/aug-cc-pVDZ
level
19 // Convert the ROA data outputted by Gaussian to "real" ROA intensities
532nm // Wavelength of incident light. This value should be consistent with actual
experimental condition
[Press ENTER button] // Assume that experimental temperature is 298.15K
3 // Adjust range of X axis of the spectrum
3200,200,400 // Lower limit, upper limit, stepsize
0 // Plot the spectrum
Now you can see below ROA spectrum:
```



If you intend to present this spectrum in paper, I suggest removing the labels in both the left and right ordinates, since they are meaningless, only the shape of the curve is of chemical interest.

4.12 Quantitative analysis of molecular surface

4.12.1 Electrostatic potential analysis on phenol molecular surface

Below I will introduce quantitative analysis of molecular surface by case study of phenol. The theoretical basis has been documented in Section 3.15.1 and thus will not be repeated here. In this section we only analyze electrostatic potential (ESP) on phenol vdW surface, in next section we will then analyze average local ionization energy on the phenol surface.

Boot up Multiwfn and input following commands

examples\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 Hartree-Fock completely ignored electron Coulomb correlation and hence results in 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 information are printed, ordinary users do not need to concern them. Below results will be printed on screen once the calculation has been finally finished:

```
Global surface minimum: -0.041203 a.u. at 1.455097 3.343708 -0.007902 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	-0.03045989	-0.828856	-19.112362	0.192185 -0.985412 1.877656
*	3 -0.04120321	-1.121196	-25.853368	1.455097 3.343708 -0.007902

Number of surface maxima: 5

#	a.u.	ev	kcal/mol	x/Y/Z coordinate(Angstrom)
1	0.02275520	0.619200	14.277975	-3.344441 -2.281045 0.047286
*	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	0.01583741	0.430958	9.937340	3.419218 1.225375 -0.019326

===== Summary of surface analysis =====

volume: 835.71041 Bohr^3 (123.83953 Angstrom^3)

Overall surface area: 476.05682 Bohr^2 (133.30951 Angstrom^2)

Positive surface area: 231.09186 Bohr^2 (64.71232 Angstrom^2)

Negative surface area: 244.96497 Bohr^2 (68.59719 Angstrom^2)

Overall average value: -0.00020233 a.u. (-0.12695332 kcal/mol)

Positive average value: 0.01877643 a.u. (11.78145591 kcal/mol)

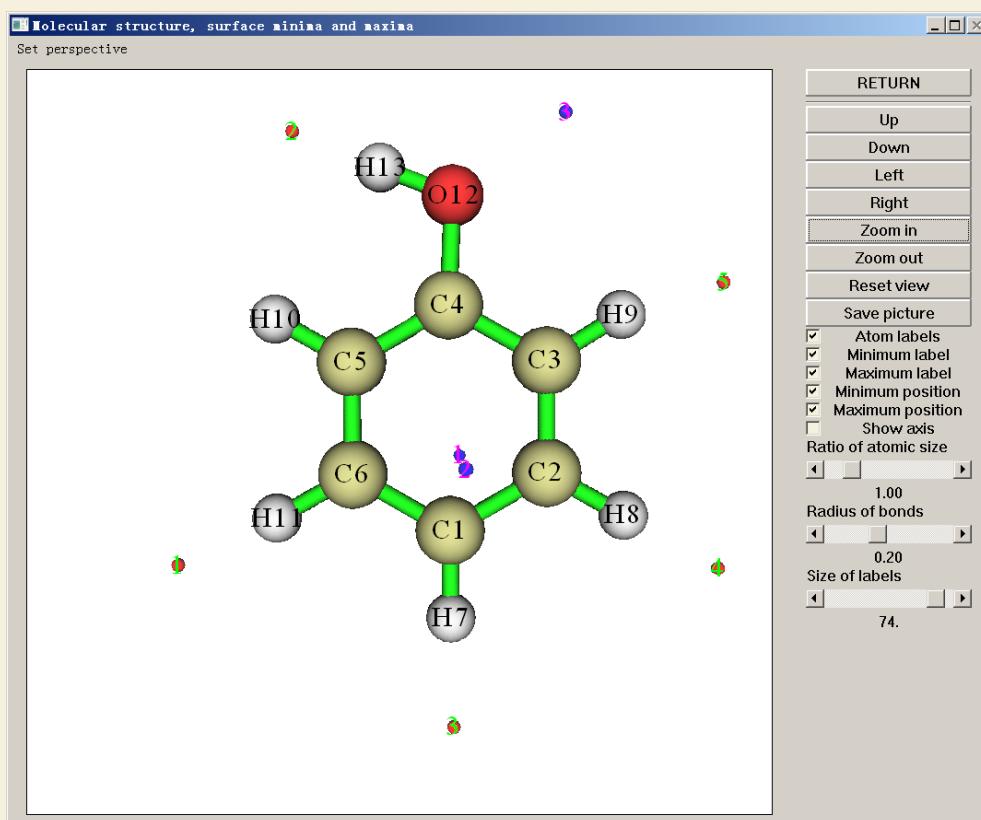
Negative average value: -0.01810626 a.u. (-11.36095315 kcal/mol)

```

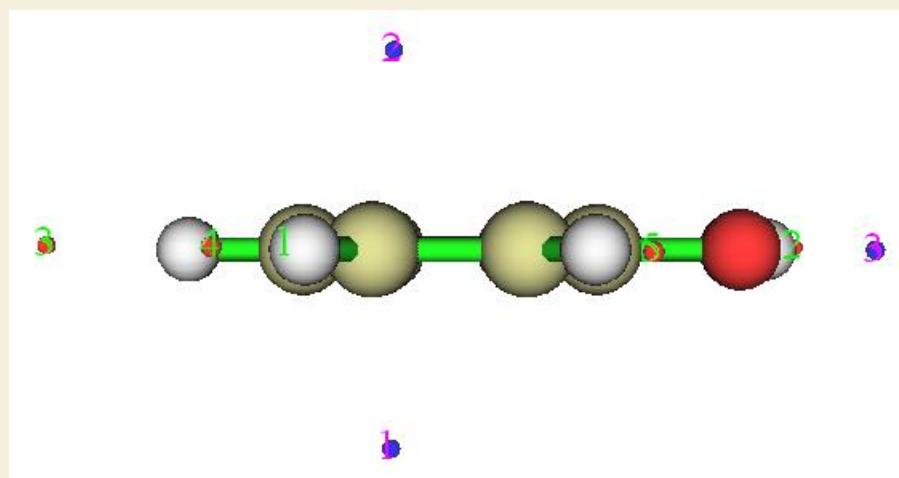
Overall variance (sigma^2_tot): 0.00041488 a.u.^2 ( 163.34165024 (kcal/mol)^2)
Positive variance:      0.00031106 a.u.^2 ( 122.46642148 (kcal/mol)^2)
Negative variance:     0.00010382 a.u.^2 ( 40.87522876 (kcal/mol)^2)
Balance of charges (nu): 0.18762182
Product of sigma^2_tot and nu: 0.00007784 a.u.^2 ( 30.6464578 (kcal/mol)^2)
Internal charge separation (Pi): 0.01842642 a.u. ( 11.56183883 kcal/mol)

```

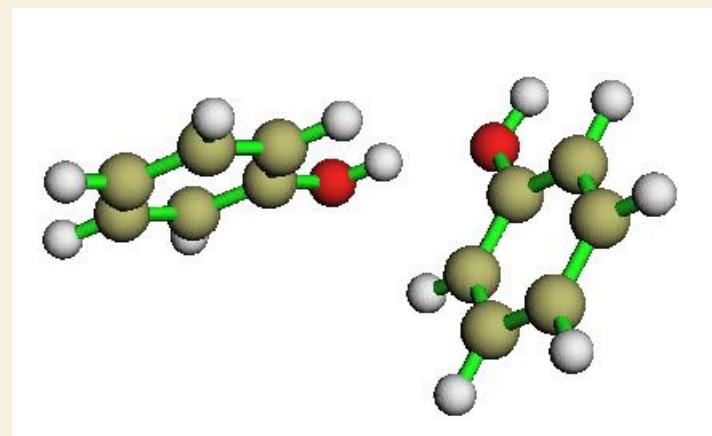
Above information includes all quantities that involving ESP mentioned in Section 3.15.1. In Multiwfn, the vdW volume can also be computed by Monte Carlo method through subfunction 3 of main function 100, see Section 3.100.3, the results match fairly well. Now select option 0 in the post-process interface to view molecular structure and surface extrema (red and blue spheres correspond to maxima and minima, respectively):



In side view:



Minimum 3 (-25.85kcal/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 arising 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 expect 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 electrophiles always prefer to attack the atom which has very low ESP around it, so C1 should be an ideal reaction site for electrophilic reaction. This conclusion is partially consistent with the general knowledge that hydroxyl is an ortho-para directing group. However, although the global minimum is closest to O12, O12 is not the electrophilic reaction site; this contradiction reveals the inherent limitation of ESP analysis method.

Note: Since the molecule has Cs symmetry, in principle, minimum 1 and 2 should have identical X and Y coordinates. However, this cannot be exactly fulfilled in numerical process, because the points scattered on molecular surface do not have the molecular symmetry, see Section 3.15.1 for detail. So X and Y coordinates of minimum 1 and 2 are slightly deviated to each other. If you want to refine the result, choose option 3 "Spacing of grid data for generating molecular surface" and input a smaller value than default value. Smaller spacing of grid points yields more accurate result, but bring higher computational burden.

The non-bonded radius defined in the framework of atoms in molecule (AIM) theory is the shortest distance between a nucleus and $\rho=0.001$ isosurface. Let us calculate the non-bonded radius for O12 and H13. Select option 10 in the post-process interface and input 12, we can see that the non-bonded radius of O12 is 1.701 Angstrom. Select 10 and input 13, the non-bonded radius of H13 is 1.172 Angstrom. In the phenol dimer, the H---O of the H-bond is 1.937 Angstrom, therefore the so-called mutual penetration distance is $1.701+1.172-1.937=0.936$ Angstrom. This is a nontrivial value, indicating the H-bond is strong.

As the final part of ESP analysis, we examine the molecular surface area in each ESP range, this is useful to quantitatively discuss ESP distribution on the whole molecular surface. We choose option 9 in the post-process interface, and then input:

all // All atoms are taken into the statistics (alternatively, if you input for example 2-4, then only the local surfaces corresponding to atom 2, 3 and 4 will be taken into account, see Section 4.12.3 for illustration of the concept of local molecular surface)

-30,55 // The ESP range you are interested in. Since we have already known that the minimum and the maximum ESP on the surface are -25.85 and 53.81 kcal/mol respectively, here we input a slightly larger range to enclose them

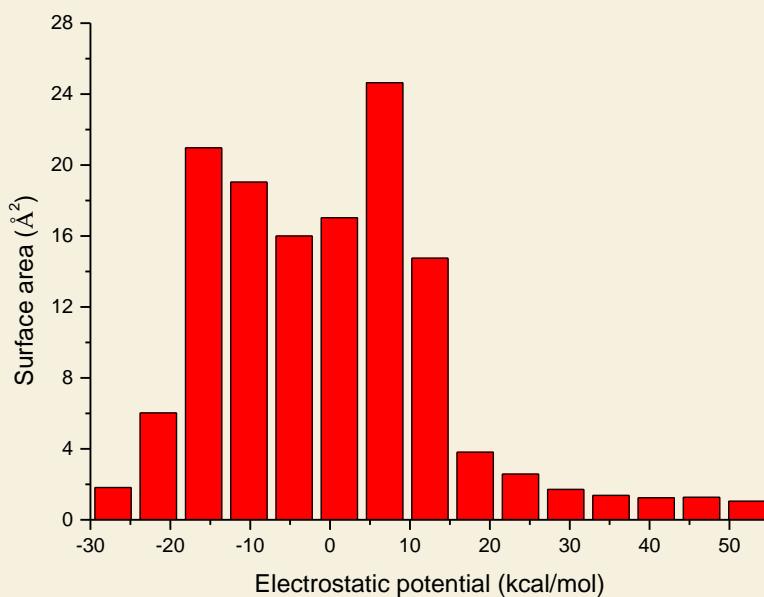
15 // The number of intervals

3 // Both the inputted and outputted units are in kcal/mol

Then you will see surface area (in \AA^2) and corresponding percentage in the whole surface area in each successive ESP range.

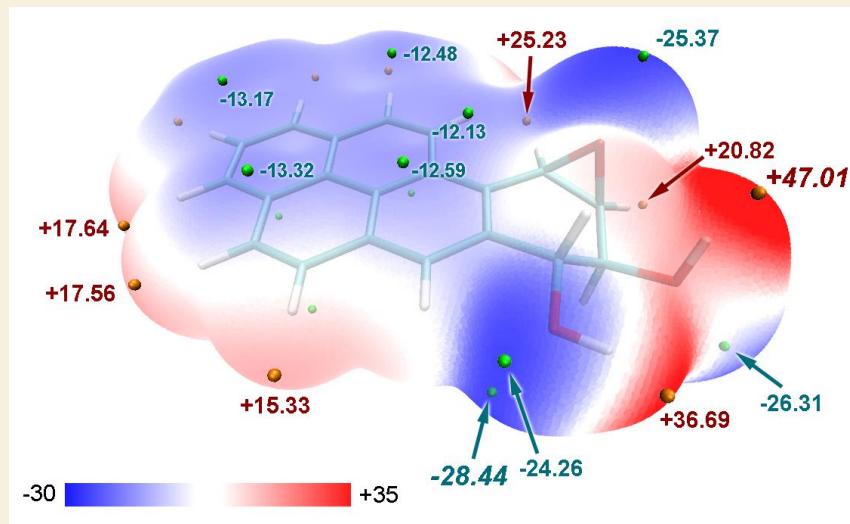
Begin	End	Center	Area	%
-30.0000	-24.3333	-27.1667	1.8192	1.3647
-24.3333	-18.6667	-21.5000	6.0284	4.5221
-18.6667	-13.0000	-15.8333	20.9732	15.7327
-13.0000	-7.3333	-10.1667	19.0390	14.2818
...				
43.6667	49.3333	46.5000	1.2690	0.9519
49.3333	55.0000	52.1667	1.0457	0.7844
Sum:			133.3095	100.0000

By using these data you can use your favourite program to draw a histogram graph. For example, we choose "center" column as X-axis and "Area" column as Y-axis to plot below graph



From the graph it can be seen that there is a large portion of molecular surface having small ESP value, namely from -20 to 20 kcal/mol. Among these areas, the negative part mainly corresponds to the surface above and below the six-membered ring and shows the effect of the abundant π -electron cloud; the positive part mainly arises from the positive charged C-H hydrogens; the near-neutral part represents the border area between the negative and positive parts. There are also small areas having remarkable positive and negative ESP value, corresponding to the regions closed to the global ESP minimum and maximum, respectively.

With the help of VMD program, one can plot very nice color-filled molecular surface map with surface extrema for various real space functions based on the output of Multiwfn. Below is such a plot for ESP, which was presented in my study of benzoapyrene diol epoxide, see *Struct. Chem.*, **25**, 1521 (2014). In which blue, white and red correspond to ESP varying from -30 to 35 kcal/mol, the green and orange spheres correspond to ESP surface minima and maxima, respectively



If you would like to plot similar graphs, please download and follow this tutorial "Plotting

electrostatic potential colored molecular surface map with ESP surface extrema via Multiwfn and VMD" (<http://sobereva.com/multiwfn/res/plotESPSurf.pdf>). However, there are a lot of steps in this tutorial, if you want to draw a map with similar effect but in a much simpler way, see Section 4.A.13. This section also illustrates how to plot penetration map of van der Waals surface of different molecules, which is useful for discussing intermolecular interactions.

Important note: The calculation speed of ESP of cubegen utility in Gaussian package is much faster than Multiwfn. If you have Gaussian installed on your system, it is **strongly** recommended to set *cubegenpath* parameter in settings.ini file to actual path of cubegen, so that cubegen could be automatically invoked by Multiwfn to evaluate ESP during the quantitative molecular surface analysis. Please check Section 5.7 for detail.

Trick: Reuse data of mapped function generated during previous analysis

Here I introduce a trick. Maybe you have noticed that evaluation of ESP on vdW surface is time-consuming, especially for large system with high-quality basis set. If you have performed ESP analysis on a system, which will be analyzed again later, in fact you can export the ESP data to a plain text file, so that next time when you analyze the same system you can directly make use of the exported data. For other type of mapped functions, this trick also works.

Let us see an example. We first perform ESP analysis on vdW surface as usual, input below commands:

```
examples\N-phenylpyrrole.fch
12 // Quantitative molecular surface analysis
0 // Start the analysis
```

Once the calculation is finished, select option 7 to export the surface vertices with ESP values to a plain text file named *vtx.txt* in current folder. After that select -1 to return to last menu.

Assume that we want to perform the analysis again. This time we can directly use the ESP data recorded in the plain text file. Input below commands

```
5 // Load mapped function values from external file during analysis
1 // Load mapped function at all surface vertices from a plain text file
0 // Start the analysis
```

Once construction of molecular surface is complete, Multiwfn will prompt you to input the path of the plain text file recording mapped function values at all surface vertices, at this point you should simply input *vtx.txt*.

Since this time the mapped function values, namely ESP values, are not calculated but loaded from *vtx.txt* directly, the analysis results immediately show up on the screen.

4.12.2 Average local ionization energy analysis on phenol molecular surface

Below we will analyze average local ionization energy \bar{I} on phenol vdW surface. Boot up Multiwfn and input

```
examples\phenol_DFT.wfn
```

12 // Quantitative molecular surface analysis

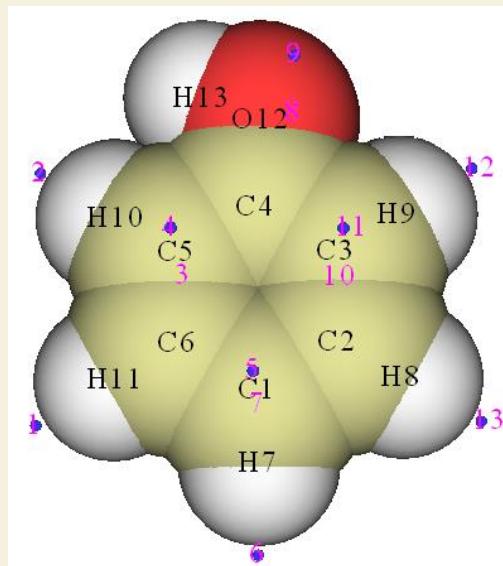
2 // Reselect mapped function

2 // Choose \bar{I} as mapped function

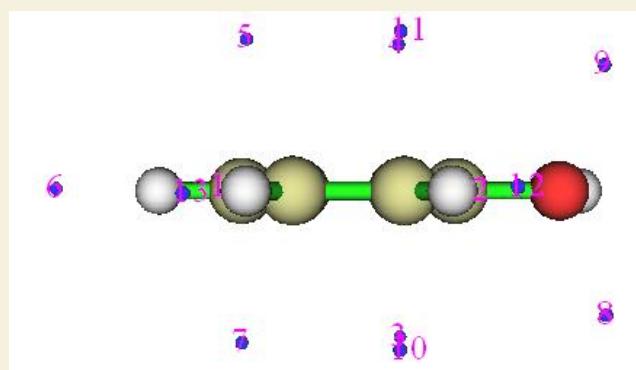
0 // Start the surface analysis.

Since calculation of \bar{I} is much simpler than ESP, the calculation is finished rapidly. Unlike surface analysis for ESP, at this time only vdW volume, surface area, average and variance of \bar{I} on vdW surface are outputted alongside extrema information.

Choose 0 to visualize extrema. In order to make the correspondence between extrema and atoms clearer, we drag the "Ratio of atomic size" scale bar to 4.0, which corresponds to vdW surface, and we disable showing of surface maxima, then we will see:



In side view

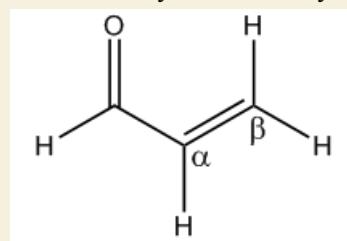


Low value of \bar{I} suggests that the electron at this position is not tightly bounded, the site with lowest \bar{I} on vdW surface is usually recognized as the most vulnerable site to electrophilic attack or to free radical attack. All sites that highly polarizable such as π electron and lone pair regions commonly have corresponding surface minima of \bar{I} . In present instance, minimum 8 and 9

correspond to lone pair of O12, from the output on screen we can find both their \bar{I} values are 10.59eV. Minimum 4,5,11, and 3,7,10 correspond to π electrons, \bar{I} values of all of them are about 8.9eV and can be viewed as degenerate 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 an ortho-para directing group. Since \bar{I} at minimum 8 and 9 are obviously larger than the \bar{I} at the minima around the carbon ring, oxygen should not be vulnerable site of electrophilic reaction.

4.12.3 Local molecular surface analysis for acrolein

It is well known that acrolein (see below) tends to undergo nucleophilic attack at carbonyl carbon and β carbon; in particular, the former is the primary site for hard nucleophilic reagent. The so-called hard means the electron cloud of the nucleophilic reagent is difficult to be polarized; the selectivity of reactive site for this case is usually dominated by ESP.



In this example, we will try to interpret the site-selectivity of acrolein by analyzing ESP on its vdW surface. Note that average local ionization energy is only useful for studying electrophilic attack, but completely useless for analyzing nucleophilic attack.

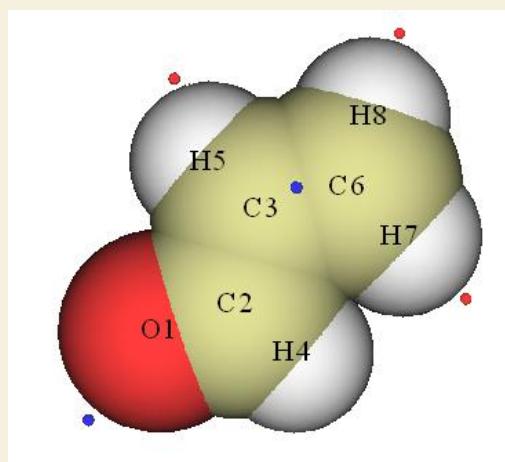
Boot up Multiwfn and input:

```
examples\acrolein.wfn // Optimized and produced at B3LYP/6-31G** level
```

```
12 // Quantitative analysis of molecular surface
```

```
0 // Start the analysis for ESP
```

After the calculation is finished, choose 0 to visualize surface extrema:



As you can see, there is a surface minimum of ESP at the boundary of α carbon and it is very close to β carbon. This observation indirectly reveals that nuclear charge of α carbon is more heavily screened by electron cloud, and hence is less probable to be the site of nucleophilic attack. However, quantitative analysis of ESP on the whole acrolein surface does not provide a direct and definitive interpretation on the preference of reactive sites, because no surface maxima are found on carbonyl and β carbons, hence we are unable to directly investigate the characteristic of carbonyl and β carbon.

In Multiwfn, the quantitative analysis can not only be applied to the whole molecular surface, but is also applicable to local molecular surface to reveal characteristic of atom or fragment. Here we select option 11 in post-process interface to calculate and output properties of the local surfaces corresponding to each atom. Some of the results are shown below

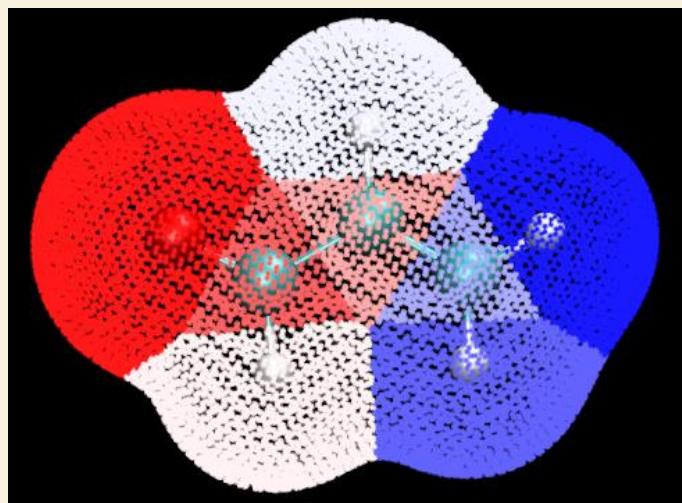
Note: Average and variance below are in kcal/mol and (kcal/mol)² respectively

Atom#	All/Positive/Negative average		All/Positive/Negative variance			
1	-24.35251	NaN	-24.35251	NaN	NaN	72.72766
2	4.65672	5.74594	-1.45401	9.79575	9.01495	0.78080
3	1.30965	2.36405	-0.88391	2.70896	2.45972	0.24925
4	8.37174	10.35813	-6.00120	36.74187	17.22299	19.51888
5	7.07040	8.67973	-6.35468	49.99108	25.33899	24.65209
6	2.21578	3.02322	-0.73880	5.15593	4.95305	0.20288
7	15.34251	15.34251	NaN	NaN	22.85377	NaN
8	14.68486	14.68486	NaN	NaN	25.92230	NaN

As you can see, the average ESP values on the local surface of carbonyl carbon (atom 2), α carbon and β carbon are 4.657, 1.310 and 2.216 kcal/mol, respectively. This result clearly explained the site-selectivity; the carbonyl carbon is the most favorable site because the average ESP on its local surface is the most positive, and hence nucleophilic reagents (especially the hard ones) tend to be attracted to this site. In contrast, on the local surface of α carbon the average of ESP is the smallest compared to the other two carbons, and thus α carbon has less capacity to attract nucleophilic reagents.

Note that some of the outputted data are NaN (Not a Number), these are not bug but understandable. For example, the average of positive part of ESP of atom 1 is NaN, this is because oxygen has large electronegativity, and thus on the local surface of atom 1 the ESP is completely negative, so the average of positive ESP is unable to be computed.

If you are confused about what is "local surface of atoms" or you want to visualize them, after you choose option 11 you can select "y" to output the surface facets to locsurf.pdb file in current folder. Each atom in this file corresponds to a surface facet, the B-factor value corresponds to its attribution. By this file you can visualize how the whole molecular is partitioned, the method is: boot up VMD program and drag the pdb file into the VMD main window, in "Graphics"- "Representation" set the "Drawing method" as "Points", set the point size to 4, and set "Coloring Method" as "Beta". In VMD main window select "Display"- "Orthographic" and deselect "Display"- "Depth Cueing". Then load the molecular structure file of acrolein into VMD and render it as CPK mode, you will see below graph

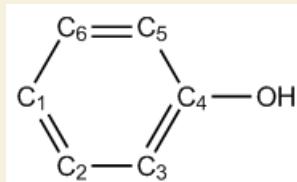


In the graph, each point represents a surface facet; different colors represent different local surface regions, and each one corresponds to an atom.

Beware that the "imolsurparmode" parameter in settings.ini directly affect the result of local surface analysis, currently we are using imolsurparmode=1.

4.12.4 Fukui function distribution on local molecular surface of phenol

I have exemplified how to study Fukui function by visualizing its isosurface (Section 4.5.4) and by condensing it to atomic value via population analysis (Section 4.7.3). In this section, I will introduce another way to study this function, namely examining its average value on local molecular surface corresponding to each atom. The advantage of this scheme over simply visualizing isosurface is that the relative magnitude of Fukui function can be obtained clearly and quantitatively. Meanwhile, unlike condensed Fukui function, this scheme is free of ambiguity of choice of population methods. I will still use phenol as example.



Boot up Multiwfn (referred to as Multiwfn A) and input following commands

examples\phenol.wfn

12

2 // Select the mapped real space function on the molecular surface

0 // The function value will be loaded from an external file

I // Set the way to define the surface

l // Use electron density isosurface as molecular surface

0.01 // Because Fukui function on default isosurface $\rho=0.001$ is often too small, enlarging the isovalue to 0.01 makes the discussion easier

θ // Start the surface analysis

Multiwfn will generate grid data of electron density and then generate the surface vertices. After the coordinate of these vertices are automatically outputted to surfptpos.txt in current folder, Multiwfn A pauses. Do not terminate Multiwfn A, we boot up another Multiwfn now (referred to as

Multiwfn B), and then input below commands in Multiwfn B

```
examples\phenol.wfn
5 // We use this module to generate Fukui function on the points recorded in surfptpos.txt
0 // Set custom operation
1
-,examples\phenol+1.wfn // Subtract the properties from phenol+1.wfn from phenol.wfn
1 // Electron density
100 // Load the coordinate of the points to be calculated from an external file
surfptpos.txt
.t.txt // Output the coordinate and calculated function values (Fukui function) of the points
(surface vertices) to this file
```

Now we terminate Multiwfn B, and return to Multiwfn A, then input

```
.txt // Load the Fukui function values at the surface vertices from this file
11 // Output the quantitative data of Fukui function distributed on the local vdW surface
corresponding to each atom
```

You can find below content from the output

Atom#	All/Positive/Negative average	
1	1.93967E-03	1.93967E-03
2	8.13794E-04	8.13794E-04
3	1.41595E-03	1.41595E-03
4	2.01898E-03	2.01898E-03
5	1.25625E-03	1.25625E-03
6	9.39694E-04	9.39694E-04
... [ignored]		

NaN means there is no negative value of Fukui function on the local molecular surfaces. From the result it is clear that the average of the Fukui function on the local molecular surface corresponding to *ortho* and *para* carbons are larger than that of *meta* carbons, this observation correctly reflects the fact that hydroxyl is an ortho-*para*- director.

4.12.5 Becke surface analysis on guanine-cytosine base pair

The concept of Hirshfeld and Becke surface analysis have been detailedly introduced in Section 3.15.5, please read them first. In this section I will exemplify how to perform Becke surface analysis on guanine-cytosine (GC) base pair to analyze the weak interaction between the two monomers.

Boot up Multiwfn and input

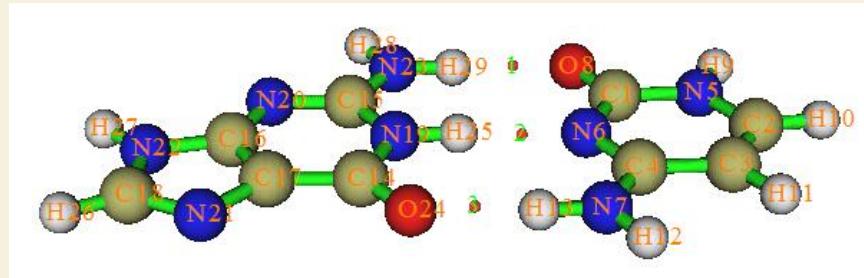
```
examples\GC.wfn // Generated at M06-2X/6-31+G** level, optimized at PM7 level
12
1 // Change the definition of surface
6 // Use Becke surface. You can also select 5 to use Hirshfeld surface
1-13 // The index range of the atoms you are interested in (cytosine in present case)
2 // Change mapped function
11 // Electron density
0 // Start calculation
```

Multiwfn found numerous surface minima, which are meaningless in this case, and at the same

time three surface maxima are found

Number of surface maxima:		3
#	value	x/Y/z coordinate(Angstrom)
1	0.042331	-0.721650 2.605906 0.043754
2	0.049281	0.767524 0.959671 -0.013820
*	3	1.628343 -1.269364 0.030387

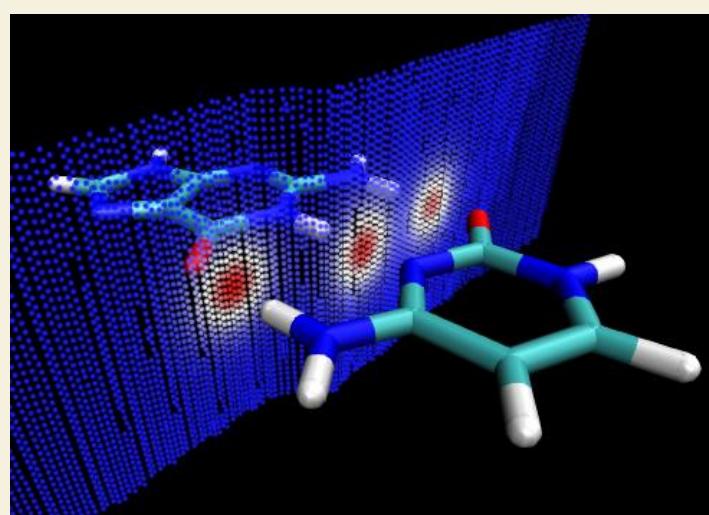
You can choose 0 to visualize them, see below



Since the sequence of electron density at these maxima is $3 \geq 2 > 1$, one can expect that the sequence of H-bond strength is $O24-H13 \geq H25-N6 \geq H29-O8$. This conclusion is identical to the analysis of AIM bond critical point analysis.

If you want to visualize the shape of Becke surface, simply choose option -3. If you want to plot the Becke surface colored by mapped function value, you need to make use of VMD, as described below. Select option 6 and input 100 to export all surface vertices to vtx.pdb in current folder and in the meantime multiply the data by 100 (since electron density in the Becke surface is small, while the B-factor in pdb format only has two decimal places). Each atom in vtx.pdb now corresponds to a surface vertex, and its B-factor stands for electron density at corresponding position.

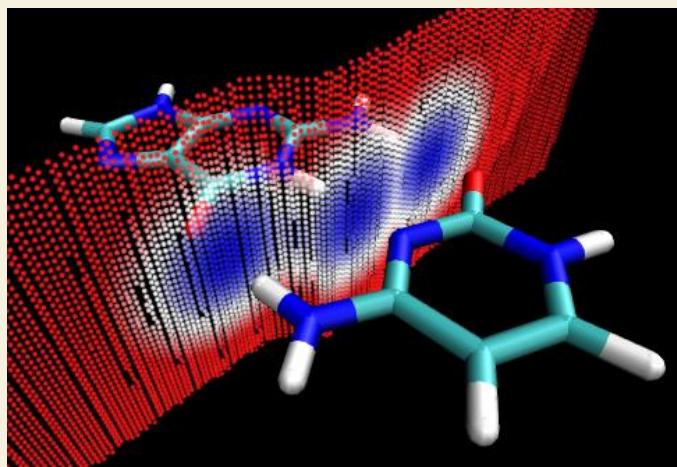
Drag examples\GC.pdb into main window of VMD program, this file contains geometry of present system. Select "Graphics"- "Representation", change the drawing method to "Licorice" and decrease bond radius to 0.2. Then drag the vtx.pdb into VMD main window, select "Graphics"- "Representation", change the drawing method from "Lines" to "Points", set coloring method to "Beta", enlarge the point size to 6, run command *color scale method BWR* in VMD console window. Now you should see



Becke surface is represented by points, and the three red zones correspond to high electron density region, which stem from H-bonds. This example demonstrated that Becke surface analysis

is useful to reveal the region where intermolecular interaction is evident.

Repeat above analysis but with d_{norm} as mapped function, and replot the Becke surface with the same procedure. Then in the graphical representation window, select "Trajectory" tab, input -0.5 and 2.0 in the "Color scale data range", you will see below graph

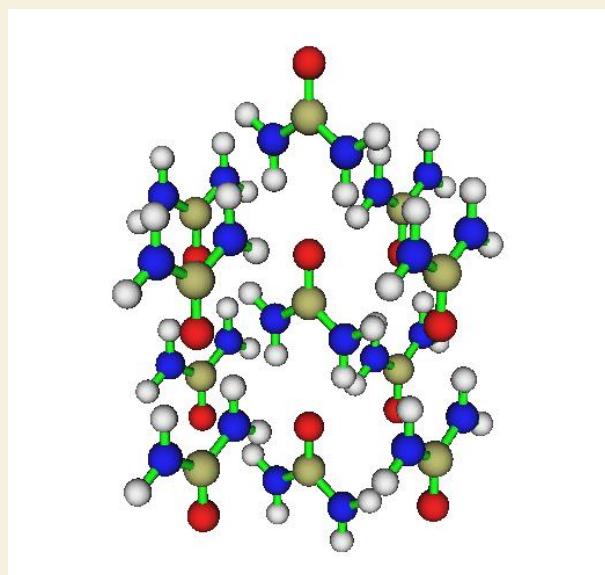


As you can see, using d_{norm} as mapped function is also capable of revealing weak interactions; the blue regions corresponds to close contact between the fragment you are interested in (cytosine) and other fragments (guanine only). The advantage of using d_{norm} over electron density is that the former does not rely on wavefunction information, so the input file you used can only contain geometry information, such as .pdb and .xyz are completely acceptable in this case. However the physical meaning and robustness of d_{norm} is not as good as electron density.

In fact, present analysis can also be realized via the Hirshfeld surface analysis, and the computational cost is lower (especially when the number of atoms is high).

4.12.6 Hirshfeld surface analysis and fingerprint plot analysis on urea crystal

This time we perform Hirshfeld surface analysis for urea crystal. .cif is the most popular crystal structure format, however currently Multiwfn does not support it. So I dug out a cluster of ureas from its crystal structure; the file *examples\Urea_crystal.pdb* contains 11 ureas, the central one will be defined as the fragment in our Hirshfeld surface analysis.



Notice that Hirshfeld surface analysis requires electron density of atoms in their free-states, in Multiwfn there are two ways to generate them: (1). Using built-in atomic densities, this is very convenient, see Appendix 3 for detail; (2). Evaluating atomic densities based on atomic .wfn files, see Section 3.7.3 for detail. In this example we will use way 2 (way 1 works equally well and is more convenient). What we need to do is simply moving the "atomwfn" folder in "examples" folder to current folder. Then during generating Hirshfeld surface, Multiwfn will automatically use the atomic wavefunction files in this folder.

Boot up Multiwfn and input

examples\Urea_crystal.pdb

12 // Quantitative molecular surface analysis

1 // Change surface type

5 // Use Hirshfeld surface

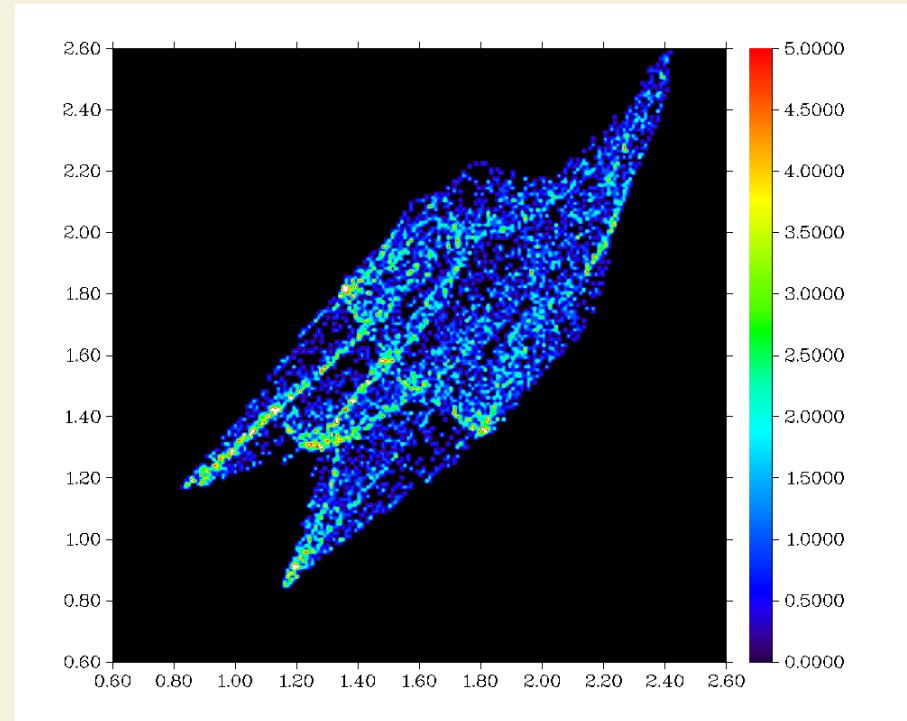
16,36,58,2,77,55,34,13 // The index of the atoms in the central urea

0 // Start calculation. Note that the default mapped function d_{norm} is used here

2 // Evaluating atomic densities based on atomic .wfn files

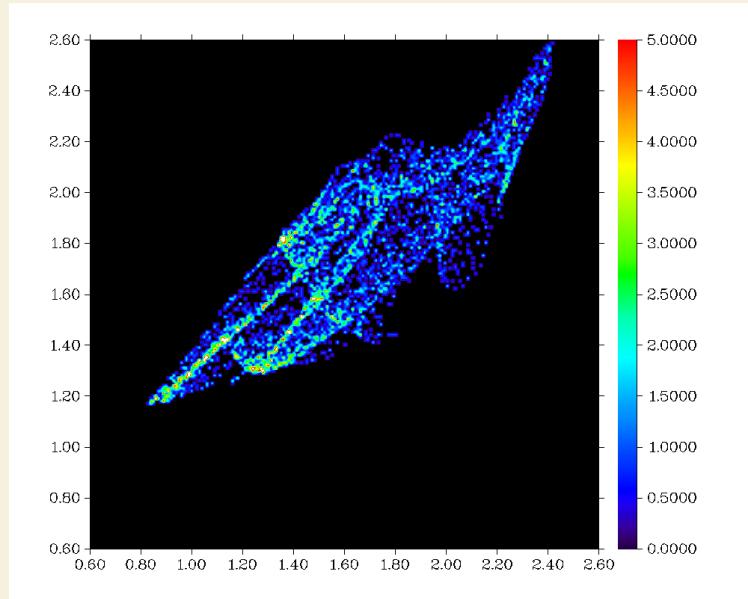
After the calculation is finished, select option 6 to export the surface vertices with d_{norm} value to *vtx.pdb*, and then plot them in VMD by using the method shown in the last section, you will see a graph very similar to the second figure in Section 3.15.5.

Next, we draw fingerprint plot. In the post-process menu select 20 and choose option 0, then choose option 0 again, you will see below graph



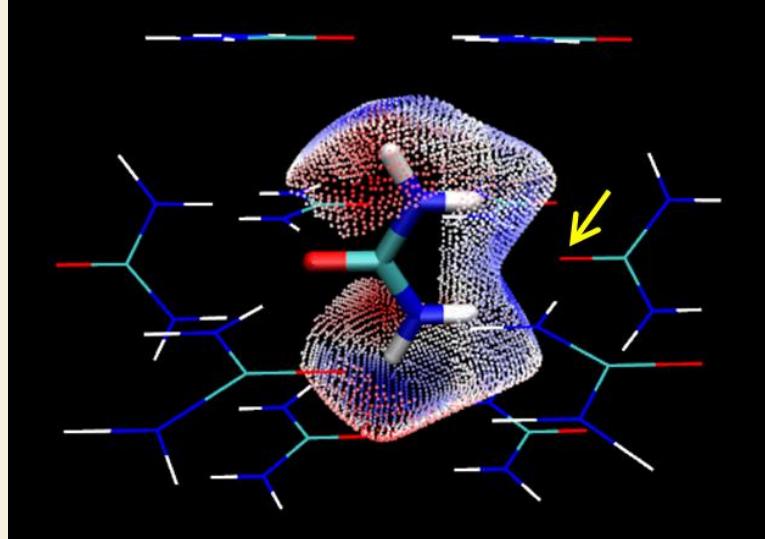
In this figure, X and Y axes correspond to d_i and d_e , respectively. It can be seen that there is a pair of spikes at the bottom left of the plot (*i.e.* short d_i and d_e), this observation suggests that urea behaves as both H-bond acceptor (the lower spike, $d_i > d_e$) and H-bond donor (the upper spike, $d_i < d_e$).

In Multiwfn, the fingerprint plot can be drawn not only for overall Hirshfeld surface, but also for local contact surface (see Section 3.15.5 for detail). Let's check the fingerprint plot of the local contact surface between the four hydrogens in central urea and all of the atoms in peripheral ureas. After closing the window showing fingerprint plot, select -1 to return, then input 20 to enter the interface of fingerprint plot analysis again. We need to first define "inside atoms" set and "outside atoms" set, because only the points on the contact surface between the two sets will be taken into account in the fingerprint plot analysis. Choose option 1 to define "inside atoms" set, then press ENTER button directly to use default atom range (*i.e.* all atoms in the central urea) and then input *H*. Now all hydrogens in the central urea are defined as the "inside atoms" set. Since the default "outside atoms" set is just all of the atoms in peripheral ureas, we do not need to modify it. Select option 0 to start the fingerprint plot analysis, then choose 0, you will see



As you can see, since this time we only considered the hydrogens in the central urea, the urea purely behaves as hydrogen donor, so only one spike at bottom left of the plot is observed

It is interesting to check the shape of the local contact surface, so after closing the window we choose option 5 to export all of the points on the surface to *finger.pdb* in current folder. Plot them in VMD by using the method shown in the last section, you will see



Clearly, the surface very well portrays the contact between the hydrogens in central urea and the atoms in peripheral ureas. The color corresponds to d_{norm} value, more blue implies more close contact.

Next, we check the fingerprint plot between the hydrogens in central urea and the oxygen marked by yellow arrow in above figure. Input below commands

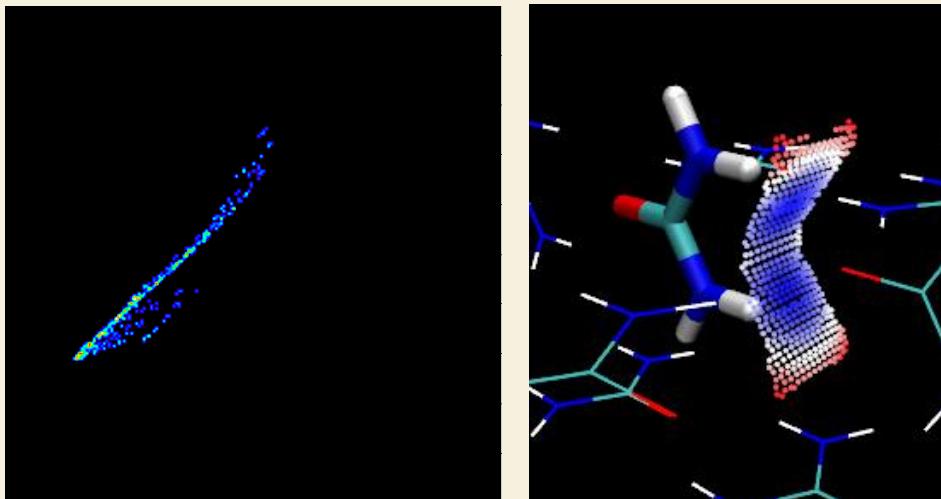
```
-I // Return
20 // Fingerplot analysis
1 // Set inside atoms set
[Press ENTER directly]
H
2 // Set outside atoms set
76 // The index of the oxygen marked by yellow arrow
```

```
[Press ENTER directly] // Namely do not use element type as filter condition
0 // Start fingerprint analysis
3 // Set color scale. Because the default color scale in this case is different to above
fingerprints, the plots are not directly comparable
```

0,5

0 // Show fingerprint graph on screen

The fingerprint plot and the corresponding surface points are shown below



In the fingerprint plot you can see that the distribution scope of surface points is narrow, and the spike is quite evident, showing strong H-bond character due to the contact of H and O.

Fingerprint is especially useful for comparison of the intermolecular interactions in different crystals, see *CrystEngComm*, **11**, 19 (2009) for discussions.

Using VMD to plot color-mapped isosurface of Hirshfeld/Becke surface

Here I describe how to easily plot very pretty Hirshfeld (or Becke) surface mapped by electron density with promolecular approximation, this map looks much better than those shown above. Urea cluster is still taken as example.

Boot up Multiwfn and input

examples\Urea_crystal.pdb

12 // Quantitative molecular surface analysis

1 // Change surface type

5 // Use Hirshfeld surface

16,36,58,2,77,55,34,13 // The index of the atoms in the central urea

2 // Select mapped function

11 // Electron density under promolecular approximation (namely the density of the system is evaluated as superposition of built-in atomic densities in their free-states)

0 // Start calculation

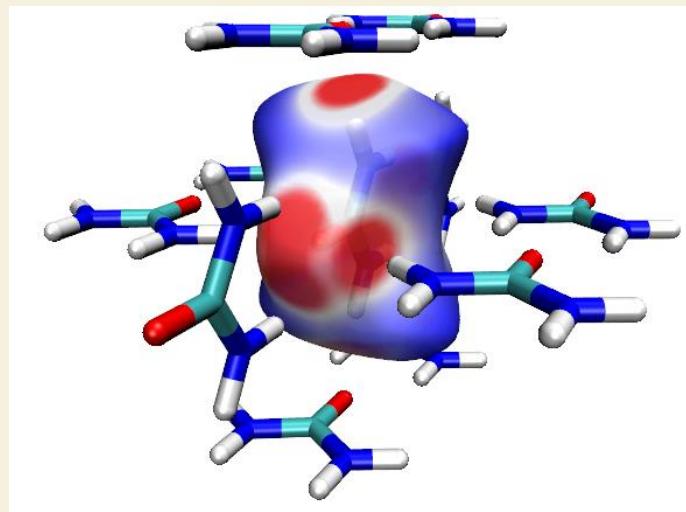
1 // Use built-in atomic densities to construct Hirshfeld surface

-2 // Export the grid data used to define Hirshfeld surface as *surf.cub* in current folder

13 // Calculate grid data of mapped function and export it to *mapfunc.cub* in current folder

Now you have *surf.cub* and *mapfunc.cub* in current folder, move them to VMD folder. Then copy the *examples\scripts\hirsh_rho.vmd* into the VMD folder. Boot up VMD, input *source hirsh_rho.vmd* in VMD console window, for present case it is better to also input *material change*

diffuse Translucent 0.8 to make the surface brighter, then you will see below graph. The plotting script sets color transition to Blue-White-Red, which corresponds to electron density varying from 0.0 to 0.015 a.u. Clearly, from the graph one can easily recognize the evident intermolecular interaction regions.



You can manually modify the .vmd plotting script to meet your expectation. Via basically the same procedure, you can also plot d_{norm} mapped Hirshfeld (or Becke) surface, in this case you should use *examples\scripts\hirsh_dnorm.vmd* script.

4.12.7 Predict density of molecular crystal of FOX-7

As introduced in Section 3.15.1, many condensed phase properties of a molecule can be predicted based on the result of quantitative molecular surface analysis of electrostatic potential (ESP). For example, in *Mol. Phys.*, **107**, 2095 (2009), Politzer et al. showed that crystal density of molecules only containing C, H, N, O can be predicted as

$$\rho = \alpha \frac{M}{V_m} + \beta(\nu\sigma_{\text{tot}}^2) + \gamma$$

where $\alpha = 0.9183$, $\beta = 0.0028$ and $\gamma = 0.0443$ when the wavefunction is generated at B3PW91/6-31G** level and the unit of M/V_m and $\nu\sigma_{\text{tot}}^2$ are g/cm³ and (kcal/mol)², respectively.

In this section, I illustrate how to use above formula to predict density of molecular crystal of FOX-7 (1,1-diamino-2,2-dinitroethene), which is an insensitive high explosive compound. More illustrations of property prediction can be found in my blog article "Using Multiwfn to predict crystal density, heat of vaporization, boiling point and solvation free energy" (in Chinese, <http://sobereva.com/337>).

First, we optimize geometry of FOX-7 and yield wavefunction file at B3PW91/6-31G** level, which is the level used by Politzer et al. in their *Mol. Phys.* paper. The resulting FOX-7.wfn has provided as *examples\FOX-7.wfn*.

Boot up Multiwfn and input below commands:

examples\FOX-7.wfn

12 // Quantitative molecular surface analysis

```
0 // Start analysis for default real space function (ESP) on default surface (0.001 a.u.
isosurface of electron density)
```

After a while, you will find below output on screen

```
volume: 942.48700 Bohr^3 ( 139.66220 Angstrom^3)
Estimated density according to mass and volume (M/V): 1.7606 g/cm^3
...[ignored]
Product of sigma^2_tot and miu: 0.00020164 a.u.^2 ( 79.40119 (kcal/mol)^2)
Internal charge separation (Pi): 0.03740373 a.u. ( 23.47121 kcal/mol)
```

From the output, we find that $M/V_m=1.7606 \text{ g/cm}^3$ and $\nu\sigma_{tot}^2 = 79.40119 \text{ (kcal/mol)}^2$, therefore the density could be predicted as $0.9183*1.7606+0.0028*79.40119+0.0443=1.883 \text{ g/cm}^3$. The experimental density of FOX-7 crystal is 1.885 g/cm^3 , which can be found at corresponding wiki page (<https://en.wikipedia.org/wiki/FOX-7>). Clearly, our prediction is extremely successful, the error is merely -0.002 g/cm^3 ! However, the surprisingly good result is fortuitous to a large extent, since according to the test in the *Mol. Phys.* paper, the RMS error using above prediction formula is 0.047 g/cm^3 .

4.12.8 Quantitative analysis of orbital overlap distance function D(r) on thioformic acid molecular surface

Content of this section was contributed by Arshad Mehmood and slightly adapted by Tian Lu.

This example is a continuation of Section 4.5.7. Here I illustrate the quantitative analysis of orbital overlap length function $D(\mathbf{r})$ on molecular electron density isosurface of thioformic acid.

Boot up Multiwfn and input following commands:

```
examples\ThioformicAcid.wfn // Thioformic acid optimized at B3LYP/6-311++G(2d,2p)
12 // Quantitative analysis of molecular surface
2 // Select mapped function
6 // Orbital overlap distance function D(r), which maximizes EDR(r;d) with respect to d
2 // Use default value of total number, start and increment of EDR exponents. Please consult
Section 4.5.7 for more information.
```

0 // Start analysis now!

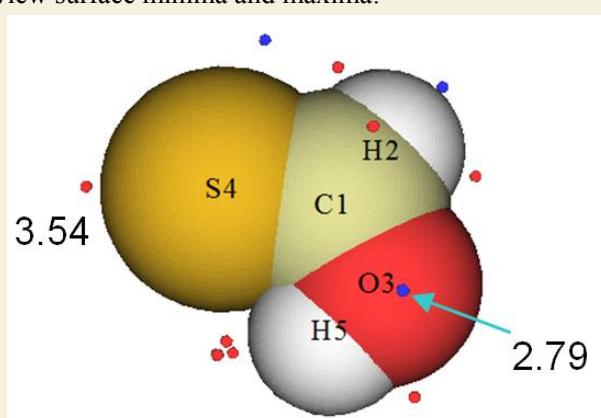
Now the analysis starts. This step will take some time. Once calculation is finished, following results will be printed on screen along with other information:

```
Global surface minimum: 2.789918 a.u. at 1.983402 -0.346198 1.757884 Ang
Global surface maximum: 3.541349 a.u. at -2.861073 -1.074395 -0.095237 Ang
```

#	value	X/Y/Z coordinate(Angstrom)		
1	3.296958	-1.622302	2.055665	0.423383
2	3.218284	0.950838	2.841231	0.010991
*	3.2789918	1.983402	-0.346198	1.757884
4	2.790103	2.078805	-0.344424	-1.731277

```
The number of surface maxima: 10
#      value      X/Y/Z coordinate(Angstrom)
*   1      3.541349  -2.861073 -1.074395 -0.095237
   2      3.401958  -0.539716  2.324441  0.014652
   3      3.502485   0.041422 -2.239222 -0.308278
   4      3.502696   0.089030 -2.257271  0.004859
   5      3.494004   0.207205 -2.075594 -0.673828
   6      3.496480   0.164501 -2.070255  0.707373
   7      3.359381   0.542328  1.465019 -1.650518
   8      3.358980   0.496994  1.456262  1.657761
   9      3.311711   2.030429  1.904600  0.023877
  10     2.918240   2.847314 -1.370280  0.021275
```

Now select 0 to view surface minima and maxima:



This graph shows molecular structure and surface extrema (red and blue spheres correspond to surface maxima and minima, respectively). It can be seen that surface minima is present on oxygen atom due to compact lone pair and surface maxima is located on sulfur atom due to its more diffuse and weakly bound lone pair electrons.

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 DMol3 grid file (.grd) or the grid data directly generated by such as 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 does not 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

Example 1

Assume that we have two cube files A.cub and B.cub, in this example we obtain their difference cube file (*viz.* A.cub minus B.cub).

Boot up Multiwfn and input following commands

```
A.cub // Load the first cube file into memory  
13 // Process grid data  
11 // Grid data calculation  
4 // Subtract the grid data in memory by another grid data
```

B.cub // The cube file containing another grid data. Notice that this cube file must have identical grid setting as the first cube file

Now the grid data in the memory has been updated, choose 0 to export it as a new cube file, which is what we need.

Example 2

Assume that we have two cube files MO1.cub and MO2.cub, each of them records wavefunction value of an orbital. In this example we will generate a cube file containing total electron density deriving from these two orbitals. According to Born's probability interpretation, square of an orbital wavefunction value is simply its density probability, therefore what we need is the sum of square of the two grid data.

Boot up Multiwfn and input following commands

```
MO1.cub
13 // Process grid data
11 // Grid data calculation
10 // Perform A^2+B^2=C operation, where A is present grid data (MO1.cub), B is another
cube file (MO2.cub), C is the new grid data
MO2.cub // Load another cube file
After calculation, the grid data in memory has been updated to C.
0 // Output the updated grid data
totdes.cub // Filename of the new cube file, which contains total electron density of the 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 in local regions

Sometimes we do not want all isosurfaces in the whole space are shown, because too many isosurfaces will confuse our eyes. This section I will show how to screen the isosurfaces of not interest

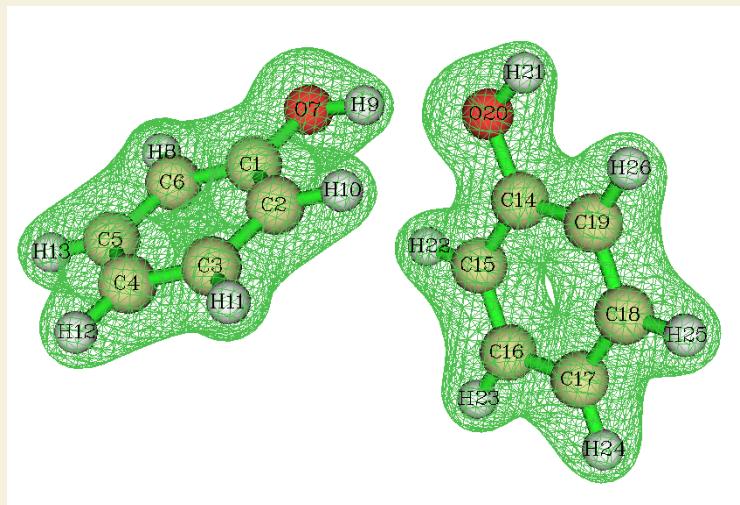
4.13.4.1 Screen isosurfaces inside or outside a region

This section I take electron density of phenol dimer as example. First we generate the grid data as follows (you can also directly load a .cub/.grd file and then enter main function 13)

```
examples\phenoldimer.wfn
```

```
5
1
2
-1 // Visualize isosurface
```

As you can see, the isosurfaces appear on both phenol molecules.



Assume that we only want the isosurface around the right phenol will be shown, we need to set the value of the grid points that close to the left phenol to a very small value, for example, zero.

Close the Multiwfn GUI window and input

```
0 // Return to main menu
13 // Process grid data
13 // Set value of the grid points that far away from / close to some atoms
-0.7 // That means we will set the value of the grid points inside 0.7 times of vdW radius of
      the atoms. If input 0.7, then the value of the grid points outside 0.7 times of the vdW radius will be
      set
0 // Set the value to 0
2 // Defining mode. 2 means inputting atomic indices by hand (if choose 1, external file
containing atomic index list will be used to define fragment, see Section 3.16.9 for the format or the
next example)
1-13 // Range of atomic indices of the left phenol
```

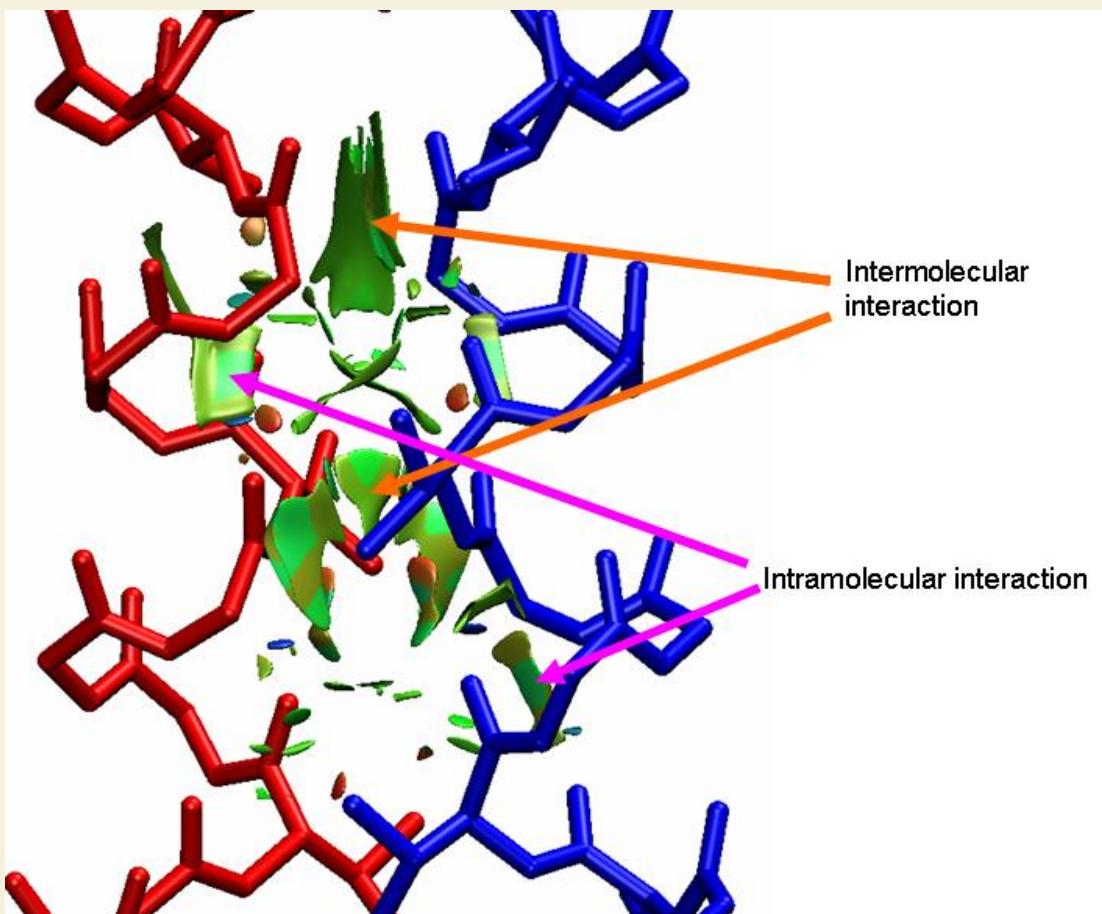
Now the grid data has been updated, let's choose option -2 to visualize the isosurface of current grid data. As you can see, the isosurface of the left phenol has disappeared.

4.13.4.2 Screen isosurfaces outside overlap region of two fragments

During analysis of inter-molecular interaction by NCI method (Section 4.20.1), what we want to study is only the isosurfaces in inter-molecular regions. 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 section 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 corresponding 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 reading Section 4.20.2), red and blue representing backbone structure of the two chains respectively. Isosurfaces of reduced density gradient exhibit weak interaction region. However, these isosurfaces include both intermolecular and intramolecular parts, they are interwinded and result in 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 prepare two atom list files for the two chains (each chain corresponds to a fragment). *atmlist1.txt* includes atom indices of chain 1, the head and tail of the file are:

```

159 <--- Total number of atoms in chain 1
1 <--- Atom index of the first atom in chain 1
2 <--- Atom index of the second atom in chain 1
...
159 <--- Atom index of the last atom in chain 1

```

Similarly, *atmlist2.txt* defines atom list for chain 2, its head and tail parts are:

```

159 <--- Chain 2 has 159 atoms too
160 <--- Atom index of the first atom in chain 2
161 <--- Atom index of the second atom in chain 2
...
318 <--- Atom index of the last atom in chain 2

```

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 try this
value many times to find a proper value
1000 // Set value of those grid points to 1000, this value is large enough

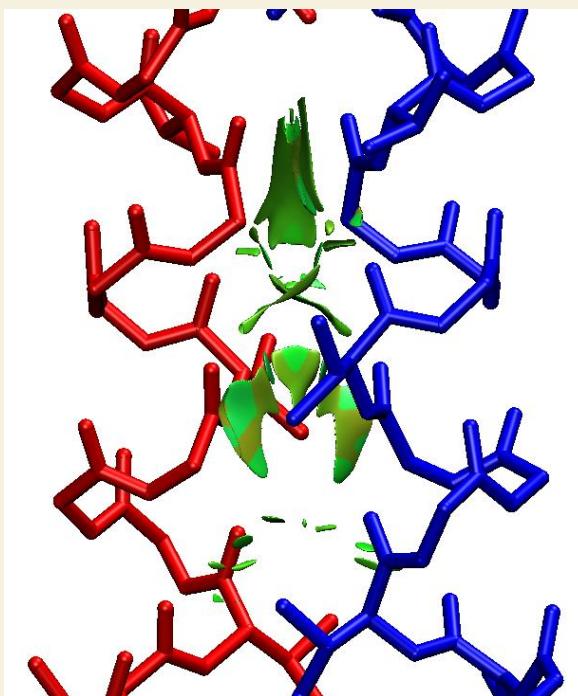
```

```
1 // Defining mode, 1 means using external file to define the fragment
```

```
atmlist1.txt // The name of the atom list file for chain 1
```

```
atmlist2.txt // The name of the atom list file for chain 2
```

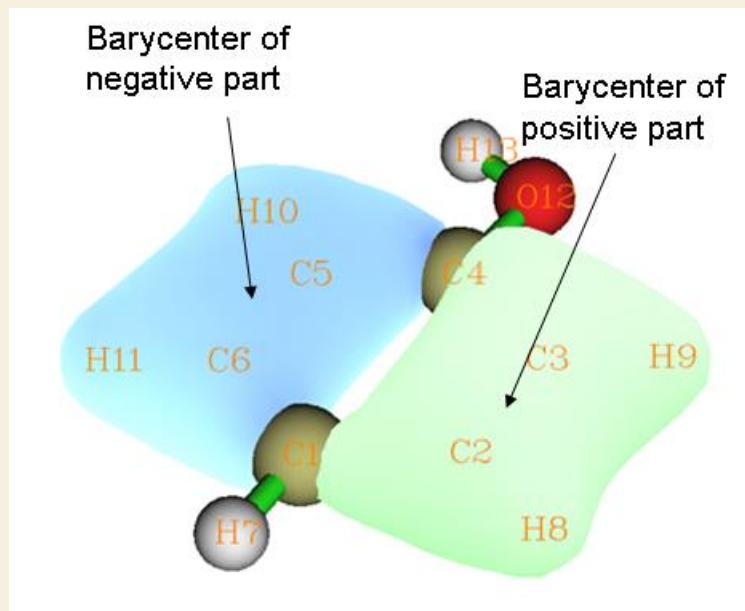
Wait for a while, the grid data will be updated. Then choose function 0 to export it as cube file. Using this new cube file to redraw above picture, we find all of intramolecular isosurfaces have disappeared, the graph becomes very clear.



In fact, when the case is not complicated (as present example), preparing atomic list files are not needed, you can choose defining mode as 2 and then directly input atomic indices (i.e. 1-159 for chain 1 and 160-318 for chain 2).

4.13.5 Acquire barycenter of a molecular orbital

In this example, we will calculate barycenter of a molecular orbital. You can also obtain barycenter of other real space functions by similar manner. The definition of barycenter is given in Section 3.16.13. Below is the isosurface of the 10th MO of phenol.



Before calculate the barycenter of the MO, we need to obtain the grid data of the MO. We can do this in Multiwfn, namely boot up Multiwfn and input following commands:

```
examples/phenol.wfn
5 // Calculate grid data
4 // Choose orbital wavefunction
10 // The 10th orbital
2 // Medium quality grid. Finer quality of grid will give rise to more accurate barycenter position
0 // Return back to main menu
```

Now the grid data has been stored in memory, we will analyze it now

```
13 // Process grid data
17 // Show statistic data
1 // Select all points
```

From the output, we can find that the X, Y, Z components of barycenter of the positive part of the MO are (in Bohr) 2.612, -0.524, 0.000 respectively, while that of the negative part are -2.586, -0.710, 0.000. The total barycenter is meaningless currently, since total integral value is zero for this MO. However, total barycenter of absolute value of the MO is useful, especially for macromolecules, from this we can understand where the MO is mainly located. In order to do this, we input:

```
11 // Grid data calculation
13 // Get absolute value
17 // Show statistic data
1 // Select all points
```

We find X, Y, Z of total barycenter of the MO are -0.015, -0.618, 0.000. Since there is no negative region now, barycenter of negative part is shown as NaN (Not a Number).

4.13.6 Plot charge displacement curve

Multiwfn is able to calculate and plot integral curve for grid data, see the introduction in Section 3.16.14. If the grid data is selected as electron density difference, then the integral curve is

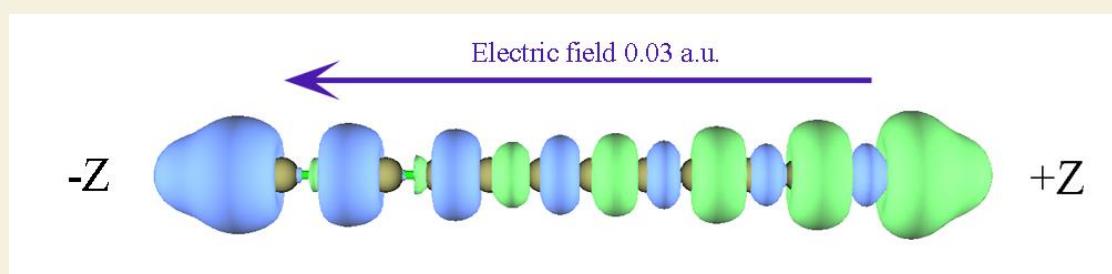
commonly known as charge displacement curve (CDC), by which the charge transfer can be studied visually and quantitatively, extremely suitable for linear systems. In this example, by means of CDC, we will investigate the intermolecular charge transfer in polyyne ($n=7$) due to the externally applied electric field of 0.03 a.u. along the molecular axis.

The polyyne.wfn and polyyne_field.wfn files in "example" folder correspond to the polyyne in its isolated state and in the case that external electric field of 0.03 a.u. is applied, respectively. B3LYP/6-31G* is used in the calculations, and the geometry optimized in isolated state is used for both cases. In Gaussian program, the field can be activated via keyword "field=z+300".

Before plotting the CDC, we must calculate the grid data of electron density difference between these two files first. Boot up Multiwfn and input following commands:

```
examples\polyyne_field.wfn
5 // Calculate grid data
0 // Custom operation
1
-examples\polyyne.wfn //Subtract the property of polyyne.wfn from polyyne_field.wfn
1 // Electron density
2 // Medium quality grid
```

We first visualize the isosurface of the electron density difference. After select -1 and set the isovalue to 0.004, we will see the graph like below

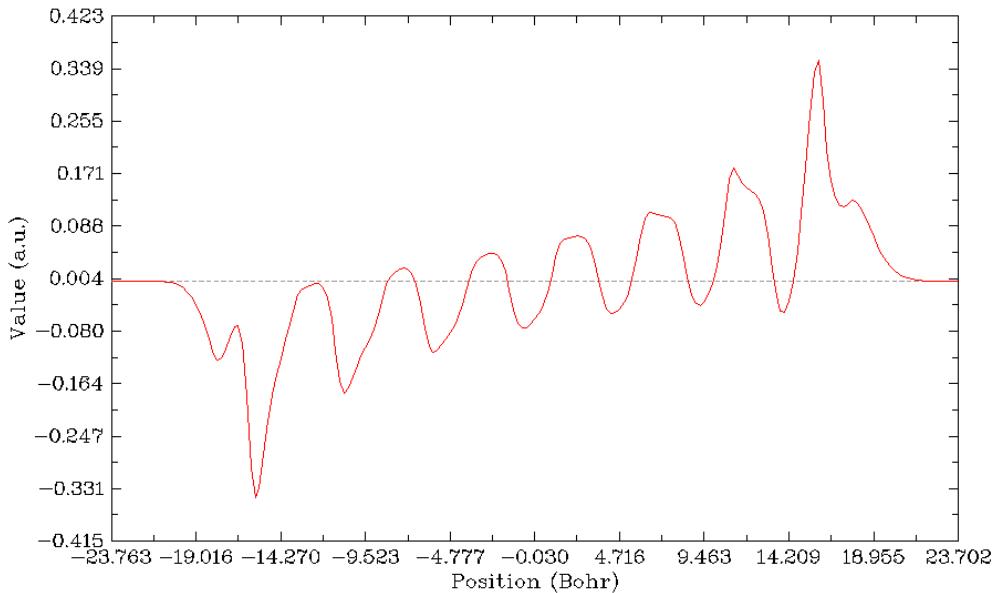


The green and blue parts represent the regions where electron density is increased and decreased after the external electric field is applied, respectively. It can be seen that although green and blue parts interlace with each other, the total trend is that electron transferred to positive side of Z-axis (namely toward the source of the electric field). Next we will plot CDC, which is able to characterize the electron transfer in different regions quantitatively.

Click "Return" button in the GUI and then input

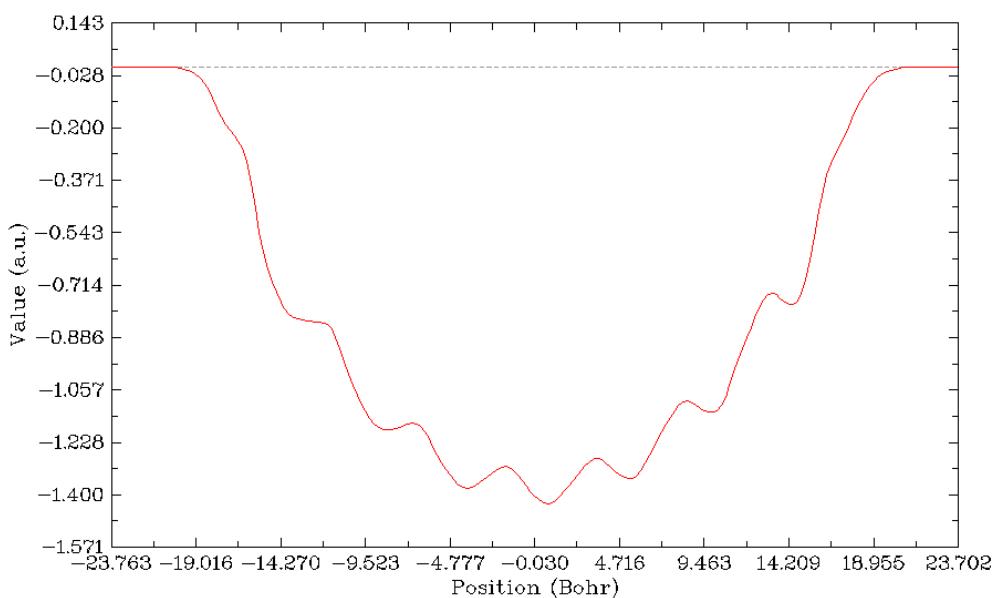
```
0 // Return to main menu
13 // Process grid data
18 // Calculate and plot integral curve
Z // The curve will be plotted in Z direction
a // Select the entire range
```

In the menu, we first choose 2 to plot local integral curve of the grid data of the electron density difference. You will see



From the graph we can examine the integral of electron density difference in the XY planes corresponding to different Z coordinates. The Z coordinate and the value correspond to X and Y axes of the graph. You can directly compare this curve with the isosurface graph shown above, the peaks lower and higher than zero (dashed line) correspond to the blue and green isosurfaces.

Clicking right mouse button on the graph to close it, and then select option 1, the CDC will be shown immediately

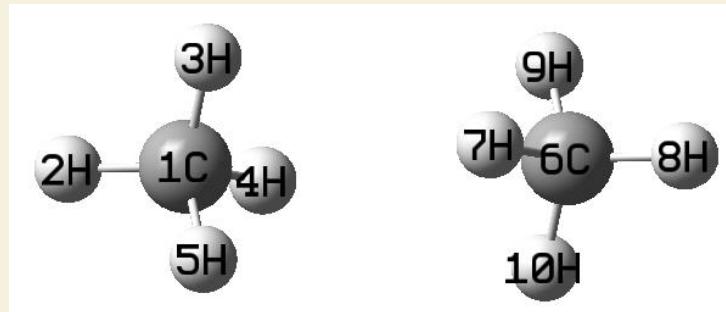


This graph is yielded by integrating the curve shown in the last graph along the molecular axis. In its left part, although there are some fluctuations, the CDC gradually becomes more and more negative and reaches minimum value of 1.4 in the midpoint of the X-axis (corresponding to the center of the polyyne), that means due to the external electric field, the number of lost electrons in the left part of polyyne is 1.4. In the right part of the graph, the CDC increases gradually from -1.4 and finally reaches zero, suggesting that 1.4 electrons are transferred to right part of the polyyne,

and due to the amount of increase and decrease of electron are cancelled with each other exactly in the whole molecular space, there is no variation of the total number of electrons (in other words, integral of the electron density difference in the whole molecular space is exactly zero).

4.13.7 Evaluation of electron density overlap

In this example we take methane dimer as example to evaluate where electron density of the two monomers overlap with each other evidently. The structure of the methane dimer is



We first obtain grid data for an arbitrary real space function for the dimer. Boot up Multiwfn and input:

```
examples\rho_overlap\dimer.pdb
5
100 // User define function, by default this function does not take any computational time
-10 // Set grid extension distance
2 // Decrease the distance to 2 Bohr to avoid waste of grid at system boundary
2 // Medium quality grid
2 // Export grid data as userfunc.cub
```

Now we calculate wavefunction file by Gaussian for the two monomers, the input files are *monomer1.gjf* and *monomer2.gjf* in *examples\rho_overlap* folder. Finally we obtained *monomer1.wfn* and *monomer2.wfn*. Beware that the *nosymm* keyword must be used in Gaussian calculation to avoid the automatic reorientation and translation.

Now we calculate electron density grid for monomer 1 over the whole space

```
monomer1.wfn
5
1 // Electron density
8
userfunc.cub // Use this cube file to define the grid, which corresponds to the whole space
2 // Export grid data
```

Then rename the resulting *density.cub* to *density1.cub*. Repeat above steps for monomer2 to obtain *density2.cub*.

Now we calculate grid data of $\min(\rho(1),\rho(2))$, namely take the minimal value of the two sets of electron density everywhere. Boot up Multiwfn and input:

```
density1.cub
13
11
21 // Take min(rho(1),rho(2))
```

density2.cub

0 // Export resulting grid data

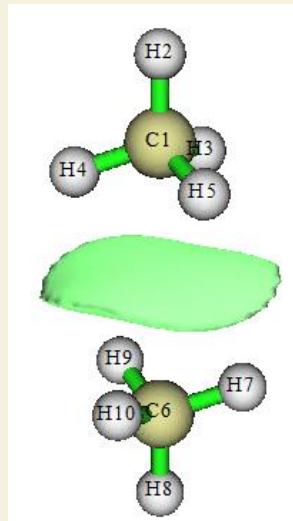
overlap.cub

we open the *overlap.cub* and *density2.cub* by text editor, copy atomic coordinates from the latter to the former, and meantime change the number of atoms. Then the head of the *overlap.cub* should look like below (the highlighted texts are modified parts)

Generated by Multiwfn

```
Totally      531846 grid points
10  -3.693194  -3.955866  -7.444301
63  0.119334  0.000000  0.000000
67  0.000000  0.119334  0.000000
126  0.000000  0.000000  0.119334
6  6.000000  0.000000  0.000000  3.371271
1  1.000000  0.000000  0.000000  5.444301
1  1.000000  0.000000  1.955867  2.677742
1  1.000000  -1.693195  -0.976988  2.677742
1  1.000000  1.693195  -0.976988  2.677742
6  6.000000  0.000000  0.000000  -3.371271
1  1.000000  1.693195  0.976988  -2.677742
1  1.000000  0.000000  0.000000  -5.444301
1  1.000000  -1.693195  0.976988  -2.677742
1  1.000000  0.000000  -1.955867  -2.677742
```

Boot up Multiwfn and load the *overlap.cub*, enter main function 0, change the isovalue to a small one such as 0.0005, you will clearly see the density overlap region:



It is noteworthy that if then you enter main function 13, select subfunction 17 and then input I , you will be able to obtain integral value of the density overlap function over the whole space, namely the "Integral of all data". Clearly, the larger the integral, the higher extent the densities of monomers overlap with each other.

By the way, if you feel the above mentioned steps are too lengthy, you can easily make use of silent mode of Multiwfn to significantly reduce the operation steps, see Section 5.2.

4.14 Adaptive natural density partitioning (AdNDP) analysis

Theory basis of AdNDP analysis has been introduced in Section 3.17.1, please read it first. Below I will show you how to use AdNDP approach to study multi-centers orbitals of several practical molecules. More detailed discussion about AdNDP analysis can be found in my blog article "Study multi-center bonds by AdNDP approach as well as ELF/LOL and multi-center bond order" (in Chinese, <http://sobereva.com/138>).

NOTICE: Using diffuse functions in AdNDP analysis is strongly deprecated, because they often cause numerical problems (which sometimes leads to crash when Multiwfn loading input file) and never improve AdNDP results at all!

4.14.1 Analyze Li₅⁺ cluster

In *Chem. Eur. J.*, **6**, 2982, the authors showed that Li₅⁺ cluster has two 4-centers 2-electrons (4c-2e) bonds by examining ELF isosurfaces. In present example, we will use AdNDP approach to study this cluster to verify their statement. We first optimize Li₅⁺ cluster under B3LYP/6-311G* level and then compile an input file of single point task for Gaussian. *pop=nboread* keyword must be specified in route section, and *\$NBO AONAO DMNAO \$END* must be added to the end of the input file. Run this file by Gaussian, and then convert check point file to .fch format. The input file, output file and .fch file have been given in "examples\AdNDP" folder.

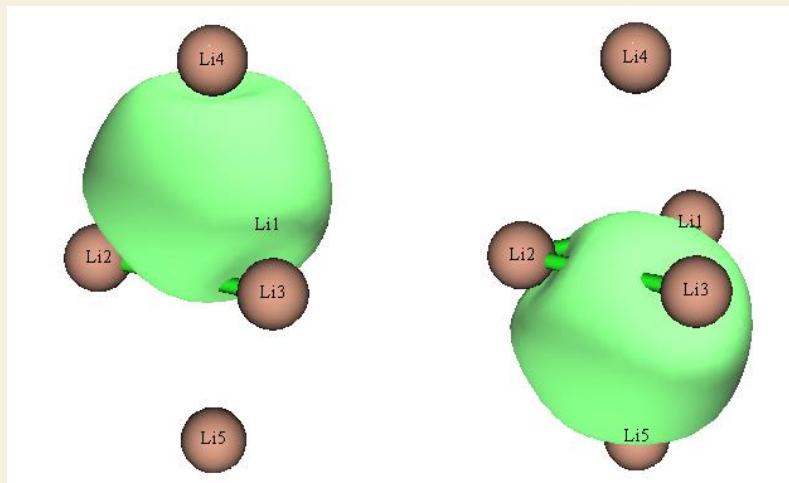
Boot up Multiwfn and input *examples\AdNDP\Li5+.out*, and then choose main function 14. After Multiwfn loaded some necessary information, a menu appears. Since this cluster is small, we can directly use exhaustive manner to search all possible 1c-2e, 2c-2e, 3c-2e, 4c-2e and 5c-2e orbitals in turn. We first choose option 2 to search 1c-2e orbitals (namely lone pairs), however, because occupation numbers of all tried 1c orbitals are lower than default threshold (which is a value close to 2.0 and can be adjusted by option 4), the candidate orbitals list shown in front of the menu is still empty. We then choose option 2 twice to search 2c-2e and 3c-2e orbitals in turn, we still can't find any orbital with high occupation numbers. Next we select option 2 again to search 4c-2e orbitals, this time the candidate orbital list is no longer empty, there are two orbitals in it:

```
# 2 OCC: 1.9966 Atom: 1Li 2Li 3Li 4Li
# 1 OCC: 1.9966 Atom: 1Li 2Li 3Li 5Li
```

Due to their high occupation number, it clear that they are ideal 4c-2e orbitals, therefore we decide to choose option 0 and input 2 to pick them out from candidate list and save as AdNDP orbitals. Whereafter the list of AdNDP orbitals can be printed by option 5.

You may have noticed that the number of residual valence electrons (shown at the top of the menu) has been updated to 0.020, which is already very close to zero, it is suggested it is meaningless to continue to search 5c-2e orbitals because they would be impossible to be found.

Now you can choose option 7 to visualize the two 4c-2e AdNDP orbitals. In order to calculate orbital wavefunction, Multiwfn needs to load basis set information from corresponding .fch file first. Since *Li5+.fch* is in the same folder and has identical name as *Li5+.out*, the .fch file will be directly loaded. When loading is finished, a GUI pops up, which is completely identical to the one of main function 0. AdNDP orbitals can be plotted by selecting corresponding numbers in the right-bottom list. The 0.05 isosurfaces of the two orbitals are shown below.



Grid data of AdNDP orbitals can be exported as Gaussian cube files by option 9, so that you can draw them by some third-part visualization programs such as VMD. You need to input orbital index range, assume that we want to output the two 4c-2e AdNDP orbitals we just found, we should input 1,2, then they will be exported as *AdNDPorb0001.cub* and *AdNDPorb0002.cub* in current folder.

By option 3, you can set the number of centers of multi-centers orbitals in the next exhaustive search. So, assume that you have already known that there are two 4c-2e orbitals in present system, you can directly choose option 3, input 4 and then select option 2 to start the exhaustive search of 4c-2e orbitals, the exhaustive searchs of 1c-2e, 2c-2e and 3c-2e orbitals will be skipped.

It is also possible to obtain energies of the AdNDP orbitals that have been picked out. To realize this, you need to provide additional plain text file containing Fock matrix of present system in lower-triangular sequence, so that orbital energies can be yielded after some transformations of this matrix. The Fock matrix can also be loaded from .47 file. The most straightforward procedure is as follows: Copy *examples\AdNDP\Li5+.gif* as *examples\AdNDP\Li5+_47.gif*, change content between \$NBO ... \$END to *archive file=C:\Li5+.47*. Then after running this file, *C:\Li5+.47* will be yielded, which is input file of GENNBO program and contains the Fock matrix that we need. Then we choose "16 Output energy of picked AdNDP orbital" in the AdNDP analysis interface, input the path of the *Li5+.47* (which has already been provided in "*examples\AdNDP*" folder), Multiwfn will load it and immediately print out AdNDP orbital energies, as shown below:

```
Energy of picked AdNDP orbitals:
orbital: 1 Energy (a.u./ev): -0.325776    -8.8648
orbital: 2 Energy (a.u./ev): -0.325776    -8.8648
```

As expected, the two orbitals are degenerate in energy, since they have exactly equivalent shape.

4.14.2 Analyze B_{11}^- cluster

This time, we will try to reproduce the AdNDP analysis result of B_{11}^- cluster that given in AdNDP original paper (*Phys. Chem. Chem. Phys.*, **10**, 5207).

The files needed by this instance, namely *B11-.out* and *B11-.fch* can be found in "examples\AdNDP" folder. The geometry was optimized under B3LYP/6-311+G*, while the wavefunction was generated under HF/STO-3G level. You may wonder whether the result is meaningful under such low level of basis set; actually, AdNDP analysis is rather insensitive to basis

set quality, even STO-3G is able to produce at least qualitative resonable result. In addition, using larger basis set will bring additional cost at AdNDP analysis stage.

Boot up Multiwfn and input *examples\AdNDP\B11-.out*, then choose 14 to enter AdNDP module. As usual, we select 2 to search 1c-2e orbitals first, but we find nothing (this is common case). Then select 2 again to exhaustively search 2c-2e orbitals from the 11 atoms, Multiwfn will totally try $11!/(11-2)!/2!=55$ combinations, finally there are nine 2c orbitals present in candidate list (sorted according to occupation number from large to small):

```
# 9 Occ: 1.9727 Atom: 6B 10B
# 8 Occ: 1.9727 Atom: 5B 11B
# 7 Occ: 1.9742 Atom: 7B 9B
# 6 Occ: 1.9742 Atom: 7B 8B
# 5 Occ: 1.9869 Atom: 2B 6B
# 4 Occ: 1.9869 Atom: 3B 5B
# 3 Occ: 1.9871 Atom: 9B 11B
# 2 Occ: 1.9871 Atom: 8B 10B
# 1 Occ: 1.9942 Atom: 2B 3B
```

All of them have occupation number close to 2.0, ostensibly one can directly pick all of them out as AdNDP orbitals, however, it is not recommended to do so, because neighboring orbitals may share the same densities. For example, the 1th and the 4th candidate orbitals share the same densities, since both of them related to atom 3. In order to avoid overcount of electrons, firstly you should pick out the first three orbitals by choosing option 0 and input 3, then the density of the first three orbitals will be depleted from density matrix, after that wavefunction and occupation number of remained candidate orbitals will be updated automatically. After that the candidate list becomes

```
# 6 Occ: 1.9538 Atom: 6B 10B
# 5 Occ: 1.9538 Atom: 5B 11B
# 4 Occ: 1.9556 Atom: 7B 8B
# 3 Occ: 1.9556 Atom: 7B 9B
# 2 Occ: 1.9750 Atom: 2B 6B
# 1 Occ: 1.9750 Atom: 3B 5B
```

Since some densities have been depleted, occupation number of remained six candidate orbitals slightly decreased. Now, we pick out the first four candidate orbitals by choosing option 0 and input 4. Although both of 3th and 4th orbitals are related to atom 7, here we have to ignore the slight overcount of electrons, otherwise their degeneration will be broken and thus the final AdNDP pattern will not be consistent with molecular symmetry anymore (you can choose option 8 to carefully inspect candidate orbitals before you decide to pick them out). Finally, we pick out the last two orbitals (*i.e.* 6B-10B and 5B-11B). Currently the number of residual valence electrons is 16.307, which reveals that it is probable to find several higher number of centers orbitals with nearly two electrons occupied.

Now we select option 2 to start the search of 3c-2e orbitals, the resulting candidate orbital list is

```
# 9 Occ: 1.7399 Atom: 1B 6B 10B
# 8 Occ: 1.7399 Atom: 4B 5B 11B
# 7 Occ: 1.7502 Atom: 1B 3B 4B
# 6 Occ: 1.7502 Atom: 1B 2B 4B
```

```
# 5 Occ: 1.8504 Atom: 1B 2B 6B
# 4 Occ: 1.8504 Atom: 3B 4B 5B
# 3 Occ: 1.8603 Atom: 1B 4B 7B
# 2 Occ: 1.8673 Atom: 4B 9B 11B
# 1 Occ: 1.8673 Atom: 1B 8B 10B
```

After we pick out two orbitals (1B-8B-10B and 4B-9B-11B), one orbital (1B-4B-7B) and two orbitals (3B-4B-5B and 1B-2B-6B) in turn, the highest occupation number of remained candidate orbitals is 1.41, which is obviously too low to be recognized as 3c-2e orbital, so they will not be concerned. Currently the number of residual valence electrons is 7.03.

Then you can start to search higher number of centers orbitals, beware that this is never a trivial task, and there is no absolute rule on how to reasonably pick out candidate orbitals, different picking manners result in different AdNDP patterns. You may have to try many times before finally obtaining an optimal AdNDP pattern. It is recommended to use option 11 to save present density matrix and AdNDP orbital list into memory, so that you need not to worry about improper pick of candidate orbitals, since saved state can be recovered anytime by choosing option 12.

Now choose option 2 to start the search of 4c-2e orbitals, the highest occupation is merely 1.71, none of them could be picked out.

Select option 2 again to search 5c-2e orbitals, you will find many 5c candidate orbitals, the first two have occupation numbers of 1.89, we pick out both of them.

Then choose option 2 to search 6c-2e orbitals, no good candidate can be found, the highest occupation number is only 1.84. Then choose option 2 to search 7c-2e orbitals, we pick the highest occupied one (1.90). Now the residual valence electron is only 1.34, which is much less than 2.0, indicating that no additional 2e AdNDP orbital could be found, therefore now we can end the AdNDP searching procedure. The amount of residual electron reflects the electrons that cannot be fully represented by present AdNDP pattern (analogous to non-Lewis electron in the NBO framework)

By choosing option 5, information of all AdNDP orbitals can be printed out:

```
# 1 Occ: 1.9942 Atom: 2B 3B
# 2 Occ: 1.9871 Atom: 8B 10B
# 3 Occ: 1.9871 Atom: 9B 11B
# 4 Occ: 1.9750 Atom: 2B 6B
# 5 Occ: 1.9750 Atom: 3B 5B
# 6 Occ: 1.9556 Atom: 7B 9B
# 7 Occ: 1.9556 Atom: 7B 8B
# 8 Occ: 1.9337 Atom: 6B 10B
# 9 Occ: 1.9337 Atom: 5B 11B
# 10 Occ: 1.8673 Atom: 4B 9B 11B
# 11 Occ: 1.8673 Atom: 1B 8B 10B
# 12 Occ: 1.8533 Atom: 1B 4B 7B
# 13 Occ: 1.8451 Atom: 3B 4B 5B
# 14 Occ: 1.8451 Atom: 1B 2B 6B
# 15 Occ: 1.8908 Atom: 1B 2B 6B 8B 10B
# 16 Occ: 1.8908 Atom: 3B 4B 5B 9B 11B
# 17 Occ: 1.9036 Atom: 2B 3B 5B 6B 7B 8B 9B
```

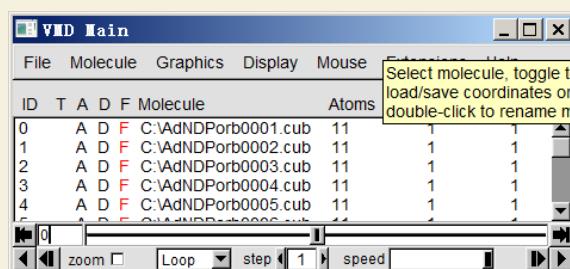
Total occupation number in above orbitals: 32.6607

Now you can use option 7 to visualize all picked AdNDP orbitals. However this time I show how to plot AdNDP orbitals via VMD, which could simultaneously plot a batch of orbitals and the graphical quality is better. VMD can be freely obtained at <http://www.ks.uiuc.edu/Research/vmd/>.

Choose option 9 and input 1,17 to export all the seventeen AdNDP orbitals as cube files, the format of the file name is *AdNDPorb[index].cub*. Assume that you have move all of them to C:\ directory, you should edit *examples\AdNDP\plotAdNDP.vmd*, change this line

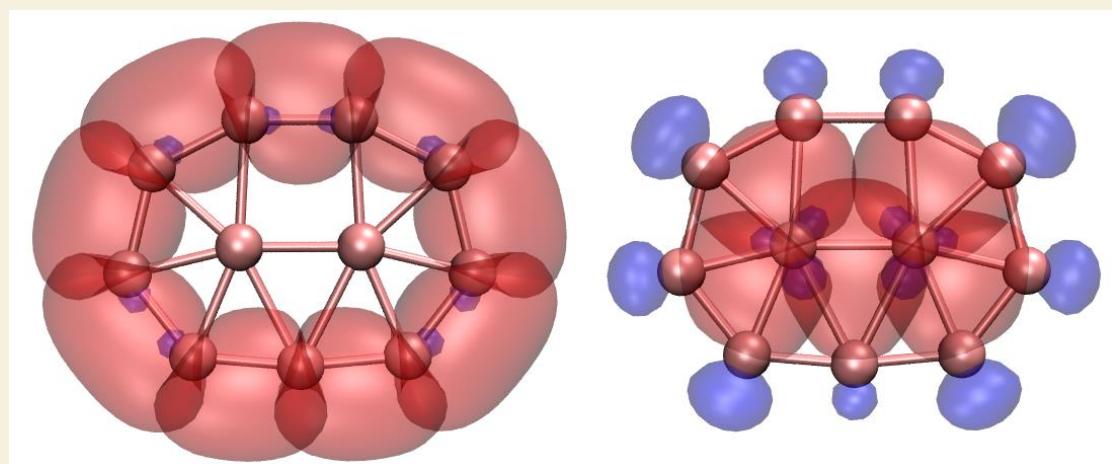
```
set name "D:\\CM\\my_program\\Multiwfn\\AdNDPorb$idx.cub"
to
set name "C:\\AdNDPorb$idx.cub"
```

Then boot up VMD, copy all content in the *plotAdNDP.vmd* to VMD console window, all cube files of AdNDP orbitals will be loaded into VMD. Now the VMD Main window looks like below

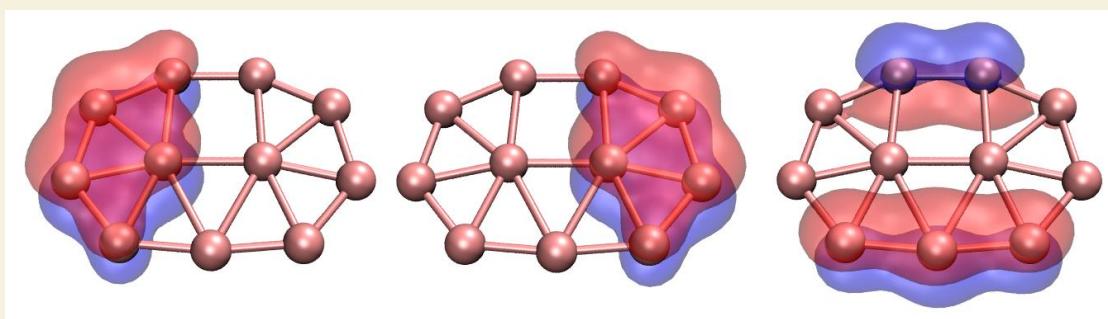


Each entry corresponds to an AdNDP orbital. Currently all the 17 orbitals are shown. If you double click a "D" label, then corresponding orbital will be hidden in the graphical window. In order to show the molecule structure, drag the *examples\AdNDP\B11-.xyz* into the VMD main window to load it, then enter "Graphics" - "Representation" and change the drawing style as CPK.

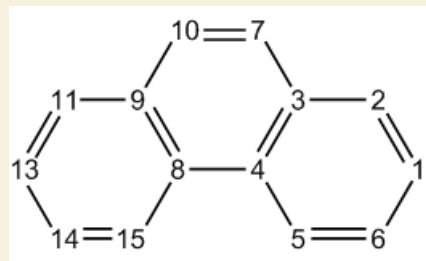
If you make VMD only display all the nine 2c-2e and all the five 3c-2e orbitals, you will see left and right parts of below graph, respectively



The two 5c-2e and one 7c-2e orbitals are shown below (In the graph the 7c-2e orbital looks like 5c orbital, the main reason is that the isovalue in the plotting script is relatively high, i.e. 0.06).



4.14.3 Analyze phenanthrene



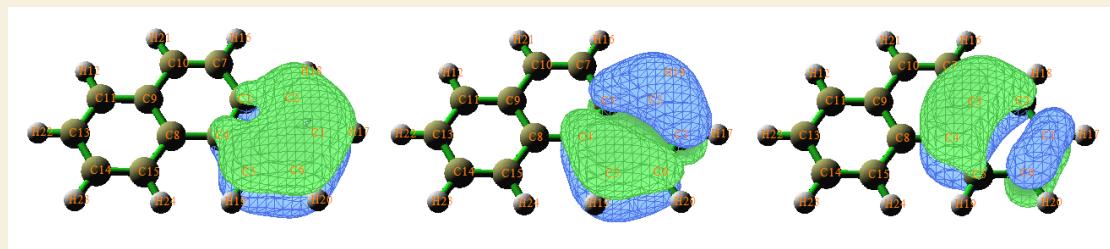
AdNDP analysis of phenanthrene ($C_{14}H_{10}$, see above) has been given in *J. Org. Chem.*, **73**, 9251, in this section we will repeat their result, you will learn how to use user-directed search. Files used in this example can be found in *examples\AdNDP* folder with "phenanthrene" prefix.

First we load *examples\AdNDP\phenanthrene.out* and enter main function 14. Consistent with the previous examples, we select option 2 twice to search 1c orbitals and then search 2c orbitals. No 1c-2e orbitals can be found, while there are 31 candidate 2c orbitals present in the list. Ten of them correspond to C-H σ -bonds and have no overlapping with each other, so we can pick them out first, namely choosing option 0, input 8,15, then choose option 0 again and input 9,10. Next, we successively pick out sixteen 2c candidate orbitals that corresponding to C-C σ -bonds (the most careful input is 0 2 0 1 0 2 0 1 0 2 0 2 0 2 0 2, where space denotes pressing ENTER button). Then there are only five orbitals remain, the first orbital with occupation number of 1.80 corresponds to the π -bond between C7 and C10, it seems worth to be picked out. The occupation numbers of the four remained orbitals are about 1.72, thus they are not ideal 2c-2e bonds, we do not concern these five orbitals at the moment.

Although we can use option 2 to exhaustively search 3c, 4c, 5c ... orbitals as usual, however this may be not a good idea for present system, user-directed search is often more effective. We first choose option 13 to check population of residual electrons on each atom, see below, this information is usually helpful for guiding users to properly set up exhaustive search list. (Note: The exhaustive search triggered by option 2 is only applied to the atoms in exhaustive search list, which contains all atoms in present system by default)

1C : 1.0250	2C : 1.0370	3C : 1.0280	4C : 1.0414
5C : 1.0339	6C : 1.0262	7C : 0.1322	8C : 1.0414
9C : 1.0280	10C : 0.1322	11C : 1.0370	12H : 0.0117
13C : 1.0250	14C : 1.0262	15C : 1.0339	16H : 0.0121
17H : 0.0113	18H : 0.0117	19H : 0.0126	20H : 0.0111
21H : 0.0121	22H : 0.0113	23H : 0.0111	24H : 0.0126

From above data it is clear that hydrogens have almost zero population numbers, hence they could be safely removed from the exhaustive search list. Due to the same reason C7 and C10 can be removed too. The other atoms, whose occupation numbers are about 1.03, are the carbons composing the two 6-member rings in both sides of the molecule. It can be expected that the two rings may be analogous to benzene ring and hence representing local aromaticity in phenanthrene. Based on this consideration, we choose option -1 to enter the interface for defining exhaustive search list, input *clean* to clean up the default content first, then input *a 1-6* to add atom 1,2,3,4,5,6 into the list, and then input *x* to save and exit. Now the search list only contains carbon atoms in a single ring. Select option 3 and input 6, then choose option 2 to search 6-centers orbitals in the ring, three candidate orbitals will be found, whose occupation numbers are 2.000, 1.985 and 1.821 respectively, they are appropriate to be picked out as 6c-2e AdNDP orbitals, so we pick them out now. Their 0.03 isosurfaces are shown below, which look like π molecular orbitals of benzene.



Next, with the same method, we define exhaustive search list as atoms 8,9,11,13,14,15 and then search out the corresponding three 6c-2e orbitals, then finally pick them out. Now all AdNDP orbitals (26^*2c-2e , 6^*6c-2e) have been found, the AdNDP analysis is finished.

With similar procedure, we evaluate AdNDP orbital energies like Section 4.14.1. The NBO .47 file containing Fock matrix of current molecule has been provided as *examples\AdNDP\phenanthrene.47*, which was yielded by *examples\AdNDP\phenanthrene_47.gjf*. We choose option 16 and input the path of this file, Multiwfn immediately loads Fock matrix from it and outputs the orbital energies:

```
... (ignored)
orbital: 23 Energy (a.u./ev): -0.697578 -18.9821
orbital: 24 Energy (a.u./ev): -0.697578 -18.9821
orbital: 25 Energy (a.u./ev): -0.685803 -18.6616
orbital: 26 Energy (a.u./ev): -0.685803 -18.6616
orbital: 27 Energy (a.u./ev): -0.348201 -9.4750
orbital: 28 Energy (a.u./ev): -0.255535 -6.9535
orbital: 29 Energy (a.u./ev): -0.264372 -7.1939
orbital: 30 Energy (a.u./ev): -0.348201 -9.4750
orbital: 31 Energy (a.u./ev): -0.255535 -6.9535
orbital: 32 Energy (a.u./ev): -0.264372 -7.1939
```

As you can see, the last six, which correspond to π orbitals, have energy much higher than σ orbitals. The three π orbitals in left six-membered ring is symmetric to the counterpart three π orbitals in right six-membered ring. In each set, the highest two are nearly degenerate and evidently higher than the lowest one, this situation is very similar to occupied π orbitals of isolated benzene.

Sometimes composition of AdNDP orbitals is interesting. In order to perform the analysis, we

select option 14 to output all of the $26+6=32$ AdNDP orbitals as *AdNDP.molden* in current folder, note that this step requests .fch file of present system. Our system has 146 basis functions, so there are 146 orbitals in the .molden file, but only the first 32 orbitals correspond to the AdNDP orbitals that we have found, the other orbitals are meaningless and should be simply ignored.

By using the *AdNDP.molden* as input file, you can use main function 8 to analyze the composition of AdNDP orbitals. Some examples of orbital composition analysis can be found in Section 4.8, related theoretical introductions can be found in Section 3.10. All orbital composition analysis methods except for NAO method can be applied to AdNDP orbitals. For example, here we analyze the composition of the 27th AdNDP orbital, namely the orbital at left side of above graph. Boot up a new Multiwfn and input below command:

```
AdNDP.molden
8 // Orbital composition analysis
1 // Mulliken analysis
27
```

Contribution of the six carbons in the ring are shown below, please compare the data with the corresponding AdNDP orbital isosurface.

Atom	1(C) :	33.144079%
Atom	2(C) :	13.320911%
Atom	3(C) :	1.585369%
Atom	4(C) :	1.925742%
Atom	5(C) :	15.297803%
Atom	6(C) :	34.805871%

4.15 Fuzzy atomic space analysis

Delocalization index, PDI, FLU and FLU- π were originally proposed for AIM atomic space, it has been shown that, if they are calculated in fuzzy atomic space, the computational cost would be significantly reduced, while the results are still reasonable. The introduction of basic concepts of fuzzy atomic space has been given in Section 3.18.0. In this section, we will calculate these quantities in the fuzzy atomic space defined by Becke. Multiwfn also supports Hirshfeld and Hirshfeld-I methods to defined atomic space, however, the Becke's definition has the advantage that the calculation is independent on reference atomic densities, so we do not need to concern the preparation of .wfn files for atoms in their free-states, and therefore the calculation procedure is somewhat simpler.

4.15.1 Study delocalization index of benzene

The definition of delocalization index (DI) has been detailedly introduced in Section 3.18.5. In present instance we will calculate DI in Becke's fuzzy atomic space to study the extent of electron delocalization between different atom pairs in benzene. Notice that the original DI is defined in Bader's AIM atomic space, so the DIs calculated in this example may somewhat differ to the ones in some literatures.

Boot up Multiwfn, and input following commands

```

examples\benzene.wfn // Generated under B3LYP/6-311G*
15 // Fuzzy atomic space analysis
4 // Calculate localization index (LI) and DI. Multiwfn first makes use of DFT numerical
quadrature scheme to calculate atomic overlap matrix (AOM) in each fuzzy atom space, and then
convert AOM to DI and LI.
n // Do not output LI and DI to plain text file

```

Multiwfn automatically checks and output the error of AOM, for present calculation the error is less than 0.001, which is completely negligible. If the error is too larger to be accepted, you can set "iautointgrid" in settings.ini to 0, and set "radpot" and "sphpot" to a large value. When "iautointgrid" is equal to 1, Multiwfn uses (40,230) grid points to calculate AOM, accuracy of which directly affect the accuracy of LI, DI, as well as of PDI and FLU.

From the DI matrix outputted on the screen, we can see that DI between adjacent two carbon atoms and adjacent C-H atoms are large (1.467 and 0.877, respectively), that means electron delocalization between bonded atoms is strong, this is mainly due to the shared electron of σ bond. In contrast, the DI between non-bonded carbon atoms is very small, about 0.1, nevertheless evidently not zero, reflecting the high-degree delocalization nature of π electrons.

The DI calculated in fuzzy atomic space is essentially the fuzzy bond order proposed by Mayer. According the DI data, we can say that the bond order between C-C bond and C-H bond in benzene is 1.467 and 0.877 respectively, the former corresponds to single σ bond + "semi" π bond, while the latter corresponds to typical single σ bond.

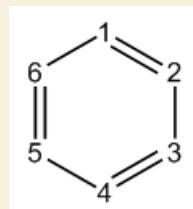
Since benzene is an exactly planar molecule, we can decompose DI to DI- σ and DI- π . Here we calculate the latter. Input 0 to return to main menu, and then following commands

```

6 //Modify wavefunction
26 //Modify occupation number
0 // Select all orbitals
0 // Make occupation number of all orbitals to zero
17,20,21 // MO 17,20,21 correspond to  $\pi$  orbitals.
2 // Set their occupation numbers to 2 (closed-shell orbitals)
q // Return to upper level of menu
-I //Return to main menu

```

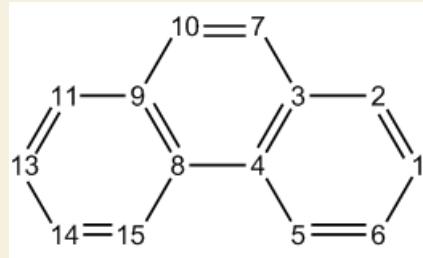
Now recalculate DI as before, since the occupation numbers of all orbitals except π orbitals have been set to zero, the result will be DI- π .



The DI- π between C1-C6, C1-C5 and C1-C4 are 0.438, 0.055 and 0.093 respectively, it is obvious that π electron delocalization is larger for para-related than for meta-related carbon atoms.

4.15.2 Study aromaticity of phenanthrene by PDI, FLU, FLU- π and PLR

PDI, FLU, FLU- π and PLR are very useful aromaticity indices, their definitions have been introduced in Section 3.18.6, 3.18.7 and 3.18.9. In present instance, we will calculate them in Becke's fuzzy atom space to study aromaticity of different rings of phenanthrene.



We first calculate PDI. Boot up Multiwfn, and input following commands:

```
examples\phenanthrene.wfn // Optimized at B3LYP/6-31G* level
```

```
15 // Fuzzy atomic space analysis
```

```
5 // Calculate PDI
```

Then Multiwfn starts to calculate atomic overlap matrix (AOM), this is a computationally intensive work. After that AOM will be converted to delocalization index (DI), then DI matrix will be outputted on screen. Finally, you will be prompted to input atom indices of the ring you are interested in, the input order should be in consistency with atomic connectivity. We first calculate PDI of the central ring, namely input 4,8,9,10,7,3, the result is

```
Delocalization index of    4(c) -- 10(c) : 0.053069
Delocalization index of    8(c) -- 7(c) : 0.053069
Delocalization index of    9(c) -- 3(c) : 0.036350
PDI value is   0.047496
```

PDI value is just the average of the DIs between C4-C10, C8-C7 and C9-C3. Now we input 8,9,11,13,14,15 to calculate PDI of the boundary ring, the result is 0.0817. From this result it is evident that the electron delocalization in the boundary rings is stronger than in the central ring, so boundary rings possess larger aromaticity. Next, we use FLU and FLU- π to study the aromaticity, and check if we can draw the same conclusion.

Input *q* to return to upper level of menu, and input 6 to calculate FLU, then input 4,8,9,10,7,3 and 8,9,11,13,14,15 in turn. The FLU of central ring and of boundary rings are 0.025311 and 0.007537 respectively, this result suggests that the boundary rings is more like to typical aromatic system (benzene), and hence possesses larger aromaticity than the central ring. Note that since the AOM has already been calculated during calculating PDI, so this time the calculation process of AOM is automatically skipped.

Next, input *q* to return to upper level of menu, and input 7 to calculate FLU- π . First you need to input the indices of π orbitals. By visually checking isosurface of all orbitals (or utilizing option 22 in main function 100), we know that 36, 40, 43, 44, 45, 46, and 47 are π orbitals, so here we input 36,40,43,44,45,46,47, then DI of π electrons will be outputted. After that you will be prompted to input the atom indices in the ring, we input 4,8,9,10,7,3 and 8,9,11,13,14,15 in turn. The FLU- π of central ring and boundary rings are 0.149288 and 0.035037, respectively. Obviously, FLU- π analysis also confirms that boundary ring is more aromatic.

Finally, let's calculate para linear response index (PLR). PLR is based on linear response kernel, which relies on virtual MOs information; however .wfn file only contains occupied MOs, therefore we must use .fch/.molden/.gms file as input. Reboot Multiwfn and input following commands

```
examples\phenanthrene.fch // Obtained at the same level as phenanthrene.wfn  
15  
10 // Calculate PLR
```

Multiwfn will calculate condensed linear response kernel (CLRK) matrix, after that input 4,8,9,10,7,3 and 8,9,11,13,14,15 in turn, the result is 0.247848 and 0.489434, respectively. Since the former is much smaller than the latter, PLR also validates the conclusion that boundary rings have larger aromaticity than central ring.

Note that both PDI and PLR can be separated as α and π parts to respectively investigate α and π aromaticity, see Section 3.18.6 and 3.18.9 for detail.

4.16 Charge decomposition analysis and plotting orbital interaction diagram

The theoretical background of charge decomposition analysis (CDA, *J. Phys. Chem.*, **99**, 9352), extended CDA (*J. Am. Chem. Soc.*, **128**, 278), and general CDA (GCDA, *J. Adv. Phys. Chem.*, **4**, 111-124 (2015)), as well as usage of CDA module of Multiwfn have been detailedly introduced in Section 3.19. In this section I present several practical examples.

The CDA module of Multiwfn supports Gaussian output file, .fch file, Molden input file (.molden) and GAMESS-US output file (.gms) as input. If the calculation conditions are identical, they will give rise to the same CDA result. In the next severals sections only Gaussian .fch files will be used to exemplify the usage of the CDA module.

4.16.1 Closed-shell interaction case: COBH₃

In COBH₃, CO makes use of its lone pair to form coordinate bond with BH₃, which is an electron-deficient system (Lewis acid). Therefore, electrons will transfer from CO to BH₃ during the formation of the complex. In this example we will employ CDA scheme to provide a deeper insight on the electron transfer.

First we generate Gaussian output file for CO (fragment 1), BH₃ (fragment 2) and COBH₃ (complex). The .fch files and corresponding input files have been provided in "examples\CDA\COBH3" folder. The calculations were performed at HF/6-31G* level. On how to prepare the input files for CDA, see Section 3.19.2 for detail

Now boot up Multiwfn, and input following contents:

```
examples\CDA\COBH3\COBH3.fch // Gaussian .fch file of the complex  
16 // Enter CDA module  
2 // We define two fragments  
examples\CDA\COBH3\CO.fch // Gaussian .fch file of fragment 1  
examples\CDA\COBH3\BH3.fch // Gaussian .fch file of fragment 2
```

Immediately, below CDA result will be outputted on screen

Orb.	occ.	d	b	d - b	r
1	2.000000	-0.000004	-0.000000	-0.000004	-0.000001
2	2.000000	0.001119	-0.000023	0.001141	0.000326
3	2.000000	-0.000002	-0.000471	0.000469	0.000313
4	2.000000	-0.013250	-0.000704	-0.012546	-0.005676
5	2.000000	0.041648	-0.003309	0.044957	0.232262
6	2.000000	0.037385	-0.020136	0.057521	0.212422
7	2.000000	-0.000543	0.000647	-0.001190	0.022166
8	2.000000	-0.000543	0.000647	-0.001190	0.022166
9	2.000000	0.171353	0.026952	0.144401	-0.741381
10	2.000000	-0.000569	0.043713	-0.044281	-0.038916
11	2.000000	-0.000569	0.043713	-0.044282	-0.038916
12	0.000000	0.000000	0.000000	0.000000	0.000000
13	0.000000	0.000000	0.000000	0.000000	0.000000
14	0.000000	0.000000	0.000000	0.000000	0.000000
15	0.000000	0.000000	0.000000	0.000000	0.000000
.....					
<hr/>					
Sum:	22.000000	0.236023	0.091027	0.144996	-0.335233

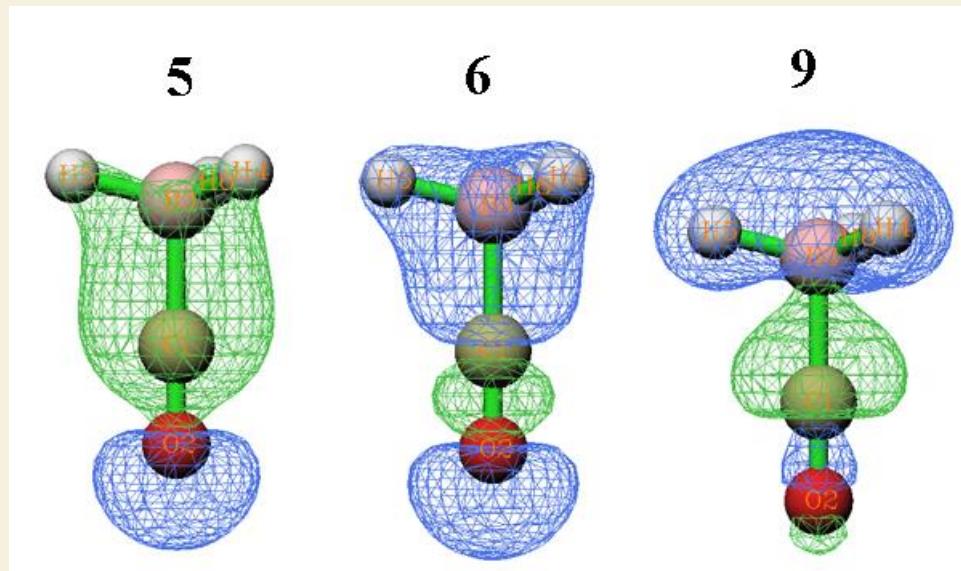
"Orb." denotes the indices of the orbitals of the complex; "occ." is corresponding occupation number. "d(i)" stands for the amount of donated electrons from fragment 1 to 2 via corresponding complex orbital, "b(i)" stands for the amount of electrons back donated from fragment 2 to 1 via corresponding complex orbital. "r(i)" corresponds to the overlap population between the occupied fragment orbitals (FOs) of the two fragments in corresponding complex orbital; its positive and negative signs imply that in this complex orbital, the electrons of occupied FOs are accumulated to and depleted (mainly due to Pauli repulsion) from the overlap region between the two fragments, respectively. The sum of r(i), namely -0.335, reveals that repulsive effect dominates the overall interaction between occupied FOs, which results in corresponding electrons moved away towards nonoverlapping regions from overlap regions.

The difference between d(i) and b(i), to some extent can be viewed as the number of net transferred electrons from fragment 1 to 2 due to formation of corresponding complex orbital. But bear in mind, electron polarization effect is also included into this value.

From the data, it can be seen that the first three complex orbitals have almost zero b, d and r values, this is because they are core orbitals of O, C and B, respectively, and hence it can be expected that they are not involved in bond formation. The virtual complex orbitals have exactly zero b, d and r terms, since their occupation numbers are exactly zero. Orbital 9 leads 0.171 electrons donate from CO to BH₃, which is the primary source of the donor-acceptor bonding, and causes as high as 0.741 electrons removed from overlap region between CO and BH₃, which stabilized the complex by diminishing electron repulsion. Orbital 5 and 6 have small contributions to electron donation, meanwhile lead evident accumulation of electrons from respective occupied FOs to the overlap region, which must be beneficial to the bonding between the two fragments. Orbital 10 and 11 are π orbitals and degenerate in energy, they exhibit π -back donation characteristics.

Isosurfaces of orbital 9, 5 and 6 are shown below. As you can see, a node occurs in the overlap

region between CO and BH₃ in orbital 9, while the isosurfaces of orbital 5 and 6 uniformly cover the overlap region. These observations largely explained why r(9) is a large negative value, and r(5) and r(6) are obvious positive values.



Notice that the definition of CDA used in Multiwfn is a generalized version proposed by me, so that CDA is also applicable to post-HF and open-shell circumstances, see corresponding part of Section 3.19.1 for detail. For the cases when original CDA is applicable (namely MO for FO, MO or natural orbital for complex orbital), the *d* and *b* terms produced by generalized CDA are exactly identical to the ones produced by original definition, while the *r* term is exactly twice of the one produced by original definition.

Note: The COBH3 example was also given in original paper of CDA, in which, although the formulae of *d*, *b* and *r* are correct, the data in their examples are incorrectly: all of the data should be divided by two. I also found that all of the results of the CDA program written by Dapprich and Frenking (<http://www.uni-marburg.de/fb15/ag-frenking/cda>), and the results of AOMix program (<http://www.sg-chem.net>) should be divided by two. So, if you want to compare the CDA results calculated by Multiwfn with theirs, the *d* and *b* terms should be multiplied by two first. But do not do this for *r* term, since the *r* term computed by Multiwfn has already been doubled with respect to its original definition.

The amount of net electron transfer between the two fragments can be characterized by *b-d* term, however it was argued that ECDA is a more reasonable method to calculate the amount of net electron transfer, since the contribution of electron polarization effect (PL) is completely excluded. ECDA result is outputted after CDA result:

```
===== Extended Charge decomposition analysis (ECDA) =====
Contribution to all occupied complex orbital:
Occupied, virtual orbitals of fragment 1:    680.4194%      8.0593%
Occupied, virtual orbitals of fragment 2:    390.3988%      21.1226%
Contribution to all virtual complex orbital:
Occupied, virtual orbitals of fragment 1:    19.5806%      2291.9407%
Occupied, virtual orbitals of fragment 2:    9.6012%      1678.8774%
PL( 1) + CT( 1-> 2) =    0.3916      PL( 1) + CT( 2-> 1) =    0.1612
PL( 2) + CT( 1-> 2) =    0.4225      PL( 2) + CT( 2-> 1) =    0.1920
The net electrons obtained by frag. 2 = CT( 1-> 2) - CT( 2-> 1) =    0.2304
```

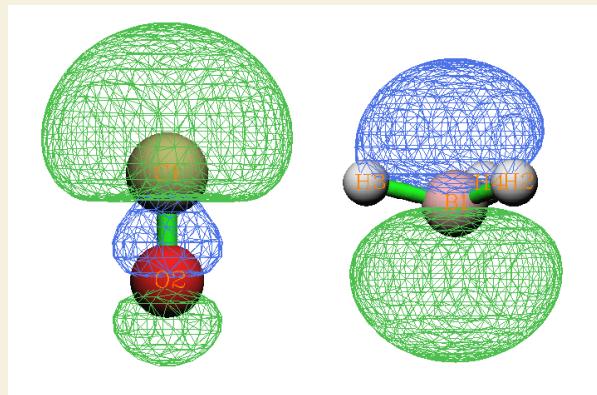
Commonly, you only need to pay attention to the last line. The data shows that the net number

of electrons transferred from fragment 1 to 2 is 0.2304.

In the menu that appears on screen, by using option 2, composition of FOs in a specific complex orbital can be outputted. Here we select this option and input 9, the composition of complex orbital 9 are shown below

```
occupation number of orbital      9 of the complex: 2.0000000
orbital    7 of fragment 1, occ: 2.00000  Contribution: 25.8560%
orbital   13 of fragment 1, occ: 0.00000  Contribution: 1.0798%
orbital    2 of fragment 2, occ: 2.00000  Contribution: 57.2921%
orbital    5 of fragment 2, occ: 0.00000  Contribution: 14.5640%
```

Only the FOs with contribution $\geq 1\%$ to the complex orbital are shown (the threshold can be altered by "comphresCDA" in settings.ini). As already mentioned, the electron transfer from CO to BH_3 is mainly due to the complex orbital 9, therefore from above data we can infer that the nature of the $\text{CO} \rightarrow \text{BH}_3$ electron transfer can be largely interpreted as the mix between FO 7 of CO (an occupied orbital) and FO 5 of BH_3 (a virtual orbital). This viewpoint can also be manifested by comparing the shape of the two FOs (see below) with the shape of complex orbital 9 given above.



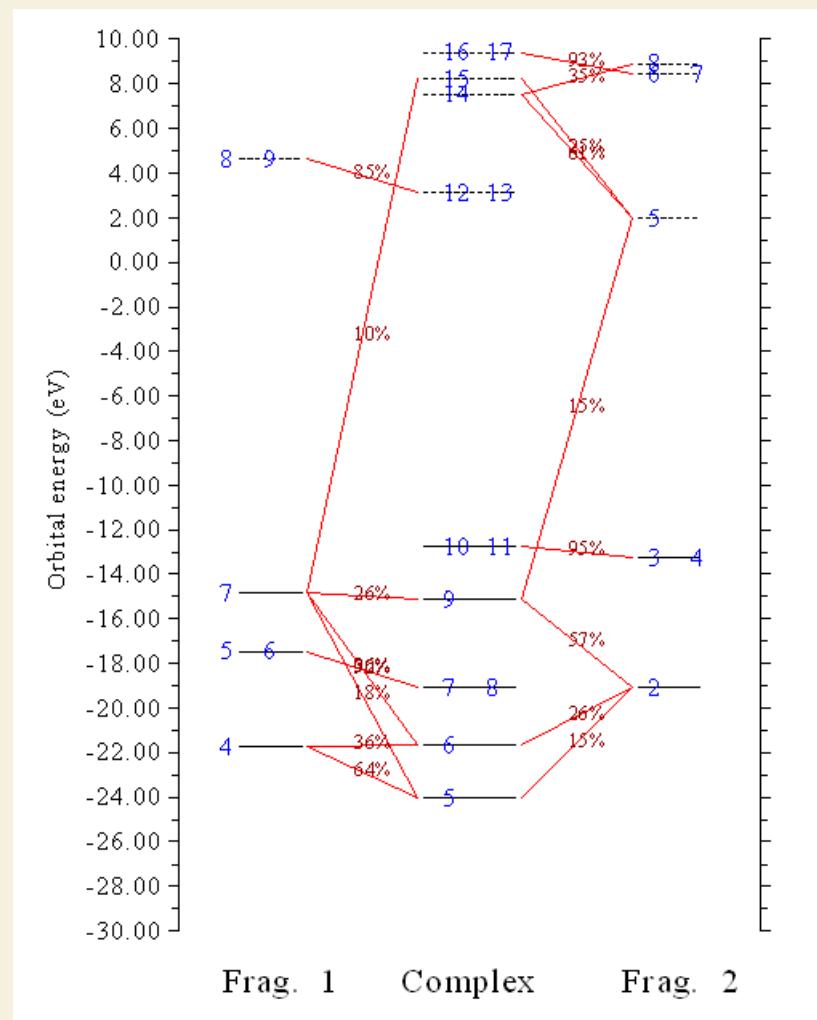
Note: Sometimes a few contributions of FOs may be negative, this is a well-known drawback of Mulliken analysis, which is the method employed in CDA module to calculate the complex orbital composition. Since the magnitudes of the negative values are often small, you can simply ignore them.

We can also directly decompose d , b , r terms of a complex orbital to FO pair contributions, let's do this for complex orbital 9. Now input 0 to return to the last menu, select option 6 and input 9, then input a threshold e.g. 0.005, then all FO pairs whose contribution to any of d , b , r term larger than 0.005 are printed:

FragA Orb(Occ.)	FragB Orb(Occ.)	d	b	$d - b$	r
4(2.0000)	2(2.0000)	0.000000	0.000000	0.000000	-0.009969
7(2.0000)	1(2.0000)	0.000000	0.000000	0.000000	-0.005759
7(2.0000)	2(2.0000)	0.000000	0.000000	0.000000	-0.723845
7(2.0000)	5(0.0000)	0.176503	0.000000	0.176503	0.000000
7(2.0000)	8(0.0000)	0.006221	0.000000	0.006221	0.000000
7(2.0000)	11(0.0000)	0.005846	0.000000	0.005846	0.000000
7(2.0000)	12(0.0000)	-0.023941	0.000000	-0.023941	0.000000
13(0.0000)	2(2.0000)	0.000000	0.021958	-0.021958	0.000000

From the output it is clear that the mix between FO 7 of CO and FO 5 of BH_3 contributes most of the d term of complex orbital 9.

Finally, we plot orbital interaction diagram. Input 0 to return to the last menu, and then select 5 to enter the menu for plotting orbital interaction diagram. Choose option 3 and input -30,10 to set lower and upper energy limits of the plot to -30eV and 10eV, respectively. Then select option 1 to plot the diagram under default settings, a graph will pop up on screen:



In above graph, occupied and virtual orbitals are represented as solid and dashed bars, respectively. Orbital indices are labelled by blue texts. If two or more labels occur in the same bar, e.g. 7 and 8, that means these orbitals are degenerate in energy. If composition of a FO in a complex orbital is larger than 10%, then the corresponding two bars will be linked by red line, and the composition will be labelled on the lines. By simply viewing the diagram we can directly understand how the MOs of COBH_3 are constructed by FOs of CO and BH_3 . For example, complex orbital 7 and 8 only link to FO 5 and 6 of CO in this diagram, hence we immediately know that these two orbitals basically remain unperturbed during formation of the complex. In fact they are π orbitals of CO, certainly they cannot participate to the σ type donor-acceptor interaction between CO and BH_3 .

There are many options used to adjust plotting parameters (such as label size, the rule for drawing and linking bars, position of composition labels, energy range), please play with them, and replot the graph to check their effects.

4.16.2 Open-shell interaction case: CH₃NH₂

In this example, I use CH_3NH_2 to illustrate how to perform CDA for the complex in which the

two fragments interact with each other covalently (open-shell interaction).

First we need to generate Gaussian output file for CH₃ (fragment 1), NH₂ (fragment 2) and CH₃NH₂ (complex). For fragment 1 and 2, in present example we use unrestricted B3LYP method; while for the complex, since this is a closed-shell system, we use restricted B3LYP method (unrestricted B3LYP can also be used, the CDA result will be the same). The .fch files and corresponding input files can be found in "examples\CDA\CH3NH2" folder, the geometry was pre-optimized under B3LYP/6-31G** level.

Note that both CH₃ and NH₂ have 5 alpha and 4 beta electrons, while CH₃NH₂ has 9 alpha and 9 beta electrons. Evidently, the total numbers of alpha and beta electrons in the two fragments, namely 5+5 and 4+4, does not match the ones of the complex. So, we must flip electron spin of one fragment (either CH₃ or NH₂). In this example, we will flip electron spin of NH₂, i.e. exchanging all information of its alpha and beta electrons.

Boot up Multiwfn and input following contents:

```
examples\CDA\CH3NH2\CH3NH2.fch // Gaussian output file of the complex
```

```
16 // Enter CDA module
```

```
2 // We define two fragments
```

```
examples\CDA\CH3NH2\CH3.fch // Gaussian output file of fragment 1
```

```
examples\CDA\CH3NH2\NH2.fch // Gaussian output file of fragment 2
```

```
n // Do not flip electron spin of fragment 1
```

```
y // Flip electron spin of fragment 2, then NH2 will have 4 alpha and 5 beta electrons.
```

CDA and ECDA results will be calculated and printed on screen for alpha electrons and beta electrons separately. As you can see, for alpha (beta) part, both d - b and CT(1->2) - CT(2->1) terms are positive (negative), showing that alpha (beta) electrons are transferred from CH₃ to NH₂ (from NH₂ to CH₃). This is mainly because CH₃ has more alpha electrons (5) than beta electrons (4), while after flipping electron spin, NH₂ has more beta electrons (5) than alpha electrons (4), hence when they combine together to form CH₃NH₂, CH₃ prefer to donate alpha electrons to NH₂ and accept beta electrons from NH₂.

Result of total electrons, namely the sum of alpha and beta results are also outputted. Below is total result of CDA and ECDA, respectively

```
d= 0.044335 b= 0.145181 d - b = -0.100847 r= -0.172318
```

```
CT( 1-> 2) - CT( 2-> 1) for all electrons: 0.1252
```

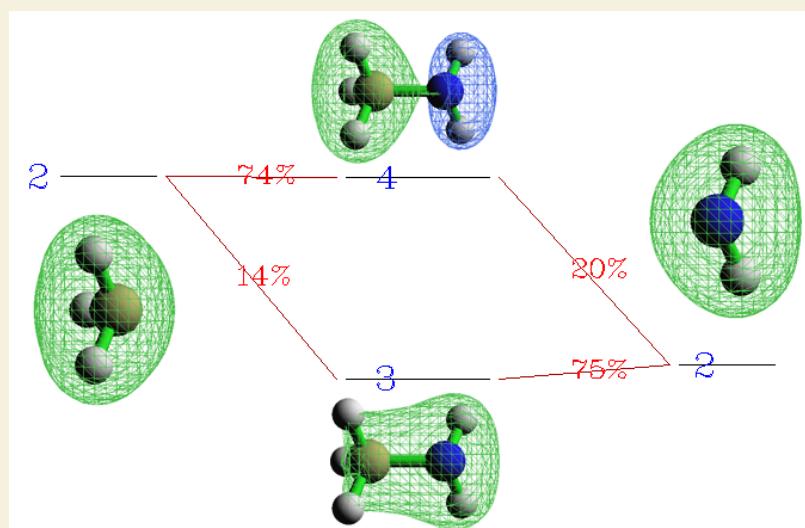
It is obvious that electron should transfer from CH₃ to NH₂, because nitrogen has evidently larger electronegativity. The total ECDA result is in line with our expectation, it shows that the amount of transferred electrons from CH₃ to NH₂ is 0.1249. However, the d - b term conflicts with our expectation, the transfer direction is totally inverted. This example illustrates that d - b term is not as reliable as ECDA to reveal total amount of net electron transfer for open-shell interaction, although d and b terms are still quite useful for decomposing electron transfer into orbital contributions.

If you select to flip electron spin for CH₃ rather than for NH₂ when loading their Gaussian output files, you will see the alpha and beta results of CDA and ECDA are exchanged, but the result for total electrons remains unchanged.

Now select 2 and input 6, you will see both the composition of the 6th alpha orbital and the 6th beta orbital of the complex are printed on screen. Though the complex is a closed-shell system and

thus the two orbitals are essentially the same, owing to the alpha and beta FOs in the two fragments are not equivalent, the printed compositions have slight difference. Then input 0 to return.

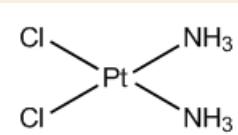
Select option 5 to enter the menu for plotting orbital interaction diagram. By option 5 in this menu, you can switch the spin of the orbitals that the diagram will be plotted for. We select it once to switch the status to "Beta", then choose option 3 and input -30,10 to change the lower and upper limits of the diagram to -30 and 10 eV, respectively. Then select option 1 to plot the orbital interaction diagram. From the graph it is very clear that, beta orbital 3 and 4 of CH_3NH_2 are formed by mixing beta FO 2 in CH_3 and beta FO 2 in NH_2 . To illustrate this point more intuitively, corresponding part is extracted from the whole diagram, and the orbital isosurfaces are attached on it, see below



It can be seen that, beta complex orbital 3 shows bonding character (this is why $r(3)$ is a positive value, namely 0.103), which is constructed by slightly mixing beta FO 2 of CH_3 into beta FO 2 of NH_2 with the same wavefunction phase. Beta complex orbital 4 is an anti-bonding orbital (this explained why $r(4)$ is a negative value, namely -0.056), formation of which is due to the mixture of beta FO 2 of NH_2 into beta FO 2 of CH_3 in terms of different phase.

4.16.3 More than two fragments case: $\text{Pt}(\text{NH}_3)_2\text{Cl}_2$

In this section I use cisplatin (see below) as instance to show how to perform CDA analysis on the system consisting of more than two fragments.



We will define Pt^{2+} cation, $(\text{Cl}_2)^{2-}$ anion and $(\text{NH}_3)_2$ as fragment 1, 2 and 3 respectively. This definition of fragments is the best choice for present system. Note that the Pt^{2+} cation is in $3d^8$ low-spin configuration, so that all fragments are in closed-shell status and thus we will not need to analyze alpha and beta spins separately.

The related Gaussian input files and resulting .fch files have been provided in *examples|CDA|Pt(NH3)2Cl2* folder. Lanl2DZ pseudo-potential basis set was used for Pt, cc-pVDZ was used for other atoms. B3LYP functional was chosen as the theoretical method. The geometry of the complex has already been optimized. In the input file of Pt^{2+} , "scf=xqc" keyword was used to

solve the unconvengence problem, but this is not needed in general.

Note 1: For certain types of basis sets, such as Pople series of basis sets (e.g. 6-31G*), by default Gaussian employs Cartesian type basis functions rather than spherical-harmonic basis functions, which may results in inconsistency problem of basis set between complex and fragment calculations. If you are not very familiar with Gaussian, I highly recommend you always add "5d 7f" keyword in all Gaussian input files when mixed basis set is used in the calculation of the complex.

Note 2: The sequence of the fragments is crucial. Because in the input file of complex, the atomic sequence is Pt--Cl₂--(NH₃)₂, we should not for example define Pt²⁺, (NH₃)₂ and (Cl₂)²⁻ as fragment 1, 2 and 3, respectively.

Now boot up Multiwfn and input:

```
examples\cda\Pt(NH3)2Cl2\Pt(NH3)2Cl2.fch // Complex
```

```
16 // Enter CDA module
```

```
3 // We define three fragments
```

```
examples\cda\Pt(NH3)2Cl2\Pt.fch // Fragment 1
```

```
examples\cda\Pt(NH3)2Cl2\Cl2.fch // Fragment 2
```

```
examples\cda\Pt(NH3)2Cl2\NH3.fch // Fragment 3
```

Then choose option 0 and input 1,2 to output CDA analysis result for fragment pair 1-2. Similarly, we get the CDA result for fragment pair 1-3 and 2-3. We cannot obtain ECDA result for present system because ECDA is only applicable to two-fragment cases. The total CDA results are summarized below.

	1-2	1-3	2-3
<i>d</i>	0.0017	0.0538	0.0538
<i>b</i>	0.5156	0.0071	0.0071
<i>d-b</i>	-0.5139	-0.0467	0.0467
<i>r</i>	0.0581	-0.1415	-0.1415

The table shows that there are 0.5156 and 0.0071 net electrons transferred from (Cl₂)²⁻ and (NH₃)₂ to Pt²⁺, respectively. Basically no electrons are donated from Pt²⁺ to its ligands. This result is in line with our chemical intuition because according to classical theory, the electrons of coordinate bonds are purely contributed from ligands. The charge transferation between (Cl₂)²⁻ and (NH₃)₂ is trivial, mainly due to there is no direct chemical interaction between them. Note that the *r* term between fragment 2 and 3 is a small but non-negligible negative value, exhibiting that there is detectable electron repulsion effect between the ligands.

Deeper analyses on this system are left for you. You can discuss orbital contribution to charge transferation based on the detail output of CDA analysis, use option 2 to study the composition of complex orbital in fragment orbital basis, and use option 5 to plot orbital interaction diagram.

4.17 Basin analysis

Below I will show how to use basin analysis module of Multiwfn to perform basin analysis for several molecules and for various real space functions. Related theory, numerical algorithms and the usage of this module have been detailedly introduced in Section 3.20. If you are not familiar with the concept of basin, please consult Section 3.20.1 first. While if you want to know more detail about basin analysis, please consult Sections 3.20.2 and 3.20.3.

You should know that Multiwfn uses a grid-based method (in particular, near-grid method) to locate attractors, generate and integrate basins; in other words, most tasks realized in the basin

analysis module rely on grid data. This is why in below sections "grid data" is frequently mentioned.

4.17.1 AIM basin analysis for HCN and Li₆

In this example we will analyze basins of electron density (also known as AIM basins) for HCN molecule, which is a very representative molecule. Only at final part of this section, I will also show how to carry out basin analysis of electron density for Li₆ atomic cluster, since this is a special case, in which there are some "pseudoatoms".

After you carefully read this section, I believe you will understand most of basic operations of the basin analysis module in Multiwfn.

Generate basins and locate attractors

Boot up Multiwfn and input following commands

examples\HCN.wfn

17 // Basin analysis

1 // Generate basins and locate attractors

1 // The grid data to be calculated and thus analyzed is for electron density

2 // Medium quality grid. This is enough for most cases, if you want to obtain a better result, you can choose "High quality grid", but much more computational time will be spent

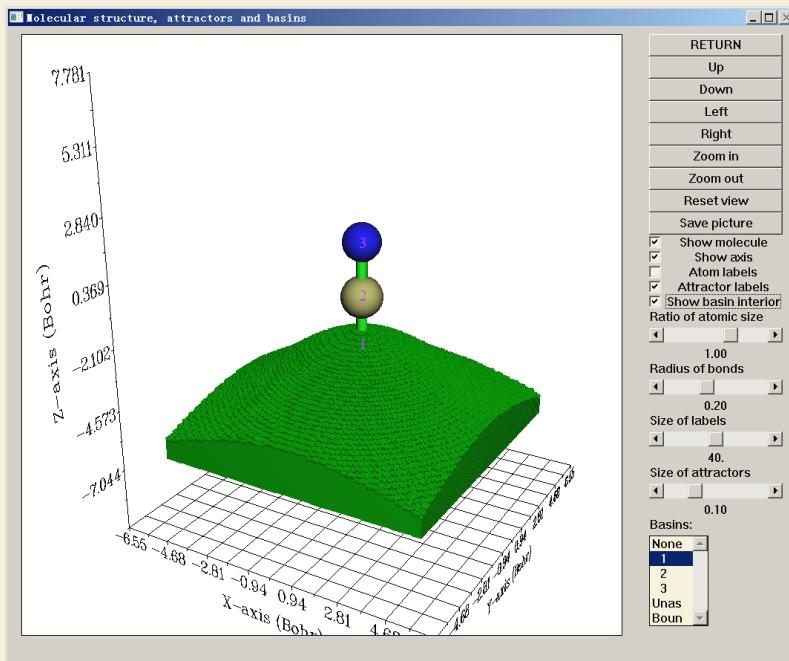
Multiwfn now starts to calculate grid data for electron density, and then generates basins and locates attractors based on the grid data; soon you will see attractor information, as shown below (notice that the attractor sequence may be different to your actual case if parallel mode is enabled, similarly hereinafter):

Attractor	X, Y, Z coordinate (Angstrom)	value
1	-0.02645886 -0.02645886 -1.52058663	0.33959944
2	0.02645886 0.02645886 -0.51514986	49.48609717
3	-0.02645886 -0.02645886 0.64904009	76.55832377

From now on you can check the information of the located attractors anytime by option -3.

Visualization of basins and attractors

Now choose 0 to visualize the basins and attractors. The purple labels indicate the index of the located attractors. Since attractors of electron density are very close to nuclei, if you want to see them, you should deselect "Show molecule" check box first. Here we select 1 from the basin list at the right-bottom corner, the basin corresponding to attractor 1 will be immediately shown. By default only interbasin part of the basin is shown, if you want to inspect the whole basin you should select "Show basin interior" checkbox. After we select it, the GUI will look like below:



Now click "RETURN" button to close the GUI.

Hint: If you are not satisfied with the graph shown in the Multiwfn GUI, you can choose function -4 to export the attractors as .pdb file, and choose option -5 to export the basins as cube file, then you can easily plot them in your favourite visualization software such as VMD. The isosurface representing the basin can be visualized by plotting an isosurface with isovalue of 0.5. Note that When Multiwfn asking you "If output internal region of the basin?", if you input "n" rather than "y", only interbasin surface will be shown when you plotting the isosurface.

You may have noticed below statements in command-line window

The number of unassigned grids: 0

The number of grids travelled to box boundary: 0

Commonly the number of these two types of grid should be zero, only in rare cases they are not zero; in such cases, you can visualize them by respectively selecting "Unas" and "Boun" in the basin list of the GUI. These grids do not belong to any basin, and generally they lack of physical meaning; to understand when and why they occur please consult Section 3.20.2.

Integrating basins

Next, we calculate the integral of electron density (electron population number) in these basins. Select function 2, then you will see many options. Each option with the serial ≥ 1 corresponds to a real space function; if you select one of them, corresponding real space function will be integrated in the generated basins. In present example we can select option 1, which corresponds to electron density. However, since we have already calculated grid data for electron density, and the grid data of electron density thus have been stored in memory, we can directly use it rather than recompute it again to reduce computational time, so here we select option 0 to use "The values of the grid data stored in memory".

Since electron density at the grids is not needed to be re-evaluated, the integrals are outputted immediately:

#Basin	Integral(a.u.)	volume(a.u.^3)
1	0.7356142812	441.70000000
2	5.3511723358	566.84900000
3	7.9020047534	829.72600000

Sum of above values: 13.98879137

There should be totally $1+6+7=14$ electrons in present system; unfortunately the sum of the integrals of electron density is 13.98879, which evidently deviates from the ideal value!

Because the basin we are studying is AIM basin, the best choice to obtain the basin integral is using function 7 rather than function 2. In function 7, mixed atomic-center and uniform grid is used, while function 2 only employs uniform grid to integrate. We input:

```
7 // Integrate real space functions in AIM basins with mixed type of grids
I // Integrate a specific function with atomic-center + uniform grids
I // Select electron density as the integrand
```

The result is

#Basin	Integral(a.u.)	vol(Bohr^3)	vol(rho>0.001)
1	0.7356461193	441.700	34.884
2	5.3552283604	566.849	102.832
3	7.9086490986	829.726	134.228

Sum of above integrals: 13.99952358

Sum of basin volumes (rho>0.001): 271.944 Bohr^3

As you can see, the sum of the integrals of electron density (13.9995) is almost exactly identical to the expected value 14.0, obviously the result is much better than using function 2. The basin volumes are also outputted. The terms "vol(Bohr^2)" do not have clear physical meaning, since they are directly affected by the spatial range of grid setting. However, the terms "vol(rho>0.001)" are useful, they exhibit the volume of the basin enclosed by the isosurface of electron density > 0.001 (Bader's vdW surface) and thus can be regarded as atomic sizes.

The atomic charge (AIM charge) of these atoms and their volumes are also outputted

```
1 (C ) Charge: 0.644589 volume: 102.832 Bohr^3
2 (N ) Charge: -0.908918 volume: 134.228 Bohr^3
3 (H ) Charge: 0.264329 volume: 34.884 Bohr^3
```

Note that above AIM charges are inaccurate! To obtain more accurate integrals in AIM basins, you should select 2 or 3 in function 7; compared to 1, they will refine the assignment of basin boundary to improve the integration result, but additional computational cost must be afforded. Here we try it, select option 2 in function 7, then input *I*, the result is

```
1 (C ) Charge: 0.748979 volume: 100.160 Bohr^3
2 (N ) Charge: -1.003917 volume: 136.304 Bohr^3
3 (H ) Charge: 0.254938 volume: 35.480 Bohr^3
```

Although the sum of the integral of electron density is still 13.9995, the charge of the atoms are somewhat varied (more accurate than before).

To further improve the integration accuracy, when generating basins one should select a grid setting better than "medium quality grid", e.g. "High quality grid" or even "Lunatic quality grid". But bear in mind that for large systems, "high quality grid" (or a better one) may consumes very large amount of computational time and memory.

Although as we have seen the integration accuracy of function 7 is much better than function 2, the former is only applicable to AIM basins, while the latter can be used for any type of basin (e.g. ELF basin).

Note: If you used option 2 or 3 in function 7, during the boundary grid refinement process, the assignment of basin boundary will be updated permanently, that means the result of later analysis (e.g. calculating LI/DI, electric multipole moment) will become more accurate.

Hint: In summary, the general steps for obtaining reliable AIM charges after you entered basin analysis module is

```
1 // Generate basin  
1 // Electron density  
2 // Medium quality grid. Select high quality grid if you wish to get more accurate result  
7 // Integrate real space functions in AIM basins with mixed type of grids  
2 // Integrate and meantime refine basin boundary  
1 // Electron density
```

Calculate electric multipole moments in basins

Enter function 8 to calculate electric multipole moments in the AIM basins (you can also use 3 to do this, but the accuracy is much lower). Only the result for carbon is pasted below

```
Result of atom    1 (C )  
Basin electric monopole moment: -5.250849  
Basin electric dipole moment:  
X= 0.000003 Y= 0.000003 Z= 1.095846 Magnitude= 1.095846  
Basin electron contribution to molecular dipole moment:  
X= 0.000003 Y= 0.000003 Z= 6.026238 Magnitude= 6.026238  
Basin electric quadrupole moment (Cartesian form):  
QXX= -0.759571 QXY= 0.000000 QXZ= -0.000002  
QYX= 0.000000 QYY= -0.759571 QYZ= -0.000002  
QZX= -0.000002 QZY= -0.000002 QZZ= 1.519142  
The magnitude of electric quadrupole moment (Cartesian form): 1.519142  
Electric quadrupole moments (Spherical harmonic form):  
Q_2,0 = 1.519142 Q_2,-1= -0.000002 Q_2,1= -0.000002  
Q_2,-2= 0.000000 Q_2,2 = 0.000000  
Magnitude: |Q_2|= 1.519142
```

The formulae used to evaluate these terms are basically identical to the ones given in Section 3.18.3, the only differences are that in that formulae the nuclear positions should be replaced by attractor positions, and the ranges of integration should be basins rather than fuzzy atomic spaces.

The electric monopole moment (-5.251) is just the negative value of electron population number in the basin. Z-component of electric dipole moment of the basin is a positive value (1.096), suggesting that in basin 2, most of electrons are distributed in the regions where Z-coordinate is more negative than attractor 2. The ZZ-component of basin electric quadrupole moment is positive (1.519), while the other diagonal components are negative, indicating that relative to attractor 2, electron cloud in this basin contracts in Z-direction but elongates in other directions.

Calculate localization index and delocalization index in basins

Delocalization index (DI) is a quantitative measure of the number of electrons delocalized (or say shared) between two regions, while localization index (LI) quantitatively measures how many electron are localized in a region. For details about LI and DI please see Section 3.18.5. The only difference relative to the statements in that section is that here the LI and DI will be calculated based on basins rather than based on fuzzy atomic spaces.

Enter function 4, Multiwf will start to calculate LI and DI, then you will see:

```
***** Total delocalization index matrix *****
```

	1	2	3
1	0.97290584	0.90037928	0.07252656
2	0.90037928	3.49303345	2.59265416
3	0.07252656	2.59265416	2.66518072

Total localization index:

```
1: 0.259    2: 3.496    3: 6.658
```

Since present molecule is a close shell system, only total LI and DI are outputted, the LI and DI for α and β electrons are not outputted separately. As you can see, between basin 2 and 3, which correspond to AIM atomic space of C and N respectively, the DI is 2.593, exhibiting that in average there are 2.593 electrons shared by the two atoms. To some extent DI can be regarded as covalent bond order, the DI value 2.593 is indeed comparable with the formal bond order (3.0) between C and N in HCN. The diagonal terms are the sums of the elements in corresponding row/column, for closed-shell cases they can be somewhat considered as atomic valency. So we can say in HCN the nitrogen atom has atomic valency of 2.665.

The LI of basin 1 is only 0.259 which conspicuously deviates from the basin electron population number. This observation reflects that in HCN, the electron in the AIM atomic space of hydrogen can easily delocalize out.

Special case: Basis analysis when pseudoatoms are presented

Pseudoatom is also known as non-nuclear attractor (NNA) of electron density, it refers to maximum of electron density that are not at nuclear position. There are various reasons that can cause the NNAs, for example, existence of metal bond or quality of wavefunction is too poor. Here I use Li₆ cluster as example to illustrate how to deal with the case when NNAs are presented.

Boot up Multiwf and input

examples\Li6.fch

17 // Basin analysis

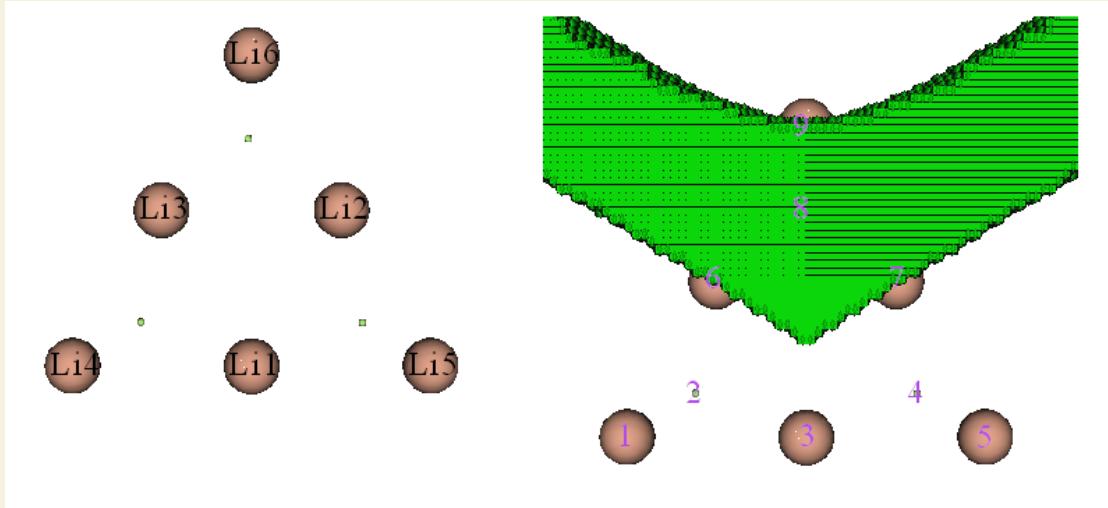
1 // Generate basins and locate attractors

1 // Electron density

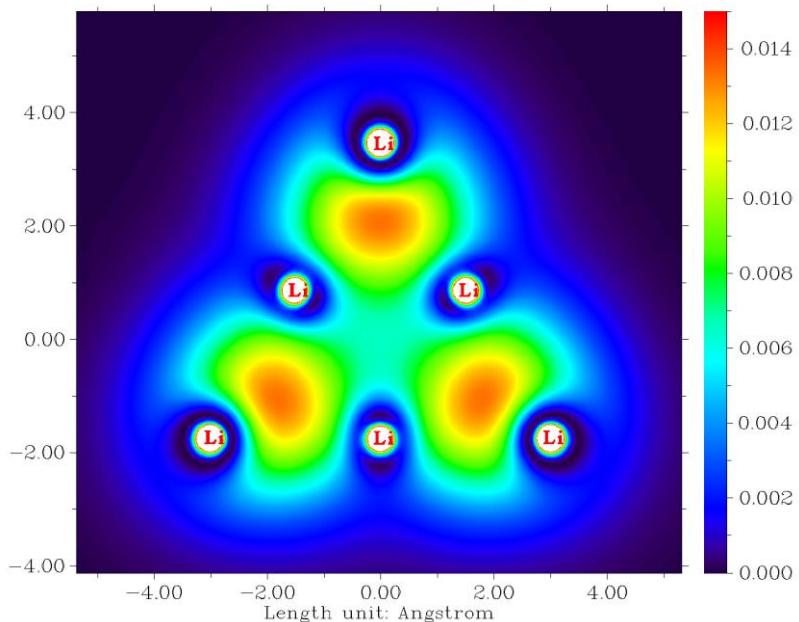
1 // For illustration purpose, here we only use low quality grid for saving time

0 // Visualize attractors and basins

Left part of below graph shows geometry of the cluster, the three green spheres indicate position of the three NNAs; right part of the graph displays corresponding basin of one of NNAs. As you can see, attractors 2, 4, 8 are NNAs.



If you plot color-filled map of valence electron density for the Li_6 via the steps illustrated in Section 4.6.2, you will immediately understand why there exists NNAs at center of the boundary Li triangles. From below graph you can clearly see that at center of each boundary triangle there is indeed a maximum of electron density, this observation also implies presence of three-center bonds



Next, we calculate population number of the AIM basins. Input below commands

```
7 // Integrate real space functions in AIM basins with mixed type of grids
2 // Integrate and meantime refine basin boundary
1 // Electron density
```

During integrating the basins, calculation will pause three times and meantime you will find prompt like below on screen. This is because program does not know how to properly deal with the three NNAs, namely attractors 2, 4, 8:

Warning: Unable to determine the attractor 2 belongs to which atom!

If this is a non-nuclear attractor, simply press ENTER button to continue. If you used pseudopotential and this attractor corresponds to the cluster of all max

ima of its valence electron, then input the index of this atom (e.g. 9). Else you should input q to return and regenerate basins with smaller grid spacing

Since we already know that attractors 2, 4, 8 are regular NNAs, according to the prompt, what we should do is simply pressing ENTER button to continue the calculation. Finally, you will find below output

```
Normalization factor of the integral of electron density is 0.999992
```

```
The atomic charges after normalization and atomic volumes:
```

2 (NNA)	Charge:	-1.249592	Volume:	273.680 Bohr^3
4 (NNA)	Charge:	-1.249536	Volume:	273.680 Bohr^3
8 (NNA)	Charge:	-1.249079	Volume:	274.048 Bohr^3
1 (Li)	Charge:	0.757502	Volume:	69.792 Bohr^3
2 (Li)	Charge:	0.756323	Volume:	70.000 Bohr^3
3 (Li)	Charge:	0.756331	Volume:	70.000 Bohr^3
4 (Li)	Charge:	0.492816	Volume:	139.296 Bohr^3
5 (Li)	Charge:	0.492759	Volume:	139.296 Bohr^3
6 (Li)	Charge:	0.492477	Volume:	139.392 Bohr^3

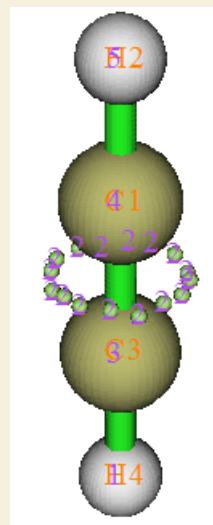
As can be seen, each NNA basin carries 1.249 electrons, therefore charge of the basin is -1.249. When NNAs are presented, it is clearly unable to rigorously define AIM atomic charges, since sum of all AIM atomic charges will be unequal to net charge of the whole system. This is one of severe limitations of AIM atomic charge.

4.17.2 ELF basin analysis for acetylene

This time we analyze ELF basin for acetylene. Boot up Multiwfn and input examples\c2h2.wfn

```
17 // Basin analysis
1 // Generate basins and locate attractors
9 // ELF
2 // Medium quality grid
0 // Visualize attractors and basins
```

In the GUI, as the graph shown below, you can see there are lots of closely placed attractors encircling the C-C bond. In fact, they have very similar ELF values, and collectively represent the ring-type ELF attractor. These attractors have been clustered together by Multiwfn automatically, therefore all of them have identical attractor index, namely 2; in another word, attractor 2 is a "degenerate" attractor, which has many member attractors. Correspondingly, basin 2 is composed by the member basins. The interested user is recommended to visualize basin 2 in the GUI.



The presence of attractor 2 signifies the π electrons shared by the two carbons. According to the well-known ELF symbolic method, basin 2 should be identified as $V(C_1, C_3)$, which means this basin is comprised by the valence electrons of C_1 and C_3 .

Attractor 3 and 4 correspond to core-type ELF attractors, their basins should be identified as $C(C_3)$ and $C(C_1)$, respectively, where the letter C out of parentheses stands for "Core". If you deselect the "Show molecule" check box and select corresponding terms in the basin list then you can visualize the basins. The graph shown below portrays $C(C_3)$.



Although basin 1 and 5 cover the whole hydrogen spaces, they should be identified as $V(C_3, H_4)$ and $V(C_1, H_2)$, rather than $C(H_4)$ and $C(H_2)$, respectively. This is because hydrogen does not have core electrons.

Now close the GUI window by clicking "RETURN" button, and then input

```
2 // Integrate a real space functions in the basins
1 // Electron density
```

Soon, we get the integrals, namely the average electron population number in each basin:

#Basin	Integral(a.u.)	volume(a.u.^3)
1	2.2159821115	768.39800000
2	5.3670485473	972.79700000

4 Tutorials and Examples

3	2.0949016050	0.83200000
4	2.0949016050	0.83200000
5	2.2159824751	768.55500000

Sum of above values: 13.98881634

Both C(C₁) and C(C₃) in average contain 2.095 electrons, which is in line with the fact that carbon has two electrons in its core. Also, the average population number in V(C₃,H₄) and V(C₁,H₂) are close to two, approximately reflecting that in average there is a pair of electron shared between C and H.

According to classical chemical bond theory there are three electron pairs and hence six electrons are shared by the two carbons, however in the V(C₁,C₃) basin the integral is only 5.37. Although the deviation is relatively large, this is a normal situation. It is senseless to require that the result of ELF basin analysis must be able to reproduce classical Lewis picture, and actually, ELF analysis is more advanced and more close to real physical picture.

Next, enter function 3 to calculate electric multipole moments. We want the result for all basins to be outputted, so we input -1. The result of C(C₁) is pasted below

```
Basin      4
Basin electric monopole moment: -2.094902
Basin electric dipole moment:
X= -0.104745 Y= -0.104745 Z= 0.020728 Magnitude= 0.149575
Basin electron contribution to molecular dipole moment:
X= 0.000000 Y= 0.000000 Z= -2.388409 Magnitude= 2.388409
Basin electric quadrupole moment (Cartesian form):
QXX= -0.003034 QXY= -0.007856 QXZ= 0.001555
QYX= -0.007856 QYY= -0.003034 QYZ= 0.001555
QZX= 0.001555 QZY= 0.001555 QZZ= 0.006067
The magnitude of electric quadrupole moment (Cartesian form): 0.006067
Electric quadrupole moments (Spherical harmonic form):
Q_2,0 = 0.006067 Q_2,-1= 0.001795 Q_2,1= 0.001795
Q_2,-2= -0.009071 Q_2,2 = 0.000000
Magnitude: |Q_2|= 0.011205
```

First you should note that although in present case, due to the symmetry, the X and Y components of the basin electric dipole moment should vanish, the actual values are not so close to zero, rendering that the integration accuracy is not very high. This is why "high quality grid" is often necessary when electric multipole moment analysis is involved. However, present result is still useful on the qualitative level. The magnitude of electric quadrupole moment in the basin measures how evidently does the electron distribution in the basin deviate from spherical symmetry. This value for C(C₁) is very small (0.0061), showing that the distribution of core electron of the carbon basically remains unperturbed during the formation of the molecule.

Note: If the basins you are interested in are only valence basins, "medium quality grid" is enough for electric multipole moment analysis, since valency density is not as high as core density, and hence does not need high accuracy of integration.

Input 0 to return. Finally, let's choose option 4 to study LI and DI. The result is shown below

***** Total delocalization index matrix *****

1	2	3	4	5
---	---	---	---	---

1	1.31231709	1.03496561	0.16085935	0.02291204	0.09358009
2	1.03496561	2.71090548	0.32048710	0.32048709	1.03496568
3	0.16085935	0.32048710	0.51131492	0.00705643	0.02291205
4	0.02291204	0.32048709	0.00705643	0.51131492	0.16085937
5	0.09358009	1.03496568	0.02291205	0.16085937	1.31231719

Total localization index:

1: 1.560 2: 4.011 3: 1.834 4: 1.834 5: 1.560

Between C(C₁) and C(C₃), namely DI(3,4), the value is trivial, reflecting the general rule that the electron delocalization between atomic cores is rather difficult. DI(1,3) and DI(2,3) are very small values but not close to zero, representing that the electrons in C(C₃) have a few probability to exchange with the ones in V(C₃,H₄) and V(C₁,C₃), which are the only two basins adjacent to C(C₃).

DI(2,1) and DI(2,5) are about 1.0, such a large value indicates that the electron delocalization between C-C bonding region and C-H bonding region is easy. Though the average electron population number in C(C₁) and C(C₃) are 2.095, their LI values are as high as 1.834, implying that the core electrons of carbon highly prefer to stay in the core region rather than delocalize out. In contrast, for V(C,H) and V(C,H), their LI values are less than their average electron population numbers distinctly, revealing that the electrons in these basins do not express very strong localization character.

In the ELF basin analyses, some researchers prefer to use the concept of variance (σ^2) and covariance (Cov) rather than LI and DI to discuss problems. Covariance of electron pair fluctuation is simply the half of the negative value of DI, for example, Cov(2,5) = -DI(2,5)/2 = -1.035/2 = -0.518. The variance of electronic fluctuation can be calculated as the half of the corresponding diagonal term of the DI matrix outputted by Multiwfn, for instance, $\sigma^2(2) = \text{DI}(2,2)/2 = 2.710/2 = 1.355$ (note that as mentioned earlier, the diagonal terms of the DI matrix outputted by Multiwfn are the sums of the elements in corresponding row/column).

4.17.3 Basin analysis of electrostatic potential for H₂O

In literatures, basin analysis technique is rarely applied on electrostatic potential (ESP). To illustrate the universality of basin analysis module of Multiwfn, in this example we will generate ESP basins for H₂O, and then evaluate electron population numbers in them. In the first part of this section, we will let Multiwfn directly calculate ESP and electron density, while in the second part we will make use of the cube files produced by third-part program to do the same things.

It is noteworthy that unlike electron density and ELF, which we have analyzed earlier, ESP has positive part and negative part simultaneously. For such real space functions, Multiwfn will automatically locate attractors (maxima) for positive part and locate "repulsors" (minima) for negative part, but in Multiwfn all of them are collectively recorded as "attractors". You can easily discriminate them by inspecting their colors in the GUI, as illustrated below.

Part 1

Boot up Multiwfn and input following commands:

examples\H2O.fch // Optimized and produced at HF/6-31G* level. Of course you can also use .wfn/.wfx file

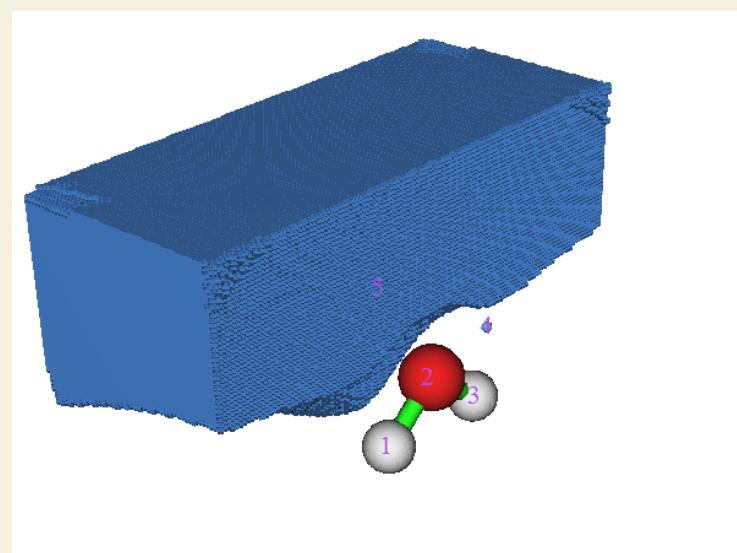
```

1
12 // ESP
2 // Medium quality grid. Note that calculation of ESP is rather time-consuming, so do not
use too high quality of grid, especially for large systems

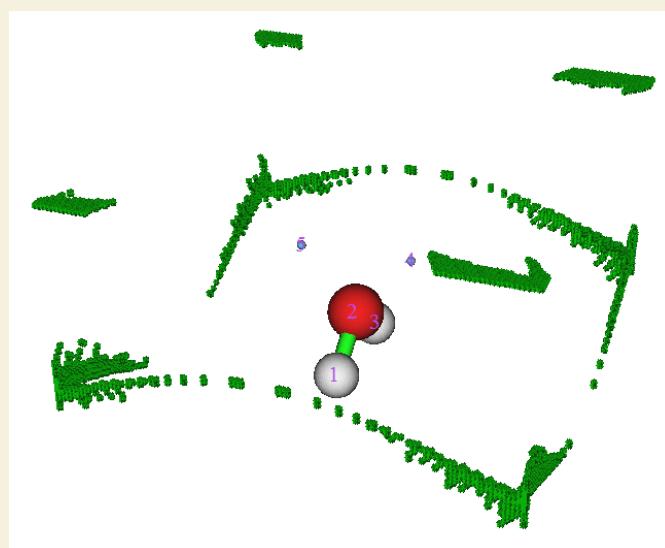
```

After the calculation of ESP grid data and the basin generation are finished, five attractors are founded. Notice that this time the number of grids travelled to box boundary is not zero again (about 1627 grids).

Choose option 0 to open GUI, you can see that attractor 1, 2 and 3 correspond to ESP maxima due to nuclear charges. It is well known that the lone pairs of oxygen make the ESP at corresponding region negative, clearly attractor 4 and 5 reflected this effect. Attractor 4 and 5 are colored by light blue because they are lying in negative region, in fact they are not attractors but "repulsors" (minima) of ESP. By clicking term 5 in the basin list and checking "show basin interior" box, you will see below graph, which exhibits the basin corresponding to attractor 5.



If you want to visualize which grids have travelled to box boundary during basin generation, you can select "Boun" in the basin list, see below graph. Evidently, these grids lack of physical meaning and hence can be simply ignored. They only present at the regions far from atoms.



Now click "RETURN" button to close the GUI.

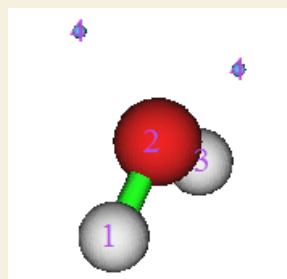
Sometimes one needs to obtain geometry information between attractors and nuclei. As an instance, we enter function -2, and input *a1 c4*, the distance between nucleus of atom 1 (namely oxygen) and attractor 4 will be shown on the screen, the value is 2.237 Bohr. Next, input *c4 a1 c5*, then the angle of "attractor 2 -- atom 1 -- attractor 3" will be outputted, the value is 80.84 degree, which in some sense can be regarded as the angle between the two lone pairs.

Input *q* to exit the geometry measurement interface.

Assume that we want to cluster attractor 4 and 5 together as a degenerate attractor, we can input

```
-6 // Set parameter for attractor clustering or manually perform clustering
3 // Cluster specified attractors
4,5 // attractor 2 and 3 will be clustered as a single one
0 // Return
```

Select option 0 to open GUI, as shown below, you can find that the index of all attractors have changed, and the two attractors corresponding to the oxygen lone pairs now sharing the same index, namely 4.



Close the GUI by clicking "RETURN" button, choose option 2 and then select 1 to integrate electron density in the ESP basins, the result is

#Basin	Integral(a.u.)	volume(a.u.^3)
1	0.8647179457	391.74200000
2	7.5147940814	25.01400000
3	0.8647179450	391.65000000
4	0.7405385242	703.11200000

Sum of above values: 9.98476850
 Integral of the grids travelled to box boundary: 0.00000002

Since currently basin 4 is just the whole negative ESP region, the result shows that in average there are 0.740 electrons in the negative ESP region. Indeed this value is not large (one may expects that there should be about four electrons due to the two lone pairs), this is because most parts of the molecular space are dominated by nuclear charges and hence having positive ESP.

Part 2

Now we redo some analyses in part 1 but using the grid data comes from external file. Since external grid data is supported, one can use third-part programs to generate the grid data, and then use basins analysis module of Multiwfn to study the basins of the real space functions recorded in the grid data. The real space function to be analyzed may be the one cannot be directly calculated by Multiwfn, or the one requires large amount of computational time, e.g. ESP (compared to some

ab initio programs, using current Multiwfn to calculate ESP grid data is a bit slow).

Here we use *cubegen* utility in Gaussian package to yield grid data of ESP and electron density for the H2O.fch file. Run below commands in Windows cmd or Linux or Mac OS console to produce density.cub and ESP.cub. *cubegen* utility is assumed to be placed in current path

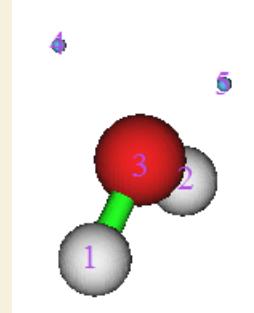
```
cubegen 0 fdensity H2O.fch density.cub 0 h
cubegen 0 potential H2O.fch ESP.cub 0 h
```

Then boot up Multiwfn and input

ESP.cub // This file contains ESP grid data. After loading it, the grid data will be stored in memory

```
17
I
2 // Generate the basins by using the grid data stored in memory
```

Now visualize the attractors in function 0



Exit GUI, after that input following commands

```
2 // Integrate real space functions in the basins
-I // Use the grid data stored in external file as integrand
density.cub // This file contains electron density grid data
```

The result is very close to the one we obtained in Part 1 of present section. For example, the electron population number in negative ESP region we get here is $0.367*2=0.734$, while the counterpart one we obtained in Part 1 is 0.740.

At last, we select function 3 to calculate electric multipole moments for the basins. Because cube file does not contain GTF (Gaussian type function) information, you will be prompted to input the path of a file containing GTF information of present system, so that electric multipole moments can be calculated. We input the path of the H2O.fch file, and then input *-I*, the electric multipole moments of all basins will be immediately outputted on screen.

NOTE: The calculation speed of ESP of *cubegen* utility in Gaussian package is much faster than Multiwfn. If you have Gaussian installed on your system, when you want to carry out basin analysis for ESP, the best way is setting *cubegenpath* parameter in settings.ini file to actual path of *cubegen*, then at the step of calculating ESP grid data in the basin analysis, *cubegen* will be automatically invoked by Multiwfn to evaluate ESP, the time cost will hence be significantly reduced compared to using internal code of Multiwfn to evaluate ESP grid data. More information about this point can be found in Section 5.7.

4.17.4 Basin analysis of electron density difference for H₂O

In this example we analyze basins of electron density difference for H₂O to quantitatively study the electron density deformation during formation of the molecule.

Before doing the basin analysis, we need to generate grid data of electron density difference first by main function 5, wavefunction file of all related elements must be available. Here we directly use the set of atomic wavefunction files provided in Multiwfn package, namely copying "atomwfn" subfolder in "example" folder to current folder, then during generating grid data of electron density difference Multiwfn will automatically use them. There are several different ways to prepare atomic wavefunction files, please recall Section 4.4.7 and consult Section 3.7.3.

After that, boot up Multiwfn and input:

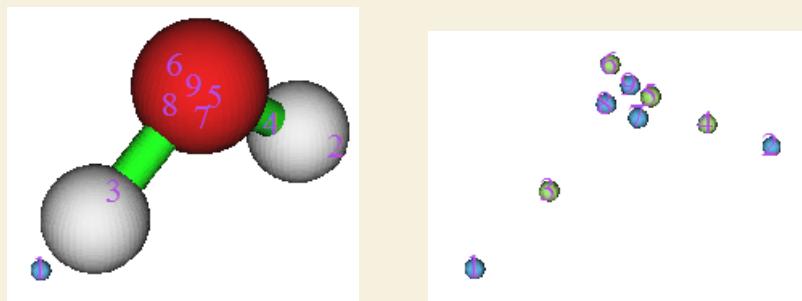
examples\H2O.fch

```
5 // Calculate grid data
-2 // Obtain deformation property
1 // Electron density
```

3 // High quality grid. Because the variation of electron density difference is complicated, using relatively high quality of grid is compulsory. Note that the "high quality grid" we selected here only defines the total number of grids, and hence has different meaning to the one involved in function 1 of basin analysis module

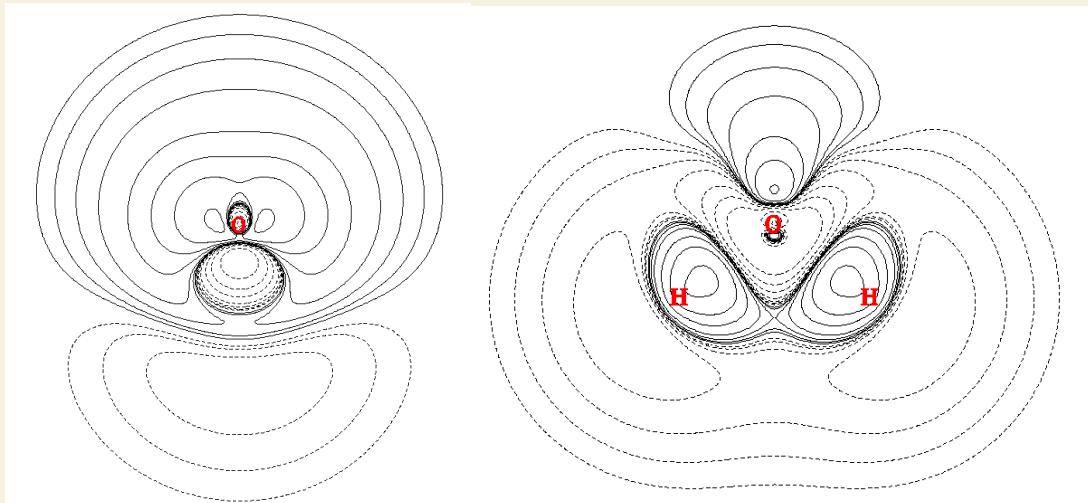
```
0 // After the calculation is finished, return to main menu
17 // Basin analysis module
1 // Generate basins and locate attractors
2 // Generate the basins by using the grid data stored in memory (namely the grid data we just calculated by main function 5)
```

Enter function 0 to visualize the result, you will see the left graph shown below. After deselect "Show molecule", the graph will look like the right one



Positive (negative) part of electron density difference corresponds to the region where electron density is increased (decreased) after formation of the molecule. Light green spheres denote the maxima of the positive part, while light blue ones denote the minima of the negative part.

If you feel difficult to imagine why the maxima and minima distribute like this, I suggest you to plot plane maps for electron density difference. The left graph shown below is the electron density difference map vertical to the molecular plane, while the right graph is the map in the molecular plane.

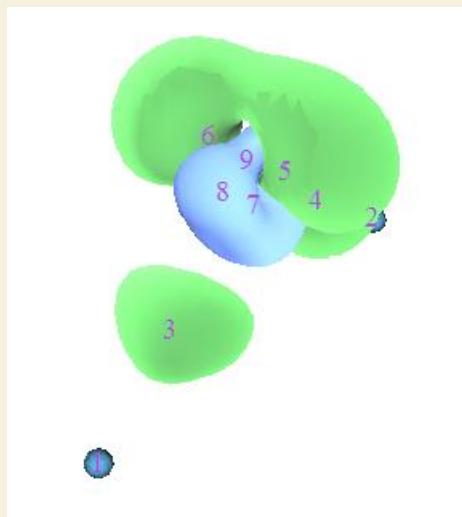


By comparing the attractors with plane maps, it is evident that attractor 3 and 4 are the maxima of the region where electron density is enhanced due to the formation of the O-H bonds. While the presence of attractor 5 and 6 mainly arise from the electron aggregation due to the formation of the lone pairs.

Note: In this example there are missing and misplaced attractors in negative regions (in fact they are repulsors). As you can see, attractor 8 does not have its counterpart in another side of molecular symmetry plane. Besides, attractor 7 deviates from its ideal position. The reason of these problems is that the grid quality is still not high enough relative to the complicated characteristic of electron density difference. However, since these attractors are not of chemical interested, these problems can be simply ignored.

It is interesting to examine how many electrons are aggregated between C and H due to the bond formation. There can be many ways to measure this quantity; the most reasonable one for present case is to integrate the electron density difference in basin 3 or in basin 4. Let's do this now. Choose function 2, and then select option 0 to take the grid data of electron density difference as integrand. From the output we can find that the integral is about 0.09e.

If you would like to compare the attractors with the isosurface of electron density difference, you can simply choose -10 to return to main menu of Multiwfn, and then choose suboption -2 in main function 13 to plot the isosurface of the grid data stored in memory, the attractors we located will be shown together, as shown below (isovalue=0.05), where green and blue parts correspond to positive and negative regions, respectively.

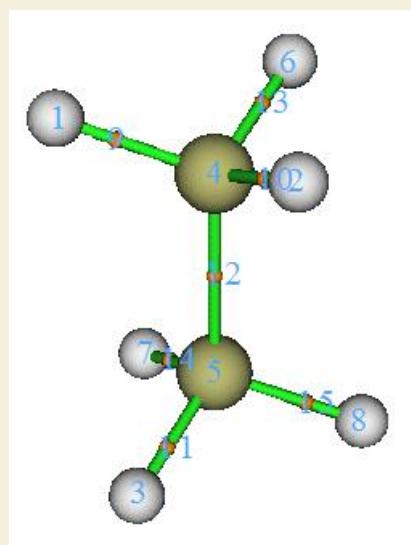


4.17.5 Study source function in AIM basins

Source function has been briefed in part 19 of Section 2.6. Commonly, bond critical point (BCP) is taken as the reference point of source function when bonding problem is discussed. In this example we calculate source function in AIM basins for ethane; in particular, based on source function we will get the contribution from methyl group to the electron density at the BCP of its C-H bond. Before calculating source function we should perform topology analysis first to find out the position of the BCP.

Boot up Multiwfn and input:

```
examples\ethane.wfn // Optimized and produced at B3LYP/6-31G*
2 // Topology analysis
2 // Search nuclear critical points
3 // Search BCPs
0 // Visualize result, see below
```



Critical point 11 will be selected as the reference point of the source function. Of course, selecting which BCP is completely arbitrary. Now close the GUI of topology analysis module

You'd better choose option 7 and then input 11 to check the electron density at this CP, because theoretically the integral of source function in the whole space should equal to the electron density at its reference point, therefore this value is important to examine if the integration of source function is accurate enough. The electron density at CP11 is 0.276277.

From the information shown in the command-line window you can find the coordinate of CP11 is (0.0,-1.199262548,-1.909104063), copy it from the window to clipboard (if you do not know how to do this please consult Section 5.4). Next, we will set CP11 as the reference point of the source function. Although you can define reference point by "refxyz" parameters in settings.ini, there is a trick can do the same thing, by which you needn't to close Multiwfn and then reboot it to make the parameters take effect!

Input below commands

-10 // Return to main menu from topology analysis module

1000 // Secret menu

1 // Set reference point

Paste the coordinate of CP11 to the window and then press ENTER button.

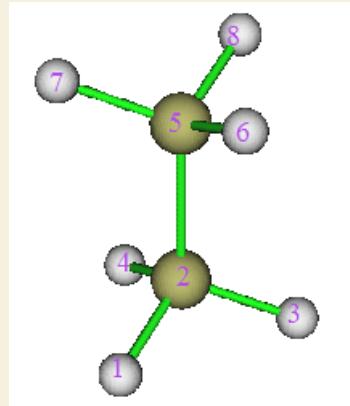
17 // Basin analysis

1 // Generate basins and locate attractors

1 // Electron density

2 // Medium quality grid

Enter GUI by choosing function 0, you will see



Now we integrate source function in the AIM basins. Input following commands

7 // Integrate real space functions in AIM basins with mixed type of grids

1 // Integrate a specific function with atomic-center + uniform grids

19 // Source function

The result is

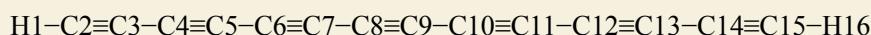
Atom	Basin	Integral(a.u.)	Vol(Bohr^3)	Vol(rho>0.001)
1 (C)	5	0.00361772	149.307	70.392
2 (H)	8	0.00289355	468.732	50.058
3 (H)	6	0.00280041	443.982	50.097
4 (H)	7	0.00280276	418.343	50.095
5 (C)	2	0.12224052	150.514	70.396
6 (H)	1	0.12040142	475.383	50.054
7 (H)	4	0.01051821	424.105	50.095
8 (H)	3	0.01051742	450.072	50.097

```
Sum of above integrals:          0.27579202
Sum of basin volumes (rho>0.001): 441.284 Bohr^3
```

The sum of the integrals is very close to the electron density at CP13 (0.276277). The sum of the integral in basin of 1, 2, 3 and 4 is $0.1204+0.1222+0.0105*2=0.2636$, which represents the integral in the space of methyl group and accounts for $0.2636/0.2758*100\%=95.6\%$ of the total integral value, exhibiting that methyl group is the main source of the electron density of the BCP of its C-H bonds.

4.17.6 Local region basin analysis for polyyne

Sometimes, the geometry of the system we studied is rather extended, for instance, polyyne C_{14}H_2 , which can be formally illustrated as



If we are only interested in the electronic structure characteristic of local region in this system, by properly setting up grid, basin analysis can be conducted only for the interesting region rather than for the whole system to save computational time. As an example, in this section we will try to acquire electron population number in the ELF basin of $V(\text{C}_7,\text{C}_8)$ and $V(\text{C}_8,\text{C}_9)$ with minimum computational cost.

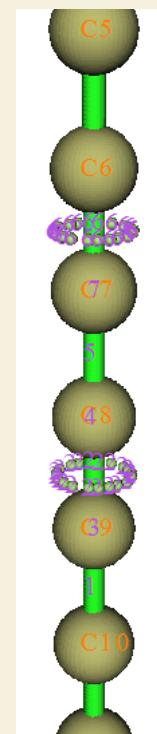
Boot up Multiwfn and input following commands:

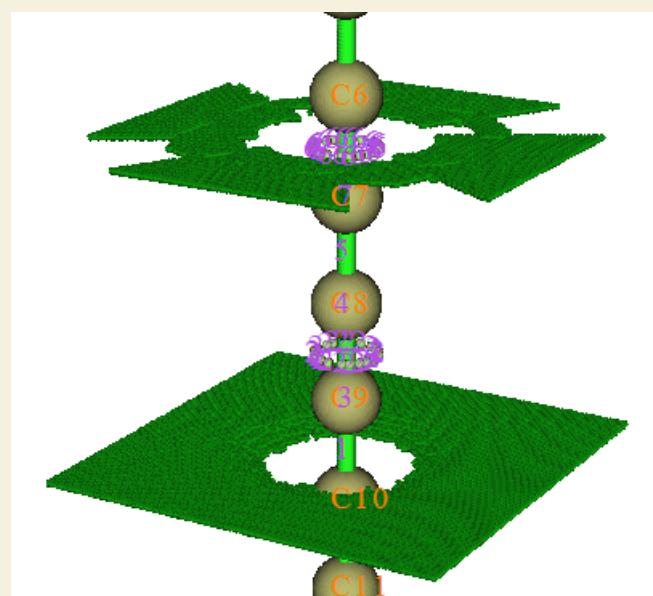
```
examples\polyyne.wfn // Optimized and produced under B3LYP/6-31G*
17 // Basin analysis
1 // Generate basins and locate attractors
9 // ELF
8 // Set the grid by inputting center coordinate, grid spacing and box length
a8 // Take the position of atom 8 as box center
0.08 // Grid spacing (Bohr)
```

10,10,8 // Box length in X, Y and Z directions (Bohr). Note that current molecule is aligned in Z-axis. Obviously, the larger the box, the longer the computational time must be spent. While the box should not be too small, otherwise the basins of interest may be truncated. Choosing appropriate box size highly relies on users' experience

After the calculation is finished, enter GUI by selecting option 0, you will see the graph shown at the right side. Clearly, only several attractors near C8 are located. Basin 5 and basin 21 correspond to $V(\text{C}_7,\text{C}_8)$ and $V(\text{C}_8,\text{C}_9)$, respectively. Notice that although attractor 1 and 6 are also located, due to their corresponding basins are not only large but also close to box boundary, it can be expected basin 1 and 6 are severely truncated and hence studying them are meaningless.

When you use above manner to study local region, you will always find there are many grids travelled to box boundary. In present example, as shown in command-line window, the number of such type of grids is 60668. You can visualize them by choosing "Boun" in basin list of GUI, see the graph below





Now close GUI, select option 2 and then select 1, the result shows that the integral of electron density in $V(C_7, C_8)$ and $V(C_8, C_9)$ are 2.78 and 5.02, respectively. Evidently, the bonding between C8-C9 is much stronger than C7-C8, this is why the bond length of the former (1.236\AA) is shorter than the latter (1.338\AA). Note that the electron population number in $V(C, C)$ of acetylene is 5.37 (see Section 4.17.2), therefore it can be expected that C8-C9 is weaker than typical C-C triple bond, mostly due to the electron global conjugation in polyyne.

The interested users can redo the basin analysis for the whole system with using the same grid spacing (0.08 Bohr), the computational amount will be much larger than current example. For $V(C_7, C_8)$ and $V(C_8, C_9)$, you will find the result does not differ from the one we obtained above detectably.

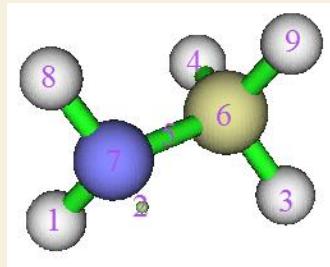
4.17.7 Evaluate atomic contribution to population of ELF basins

In this section, I will use CH_3NH_2 as example to show how to obtain contribution of C and N to population of $V(\text{C}, \text{N})$ ELF bond basin based on AIM partition of atomic space, this is useful to examine bond polarity. You can also use the similar way to obtain atomic contribution to population of any other kind of basins (e.g. LOL basin, ESP basin).

First, we need to generate a cube file named `basin.cub`, whose grid value corresponds to index of ELF basins. Boot up Multiwfn and input

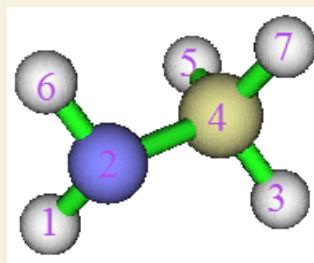
```
examples\CH3NH2.wfn
17 // Basin analysis
1 // Generate basins and locate attractors
9 // ELF
2 // Medium quality grid
```

Now enter option 0 to examine the basin index



Evidently basin 5 corresponds to V(N,C), which is the one we will study. Then close the GUI and input

```
-5 // Export basin as cube file
0,0 // Export basin.cub in current folder, whose grid value directly defines ELF basin
Next, we generate AIM basins as usual, the grid setting must be exactly identical to basin.cub
1 // Regenerate basins
1 // Select real space function
1 // Electron density
9 // Use grid setting of another cube file, this is the safest way to ensure the grid data to be
generated has the same grid setting as basin.cub
basin.cub
0 // Check attractors
```



It is clear that the attractor index corresponding to N and C are 2 and 4, respectively. Then we evaluate atomic contribution to population of the basins defined in basin.cub

```
9 // Then program loads basin.cub in current folder
2 // The index of the attractor corresponding to N
5 // The 5th ELF basin, i.e. V(N,C) basin
The result is 1.15866, namely N contributes 1.159 electrons to V(N,C) basin. Then input
4 // The index of the attractor corresponding to C
5 // The 5th ELF basin, i.e. V(N,C) basin
```

From the result we know that C contributes 0.463 electrons to the V(N,C) basin.

Since N contributes much more electrons than C to their ELF bond basin, it may thus be concluded that C-N is a bond with significant polarity.

4.18 Electron excitation analysis

Main function 18 of Multiwfn is very powerful, it is a collection of electron excitation analysis methods and able to provide very deep insight into all aspects of electron transition characters. In this section I will illustrate most of them using practical instances. Before following below examples,

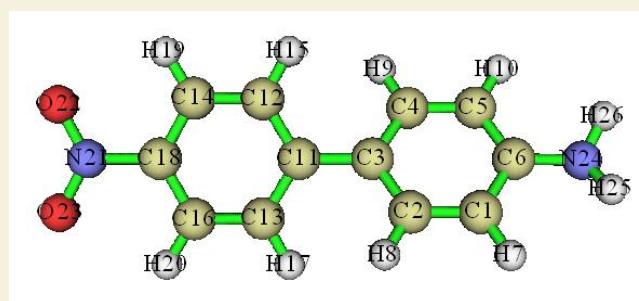
please at least first read beginning of Section 3.21 to understand requirement on the input files. Although I illustrate the analyses based on Gaussian output files, these functions are by no means limited to Gaussian users!

4.18.1 Using hole-electron analysis to fully characterize electron excitations

The hole-electron analysis module of Multiwfn is quite powerful, it is able to present very comprehensive characterization for all kinds of electron excitations. If you are not familiar with basic theories and ideas of hole-electron analysis, please read Section 3.21.1. The requirement on input file for the hole-electron analysis has been described at the beginning of Section 3.21, please carefully check it. Below I will employ two systems to illustrate the use of the hole-electron analysis, the first one is a typical donor- π -acceptor system, the second one is a typical coordinate. The Chinese version of this section is my blog article "Using Multiwfn to perform hole-electron analysis to fully investigate electronic excitation character" (<http://sobereva.com/434>), in which there is an additional example, namely studying Rydberg excitation of H₂CO.

4.18.1.1 Example 1: NH₂-biphenyl-NO₂

The example in this section is quite long, please carefully and patiently read it. In this section I will take the NH₂-biphenyl-NO₂ as example, its geometry is shown below.



In this system the biphenyl moiety behaves as a π -linker, and it is well-known that the nitro group and amino group act as electron acceptor and donor during electron excitation, respectively, therefore it is expected that there must be some charge-transfer (CT) states corresponding to overall electron displacement from the amino group side to the nitro group side.

Preparation

Here I assume that you are a Gaussian user (other quantum chemistry codes users can also utilize the hole-electron analysis). We optimize the geometry at the B3LYP/6-31G* level, then carry out TDDFT calculation using below settings (the input file has been provided as *examples\excit\D-pi-A.gjf*), then five lowest singlet excited states will be evaluated. Note that *IOp(9/40=4)* must be specified, the reason has been clearly mentioned at the beginning of Section 3.21. The CAM-B3LYP is employed here because it is able to faithfully represent CT excitations.

```
%chk=D-pi-A.chk
# CAM-B3LYP/6-31g(d) TD(nstates=5) IOp(9/40=4)
```

After calculation, convert the .chk file to .fch/fchk file, the resulting .fchk file has been provided as *examples\excit\D-pi-A.fchk*. The output file of this task has been provided as *examples\excit\D-pi-A.out*.

Examining quantitative indices defined in the hole-electron analysis framework

Boot up Multiwfn and input below command

```
examples\excit\D-pi-A.fchk
```

```
18 // Electron excitation analysis
```

```
1 // Hole-electron analysis
```

```
examples\excit\D-pi-A.out
```

```
1 // Study excitation between ground state (S0) and the first excited state (S1)
```

```
1 // Calculate distribution of hole, electron and so on as well as various indices
```

2 // Medium quality grid (this is suited for small and medium sized systems. For large systems, you should use at least "high quality grid", or manually input a proper grid spacing)

Once calculation is finished, you will find below information on screen. The data (except for the excitation energy) are calculated by grid-based integration. Clearly, for the same system, the higher number of grids, the better accuracy of the data.

```
Integral of hole: 1.000343
Integral of electron: 0.999793
Integral of transition density: -0.000033
Transition dipole moment in X/Y/Z: 0.444063 -0.000186 -0.001753 a.u.
Sm index (integral of Sm function): 0.27369 a.u.
Sr index (integral of Sr function): 0.51896 a.u.
Centroid of hole in X/Y/Z: -4.531729 0.000425 0.003252 Angstrom
Centroid of electron in X/Y/Z: -4.010033 0.001760 0.002643 Angstrom
D_x: 0.522 D_y: 0.001 D_z: 0.001 D index: 0.522 Angstrom
Variation of dipole moment with respect to ground state:
X: -0.985930 Y: -0.002523 Z: 0.001150 Norm: 0.985934 a.u.
RMSD of hole in X/Y/Z: 1.443 1.160 0.437 Norm: 1.902 Angstrom
RMSD of electron in X/Y/Z: 1.596 0.974 0.634 Norm: 1.974 Angstrom
Difference between RMSD of hole and electron (delta sigma):
X: 0.153 Y: -0.186 Z: 0.197 Overall: 0.072 Angstrom
H_x: 1.520 H_y: 1.067 H_z: 0.536 H_CT: 1.520 H index: 1.938 Angstrom
t index: -0.998 Angstrom
Ghost-hunter index (def 1): -18.018 ev, 1st/2nd terms: 9.583 27.601 ev
Ghost-hunter index (def 2): -17.466 ev, 1st/2nd terms: 10.135 27.601 ev
Excitation energy of this state: 3.907 ev
```

In the output, the "Integral of hole" and "Integral of electron" are the integrals of hole and electron over the whole space, respectively, they should be exactly 1.0 in theory. However, due to the unavoidable numerical integration error, the calculated values have slight deviation from 1.0. Since the deviation is extremely small, we can say that for the current excited state of the current system, the grid setting we used is completely appropriate.

Note: If you find the integral of hole or electron deviates from 1.0 evidently, then the outputted indexes will be unreliable. There are three possibilities: (1) You forgot to use $IOP(9/40=3 \text{ or } 4)$ (2) The grid quality is too low (3) The extension distance is not large enough, therefore the spatial region of the grid points does not fully cover the main distribution region of hole or electron (when Rydberg excitation is investigated, the default extension distance should always be enlarged).

The rest terms in the above output in turn are: The integral of transition density over the whole space (ideal value is 0), transition electric dipole moment, S_m and S_r indices, centroid coordinate of hole and electron, D_λ and D indices, X/Y/Z components and norm of variation of excited state dipole

moment with respect to the ground state one, RMSD (σ) of hole and electron, $\Delta\sigma_\lambda$ and $\Delta\sigma$ indices, $H_\lambda/H_{CT}/H$ indices, t index, Ghost-hunter index, excitation energy (which is loaded from Gaussian output file directly).

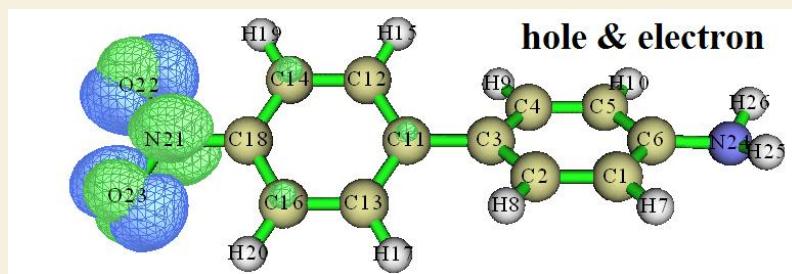
Note that the "Ghost-hunter index (def 1)" is the definition of this quantity in its original paper, while the one with "(def 2)" suffix is the form modified by me, which should be more reasonable. The two values after the "1st/2nd terms" are the first term (dependent on configuration coefficients) and second term (namely $-1/D$) of the Ghost-hunter index, respectively.

The transition dipole moment outputted above is obtained by integrating evenly distributed grids of transition dipole moment density. It can also be directly read from the Gaussian output file, the X/Y/Z components are 0.4427, -0.0005, -0.0012 a.u., which are very close to the ones output by Multiwfn, namely 0.444063, -0.000186, -0.001753. This observation further reflects that the grid setting we employed is appropriate.

For the $S_0 \rightarrow S_1$ excitation under current study, from the above output, it can be seen that the D index is merely 0.522 Å, which is obviously a very small value since it is even less than half length of a typical C-C bond. The S_r index reaches 0.519 (the theoretical upper limit is 1.0), which is a large value, implying that about half part of hole and electron has perfectly matched. So, by simply examining S_r and D indices, we are already able to judge that this excitation should be a typical local excitation (LE). Then let us look at the t index, its total value is -0.998, which is much less than 0, meaning that there is no significant separation of hole and electron distributions, futher implying that this excitation should be attributed to LE type.

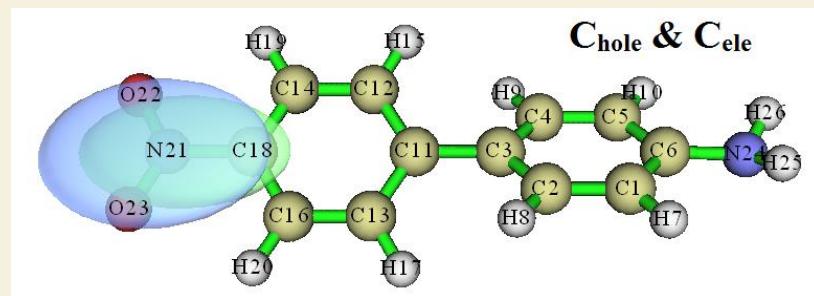
Visual study of various real space functions in the hole-electron framework

Now you should see the post-process menu on the screen. The meaning of each option is self-explanatory. Please read through each option carefully. Here we choose option 3, we will see distribution of hole and electron at the same time:



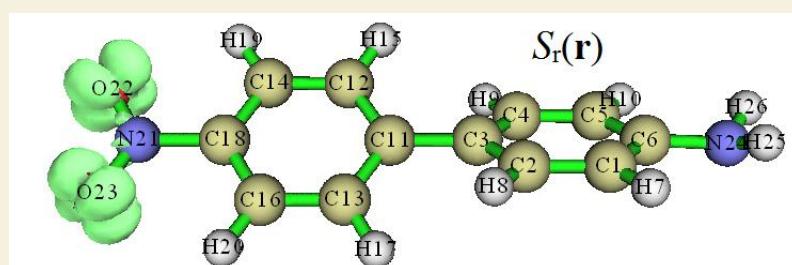
In above figures, green represents the electron distribution, and blue represents the hole distribution, isovalue has been set to 0.005. Both the hole and electron appear almost exclusively in the nitro group, so there is no doubt that $S_0 \rightarrow S_1$ is a LE excitation, well verifying our conclusion based on the D , S_r , and t indices. In addition, according to the above hole distribution map, the hole appears to be composed of lone pair orbitals of oxygens since there is one lobe on each side of each oxygen. Electron distribution has a nodal plane along the nitro group, therefore we can infer that the electron distribution should be composed of π^* orbital. Now we can draw the conclusion that that $S_0 \rightarrow S_1$ is a LE excitation with $n \rightarrow \pi^*$ feature.

Then close the graphical window and select option 8 to visualize C_{hole} and C_{ele} , which are transformed from hole and electron distributions respectively to make their distribution behavior smoother. The isosurface map is show below (In order to see clearly, transparent style is used).



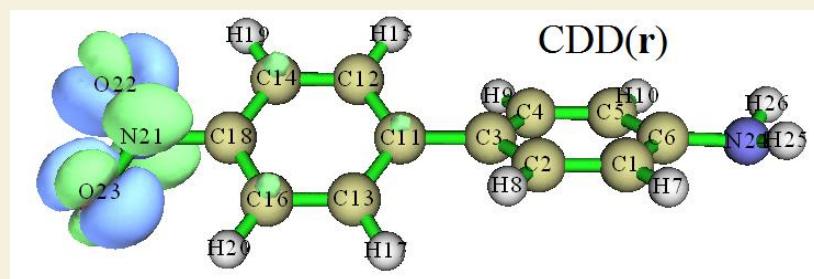
As can be seen, the graph of C_{hole} and C_{ele} look obviously more intuitive, they are very sleek and do not have any nodal character as hole and electron. Therefore using $C_{\text{hole}}/C_{\text{ele}}$ map to replace the hole/electron map is a good choice in many cases.

Next, let us take a look at the overlap function of hole and electron, namely the S_r function. Close current graphical window, select option 4 in the post-process menu, and then select option 2 to display the S_r function, you will see below map (isovalue is set to 0.005)



From the graph one can clearly find where the hole and electron overlap significantly. As can be seen, around each oxygen there are four regions where the hole and electron are highly overlapped. It is easy to understand why the S_r graph looks like this by comparing the hole and electron isosurfaces shown earlier.

Then close the window and select option 7, charge density difference (CDD) between the excited state and the ground state will be shown, see below. In this map, the isovalue is set to 0.005, green and blue correspond to increase and decrease of the excited state density with respect to the ground state density, respectively.



The CDD map and the map simultaneously showing hole and electron distributions (referred to as "hole&electron map" later) are similar, but there are also differences. The key difference is that in the CDD map, the hole and electron have been largely cancelled in their overlapping region; in contrast, in the hole&electron map, the overlapping between hole and electron can be faithfully exhibited. I think the hole&electron map is more useful than the CDD map to investigate the intrinsic characteristics of electron excitation because it directly exhibits the pristine distribution of hole and electron.

By the way, if we select option 18 in the post-process menu, the program will start to calculate the Coulomb attractive energy (also known as exciton binding energy) between the hole and electron.

The calculation is quite time-consuming even for medium sized system, so please wait patiently. The final output is:

```
Coulomb attractive energy: 0.287031 a.u. ( 7.810524 ev )
```

Examining quantitative contributions to hole and electron

Next, I demonstrate how to evaluate contribution of MOs to hole and electron. After selecting option 0 in the post-process menu to return to the hole-electron analysis interface, we select subfunction 2 and input an outputting threshold. Here we input 1, then MOs with contribution to hole or electron higher than 1% will be shown on screen:

```
MO 52, occ: 2.00000 Hole: 96.207 % Electron: 0.000 %
MO 56, occ: 2.00000 Hole: 3.415 % Electron: 0.000 %
MO 57, occ: 0.00000 Hole: 0.000 % Electron: 85.411 %
MO 59, occ: 0.00000 Hole: 0.000 % Electron: 12.222 %
MO 61, occ: 0.00000 Hole: 0.000 % Electron: 2.163 %

Sum of hole: 100.001 % Sum of electron: 100.001 %
```

It can be seen from the data that MO52 is absolutely dominant for hole, it contributes as high as 96.2%, while electron is mainly composed of MO57, with a contribution of 85.4%. This observation implies that if one discusses electron excitation solely based MO52 and MO57, although in this case the electron excitation can be qualitatively described, there are still non-negligible deviations. The "Sum of hole" and "Sum of electron" shown above are the sum of the contributions of all orbitals to hole and electron, respectively (including the terms not outputted), these two values in principle should be exactly 100%, but currently there are 0.001% error. Such a small error can be completely ignored, it comes from the fact that not all configuration coefficients are printed by Gaussian (only configuration coefficients greater than 0.0001 are requested to be outputted during Gaussian calculation via *IOp(9/40=4)*)

Then we check contribution of atoms or fragments to hole and electron. Select subfunction 3 in hole-electron analysis interface, you will see

```
The number of non-hydrogen atoms: 16

Contribution of each non-hydrogen atom to hole and electron:

1(C ) Hole: 0.19 % Electron: 0.02 % overlap: 0.07 % diff.: -0.17 %
2(C ) Hole: 0.18 % Electron: 0.48 % overlap: 0.30 % diff.: 0.30 %
3(C ) Hole: 0.59 % Electron: 0.13 % overlap: 0.28 % diff.: -0.46 %
4(C ) Hole: 0.18 % Electron: 0.49 % overlap: 0.30 % diff.: 0.32 %
5(C ) Hole: 0.19 % Electron: 0.02 % overlap: 0.06 % diff.: -0.18 %
6(C ) Hole: 0.31 % Electron: 0.37 % overlap: 0.34 % diff.: 0.06 %
11(C ) Hole: 0.15 % Electron: 4.17 % overlap: 0.78 % diff.: 4.03 %
12(C ) Hole: 0.41 % Electron: 0.14 % overlap: 0.24 % diff.: -0.26 %
13(C ) Hole: 0.37 % Electron: 0.14 % overlap: 0.23 % diff.: -0.23 %
14(C ) Hole: 0.94 % Electron: 5.03 % overlap: 2.18 % diff.: 4.09 %
16(C ) Hole: 0.92 % Electron: 5.01 % overlap: 2.15 % diff.: 4.09 %
18(C ) Hole: -0.01 % Electron: 1.16 % overlap: 0.00 % diff.: 1.17 %
21(N ) Hole: 2.39 % Electron: 33.90 % overlap: 9.00 % diff.: 31.52 %
22(O ) Hole: 46.18 % Electron: 24.38 % overlap: 33.55 % diff.: -21.80 %
23(O ) Hole: 46.12 % Electron: 24.39 % overlap: 33.54 % diff.: -21.73 %
24(N ) Hole: 0.46 % Electron: 0.15 % overlap: 0.26 % diff.: -0.32 %
```

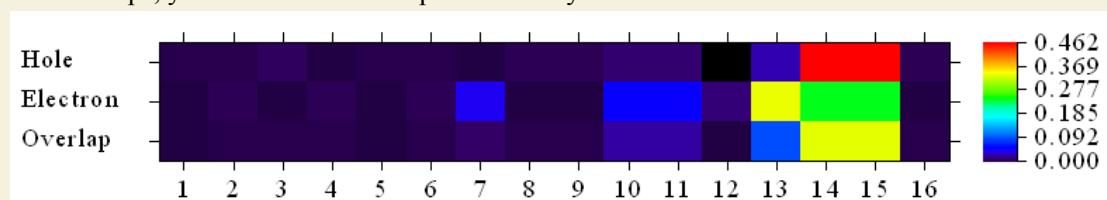
Since hydrogen atoms generally do not participate in electron excitation of chemical interest, only the information of non-hydrogen atoms is outputted, including the atomic contributions to hole, electron, hole-electron overlap, electron-hole difference (*i.e.* CDD). The indices of the atoms in the nitro group are 21, 22, and 23, it can be seen from the data that the two oxygens of the nitro group contribute most to the hole, the sum of their contributions is $2 \times 46.1 \approx 92\%$. The spatial delocalization of the electron is relatively stronger, the three atoms in the nitro group contribute a total of $2 \times 24.4 + 33.9 \approx 83\%$, the rest part of electron is basically contributed by the atoms in the biphenyl moiety.

Although the distribution characteristics of hole and electron can be examined by visualizing isosurface map of hole and electron, the observed isosurfaces are obviously dependent on the choice of isovalue. So, it is impossible to fully display the hole and electron distribution in all regions by only one image. On the contrary, the quantitative atomic contributions given above are very definitive.

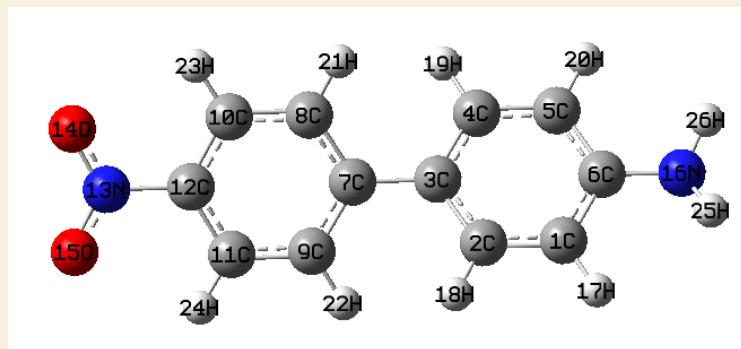
The sum of "Diff." of 21N, 22O and 23O is about -12%, which shows that the integral value of the density difference (CDD) in the nitro group is -0.12, revealing that the nitro moiety lost 0.12 electrons during electron excitation, and some of them transferred to the biphenyl moiety (if you want to investigate charge transfer amount between specific fragments, it is recommended to use the IFCT method, as illustrated in Section 4.18.8)

Showing atomic contributions to hole and electron in terms of heat map

We can also plot the atomic contributions as heat map, so that the major character can be immediately and easily captured. Select "4 Set stepsize between labels in X axis" in current menu and then input 1 to change the step of abscissa to 1, and then select "1 Plot hole/electron composition as heat map", you will see below map immediately



In the figure, the numbers in the abscissa are indices of non-hydrogen atoms. This figure describes the contribution of each non-hydrogen atom to hole, electron and their overlap by color (*e.g.* 0.4 corresponds to 40%). For example, based on the list of contributions of non-hydrogen atoms to hole/electron that shown earlier, we can know that the position 16 in the abscissa of the figure actually corresponds to atom N24. In order to more conveniently find the correspondence between the indices in the abscissa and actual atomic indices, you can open corresponding .fch/.gif/.out file by GaussView, Select "Edit" - "Atom List", then select "Edit" - "Reorder" - "All atoms: Hydrogens Last", then you will see below graph in GaussView, in which indices of all hydrogen atoms have been placed behind indices of non-hydrogen atoms. Clearly, in this case the atomic indices in the molecular structure directly corresponds to the indices in the abscissa of the heat map.

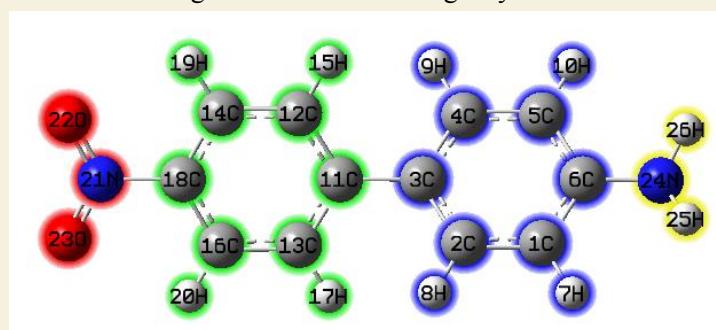


From indices in the above map, we find that the position of 13, 14 and 15 in the heat map correspond to the two oxygens and one nitrogen atoms in the nitro group, the position 16 corresponds to the nitrogen in amino group, the rest are carbons in the biphenyl moiety. In the heat map, the row corresponding to hole clearly revealed that the hole is almost solely contributed by the two oxygens in the nitro group, since corresponding matrix elements are red (large value). Electron is also mainly contributed by the nitro group, but other molecule regions also have non-negligible contributions, this is why in the row corresponding to electron, blue color appears in the matrix elements other than positions 13~15. The information conveyed by colors of the row corresponding to overlap is that there is significant overlap between hole and electron on the oxygens of the nitro groups, while the overlap in other areas of the systems is far from being so remarkable.

There are many options in the hole/electron composition analysis interface, they can be used to adjust plotting effect of the heat map, save the heat map as a image file, switch whether to include the hydrogen atoms into the heat map, and export the data to *he_atm.txt* in current folder so that you can draw the heat map yourself in other programs such as Origin. These options will not be explained one by one here, please try it yourself.

Investigating fragment contributions to hole and electron

The option -1 in the hole/electron composition analysis interface is important, it is used to load fragment definition, then fragment contribution to hole and electron will be shown, and fragment-based heat map could be drawn, which makes the discussion significantly more convenient. Here we divide the system into four fragments in the following way.



Select option "-1 Load fragment definition", and then input

4 // Define four fragments

21-23 // The atomic indices of fragment 1 (the nitro group)

11-20 // The atomic indices of fragment 2 (the benzene neighbouring to nitro group)

1-10 // The atomic indices of fragment 3 (the benzene neighbouring to amino group)

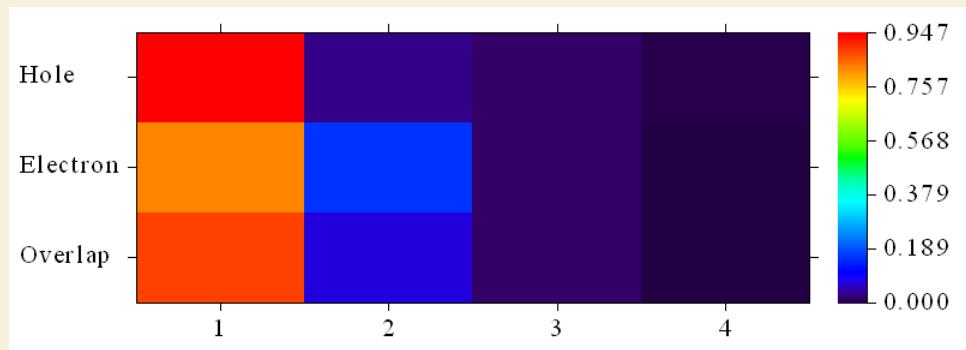
24-26 // The atomic indices of fragment 4 (the amino group)

You will see below output immediately

Contribution of each fragment to hole and electron:

```
# 1 Hole: 94.68 % Electron: 82.67 % overlap: 88.47 % diff.: -12.01 %
# 2 Hole: 3.20 % Electron: 15.67 % overlap: 7.09 % diff.: 12.47 %
# 3 Hole: 1.64 % Electron: 1.53 % overlap: 1.59 % diff.: -0.12 %
# 4 Hole: 0.47 % Electron: 0.13 % overlap: 0.25 % diff.: -0.34 %
```

The data shows that 94.68% of hole is located on the nitro group, while 82.67% and 15.67% of electron are located on the nitro group and neighbouring benzene ring, respectively. The degree of overlap between hole and electron on the nitro group is about 90%. Since "Diff." of the nitro group is -12.01%, and the excitation under current investigation is a single electron excitation, therefore, it can be said that the electron on the nitro group is reduced by 0.1201 during the electron excitation process, while the benzene ring neighbouring to the nitro group gained 0.1247 electrons. Then, if you want to make representation of the above data more intuitive, we can select option 1 to draw the fragment-based heat map. The abscissa at this time corresponds to fragment index, as shown below:



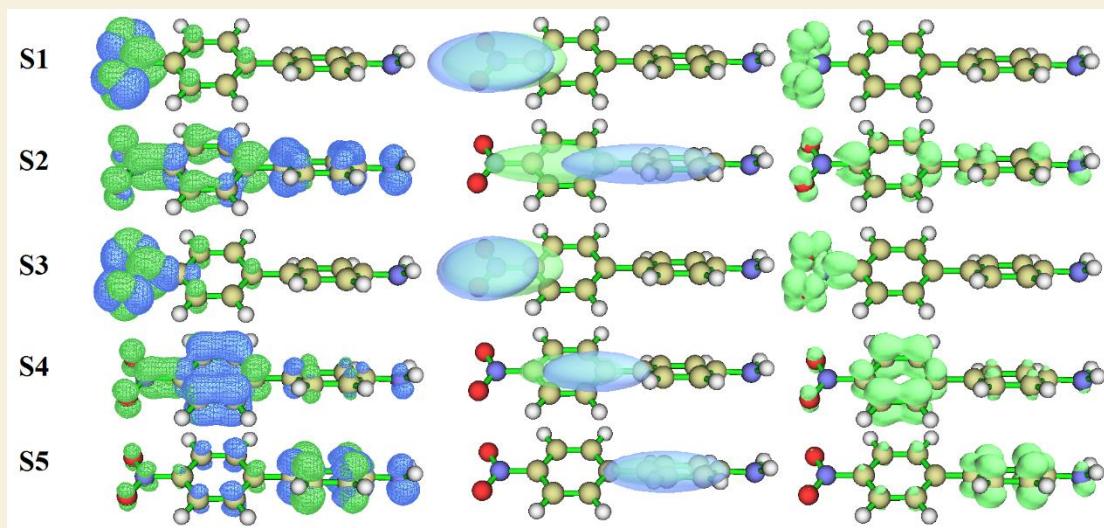
It can be seen from the figure that the spatial distribution range of electron is larger than hole.

Collective comparsion of all electron excitations

At this point, various analyses in the hole-electron framework for the S0→S1 excitation of the NH₂-biphenyl-NO₂ system have been completely completed. If you also want to analyze other excited states, return to the menu of main function 18 via option 0, then enter the hole-electron analysis function again, and then select the corresponding excited state. Here we put together the D , S_r , H , t index and hole-electron Coulomb attraction energy of all the five excited states calculated in this system:

	D (Å)	S_r	H (Å)	t (Å)	E_{coul} (eV)
S0→S1	0.52	0.52	1.94	-1.00	7.81
S0→S2	3.48	0.65	3.15	0.56	4.71
S0→S3	0.57	0.55	1.70	-0.68	8.54
S0→S4	0.97	0.87	2.88	-1.55	5.56
S0→S5	0.54	0.87	2.93	-2.06	5.56

Below are hole&electron map, $C_{\text{hole}}\&C_{\text{ele}}$ map and S_r function map of all the five excitations. The isovalue are set to 0.003. It can be seen that the $C_{\text{hole}}\&C_{\text{ele}}$ map can always display the main distribution features of the hole&electron map in a clearer and more intuitive way. However, many details are lost during the transformation; for example it is impossible to determine the specific type of the electron excitation (such as n-π*, π-π*) solely based on the $C_{\text{hole}}\&C_{\text{ele}}$ map.



Now we look at the indices in combination with the isosurface graph. For the D index, only the $S_0 \rightarrow S_2$ value has very large (3.48 \AA), so it can obviously be considered as CT excitation. Indeed, from above graph it can be seen that the distance between the center of the blue and green isosurfaces (namely centroids of C_{hole} and C_{ele}) is large. While for other excitations, the centers of the blue and green isosurfaces are close together, therefore they should be regarded as LE excitations.

Then we look at the S_r index. We find that the S_r indices of all excited states are relatively large. In particular, the values of the $S_0 \rightarrow S_4$ and $S_0 \rightarrow S_5$ are rather large, up to 0.87, the main reason is that these two excitations are highly localized $\pi-\pi^*$ type of excitation on the benzene ring. It is worth to mention that although $S_0 \rightarrow S_1$ is also a highly localized excitation, its S_r (0.52) is even smaller than the S_r (0.65) of $S_0 \rightarrow S_2$, which is a CT excitation. The reason why the S_r index of the $S_0 \rightarrow S_1$ is not as large as expectation is not difficult to understand. As mentioned earlier, $S_0 \rightarrow S_1$ shows $n \rightarrow \pi^*$ feature, the main body of lone pair is on the NO_2 plane, while π^* orbital has a nodal plane on the NO_2 plane, hence the overlap of hole and electron should be limited.

Next, look at the H index, which reflects the breadth of the average distribution of hole and electron. It can be seen from the hole&electron map that both the hole and electron of $S_0 \rightarrow S_1$ and $S_0 \rightarrow S_3$ are distributed in a local region, this is why their H indices are not large. Since the distribution of hole and electron corresponding to the excitations from S_0 to S_2 , S_4 and S_5 are evidently wider than $S_0 \rightarrow S_1$, their H indices are evidently larger.

One can see that only the t index of $S_0 \rightarrow S_2$ is a slightly positive value, indicating that the separation of hole and electron is obvious, so it is more reasonable to consider $S_0 \rightarrow S_2$ as a CT excitation. The t indices corresponding to the excitations from S_0 to other excited states are evident negative values, suggesting that degree of separation of their hole and electron is very low.

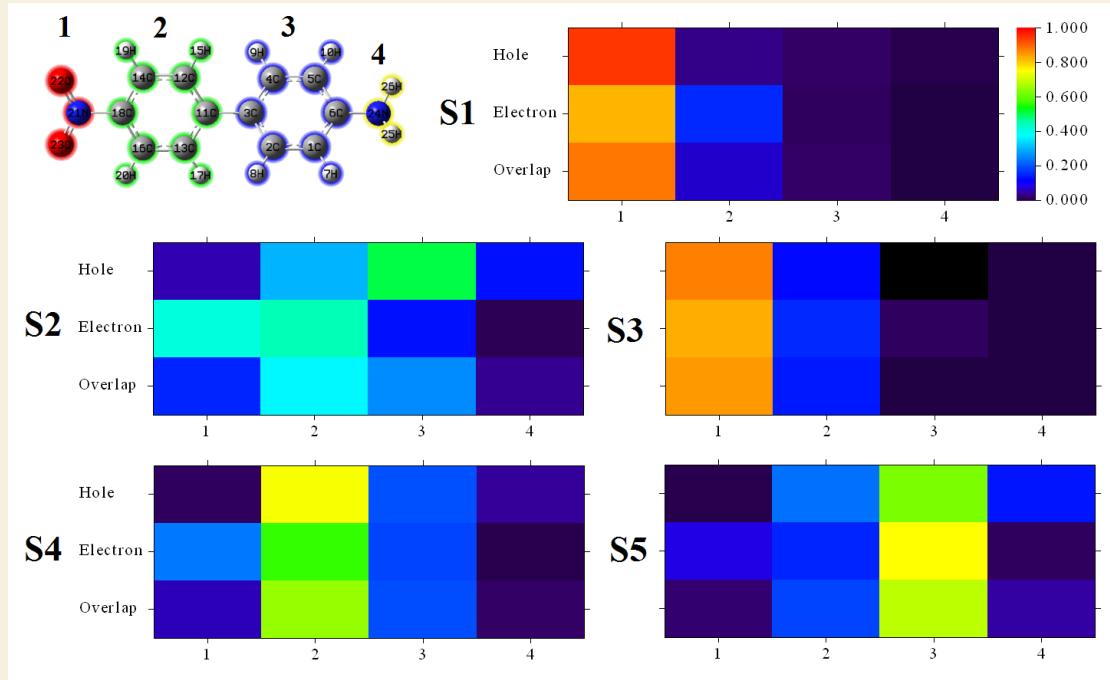
Combined with the above isosurface maps and quantitative data, we can identify the characteristics of the five excitations:

- $S_0 \rightarrow S_1$: LE excitation of $n-\pi^*$ type on the nitro group
- $S_0 \rightarrow S_2$: CT excitation of $\pi-\pi^*$ type from amino group towards nitro group
- $S_0 \rightarrow S_3$: The same as $S_0 \rightarrow S_1$
- $S_0 \rightarrow S_4$: $\pi-\pi^*$ LE excitation occurring on the benzene ring attached to the nitro group
- $S_0 \rightarrow S_5$: $\pi-\pi^*$ LE excitation occurring on the benzene ring attached to the amino group

The hole-electron Coulomb attractive energy given in the table is closely related to the electron excitation characteristics, and the most influential factor should be the D index. It is easy to

understand that the larger the D index is, the farther the distance between the main distribution region of hole and electron, and thus the weaker the Coulomb attractive energy. From the data, it is indeed found that Coulomb attraction energy of $S0 \rightarrow S2$ (the only CT excitation) is the smallest one of that of all the five electron excitations. While for the excitations of $S0 \rightarrow S1$ and $S0 \rightarrow S3$, since their D indices are very small and according to the H index the spatial extent of their hole and electron is very narrow, one can easily imagine that the corresponding Coulomb attraction should be very strong. Indeed, as can be seen from the previous table, their hole-electron Coulomb attraction energies are the most negative ones (-7.81 and -8.54 eV).

Here, the heat maps that exhibit contributions to the hole, electron and overlap from all fragments for all the five excitations are given together. In order to facilitate parallel comparison, the color scale for all excited states is uniformly set to 0.0~1.0.



From these heat maps, one can immediately make clear where the excited electrons come and where they go by viewing color of the matrix elements. For example, from the graph of $S0 \rightarrow S2$, one can easily recognize that the excited electron is mainly originated from fragment 3 (the benzene attached to amino group), most of them is transferred to fragment 1 (the nitro group), and a smaller part is transferred to fragment 2 (the benzene attached to the nitro group). Another example, from the $S0 \rightarrow S4$ map one can find that the excited electrons come from fragment 2, after excitation most of them remain in fragment 2, but a few of them transferred to fragment 1, and the overlap between hole and electron on fragment 2 is significantly higher than the other regions.

It is worth to note that the detailed characteristics of charge transfer between fragments during electron excitation can be even better revealed by the IFCT method, see example in Section 4.18.8.

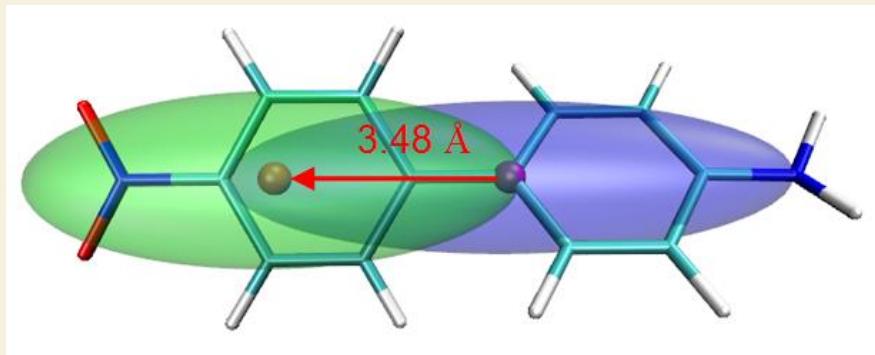
Plotting grid data in VMD

The grid data of hole, electron, C_{ele} , C_{hole} and so on can be exported to cube files by corresponding options in the post-process menu, and then you can render them in VMD program to get better visualization effect. If you do not know how to do, please refer to the description of the operations in VMD at the end of Section 4.18.3. Furthermore, one can first use VMD to plot isosurface map for C_{ele} and C_{hole} , and then input such as below command to draw centroids of hole

and electron as purple and orange spheres to make the graph more informative

```
draw color purple
draw sphere {1.411500 -0.007015 -0.025494} radius 0.25 resolution 20
draw color orange
draw sphere {-2.069784 -0.000346 -0.000830} radius 0.25 resolution 20
```

For the S0→S2 excitation, the graph plotted in above way in VMD is shown below



In order to make understanding easier, an arrow is appended on the graph to highlight the CT direction, the *D* index is also labelled together to make the graph more informative.

Tip: Obtaining a variety of indices for a range of excited states via script

By Linux shell script, a variety of indices for a range of excited states can be obtained at once. For example, we want to obtain all indices for excitations of S0→S1,S2,S3, we should copy the the input files *D-pi-A.fchk* and *D-pi-A.out* as well as the *all_index.sh* from the "examples\excit" folder to a proper folder, then enter this folder in the Linux terminal, run *chmod +x ./batch.sh* to add executable permission, then run *./all_index.sh* to execute the script. Each excitation will be analyzed in turn, the status will be shown on screen until the appearence of "Finished!". Then open the resulting *result.txt* file in current folder, you will find

```
1 Sr index (integral of sr function): 0.51896 a.u.
2 Sr index (integral of sr function): 0.64906 a.u.
3 Sr index (integral of sr function): 0.54538 a.u.

1 D_x: 0.522 D_y: 0.001 D_z: 0.001 D index: 0.522 Angstrom
2 D_x: 3.481 D_y: 0.007 D_z: 0.025 D index: 3.481 Angstrom
3 D_x: 0.574 D_y: 0.001 D_z: 0.001 D index: 0.574 Angstrom

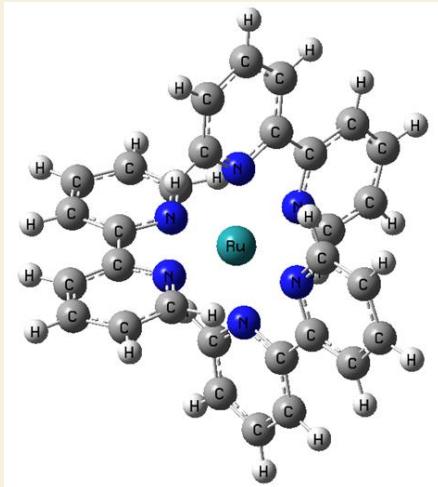
1 RMSD of hole in x/Y/Z:      1.443  1.160  0.437  Norm:  1.902 Angstrom
2 RMSD of hole in x/Y/Z:      3.055  0.826  0.740  Norm:  3.251 Angstrom
3 RMSD of hole in x/Y/Z:      0.984  1.032  0.416  Norm:  1.486 Angstrom

[ignored...]
```

This summary of all indices. You can easily modify the script to meet your practical requirement, if you do not familiar with this point, please check Section 5.3.

4.18.1.2 Example 2: Ru(bpy₃)²⁺ cation in water

Below we examine several excited states of the Ru(bpy₃)²⁺ cation complex in water by hole-electron analysis based on TDDFT output.



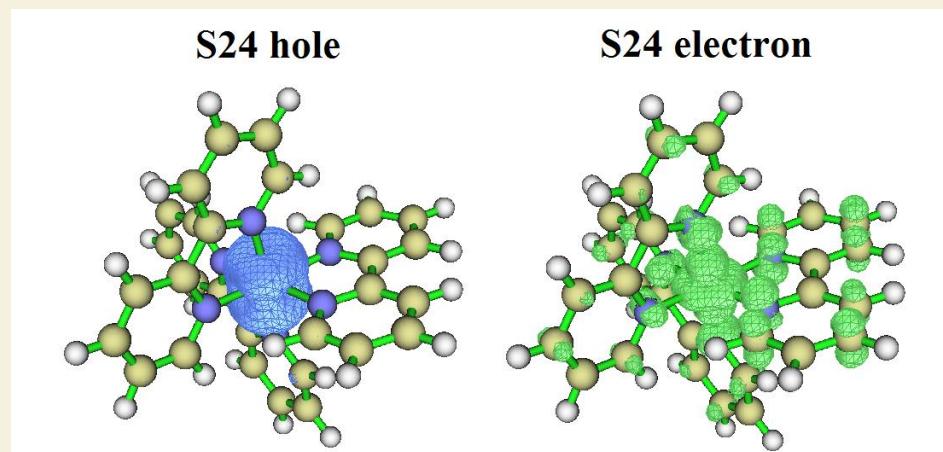
The corresponding Gaussian input file has been provided as *examples\excit\Ru(bpy3)2+.gjf*, please calculate it by Gaussian. If you want to directly obtain the resulting output file and .fchk file, you can download them at http://sobereva.com/multiwfn/extrafiles/Ru_bpy3_2+_TDDFT.zip. As you can see from the .gjf file, the keywords used are *B3LYP/genecp TD(nstates=50) scrf IOp(9/40=3)*, where *scrf* requests Gaussian to employ IEFPCM solvation model to represent water environment. Since this system is not small, and as many as 50 excited states are evaluated, in order to avoid too high calculation time in hole-electron analysis and too large Gaussian output file, *IOp(9/40=3)* is used instead of the *IOp(9/40=4)* employed in last section, the analysis accuracy at this time is still completely sufficient.

We arbitrarily select three excited states to perform hole-electron analysis, the results are

	<i>D</i> (Å)	<i>S_r</i>	<i>H</i> (Å)	<i>t</i> (Å)	hole (Ru%)	ele (Ru%)	MLCT(%)
S0→S24	0.30	0.71	2.73	-1.35	77.3	19.6	57.7
S0→S37	0.11	0.84	3.52	-2.10	16.9	8.8	8.1
S0→S40	0.13	0.71	2.00	-1.05	80.3	42.4	38.0

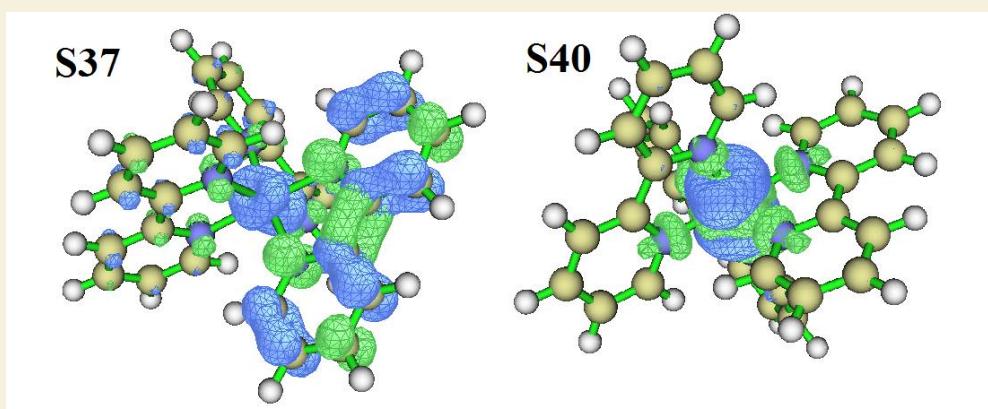
All *D* indices in this table is very small, while all *S_r* indices are fairly large. The main reason is that the current molecule is a symmetric system, thus the CT transitions are multiple directional. The MLCT(%) in the table denotes percent of metal-to-ligand charge transfer character, which can be easily evaluated in terms of subtracting the percentage of metal in hole (namely hole(Ru%)) by that in electron (namely ele(Ru%)). Notice that, properly speaking, what we obtained is net MLCT percentage, it has been somewhat cancelled with LMCT (ligand-to-metal charge transfer).

Below is hole&electron map of S0→S24 excitation with isovalue of 0.002. Since the hole and electron distributions have a large overlap, for the sake of clarity, the isosurfaces of hole and electron are given separately.



Combining the graph with the quantitative data in the table, it is clear that the $S_0 \rightarrow S_{24}$ excitation not only has metal-centered (MC) character, namely electrons of the metal are excited into the metal's own empty orbital, but also have evident MLCT character. As shown in the hole&electron isosurface map, the main body of both the hole and electron is on the metal, and it can be seen that the isosurface of electron also has a large portion on the ligand. The calculated percentage of MLCT feature is $77.3 - 19.6 = 57.7\%$, this value should be said to be very consistent with the information conveyed by the hole-electron isosurface map. Note: The hole also has non-negligible distribution on the ligand, which is $100\% - 77.3\% = 22.7\%$. The reason why the hole isosurface is invisible on the ligands is because its distribution is very diffuse in this region, the hole on the ligands can only be clearly seen by decreasing the isovalue to a smaller value, such as 0.0005.

Then look at $S_0 \rightarrow S_{37}$ excitation. From the hole&electron isosurface map shown below we can see that main body of both hole and electron is located at one of the ligands, therefore it is no doubt that this is a LC (ligand-centered) excitation, corresponding to the case that electrons excited from ligand to its own π^* orbitals. Because there is also hole and electron distribution on the metal, therefore this excitation also shows some MLCT character, which is calculated to be $16.9\% - 8.8\% = 8.1\%$.



Finally, look at the $S_0 \rightarrow S_{40}$ excitation, whose hole and electron isosurfaces are shown at right part of above map. From the hole isosurface and the percent of hole on metal shown in the previous table (16.9%), it is found that its hole distribution character is very similar with $S_0 \rightarrow S_{24}$, but the amount of electron on the ligand is obviously not as large as $S_0 \rightarrow S_{24}$, only tiny part of it is distributed on the four nitrogens directly coordinating with Ru, so the MLCT feature of $S_0 \rightarrow S_{40}$ is conspicuously weaker than $S_0 \rightarrow S_{24}$. In contrast, its MC characteristic is definitely higher than $S_0 \rightarrow S_{24}$.

By the way, for coordinate systems, if you want to obtain MLCT, LMCT, LLCT, MC, LC separately, you should resort to the IFCT analysis illustrated in Section 4.18.8.

We already know that the electron in the excitations from S0 to S24 and S40 mainly sources from Ru atoms, but how to unveil which are the atomic orbitals that electron excited from? Although it can be more or less judged from the isosurface map of the hole, there is still some subjectivity. To figure this out, we can calculate contributions of basis functions to hole and electron. For example, after entering the hole-electron analysis function and selecting the 40th excited state, we select "4 Show basis function contribution to hole and electron" and then input an outputting threshold, such as 2, then information of the basis functions contributing to hole or electron higher than 2% will be printed:

Basis	Type	Atom	Shell	Hole	Electron	Overlap	Diff.
22	D 0	1(Ru)	12	23.81 %	0.00 %	0.33 %	-23.81 %
23	D+1	1(Ru)	12	8.90 %	13.63 %	11.01 %	4.72 %
24	D-1	1(Ru)	12	9.06 %	25.12 %	15.09 %	16.05 %
25	D+2	1(Ru)	12	13.96 %	6.94 %	9.84 %	-7.02 %
26	D-2	1(Ru)	12	14.06 %	15.57 %	14.80 %	1.51 %
27	D 0	1(Ru)	13	3.66 %	0.00 %	0.06 %	-3.66 %
30	D+2	1(Ru)	13	2.32 %	-0.01 %	0.00 %	-2.32 %
31	D-2	1(Ru)	13	2.33 %	-0.03 %	0.00 %	-2.35 %
Sum of above printed terms:				78.10 %	61.22 %		16.88 %

It can be seen from the data that the main contribution to the hole is the D basis function of the Ru atom. The SDD pseudopotential basis set we currently used only describes 4s, 4p, 4d and 5s electrons for Ru, so the excited electron of S0→S40 excitation must come from the 4d atomic orbitals. The basis functions of Ru that have contributions to electron are also D type. Therefore now we know that the MC component in the S0→S40 corresponds to *d-d* transition on Ru.

Notice that this is not the only way to determine which orbitals the excited electrons come from and move to, for example you can also use Multiwfn to perform NTO analysis and examine the pattern of the NTO pair with largest eigenvalue. However, if the electron excitation under study cannot be well described by any pair of NTOs, then this approach will not work. In contrast, the hole-electron analysis does not have any limitation.

4.18.2 Illustration of transition density and transition dipole moment density analysis

The transition density and transition dipole moment density are very important quantities involved in electron transition analysis, they can be studied in the form of real space functions by hole-electron module, or studied in Hilbert space by plotting as colored matrix maps. The two examples in Sections 4.18.2.1 and 4.18.2.2 illustrate these two forms of analyses, respectively. In addition, although the commonly studied transitions are those from ground state to excited states, the transition density and transition dipole moment density between two excited states is also useful in some special studies, Section 4.18.2.3 mentioned how to realize this analysis.

4.18.2.1 Analyzing transition density in real space for N-phenylpyrrole

Theory of real space function form of transition density, namely $T(\mathbf{r})$, has been introduced as "Theory 4" in Section 3.21.1.1, isosurface map of $T(\mathbf{r})$ is capable of revealing apparent coherence

region between hole and electron. While the real space function form of transition electric dipole moment density, namely $T_x(\mathbf{r})$, $T_y(\mathbf{r})$ and $T_z(\mathbf{r})$, are able to exhibit contribution of various regions to transition electric dipole moment (D_x , D_y , D_z), this point has been introduced as "Theory 5" of Section 3.21.1.1. In this section, N-phenylpyrrole will be taken as instance to illustrate this kind of analysis, involved files are completely identical to those utilized in the example in Section 4.18.1.

Boot up Multiwfn and input

```
examples\excit\N-phenylpyrrole.fch // The .fch file yielded by Gaussian TDDFT task
18 // Electron excitation analysis
1 // hole-electron analysis module
examples\excit\N-phenylpyrrole.out // The output file of Gaussian TDDFT task with
IOP(9/40=4) keyword
1 // Analyze electron transition from ground state to the 1st excited state (S0→S1)
1 // Visualize and analyze hole, electron, transition density and so on
2 // Medium quality grid
```

Now you can find below information in the output, they are D_x , D_y and D_z evaluated based on integrating the grid data

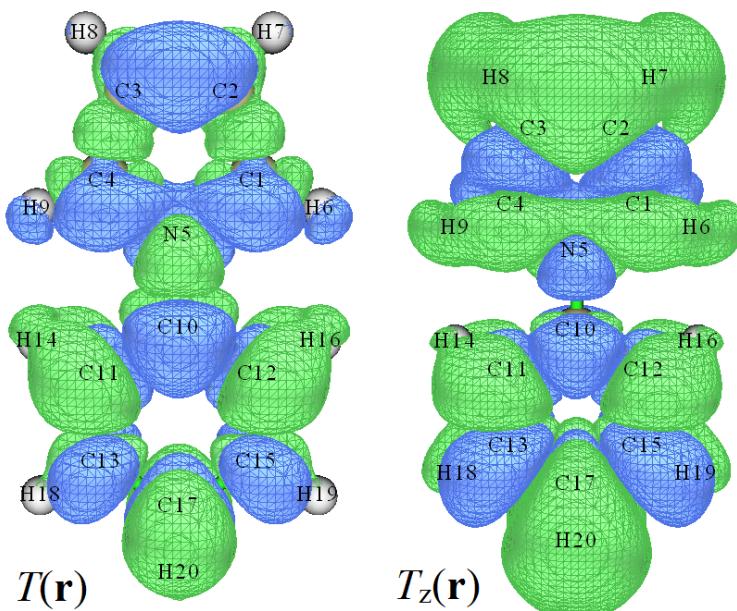
```
Transition dipole moment in X/Y/Z: -0.000021 -0.000045 1.767332 a.u.
```

It is worth to note that these values are very close to those printed in Gaussian output file, as shown below (line 773 of *N-phenylpyrrole.out*), indicating that the grid quality we currently employed is good enough

state	x	y	z	dip. s.	osc.
1	0.0000	0.0000	1.7813	3.1729	0.3935

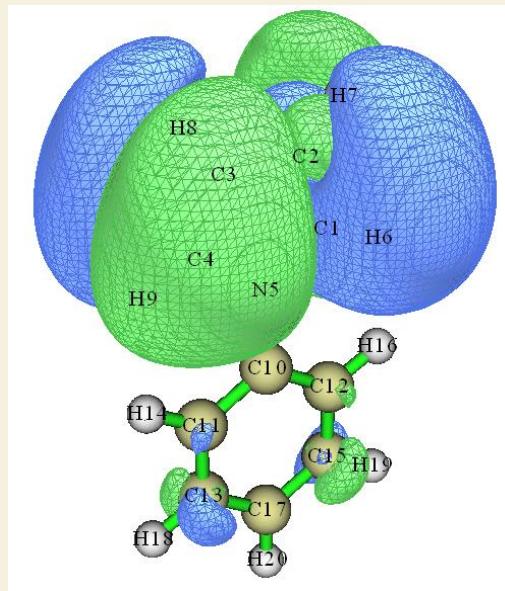
...

We choose option 5 in the post-process menu to show isosurface of transition density, you will then see left map in the image shown below, which exhibits the transition density in real space representation. If the transition density is multiplied by negative of Z coordinate variable, then we will obtain Z component of transition dipole moment density, namely $T_z(\mathbf{r})$. To visualize it, we close current GUI window and choose "6 Show isosurface of transition dipole moment density" and then select "3: Z component", you will see right graph below



From the above $T(\mathbf{r})$ map, we can see that hole and electron have strong coherence everywhere, implying that distribution of both hole and electron covers the whole molecule (this can be easily further confirmed by visualizing hole and electron distributions, as illustrated in Section 4.18.1). From the right map shown above one can see that its positive (green) part is obviously larger than negative (blue) part, recall that the integral of $T_z(\mathbf{r})$ over the whole space is just the Z component of transition electric dipole moment (D_z), this observation explains why D_z of S0→S1 is a large positive value (1.7813 a.u.). If you have interesting, you can try to plot $T_x(\mathbf{r})$ or $T_y(\mathbf{r})$ maps to interpret why the electron excitation under study has vanished D_x and D_y .

Next, we check isosurface of $T_z(\mathbf{r})$ for S0→S4 excitation. We first return to menu of main function 18, then repeat above steps, finally you will see



The green and blue isosurfaces occupy the same amount of space, indicating that positive and negative contributions to D_z are exactly the same, this is why D_z of S0→S4 is zero. Note that the $T_z(\mathbf{r})$ almost solely distributes on the pyrrole region, this is because this electron excitation corresponds to local excitation on pyrrole moiety (as shown in Section 4.18.1).

You may have felt that visual study of transition dipole moment density is interesting and useful; indeed, via this way you can clearly identify contribution to transition dipole moment from different molecular regions. The cube file of the $T(\mathbf{r})$ as well as $T_x(\mathbf{r})$, $T_y(\mathbf{r})$ and $T_z(\mathbf{r})$ can be exported via post-process menu, so that you can also plot them using other visualization software such as VMD, or use such as basin analysis module (Section 3.20) or domain analysis module (Section 3.200.14) to further quantify their distributions, or plot them as plane map via main function 4 (using interpolated function based on loaded grid data, *i.e.* user-defined function -1).

Visualizing transition magnetic dipole moment density

The transition dipole moment we discussed above is transition **electric** dipole moment. There are also other kinds of transition dipole moments, such as transition **velocity** dipole moment and transition **magnetic** dipole moment. Multiwfn is also capable of calculating transition magnetic dipole moment and plotting the corresponding density isosurface map, related knowledges can be found in "Theory 5" of Section 3.21.1. Since this quantity is less important than transition electric

dipole moment, I will not discuss it deeply, but only give a simple example. Using the .out and .fch files of N-phenylpyrrole, we first enter hole-electron module and select the second excited state, then input

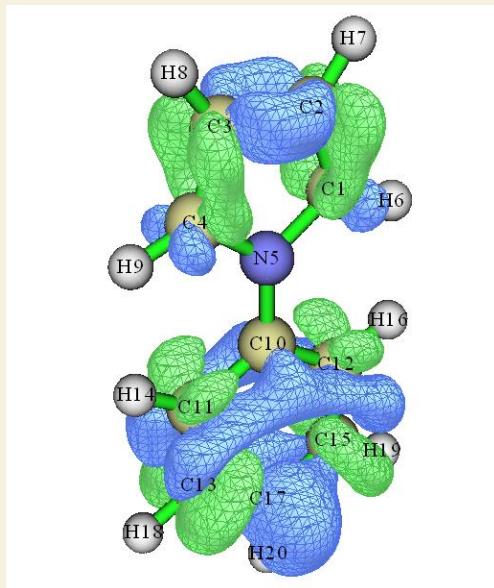
-1 // By default, transition magnetic dipole moment density is not calculated by option 1 of hole-electron module for saving time, we select this option now to make option 1 also calculate this quantity

```
1 // Visualize and analyze hole, electron, transition density and so on
2 // Medium quality grid
```

Once the calculation is finished, from screen you can find the transition magnetic dipole moment evaluated based on the grid data:

```
Transition magnetic dipole moment in x/y/z: -0.503315 0.000119 -0.000177 a.u.
```

Then we select option 9 and select the component of transition magnetic dipole moment density that you are interested in, after that you will see the corresponding isosurface map. Below map is X component of transition magnetic dipole moment density plotted under iso value of 0.005. The relatively larger blue region compared to green region explains why the X component of transition magnetic dipole moment is a negative value (-0.503 a.u.).



4.18.2.2 Analyzing transition density in terms of transition density matrix (TDM) plot

This section will be written soon.

Currently, see "Using Multiwfn to plot transition density matrix and charge transfer matrix to investigate electron excitation characteristics" (in Chinese, <http://sobereva.com/436>).

4.18.2.3 Generate grid data of transition density between two excited states

As shown in Section 4.18.2.1, Multiwfn is able to easily generate grid data of transition density between ground state and a selected excited state. If you make use of Multiwfn flexibly, in fact transition density between two excited states can also be generated. The basic idea is that, we should

first generate transition density matrix (TDM) between the two states, however, because all analyses related to real space function in Multiwfn are based on orbitals, we then need to transform the TDM to corresponding natural orbitals. Finally, the electron density evaluated based on these natural orbitals will directly correspond to transition density.

Here S2→S3 transition of N-phenylpyrrole is taken as an example, below procedure will generate cube file of corresponding transition density. Boot up Multiwfn and input

```
examples\excit\N-phenylpyrrole.fch
18 // Electron excitation analysis
9 // Generate and export transition density matrix
2 // Generate transition density matrix between (TDM) two excited states
examples\excit\N-phenylpyrrole.out
2,3 // Assume that you want to analyze S2-S3 transition
[Press ENTER directly to use default threshold]
y // Symmetrize the resulting TDM in usual way
y // Export wavefunction information including the newly generated TDM to TDM.fch in current folder

Reboot Multiwfn and input
TDM.fch
200 // Other function, part 2
16 // Generate natural orbitals based on the density matrix in .fch/.fchk file
SCF // We input this because the "Total SCF Density" field in the TDM.fch currently correspond to S2→S3 TDM
y // Export new.molden, which contains natural orbitals corresponding to S2→S3 TDM, and then let directly load it
```

```
0 // Return to main menu
5 // Calculate grid data
1 // Electron density
2 // Medium quality grid
2 // Export cube file
```

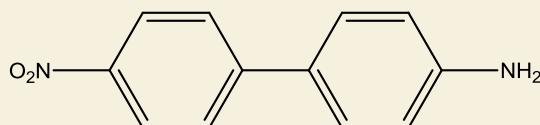
Now the generated *density.cub* in current records transition density between S2 and S3.

It is also possible to generate grid data of transition dipole moment density between excited states. For example, if you set “iuserfunc” parameter in settings.ini to 22, which sets user-defined function to $-x\rho(\mathbf{r})$, then when you use the *new.molden* generated previously as input file, the user-defined function will correspond to X component of transition dipole moment density of S2→S3 transition. Clearly, what you should do next is calculating grid data of user-defined function.

4.18.3 Analyze charge-transfer during electron excitation based on electron density difference

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 to as P2. The related theory has been introduced in Section 3.21.3. The discussions in this example are somewhat related to the ones involved in Section 4.18.1, however the methods employed in this section are

purely based on electron density difference.



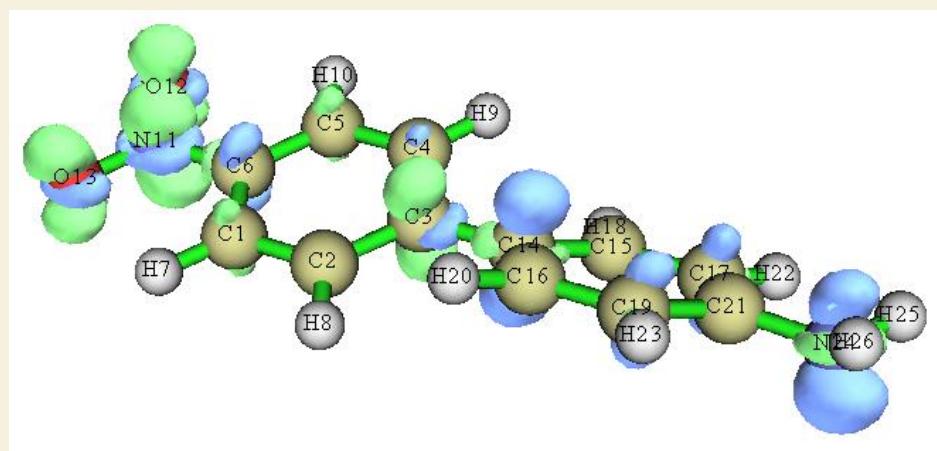
Since the .wfn files corresponding to the excited state and ground state are large, they are not provided. Instead, the input files of Gaussian for generating the two .wfn files are provided in "example\excit" folder (*extP2.gjf* and *basP2.gjf*). I assume that the corresponding .wfn files are produced at "CT" subfolder in current 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 will be meaningless! If you do not have Gaussian in hand, you can also directly download the *extP2.wfn* and *basP2.wfn* from http://sobereva.com/multiwfn/extrafiles/extP2_basP2.zip.

First, we calculate grid data of electron density variation $\Delta\rho$ during the excitation. Boot up Multiwfn and input:

```
CT\extP2.wfn // Excited state wavefunction file
5 // Generate grid data
0 // Set custom operation
1 // Only one file will be dealt with
-CT\basP2.wfn // Ground state wavefunction file. Corresponding density will be subtracted
from the excited one to generate  $\Delta\rho$ 
```

1 // Electron density
2 // Medium quality grid. If the system is much larger than present one, more grid points is required (e.g. using high quality grid)

Once the calculation is normally completed, you can choose option -1 to view the electron density variation during the electron excitation (default isovalue is too large for visualizing density difference, 0.005 is recommended for present case). Green and blue regions correspond to positive and negative regions, respectively, they represent increase and decrease in electron density due to the excitation.



However this density difference graph is not quite intuitive, since positive and negative parts intertwined together and there are many nodes. We will see that *C+* and *C-* functions make the image much clearer.

```
0 // Return to main menu
18 // Electron excitation analysis
```

3 // Analyzing CT based on electron density difference grid data

The following information are displayed immediately. Note that if positive and negative parts of q_{CT} are obvious unequal, that means the grid setting used in generating $\Delta\rho$ is too coarse, and you need to calculate again with finer grid setting.

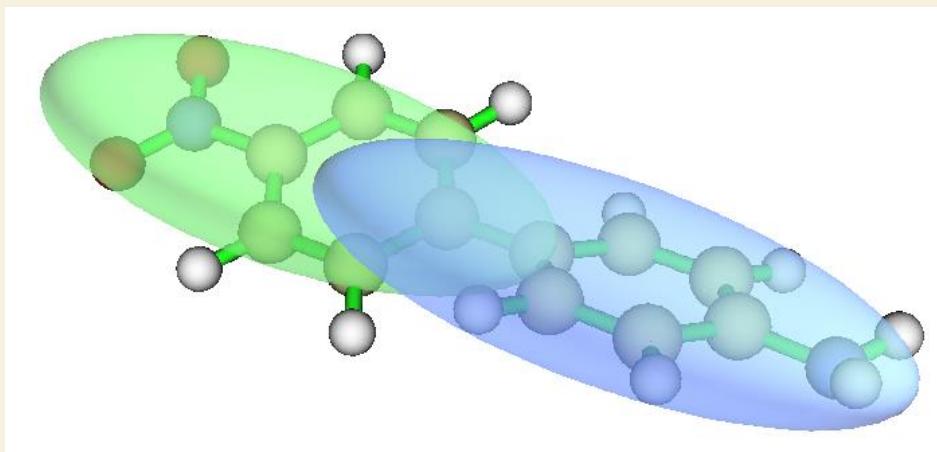
```

q_CT (positive and negative parts):  0.844 -0.844 a.u.
Barycenter of positive part in x,y,z (Angstrom): -2.659 -0.001 -0.000
Barycenter of negative part in x,y,z (Angstrom):  2.294 -0.009 -0.029
Distance of CT in x,y,z (Angstrom):  4.953  0.009  0.029 D index:  4.953
Dipole moment variation (a.u.) :  7.896 -0.014 -0.046 Norm:  7.896
Dipole moment variation (Debye): 20.070 -0.035 -0.117 Norm: 20.070
RMSD of positive part in x,y,z (Angstrom): 2.993 1.250 0.821 Total: 3.346
RMSD of negative part in x,y,z (Angstrom): 3.290 1.144 0.881 Total: 3.593
Difference between RMSD of positive and negative parts (Angstrom):
X: -0.297 Y:  0.106 Z: -0.060 delta_sigma index: -0.247
H_x: 3.141 H_y: 1.197 H_z: 0.851 H_CT: 3.141 H index: 3.469 Angstrom
t index: 1.811 Angstrom
overlap integral between C+ and C- (i.e. S+- index): 0.742365

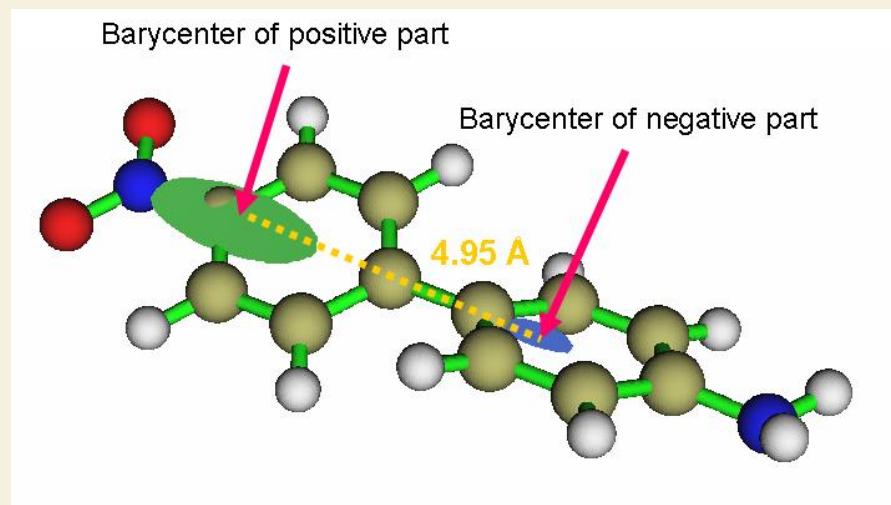
```

Above information are self-explained, if you are confused, please consult Section 3.21.3. The evident positive value of t index implies that the distribution of positive and negative of $\Delta\rho$ has been significantly separated due to strong CT. The large D index (4.95 Å) shows that the CT distance is quite long. Clearly, S0→S1 transition of this system should be identified as typical CT excitation. The excitation caused significant variation of dipole moment, as shown in the data, it is as high as 20.07 Debye. The distribution spatial distribution breadth of positive and negative parts of $\Delta\rho$ are similar, therefore the outputted $\Delta\sigma$ index, which measures difference of their RMSD, is merely -0.247 Å.

By selecting option 1, isosurface of 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 (barycenters of C_+ and C_- exactly correspond to center of their isosurfaces).



From the graph it is evident that the direction of electron transfer is from the amino group side (electron donor) to nitro group side (electron acceptor). However, the barycenters are not exactly located at the two substituents, this observation suggests that the actual electron donor in this electron excitation is not amino group but phenyl. This finding parallels to the fact that phenyl is a weak electron donor.

Hint: If you would like to get better display effect of the C₊ and C₋ isosurfaces, you can use VMD program (freely available at <http://www.ks.uiuc.edu/Research/vmd/>) to display them, the procedure is: Boot up VMD first, drag Cpos.cub into VMD main window, and then drag Cneg.cub into it. Select "Graphics"- "Representations", choose the first term in "Selected Molecule", click "Create Rep" button to create a new representation (the existing representation is used to show molecular structure), change the "drawing method" to "isosurface", set "Draw" to "solid surface", change the isovalue to 0.0015, set "coloring method" to "ColorID" and choose "7 green". Now the isosurface of Cpos has been properly displayed. Next, choose the second term in "Selected Molecule", use the similar methods to set each options, but select "0 blue" in "ColorID", and use isovalue of -0.0015. Finally, the graph will look like the one shown above. You can also set "Material" to "transparent" so that the overlap region of C₊ and C₋ can be clearly distinguished.

4.18.4 Calculate Δr and Λ indices to characterize various electron excitations for N-phenylpyrrole

In this section I will illustrate how to calculate the Δr index proposed in *J. Chem. Theory Comput.*, **9**, 3118 (2013) and the Λ index proposed in *J. Chem. Phys.*, **128**, 044118 (2008) to characterize electron excitations for N-phenylpyrrole. If you are not familiar with these two indices, please check Section 3.21.4 and Section 3.21.14, respectively.

In my personal view of point, using the quantities such as D and S_r indices defined in hole-electron framework to characterize electron excitation is already absolutely sufficient, as illustrated in Section 4.18.1. Theoretically, the Δr and Λ indices may be regarded as approximations of D and S_r , respectively. The only advantage of Δr and Λ is that in Multiwfn, they can be outputted for all selected excited states at the same time and can be decomposed into orbital pair contributions. In addition, calculation cost of Δr index is almost negligible.

The files used in this section is the *N-phenylpyrrole.fch* and *N-phenylpyrrole.out* in "examples\excit" folder, they were yielded by Gaussian, the keywords are *CAM-B3LYP/6-31+G(d) TD(nstates=5) IOp(9/40=4)*. Since the optimized ground state geometry was used in the calculation, therefore the analysis results can be regarded as corresponding to vertical absorption process.

Calculating Δr index

The Δr index is a quantitative indicator for measuring charge transfer (CT) length of electron excitation, larger Δr index implies longer CT distance.

Boot up Multiwfn and input

examples\excit\N-phenylpyrrole.fch

18 // Electron excitation analysis

4 // Calculate Δr index

examples\excit\N-phenylpyrrole.out

1-5 // Assume that we want to calculate Δr index for all the five calculated singlet excited states

Immediately, the results are printed on screen:

```
Excited state 1: Delta_r = 1.499249 Bohr, 0.793368 Angstrom
Excited state 2: Delta_r = 3.489064 Bohr, 1.846333 Angstrom
Excited state 3: Delta_r = 4.641132 Bohr, 2.455982 Angstrom
Excited state 4: Delta_r = 5.869424 Bohr, 3.105966 Angstrom
Excited state 5: Delta_r = 7.091127 Bohr, 3.752463 Angstrom
```

The Δr indices imply that the excitations from ground state (S_0) to the 3th, 4th and 5th excited states possess strong CT character since they have large Δr , while the excitations of $S_0 \rightarrow S_1$ and $S_0 \rightarrow S_2$ should be basically regarded as LE excitations because their Δr indices are not quite large (the original paper of Δr suggests using 2.0 Å as criterion for distinguishing LE and CT excitations). Bear in mind, definitive conclusion about the excitation character can only be finally drawn after visualizing the hole and electron distributions using the hole-electron analysis module of Multiwfn.

In Multiwfn it is possible to decompose the Δr index as contributions of orbital pair transitions. For example, we want to do this for the $S_0 \rightarrow S_4$ excitation, we should first enter the Δr index calculation interface and then input

4 // Only calculate Δr index for a single excitation ($S_0 \rightarrow S_4$), in this case the result can be decomposed

y // Print orbital pair contributions

0.01 // Only the orbital pairs having contribution larger than 0.01 Å will be printed

You will immediately see below information

Note: The configuration coefficients shown below have combined both excitation and de-excitation parts

Sum of square of configuration coefficients: 0.497953

#Pair	Orbitals	Coefficient	Contribution (Bohr and Angstrom)
378	37 41	0.5004500	3.7301590 1.9739153
379	37 43	0.4452400	1.8477898 0.9778083
381	37 47	-0.1067900	0.0929645 0.0491947
382	37 49	-0.0782100	0.0700060 0.0370456
383	37 53	-0.0639300	0.0285116 0.0150877
389	37 72	0.0436900	0.0215865 0.0114231

As you can see, MO37→MO41 transition has predominating contribution (1.97 Å) to the Δr index of $S_0 \rightarrow S_4$ (3.11 Å), while the MO37→MO43 transition also has nonnegligible contribution (0.97 Å).

Calculating Λ (lambda) index

The Λ index essentially measures overlapping degree of hole and electron of electron

excitations. Here we calculate it for all the five excitations for N-phenylpyrrole.

Boot up Multiwfn and input

```
examples\excit\N-phenylpyrrole.fch
18 // Electron excitation analysis
14 // Calculate Λ index
examples\excit\N-phenylpyrrole.out
1-5 // Analyze all the five calculated singlet excited states
```

Immediately, the results are printed on screen:

```
Excited state 1: lambda = 0.684853
Excited state 2: lambda = 0.563804
Excited state 3: lambda = 0.530928
Excited state 4: lambda = 0.198710
Excited state 5: lambda = 0.235255
```

From above output, it can be found that the Λ indices are nearly inversely proportional to the Δr indices, because the larger the hole-electron overlapping extent, usually the shorter the hole-electron separation distance (but bear in mind, this relationship is not always true).

Then we decompose the Λ index for the fourth excitation. Input below commands

```
y // Do the Λ index analysis again
4 // The fourth excitation
y // Decompose analysis on Λ index
0.01 // Printing threshold
```

Then you will see all MO pairs having contribution to Λ index larger than 0.01:

```
Sum of square of configuration coefficients: 0.497953
#Pair      orbitals      Coefficient      Contribution
  378       37     41      0.5004500      0.0865190
  379       37     43      0.4452400      0.0915297
```

The data indicates that only occupied MO 37 has non-negligible overlap with unoccupied MOs; specifically, only the overlap between MO37-MO41 and between MO37-MO43 is relatively detectable.

4.18.6 Generate and analyze natural transition orbitals (NTOs) for uracil

In this section I illustrate how to use Multiwfn to perform the very popular natural transition orbital (NTO) analysis with uracil as example. Please first read Section 3.21.6 to acquire basic knowledge of NTO. Although in this example the files outputted by Gaussian were used as input file, in fact the files outputted by ORCA are also fully supported, see Section 3.21.1.2 for detailed requirement about the input file.

Before showing how to perform NTO analysis, I would like to let you apprehend why NTO analysis is meaningful. As an instance, we use Gaussian to perform TDDFT calculation at PBE0/6-31G* level for singlet excited states of uracil, you will find below information

```
Excited State 3: Singlet-A"   6.0180 ev  206.02 nm  f=0.0000 <s**2>=0.000
26 -> 30          0.54135
```

26 -> 31	-0.20634
28 -> 30	-0.15424
28 -> 31	0.36715

Clearly, in excitation of S0→S3, there is no dominant MO transition, the largest contribution of a single MO pair is merely $0.541^2 * 2 * 100\% = 58.5\%$, therefore it is impossible to identify the nature of this excitation by viewing only one MO pair. In such difficult cases, NTO analysis is often useful, because after transforming MOs to NTOs, commonly you will be able to find only one pair of NTO having eigenvalue very close to 1, transition between the two NTOs in this pair faithfully represents the real character of the electron excitation.

The files needed by NTO analysis have mentioned at the beginning of Section 3.21. Briefly, assume that you are a Gaussian user and you want to study electron excitation from ground state to the lowest three singlet excited states for uracil at TD-PBE0/6-31G* level, what you need to do is carrying out a normal TDDFT calculation with these keywords: # PBE1PBE/6-31G* TD IOP(9/40=4), also you need to make Gaussian generate corresponding .fch file. The input file, output file and .fch file have already been provided in "examples\excit\NTO" folder. The keyword IOP(9/40=4) is very important, without it the NTO result will be evidently inaccurate, the meaning of this IOp has been mentioned in Section 4.18.1.

Now we start to carry out NTO analysis. Boot up Multiwfn and input
examples\excit\NTO\uracil.fch

18 // Electron excitation analysis

6 // Generate NTOs

examples\excit\NTO\uracil.out // Gaussian calculated three lowest excited states, you can analyze any one of them

3 // Study transition from ground state (S0) to the 3rd excited state (S3)

Now Multiwfn loads transition information of S0→S3 from the Gaussian output file and generate NTOs, the eigenvalues of occupied and virtual NTOs are shown below

The highest 10 eigenvalues of occupied NTOs:

0.000006	0.000007	0.000015	0.000016	0.000024
0.000063	0.000121	0.000582	0.134025	0.865529

The highest 10 eigenvalues of virtual NTOs:

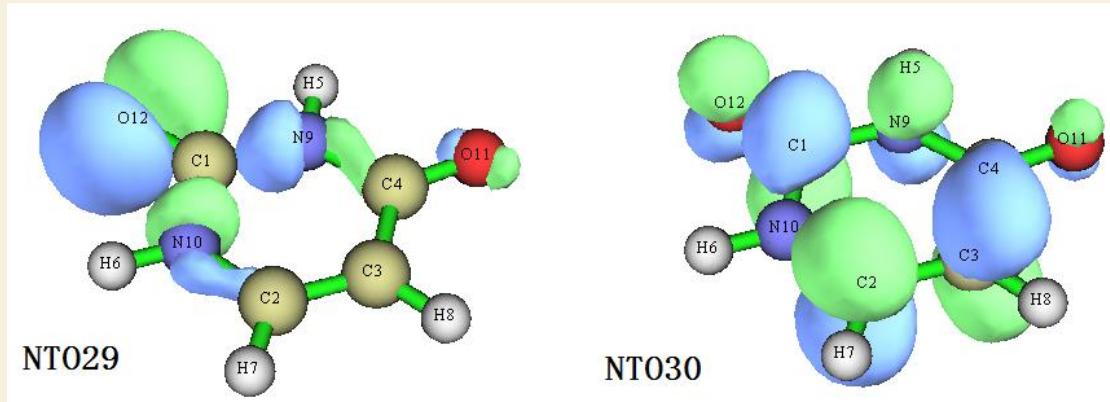
0.865529	0.134025	0.000582	0.000121	0.000063
0.000024	0.000016	0.000015	0.000007	0.000006

It can be seen that, the eigenvalues of occupied and virtual NTOs have one-to-one correspondence, the largest eigenvalue is 0.8655, that means that NTO pair contributes as high as 86.55% of the S0→S3 transition. So, if we would like to characterize the nature of this transition, we can only study the occupied NTO and virtual NTO in this NTO pair.

Now you can select if outputting .fch or .molden file containing the NTOs. We choose "2 Output NTO orbitals to .fch file" and input the path to output, such as *C:\S3.fch*. After the .fch has been successfully generated, you can reboot Multiwfn and load the *S3.fch*, in main function 0 you can visualize the NTOs, the orbital energies now correspond to NTO eigenvalue. To plot the two NTOs in the NTO pair with 86.55% contribution, in the GUI of main function 0 you can select "orbital info." - "Show up to LUMO+10" in the menu, in the text window you will find output like below

orb: 27	Ene(au/ev):	0.000582	0.0158	occ: 2.000000	Type: A+B
orb: 28	Ene(au/ev):	0.134025	3.6470	occ: 2.000000	Type: A+B
orb: 29	Ene(au/ev):	0.865529	23.5522	occ: 2.000000	Type: A+B
orb: 30	Ene(au/ev):	0.865529	23.5522	occ: 0.000000	Type: A+B
orb: 31	Ene(au/ev):	0.134025	3.6470	occ: 0.000000	Type: A+B
orb: 32	Ene(au/ev):	0.000582	0.0158	occ: 0.000000	Type: A+B

We can see that the occupied NTO with index of 29 and the virtual NTO with index of 30 constitute the NTO pair with eigenvalue of 0.8655, there we select corresponding index in the GUI to visualize them, the isosurfaces are shown below



Undoubtedly, this $S_0 \rightarrow S_3$ excitation can be regarded as transition from lone pair of O12 to anti-bonding π orbital of the uracil ring, at least we have 86.55% confidence to say that. From NTO eigenvalues we notice that NTO28→NTO31 transition also has small contribution (13.40%) to the excitation, please plot corresponding orbitals and discuss their characteristic.

The NTOs can also be subjected to quantitative analyses. For example, you can enter main function 8 and use suitable options to analyze their orbital composition at quantitative level, or you can use subfunction 11 of main function 100 to evaluate overlap extent and centroid distance between selected two NTOs.

Note that NTO analysis has both advantage and disadvantage with respect to the hole-electron analysis, this point has been mentioned at the end of Section 3.21.6. For many systems and excitations, even if NTO transformation has been applied, there are still no NTO pair with predominant contribution, in this case you have to use hole-electron analysis to facilitate discussion of excitation characteristic. The drawback of hole-electron analysis is that it is unable to present orbital phase information like NTO analysis, and it takes more computational cost.

Sometimes we want to perform NTO analysis for a batch of excitations, for example, we want to generate a batch of .fch files that respectively contain NTOs of $S_0 \rightarrow S_1$, $S_0 \rightarrow S_2$ and $S_0 \rightarrow S_3$ transitions, although you can manually do this in the interactive interface of Multiwfn, a more efficient and clever way is employing shell script. If you are a Linux user, we can run below shell script to yield $S1.fch$, $S2.fch$ and $S3.fch$, which contain NTOs corresponding to $S_0 \rightarrow S_1$, $S_0 \rightarrow S_3$ and $S_0 \rightarrow S_3$. This script is very easy to understand as long as if you have basic level of knowledge about shell programming.

```
#!/bin/bash
cat << EOF > allNTO.txt
```

```

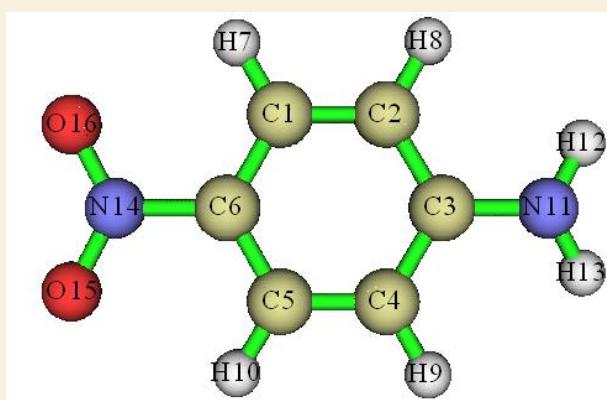
6
examples/NTO/uracil.out
EOF
for ((i=1;i<=3;i=i+1))
do
cat << EOF >> allNTO.txt
$i
2
$Si.fch
6
EOF
done
./Multiwfn examples/excit/NTO/uracil.fch < allNTO.txt
rm ./allNTO.txt

```

This script is also provided as *examples\excit\NTO\allNTO.sh*. If you do not make any modification to the script, this script should be copied to the Multiwfn folder and run as *./allNTO.sh* in Multiwfn folder, then *S1.fch*, *S2.fch* and *S3.fch* will be yielded at the same folder. In practical studies, you should properly modify the script according to actual situation, the range of the excitations to be studied is determined by "i=1;i<=3".

4.18.8 Using IFCT method to study interfragment charge transfer during electron excitation for 4-nitroaniline

Multiwfn supports IFCT method to study interfragment charge transfer (CT) during electron excitation, please read Section 3.21.8 first before following this example. In this example, I will illustrate how to use this method to study CT between three fragments (-NO₂, -NH₂ and the linker benzene) of 4-nitroaniline during its electron excitation.



The Gaussian input file of TDDFT task at PBE0/6-311G* level for 4-nitroaniline has been provided as *examples\excit\4-nitroaniline.gjf*. Note that *IOp(9/40=4)* is used so that Gaussian can print enough configuration coefficients. Run it by Gaussian to obtain .chk file and .out file, then convert the .chk file to .fch via formchk. The .fch and .out files have been provided in "examples\excit" folder.

Boot up Multiwfn and input below commands

examples\excit\4-nitroaniline.fch

18 // Electron excitation analysis

8 // Calculate interfragment charge transfer in electron excitation via IFCT method

examples\excit\4-nitroaniline.out

2 // We first analyze transition from ground state (S0) to the second excited state (S2)

3 // Define three fragments

11-13 // Atomic indices of amino group (fragment 1)

1-10 // Atomic indices of benzene group (fragment 2)

14-16 // Atomic indices of nitro group (fragment 3)

Then you will see

Contribution of each fragment to hole and electron:

1 Hole: 28.17 % Electron: 3.02 %

2 Hole: 65.89 % Electron: 29.52 %

3 Hole: 5.94 % Electron: 67.46 %

Construction of interfragment charge-transfer matrix has finished!

Variation of population number of fragment 1: -0.25150

Variation of population number of fragment 2: -0.36369

Variation of population number of fragment 3: 0.61519

Intrafragment electron redistribution of fragment 1: 0.00849

Intrafragment electron redistribution of fragment 2: 0.19453

Intrafragment electron redistribution of fragment 3: 0.04009

Transferred electrons between fragments:

1 → 2: 0.08315 1 ← 2: 0.01987 Net 1 → 2: 0.06328

1 → 3: 0.19001 1 ← 3: 0.00179 Net 1 → 3: 0.18821

2 → 3: 0.44452 2 ← 3: 0.01754 Net 2 → 3: 0.42697

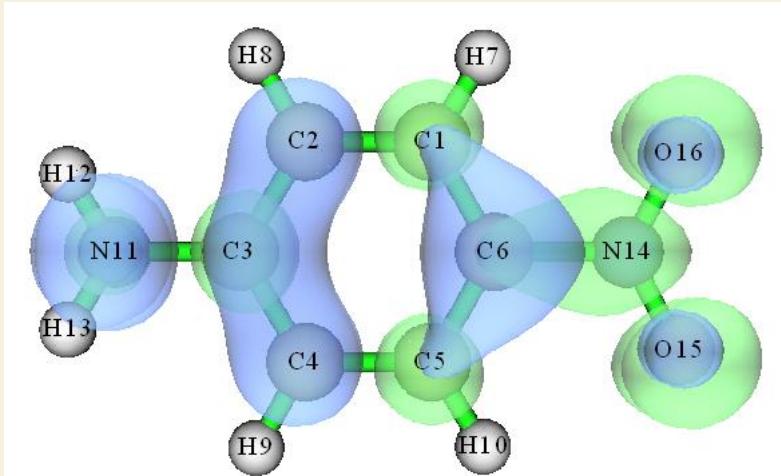
The output is very easy to understand, if you are confused, please check corresponding explanation in Section 3.21.8. For example, the data shows that during the S₀→S₂ excitation, nitro group (fragment 3) donates 0.00179 electrons to amino group (fragment 1) and meantime accept 0.19001 electrons from amino group, therefore nitro group totally gains 0.18821 electron from amino group. If the benzene fragment is also taken into account, the electron excitation totally increases electron population of nitro group by 0.61519. The electron redistribution phenomenon in the two terminal groups is not prominent, however, as the data shown (0.19453), the electron redistribution within the benzene, which behaves as π linker, is remarkable.

For facilitating discussion, I summarized all IFCT analysis data in below table. The diagonal terms correspond to amount of intrafragment electron redistribution.

Donor	Acceptor		
	amino group	benzene	nitro group
amino group	0.008	0.083	0.190
benzene	0.020	0.195	0.445
nitro group	0.002	0.018	0.040

As you can see from the table, there are three prominent interfragment CT terms (sorted according to magnitude): benzene \rightarrow nitro, amino \rightarrow nitro, amino \rightarrow benzene, all of them direct from amino group towards nitro group.

For better and intuitively understanding above data, we plot hole-electron isosurface map using the method described in Section 4.18.1:



In the graph, main distribution regions of hole and electron correspond to blue and green, respectively. As you can see, hole mainly distributes on the amino group and benzene moiety, while electron mainly locates at the benzene linker and nitro group, clearly the overall CT direction is from amino group to nitro group, this is completely in line with our observation of quantitative interfragment CT analysis.

Notice that by default Multiwfn employs Mulliken-like partition to evaluate atomic contributions to hole and electron distributions, which are needed by the IFCT analysis. Beware that since this method is incompatible with diffuse functions, if basis set containing diffuse functions was used, the calculated amount of CT transfer may be unreliable or completely meaningless. Fortunately, it is also possible to employ Becke partition to evaluate interfragment CT quantities, in this case diffuse functions could be safely used, but additional steps are needed, namely cube file of hole and electron must be first generated using the hole-electron analysis module. Next, we will redo above analysis in this way. Boot up Multiwfn and then input below commands

```
examples\excit\4-nitroaniline.fch
18 // Electron excitation analysis
1 // Evaluate and analyze hole and electron
examples\excit\4-nitroaniline.out
2 // Analyze S0 $\rightarrow$ S2
1 // Visualize and analyze hole, electron, transition density and so on
2 // Medium quality grid
10 // Export cube file for hole
1 // Total hole (i.e. hole distribution in complete form)
11 // Export cube file for electron
1 // Total electron (i.e. electron distribution in complete form)
```

Now *hole.cub* and *electron.cub* have been generated in current folder, so we can start IFCT analysis. Input below commands

```
0 // Return
0 // Return to menu of main function 18
8 // IFCT analysis
```

Now the *hole.cub* and *electron.cub* in current folder have been detected, here we input y to make Multiwfn carry out IFCT analysis based on them using Becke partition method to evaluate their compositions. Then define fragments:

```
3 // Define three fragments
11-13 // Atomic indices of amino group (fragment 1)
1-10 // Atomic indices of benzene group (fragment 2)
14-16 // Atomic indices of nitro group (fragment 3)
```

Now the result is

Transferred electrons between fragments:

1 -> 2:	0.08103	1 <- 2:	0.02383	Net 1 -> 2:	0.05720
1 -> 3:	0.18532	1 <- 3:	0.00285	Net 1 -> 3:	0.18248
2 -> 3:	0.43360	2 <- 3:	0.02264	Net 2 -> 3:	0.41097

The data is in good agreement with the result shown earlier, which was calculated based on Mulliken-like partition.

Commonly I suggest using the default way to realize the IFCT analysis, since no additional procedure is needed and the cost is relative low. However, if due to some special reasons diffuse functions must be employed (*e.g.* anion systems, Rydberg excitation), you have to perform IFCT analysis based on Becke partition.

More discussions and illustrations about the IFCT analysis can be found from my blog article "Using the IFCT method in Multiwfn to evaluate amount of electron transfer between arbitrarily defined two fragments during electron excitation" (in Chinese, <http://sobereva.com/433>).

4.18.10 Gain molecular orbital pair contributions to transition dipole moment

In order to gain a deeper insight into transition (electric) dipole moment, Multiwfn provides a function used to decompose it to contributions from various MO pair transitions, see Section 3.21.10 for introduction. Here I present an example. The .fch and .out files involved in this example were produced by TDDFT calculation of Gaussian.

Boot up Multiwfn and input

examples\excit\N-phenylpyrrole.fch

18 // Electron excitation analysis

10 // Decompose transition dipole moment as molecular orbital pair contributions

examples\excit\N-phenylpyrrole.out

1 // Select the excitation from ground state (S0) to the first singlet excited state (S1)

Now below information about this excitation is shown on screen

Transition dipole moment in x/Y/Z: -0.000000 -0.000000 1.781438 a.u.

Norm of transition dipole moment: 1.781438 a.u.

Oscillator strength: 0.3935306

Then you can find several options on screen, they are self-explained. We first choose option 1

and input for example 0.02, then all MO pairs having contribution larger than 0.02 are printed:

#Pair		Coefficient	Transition dipole x/Y/Z (a.u.)		
1213	35 ->	46	0.040230	0.000000	0.000000
1214	35 ->	50	-0.047670	0.000000	-0.000000
1239	36 ->	40	-0.101270	0.000000	-0.000000
1259	37 ->	40	-0.127550	0.000000	0.000000
1260	37 ->	52	0.069960	-0.000000	-0.000000
1262	37 ->	58	-0.036060	-0.000000	0.000000
1278	38 ->	39	0.672690	-0.000000	0.000000
1280	38 ->	50	0.052570	-0.000000	0.000000
2489	36 <-	40	-0.014330	0.000000	-0.000000
2506	37 <-	52	0.015140	-0.000000	-0.000000
2522	38 <-	39	-0.027240	0.000000	-0.000000
Sum of the 11 pairs shown above:			-0.000000	-0.000000	2.165763

From the output, we can immediately find that transition of MO38→MO39 has dominating contribution (2.796678 a.u.) to this S0→S1 excitation.

When there are too many MO pairs having nonnegligible contributions to transition dipole moment and thus difficult to identify important MO transitions, you can let Multiwfns sort the MO pairs according to their contributions to specific component of transition dipole moment. For example, here we choose the option "4 Print orbital pairs according to absolute contribution to Z component" and then input 5, then you will see the five MO pairs having largest contribution to Z component of transition dipole moment:

#Pair		Coefficient	Transition dipole x/Y/Z (a.u.)		
1278	38 ->	39	0.672690	-0.000000	0.000000
1239	36 ->	40	-0.101270	0.000000	-0.000000
1259	37 ->	40	-0.127550	0.000000	0.000000
1260	37 ->	52	0.069960	-0.000000	-0.000000
2522	38 <-	39	-0.027240	0.000000	-0.000000

By the way, oscillator strength (f) directly relates to norm of transition dipole moment, therefore it can be expected that if the configuration coefficient corresponding to MO38→MO39 is set to zero, namely ignoring its contribution, then f will be lowered evidently. As shown earlier, the original f of S0→S1 is 0.39353. Let us quantitatively check how MO38→MO39 affects the f . To do this, we can manually set configuration coefficient of this transition to zero and then re-examine the f value. We input following commands

```

0 // Return to menu of electron excitation analysis
-I // Check, modify and export configuration coefficients of an excitation
I // Choose the first excited state
I // Set coefficient of a MO pair
38,39 // The MO indices of the MO pair
I // The transition type is chosen as "Excitation", hence MO38→MO39 is selected (if
inputting 2, then what we selected will be MO38←MO39)
0 // Set the configuration coefficient to zero
-3 // Export current excitation information to a plain text file

```

SI.txt // The path of the file to store excitation information of S0→S1

Now *SI.txt* has been generated in current folder, if you open it with text editor, you will find the coefficient corresponding to MO38→MO39 is indeed zero.

Then reboot Multiwfn and input

o // Load the file used at the last time, namely *examples\excit\N-phenylpyrrole.fch*

18 // Electron excitation analysis

10 // Decompose transition dipole moment as molecular orbital pair contributions

SI.txt

Now the printed *f* is only 0.1278, which is less than 1/3 of its original value (0.39353), showing that MO38→MO39 has crucial influence on strength of S0→S1 excitation.

Since the coefficient of MO38→MO39 is as large as 0.6727, after setting it to zero, now the sum of the square of remaining coefficients has been much less than 0.1, which is far from the ideal value of closed-shell case (0.5).

4.18.11 Plot transition dipole moment vector contributed by molecular fragments as arrows

In Section 4.18.2.1, I have shown how to plot transition dipole moment density in real space, which is extremely useful for studying contribution of different regions in three-dimension space. In fact, if using a special plotting script of VMD (<http://www.ks.uiuc.edu/Research/vmd/>) provided below, transition dipole moments contributed by molecular fragments can be easily drawn as arrows, which greatly facilitates discussion of composition of total transition dipole moment.

Here, azobenzene is taken as example. The input file of TDDFT task of Gaussian for azobenzene is provided as *example\excit\Azobenzene.gjf*. Note that *IOp(9/40=4)* is used and .chk file is saved after calculation. Run it by Gaussian, and then convert azobenzene.chk to azobenzene.fch. (If you do not have Gaussian in hand, you can also directly download the .out and .fch files from http://sobereva.com/multiwfn/extrafiles/Azobenzene_exc.zip)

Boot up Multiwfn, load the *azobenzene.fch*, then input

18 // Electron excitation analysis

11 // Decompose transition dipole moment as basis function and atom contributions

Azobenzene.out // The Gaussian output file obtained by running Azobenzene.gjf

2 // Assume that we want to study is electron excitation from ground state to the second excited state

1 // The type of transition dipole moment to be decomposed is electric

n // Do not generate *AAtrdip.txt* currently

Now *trdipcontri.txt* is outputted to current folder, which contains transition dipole moment contributed by each basis function and each atom. Move this file to VMD folder.

Return to main menu, then enter subfunction 2 of main function 100, export current molecular geometry to *azobenzene.pdb*.

Copy *examples\excit\loadip.tcl* to VMD folder, this is a VMD script written by me, it can load data from *trdipcontri.txt*. It also defines custom commands "dip" and "dipatm" used to draw transition dipole moment contributed by specific molecular fragment as arrow.

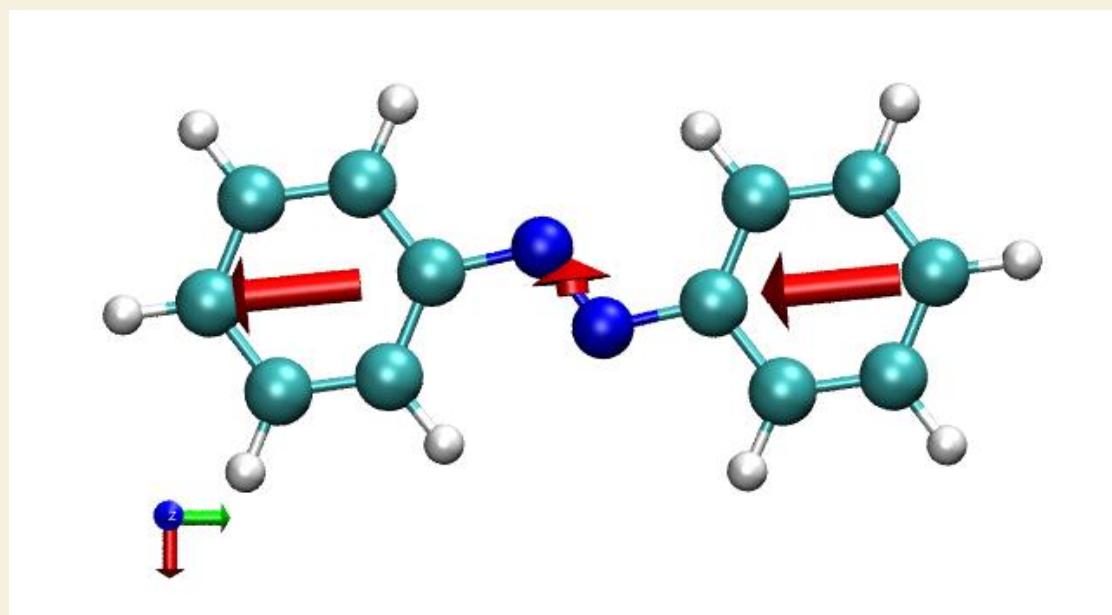
Boot up VMD, drag the file *azobenzene.pdb* into VMD main window to load it, then run *source loadip.tcl* in VMD console window to execute the script. Assume that we want to divide the molecule as three parts to separately investigate their contributions to transition dipole moment,

namely the first phenyl group (atoms 1~11), N₂ part (atoms 12 and 13) and the second phenyl group (atoms 14~24), we should run below commands in VMD console window

```
draw color red
dip "serial 1 to 11"
dip "serial 12 13"
dip "serial 14 to 24"
```

Now you will see three red arrows in the VMD graphical window. The length of cylindrical part of the arrows correspond to magnitude of fragmental transition dipole moments, the center of the arrows correspond to geometric center of the fragments. Note that when we use "dip" command, the fragment geometry center and quantitative contribution to transition dipole moment by the selected fragment are also shown in VMD console window.

In order to improve the graphical quality, we input *color Display Background white* in console window to set white as background color, enter Graphics - Representation and set Drawing method to CPK, and then choose Display - Orthographic in VMD main window. The final graph will look like below.

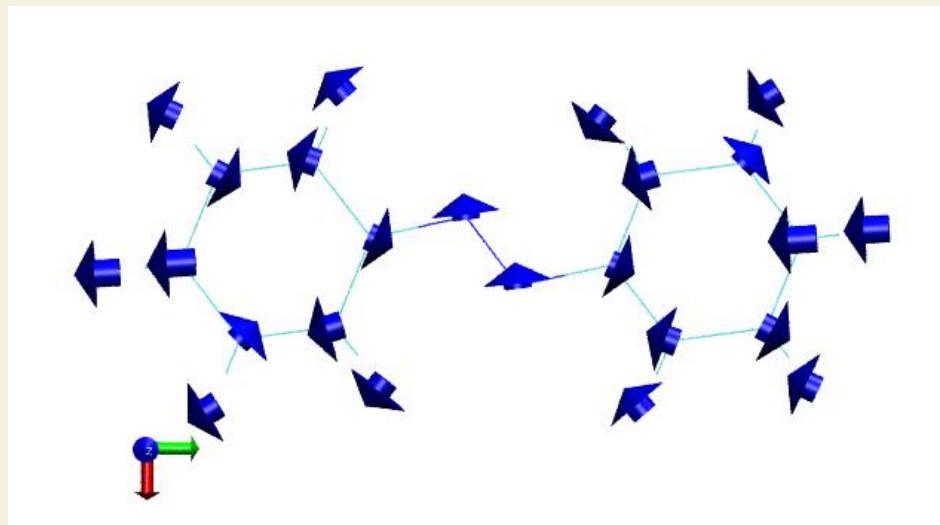


As you can see, both the two phenyl groups have significant contribution to Y component of total transition dipole moment (the red, green and blue of the axis shown at left-bottom part of the graph correspond to X, Y and Z directions, respectively). For quantitative comparison purpose, total transition dipole moment vector and its compositions are also listed below

```
Total:      0.1155 -2.8868 0.0
Phenyl group 1: 0.14262 -1.43288 0.0
N2:          -0.16948 -0.02138 0.0
Phenyl group 2: 0.14262 -1.43288 0.0
```

If you also want to plot total transition dipole moment as green arrow on the graph, you can input *draw color green* and then input *dip all*.

It is also possible to plot transition dipole moment contributed by each atom. To do that, we input *draw delete all* to remove all existing arrows, and then input *dipatm*, you will immediately see



There is a very important point that should be paid attention to when using above method to decompose transition dipole moment, namely contribution of a fragment is often dependent of choice of origin, because transition charge of a fragment is often non-zero. For example, if we use subfunction 6 of hole-electron analysis module to export atomic transition charges and then sum them as fragment transition charges, you will find the value of the first phenyl group is 0.2116. Since it is non-zero, it can be proved that if overall coordinate of the azobenzene is translated, the transition dipole moment corresponding to this fragment must be varied; in other words, the result is not definite. Therefore, one should carefully discuss fragmental transiton dipole moment in papers.

Another very important point is that since the transition dipole moment is decomposed via Mulliken method, the analysis method shown above will be meaningless when diffuse functions are presented in the electron excitation calculation.

4.18.13 Study electronic structure of a single excited state and difference between two excited states

Most other subsections in Section 4.17 focus on exemplifying how to study electron transition characters, however, sometimes we want to study character of a single excited state or difference between two excited states in specific property. In Multiwfn, one can perform various kinds of wavefunction analysis for an excited state as usual, however, the input file must contain wavefunction of this excited state. For multi-configuration methods that can study excited state, such as CIS and TDDFT, the excited state wavefunction must be recorded as natural orbitals (NOs), because Multiwfn always load wavefunction in terms of orbitals.

The main purpose of this section is illustrating the function used to generate .molden file containing NOs of an excited state, so that we can analyze wavefunction character of this state. I strongly suggest you read Section 3.21.13 first, in which the details of generating NOs of excited states are described.

Note that the NOs generated in this way correspond to unrelaxed density of excited state, which is in principle not as accurate as relaxed excited state density, but the cost is fairly low.

For Gaussian users, one can use such as “# PBE1PBE/6-31G* out=wf TD(root=x)” keywords to export NOs of excited state x to specific .wfn file, which can also be employed as input file for performing wavefunction analysis of the excited state. However, do not forget that many functions in Multiwfn require basis function information, which cannot be provided by .wfn file, thus in this case the kind of analyses can used is severely limited. In addition, by solely using Gaussian it is also possible to yield and store the NOs to .fch file, as explicitly described at the

beginning of Chapter 4, however this procedure is relatively cumbersome. Notice that the NOs generated in these ways correspond to relaxed excited state wavefunction, the cost is much higher than regular excited state calculation. If you only need the NOs corresponding to the unrelaxed excited state wavefunction, simply adding “density=rhoc” keyword in route section.

Example of wavefunction analysis of an excited state

Here I take N-phenylpyrrole as example, assume that we want to examine Mayer bond orders for the second singlet excited state. To do so, we first carry out a regular TDDFT calculation with *IOP(9/40=4)*, the *examples\excit\N-phenylpyrrole.out* is output file and *examples\excit\N-phenylpyrrole.fch* is corresponding .fch file. The geometry was previously optimized for ground state.

Boot up Multiwfn and input below commands

```
examples\excit\N-phenylpyrrole.fch
18 // Electron excitation analysis
13 // Generate natural orbitals of specific excited states
examples\excit\N-phenylpyrrole.out
2 // Choose the 2nd excited state
```

Now, *NO_0002.molden* has been generated in current folder, it records wavefunction in terms of NOs for the second excited state.

Reboot Multiwfn and input

```
NO_0002.molden
9 // Bond order analysis
1 // Mayer bond order
```

From the output you will find the bond order of the N5-C10 bond, namely the bond linking pyrrole and benzene moieties, is 0.796. If you repeat the calculation for *examples\excit\N-phenylpyrrole.fch*, the result will correspond to ground state, and you will find the Mayer bond order is 0.713. Clearly, the vertical excitation from S0 to S2 at minimum point of S0 weakens the strength of N5-C10 detectably.

Plotting density difference between excited states

Next I illustrate how to plot density difference between various excited state. In fact this is very easy, you simply need to generate Multiwfn input files containing NOs of the two excited states respectively, and then get their difference via the steps illustrated in Sections 4.5.5 or 4.18.3.

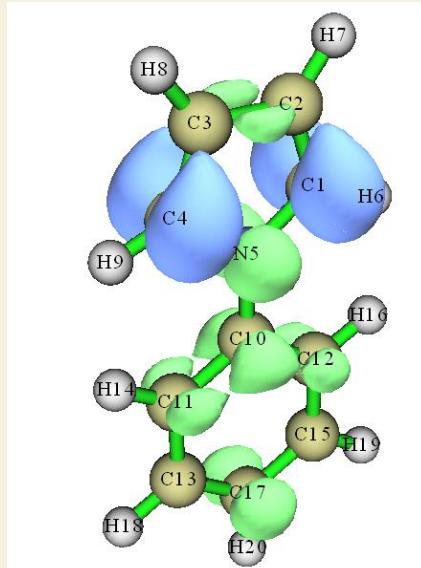
I still take N-phenylpyrrole as example. We repeat aforementioned steps using the *N-phenylpyrrole.fch* and *N-phenylpyrrole.out* to generate .molden files, when Multiwfn asks you to input the index of excited states, we input 1-3, then *NO_0001.molden*, *NO_0002.molden* and *NO_0003.molden* will be generated in current folder, clearly now we can study density difference between 1-2, 1-3 and 2-3.

Assume that currently we want to visualize isosurface map of electron density difference between the third and the first excited state, we reboot Multiwfn and input

```
NO_0003.molden
5 // Calculate grid data
0 // Custom operation
1 // One file will be dealt with the firstly loaded file
-,NO_0001.molden
1 // Electron density
2 // Medium quality grid
```

-I // Visualize isosurface

We set isovalue to 0.005, we will obtain below graph



The density difference map between other excited states can be obtained similarly.

Although you may also directly use your quantum chemistry program to generate wavefunction file containing NOs for various excited states, the procedure is evidently much more cumbersome than using Multiwfn, because as shown above, the advantage of Multiwfn is that it is able to simultaneously generate .molden file containing NOs for a batch of excited states.

Calculate difference in fragment charge between excited states

Finally, as an example, we will study difference of electron distribution at quantitative level by comparing fragment charge of the pyrrole ring between excited states 3 and 1.

Boot up Multiwfn and input

NO_0003.molden

7 // Population analysis

-I // Define fragment

I-9 // The pyrrole fragment, which is composed of atoms 1~9

I1 // ADCH charge

I // Use built-in atomic densities

You will find

Fragment charge: 0.598263

Namely the fragment charge of the pyrrole ring is 0.598 at the 3rd excited state. Repeat the calculation for the *NO_0001.molden*, the charge of the pyrrole ring will be found to be 0.148. The data shows that during the (hypothetical) transition from the 1st to the 3rd excited state, the pyrrole fragment will lose $0.598 - 0.148 = 0.45$ electron, this well explains why in the corresponding density difference map there are obvious isosurfaces around the pyrrole ring and most of them are in blue color.

4.19 Orbital localization analysis

Orbital localization is a very powerful and useful technique, it can transform canonical molecular orbitals, which often show highly delocalized character, to localized form, which is known as localized molecular orbital (LMO). The LMOs have close relationship with many classical chemistry concepts such as chemical bond and lone pair, therefore they can be used to analyze and unveil many problems of chemical interest. Before reading this section please read Section 3.22 first to gain some basic knowledges. Some examples in other sections, such as the example in Sections 4.8.4 and 4.100.22, also utilized orbital localization function.

CAUTION: The default orbital localization method cannot be used when diffuse functions are presented, otherwise the result may be misleading or completely meaningless! If diffuse functions have to be employed, you should change to Foster-Boys orbital localization method.

4.19.1 Localizing molecular orbital of 1,3-butadiene by Pipek-Mezey method

This section illustrates the use of orbital localization analysis of Multiwfn with *trans*-1,3-butadiene as example.

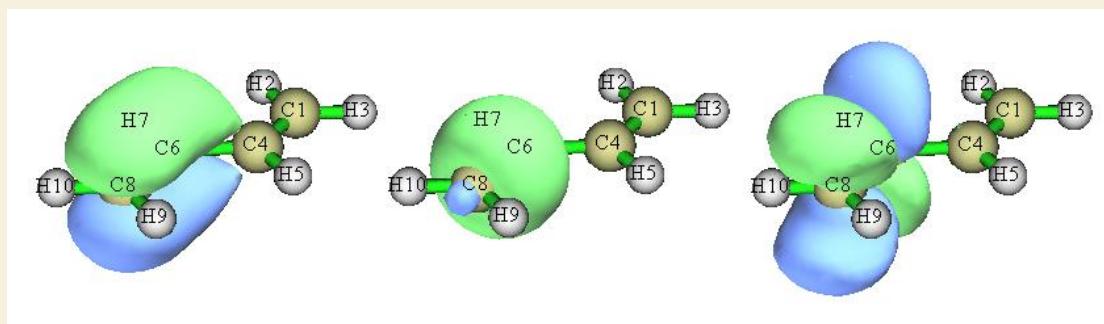
Basic steps of performing LMO analysis

The input file is *examples\butadiene.fch*, which was yielded at B3LYP/6-31G** level. You can first visualize its MOs via main function 0, you will find all MOs are highly delocalized, none of them have direct correlation with classical concept of chemical bond theory. Now we perform orbital localization to transform them to more chemically meaningful orbitals. Boot up Multiwfn and input *examples\butadiene.fch*

19 // Orbital localization. By default, Pipek-Mezey localization method based on Mulliken population is used

2 // Perform localization for both occupied and unoccupied MOs

Since this system is small, convergence of localization iteration immediately achieved, and the major character of the resulting orbitals are printed. After that Multiwfn automatically exports the localized orbitals as *new.fch* in current folder and then load it. Now, the orbital coefficients in memory have been completely updated to the localized orbitals. You can use different ways to characterize them. We enter main function 0 to visualize these orbitals, you will find all orbitals show strong localization character, in particular the occupied ones. Below are screenshot of three localized orbital involving C6-C8, the first two are occupied, and they correspond to σ bond and π bond, respectively; the third one is unoccupied, it can be ambiguously identified as anti- π bond orbital.



Obtaining LMO energies

It is also possible to obtain energy for the localized MOs, you should provide Fock matrix (strictly speaking, Kohn-Sham matrix in current case). To realize this for present system, we enter main function 100 and choose subfunction 2, and then select option 10 to write current molecular geometry as a Gaussian input file. We properly modify it to make Gaussian output NBO .47 file, which contains Fock matrix, and we need to ensure that the calculation level is identical to the .fch we used (*i.e.* B3LYP/6-31G**). The .gjf file after modification has been provided as *examples\butadiene_47.gjf*. Run it by Gaussian, you will find the resulting *BUTADIENE.47* at C:\ (this file has already been provided as *examples\butadiene.47*).

We have finished preparation. Now reboot Multiwfn and input

examples\butadiene.fch

19 // Orbital localization

-4 // Allow Multiwfn to generate and print energy for localized orbitals

examples\butadiene.47

2 // Localize both occupied and unoccupied orbitals

After convergence finished, Multiwfn evaluates energy of the localized MOs and sort orbitals according to their energies from low to high. The information of the highest occupied localized orbitals and lowest unoccupied localized orbitals are shown below

14	Energy:	-0.2750727 a.u.	-7.4851 eV	Type: A+B	Occ: 2.0
15	Energy:	-0.2750727 a.u.	-7.4851 eV	Type: A+B	Occ: 2.0
16	Energy:	0.2185910 a.u.	5.9482 eV	Type: A+B	Occ: 0.0
17	Energy:	0.2185912 a.u.	5.9482 eV	Type: A+B	Occ: 0.0

Please plot these orbitals in main function 0 to try to identify their characters.

Showing center position of all LMOs

It is possible to show center position of all generated LMOs, so that the distribution of the LMOs can be immediately understood. Here I illustrate how to realize this.

Reboot Multiwfn and input

examples\butadiene.fch

19 // Orbital localization

-8 // Switch status of “If calculating center position and dipole moment of LMOs” to “Yes”

-6 // Change localization method

10 // Foster-Boys

1 // Localize occupied orbitals (center position of unoccupied LMOs is generally uninteresting therefore we do not localize unoccupied MOs)

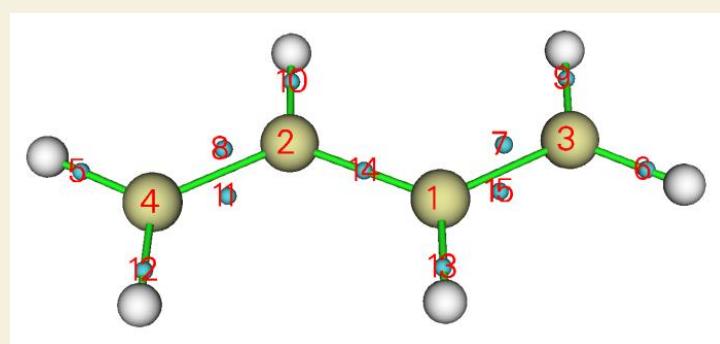
n // Skip the dipole moment analysis

As you can see from the prompts, after orbital localization has been finished as usual and the

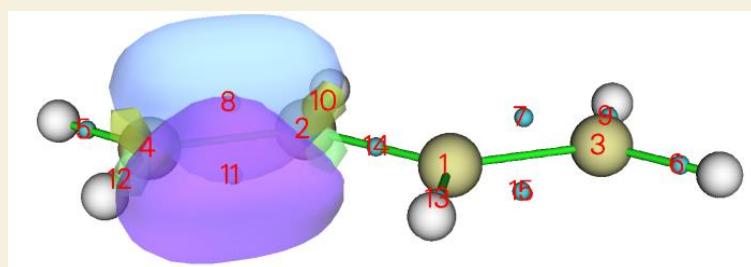
newly generated *new.fch* has been automatically loaded into Multiwfn, the program calculates center position of each LMO. In order to make visualization of the centers easy, Multiwfn adds Bq atoms (*i.e.* ghost atoms") into the current system to represent the centers of the LMOs. The center coordinates as well as correspondence between LMO indices and Bq indices are automatically exported to current folder as *LMOcen.txt*. The content of this file of current instance is

```
LMO    1 corresponds to Bq    11, x,Y,Z:   1.1372   0.7759   -0.0000 Bohr
LMO    2 corresponds to Bq    12, x,Y,Z:  -1.1372  -0.7759   -0.0000 Bohr
LMO    3 corresponds to Bq    13, x,Y,Z:   1.1372   3.3081   -0.0000 Bohr
...[ignored]
```

Then we return to main menu and enter main function 0 to visualize the LMO centers and orbital isosurfaces. The plotting settings have been automatically set to a special status for best visualizing LMOs purpose, currently you can see:

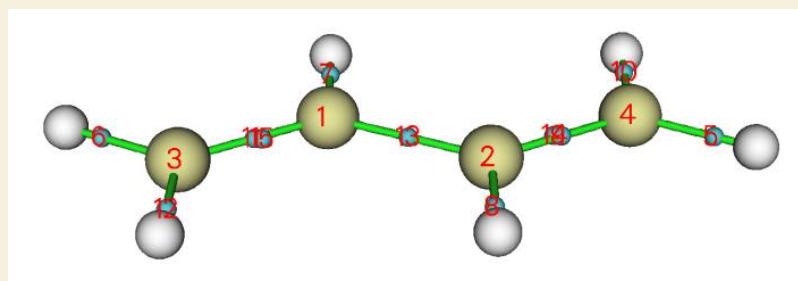


From this graph we can very clearly understand distribution of the generated LMOs. Each cyan sphere is a ghost atom, representing center of a LMO. Under current setting the index of the ghost atoms starts from 1, therefore the index shown in the graph directly corresponds to the LMO index. For example, we want to simultaneously visualize the two LMOs at boundary C-C bond, since the cyan spheres with labels 8 and 11 locate around the bond, we choose orbital 8 in the orbital list to display it, then select "Show+Sel. isour#2" and choose orbital 11 to. After that, change isovalue to 0.13 and set the drawing style as transparent, you will see



Evidently, the two LMOs obtained via Foster-Boys method collectively represent the double-bond character of the boundary C-C bond. The two LMOs are known as "banana" orbitals and do not exhibit σ - π separation character.

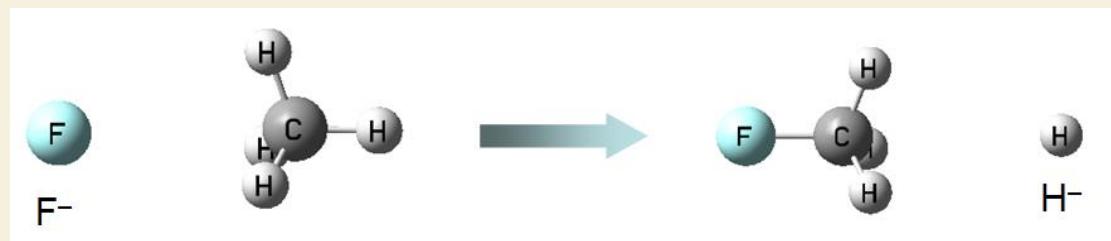
You can also try to visualize the center position of the LMOs generated via Pipek-Mezey algorithm. Because this method represents each double-bond as a pair of separated σ and β LMOs, whose centers should be very close to each other, from below graph you can find the centers indeed can hardly be distinguished:



It is worth to note that there are two ways to rapidly find the index of the LMO corresponding to the bond of your interest. The first one is examining the orbital compositions printed during the orbital localization, the second one is directly visualizing the LMO centers and find the index showing above the cyan sphere at proper place, as I just illustrated.

4.19.2 Analyze variation of localized molecular orbitals for SN2 reaction

In this example, we study variation of localized molecular orbitals (LMOs) along with reaction path to visualize variation of chemical bonds, a typical SN2 reaction is taken as instance.



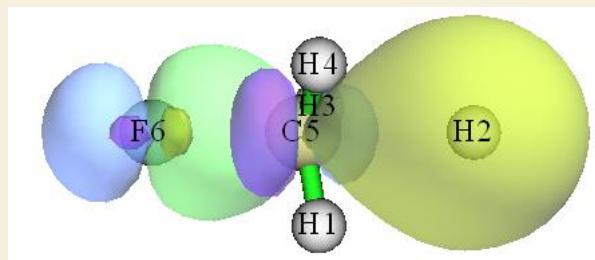
This SN₂ reaction involves formation of F-C bond and break of C-H bond, therefore in the following analysis we will focus on corresponding two LMOs. The IRC of the SN₂ reaction is shown below, five points are taken into account, their .fch files have been provided in *examples\SN2* folder.



First, we generate LMOs for transition state (TS) geometry. Boot up Multiwfn and input *examples\SN2\TS.fch*

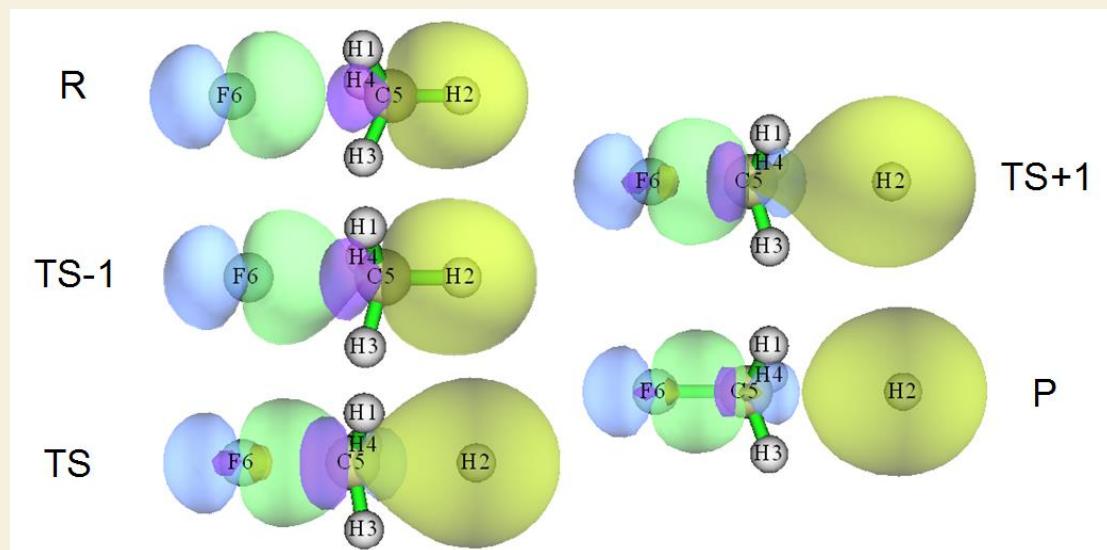
```
19 // Orbital localization
1 // Localize occupied orbitals
0 // Visualize orbitals
```

We can find there are two orbitals respectively corresponding to F-C and C-H bonds, we draw them together for easier comparison



The green-blue isosurface and purple-yellow isosurfaces clearly portray the orbitals corresponding to F-C and C-H bonds, respectively.

We draw the same kind plot for R.fch, TS-1.fch, TS+1.fch and P.fch, then put the graph together, as shown below



From the graph, the variation of chemical bonds during the SN₂ reaction is quite clear (R→TS-1→TS→TS+1→P). In the reactant state, the green-blue isosurface corresponds to the lone pair of fluorine atom, while the purple-yellow isosurface shows typical covalent bond character of C-H. As the reaction proceeds, the two LMOs vary smoothly, the C-F covalent bond character becomes more and more prominent, and in the final product state, the LMO represented by purple-yellow isosurface has already corresponded to 1s orbital of H⁻ anion.

4.19.3 Characterize Re-Re bond of [Re₂Cl₈]²⁻ anion

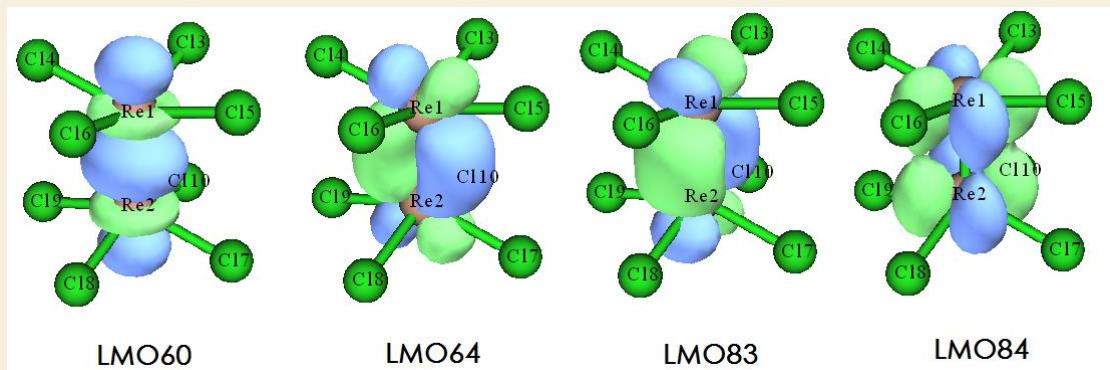
It is widely accepted that Re-Re bond in [Re₂Cl₈]²⁻ anion is a quadruple bond, with configuration of ($\sigma^2\pi^4\delta^2$). The σ bond results from overlap of d_{z^2} – p_z hybrid orbitals of the two rhenium atoms, the two π bonds stem from overlap of their d_{xz} and d_{yz} orbitals, while the δ bond is formed by overlap of their d_{xy} orbitals. Can this classic concept be validated via orbital localization analysis?

The .fchk file of [Re₂Cl₈]²⁻ anion produced under B3LYP with 6-31G* for Cl and SDD for Re is provided as *examples\Re2Cl82-.fchk*. Load it into Multiwfn and carry out orbital localization for occupied orbitals, from the output we can immediately identify the four LMOs corresponding to the Re-Re bond:

Almost two center LMOs: (Sum of two largest contributions > 85.0%)

57: 7(c1) 75.9% 2(Re) 21.8%	58: 10(c1) 75.9% 2(Re) 21.8%
59: 4(c1) 75.9% 1(Re) 21.8%	60: 2(Re) 49.7% 1(Re) 49.7%
61: 8(c1) 75.9% 2(Re) 21.8%	62: 3(c1) 75.9% 1(Re) 21.8%
64: 1(Re) 50.4% 2(Re) 50.4%	66: 6(c1) 75.9% 1(Re) 21.8%
67: 9(c1) 75.9% 2(Re) 21.8%	69: 5(c1) 75.9% 1(Re) 21.8%
83: 1(Re) 50.4% 2(Re) 50.4%	84: 2(Re) 50.9% 1(Re) 50.9%

Then we use main function 0 to visualize them, you will see (with default isovalue of 0.05)



From the isosurface maps of LMOs, it is clearly that LMO60 corresponds to σ bond, LMO64 and LMO83 correspond to π bond and LMO84 corresponds to δ bond. This observation supports the quadruple bond argument.

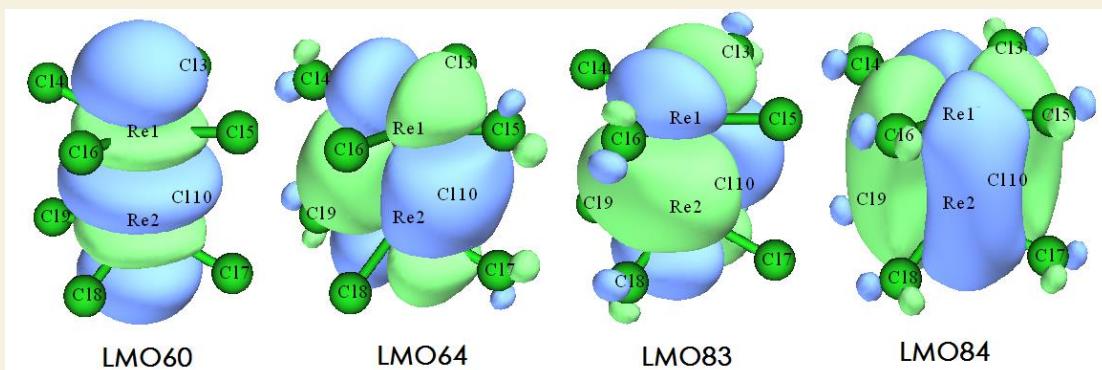
However, if we calculate Mayer bond order, we will find the situation is not so simple. The Mayer bond order of Re-Re bond calculated by main function 9 is 2.94, why the value is significantly lower than 4.0, which is expected?

To gain deeper insight, we perform "Orbital occupancy-perturbed Mayer bond order" analysis for Re-Re bond using main function 9, the output is

```
Mayer bond order after orbital occupancy-perturbation:
orbital    occ     Energy   Bond order  variance
...[Ignored]
      60    2.00000  -0.14354   2.001481  -0.935449
...[Ignored]
      64    2.00000  -0.12793   2.276388  -0.660542
...[Ignored]
      83    2.00000  -0.01749   2.276388  -0.660542
      84    2.00000   0.02809   2.466317  -0.470613
```

The result shows that the σ LMO has contribution of 0.935, each π LMO contributes 0.661, and the δ LMO contributes 0.471. Although in principle Mayer bond order cannot be exactly decomposed, these data is sufficient to help us to understand relative importance of each bonding LMO. Clearly σ bond is of most importance to the Re-Re bond, while the importance of the δ bond is relatively lowest.

Now a new problem arises, why the four LMOs have different contributions to Mayer bond order of Re-Re bond? This may be answered by visualizing their isosurfaces with lower value of isovalue. The graphs of the four LMOs with isovalue of 0.01 are illustrated below



From the graph we find that LMO60 basically only occurs around the two Re atoms, therefore its contribution to Re-Re Mayer bond order should be close to 1.0. Both LMO64 and LMO83 slightly delocalize to four Cl atoms, therefore they do not purely show Re-Re bond character and thus have lower contribution to Re-Re Mayer bond order. The LMO84 delocalizes to all the eight Cl atoms, it is naturally expected that its contribution to the Re-Re Mayer bond order should be the smallest.

By the way, if you have interesting, you can carry out orbital occupancy-perturbed Mayer bond order analysis for Cr_2 , you will find the Mayer bond order is almost exactly 6.0, and all the six MOs corresponding to Cr-Cr bond basically have contribution of 1.0, this is mainly because these orbitals do not delocalize to any other atoms.

4.19.4 Study bond dipole moment based on two-center LMOs for CH_3NH_2

Multiwfn supports a few methods for evaluating bond dipole moment, as mentioned in Section 4.A.11. In this section I use CH_3NH_2 as an example to show that based on two-center LMOs it is possible to study bond dipole moment, which somewhat reflects bond polarity. Introduction of related theory is given in Section 3.22.

Boot up Multiwfn and input

examples\CDA\CH3NH2\CH3NH2.fch

19 // Orbital localization

-8 // Switch status of “If calculating center position and dipole moment of LMOs” to “Yes”

1 // Localize occupied orbitals

y // Perform dipole moment analysis for the LMOs

Now *LMOdip.txt* has been generated in current folder, in this file you can find below content, which are calculated in a special method as shown in “Special topic 3” of Section 3.22:

Single-center orbital dipole moments (a.u.):

1 (5N)	x/Y/Z:	-0.04465	0.01500	0.00001	Norm:	0.04710
2 (1C)	x/Y/Z:	0.00010	0.00014	0.00000	Norm:	0.00017
9 (5N)	x/Y/Z:	-1.28448	0.41598	0.00001	Norm:	1.35016
Sum	x/Y/Z:	-1.32903	0.43112	0.00001	Norm:	1.39721

Two-center bond dipole moments (a.u.):

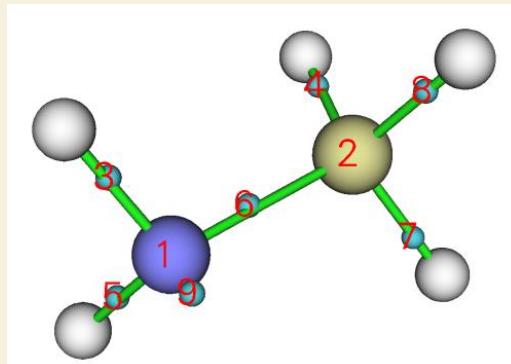
3 (5N - 6H)	x/Y/Z:	-0.08951	-0.20801	0.30542	Norm:	0.38021
4 (1C - 4H)	x/Y/Z:	-0.06753	0.08375	-0.00000	Norm:	0.10758

```

5 ( 5N - 7H ) X/Y/Z: -0.08950 -0.20802 -0.30543 Norm: 0.38022
6 ( 5N - 1C ) X/Y/Z: 0.17708 0.25312 -0.00000 Norm: 0.30891
7 ( 1C - 3H ) X/Y/Z: 0.08532 0.06534 -0.09566 Norm: 0.14387
8 ( 1C - 2H ) X/Y/Z: 0.08532 0.06534 0.09566 Norm: 0.14387
Sum X/Y/Z: 0.10119 0.05152 -0.00001 Norm: 0.11355

```

After entering main function 0, we can see below graph



By comparing the graph and content of *LMOdip.txt*, you can find LMO1 and LMO2, which respectively correspond to core orbitals of N5 and C1, have negligible polarity (*i.e.* the “norm” are basically zero), reflecting that the LMO centers are very close to the nuclear positions. The LMO9 corresponds to lone pair orbital of N5, its “norm” is as high as 1.35 a.u., showing that the LMO center deviates from N5 nucleus significantly. LMOs 3 and 5 correspond to the N-H bonds, LMOs 4, 7 and 8 correspond to the C-H bonds, it is well known that polarity of C-H should be lower than N-H, this point is well reflected by the difference in their “Norm” values. The “Norm” of LMO6, which corresponds to the C-N bond, is 0.3089, well indicating the fact that C-N is a polar bond.

In addition, in current system the Y coordinates of N5 and C1 are -1.438 and 1.330 Bohr, respectively. The Y component of the bond dipole moment of LMO6 is 0.253, which is an evident positive value. This observation shows that the negative and positive charge centers are on the N5 and C1 sites, respectively, corresponding to the fact that nitrogen has larger electronegativity than carbon.

4.20 Visual study of weak interactions

4.20.1 Studying weak interaction in 2-pyridoxine 2-aminopyridine by NCI method

Please carefully read Section 3.23.1 first to understand theory and how to use Multiwfn to carry out NCI analysis. Weak interaction character in 2-pyridoxine 2-aminopyridine system has already been studied using AIM theory in Section 4.2.1, in this section we perform NCI analysis, and meantime I will show how to plot color-filled map and AIM graph in a single map.

Boot up Multiwfn and input

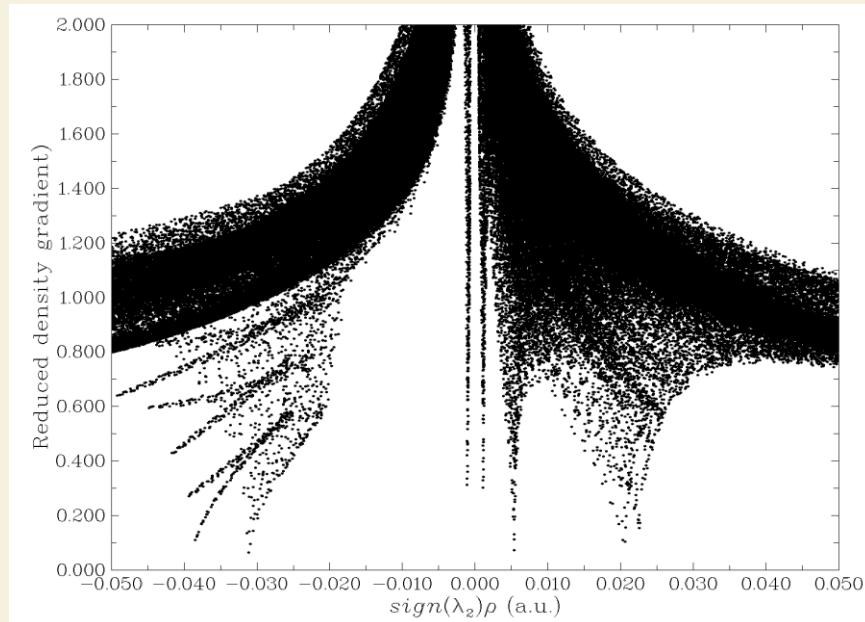
examples\2-pyridoxine_2-aminopyridine.wfn

20 // Visual study of weak interaction

1 // NCI analysis

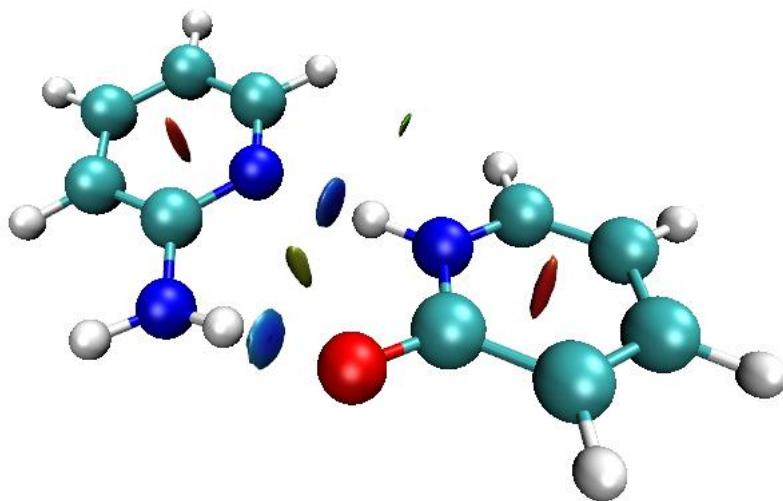
2 // Medium quality grid

After a while, calculation of grid data is finished. You can then select -1 to visualize scatter map, from which interactions in the system can be examined preliminarily.



Since there are spikes (points nearly approaching bottom) at very negative region of $sign(\lambda_2)\rho$, according to the description of NCI method given in Section 3.23.1, we immediately know that this dimer system must contain evident attractive intermolecular interaction. There is also a spike at very positive side, therefore steric effect should exist in present system.

Then select option 3 to export *func1.cub* and *func2.cub*, and use VMD to plot color-filled RDG map using the method described in Section 3.23.1, we obtain below graph



The type of interactions in this system now is very clear. Steric effect exists within the aromatic ring because the color of corresponding isosurfaces is red. The two hydrogen bonds N.....H-N and N-H.....O should be strong, since corresponding RDG isosurfaces have blue color. Between two hydrogens there is also a RDG isosurface, since its color is green, it should be regarded as van der Waals interaction, which is very weak. This region corresponds to the spikes in the middle of the

scatter map.

Below I will illustrate how to plot AIM critical points (CPs) and paths on the color-filled RDG map, the resulting graph will be more informative, since the trace of interactions can be vividly shown, while this kind of information is not explicitly revealed by NCI analysis.

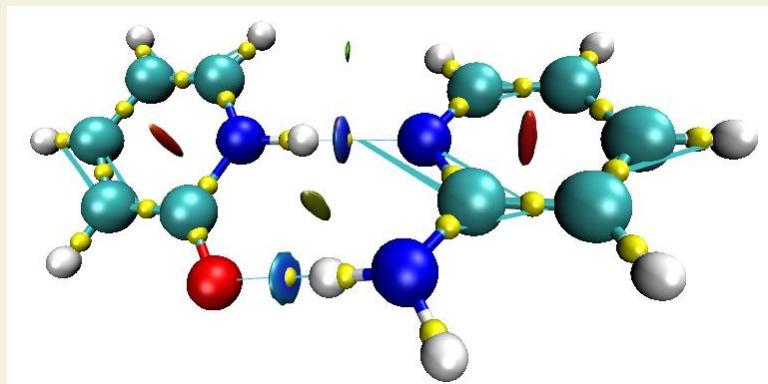
Return to main menu and input below commands to search CPs, generate paths and then export them as *CPs.pdb* and *paths.pdb* in current folder, respectively.

```

2 // Topology analysis
2 // Search nuclear CPs
3 // Search bond CPs
8 // Generate bond path
-4
6 // Export CPs as CPs.pdb
0 // Return
-5
6 // Export paths as paths.pdb

```

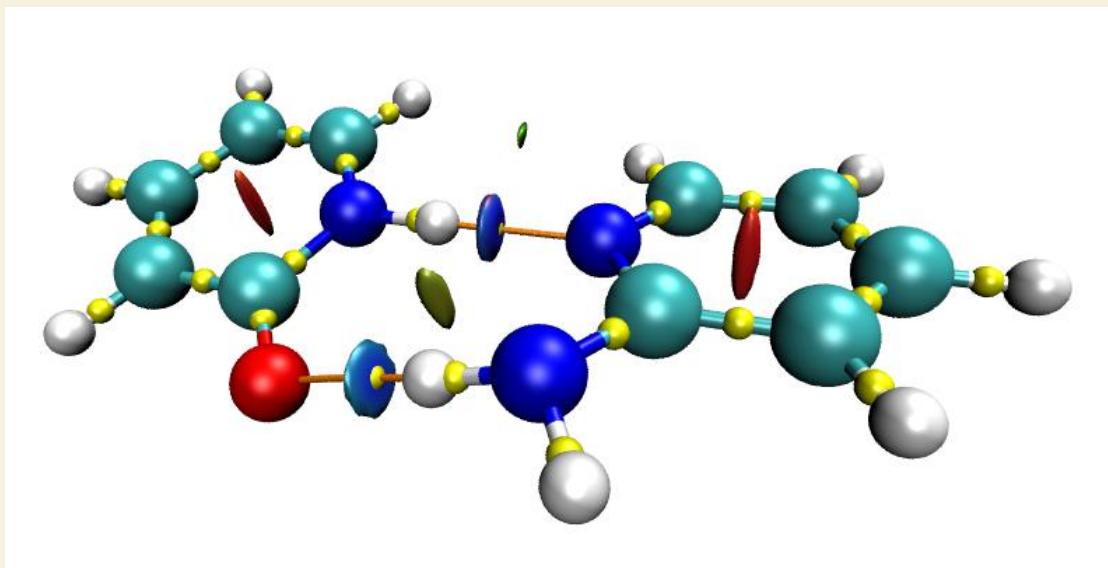
Then we close Multiwfn. Drag *CPs.pdb* and *paths.pdb* into VMD main window in turn to load them, select "Graphics"- "Representation", change "Selected molecules" to the second term (corresponding to *CPs.pdb*), change "Drawing Method" to "VDW" and set "Sphere Scale" from the default 1.0 to the minimal value 0.1. Note that in the *CPs.pdb* file, C, N, O, F atoms correspond to (3,-3), (3,-1), (3,+1), (3,+3), respectively. Here we only want to draw bond CPs (*i.e.* (3,-1) type of CPs) on the graph with yellow color, therefore we input "nitrogen" in "Selected Atoms" text box and press ENTER button, then change "Coloring Method" to "Color ID" and select "4 yellow" in the drop-down box. Currently, the graph looks like below



Probably you feel that the spheres corresponding to CPs are too large, however we cannot further decrease the "Sphere Scale" using graphical window due to limitation of VMD. To make the spheres smaller, you have to use corresponding command in VMD console window. To find proper command to do this, we select "File"- "Log Tcl Commands to Console", then change "Sphere Scale" to other value (*e.g.* 0.2), you will immediately see corresponding text-line command in the VMD console window, for present the command is *mol modstyle 0 1 VDW 0.200000 12.000000*, where the argument 0.2 corresponds to size of the spheres. Therefore, to decrease the sphere size to *e.g.* 0.09, we should input *mol modstyle 0 1 VDW 0.09 12.000000* in the console window, then in the VMD graphical window you will see the spheres have already become smaller.

Next, we change the appearance of paths. In the "Graphics"- "Representation" panel, select the third term in "Selected molecules" (corresponding to *paths.pdb*), change the drawing method to

"VDW", set coloring method as "Color ID" and select "3 orange", then use abovementioned skill to set the sphere scale to 0.02. The final graph is shown below.

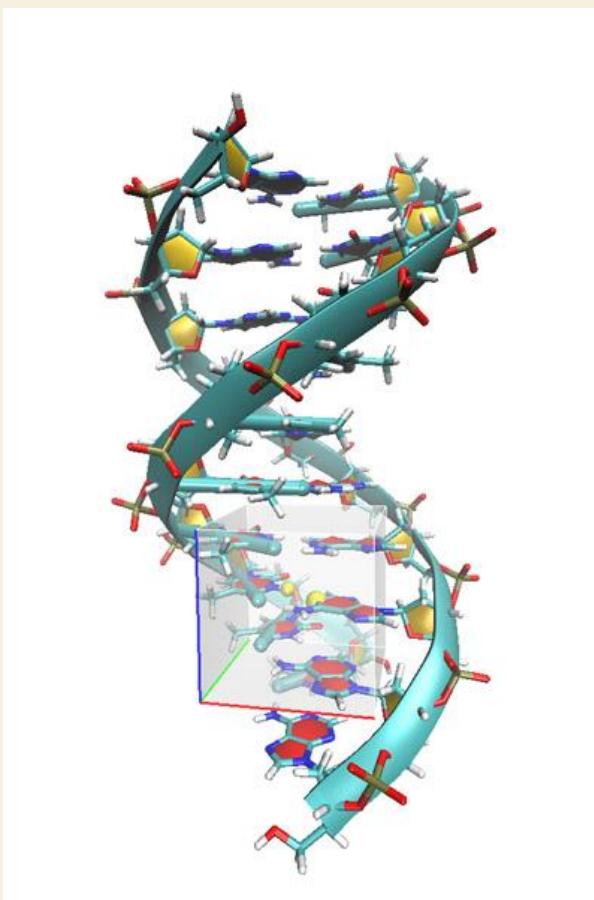


From this graph, not only the weak interaction regions are clearly revealed, but also the interaction paths are vividly exhibited. Notice that there is no CP and path corresponding to the hydrogen-hydrogen interaction, because in this region there is no position having vanishing electron density gradient, and this is why in the scatter map the spike corresponding this H-H interaction does not completely approach the bottom of the map. This observation reflects an advantage of NCI analysis over AIM analysis, namely interaction can be revealed even if there is no corresponding bond CP.

If you feel the steps of showing CPs in VMD are too lengthy, you can use the pre-provide scripts to do this, see Section 4.2.5.

4.20.2 Studying weak interaction in DNA by NCI method based on promolecular density

Please read Sections 3.23.1 and 3.23.2 if you are not familiar with NCI analysis and the concept of promolecular approximation. In this example, we will carry out NCI analysis with for a DNA segment consisting of 10 base-pairs. Since this system is fairly large, promolecular approximation is used for fast construction of its electron density, which is needed by NCI analysis. Here we only study weak interaction character of a local DNA region, which is enclosed in the transparent box:



Boot up Multiwfn and input:

examples\DNA.pdb

20 // Visual study of weak interaction

2 // NCI analysis based on promolecular density

7 // Use mode 7 for defining grid data

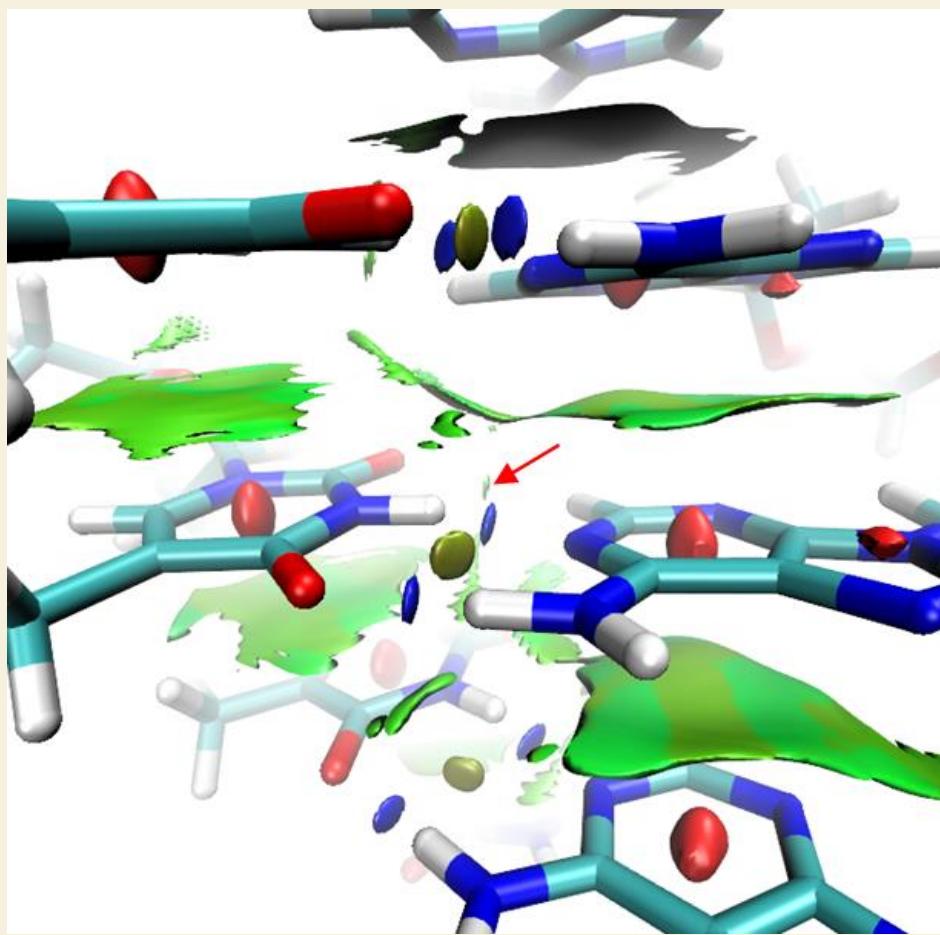
84,565 // Use midpoint of atom 84 and 565 as center of grid data. You can view molecule structure in your favourite visualization tool to find two proper atoms used to define center

120,120,120 // Because the spatial scope of grid data is large, we need relatively large number of grid points, otherwise the grid spacing will be too large, which results in bad quality of RDG isosurfaces

9,9,9 // Set the extension distances in all directions to 9 Bohr

Hint: You can also use mode 10 to set up box interactively in a GUI window, the box size and position of box center is more controllable

After the calculation of grid data is finished, choose option 3 to export $\text{sign}(\lambda_2)\rho$ and RDG as *func1.cub* and *func2.cub* respectively in current folder, and then copy them as well as *examples\RDGfill_pro.vmd* to VMD installation folder. Boot up VMD and input *source RDGfill_pro.vmd* in console window to draw color-filled RDG isosurface map, the resulting graph after some adjustments is shown below. (For better visualization effect, open "graphics"- "Representation" and change the drawing method of DNA to Licorice, change bond radius to 0.2. Then enter "Display"- "Display settings..." to set "Cue Mode" to "Linear" and set "Cue Start/End" to 2.25 and 3.75, respectively, so that distant atoms can be substantially screened). Finally, you will get the graph below



It is clear that there are π - π stacking interaction between neighbouring base pairs (big flat isosurfaces), and there are two strong hydrogen bonds among 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 it can only be regarded as vdW interaction.

The default isovalue 0.3 in RDGfill_pro.vmd is suitable for present case, but may be not suitable for exhibiting weak interaction region of other systems, in that situations you need to adjust it manually. You can either edit the .vmd file, or choose “Graphics”-“Representation” in VMD, then select the representation with style “Isosurface” and reset isovalue by inputting expected value in text box.

4.20.3 Visually studying weak interaction for water in bulk environment by aNCI method

If you are unfamiliar with NCI and aNCI methods, please first read the introduction given in Section 3.23.1, 3.23.2 and 3.23.3. The aNCI method illustrated in this section is a generalization of NCI analysis method for fluctuation environment, *e.g.* molecular dynamics (MD) process.

In this example I will show how to use Multiwfn to visually study the weak interaction between water molecules in the MD simulation of bulk water system. You can use any program to perform the MD simulation, as long as you know how to convert the resulting trajectory from private format to the general .xyz format, which can be recognized by Multiwfn and utilized in aNCI analysis.

Here I assume that you are a GROMACS 4.5 user. The detailed steps of the MD process are given below (very different to GROMACS ≥ 5.0), all of the related files can be found in *examples\aNCl* folder. If you do not want to perform the MD simulation by yourself, you can directly download the *wat.xyz*, which will be utilized in aNCI analysis later: http://sobereva.com/multiwfn/extrafiles/aNCI_wat_xyz.zip.

Generating MD trajectory by GROMACS

First, build a file named "emptybox.gro", which records a blank box, and the side length in each direction is 2.5nm. Then run below command to fill the box with waters.

```
genbox -cp emptybox.gro -cs spc216.gro -o water.gro
```

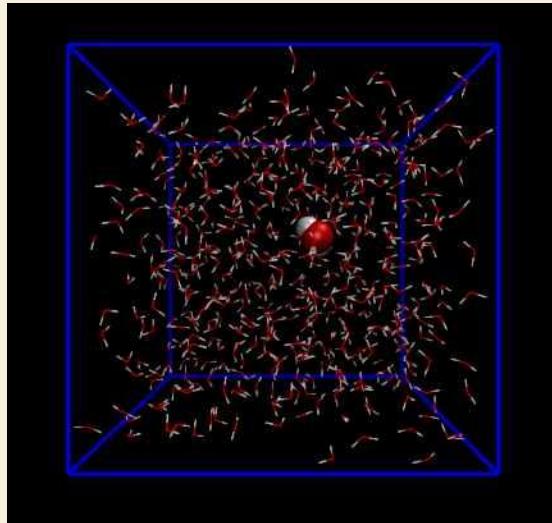
Run below command and select "GROMOS96 53a6 force field" to obtain the top file of the bulk water system. SPC/E water model is employed.

```
pdb2gmx -f water.gro -o water.gro -p water.top -water spce
```

Then carry out NPT MD by 100ps to equilibrate the bulk water at 298.15K, 1atm environment.

```
grompp -f pr.mdp -c water.gro -p water.top -o water-pr.tpr
mdrun -v -deffnm water-pr
```

Using VMD program to load water-pr.gro, select a water close to the center of the box. We select the water with resid index of 101, which is highlighted in below graph. Note that the two hydrogens in this water have index of 302 and 303, and the index of the oxygen is 301.



This water will be freezed in the following MD simulation. In order to do so, we generate index file, namely inputting below commands

```
make_ndx -f water-pr.gro
ri 101
q
```

Run following command to do 1ns equilibrium MD simulation at 298.15K, the trajectory will be saved every 1ps, and finally we will obtain 1000 frames. The water with resid index of 101 is freezed via the keyword "freezegrps = r_101". Note that NVT ensemble instead of NPT is used,

because NPT process will scale the coordinate of the atoms, which somewhat destroys the effect of freezing.

```
grompp -f mdmdp -c water-pr.gro -p water.top -o water-md.tpr -n index.ndx  
mdrun -v -deffnm water-md
```

Load the *water-pr.gro* into VMD, then load *water-md.xtc* to the same ID, select "File"- "Save Coordinate..." option and set the file type as xyz, then input *all* in the "Selected atoms" box, input *1* and *1000* in the "First" and "Last" window, respectively. Finally, click "Save" button to convert the GROMACS trajectory to *wat.xyz*.

IMPORTANT NOTICE: The *wat.xyz* currently records atom names rather than element names. For example, if you open this file via text editor, you will find each water contains OW, HW1 and HW2, which are atom names. However, in standard .xyz file, only atom elements should be recorded. Therefore, in general cases, you should manually replace all atom names in the .xyz file generated by VMD with element names. Fortunately, this step can be skipped in present example because there is no element in the periodic table named OW, HW1 and HW2, therefore, only the first letter of atom names will be employed by Multiwfn to try to identify their elements, and they can be properly recognized as oxygen and hydrogens, since after loading the .xyz file, you can find prompt "Formula: H1022 O511" on the screen, which is what we expected. If you find there are undesired elements in the the "formula", that means you have to replace the corresponding atom names in the .xyz file as their actual element names.

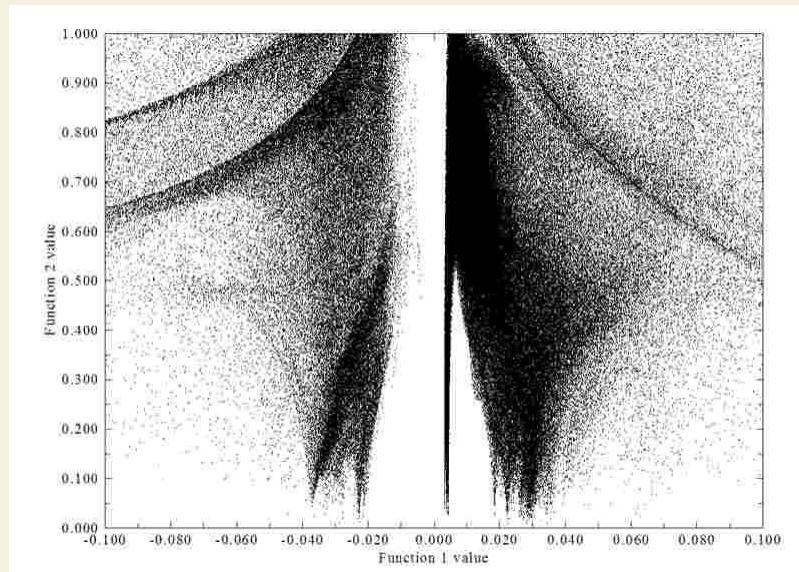
Generating grid data by Multiwfn

Boot up Multiwfn and input following commands

```
wat.xyz  
20 // Visual study of weak interaction  
3 // aNCI analysis  
1,1000 // The range of the frames to be analyzed  
7  
301,301 // Using atom 301 (the oxygen of the freezed water) as the box center of the grid  
data  
80,80,80 // The number of grid points in each side  
4.5,4.5,4.5 // Extend 4.5 Bohr in each side
```

Now Multiwfn starts to calculate electron density, its gradient and Hessian of each frame, then their average quantities will be obtained, and finally Multiwfn calculates average RDG and average $\text{sign}(\lambda_2)\rho$. The whole process is time-consuming; at a common Intel 4-cores computer about half an hour will be consumed. (Note that the electron density I referred here is produced by promolecular approximation, which is constructed by simply superposing the density of the atoms in their free-states)

After the calculation is finished, you can select option 1 to check the scatter plot between average RDG (X-axis) and average $\text{sign}(\lambda_2)\rho$ (Y-axis), see below, there is also option used to export the corresponding data points to plain text file.



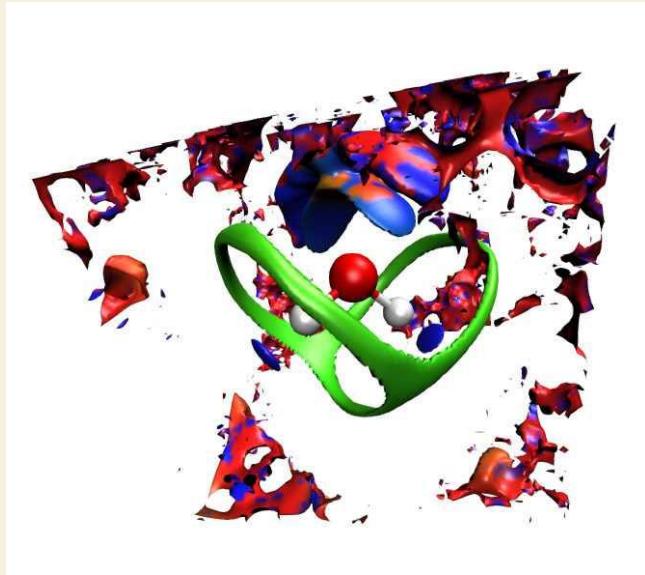
Select option 6 to export the grid data of average RDG and average sign(λ_2) ρ as *avgRDG.cub* and *avgl2r.cub* in current folder, respectively.

Since we wish to check the stability of weak interaction, we also select 7 to export thermal fluctuation index to *thermflu.cub* in current folder. Note that this process requires recomputing electron density of each frame, and thus is time-consuming.

Analysis

Copy *avgRDG.cub*, *avgl2r.cub*, *thermflu.cub* as well as *avgRDG.vmd* and *avgRDG_TFI.vmd* in "examples\aNCl" folder to the directory of VMD program.

Simply boot up VMD and input *source avgRDG.vmd* in its console window, the average RDG isosurface will be shown with isovalue of 0.25, meantime the average sign(λ_2) ρ is mapped on the isosurface by various colors. In order to make the graph clearer, one should screen unrelated atoms, that is enter "Graphics" - "Representation", then select the entry whose "style" is "CPK", and input *serial 301 302 303* in the "Selected Atoms" box and then press ENTER button. Now only the water with resid index of 101 presents in the graph. After proper rotation and translation of view, you will see



Unfortunately, around the water of interest, there are large amount of noisy isosurfaces, which somewhat messed up the graph, thus it is better to shield them. This aim can be achieved by main function 13 of Multiwfn, the steps are described below.

Then boot up Multiwfn and input

avgRDG.cub

13 // Process grid data

13 // Set the value of the grid points far away from specific atoms

1.5 // If the distance between a grid point and any selected atoms is longer than 1.5 times of vdW radius of corresponding atom, then the value of the grid point will be set as given value

100 // An arbitrarily large value (should be larger than the iso value of the RDG isosurfaces)

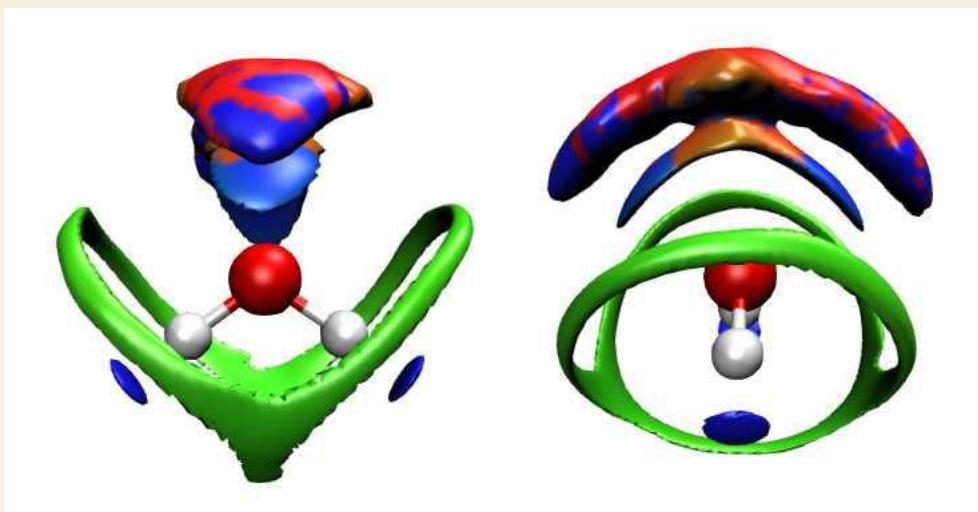
2 // Inputting selected atoms by hand

301-303 // The index of the atoms are 301, 302 and 303

0 // Export the updated grid data to a new cube file

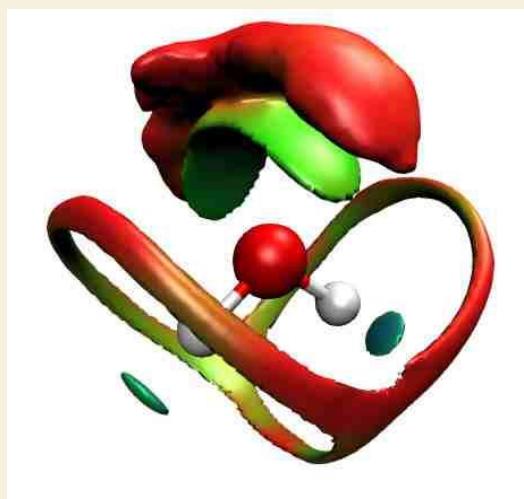
avgRDG.cub // The name of the new cube file

Copy the newly generated *avgRDG.cub* to the folder of VMD program to overwrite the old one, then use the script *avgRDG.vmd* again to plot the graph, after some adjustments you will see (for clarity, the view of two sides are shown at the same time)



The graph we obtained this time is very clear. The color scale is from -0.25 to 0.25, corresponding to the color variation of Blue-Green-Red. More blue denotes electrostatic interaction or H-bond effect in corresponding region is stronger, and more red suggests more intensive steric effect. Green region implies low electron density, corresponding to vdW interaction. From the graph one can see that there are two blue ellipses near the two hydrogens, rendering that in the MD process, strong H-bonds are formed due to the O-H group. The slender green isosurface exhibits in which direction this water prefers to interact with other waters by vdW interaction. There is a big lump of isosurface above the oxygen, on which the red color appears in the middle part, while blue color occurs at the two ends; the latter reflects that the two lone pairs of the oxygen act as H-bond acceptors during the simulation, while the former reveals the repulsive interaction zone between waters.

Next, we study the stability of the weak interactions. First disable present isosurfaces, and then input the command *source avgRDG_TFI.vmd* in the console window, after some adjustments you will see



The color scale is 0~1.5, still corresponding to the color transition of Blue-Green-Red. More blue (red) means the thermal fluctuation index (TFI) is smaller (larger), and hence the weak interaction in corresponding region is more stable (unstable). The graph shows that it is stable that the water behaves as H-bond donor, while the stability of the water acting as H-bond acceptor is slightly weaker; the vdW interaction region is totally red, rendering that vdW interaction is evidently unstable compared to hydrogen bond.

4.20.5 Simultaneously revealing covalent and noncovalent interaction in phenol dimer by DORI analysis

Before reading this section please read Section 3.23.5 to understand basic knowledge about DORI (Density Overlap Regions Indicator). Here I use phenol dimer as instance to show how to realize DORI analysis, you will find almost all steps are identical to the NCI analysis described in Section 3.23.1.

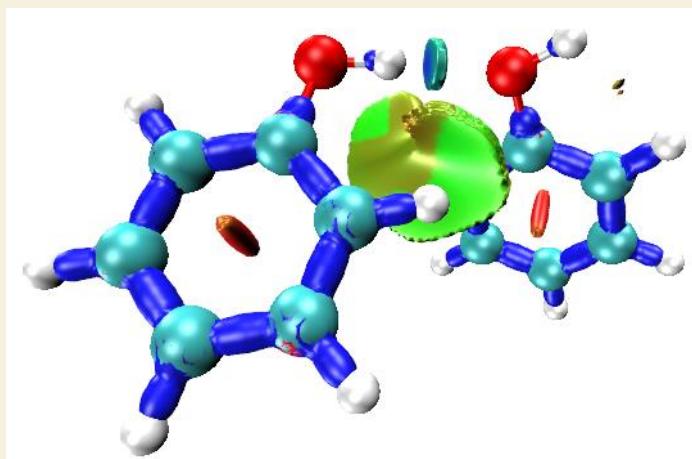
Boot up Multiwfn and input
examples\PhenolDimer.wfn

```

20 // Visual study of weak interaction
5 // DORI analysis
-10 // Modify extension distance
0 // Set extension distance to zero, namely let the box of grid data just enclose the system
2 // Medium quality grid
3 // Export cube file

```

Move func1.cub, func2.cub and plotting script *examples\DORIfill.vmd* to VMD folder. Then boot up VMD and input *source DORIfill.vmd* in console window, you will immediately see below graph. It is suggested to enable more lightings by choosing "Display" - "Light 3" so that the map becomes brighter.



The graphic effect is clearly satisfactory. The weak interaction regions are exhibited almost the same as RDG, while the strong chemical bonding regions are nicely portrayed as blue isosurfaces. Note that the most suited isovalue of DORI must be different for different systems, the default isovalue is 0.95. You can modify it in the .vmd script file, or manually adjust it in "Graphics" - "Representations" panel of VMD.

4.20.10 Visualize and quantify intermolecular and intramolecular interactions by means of Independent Gradient Model (IGM)

Please read Section 3.23.5 to gain basic knowledges about the Independent Gradient Model (IGM) method proposed in *Phys. Chem. Chem. Phys.*, **19**, 17928 (2017). If you are not familiar with NCI analysis, you should also first read Section 3.23.1, since many aspects of IGM analysis are closely related to the NCI analysis. In this section I will illustrate the use of IGM analysis in Multiwfn via several examples. More discussions and instances can be found in my blog article "Investigating intermolecular weak interactions via Independent Gradient Model (IGM)" (in Chinese, <http://sobereva.com/407>).

Notice that the IGM method in Multiwfn is based on promolecular approximation, hence you can use any kind of input file containing atomic coordinate information, such as .xyz and .pdb (see Section 2.5 for details). The VMD program used throughout this section is version 1.9.3, it is freely available at <http://www.ks.uiuc.edu/Research/vmd/>.

Example 1: Guanine-cytosine (GC) base pair

The IGM framework includes many useful ideas and defined many useful concepts, a series of analyses will be conducted in this example. A simple system guanine-cytosine (GC) base pair is taken as instance here. Some analyses may be ignored in practical studies, and sequence of analyses is completely arbitrary.

(1) Studying δg function

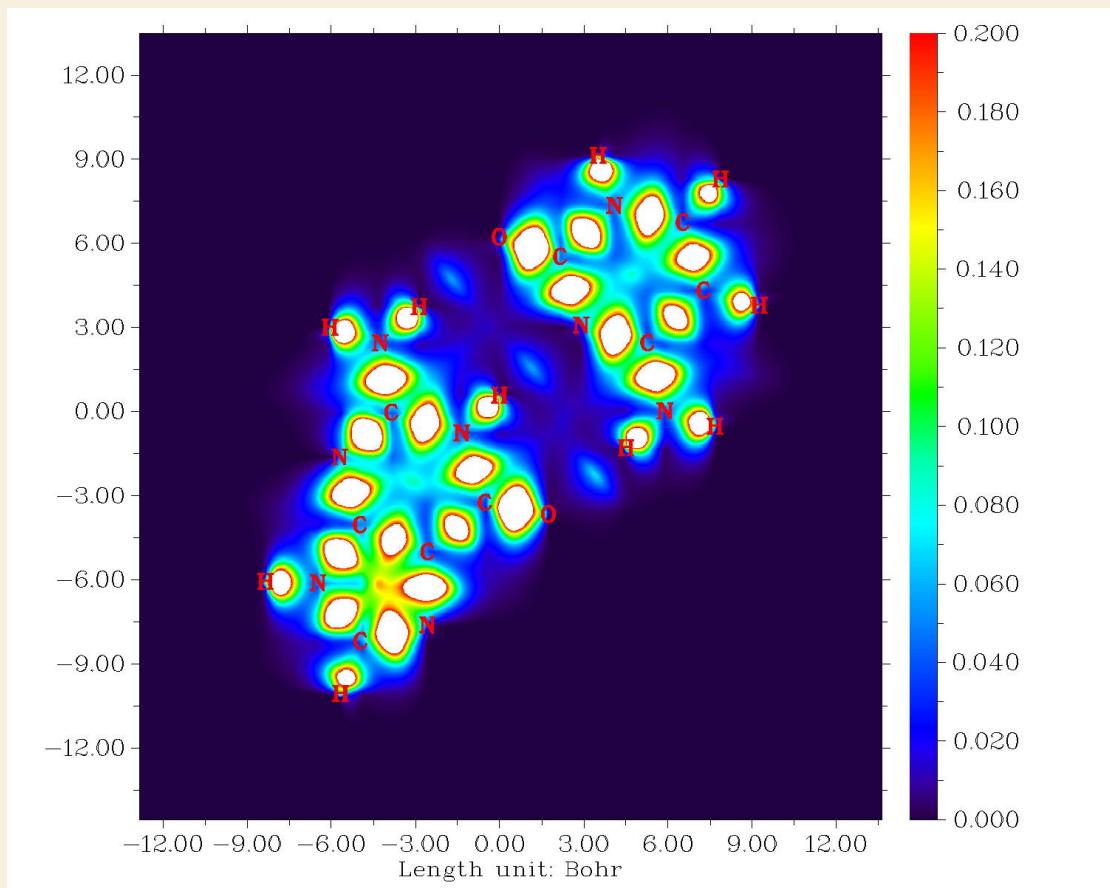
We first study the distribution character of δg function by plotting it as color-filled plane map. Boot up Multiwfn and input below commands

```
examples\GC.pdb
4 // Plot plane map
22 //  $\delta g$ 
1 // Color-filled map
Press ENTER to use default grid setting
1 // XY plane
0 // Z=0
```

The graph shown on screen currently looks obscure, this is because the default color scale is not suitable for present case, so we close the graph and input

```
1 // Set color scale
0,0.2 // Lower and upper limits
4 // Show atomic labels
1 // Red color
-2 // Set stepsize of axes
3,3,0.02 // Step size for X, Y and color bar
-1 // Plot again
```

You will see below graph

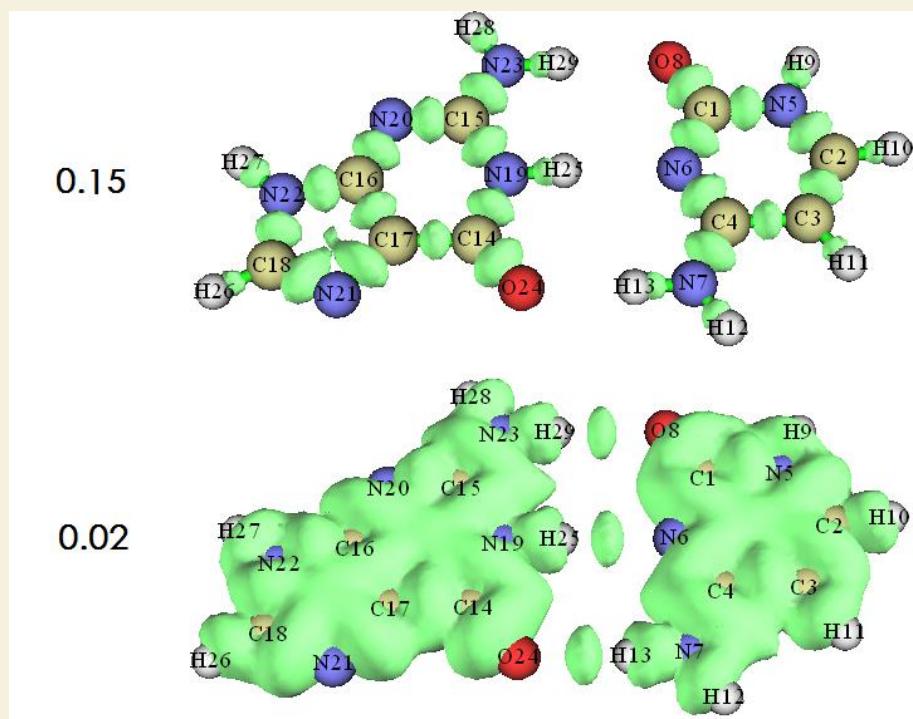


Above graph clearly reveals all interatomic interactions, and the magnitude of δg is positively relevant to interaction strength. As can be seen from the graph, all chemical bond regions have large δg value (the region with value higher than 0.2 is shown as white). The δg function also outlines three hydrogen bond regions among the base pair, where the δg function has evidently smaller value compared to the chemical bond regions.

The δg can also be plotted as isosurface map. Return to main menu and input

```
5 // Calculate grid data
22 //  $\delta g$ 
2 // Medium quality grid
-1 // Show isosurface
```

The isosurface with isovalues of 0.15 and 0.02 are shown below (you can use higher quality of grid or set the extension distance of grid data smaller to make the graph more smooth)



Since chemical bond regions have relatively large value of δg , only chemical bonding interactions are visible when isovalue is set to 0.15. Clearly, δg may be used as a function to exhibit chemical bonds like ELF and DORI functions, with additional advantage that only geometry information is needed. The weak interaction regions can also be simultaneously visualized when isovalue is decreased to a small value, e.g. 0.02.

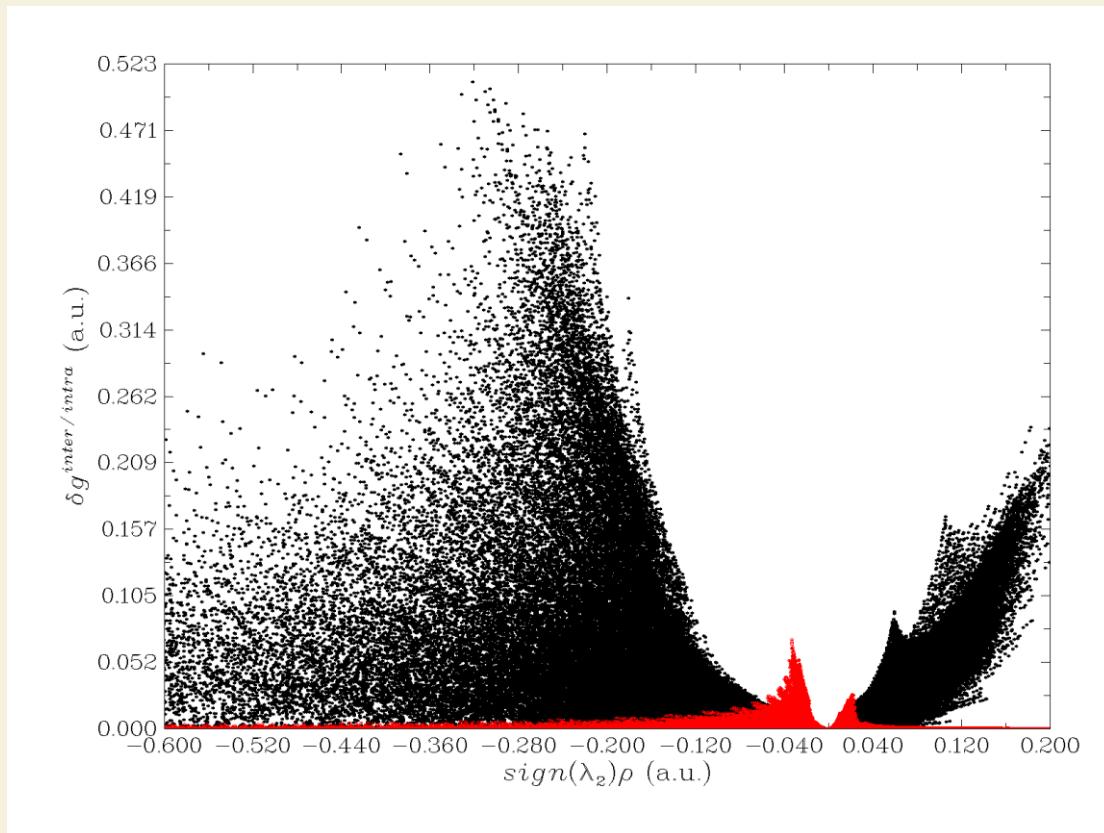
(2) Studying δg_{inter} function between base pair

The δg^{inter} is a key function in the IGM analysis framework, it is designed to reveal interaction regions between two (or even more) fragments defined by users. Here we plot this function to study the interactions between the two bases. Although as shown earlier, these interactions can also be revealed by simply drawing δg , the isosurfaces corresponding to intrafragment interactions severely polluted the graph. Fortunately, the IGM analysis allows us to separate the δg as δg^{inter} and δg^{intra} , which solely reflect the contribution to δg due to interfragment and intrafragment interactions, respectively.

Return to main menu and input following commands

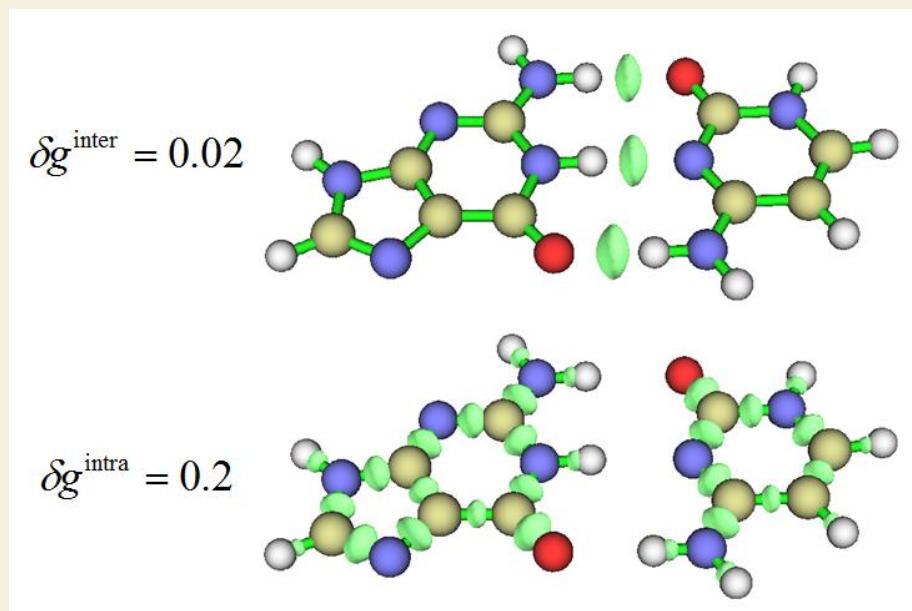
```
20 // Visual study of weak interactions
10 // IGM analysis
2 // Define two fragments
1-13 // Range of atoms in the first base
14-29 // Range of atoms in the second base
2 // Medium quality grid
```

After calculation, you will see a post-process menu. The meaning of each option has been explained in Section 3.23.5. You can use option -1 to directly draw scatter map between δg or δg^{inter} or δg^{intra} versus $\text{sign}(\lambda_2)\rho$. We choose option -1 and then select suboption 4, in the resulting map the red and black points correspond to δg^{inter} and δg^{intra} , respectively.



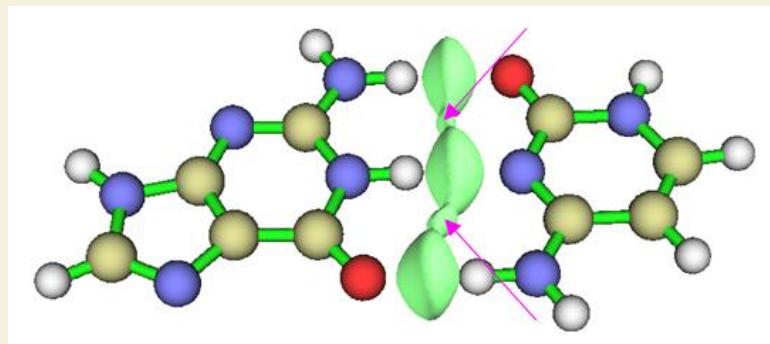
If you are familiar with NCI method, you will naturally know how to discuss this graph, now we try to identify character of peaks in the scatter graph. In the region where $\text{sign}(\lambda_2)\rho$ is about -0.04, you can find that the δg^{inter} has a remarkable peak (with height about 0.06), which implies presence of hydrogen bonds. If δg^{inter} isosurface is set to an isovalue lower than about 0.06, the corresponding isosurfaces should be visible in the graph. In the region where $\text{sign}(\lambda_2)\rho$ is approximately +0.02, there is also a small peak of δg^{inter} . Since positive $\text{sign}(\lambda_2)\rho$ implies repulsive interaction, the peak may reflect weak steric regions in the center of the two rings between the two bases. In above scatter map, there is a very prominent peak of δg^{intra} around $\text{sign}(\lambda_2)\rho = -0.3$. Since this peak corresponds to intrafragment interaction, and corresponding $\text{sign}(\lambda_2)\rho$ is not only negative but large, rendering attractive and strong interaction, the peak must result from chemical bond.

Using Multiwfns you can directly visualize isosurface of δg^{inter} and δg^{intra} . To do this, we close the scatter map, select "4 Show isosurface of grid data", then choose corresponding options and properly set isovalue, you will obtain below isosurface graphs



As can be seen, δg^{inter} and δg^{intra} indeed solely exhibit inter- and intra-fragment interactions, respectively. This greatly facilitates separate discussion of the two kinds of interactions.

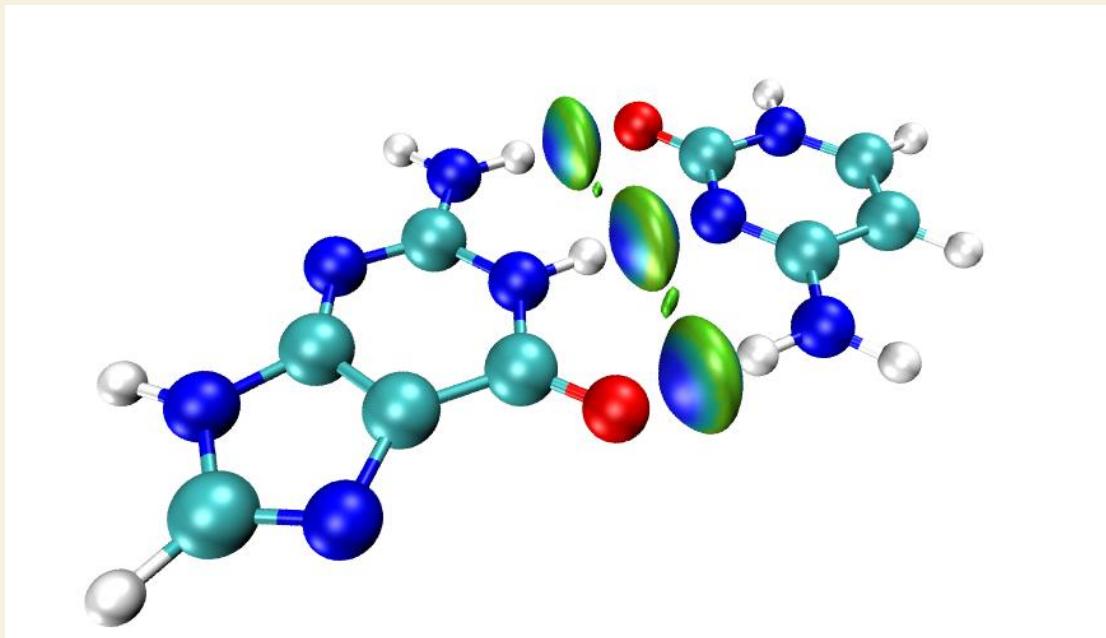
From the above $\delta g^{\text{inter}}=0.02$ isosurface map we are only able to visualize hydrogen bond regions. To find steric region in the ring center between the two bases, we should further decrease the isovalue of δg^{inter} to e.g. 0.08, as shown below. The ring-center steric regions are highlighted by arrows.



(3) Drawing $\text{sign}(\lambda_2)\rho$ mapped δg^{inter} isosurfaces

If $\text{sign}(\lambda_2)\rho$ is mapped to δg^{inter} isosurfaces by different colors, then one can not only recognize where weak interactions occur, but also immediately capture the character of the interactions. Multiwfn itself is currently unable to plot color-filled isosurface map, we need to use VMD program to do this, just like what we do in the NCI analysis.

Select "3 Output cube files to current folder" in IGM post-process menu, then $\text{sign}(\lambda_2)\rho$, δg , δg^{inter} and δg^{intra} will be exported to sl2r.cub, dg.cub, dg_inter.cub and dg_intra.cub in current folder, respectively. Move sl2r.cub and dg_inter.cub as well as VMD plotting script examples\IGM_inter.vmd into VMD folder. Boot up VMD, input source IGM_inter.vmd in the console window, you will immediately see below graph (I also selected "Display"- "Light 3" in VMD main menu to make the graph brighter)



The default isovalue employed by the IGM_inter.vmd script is 0.01, you can manually change isovalue by dragging the isovalue bar in "Graphics"- "Representation" panel. In the script the default color scale of $\text{sign}(\lambda_2)\rho$ ranges from -0.05 to 0.05, and the default color transition is Blue-Green-Red. Therefore, the more blue the isosurface, the stronger the attractive interaction, while the more red the isosurface, the larger the steric effect. Green zone in a isosurface implies that the corresponding interaction is weak and may be regarded as van der Waals interaction.

Similarly, you can plot $\text{sign}(\lambda_2)\rho$ mapped δg^{intra} isosurfaces. Just move sl2r.cub and dg_intra.cub as well as the corresponding VMD plotting script *examples\IGM_intra.vmd* into VMD folder, then input *source IGM_intra.vmd* in the console window to execute it.

(4) Decomposing interfragment interaction as atom and atom pair contributions

The atom pair δg index is a quantitative indicator of contribution of an atomic pair to total δg^{inter} between two fragments, while sum of all atom pair δg indices of a given atom yields its atom δg index, which represents its importance for interfragment interaction. In Multiwfn the atom and atom pair δg indices can be easily evaluated. In the IGM post-process menu, select option 6, then Multiwfn will calculate these indices. The result will be exported to atmdg.txt in current folder, and then the program asks you if also exporting atmdg.pdb in current, we choose *y* in present case.

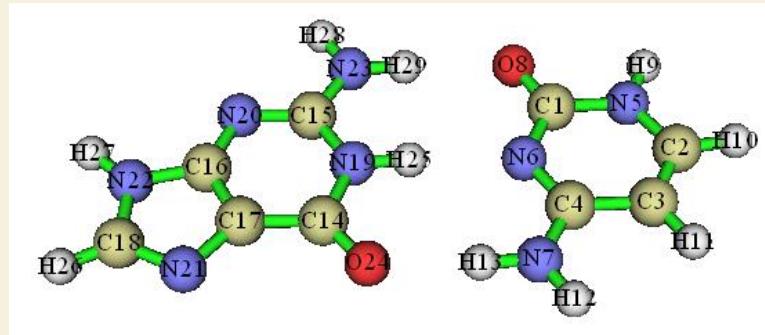
Part of content of atmdg.txt is pasted here:

```
Atom dg index in fragment 1 (zero terms are not shown)
Atom   6 :  0.388508
Atom  13 :  0.369710
Atom   8 :  0.322453
[ignored...]

Atom dg index in fragment 2 (zero terms are not shown)
Atom  25 :  0.458048
Atom  24 :  0.332201
Atom  29 :  0.297980
[ignored...]
```

```
Atom pair dg index (zero terms are not shown)
 13 24 : 0.187852
  6 25 : 0.175460
  8 29 : 0.163365
```

If you compare above data with the structure graph of present system shown below, you will find the atom and atom pair δg indices very meaningful and useful for discussing interfragment interactions

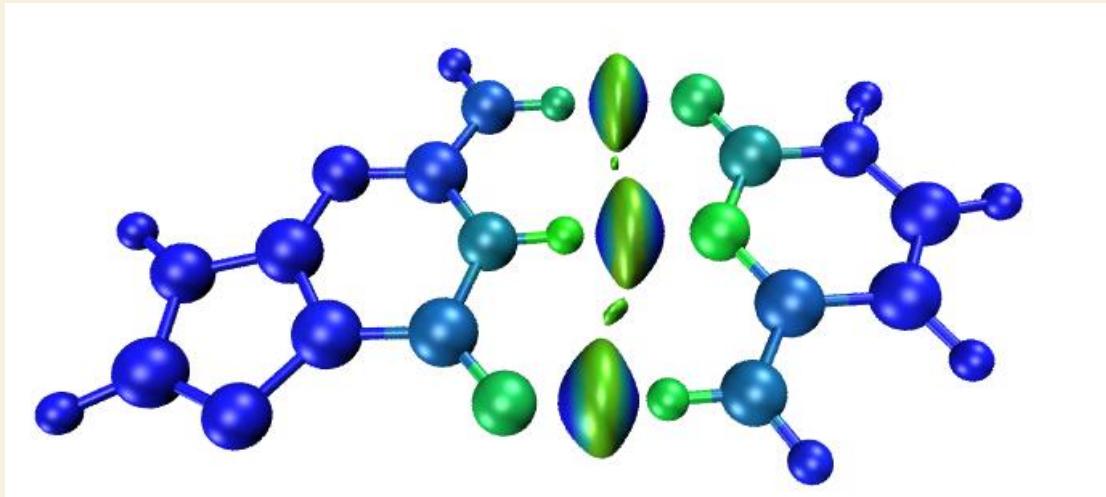


The largest three atom δg indices of fragment 1 are 25, 24 and 29, while that of fragment 2 are 25, 24 and 29, they are just the atoms closest to another fragment, undoubtedly they should have the most important contributions to the interfragment interactions. H13-O24, N6-H25 and O8-H29 have the largest atom pair δg indices, reflecting that they are the most crucial interactions for formation of the base pair.

(5) Coloring molecular structure by atom δg indices

Using VMD, it is also possible to map atom δg indices on molecular structure, so that relative importance of various atoms for interfragment interaction can be vividly exhibited. If needed, the δg^{inter} isosurfaces can also be shown together. Now we plot such a map.

First, plot color-filled δg^{inter} isosurfaces using the IGM_inter.vmd script as mentioned earlier. After that, we need to remove the default representation showing molecular structure, so we enter "Graphics"- "Represenation", choose the first term (its current style is CPK), click "Delete Rep" button. Then we drag the previously generated atm dg.pdb into VMD main window to load it. In this file, the B-factor field records atom δg indices; therefore, we should let VMD color the atoms according to their B-factors. We enter "Graphics"- "Representation" again, set "Drawing method" to CPK, and set "Coloring method" to Beta, then click "Trajectory" tab, set upper limit of color scale to 9.0 and press ENTER button, now the system in graphical window should look like below

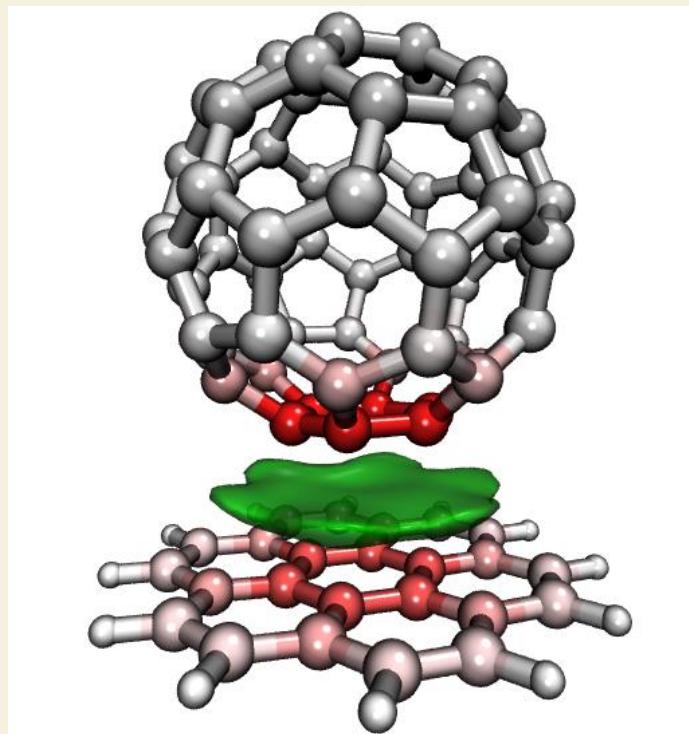


Since the color transition set by IGM_inter.vmd is Blue-Green-Red, the largest atom δg index in present case is 0.45 (as can be seen in atmddg.txt), while currently the range of color scale for mapping atom δg indices is set to 0.0~9.0, therefore in above map, the more green the atom, the larger the atom δg indices. The green atoms may be viewed as "hot atom" for interfragment interactions. Contribution to interfragment interactions due to the blue atoms can be safely ignored, since their δg indices are very close to zero.

This example ends here, through this example I think you have already recognized basic steps of IGM analysis. In next several examples I will illustrate more.

Example 2: C₆₀-coronene dimer

In this example, we will carry out IGM analysis for C₆₀-coronene dimer, and finally plot below map using VMD based on the data outputted by Multiwfn.



In above map, the major van der Waals interaction region (more specifically, the π - π stacking region) is exhibited as green isosurface, the more red-colored atoms contribute to the interaction

more. If you think this graph is pretty and want to reproduce it, just follow below steps.

The pdb file of the dimer optimized at PM6-D3 level using Gaussian is provided as *examples\c60_coronene.pdb*. Boot up Multiwfn and load it, then input below commands

```
20 // Visual study of weak interactions  
10 // IGM analysis  
2 // Define two fragments  
1-60 // C60 is fragment 1  
61-96 // Coronene is fragment 2  
2 // Medium quality grid  
3 // Output cube files in current folder  
6 // Evaluate atom and atom pair δg indices  
y // Export atmrdg.pdb in current folder
```

Boot up VMD, enter below commands into VMD console window:

```
color scale method BWR  
color Display Background white  
axes location Off  
display depthcue off  
display rendermode GLSL
```

Then drag atmrdg.pdb into VMD main window, enter "Graphics"- "Representation", set "Drawing Method" to "CPK", change bond radius from the default 0.3 to 0.8, set "Coloring Method" to "Beta", set "Material" to "EdgyShiny". Then go to "Trajectory" tab, set lower and upper limit of color scale to -5.0 and 5.0, respectively.

Next, we need to draw the δg^{inter} isosurface on the map. Drag dg_inter.cub into VMD main window to load it, then enter "Graphics"- "Representation", change the default style from "lines" to "Isosurface", set "Draw" to "Solid Surface", set "Show" to "Isosurface", then input 0.004 in the "Isovalue" box and press ENTER button. Change "Coloring Method" to "ColorID" and select "7 green". Change the "Material" to "EdgyGlass".

Finally, we render the map. Select "File"- "Render", choose "Tachyon (internal, in-memory rendering)" and click "Start Rendering" button, then you will obtain the graph shown at the beginning of this section. The resulting graphical file is .tga format, you can use such as IrfanView or photoshop to visualize it.

Note that in present example, color-filled effect is only applied to molecular structure, but not applied to isosurface. This is because in VMD, setting of color transition is shared by all representations, that means we cannot use Blue-White-Red color transition for molecular structure but use conventionally employed Blue-Green-Red color transition for isosurface. Considering that in present system there is only one kind of interaction, namely van der Waals interaction, and in conventional color-filled IGM map this region is basically colored by green, I decide directly assign green color for the whole isosurface, so that we are able to freely set color transition mode for coloring molecular structure.

Example 3: Oxazolidinone trimer

The IGM module of Multiwfn is extremely flexible, it can be applied to any number of fragments. In this example I use oxazolidinone trimer to show this point. The geometry was taken from *J. Chem. Theory Comput.*, **11**, 3065 (2015).

We first use δg^{inter} to reveal all interactions between the three monomers. Boot up Multiwfn and input

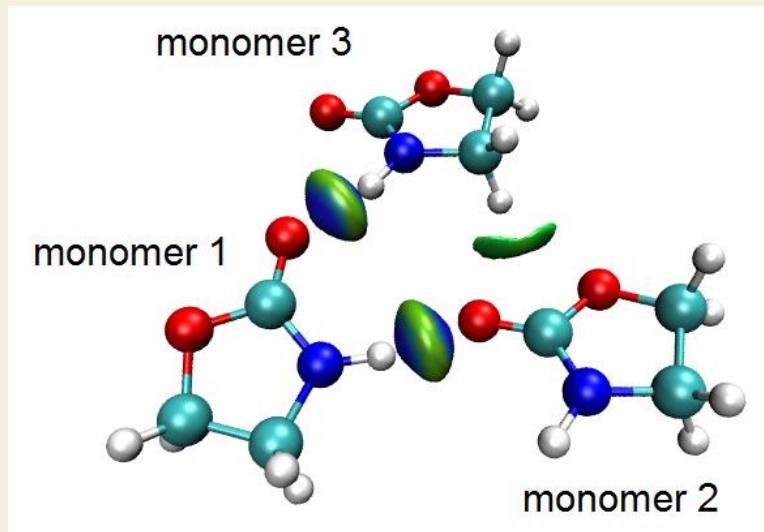
```
examples\oxazolidinone_trimer.xyz
```

```

20 // Visual study of weak interactions
10 // IGM analysis
3 // Define three fragments
1-11 // Fragment 1: Monomer 1
12-22 // Fragment 2: Monomer 2
23-33 // Fragment 3: Monomer 3
2 // Medium quality grid
3 // Output cube files in current folder

```

Then we use aforementioned method to plot color-filled δg^{inter} isosurface map via IGM_inter.vmd script, you will see below graph



From color of isosurfaces in the graph it is found that the 1-2 and 1-3 interactions correspond to typical hydrogen bonding, while 2-3 interaction is significantly weaker and thus more appropriate to be assigned as van der Waals interaction.

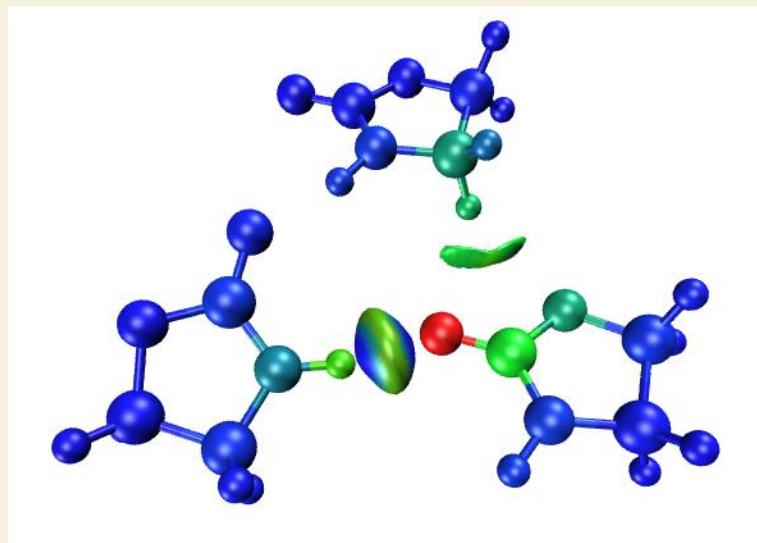
Assume that we only want to study interactions between 1-2 and 2-3, and meantime wish to screen the δg^{inter} isosurface corresponding to 1-3 interaction, how to do that? The answer is: Only define two fragments, making fragment 1 correspond to monomer 2, while making fragment 2 correspond to monomers 2 and 3. Now we do this, input below commands

```

0 // Return to last menu
10 // IGM analysis
2 // Define two fragments
12-22 // Fragment 1: Monomer 1
1-11,23-33 // Fragment 2: Monomers 2 and 3
2 // Medium quality grid
3 // Output cube files in current folder
6 // Evaluate atom and atom pair  $\delta g$  indices
y // Export atmrdg.pdb in current folder

```

Then plot δg^{inter} isosurface again using the newly generated sl2r.cub and dg_inter.cub via IGM_inter.vmd, and mean time color the structure according to atom δg indices based on atmrdg.pdb file. You will finally obtain below graph.



Now the isosurface corresponding to monomer 1-3 interaction is invisible. Since the color transition set by IGM_inter.vmd is Blue-Green-Red, and I did not manually adjust the automatically determined color range for mapping atom δg indices, therefore, in present graph, the site having largest atom δg index is rendered as red, it should be regarded as "hottest atom" for the interactions under study. The green or cyan atoms have modest magnitude of atom δg index, while contribution of blue atoms to the interaction is completely negligible.

Finally, let us only highlight interaction between monomer 1 and 2 while completely ignore monomer 3. Input below command

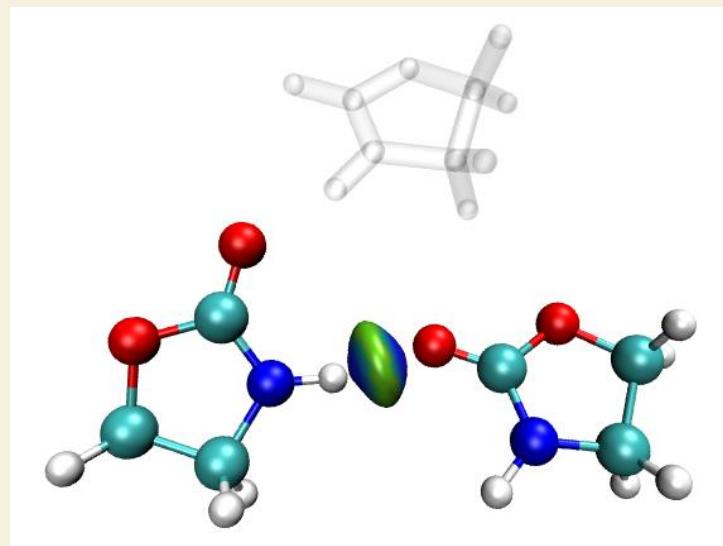
```

0 // Return to last menu
10 // IGM analysis
2 // Define two fragments
1-11 // Fragment 1: Monomer 1
12-22 // Fragment 2: Monomers 2
2 // Medium quality grid
3 // Output cube files in current folder

```

Move the resulting dg_inter.cub and sl2r.cub to VMD folder and use the IGM_inter.vmd script to draw corresponding color-filled isosurface map. It is better to make monomer 3 transparent, since currently it is uninteresting. So we enter "Graphics"- "Representation", click the existing representation with CPK style, input *fragment 0 1* in the "Selected Atoms" box and press ENTER button, now the monomer 3 is invisible. Next, click "Create Rep" button, input *fragment 2* in the "Selected Atoms" box and press ENTER button, then set the "Drawing method" as "Licorice" and change the "Bond Radius" to 0.2, then set "Material" to "Ghost". Now you should see below graph, in which only the interaction between monomer 1 and 2 is visible, while all interactions related to monomer 3 are ignored.

PS: The concept of "fragment" in VMD is different to the "fragment" in IGM analysis of Multiwfn. In VMD, when structure file is loaded into VMD, the bonding relationship is automatically determined, and then each unlinked fragment is assigned to a unique fragment index. The index starts from 0.



As can be seen in this example, the partition of fragments is highly arbitrary. The union set of all fragments is not necessarily equal to the whole system. When you intend to study intramolecular interaction, a whole molecule can also be divided into multiple fragments to reveal interesting interaction regions.

In this example only a simple system is taken as instance, however I think it is enough to substantially exhibit the extreme flexibility and powerfulness of IGM analysis, Multiwfn and VMD programs. The IGM method can also be easily applied to much more complicated systems; for example, in my blog article <http://sobereva.com/407> (in Chinese), I showed that IGM can clearly reveal the interaction between two monomers in a tetramer consisted of four large flexible molecules. Please play with the IGM analysis more!

It is worth to mention that δg value at bond critical points (BCP) is positively correlated to interaction strength (see Table 1 of IGM original paper), Multiwfn is also able to calculate it. First, load a file containing wavefunction information into Multiwfn, then use main function 2 to carry out topology analysis and locate BCPs, then using option 7 to examine properties of the BCPs, from screen you can directly read δg value. I do not explicitly present a corresponding analysis example here, please try this kind of analysis by yourself.

4.21 Energy decomposition analysis

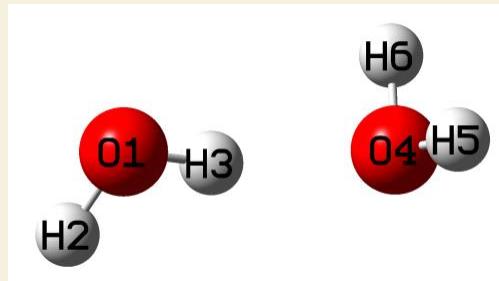
4.21.1 Examples of energy decomposition analysis based on forcefield (EDA-FF)

In this section I will illustrate how to perform energy decomposition analysis between specific fragments based on classical forcefield, this method will be referred to as EDA-FF. Please carefully read Section 3.24.1 first to gain basic knowledges about the underlying ideas and implementations of EDA-FF. If you have carefully read below examples, you should be able to easily apply this method onto various kinds of systems.

More in-depth discussions can be found from my blog article "Using Multiwfn to perform energy decomposition analysis based on forcefield" (in Chinese, <http://sobereva.com/442>).

Example 1: Water dimer

As first example, we perform the EDA-FF based on AMBER forcefield for a very simple system, water dimer, whose most stable geometry is shown below



The related files have been provided in "examples\EDA\EDA_FF\waterdimer" folder, as shown below:

- *dimer.mol*: The .mol file of water dimer containing its optimized geometry at B3LYP-D3(BJ)/6-311G** level, which is very reliable for optimizing molecular clusters. Note that in this file, the atomic sequence is O1 H2 H3 O4 H5 H6
- *water.fchk*: The .fchk file produced by optimization task of water monomer at B3LYP-D3(BJ)/6-311G** level
- *mollist.txt*: Molecular list file corresponding to the water dimer
- *water.txt*: Molecular type file of water monomer

As you can see, the content of *mollist.txt* is simply

```
water.txt 2
```

corresponding to the fact that the water dimer has two water molecules, which are described by the *water.txt* in current folder.

The content of *water.txt* is

```
OW -0.737121
HW 0.368560
HW 0.368560
```

indicating that the oxygen and hydrogens in the water have atom type of OW and HW, respectively. The second column are atomic charges evaluated by Merz-Kollman (MK) method.

Here I describe how the *water.txt* was constructed. The atom types can be manually assigned according to practical chemical environment of the atoms and definition of atom types in the forcefield original paper, but this process is troublesome if there are lots of atoms in a molecule. Therefore, here I show how to use the popular GaussView to automatically assign the atom types. Load the *water.fchk* into GaussView, click the icon with bold A letter to enter "atom list editor", then click the icon with bold orange M letter to show atom types, then click the title of "AMBER Type" column twice, the current status of the window should be

The screenshot shows a software interface titled "G1:13:VI - Atom List Editor". The menu bar includes "File", "Edit", "View", "Rows", "Columns", "Help", and a toolbar with various icons. A search bar at the top right says "Go To [3]". Below is a table with the following data:

Row	Highlight	Display	Tag	Symbol	Any AN	UFF Type	Dreiding Type	AMBER Type*	MM Charge	AMBER Fragment
1	●	Show	1	O	No	O_3	O_3	OW	-0.8340000	HOH
2	●	Show	2	H	No	H_	H_	HW	0.4170000	HOH
3	●	Show	3	H	No	H_	H_	HW	0.4170000	HOH
4	Add	Show	-	?	No	?	?	?	?	

Active Sublist Filters: None

Then click "File" - "Export Data", save the file as *water.txt*. Next, via column mode of advanced text editor such as Ultraedit, delete all columns except the "AMBER Type" column, and then delete the first row in the file. Now, only atom types of all atom are presented in the *water.txt*. After that, calculate MK (or CHELPG) charges based on the *water.fchk* using subfunction 13 (or 12) of main function 7 (you can consult the example in Section 4.7.1), then copy the outputted charges from screen (or from the exported .chg file) as the second column of the *water.txt*. At this point, preparation of *water.txt* is finished.

Now, we start to perform the EDA-FF analysis. Copy the *water.txt* to current folder, then boot up Multiwfn and input

```
dimer.mol // The file containing dimer structure information (you can also use other formats containing the geometry information as input file, such as the .fch file produced during optimization task of the dimer)
```

```
21 // Energy decomposition analysis
1 // Energy decomposition analysis based on forcefield
3 // Load atom types and atomic charges
mollist.txt // The actual path of the molecular list file. At this point, the program read atom types and charges from the water.txt and assign them to the two water molecules in the current system
2 // Define fragments
2 // Two fragments will be defined
1-3 // The atomic indices of the fragment 1
4-6 // The atomic indices of the fragment 2
```

If you want to check if atom types and charges of all atoms in current system have been set up properly, you can choose option 4, the output is

```
*** Fragment 1:
Atom: 1(O ) Charge: -0.737121 Type: OW
Atom: 2(H ) Charge: 0.368560 Type: HW
Atom: 3(H ) Charge: 0.368560 Type: HW
*** Fragment 2:
Atom: 4(O ) Charge: -0.737121 Type: OW
Atom: 5(H ) Charge: 0.368560 Type: HW
Atom: 6(H ) Charge: 0.368560 Type: HW
```

It is clear that the atom types and charges are all correctly assigned.

Now select option 1 to start the EDA-FF analysis, the result shows up immediately on screen:
Contribution of each atom in defined fragments to overall interfragment interac

tion energies:

Atom	1(O)	Elec:	12.53	Rep:	3.85	Disp:	-2.21	Total:	14.17
Atom	2(H)	Elec:	-6.24	Rep:	0.00	Disp:	0.00	Total:	-6.24
Atom	3(H)	Elec:	-16.87	Rep:	0.00	Disp:	0.00	Total:	-16.87
Atom	4(O)	Elec:	-23.52	Rep:	3.85	Disp:	-2.21	Total:	-21.88
Atom	5(H)	Elec:	6.47	Rep:	0.00	Disp:	0.00	Total:	6.47
Atom	6(H)	Elec:	6.47	Rep:	0.00	Disp:	0.00	Total:	6.47

Interaction energy components between all fragments:

		Electrostatic	Repulsive	Dispersion	Total
Frag	1 -- Frag 2:	-21.15	7.71	-4.43	-17.87

The units in the output are all kJ/mol. The above information shows that the total interaction energy between the two water molecules is -17.87 kJ/mol, which is close to the result -20.58 kJ/mol obtained by highly accurate level CCSD(T)/CBS (see original paper of the S66 weak interaction test set, *J. Chem. Theory Comput.*, **7**, 2427 (2011)). Although the given result has some error, at least it is adequate for qualitative discussion purpose. The above data also indicates that electrostatic interaction (-21.15 kJ/mol) has a decisive contribution to the binding energy between the two waters, obviously the main essence of general hydrogen bond is dominated by electrostatic interaction. Dispersion interaction also contributes to the binding, but the magnitude is relatively minor. The exchange-repulsion effect (7.71 kJ/mol), to some extent, offsets the attractive interaction due to electrostatic and dispersion effects.

In the original text of the S66 test set, the ratio of the dispersion interaction energy to the electrostatic interaction energy of water dimer given by the very ideal DFT-SAPT method is 0.29, which is qualitatively consistent with the value yielded by EDA-FF ($4.43/21.15=0.21$). Therefore, with the very simple water dimer as instance, it can be seen that as long as the choice of forcefield and atomic charges are suitable, the result of EDA-FF is generally reliable. For some systems, the total interaction energy calculated by forcefield is not quite close to that evaluated by reliable quantum chemistry method, but even so, in general the ratio between various physical components provided by the EDA-FF is still meaningful. In my view of point, it is not a bad idea to approximately estimate electrostatic interaction energy (ΔE^{ele}) via multiplying the total interaction energy (ΔE^{tot}) obtained using quantum chemistry method by the ratio of the ΔE^{ele} and ΔE^{tot} that evaluated by proper forcefield.

The above output also shows contribution of each atom to the total interaction between all the defined fragments, so that you can easily recognize which atoms have a critical impact on the interfragment interaction. The sum of all atomic contributions is equal to the total interaction energy (if the system only has two atoms *A* and *B*, and each one is defined as a fragment, then the contribution of atom *A* will be half of the interaction energy between *A-B*). From the data given above, it can be seen that influence of each atom is not negligible. After all, the distance between the atoms in the system is not far. The most important contribution to the attraction is the electrostatic interaction of the O4 atom (-23.52 kJ/mol), this result is easy to understand since O4 is the acceptor atom of H-bond. The H3, which directly acts with O4 to form the H-bond, also contributes greatly to the binding (-16.87 kJ/mol) due to significant electrostatic effect. The data shows that only oxygen atoms have nonvanishing repulsion and dispersion terms, this is because the parameters of van der Waals potential of atom type HW is zero, hence HW atoms only behave

as point charges to exhibit electrostatic effect.

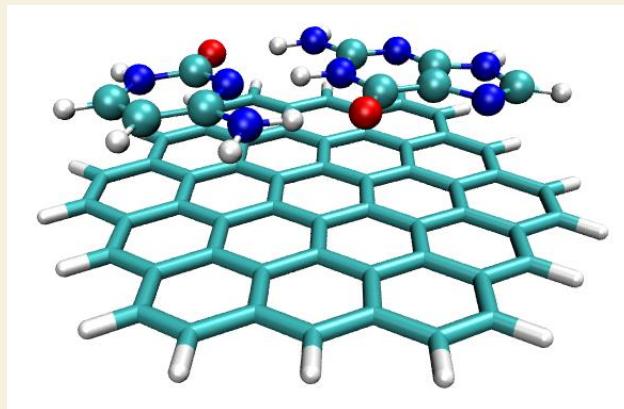
If you choose option -3 once to switch its status from the default "No" to "Yes", then during EDA-FF analysis via option 1, the program also outputs distance (Å), interaction energy (kJ/mol) and its components of each atom pair to *interatm.txt* in current folder. The file content of present example is

***** Between fragment 1 and fragment 2:						
Atom_i	Atom_j	Dist(Ang)	Electrostatic	Repulsive	Dispersion	Total
1	4:	2.873	262.78	7.71	-4.43	266.05
1	5:	3.176	-118.86	0.00	0.00	-118.86
1	6:	3.176	-118.86	0.00	0.00	-118.86
2	4:	3.346	-112.79	0.00	0.00	-112.79
2	5:	3.762	50.16	0.00	0.00	50.16
2	6:	3.762	50.16	0.00	0.00	50.16
3	4:	1.916	-197.02	0.00	0.00	-197.02
3	5:	2.312	81.64	0.00	0.00	81.64
3	6:	2.312	81.64	0.00	0.00	81.64

From the above data, we can find that the interaction energy between each pair of atoms is very large, which mainly comes from electrostatic interaction. For example, since the charges of the two oxygen atoms O1 and O4 are large and have the same sign, the electrostatic mutual exclusion energy is as high as 262.78 kJ/mol. The binding energy between the fragments appear to be orders of magnitude far less than the above values, this is because when interaction energy between the fragments is calculated, the electrostatic interactions of the atom pairs are largely positively and negatively offset.

Example 2: Circumcoronene-Cytosine-Guanine trimer

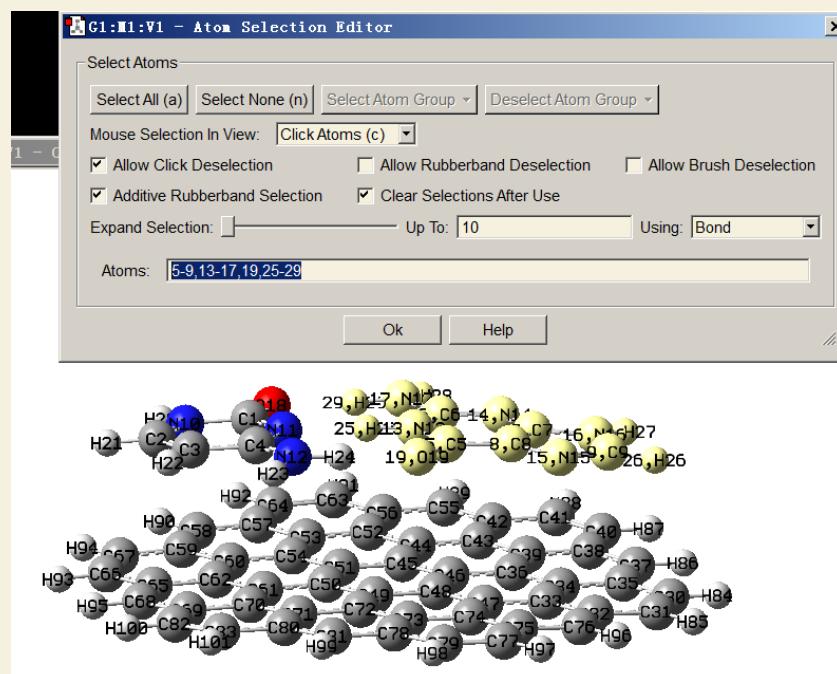
In the L7 weak interaction test set given in *J. Chem. Theory Comput.*, **9**, 3364 (2013), a system C3GC is a trimer consisted of circumcoronene (hereinafter abbreviated as C3), guanine (G) and cytosine (C). The geometry has been optimized by the authors at TPSS-D/TZVP level, as shown below. The GC base pair has formed triple H-bonds, and it is physically adsorbed on the C3 via π - π stacking interaction.



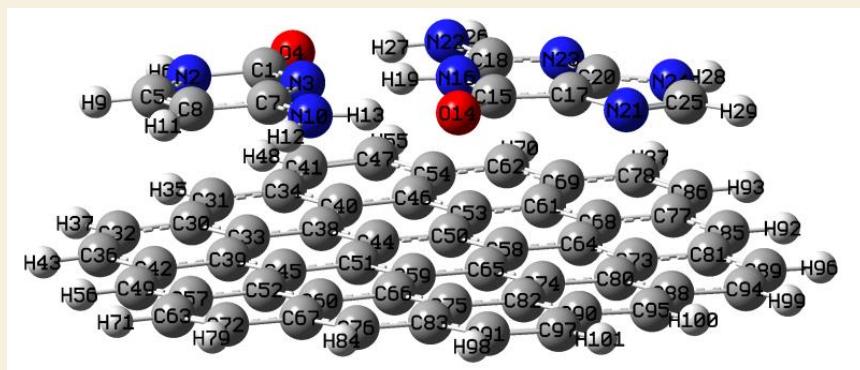
In this section, we will perform EDA-FF analysis on this system based on the AMBER force field. The relevant files are provided in the "examples\EDA\EDA-FF\C3GC" directory.

Notice that Multiwfn can perform EDA-FF only when atomic indices in any molecule type are contiguous. Otherwise, the atomic charges and types cannot be set for each atom in the system

through the molecular list file and molecule type files. The structure file given in the supplementary material of the L7 test set is *C3GC.xyz*. This file cannot be directly used because the atomic indices in each monomer is not contiguous. One of the simplest way to judge if the atomic indices are contiguous is as follows: First load *C3GC.xyz* into Multiwfn, use subfunction 2 in main function 100 to convert it to *C3GC.pdb* (we do the conversion because GaussView does not support .xyz format), then load this pdb file into GaussView, right click on arbitrary atom in a arbitrary molecule (*e.g.* atom C5), choose "Select Fragments of Atom C5" (this option is available since GaussView 6). At this point, all atoms in this molecule are selected as yellow color, then click "Tools" - "Atom Selection". As can be seen in below screenshot, the indices displayed in the text box are 5-9, 13-17, 19, 25-29, clearly the atom indices are not contiguous and should be rectified.



The easiest way of making atom indices contiguous in each molecule is entering "Atom list editor" of GaussView, then select "Edit" - "Reorder" - "All Atoms (Except the First) by Bonding", after that the atom indices are reordered according to connectivity, and you will see the atom indices in each monomer have become contiguous, as shown below. Now save this structure to *C3GC.pdb* to replace the old one.



Then we copy each monomer from the trimer to individual GaussView window, save them to respective .gjf files, change the keyword to "B3LYP/6-311G**" and use Gaussian to run them, then calculate MK charges by Multiwfn based on the resulting .fch files. Also we make use of GaussView

to determine atom types for each monomer. Finally, combine the atom types and charges as single file for each monomer, then we have the *C.txt*, *G.txt* and *C3.txt*, which have already been provided in "examples\EDA\EDA-FF\C3GC" directory.

Finally, create a molecular list file *mollist.txt* (other name is also acceptable), the content is actual paths of *C.txt*, *G.txt* and *C3.txt* as well as the number of corresponding molecule, notice that the order of the file paths must be exactly in line with the molecule order in the geometry provided in *G3GC.pdb*. Clearly, the content of *mollist.txt* should be (all the molecule type files are assumed to be placed in C:\)

```
C:\C.txt 1
C:\G.txt 1
C:\C3.txt 1
```

All preparation works have completed, now we start the EDA-FF analysis. Boot up Multiwfn and input

```
C3GC.pdb
21 // Energy decomposition analysis
1 // EDA-FF
3 // Load atom types and charges
mollist.txt // Input actual path of mollist.txt
2 // Define fragments
3 // Three fragments will be defined
1-13 // Atom indices in fragment 1, namely cytosine (C)
14-29 // Atom indices in fragment 2, namely guanine (G)
30-101 // Atom indices in fragment 3, namely C3
```

Select option 1 to carry out the EDA-FF calculation, the results are as follows (atomic contribution part is ignored)

		Electrostatic	Repulsion	Dispersion	Total
Frag	1 -- Frag 2:	-120.98	60.26	-45.54	-106.27
Frag	1 -- Frag 3:	1.88	44.86	-94.69	-47.95
Frag	2 -- Frag 3:	0.71	62.08	-132.62	-69.84

The data shows that the G-C binding is very strong, reached as high as -106.27 kJ/mol, mainly because the electrostatic component is very large (-120.98 kJ/mol), which is the consequence of the formation of the three pairs of H-bonds between the G and C. The total interaction energy between C3 and G (-47.95 kJ/mol) as well as between C3 and C (-69.84 kJ/mol) are not small, mainly due to the significant π - π stacking between them. Since the nature of π - π stacking is purely dispersion effect, it can be seen that the dispersion interactions of C3-C and C3-G are very strong (-94.69 and -132.62 kJ/mol, respectively), which are much higher than that between G-C (-45.54 kJ/mol). Since C3 is essentially a finite graphene sheet, its interaction region with G and C is obviously non-polar (*i.e.* the atomic charges are very small), so the electrostatic component in C3-C and C3-G interactions is negligible (merely 0.71 and 1.88 kJ/mol, respectively).

I also calculated binding energies between G-C, C3-G and C3-C using dimer models at B3LYP-D3(BJ)/6-311+G** level, which is very robust for evaluating weak interactions, the results are

```
G-C (Frag 1 - Frag 2): -143.97 kJ/mol
C3-C (Frag 1 - Frag 3): -56.69 kJ/mol
```

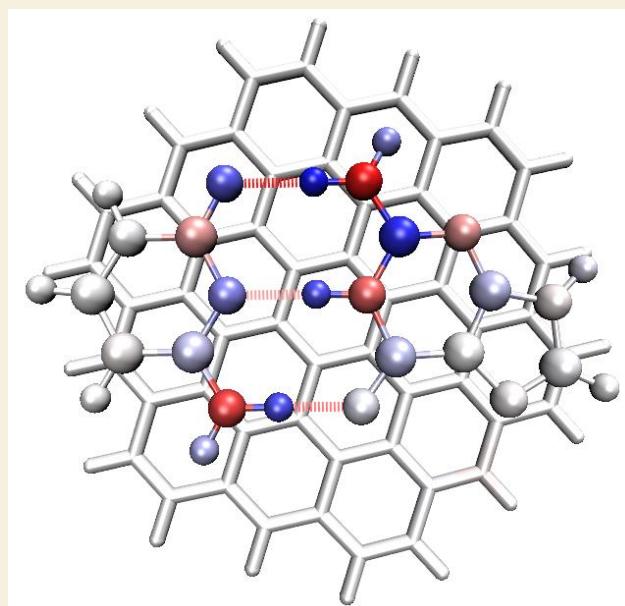
C3-G (Frag 2 - Frag 3): -76.27 kJ/mol

For C3-C and C3-G, it can be seen that the results calculated by AMBER forcefield and that by quantum chemistry method are very close, but the quantitative difference for G-C is quite conspicuous. This observation manifests the limited quantitative accuracy of forcefield when it is applied to the weak interactions with great strength. However, in this case, if we only focus on the ratio between various physical components, the result of EDA-FF is still useful and will not cause evident misleading conclusion.

By making use of .pqr file format, atomic properties can be easily visualized by coloring atoms via different colors in VMD visualization program, this strategy is detailedly introduced in Section 4.A.10. Obviously, if atomic contributions to interaction energy between fragments are stored into the .pqr files, then the importance and role of each atom will be able to be vividly shown in a graph. Similar idea is also employed in the IGM analysis, as illustrated in Section 4.20.10.

Now we select "-4 Toggle if outputting atom contributions to .pqr files" option in the EDA-FF interface to switch its status to "Yes", then select option 1 to carry out EDA-FF analysis. After the calculation, *atmint_tot.pqr*, *atmint_ele.pqr*, *atmint_rep.pqr* and *atmint_disp.pqr* appear in the current directory (they have been provided in "examples\EDA\EDA-FF\C3GC\pqr"). The data of the atomic charge column in the four .pqr files correspond to the contribution of each atom to the total/electrostatic/repulsion/dispersion interaction energy between the fragments, respectively, the values are identical to those printed on screen. For example, the value of the atomic charge column of the C1 atom in fragment 1 of the *atmint_disp.pqr* corresponds to half of the dispersion interaction energy between C1 atom and all atoms of fragments 2 and 3.

We load the *atmint_tot.pqr* into VMD program, enter "Graphics" - "Representation", coloring the atoms according to the "Charge" property, changing the lower and upper limits of the color scale from the default ones to -50 and 50, respectively. Setting drawing method of the G and C parts as CPK mode (using *fragment 0 1* in the "Selected Atoms" box to choose them), and making C3 part displayed in Licorice style (using *fragment 2* to choose it). We also set the color scale method to BWR (Blue-White-Red). Finally, the graph will look like below (if you are not familiar with VMD and do not know how to realize these settings in VMD, please consult Section 4.A.10):



Since the current color scale used is Blue-White-Red, therefore the more blue the atom color in the figure, the more negative the atomic contribution to the total binding energy between the trimer (*i.e.* the more significant the attractive effect); while the more red the atom, the stronger the repulsive effect it acts. Relatively white atoms only play trivial role on the trimer binding. As can be seen from this figure, the colors of each H-bond acceptor atom and the hydrogen atom directly acting with it are obviously blue, thus they contribute greatly to the stability of the G-C binding. The color of all H-bond donor atoms is red, indicating that their existences are not conducive to the binding, this is because the H-bond donor and acceptor atoms have large magnitude and same sign of atomic charges, therefore there is a significant electrostatic mutual exclusion between them. From above map, it can also be seen that all atoms of C3 as well as the atoms in the G and C that far away from the hydrogen bonding region only have very light or purely white color, this observation does not imply that they have nearly vanished contributions to the trimer binding, but indicates that their contributions are relatively weak and thus difficult to be revealed under current color scale setting.

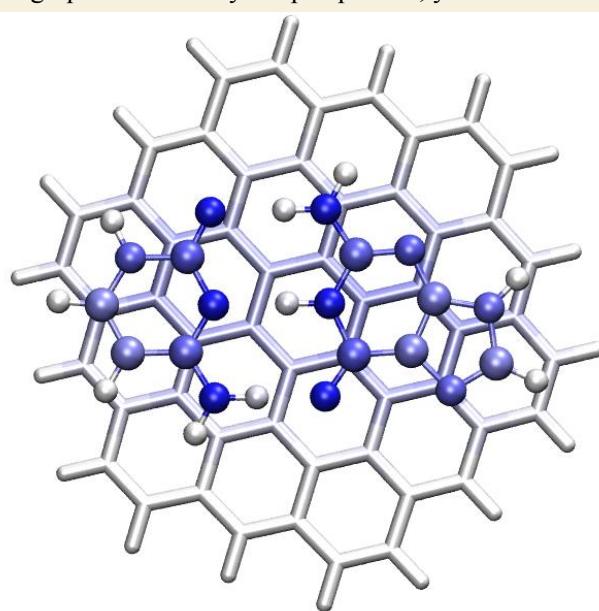
Assume that we want to vividly exhibit the dispersion interaction between C3 and GC base pair, then we input below commands in the EDA-FF interface

```
2 // Redefine fragments
2 // Two fragments will be defined
1-29 // Fragment 1, the GC base pair
30-101 // Fragment 2, the C3 part
1 // Start the EDA-FF calculation
```

Then below information is shown on screen, the data equals to the sum of C3-C and C3-G interaction energies

		Electrostatic	Repulsion	Dispersion	Total
Frag 1 -- Frag 2:		2.59	106.94	-227.31	-117.79

In the meantime, four new .pqr files are generated in current folder (they have been provided in "examples\EDA\EDA-FF\C3GC\pqr2"). Load the *atmint_disp.pqr* among them into VMD, coloring the atoms according to the way described above but using color scale of -10 to 10, then select "Display" - "Orthographic" to modify the perspective, you will see



In the above figure, the more blue the atomic color, the greater it contributes to the dispersion interaction between C3 and GC. It can be seen from the figure that each heavy atom in the GC part contributes nearly equally to the dispersion interaction, mainly because they have almost the same vertical distance to the C3 plane and the number of electrons carried by these atoms are not very different. The hydrogen atoms in the GC pair contribute very little to the C3-GC dispersion interaction, this is because the hydrogen atoms only have very few number of electrons. On the C3 part, the color of the carbons that directly contact with the GC pair is light blue, indicating their notable contributions to the dispersion interaction. The color of the C3 atoms that far away from the GC pair is white, reflecting that their influences on dispersion interaction are negligible (recall the fact that dispersion attraction attenuates sharply with distance, it has $1/r^6$ asymptotic behavior).

Note: The heavy atoms in the GC pair in the above graph are very blue, while the atoms in the equivalent position of C3 are not so blue, the reason is that: Because there are many atoms in C3, each heavy atom in the GC pair can form dispersion interaction with a large range of C3 atoms, thus the sum of the terms is large. Since the number of atoms in the GC pair is small, each atom of C3 can only interact with relatively few number of atoms in the GC pair, so the sum of terms is not large. If you want to make atomic color of the C3 part more prominent, you can set the color scale range of the representation corresponding to the C3 part to a smaller value than the -10~10 we previously used; for example, changing to -6 to 6 will yield satisfactory graph.

Some readers may have thought that it would be great if the binding energy of each of the three H-bonds between the G-C could be independently determined. There is no unique way to achieve this goal, since this is equivalent to dividing the system into parts and must not be free of artifacts. An seemingly easy way to realize this purpose is to directly define the donor and acceptor parts of a H-bond as two fragments. For example, let us examine the H-bond of N10-H13...O14, we input.

```
2 // Redefine fragments
2 // Two fragments will be defined
10,13 // Atomic indices of donor part of N10-H13...O14
14 // Atomic index of acceptor part of N10-H13...O14
1 // Perform EDA-FF analysis
```

The result is

		Electrostatic	Repulsion	Dispersion	Total
Frag 1 -- Frag 2:		71.27	21.78	-9.40	83.65

Clearly the result is unreasonable, since the total binding energy was predicted to be a positive value! The underlying reason is that, the electrostatic interaction is a kind of long-range effect ($1/r$ asymptotic behavior), therefore consideration of other atoms should not be simply ignored. I also attempted to employ other ways to evaluate the N10-H13...O14 H-bond binding energy, although some of them give seemingly acceptable result (for example, summing up atomic contribution of N10, H12, H13 and O14 to total G-C binding energy), unfortunately the relative strength of the three H-bonds cannot be faithfully explained. In my opinion, deriving individual H-bond interaction energy is impossible for this system based on EDA-FF, the reason is that the three H-bonds are too close together and thus the coupling is very strong, the polarization effect is obvious, and meantime resonance-assisted effect is involved in these H-bonds, these factors make the total interaction energy of the three H-bonds very difficult to be reasonably decomposed. However, if there are several H-bonds and the sites are far away from each other, it should be possible to individually evaluate the strength of each H-bond by estimation of interaction energy between the atoms in the corresponding local region.

It is worth to note that a possibly viable way aside from SAPT to evaluate dispersion interaction

energy is employing DFT-D3 dispersion correction using the zero-damping parameters fitted for exchange-correlation functionals that completely failed to represent dispersion interaction. This strategy has been utilized in the energy decomposition analysis example shown in Section 4.100.8. The DFT-D3 dispersion correction for interfragment interaction energy using zero-damping BLYP functional parameter is shown below

```
C -- G: -26.65 kJ/mol
C -- C3: -87.78 kJ/mol
G -- C3: -115.10 kJ/mol
```

It can be seen that the three values are close to the corresponding dispersion interaction energies evaluated by AMBER forcefield (-45.54, -94.69, -132.62, respectively), rendering that using either AMBER forcefield or DFT-D3 to estimate dispersion interaction energy is a reasonable approach.

Using UFF to conduct the EDA-FF analysis is often unsatisfactory. If we do this for present system, the result is

		Electrostatic	Repulsion	Dispersion	Total
Frag	1 -- Frag	-120.98	848.28	-80.36	646.94
Frag	1 -- Frag	3:	1.88	52.65	-107.41
Frag	2 -- Frag	3:	0.71	68.89	-143.58
					-73.99

It can be seen that the interaction predicted by UFF for C3-C and C3-G is normal, and the result magnitude is close to that calculated by AMBER, but the interaction energy of G-C is extremely positive, evidently this is completely unreasonable. From the data it is easy to find that the reason is that the exchange-repulsion component is seriously overestimated. This is not an individual phenomenon, but a common phenomenon of UFF, this is why I generally do not recommend performing EDA-FF based on UFF (although optimizing the geometry using UFF prior to the EDA-FF could alleviate this problem).

As mentioned in Section 3.24.1, for very large systems, using the very cheap EEM charges (with parameters fitted for CHELPG charges at B3LYP/6-31G* level) instead of the rigorously derived ESP fitting charges to perform the EDA-FF analysis may be a viable choice; however, my test showed that this treatment can cause remarkable error in electrostatic interaction for present system. The electrostatic interaction energy between G and C evaluated based on EEM charges is merely -74.26 kJ/mol, which is much lower than that evaluated based on MK charges (-120.98 kJ/mol). Therefore, whenever possible, using ESP fitting charges (MK or CHELPG) in EDA-FF analysis is strongly recommended, this is crucial for yielding electrostatic interaction energy with satisfactory accuracy.

4.100 Other functions (Part 1)

4.100.4 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. Boot up Multiwfn and input:

```
examples\COCl2.wfn // HF/6-31G* wavefunction
100 // Other functions (Part 1)
4 // Integrate a function in whole space
6 // Hamiltonian kinetic density K(r)
```

The result is 1031.1092, which is very close to the Gaussian outputted value 1031.1107.

Because I do not 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 still easily make these functions available by hacking source code, please consult Appendix 2. First, search “function userfunc” in function.f90, change the default content, namely userfunc=1.0D0, 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 -2839.1668, the value outputted by Gaussian is -2839.1629, evidently they are rather close. If you want to obtain nuclear attraction potential energy contributed from a specific orbital, use subfunction 26 of main function 6 to set occupation number of other orbitals to zero, and then do the integration as before.

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 recompile Multiwfn and redo the integration, you will get 444.6523, which is in excellent agreement with the Gaussian outputted value 444.6524 (behind “Electronic spatial extent (au): <R**2>”).

Tip: You may have already noticed that the two lines of the codes colored by blue above have presented in “userfunc” routine, and they correspond to iuserfunc==12 and iuserfunc==3 respectively. So actually you needn’t to modify and recompile the source code of Multiwfn, by simply changing “iuserfunc” parameter in settings.ini from the default value 0 to 12 and 3 respectively, the user defined real space function will be equivalent to the two functions. For more about the built-in real space functions in “userfunc” routine, see Section 2.7.

4.100.8 Perform simple energy decomposition by using combined fragment wavefunctions

The principle of the simple energy decomposition has been introduced in Section 3.100.8, please read it first. Here we try to use this method to analyze components of interaction energy between NH₃ and BH₃ in the adduct NH₃BH₃ at HF/6-31G* level. The Gaussian input and output files involved in this section can be found in "example\EDA\EDA_gau" folder. The Gaussian used here is Gaussian 09 E.01.

(1) Perform structure optimization task for NH₃BH₃, and then create a new input file with the optimized coordinate, named NH3BH3.gjf.

(2) Duplicate NH3BH3.gjf as NH3.gjf and BH3.gjf, then remove BH3 fragment in NH3.gjf, and remove NH3 fragment in BH3.gjf. Add "pop=full nosymm" keywords in both NH3.gjf and BH3.gjf.

(3) Run the input file of the two fragments by Gaussian respectively to generate NH3.out and BH3.out.

(4) Boot up Multiwfn, input following commands

```
NH3.out //Fragment 1
100 // Other functions (Part 1)
8 // Generate Gaussian input file with initial guess from fragment wavefunctions
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 information can be found

```
E(NH3) = -56.184295 a.u.
E(BH3) = -26.368199 a.u.
E(NH3BH3) = E_SCF_last = -82.611817 a.u.
E_SCF_1st = -82.535276 a.u.
```

PS: If you don't know where to find E_SCF_1st, search below information from new.out:

```
Cycle 1 Pass 1 IDiag 1:
```

```
E= -82.5352762791642
```

```
DIIS: error= 5.16D-02 at cycle 1 NSaved= 1.
```

The 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 can calculate each energy term:

$$\begin{aligned}\Delta E_{\text{tot}} &= E(\text{NH}_3\text{BH}_3) - E(\text{NH}_3) - E(\text{BH}_3) = -0.059323 \text{ a.u.} = -155.75 \text{ KJ/mol} \\ \Delta E_{\text{orb}} &= E_{\text{SCF_last}} - E_{\text{SCF_1st}} = -0.076541 \text{ a.u.} = -200.96 \text{ KJ/mol} \\ \Delta E_{\text{steric}} &= \Delta E_{\text{els}} + \Delta E_{\text{ex}} = \Delta E_{\text{tot}} - \Delta E_{\text{orb}} = 0.017218 \text{ a.u.} = 45.20 \text{ KJ/mol}\end{aligned}$$

namely the total interaction energy between NH₃ and BH₃ is -155.75 KJ/mol, the orbital interaction energy -200.96 KJ/mol stabilized the adduct, while the steric energy, which is the sum of electrostatic and exchange repulsion energy, destabilized the adduct by 45.20 KJ/mol.

In present system, compared to other energy components the dispersion interaction can be safely neglected, so we did not discuss it. However if the system you studied is the complex bounded by weak interaction, you must evaluate the dispersion component in total interaction energy, see the discussion in Section 3.100.8 on how to do this.

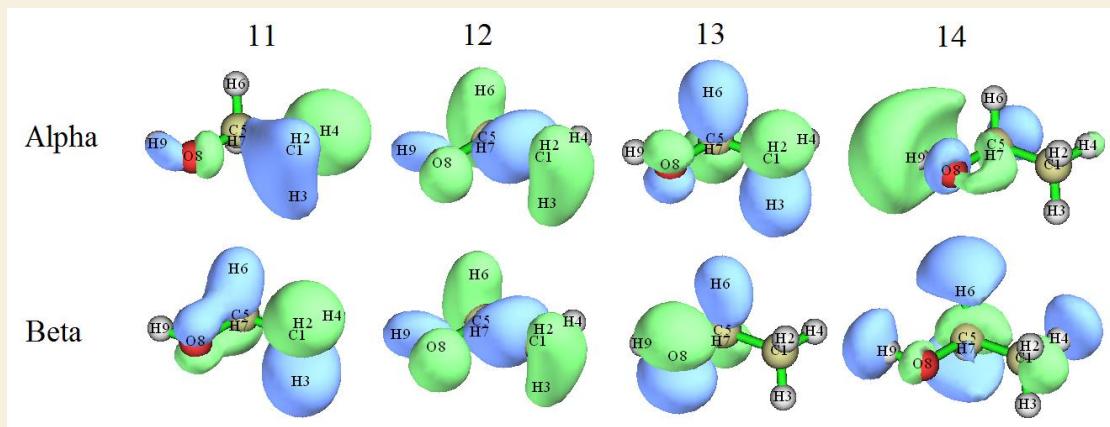
4.100.12 Perform biorthogonalization analysis for orbitals of unrestricted open-shell wavefunction

Chinese version of this example and related introduction is "Principle and application of biorthogonalization method for unrestricted open-shell wavefunctions" <http://sobereva.com/448>.

The biorthogonalization algorithm between alpha and beta orbitals has been introduced in Section 3.100.12, please read it first, this transformation makes interpretation of orbitals generated by UHF or UKS calculation much easier, since only one set of orbitals will then need to be examined. In this section I will illustrate the value of this analysis, *examples\ethanol_triplet.fch* is taken as

example.

This system has 14 occupied alpha and 12 occupied beta orbitals, first let us look at some of them:



It can be seen that only alpha orbital 12 pairs well with beta orbital 12, while other alpha and beta orbitals with the same index do not like with each other. Clearly, it is troublesome when we discuss orbital characteristics of this wavefunction, because we must simultaneously inspect two sets of orbitals.

Now we carry out the biorthogonalization for this wavefunction. Boot up Multiwfn and input *examples\ethanol_triplet.fch*

```
100 // Other function (Part 1)
12 // Perform biorthogonalization between alpha and beta orbitals
2 // Do biorthogonalization for all orbitals
```

The biorthogonalization for this wavefunction consists of three successive steps. For example, the outputted information of the first step is

```
Doing biorthogonalization for alpha 1 to 14, Beta 1 to 12
singular values of orbital overlap matrix:
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 0.9999
0.9999 0.9998 0.9995 0.9992
```

As shown, this step performs biorthogonalization between alpha orbitals 1~14 and beta orbitals 1~12. The overlap integrals between the resulting alpha and beta orbitals with the same index are all very close to 1.0, showing that the first 12 alpha orbitals have almost perfectly paired with the first 12 beta orbitals.

Once all the three steps have been finished, we input *n* to skip the step of generation of orbital energies, then *biortho.txt* and *biortho.fch* are generated in current folder.

The content of *biortho.txt* is shown below:

```
S = singular value, E = Energy (in eV), O= Occupancy, A=Alpha, B=Beta
```

```
orb: 1 S= 1.0000 O(A)= 1.0 O(B)= 1.0
...[ignored]
orb: 11 S= 0.9995 O(A)= 1.0 O(B)= 1.0
orb: 12 S= 0.9992 O(A)= 1.0 O(B)= 1.0
-----
orb: 13 S= 1.0000 O(A)= 1.0 O(B)= 0.0
```

```

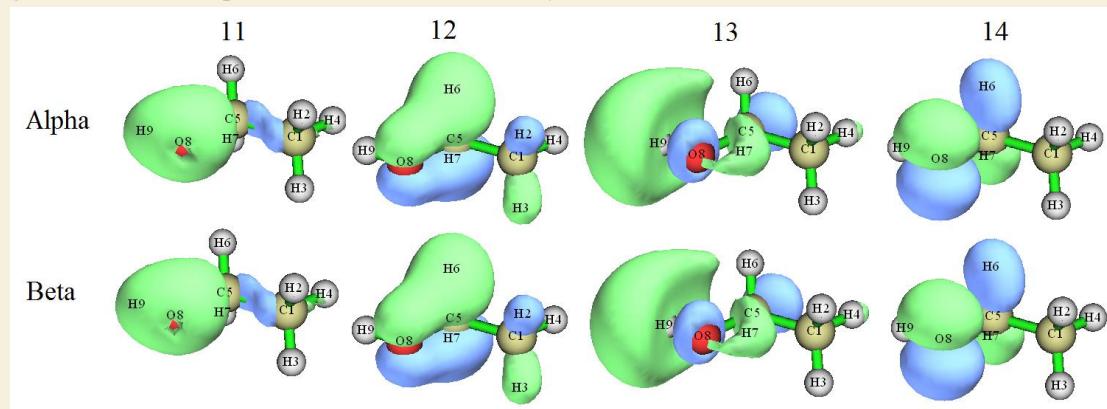
orb: 14 S= 1.0000 o(A)= 1.0 o(B)= 0.0
-----
orb: 15 S= 1.0000 o(A)= 0.0 o(B)= 0.0
orb: 16 S= 1.0000 o(A)= 0.0 o(B)= 0.0
...[ignored]

```

The information is very easy to understand. Since singular value (S) of all orbitals shown above is very close to 1.0, therefore the alpha orbitals match well with the beta orbitals having the same index.

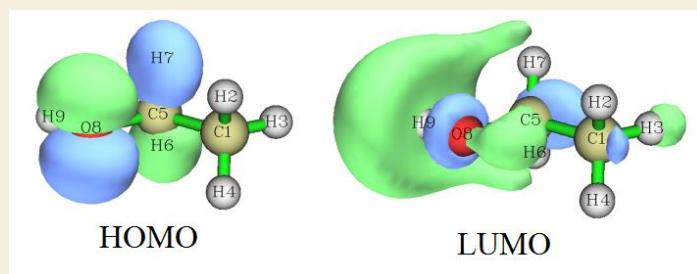
The exported *biortho.fch* contains wavefunction of the biorthogonalized orbitals, the “orbital energies” information of these orbitals now correspond to the singular values. Next, if you input *y*, this file will be immediately loaded, and then the orbitals in memory will correspond to the biorthogonalized orbitals.

Here we input *y* to load the *biortho.fch*, then enter main function 0 to examine the newly generated 11~14 alpha and 11~14 beta orbitals, you will see



From the isosurface map it can be seen that the current alpha and beta orbitals match with each other perfectly. The concept "singly occupied molecular orbitals" (SOMO) was originally defined for restricted open-shell wavefunction, but now it can also be ideally applied to current wavefunction, the orbitals 13 and 14 now could be regarded as SOMO. It is well-known that spin density of unrestricted open-shell wavefunction is determined by all occupied MOs, but after the biorthogonalization, the spin density will only be contributed by the SOMOs. Obviously, the spin density of current system directly corresponds to the sum of density of orbitals 13 and 14.

The HOMO and LUMO of singlet ethanol calculated at B3LYP/6-31G** level are shown below. By comparing this graph and the last graph, one can find that the HOMO and LUMO look very like the biorthogonalized alpha orbital 14 and 13, respectively. Hence it can be immediately understood that the transition of ethanol between singlet and triplet states can be well represented as HOMO→LUMO transition, because after such orbital transition both the HOMO and LUMO will be occupied by one electron, this electronic structure just corresponds to that represented by the biorthogonalized orbitals. Clearly, such valuable information cannot be obtained without the biorthogonalization analysis, rendering importance of biorthogonalization in practical studies of open-shell systems.



It worth to emphasize that the biorthogonalization does not alter any observable properties of the system, such as total electronic energy, total density, spin density and so on. In addition, during the biorthogonalization, the program does not update orbital energies (actually the orbital energies should be changed), therefore after the biorthogonalization the orbital energies will be meaningless, they are still the energies of the original molecular orbitals, and the order of the biorthogonalized orbitals does not reflect order of their actual energies.

Biorthogonalization can also be applied to spin polarized singlet systems such as diradicals, an example is given in my blog article "Principle and application of biorthogonalization method for unrestricted open-shell wavefunctions" (in Chinese, <http://sobereva.com/448>).

Evaluating energy of biorthogonalized orbitals and ordering the orbitals

Next, I illustrate how to also make Multiwfn evaluate energies of the biorthogonalized orbitals and order them according to their energies, the triplet ethanol is still taken as example. Run *examples\ethanol_triplet_47.gif* by Gaussian, then you will get *ETHANOL_TRIPLET.47* in C:\ folder (which has been provided as *examples\ETHANOL_TRIPLET.47*). This file contains Fock/KS matrix, which is needed during evaluation of orbital energies. Boot up Multiwfn and input

```
examples\ethanol_triplet.fch // Generated during calculation of ethanol_triplet_47.gjf
100 // Other function (Part 1)
12 // Perform biorthogonalization between alpha and beta orbitals
2 // Do biorthogonalization for all orbitals
y // Evaluate energy of the biorthogonalized orbitals
examples\ETHANOL_TRIPLET.47 // Multiwfn will load Fock matrix from it and yield
energies of biorthogonalized orbitals
y // Ordering the biorthogonalized orbitals according to their energies
```

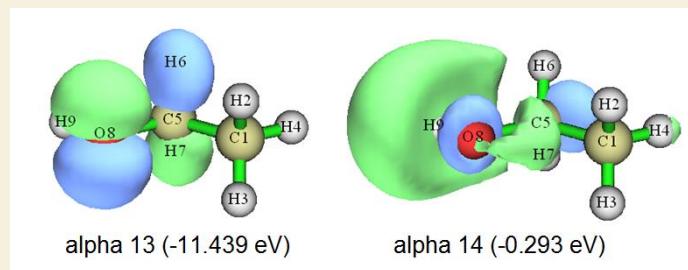
Now you can find the exported *biortho.fch* and *biortho.txt* in current folder. The content of the later one is:

```
...[ignored]
orb: 11 S= 0.9992 E(A)= -13.057 O(A)= 1.0 E(B)= -11.821 O(B)= 1.0
orb: 12 S= 1.0000 E(A)= -11.918 O(A)= 1.0 E(B)= -11.866 O(B)= 1.0
-----
orb: 13 S= 1.0000 E(A)= -11.439 O(A)= 1.0 E(B)= -5.806 O(B)= 0.0
orb: 14 S= 1.0000 E(A)= -0.293 O(A)= 1.0 E(B)= 2.817 O(B)= 0.0
-----
orb: 15 S= 0.9992 E(A)= 13.876 O(A)= 0.0 E(B)= 15.431 O(B)= 0.0
orb: 16 S= 1.0000 E(A)= 15.057 O(A)= 0.0 E(B)= 15.305 O(B)= 0.0
...[ignored]
```

Because in present example we have evaluated orbital energies, therefore the energies are also explicitly shown in the *biortho.txt*. It can be seen that the basically paired alpha and beta

biorthogonalized orbitals have detectably different energies, for example $E(\alpha 11) = -13.057\text{eV}$ while $E(\beta 11) = -11.821\text{ eV}$, this is because alpha and beta orbitals feel different effective potentials (*i.e.* alpha and beta Fock operators are different). In addition, one can find that the order of orbital index has in line with order of orbital energy, because we have requested Multiwfn to order the orbitals according to their energies (more specifically, the average energy of alpha and beta orbitals with the same index). Note that the orbital energy information in the *biortho.fch* now also corresponds to the actual energy of the biorthogonalized orbitals.

Then, you can input `y` to load the just generated *biortho.fch* so that we can then directly use main function 0 to visualize the biorthogonalized orbitals. The two occupied alpha biorthogonalized orbitals with highest indices, namely the two orbitals formally occupied by the two unpaired electrons in present system, are shown below.



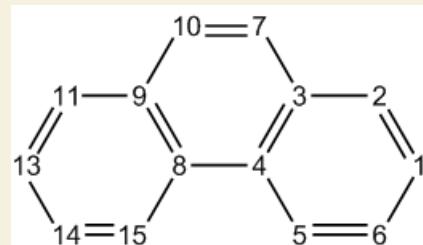
Their shapes are very close to the alpha MO 13 (-10.083 eV) and alpha MO 14 (-0.276 eV), and meantime the energies are not quite different.

Note that as mentioned in Section 3.100.12, the orbitals in each batch are ordered individually, therefore the relative order of orbitals in different batch will not alter due to the ordering. In addition, the ordering is based on average energy of each alpha orbital and its beta counterpart, thus alpha-beta orbital correspondence is kept during the ordering process.

4.100.13 Calculate HOMA and Bird aromaticity index for phenanthrene

HOMA is the most prevalently used aromaticity index based on geometry equalization, see Section 3.100.13 for detail. Here we use HOMA to study which ring of phenanthrene has stronger aromaticity.

Since calculation of HOMA only requires molecular coordinate, you can simply use such as .pdb and .xyz as input file. Of course, other files containing molecular coordinate, such as .wfm and .fch files are acceptable too. The geometry in present instance is optimized under B3LYP/6-31G* level.



Boot up Multiwfn and input following commands

examples/phenanthrene.pdb

```
100 // Other functions (Part 1)
13 // Calculate HOMA and Bird aromaticity index
0 // Start the calculation
```

You will see the default parameters are printed, they are taken from *J. Chem. Inf. Comput. Sci.*, **33**, 70, you can also change these parameters by yourself via option 1 before the calculation.

Now we input the atom indices in the ring that we are interested in, we calculate HOMA for the central ring first, so input 3,4,8,9,10,7, the input order must be consistent with atom connectivity. You will immediately obtain the result shown below

Atom pair	Contribution	Bond length(Angstrom)
3(C) -- 4(C):	-0.065698	1.427111
4(C) -- 8(C):	-0.210455	1.458000
8(C) -- 9(C):	-0.065698	1.427111
9(C) -- 10(C):	-0.096016	1.435282
10(C) -- 7(C):	-0.033673	1.360000
7(C) -- 3(C):	-0.096016	1.435282

HOMA value is 0.432442

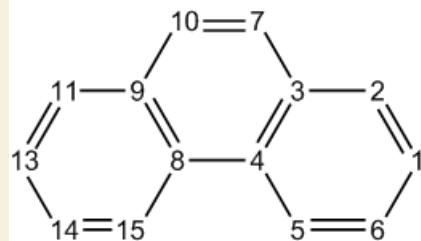
Obviously, C4-C8 deviates to ideal bond length 1.388 most significantly, giving rise to large negative contribution to HOMA, in other words, significantly breaked aromaticity. The HOMA value is calculated as 1 plus the contributions from all bonds in the ring.

Then input 8,15,14,13,11,9 to calculate HOMA for the boundary ring, the result is 0.855126. Since this value is much closer to 1 than the one for central ring, HOMA suggests that the two boundary rings have stronger aromaticity.

Bird index is another quantity used to measure degree of aromaticity, see Section 3.100.13 for a brief description. Now choose option 2 to calculate Bird index for the two rings, you will find the value for boundary ring is more close to 100 than the central ring. Likewise HOMA, Bird index also indicates that the two boundary rings have stronger aromaticity.

4.100.14 Calculate LOLIPOP for phenanthrene

In this instance, we use LOLIPOP (Localized Orbital Locator Integrated Pi Over Plane) index to evaluate which ring of phenanthrene has stronger π -stacking ability. For detail about LOLIPOP, see Section 3.100.14 and the paper *Chem. Commun.*, **48**, 9239 (2012).



Boot up Multiwfn and input following commands

examples/phenanthrene.wfn

```
100 // Other functions (Part 1)
14 // Calculate LOLIPOP
```

I // Specify π orbitals

*36,40,43,44,45,46,47 //*These orbitals are π orbitals. You can find out π orbitals by visualizing orbital isosurfaces via main function 0 or by function 22 of main function 100

0 // Start the calculation

*8,9,11,13,14,15 //*The indices of the atoms in the ring you are interested in. This ring is boundary ring

Wait for a while, from screen we can see that the LOLIOP value is 8.233.

0 // Start the calculation again

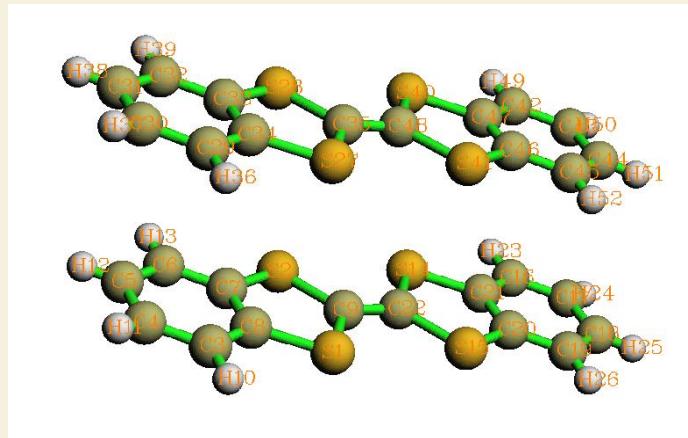
*7,3,4,8,9,10 //*Central ring

The LOLIOP value is 6.183

Since smaller LOLIOP value corresponds to stronger π -stacking ability, we can expect that the tendency of forming π -stacking structure for central ring is stronger than for boundary ring.

4.100.15 Calculate intermolecular orbital overlap integral of DB-TTF

Intermolecular orbital overlap integral is important in discussions of intermolecular charge transfer, see Section 3.100.15 for introduction. In this example, we calculate HOMO-HOMO and LUMO-LUMO overlap integrals between the two DB-TTF (Dibenzotetrathiafulvalene) monomers in below dimer. The dimer structure was extracted from CSD (Cambridge Structural Database). The wavefunction level we used is PW91PW91/6-31G*, MO78 and MO79 correspond to HOMO and LUMO, respectively.



First, we construct a Gaussian input file for dimer according to the structure. *IOp(3/33=1) nosymm guess=only* should be present in route section. *IOp(3/33=1)* tells Gaussian to output overlap matrix in original basis; *nosymm* instructs Gaussian to do not adjust the coordinate to standard orientation; *guess=only* is specified because we do not require converged wavefunction of the dimer. The input file is provided as *examples\intermol\DB-TTFdimer.gjf*. Run it by Gaussian, and then we obtain *DB-TTFdimer.out*.

Next, we extract the first and the second monomer coordinates in turn from the dimer input file to construct two Gaussian input files for the two monomers, named as *DB-TTF1.gjf* and *DB-TTF2.gjf*, the two files can also be found in *examples\intermol* folder. *nosymm* and *pop=full* should be specified in route section. Use Gaussian to run the two files to yield *DB-TTF1.out* and *DB-TTF2.out*.

After that, boot up Multiwfn and input below commands:

DB-TTFdimer.out

100 // Other functions (Part 1)

15 // Then Multiwfn will read overlap matrix (in original basis) from DB-TTFdimer.out

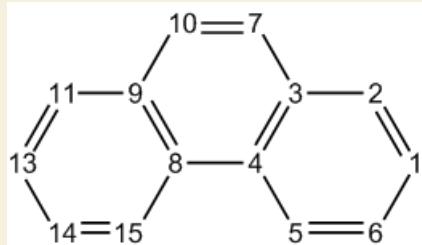
DB-TTF1.out // Multiwfn will read MO coefficients of monomer 1 from the file

DB-TTF2.out // Multiwfn will read MO coefficients of monomer 2 from the file

Now, if we input *i,j*, then the intermolecular orbital overlap integral between MO *i* in monomer 1 and MO *j* in monomer 2 will be printed. To obtain the integral between HOMO-HOMO, input 78,78, we immediately get the value -0.01389242; Then input 79,79, we will find the integral between LUMO-LUMO is 0.00906935.

4.100.18 Yoshizawa's electron transmission route analysis for phenanthrene

In this example, we will use Yoshizawa's formula (*Acc. Chem. Res.*, **45**, 1612 (2012)) to analyze favourable electron transmission routes for phenanthrene. Related theory, requirement of input file and program options have been introduced in Section 3.100.18. The numbering scheme of the carbons is shown below.



Boot up Multiwfn and input following commands:

examples\phenanthrene_NAOMO.out //The Gaussian output file containing "NAOMO" information

100 // Other functions (Part 1)

18 // Yoshizawa's electron transport route analysis

2 // Select YZ plane, which is the molecular plane

Then program will detect which atom has expected p_z atomic orbitals, and load their expansion coefficients in all MOs.

Now we select 1, and input 2,11 to check the transmission probability between 2 and 11. From the output we can know that the transmission probability is 0.855879. The contributions from each MO are also shown. From the output we also know that the probability will be 2.144432 if only HOMO and LUMO are considered. Although Yoshizawa's paper said that in common one only need to take HOMO and LUMO into account, it seems that this approximation is not true in quantitative level. The distance route 5.674656 Angstrom is the distance between atom 2 and 11.

Next, we examine which transport routes are the most favourable. Suppose that this time we only want to consider HOMO and LUMO, so we choose option -1 and input 47,48. Then choose option 2, you will see

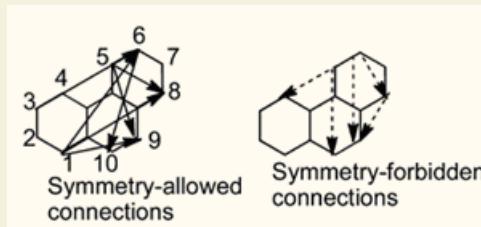
Note: The routes whose absolute value < 0.010000 will not be shown

Note: The routes whose distance < 0.0000 or > 9999.0000 Angstrom will not be shown

Atom	7 -- Atom	10	value and distance:	3.743670	1.359562
Atom	6 -- Atom	10	value and distance:	3.098919	5.081382
Atom	7 -- Atom	14	value and distance:	3.098919	5.081382
Atom	7 -- Atom	11	value and distance:	-2.831680	3.720990
Atom	2 -- Atom	10	value and distance:	-2.831680	3.720990
Atom	6 -- Atom	14	value and distance:	2.565337	5.765892
...					
Atom	6 -- Atom	9	value and distance:	-0.016585	4.927289
Atom	3 -- Atom	14	value and distance:	-0.016585	4.927289
Atom	2 -- Atom	4	value and distance:	-0.014931	2.456189
Atom	8 -- Atom	11	value and distance:	-0.014931	2.456189

The routes are ranked by transmission probability (absolute value). If we ignore the atoms linking three carbons and the duplicated routes due to symmetry, the most favourable four routes are 7-10, 6-10, 7-11, 6-14. This conclusion is completely in line with below graph in Yoshizawa's paper!

Note that Yoshizawa stated that connection 9-10 (corresponding to 7-10 in our numbering scheme) is predicted to be the best route from the rule, but it is too close to construct a metal-molecule-metal junction for connection, so the route was not marked on his graph.



Now we check the feasible routes started from atom 2. Choose option 3, and input 2, you will see

To atom	10	value and distance (Angstrom):	-2.831680	3.720990
To atom	14	value and distance (Angstrom):	-2.344569	6.206600
To atom	11	value and distance (Angstrom):	2.144432	5.674656
To atom	5	value and distance (Angstrom):	-1.571512	2.787501
...				

Evidently, 2-10 is the most favourable route.

4.100.19 ELF analysis on the promolecular wavefunction combined from fragment wavefunctions

Before reading this section please read Section 3.100.19 first to gain basic knowledge.

It is usually interesting to analyze the characteristic of promolecular wavefunction, which corresponds to the state without any electron transfer and polarization due to the interaction between the fragments constituting the system. Commonly, we can use "custom operation" in main function 3, 4 and 5 to realize this purpose, see Section 3.7 for introduction of custom operation and some illustrative applications in Sections 4.5.4 and 4.5.5. For example, we want to study electron density

distribution of promolecular state of a complex AB , by using custom operation, we can very conveniently instruct Multiwfn to calculate electron density of A and that of B respectively, and then sum them up as electron density of the promolecule state. However, this process is not applicable to non-linear real space functions such as ELF; that is to say, ELF of promolecular state of AB is not equal to the sum of ELF of A and ELF of B , the result is completely meaningless. For such cases, we should combine fragment wavefunctions first as promolecular wavefunction, and then calculate ELF for it to obtain the ELF distribution in promolecular state.

Below I will show how to use Multiwfn to produce promolecular wavefunction for COBH_3 based on fragment wavefunctions of CO and BH_3 , and then discuss the corresponding ELF character. The .wfn files used below and the corresponding Gaussian .gjf files can be found in "examples\genpromol" folder.

Boot up Multiwfn and then input

```
examples\genpromol\COBH3\CO.wfn // The path of wavefunction file of fragment 1
```

```
100 // Other functions (Part 1)
```

```
19 // Generate promolecular .wfn file from fragment wavefunctions
```

```
2 // Two fragments in total
```

```
examples\genpromol\COBH3\BH3.wfn // The path of wavefunction file of fragment 2
```

Now the promolecular wavefunction file of COBH_3 has been outputted to promol.wfn in current folder.

Let us plot ELF for this promolecular wavefunction. Reboot Multiwfn and input

```
promol.wfn
```

```
4 // Draw plane map
```

```
9 // ELF
```

```
1 // Color-filled map
```

```
[Press ENTER button]
```

```
2 // XZ plane
```

```
0 // Y=0
```

Interestingly, even in the promolecular state, from the resultant graph it looks as if the carbon and boron have been bonded to each other. In order to make clear how the relaxation of electron distribution affects the ELF character of COBH_3 , we decide to draw difference map of ELF between the actual state and promolecular state.

Reboot Multiwfn and input

```
examples\genpromol\COBH3\COBH3.wfn // Wavefunction file of actual state of COBH3
```

```
4 // Draw plane map
```

```
0 // Custom operation
```

```
1 // Deal with only one file
```

```
-.promol.wfn // Subtracting property of COBH3.wfn by that of promol.wfn
```

```
9 // ELF
```

```
1 // Color-filled map
```

```
[Press ENTER button]
```

```
2 // XZ plane
```

```
0 // Y=0
```

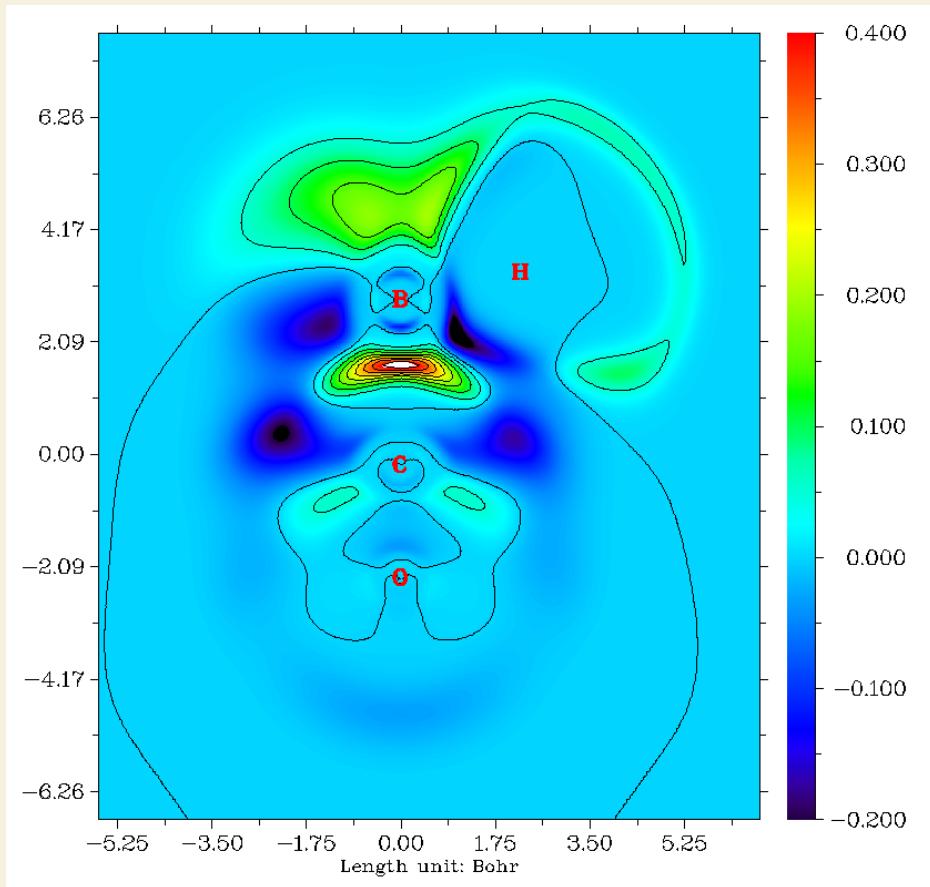
Close the graph and then input

```
1
```

-0.2,0.4 // Set the color scale from -0.2 to 0.4, since as you can see from the command-line window, in this plane the data range is from -0.248 to 0.436

2 // Enable showing contour lines

-I // Show the graph again



From this difference map of ELF, it is very clear that after electron relaxation, the electron localization character between the C-B bond enhanced evidently.

Next I will show how to create promolecular wavefunction when open-shell fragments are involved. CH₃NH₂ is taken as example, the two fragments are CH₃ and NH₂ free-radicals. Boot up Multiwfn and input

examples\genpromol\CH3NH2\CH3.wfn

100 // Other functions (Part 1)

19 // Generate promolecular .wfn file from fragment wavefunctions

2 // There are totally two fragments (including the loaded one)

examples\genpromol\CH3NH2\NH2.wfn // Wavefunction file of the second fragment

n // Do not flip spin of orbitals of CH₃

y // Flip spin of orbitals of NH₂. If you do not understand why the spin should be flipped, you can consult the corresponding CDA example in Section 4.16.2.

Now you have promolecular wavefunction of CH₃NH₂ in current folder. Try to plot ELF for it and compare the result with actual state (i.e. *examples\genpromol\CH3NH2\CH3NH2.wfn*)

4.100.21 Calculate molecular diameter and length/width/height

As described in Section 3.100.21, subfunction of 21 of main function 100 can be used to calculate a variety of that solely based on molecular geometry. This function can also calculate molecular diameter and length/width/height, in this section I will use *examples/alpha-cyclodextrin.pdb* an instance to illustrate this point. Please first read Section 3.100.21 to understand how this function work.

Boot up Multiwfn and input

examples/alpha-cyclodextrin.pdb

100 // Other function (Part 1)

21 // Calculate quantities that purely based on geometry

size // Calculate molecular diameter and length/width/height

You will see below information on screen

Farthest distance: 44(H) --- 123(H): 14.027 Angstrom

vdW radius of 44(H): 1.200 Angstrom

vdW radius of 123(H): 1.200 Angstrom

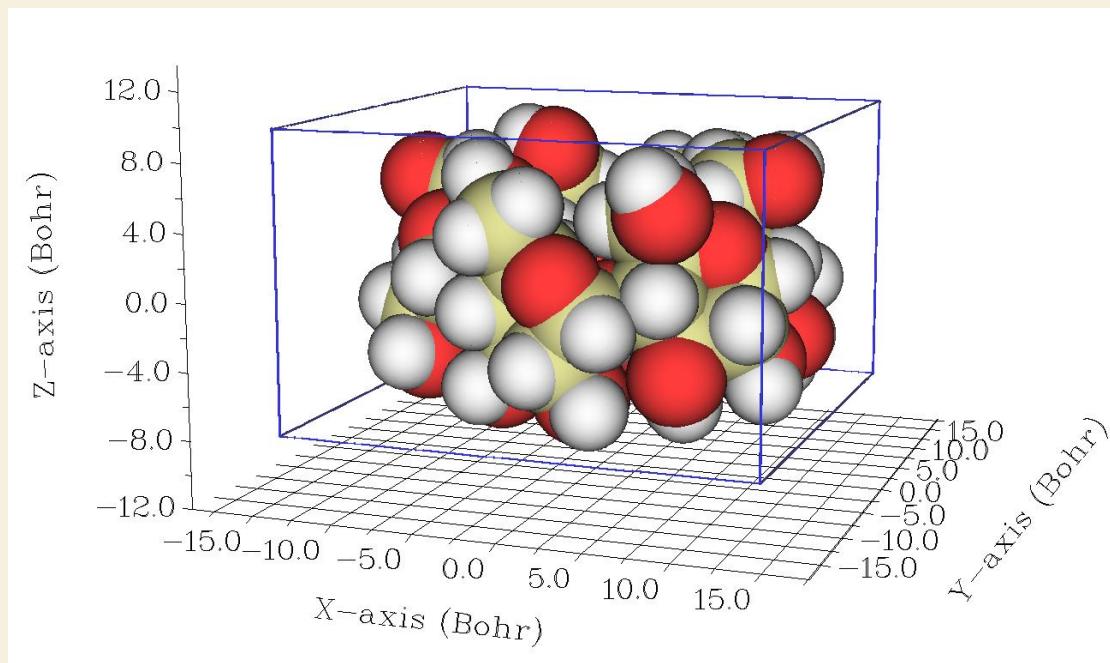
Diameter of the system: 16.427 Angstrom

Radius of the system: 8.213 Angstrom

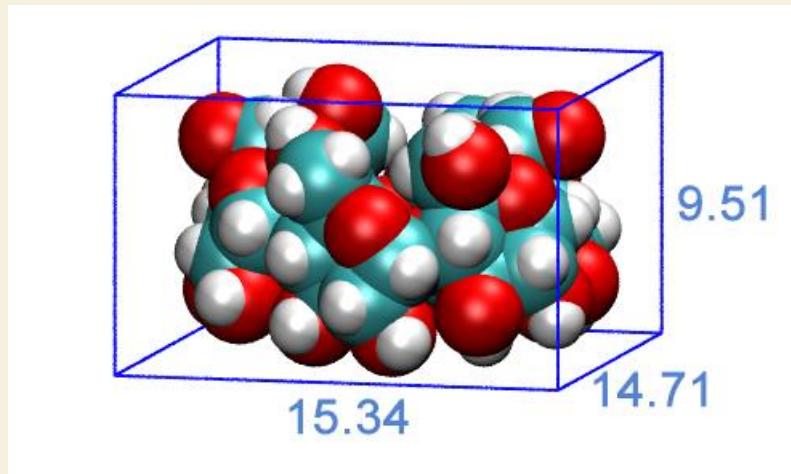
Length of the three sides: 15.341 14.714 9.511 Angstrom

The diameter 16.427 Å printed on screen is calculated as $14.027 + 2 \times 1.2$. The radius 8.213 Å is simply the half of the diameter. The length/width/height of the molecule, namely the three values after "Length of the three sides" are calculated as follows: Multiwfn first automatically rotates the molecule so that its three principal axes just parallel to the three Cartesian axes, and meantime translates the molecule to put its geometry center to original point. Then according to the position of boundary atoms (*i.e.* the atoms having maximum/minimum value of X/Y/Z coordinate) and atomic Bondi van der Waals radii, the length/width/height of the molecule can be derived straightforward.

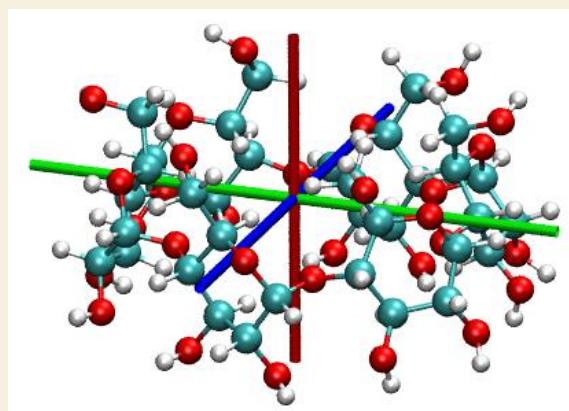
If you want to visually depict the length/width/height, you can then choose option 1, you will see below graph (the "Ratio of atomic size" has been set to 4.0, in this situation radius of atomic sphere just equals to atomic van der Waals radii). As you can see, the blue box tightly encloses the molecular van der Waals surface, the length of its three sides are simply the length/width/height of the molecule, *i.e.* 15.341, 14.714 and 9.511 Å, respectively.



You can also select option 2 to export the molecule in rotated and translated coordinate to *new.pdb* in current folder. This file contains "CRYST1" field, which records cell lengths. You can directly load this file into VMD (<http://www.ks.uiuc.edu/Research/vmd/>) visualization program, and then input *pbc box* command in VMD console window to show box. After slight adjustment of plotting effect and manually editing the graph, you will obtain below graph, which clearly illustrates the molecular geometry character.



It is also possible to visualize molecular principal axes in VMD. Simply copy all content of *examples\principal_axes.tcl* into VMD console window, you will see below graph, in which the red, blue and green axes respectively represent the three molecular principal axes.



More discussions about this topic are given in my blog article "Using Multiwfn to calculate the length, width and height of molecules" (in Chinese, <http://sobereva.com/426>).

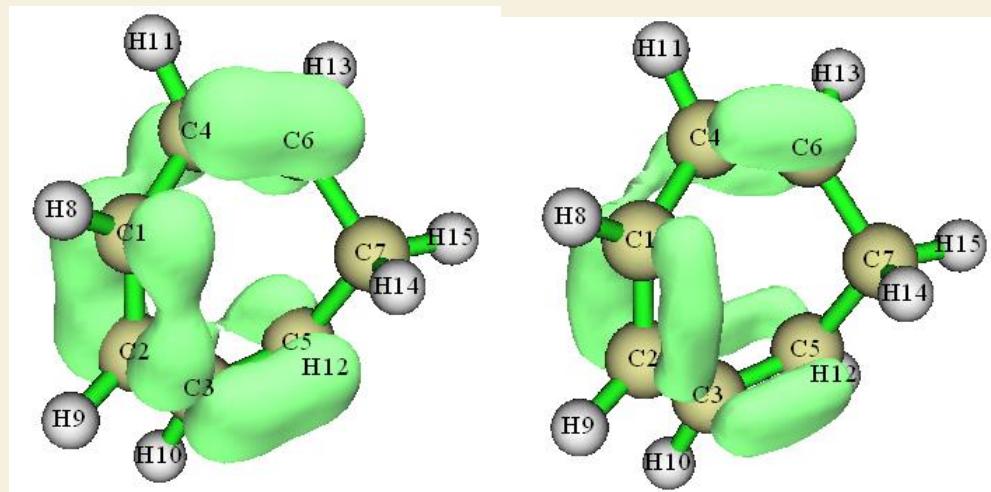
4.100.22 Analyze π electron character of non-planar system: cycloheptatriene

Before reading this section please read Sections 3.100.22 and 4.5.3 first. In Section 4.5.3, I have mentioned how to let Multiwfn automatically detect π orbitals for a exactly planar system and then separately study σ and π electron characters by means of ELF- σ and ELF- π . In addition, in Section 4.4.9, I illustrated how to study π electron delocalization path for a planar system porphyrin by plotting LOL- π map. In this section, I will show that it is also possible to make Multiwfn automatically identify π type of localized molecular orbitals for *non-planar* systems, so that separately studying σ and π electrons can be easily conducted. We will use a simple non-planar molecule cycloheptatriene as example.

For non-planar systems, to separate σ and π electrons, the molecular orbitals must be firstly transformed to localized molecular orbitals (LMOs). Boot up Multiwfn and input:

```
examples\cycloheptatriene.fch
19 // Orbital localization
1 // Localize occupied orbitals
100 // Other function (Part 1)
22 // Detect  $\pi$  orbitals
-1 // This mode is specific for LMOs
0 // Detect  $\pi$  LMOs under default settings
```

There are three π LMOs identified, assume that we will study π electron character, so we choose 2 to set occupation number of all other orbitals to zero. Then we can carry out any kind of analysis, the result will be only contributed by π electrons. For example, we plot electron density isosurface map (isovalue=0.03 a.u.) and LOL isosurface map (isovalue=0.5) using main function 5, they are shown below as left and right graphs, respectively.



From both the graphs, it is easy to recognize that C4-C6, C1-C2 and C3-C5 must be stronger than C1-C4 and C2-C3, since π electrons are delocalized much more substantially among the corresponding two atoms in the formers

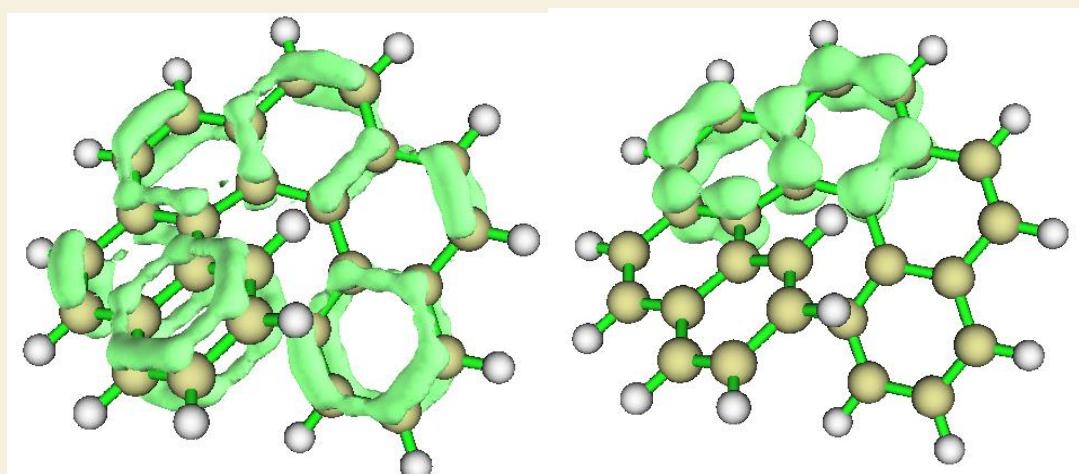
Based on π type of LMOs, we can further carry out other type of analyses, for example, Mayer bond order calculation. We calculate Mayer bond order as illustrated in 4.9.1, the result is

The total bond order ≥ 0.050000

#	1:	1(c)	2(c)	0.70534437
#	2:	1(c)	4(c)	0.18472895
#	3:	1(c)	5(c)	0.10215925
#	4:	2(c)	3(c)	0.18472484
#	5:	2(c)	6(c)	0.10215398
#	6:	3(c)	5(c)	0.80861164
#	7:	4(c)	6(c)	0.80860020
#	8:	5(c)	6(c)	0.07556895

The order of the π bond order is C4-C6 > C1-C2 > C1-C4, this result is in full agreement with our expectation via visualizing the ELF- π map.

The function illustrated in this section is quite powerful, and can also be applied to large systems. For example, below left graph is LOL- π isosurface map (isovalue=0.55) of helicene plotted using above procedure. Moreover, using option "5 Set constraint of atom range", one can define constraint of the region for identifying π LMOs. For example, below right graph shows the electron density of helicene contributed by five π LMOs located at the central two six-membered rings.



Note that multi-center bond order calculation can also be normally carried out under LMO representation, therefore you can easily identify strength of π conjugation of different rings in above system.

More discussions and illustrations about analyzing π electron character can be found in my blog article "Separate investigation of π electronic structure in Multiwfn" (in Chinese, <http://sobereva.com/432>).

4.200 Other functions (Part 2)

4.200.4 Study iso-chemical shielding surface (ICSS) and magnetic shielding distribution for benzene

Iso-chemical shielding surface (ICSS) denotes isosurface of magnetic shielding value, which presents intuitive picture on aromaticity. If you are familiar with NICS, you can also simply view ICSS as the isosurface of NICS with inverted sign. Please see Section 3.200.4 for more information. In this example we will study benzene, since this is a planar system, we will plot ICSS_{ZZ} instead of ICSS, namely only the component of magnetic shielding tensor perpendicular to molecular planar will be taken into account. ICSS_{ZZ} must be more physically meaningful than ICSS, just like NICS_{ZZ} is a better aromaticity index than NICS (proved in *Org. Lett.*, **8**, 863). Meanwhile I will also show how to plot magnetic shielding value in a line and in a plane.

Note that Multiwfn itself is incapable of calculating magnetic shielding tensor and requires Gaussian to do that.

Before running Multiwfn, you should prepare a Gaussian input file of standard single point task for present system, which later will be taken as template. This file has already been provided as "examples\ICSS\benzene.gjf". Meantime, you should prepare a file containing the structure of corresponding system, for benzene this file is provided as "examples\ICSS\benzene.pdb"

Boot up Multiwfn and input below commands

```
examples\ICSS\benzene.pdb // Note that molecular plane is in XY plane
200 // Other functions (Part2)
4 // Generate grid data of ICSS or related quantities
-10 // Change extension distance, because the default value is somewhat too small for this molecule
12 // 12 Bohr of extension distance
1 // Low quality grid, magnetic shielding tensor at 130910 points will be calculated by Gaussian later. Since present system is not very small, using medium or better quality grid will be expensive
n // Do not skip the step of generating Gaussian input file, because this is the first time we carry out analysis and thus currently we do not have Gaussian input/output files in hand
examples\ICSS\benzene.gjf // The path of template input file
```

Now Multiwfn generates a lot of Gaussian input files of NMR task in current folder based on the template file. The files are named as NICS0001.gjf, NICS0002.gjf ... NICS0017.gjf. Run these files by Gaussian.

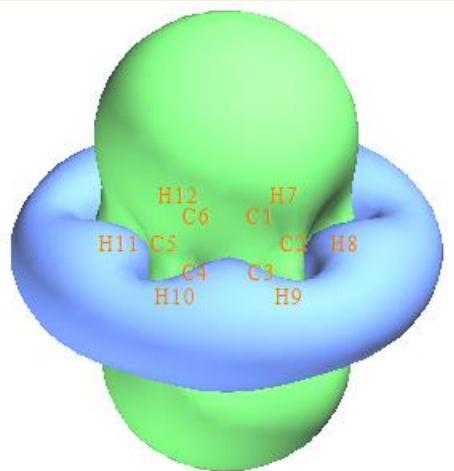
Note: If these files cannot be runned by your Gaussian normally, please check the tail of the output file, there are two common reasons:

(1) The %mem is too small to finish the task, you need to set %mem in the template .gif file to a large value and retry.

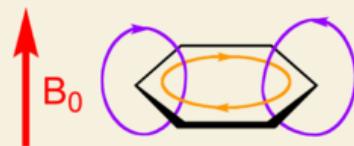
(2) The "NICSnptlim" in settings.ini is too large, you should properly reduce it and try again. The reason is that in Gaussian there is a limit on the number of Bq atoms, and it is somewhat dependant of the version of Gaussian and your computer. For G09 D.01 and E.01, you should add "guess=huckel" keyword, otherwise due to memory allocation bug, the NICSnptlim has to be reduced to a very small value to make Gaussian run normally (in this case the overall computational cost will be quite high). If error occurs in Link401 when "guess=huckel" is used, then try to use "guess=core" instead. For G16, the guess keyword is not needed.

Hint: You can make use of the script "examples\runall.sh" (for Linux) or "examples\runall.bat" (for Windows), which invokes Gaussian to run all .gif files in current folder to yield output files with the same name but with .out suffix.

Assume that the output files (NICS0001.out, NICS0002.out...) have been placed in "C:\benzene" folder, in the window of Multiwfn input C:\benzene\NICS and then input 5 to select ZZ component, Multiwfn will load these output files and convert magnetic shielding tensors to grid data of ICSS_{ZZ}. After that you can see a new menu, you can directly visualize isosurface of the grid data by option 1, export it as cube file by option 2 or re-select the property you are interested in, for example ICSS or ICSS_{XX}. The isosurface of ICSS_{ZZ}=2.0 is shown below



As you can see, the green isosurface (positive Z-component shielding value), completely covers the region above and below the benzene ring, suggesting that due to the induced ring current originated from the globally delocalized π -electrons, the Z-direction external magnetic field is largely shielded in these regions, this observation implies the strong aromaticity of benzene. From below scheme we can understand the ICSS_{ZZ} more deeply; in the cylindrical region perpendicular to and through the benzene, the direction of induced magnetic field (purple arrows) is exactly opposite to external magnetic field (B_0), this is why in this region Z-component of magnetic shielding value is large.



You can also see, blue isosurface (negative Z-component shielding value) presents in the outlier region of benzene, exhibiting de-shielding effect. This is mostly because the induced magnetic field is parallel to B_0 and thus enhances B_0 in this region.

If you properly rotate viewpoint, you will clearly find the C-H bond is also completely covered by the green surface. The reason is that the σ -electrons involved in the C-H bonding form

conspicuous local induced ring current, so the external magnetic field is also strongly shielded around the C-H bond.

Now close the GUI window, and select 2 to export the grid data to ICSSZZ.cub in current folder.

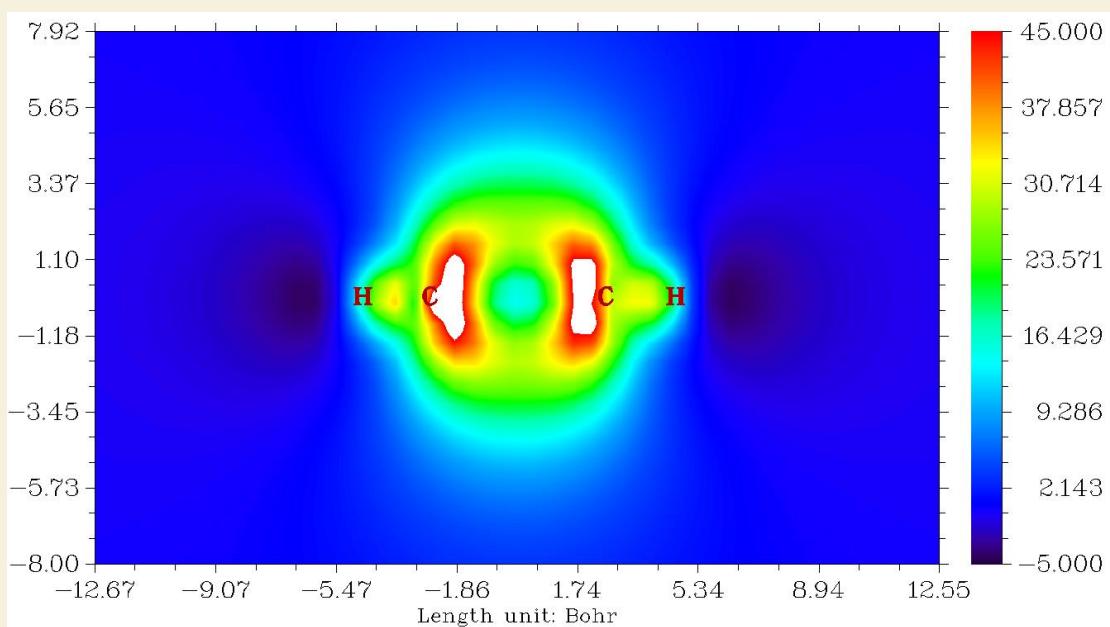
Digression: Assume that you have obtained the Gaussian output files, and the Multiwfn instance has been terminated due to some reasons, but now you would like to re-analyze ICSS for this system, you should exactly repeat the steps what you have done before, the grid must be set in the same manner, then when Multiwfn asks you "If skip generating Gaussian input file of NMR task?", you should input "y", then the Gaussian input files will not be re-generated, and you can directly feed the Gaussian output files you have obtained into Multiwfn.

Next I will show how to plot magnetic shielding value in a line and in a plane. Since we already have grid data of ICSSZZ in hand, magnetic shielding value at any point in a line/plane can be easily obtained by means of linear interpolation technique based on the grid data.

We first plot color-filled map for ICSSZZ in the YZ plane with X=0. This plane is normal to benzene and crosses C4-H10 and C1-H7. Set "iuserfunc" in settings.ini to -I, and then boot up a new Multiwfn instance and input below commands

```
ICSSZZ.cub
4 // Plot plane map
100 //User-defined function, which now corresponds to the function interpolated by the grid
data of ICSSZZ.cub
1 // Color-filled map
[Press Enter]
0 // Set extension distance of the plot
8
3 // YZ plane
0 // X=0

Now the graph pops up, close it and then input
4 // Show atom labels
11 // Crimson text
1 // Change lower and upper limit
-5,45
-I // Replot the map
Now you can see below map
```

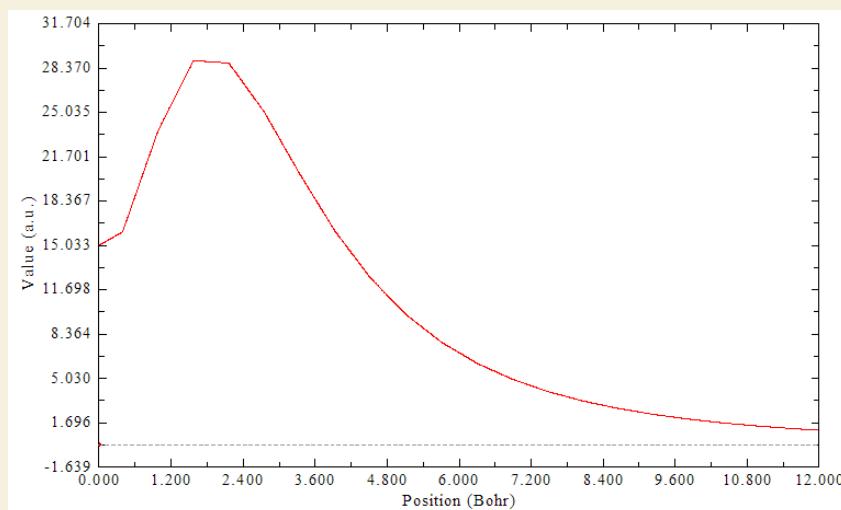


From the graph one can find that although Z-component of magnetic shielding in the center of benzene is an evident positive value, the magnitude is by far less than in the region above and below the ring plane. The reason is clear, that is benzene only has π -aromaticity, while its σ -electrons are not globally delocalized to form σ -aromaticity, so the shielding effect is relatively weak in the plane due to lack of contribution of σ -ring current.

Next, we plot curve map to study the variation of magnetic shielding in the line perpendicular to ring plane and starting from ring center. Choose -5 to return to main menu and input

```
3 // Plot curve map
100
2 // Input coordinate of two points to define a line
0,0,0,0,0,12 // The line starts from ring center (0,0,0) and ends at 12 Bohr above the ring
plane
```

You will immediately see



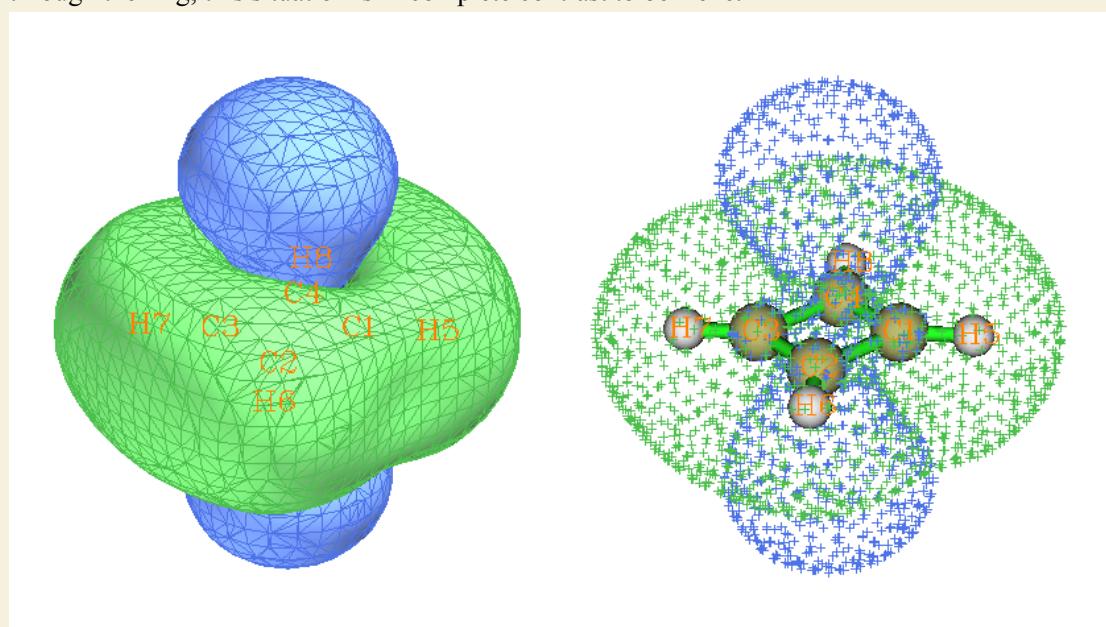
It can be seen that the maximum of Z-component of magnetic shielding value occurs about 1.8 Bohr above the ring plane.

Probably you have noticed that in some regions the curve map and plane map are not smooth

enough, this is because the grid of ICSSZZ.cub is relative coarse (the grid spacing is large). If we use denser grid when calculating ICSSzz, the graphs will be improved. Also beware that since the extension distance used in the calculation of grid data of ICSSzz is only 12 Bohr, when we plot curve or plane map based on the interpolated data of ICSSzz, the spatial range involved in the map should not be too large. For example, we cannot plot the curve map from (0,0,0) to (0,0,20).

ICSS is very useful for discussing aromaticity and anti-aromaticity, many instances can be found in the original paper of ICSS (*J. Chem. Soc. Perkin Trans. 2*, **2001**, 1893), and in some applicative papers, such as *J. Phys. Chem. A*, **116**, 5674 (2012). AFAIK, ICSSzz is first proposed by me in this manual and has not been discussed in any literature yet.

I strongly recommend the users do some more practices about plotting and analyzing ICSS/ICSSzz, I provided some ideal exercise systems in "examples\ICSS" folder, including azulene, cyclobutadiene, cycloheptatriene, porphyrin, propane and pyracylene; among them cyclobutadiene is the most simple one. Below is the ICSS=0.5 isosurface of cyclobutadiene showing in two representations; from the graph it is clear that this system shows strong anti-aromaticity character, the $4n$ π -electrons cause evident de-shielding effect in the cylindrical region perpendicular to and through the ring, this situation is in complete contrast to benzene.



I wrote a very detailed post to discuss ICSS, in which all systems in "examples\ICSS" folder are involved, see "Using Multiwfn to study aromaticity by drawing iso-chemical shielding surfaces" (in Chinese, <http://sobereva.com/216>).

4.200.5 Plot radial distribution function of electron density

Multiwfn is capable of plotting radial distribution function (RDF) for any real space function, see Section 3.200.5 for detail. This function is particularly useful for studying electronic structure character of sphere-like system.

This section consists of two parts. In part 1, we will plot RDF for electron density of fullerene (C_{60}); while in part 2, I will show how to plot RDF of electron density for a Rydberg orbital to characterize it quantatively.

Part 1: RDF of electron density for fullerene

Since .wfn file of fullerene at B3LYP/6-31G* level is large, I only provide the corresponding Gaussian input file for you (C60.gjf in example folder), please properly modify and run it by Gaussian to produce C60.wfn.

Boot up Multiwfn and input:

C60.wfn

200 // Other functions (Part 2)

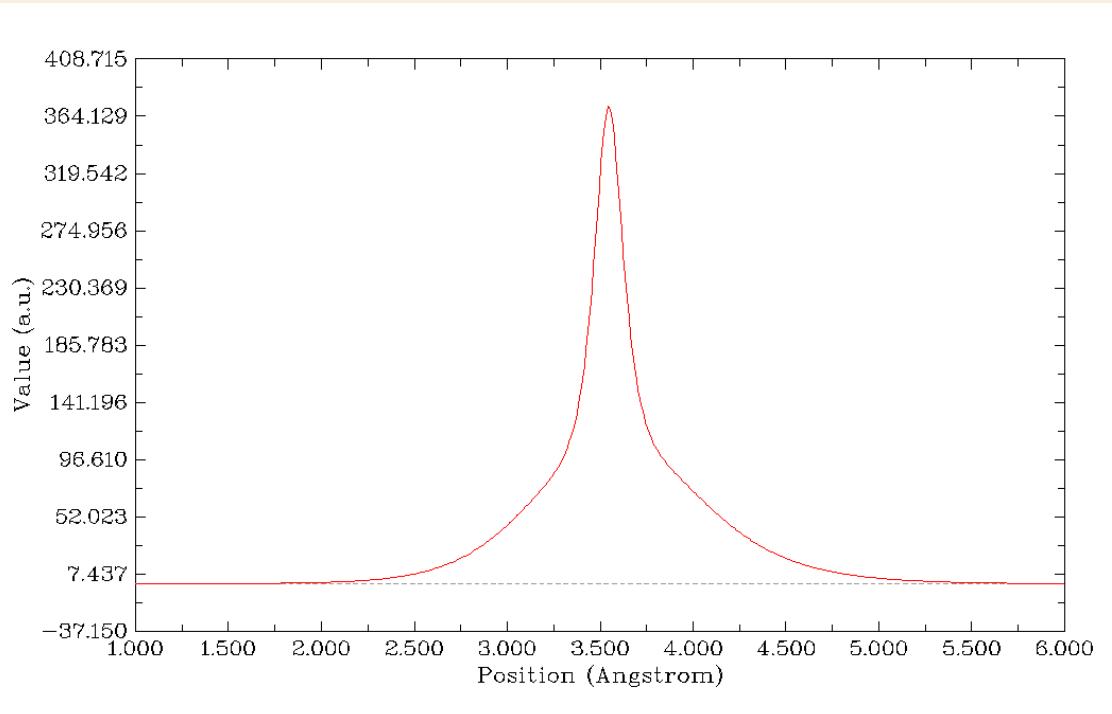
5 // Plot RDF for a real space function

3

1,6 // Set the lower and upper limit of RDF to 1.0 and 6.0 Angstrom, respectively

0 // Calculate RDF and its integration curve

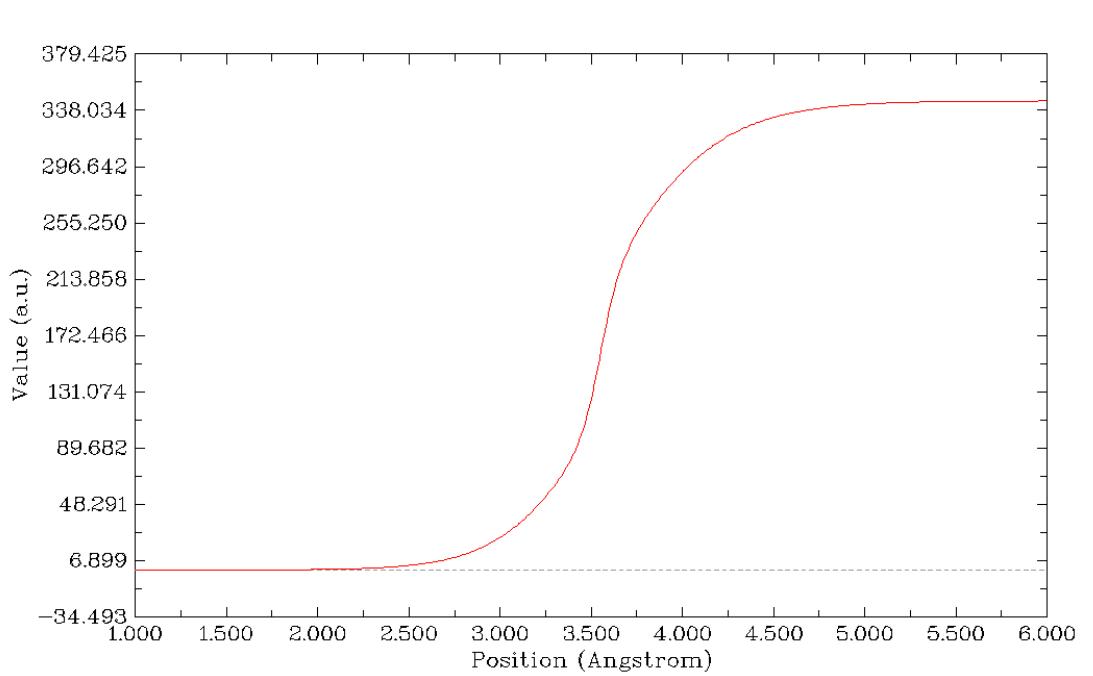
After the calculation is finished, select option 0, below RDF map will be shown on the screen, the X-axis corresponds to radial distance



As you can see, the peak of RDF is about 3.5 Å, this is because the distance between nucleus of carbons and the sphere center is 3.545 Å. It is known that electron density has maximum at nuclear position for any atom except for hydrogen.

If you carefully examine the RDF curve, you will find that the curve on the right side of the peak is slightly higher than that on the left sight. The reason is that the amount of π electrons at outer side of fullerene is richer than inner side. You can also draw and analyze ELF isosurface map to confirm this point.

You can also choose option 2 to plot integration curve of RDF, as shown below



Probably you have noticed that after the calculation there is a prompt on screen:

Integrating the RDF in the specified range is 344.9321931513

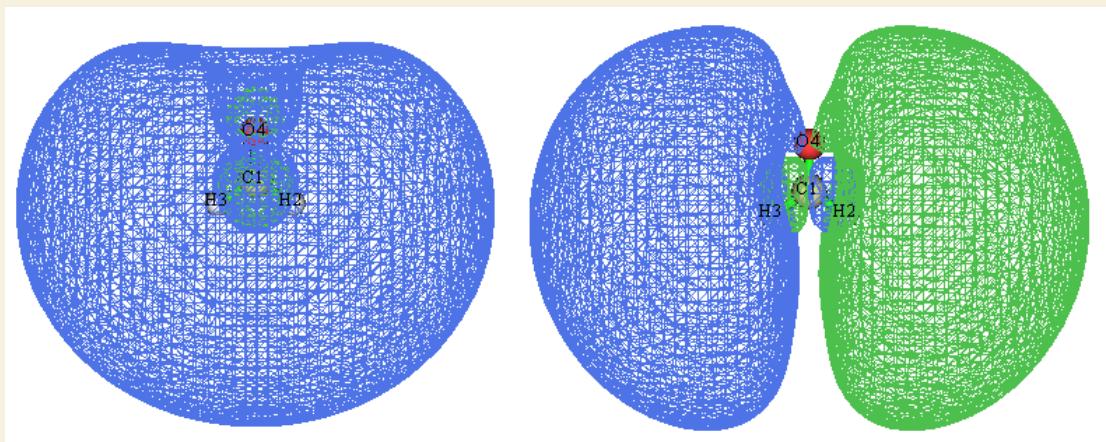
which means integrating the RDF from $r=1.0$ to $r=6.0$ Å yields 344.932, it also corresponds to the value of integration curve at $r=6.0$. This value deviates evidently from our expectation, namely the number of electrons in current system (360). One reason is that the default number of integration points is not large enough (500 and 2030 for radial and angular parts, respectively. You can manually increase them), while another reason is more important, that is the size of current system is too large, using single-center integration method is too difficult to yield very accurate result, at least for integrating electron density.

Part 2: Using RDF of electron density to quantitatively characterize Rydberg orbital for acetone

Rydberg orbitals denote the spatially very diffuse MOs, their orbital shapes are akin to atomic orbitals, since electrons in Rydberg orbitals can be regarded as weakly bounded by a small cation core, which behaves as an atomic nucleus. In order to faithfully represent the diffuse character of Rydberg orbitals, basis set with substantial diffuse functions must be employed, e.g. aug-cc-pVTZ.

The example file we used below is formaldehyde calculated at B3LYP/aug-cc-pVTZ level by Gaussian. We will first visualize Rydberg orbitals as isosurfaces, and then calculate RDF of electron density corresponding to these orbitals to characterize them quantitatively.

Boot up Multiwfn, load *examples\H2CO_aVTZ.fch*, then enter main function 0. Since Rydberg orbitals are very diffuse, in order to avoid truncating their isosurfaces when viewing them, you should first select "Other settings"- "Set extension distance" and input a large value, here we input 12. After that, change isovalue to a much smaller value than default, e.g. 0.01. Then we arbitrarily select some virtual MOs to examine their features. You will find a lot of virtual MOs show very diffuse character, for example, MO10 and MO11 are shown below:



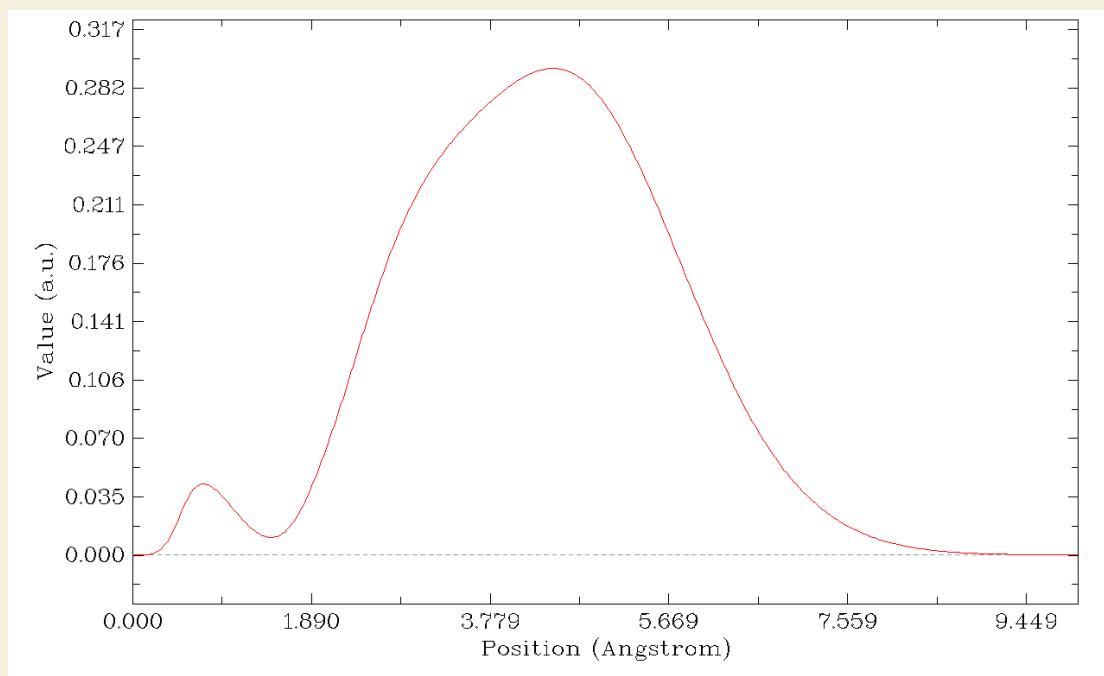
The main distribution region of both of them are far from the molecule. MO10 is almost spherically symmetric, thus is can be denoted as *s* type of Rydberg orbital. While MO11 has two phases, they equally distribute at the two sides, the overall shape is very close to atomic *p* orbitals, therefore MO11 can be identified as *p* type of Rydberg orbital.

How to quantitatively demonstrate that the main distribution region of these Rydberg orbitals is far from the molecular center? One of the best way is plotting RDF of electron density corresponding to these orbitals. Here we plot this kind of RDF map for MO11. We close the GUI of main function 0, and then input below commands:

```

6 // Modify wavefunction
26 // Modify orbital occupation number
0 // Select all orbitals
0 // Select occupation number of all orbitals to zero
11 // Select orbital 11
2 // Set occupation number of orbital 11 to 2.0 (assume it is doubly occupied)
q // Return
-1 // Return to main menu
200
5 // Plot RDF
3 // Set lower and upper limit of radial plotting
0,10 // From 0 to 10 Å
4 // Set angular number of integration points. The default value is unnecessarily high for
present purpose, therefore we set it to a smaller value to reduce computational time
302 // 302 angular points
0 // Calculate RDF for electron density (which is the default real space function)
1 // Plot the RDF map

```



From the map it can be seen that the global maximum peak is at about 4.6 Å, indicating that the major distribution region of this orbital wavefunction is very far from origin (in present .fch file, Cartesian origin corresponds to molecular center), and thus MO11 can be unambiguously identified as a Rydberg orbital.

Please also plot such a RDF map for a regular valence virtual MO, *e.g.* MO9, which is π^* orbital. Where is its peak position?

4.200.7 Parse output file of “polar” task of Gaussian to obtain (hyper)polarizability and related quantities

As introduced in Section 3.200.7, Multiwfn has ability to parse the abstract “polar” task output file of Gaussian and reorganize the data to a much more readable format, and meantime many useful quantites are outputted together. In this section an example is given.

In "example\polar" folder, *NH3_polar_static.out* and *NH3_polar_dynamic.out* are output files of static and frequency-dependent (hyper)polarizability calculations for NH₃ via a normal exchange-correlation functional PBE0. The corresponding Gaussian input files are also provided in the same folder. In the calculation, a large basis set with abundant diffuse functions (aug-cc-pVTZ) was used, notice that diffuse functions play a vital role in yielding reasonable (hyper)polarizability and thus they are absolutely indispensable.

First, we use Multiwfn to parse data from the *NH3_polar_static.out*. Boot up Multiwfn and input

```
examples\polar\NH3_polar_static.out
200 // Main function 200
7 // Parse (hyper)polarizability task of Gaussian
1 // Start parsing. We select option 1 because third-order derivative of current level (normal
DFT functional) is supported by Gaussian
```

Now you can see below information on screen:

Dipole moment:

```
X,Y,Z= 0.000000 0.000000 -0.596558 Norm= 0.596558
```

Static polarizability:

```
XX= 13.692700
XY= 0.000000
YY= 13.693300
XZ= 0.000000
YZ= -0.000334
ZZ= 15.323600
```

Isotropic average polarizability: 14.236533

Isotropic average polarizability volume: 2.109636 Angstrom^3

Polarizability anisotropy (definition 1): 1.630600

Polarizability anisotropy (definition 2): 1.630600

Eigenvalues of polarizability tensor: 13.692700 13.693300 15.323600

Polarizability anisotropy (definition 3): 1.630600

Note: It is well known that the sign of hyperpolarizability of Gaussian 09/16 should be inverted, the outputs shown below have already been corrected

Static first hyperpolarizability:

```
XXX= 0.000000
XXY= 12.903200
XYY= 0.000000
YYY= -12.900200
XXZ= 9.617100
XYZ= 0.000000
YYZ= 9.564710
XZZ= 0.000000
YZZ= 0.016285
ZZZ= 26.613200
```

Beta_X= 0.00000 Beta_Y= 0.01929 Beta_Z= 45.79501

Magnitude of first hyperpolarizability: 45.795014

Projection of beta on dipole moment: -45.795010

Beta || : -27.477006

Beta ||(z) : 27.477006

Beta _||_(z) : 9.159002

The output is easy to understand, all of the outputted quantities have been detailedly explained in Section 3.200.7.

Next, we extract frequency-dependent (hyper)polarizability from the *NH3_polar_dynamic.out*. Boot up Multiwfn and input

examples\polar\NH3_polar_dynamic.out

```
200 // Main function 200
7 // Parse (hyper)polarizability task of Gaussian
-1 // Let Multiwfn try to load frequency-dependent (hyper)polarizability
1 // Start parsing
```

Multiwfn detected there are three set of data:

```
1 w= 0.000000 ( static )
2 w= 0.065600 ( 695.04nm )
3 w= 0.071900 ( 634.14nm )
```

If we input 2, then the (hyper)polarizability data corresponding to 0.0656 a.u. incident light will be loaded and parsed.

After outputting information about dipole moment and polarizability, Multiwfn asks you to choose the type of hyperpolarizability to be outputted:

```
1: Beta(-w;w,0) 2: Beta(-2w;w,w)
Note: Option 2 is meaningless if "DCSHG" keyword was not used
```

You can select either one according to your requirement. Since as can be seen in *NH3_polar_dynamic.gif*, the keyword we used is *polar=DCSHG*, both the two types are available.

4.200.8 Calculate polarizability and hyperpolarizability for NH₃ via sum-over-states (SOS) method

In this example I will show how to use Multiwfn to calculate polarizability and hyperpolarizability based on sum-over-states (SOS) method for HF molecule. Please make sure that you have read Section 3.200.8.

As introduced in Section 3.200.8, SOS calculation requires information of a large amount of electronic states, including electric dipole moments, excitation energies, and transition dipole moments between these states. Commonly these information can be obtained by ZINDO, CIS, TDHF and TDDFT calculations. In this example we use the very popular CIS method. According to my experiences, the more expensive methods TDHF and TDDFT, which can produce more accurate excitation energies, do not necessarily give rise to better (hyper)polarizability than CIS when used in combination with SOS technique.

In this example we use Gaussian program to carry out the CIS calculation. However, Gaussian itself cannot output enough information for SOS calculation. Though for CIS (and ZINDO) there is a keyword *alltransitiondensities*, which makes Gaussian output transition density moment between each pair of excited states, however electric dipole moment of all states still cannot be obtained in a single run. Fortunately, we can use the very powerful electron excitation analysis module of Multiwfn to generate all information needed by SOS based on the output file of CIS/TDHF/TDDFT task of Gaussian or ORCA program.

First, run *examples\NH3_SOS.gjf* by Gaussian to produce output file *NH3_SOS.out*, and use formchk utility to convert the checkpoint file to *NH3_SOS.fch*. If you do not have Gaussian in hand, you can directly download them at http://sobereva.com/multiwfn/extrafiles/NH3_SOS.rar.

Since the SOS results converge often slow with respect to the number of excited states taken into account, we produce as high as 150 excited states in this example to substantially avoid truncation error. Of course, employing higher number of excited states needs more computational time in both of the CIS calculation and the subsequent SOS calculation in Multiwfn. In most practical studies, 100 states is generally large enough, and even 70 is often

enough to provide usable results. Calculation of hyperpolarizability, especially the high-order ones, has very stringent requirements on the quality of basis set, abundant diffuse functions are absolutely indispensable. In this example we employ def2-TZVPPD (*J. Chem. Phys.*, **133**, 134105), which is a high-quality basis set optimized for calculation of molecular response properties. Since this is not a built-in basis set in current version of Gaussian, it was picked from EMSL website (<https://bse.pnl.gov/bse/portal>). The keyword *IOP(9/40=5)* is important, because in the ZINDO/CIS/TDHF/TDDFT task by default Gaussian only outputs the transition coefficients whose absolute values are larger than 0.1, while *IOP(9/40=5)* lowers the criterion to 0.00001, so that much more coefficients can be outputted, and thereby we can obtain accurate transition dipole moments by Multiwfn at next step. Worthnotingly, you can also use TDHF or TDDFT instead of CIS, for example you can write *#P TD(nstates=150) CAM-B3LYP/gen IOP(9/40=5)*.

Boot up Multiwfn and input below commands

```
C:\ NH3_SOS.fch
18 // Electron excitation analysis module
5 // Calculate transition dipole moments between all excited states
C:\ NH3_SOS.out
3
```

The file *SOS.txt* generated in current folder contains all information needed by SOS (hyper)polarizability calculation. This file can be directly used by SOS module of Multiwfn.

Reboot Multiwfn and input

```
SOS.txt
200 // Other function (Part 2)
8 // Calculate (hyper)polarizability by sum-over-states (SOS) method
```

Note that all units used in the SOS module are atomic units.

Select 1 and input 0 to calculate static polarizability $\alpha(0;0)$ first, the result is

	1	2	3
1	14.610687	-0.000000	0.000000
2	-0.000000	14.610687	0.000000
3	0.000000	0.000000	14.552482

```
Isotropic average polarizability:      14.591285
Isotropic average polarizability volume:   2.162205 Angstrom^3
Polarizability anisotropy (definition 1):    0.058205
Eigenvalues:      14.552482    14.610687    14.610687
Polarizability anisotropy (definition 2):    0.029103
```

As can be seen, not only the polarizability tensor, but also some related quantities are outputted. Their definitions can be found in Section 3.100.20. The isotropic average polarizability we obtained here is 14.59, which is in perfect agreement with the experimentally determined value 14.56! (*Mol. Phys.*, **33**, 1155)

Then we calculate dynamic polarizability $\alpha(-\omega;\omega)$ and assume the frequency of external field to be 0.0719 a.u. Select option 1 again, input 0.0719, from the output we can see the dynamic isotropic polarizability at $\omega=0.0719$ is 14.86, which is slightly larger than the static counterpart.

Next, we calculate first hyperpolarizability and consider the static case $\beta(0;0,0)$. Select option 2 and input 0,0. Only the β component along the dipole moment direction, namely $\beta_{||}$, is what we are particularly interested in, since only this quantity can be determined experimentally. From the output we find $\beta_{||}(0;0,0)$ is -38.98. The corresponding experimental value is not available, however this value is close to the highly accurate value calculated by CCSD(T) method (-34.3, see *J. Chem. Phys.*, **98**, 3022).

Now we calculate $\beta(-2\omega;\omega,\omega)$ at $\omega=0.0656$ a.u. Select option 2 again and input 0.0656, 0.0656 to set the frequency of both two external fields as 0.0656 a.u. This time the $\beta_{||}$ value is -49.69, which

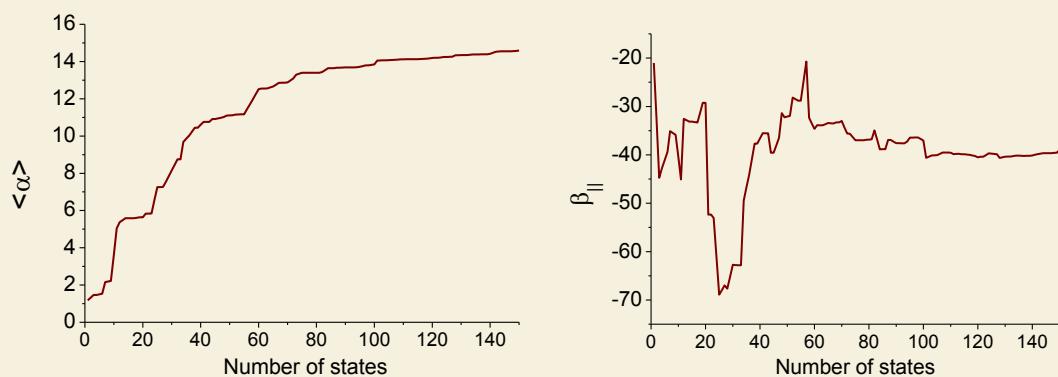
is again in excellent agreement with the experimentally determined value -48.9 ± 1.2 (see A. Hernández-Laguna et al. (eds.), *Quantum Systems in Chemistry and Physics*, Vol. 1, p111)

Note: Although in this example the agreement between our SOS/CIS calculations and the reference values is surprisingly good, this is not always hold for other systems. The SOS/CIS method sometimes severely overestimates β value!

Finally, we tentatively calculate second hyperpolarizability γ . Select 3 and input 0,0,0 to assume static electric fields. Since calculation of γ is evidently more time-consuming than β , Multiwfn does not automatically use all states but allow you to set the number of states to be taken into account. Larger value in principle gives rise to better result, but of course more time will be consumed in the calculation. Here, we input 150 to use all states. After a while, result is shown on screen, the average of γ is 928.74. Beware that this value may be inaccurate (reference value is not available, so I am not sure if this is a good result), one of the main reasons is that the basis set we used in the electron excitation calculation is not large enough. Accurate calculation of γ usually requests a basis set like d-aug-cc-pVTZ (or a even better one), which has an additional shell of diffuse functions compared to the commonly used aug-cc-pVTZ.

Multiwfn is also capable of calculating third hyperpolarizability $\delta(-\omega; \omega_1, \omega_2, \omega_3, \omega_4)$ where $\omega = \omega_1 + \omega_2 + \omega_3 + \omega_4$, but we do not do this in present example, because this quantity is fairly unimportant, and the calculation is terribly expensive when the number of states in consideration is large; moreover, a sky-high quality of basis set must be employed in the calculation...

A very important point in SOS calculation is that the number of states used must be high enough; in other words, if n states are involved in your SOS studies, the variation of the (hyper)polarizability with respect to the number of states have to be converged before n , otherwise n must be enlarged. By using Multiwfn we can readily examine if the convergence condition is satisfied. Here we check the convergence of static β . Select option 6 and input 0,0, after a while, the result is exported to "beta_n.txt" in current folder, the meaning of each column is clearly indicated on screen. Use your favourite tool to plot the data in "beta_n.txt". If you are an Origin user, you can directly drag this file into Origin window and plot the data as curve maps. You can also similarly use option 5 and 7 to study the convergence of α and γ . Below graphs show the variation of static $\beta_{||}$ as well as static isotropic average polarizability $\langle\alpha\rangle$ with respect to the number of states.



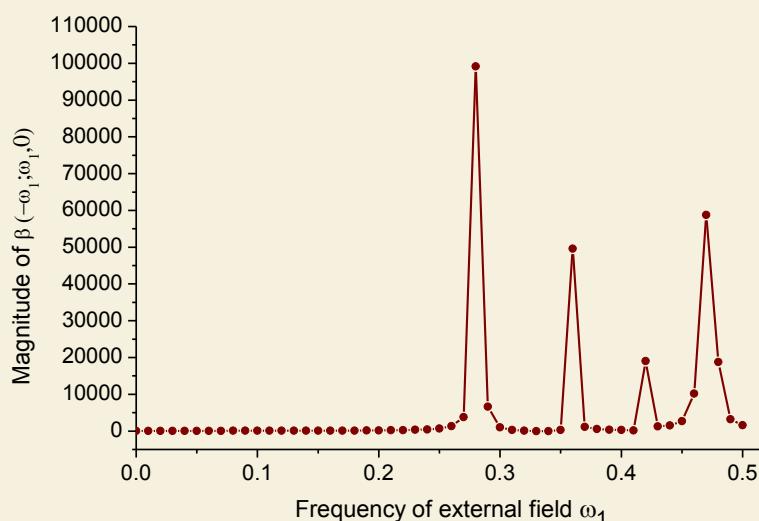
It is clear that both of the two quantities have basically converged at $n=100$. Since we employed 150 states in our calculations, the error due to the truncation of states can be safely ignored. From the graph one can also see that if the number of states is truncated at 70, the results are still qualitatively correct.

In the SOS module of Multiwfn one can also easily study the variation of dynamic (hyper)polarizability with respect to the frequency of external fields. For example, we investigate the variation of $\beta(-\omega_1; \omega_1, 0)$ as ω_1 varies from 0 to 0.5 a.u. with stepsize of 0.01. Write a plain text file, each row corresponds to a pair of ω_1 , ω_2 (ω_2 is fixed at zero in this example), for example

```
0.00 0
0.01 0
0.02 0
...
0.5 0
```

Tips: For convenience, you can utilize Microsoft Excel program to generate frequency list, and save the table as .txt file (you can select such as "Text (tab delimited)", but do not choose "Unicode text").

Then choose option 16, input the path of the plain text file, the β will be calculated at each pair of frequencies, the result will be outputted to "beta_w.txt" in current folder; the meaning of each column of the data is clearly indicated on screen. The data can be directly plotted as curve map by Origin, like below:



4.200.12 Calculate energy index (EI) and bond polarity index (BPI)

In this section I will illustrate how to calculate EI and BPI indices, which were defined in *J. Phys. Chem.*, **94**, 5602 (1990). If you are not familiar with these two quantities, please read Section 3.200.12 first. The geometry and wavefunction involved in this example were produced at HF/6-31G* level, which is the one used in above mentioned paper.

We will calculate BPI for C-N bond of CH₃NH₂, before this we first need to calculate reference EI value for C and N atoms, which correspond to EI of C in ethane and N in H₂N-NH₂, respectively. Boot up Multiwfn and input

```
examples\EI_BPI\ethane.fch
200
12 // Calculate energy index (EI) or bond polarity index (BPI)
1 // C1 atom
```

You will see the EI value for C in reference molecule ethane is -0.667639 a.u. Reboot Multiwfn and

input

examples\EI_BPI\N2H4.fch

200

12

1 // N1 atom

You can see the EI value for N in reference molecule H₂N-NH₂ is -0.718126 a.u.

Next we calculate EI for C and N in CH₃NH₂. Reboot Multiwfn and input

examples\EI_BPI\CH3NH2.fch

200

12

1 // C1, the result is -0.693374 a.u.

5 // N5, the result is -0.698092 a.u.

The BPI_{CN} in CH₃NH₂.is computed as

$$\begin{aligned} \text{BPI}_{\text{CN}} &= (\text{EI}_C - \text{EI}_C^{\text{ref}}) - (\text{EI}_N - \text{EI}_N^{\text{ref}}) \\ &= -0.693374 + 0.667639 + 0.698092 - 0.718126 \\ &= -0.046 \end{aligned}$$

As a comparison, use *examples\EI_BPI\F2.fch* to calculate reference value for F, the result should be -0.992542 a.u., and use *examples\EI_BPI\CH3F.fch* to calculate EI for C and F in CH₃F molecule, the result should be -0.750302 and -0.885961, respectively. Then compute the BPI_{CF} value as -0.750302+0.667639+0.885961-0.992542= -0.189. Since BPI_{CF} in CH₃F is evidently more negative than BPI_{CN} in CH₃NH₂, it can be concluded that C-F bond in CH₃F is more polar than C-N bond in CH₃NH₂.

Via EI index we can also evaluate the so-called group electronegativity, which is often more useful than atomic electronegativity. Here we calculate electronegativity for -CH₃ group, which is simply the negative of EI_C for CH₃ radical. Boot up Multiwfn and input

examples\EI_BPI\CH3.fch // Optimized and produced at UHF/6-31G*

200

12

1 // Carbon atom

The result is -0.630656 a.u., corresponding to electronegativity of CH₃ group of 0.631. Then we use *examples\EI_BPI\F.fch* to calculate group electronegativity for -F, the result is 0.957. It is clear that -F group has much higher electronegativity, and thus has stronger capacity to attract electrons than -CH₃ group due to its lower average energy per valence electron.

4.200.14.1 Integrate real space functions within reduced density gradient (RDG) isosurface to study weak interaction quantitatively

Before reading this, please read Section 3.23.1 to understand how to use reduced density gradient (RDG) to reveal weak interaction regions. In this section, I show the possibility of characterizing weak interaction by integrating domains enclosed by RDG isosurfaces.

System 1: Phenol dimer

First, we use phenol dimer as example. Boot up Multiwfn and input

examples\phenoldimer.wfn

200 // Main function 200

14 // Integrate real space functions within isosurfaces of a real space function

Here we want to study RDG domains defined as regions enclosed by isosurface of RDG=0.5; in other words, these domains composed of grid points where RDG<0.5. Therefore, we select option 2 and choose "13 Reduced density gradient", and then select option 3 and input criterion, namely <0.5 (In fact, RDG<0.5 is the default setting and you do not need to manually do these steps). Next, input below commands:

1 // Start calculation grid data and generate domains

-10 // Adjust extension distance

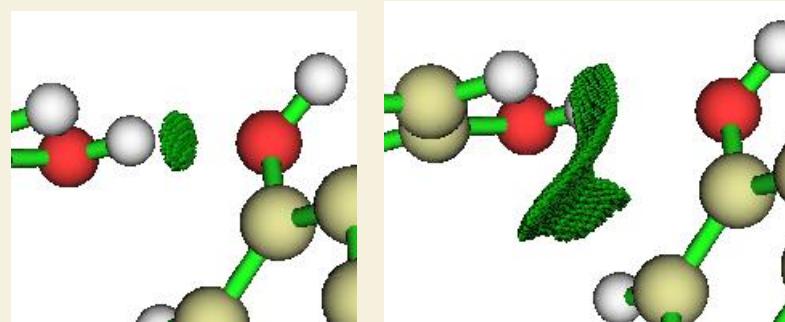
0 // Set extension distance to zero to avoid wasting of grid points at boundary area, where RDG isosurfaces commonly do not occur

2 // Medium quality grid (i.e. grid spacing=0.1 Bohr), generally this is accurate enough

Now Multiwfn starts calculation of grid data for the selected real space function (*i.e.* RDG), and then identifies individual RDG domains according to the criterion of RDG<0.5. Finally, four domains are found, the number of grid points constituting the domains are shown as the last column:

Domain:	1	Grids:	208
Domain:	2	Grids:	290
Domain:	3	Grids:	200
Domain:	4	Grids:	2597

To visualize them, select "3 Visualize domains". In the GUI, you can select domain index at right-bottom list. The 2nd and 4th domains are shown below:



If you have read Section 3.23.1, you must know these domains correspond to H-bond and van der waals (vdW) interactions between the two phenols, respectively. We can study properties of these domains in terms of integrating specific real space functions within corresponding regions. We select option 1 and input 2 (domain index), then input 1 to select electron density as the function to be integrated, the result is:

```
Integration result: 0.6932973049E-02 a.u.
volume: 0.290000 Bohr^3 ( 0.042974 Angstrom^3 )
Average: 0.2390680362E-01
Maximum: 0.2766231164E-01 Minimum: 0.1994457517E-01
```

Similarly, we do this for domain 4:

```
Integration result: 0.1181667175E-01 a.u.
volume: 2.597000 Bohr^3 ( 0.384836 Angstrom^3 )
```

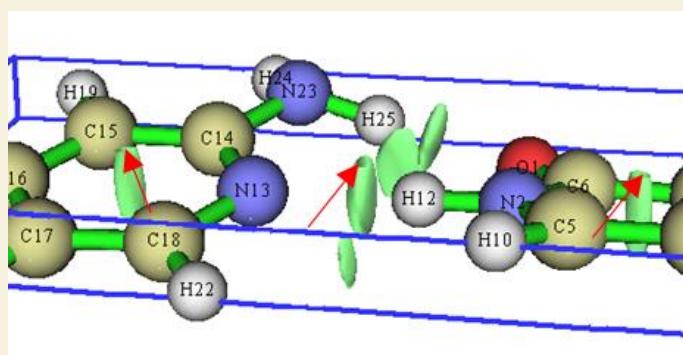
Average: 0.4550123895E-02
 Maximum: 0.6217203920E-02 Minimum: 0.3102088666E-02

From the output we know the number of electrons involved in the domains corresponding to H-bond and vdW interactions are 0.006933 and 0.011817, respectively. They can be interpreted as overlapping electrons and are closely related to strength of same type of interactions (see discussion in Section 5.2 of DORI original paper *J. Chem. Theory Comput.*, **10**, 3745 (2014)). However, since these two domains correspond to different type of weak interactions, the magnitude of overlapping electrons is not positively correlated to their strengths, namely we are unable thus to say that the vdW interaction between the two phenols is stronger than the intermolecular H-bond. The "Volume" in the output denotes volume of the domain, we can find that vdW interaction involves much wider spatial region than H-bond. "Average" correspond to average value of real space function in the domain, from this quantity one can easily infer that the strength of interaction per contact region of H-bond must be significantly higher than that of vdW interaction, since as shown above, their average values are 0.0239 and 0.0045, respectively, the former is much larger than the latter.

System 2: 2-pyridoxine 2-aminopyridine

Intermolecular H-bonds of 2-pyridoxine 2-aminopyridine (PP) has been investigated in Section 4.2.1 by means of AIM analysis, while this time we will analyze them by means of q_{int} index. This index was proposed in *J. Phys. Chem. A*, **115**, 12983 (2011) for judging interaction of H-bond at various intermolecular distance, please check Section 3.200.14 for its definition. Commonly, the more negative of the q_{int} index, the more stable the interaction.

q_{int} index is defined based on integrating domains enclosed by RDG=0.6 isosurfaces, therefore we need to first calculate RDG grid data and generate corresponding domains. Note that it is not always appropriate to set extension distance of RDG grid data to zero. For present system, if you calculate RDG grid data by main function 5 with extension distance of zero, you will see some RDG isosurfaces are truncated by box boundary, as shown below and highlighted by red arrows. In this case domain integrating module of Multiwfn does not work.



Therefore, when we calculate RDG grid data for this case, extension distance should be set somewhat larger than zero, 3 Bohr is safe enough for avoiding unexpected truncation. Extension distance should also never be set to a too large value, otherwise the number of grid points to be calculated will be very high and thus very time-consuming.

Boot up Multiwfn and input below commands:

examples\2-pyridoxine_2-aminopyridine.wfn

200

14

```
3 // Change the default criterion of defining domain
```

```
<0.6
```

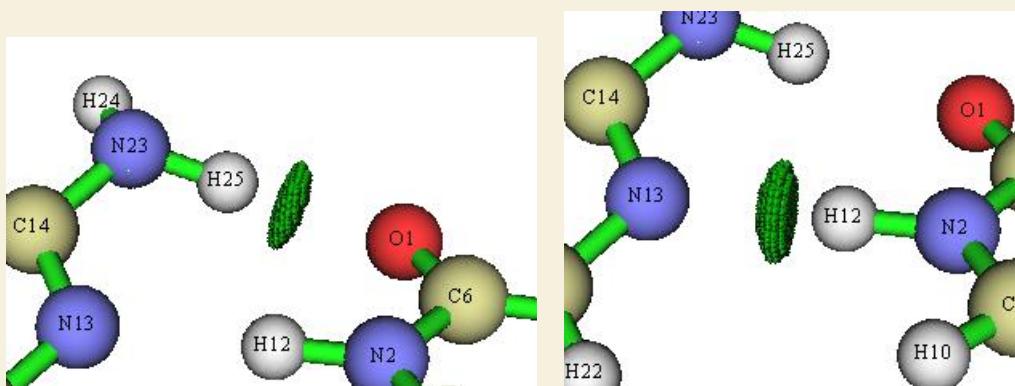
```
1 // Start calculation of grid data
```

```
-10 // Change extension distance
```

```
3 // 3.0 Bohr of extension distance
```

```
2 // Medium quality grid
```

Now visualize resulting domains. Domain 2 and 4 are shown below, clearly they correspond to H-bond of N23-H25.....O1 and N2-H12.....N13, respectively.



Now select "5 Calculate q_{bind} index for a domain" and input 2, the resulting q_{int} and related details of the domain are shown below

```
q_att:      0.00490957 a.u.
q_rep:      0.00005590 a.u.
q_bind:     -0.00485367 a.u.

volume (lambda2<0):    0.599000 Bohr^3
volume (lambda2>0):    0.010000 Bohr^3
volume (Total):        0.609000 Bohr^3

Similarly, we obtain results for domain 4

q_att:      0.00805204 a.u.
q_rep:      0.00037346 a.u.
q_bind:     -0.00767858 a.u.

volume (lambda2<0):    0.745000 Bohr^3
volume (lambda2>0):    0.052000 Bohr^3
volume (Total):        0.797000 Bohr^3
```

Since the q_{bind} corresponding to N2-H12.....N13 (-0.007678) is much more negative than that of N23-H25.....O1 (-0.004854), the former should be stronger than the latter.

Since both the interactions are H-bond, it is also possible to simply compare the number of electrons contained in the domains to estimate their relative strength. We choose "2 Perform integration for all domains" and then select electron density to print out integral of electron density in all domains:

Domain	Integral (a.u.)	Volume (Bohr^3)	Average
1	0.1212147834E-03	0.111000	0.1092025076E-02
2	0.1647945811E-01	0.609000	0.2705986553E-01
3	0.1714432107E-02	0.319000	0.5374395319E-02
4	0.2618188584E-01	0.797000	0.3285054685E-01

```

5   0.8932587581E-02      0.412000  0.2168103782E-01
6   0.7764521413E-02      0.394000  0.1970690714E-01
Integration result of all domains:  0.6119409984E-01 a.u.
volume of all domains:  2.642000 Bohr^3      0.391504 Angstrom^3

```

Not only the integration value of domain 2 (0.01648) is evidently smaller than domain 4 (0.02618), but also the average value of domain 2 (0.02706) is smaller than domain 4 (0.03285), therefore we have strong evidence to say N2-H12.....N13 is stronger than N23-H25.....O1.

With similar steps illustrated in this section, you can also integrate other real space functions such as potential energy density and spin density in the domains enclosed by isosurfaces of other real space functions, e.g. DORI and ELF.

Note that the accuracy of integration in the domains is directly determined by grid setting, higher quality of grid leads to better accuracy. For example, when visualizing domains, if you found a domain only consists of very few grid points, and its profile is very coarse, then integration accuracy of this domain must be very low.

4.200.14.2 Visualize molecular cavity and calculate its volume by domain analysis module

This is an application instance of domain analysis module, I will briefly illustrate how to use Multiwfn to visualize molecular cavity and calculate cavity volume. More information about this topic can be found in my blog article <http://sobereva.com/408> (In Chinese).

The idea of using domain analysis to study molecular cavity is very simple: We first calculate promolecular density, then we define the regions having electron density lower than a threshold *e.g.* 0.0001 as molecular cavity. More than one such regions may exist, the domain analysis module automatically assigns them with different domain indices. After that, by visualizing the domains, it should be easy to find the domain corresponding to the molecular cavity. The choice of threshold is somewhat arbitrary, commonly 0.001~0.0001 a.u. is appropriate.

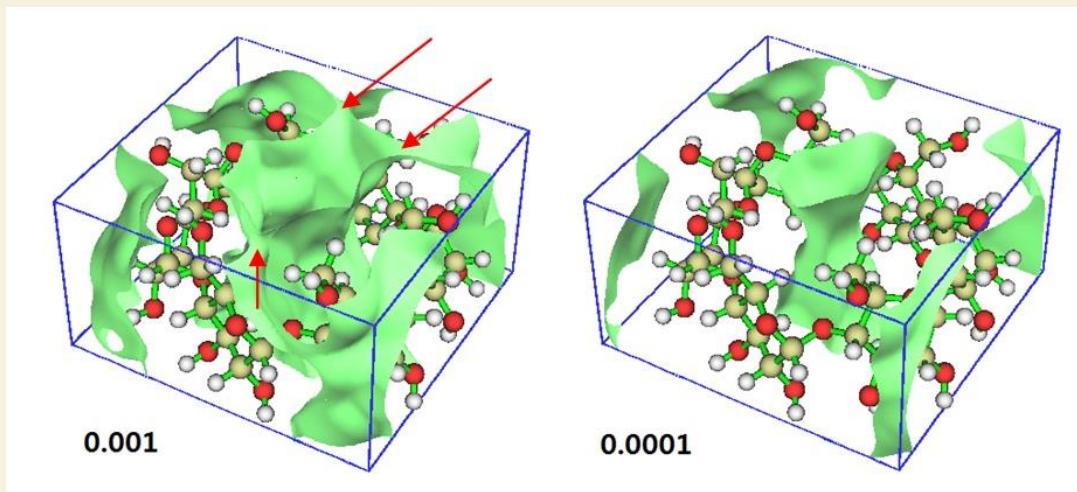
α -cyclodextrin is used as an example in this section. Before studying the cavity using domain analysis module, it is suggested to first visualize promolecular density under various isovalue. Boot up Multiwfn and input

```

examples\alpha-cyclodextrin.pdb
5 // Calculate grid data
1 // Promolecular density
-10
0 // Set extension distance to zero, namely let the box just enclose the molecule
1 // Low quality grid
-1 // Show isosurface map

```

Click "Show data range" in the GUI window to show the box of grid data as blue frame, and set isovalue to 0.01 and 0.001 respectively, you will see



It is easy to understand, if we use threshold of 0.001 a.u., then the domain corresponding to the cavity in the center of the molecule cannot be defined, because the internal region and external region are connected via the three channels pointed by red arrows. While in the case of 0.0001 a.u., the molecular cavity is clearly identifiable and thus we could use domain analysis module with this threshold to study the cavity.

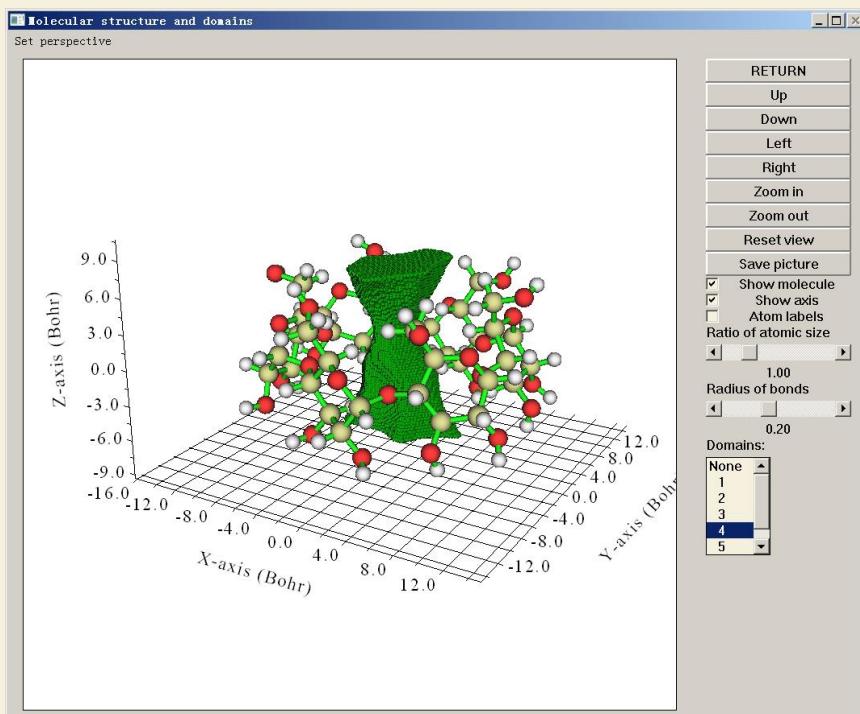
Return to main menu, and then input below commands

```
200 // Other functions (part 2)
14 // Domain analysis
2 // Choose the real space function to be calculated and used for partitioning domains
1 // Promolecular density
3 // Define the rule of determining domains
<0.0001 // Regions with electron density less than 0.0001 will be defined as domains
1 // Calculate grid data and assign domains
-10 // Change extension distance
0 // No extension distance
1 // Low quality grid
```

After calculation is finished, you will see below information from screen. There are totally six domains found, the number of grids in each domain is also shown

Domain:	1	Grids:	4
Domain:	2	Grids:	12010
Domain:	3	Grids:	5822
Domain:	4	Grids:	39654
Domain:	5	Grids:	13949
Domain:	6	Grids:	7438

Then we can visualize each domain by option 3. After examining each domain, we find that the domain with index of 5 is the one corresponding to molecular cavity, as shown below:



Since this domain well represents shape of actual molecular cavity, its volume is a good indicator of cavity size.

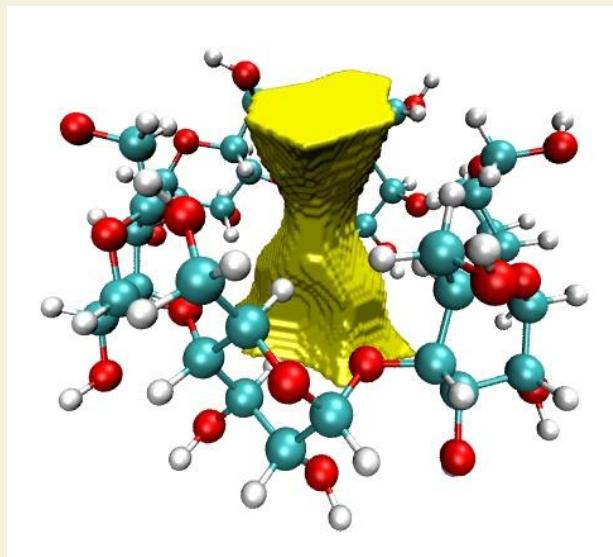
Close the GUI, select option 1 to carry out integration within domain, and input 4 to choose to integrate the domain corresponding to the cavity. Since currently we are only interested in the volume of the domain, we choose "100 User-defined real space function" as integrand (By default, this function is 1.0 everywhere and thus does not take any computational cost), then below information are printed on screen

```
Integration result: 0.3172320000E+03 a.u.
Volume: 317.232000 Bohr^3 ( 47.008943 Angstrom^3 )
Average: 0.1000000000E+01
Maximum: 0.1000000000E+01 Minimum: 0.1000000000E+01
```

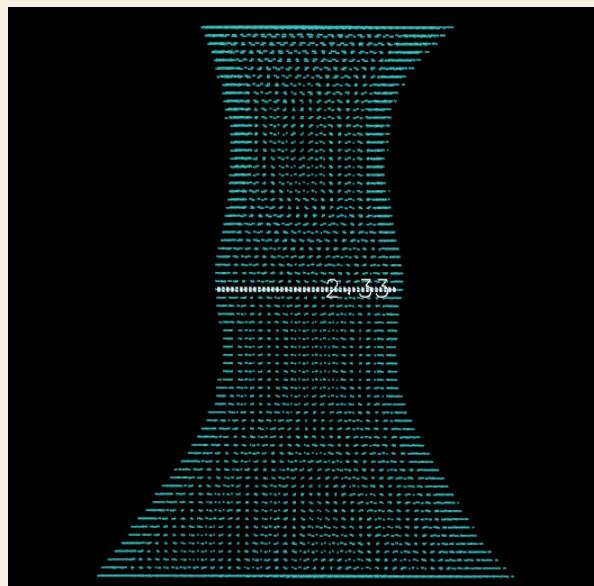
```
Position statistics for coordinates of domain points (Angstrom):
X minimum: -2.7823 X maximum: 3.1445 Span: 5.9268
Y minimum: -2.6823 Y maximum: 2.6095 Span: 5.2918
Z minimum: -3.1770 Z maximum: 3.9140 Span: 7.0910
```

From the output we find the volume of this domain, namely the size of the cavity, is 47.0 \AA^3 . The length of the cavity is 7.09 \AA (*i.e.* the span distance between the domain point with maximum Z and that with minimum Z coordinate).

Furthermore, we can export selected domain as domain.cub in current folder by using option 10, so that the domain can be portrayed as isosurface in third-part visualization tools, such as VMD. In the resulting domain.cub, the grids within the selected domain have value of 1, while grids in other regions (as well as boundary grids) have value of 0, therefore isosurface of domain can be rendered with isovalue between 0 and 1 (commonly 0.5 is used). We do this for the 4th domain and plot is as isosurface in VMD, below graph will be obtained



Alternatively, we can use option 11 to export boundary grids of selected domain to domain.pdb, in which each particle corresponds to a boundary grid. You can load this file into VMD and render the particles as spheres. Then if you want to measure domain, you can click keyboard button 2, then click two spheres in the graphical window, the linking line and distance between the two selected spheres will be shown, as illustrated below ("Display" - "Orthographic" was selected for easier inspection)

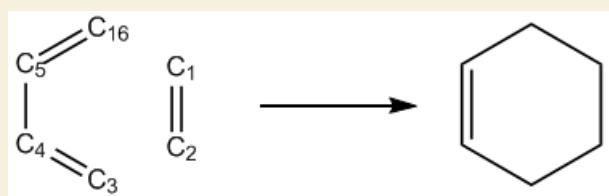


4.A Special topics and advanced tutorials

The contents in this section involve more than one main functions of Multiwfn, or contain special usages.

4.A.1 Study variation of electronic structure along IRC path

In this tutorial, I will briefly show you how to use Multiwfn to study variation of electronic structure along the IRC path of Diels-Alder adduction. We will study the variation of Mayer bond order, and will animate the deformation of ELF isosurface. With the similar fashion you can also easily investigate variation of other properties, such as atomic charges, electron density, aromaticity and so on.



Gaussian 09 A.02 was used throughout this tutorial. Unless otherwise specified, all calculations will be performed under Windows 7 64bit system. In this tutorial the files marked by crimson can be found in "examples\IRC" or "examples" folder.

Before starting this tutorial, you should setup running environment for Gaussian first, otherwise Gaussian cannot be properly invoked in Windows environment. The setup method is: Enter “control panel”-“System properties”-“Advanced”, click “Environment variables” button, then click “New” button in “User variables” frame, input GAUSS_EXEDIR as variable name, input the install directory of Gaussian as variable value (e.g. D:\study\g09w\, assuming that g09.exe is in this folder). After that modify "PATH" environment variable to add the install directory of Gaussian into it.

1 Perform IRC calculation

Run **DA_IRC.gjf** by Gaussian to produce **DA_IRC.out**. We will find this IRC path actually contains 18 and 13 points in the two directions, respectively. B3LYP/6-31+G* is used in this calculation.

2 Generate wavefunction file for each point of IRC

Write an input file of single point task of Gaussian (**DA_SP.gjf**), which will be used as "template" later. The geometry in fact can be arbitrarily filled.

DO NOT write anything here (e.g. %chk)

```
#p B3LYP/6-31G* nosymm
```

```
DA adduction
```

```
0 1
C -0.26156800 1.56679300 0.69509600
```

C	-0.26156800	1.56679300	-0.69509600
C	0.50031400	-0.43279300	-1.43864500
C	-0.26156800	-1.32826500	-0.70392000
C	-0.26156800	-1.32826500	0.70392000
H	-1.19341600	1.44689700	-1.23752300
H	-1.19341600	1.44689700	1.23752300
H	0.52507600	2.08588200	1.23621300
H	0.38154400	-0.37781200	2.51847500
H	0.38154400	-0.37781200	-2.51847500
H	1.46467600	-0.09643600	-1.07409700
H	-1.04094400	-1.89294000	-1.21418700
H	-1.04094400	-1.89294000	1.21418700
H	1.46467600	-0.09643600	1.07409700
H	0.52507600	2.08588200	-1.23621300
C	0.50031400	-0.43279300	1.43864500

blank line**blank line**

Notice that the basis set we used here (6-31G*) is different to the one used in IRC task (6-31+G*), because Mayer bond order does not work well when diffuse functions are presented. By the way, ignoring diffuse functions will not lead to detectable change of ELF isosurface. Also note that the "nosymm" keyword is specified, because if we do not do this Gaussian will automatically translate and rotate the molecule to put it to standard orientation, which may leads to discontinuity problem in the animation of ELF (You will see molecule suddenly jumps in certain frames of the animation).

IRCSplit.exe is a tool used to produce .wfn/.chk file for each point of IRC and SCAN tasks of Gaussian 09, **IRCSplit.f90** is the corresponding source code, by which you can compile Linux version of IRCSplit. Boot up IRCSplit.exe by double click its icon and then input

```
DA_IRC.out //The file of the output file of the IRC task
DA_SP.gjf //The template file used to generate single point input files
2 //Only yield .chk files
C:\DA_IRCchk\DA //The path and prefix of the finally generated .chk files
18,13 // The program detected that in DA_IRC.out there are 18 and 13 points in the two directions of IRC, respectively. Here we extract all of them, together with the TS point
```

Now you can find DA_SP0001.gjf, DA_SP0002.gjf ... DA_SP0032.gjf in current folder. Please manually check one of them to verify the reasonableness of these input files. Note that DA_SP0014.gjf corresponds to the TS geometry.

Build a new folder "C:\DA_IRCchk" and copy the .gjf files as well as the script **runall.bat** into it. Double clicking the icon of runall.bat, which will invoke Gaussian 09 to run all of the .gjf files.

Now you have DA0001.chk, DA0002.chk ... DA0032.chk in "C:\DA_IRCchk" folder. Copy **chk2fch.bat** to this folder and run it, then the formchk utility in Gaussian package will be automatically invoked to convert all .chk files to .fch files.

3 Calculate Mayer bond orders for all IRC points

Mayer bond order of C1-C16 is the one we are particularly interested in, whose formation is the key process of the DA adduction. Since by default Multiwfn only outputs Mayer bond orders with value > 0.05 , while C1-C16 must be very weak at the initial stage of DA adduction, we need to set "bndordthres" parameter in the settings.ini file in Multiwfn folder to 0.0, so that all of the bond orders larger than 0.0 can be outputted.

Write a plain text file (**MBObatch.txt**) and put it into Multiwfn folder, the content is

```
9 // Enter bond order analysis module  
I // Calculate Mayer bond order
```

Note: If you are confused why this file is written in such manner, please read Section 5.2 to study how to run Multiwfn in silent mode.

Then write a plain text file with .bat suffix (**MBObatchrun.bat**) and put it into Multiwfn folder, the content should be

```
for /f %%i in ('dir C:\DA_IRCchk\*.fch /b') do Multiwfn C:\DA_IRCchk\%%i <  
MBObatch.txt > C:\DA_IRCchk\%%~ni.txt
```

batchrun.bat in fact is a Windows batch script. Double clicking its icon to run it, the .fch files in "C:\DA_IRCchk\" folder will be sequentially loaded into Multiwfn, and the calculated Mayer bond orders will be exported to .txt files in the same folder.

4 Plot Mayer bond order

Now what we should do next is to extract the bond order of C1-C16 from the DA0001.txt, DA0002.txt ... DA0032.txt. The most convenient way is to utilize "grep" command in Linux. So we copy all of these .txt files to a folder in Linux system, then in this folder we run

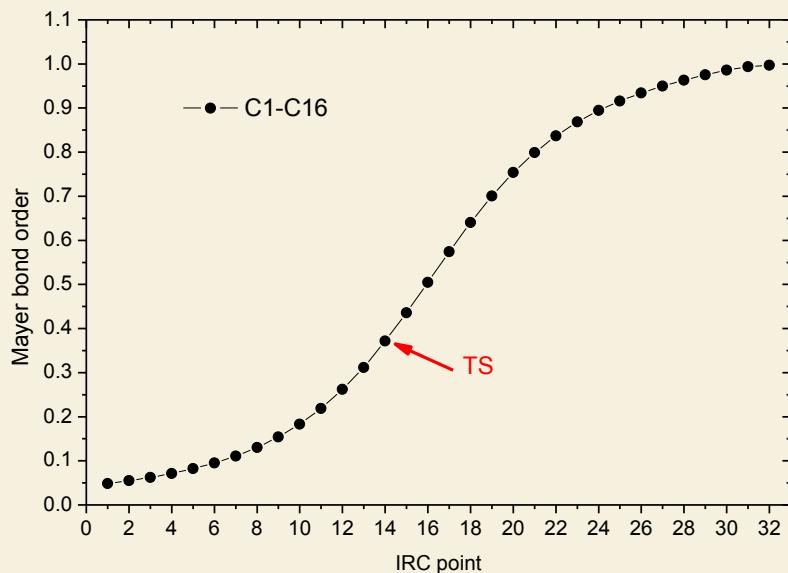
```
grep "1(C ) 16(C )" * > out.txt
```

The **out.txt** file now contains C1-C16 bond order of all points in the IRC:

```
DA0001.txt:# 9:      1(C ) 16(C ) 0.05929972  
DA0002.txt:# 7:      1(C ) 16(C ) 0.06877306  
DA0003.txt:# 7:      1(C ) 16(C ) 0.07926829  
DA0004.txt:# 7:      1(C ) 16(C ) 0.09089774  
DA0005.txt:# 7:      1(C ) 16(C ) 0.10380144  
DA0006.txt:# 7:      1(C ) 16(C ) 0.11815120  
DA0007.txt:# 7:      1(C ) 16(C ) 0.13417828  
DA0008.txt:# 7:      1(C ) 16(C ) 0.15218555
```

...

The last column is the values of Mayer bond order of C1-C16, you can plot them by your favorite program now, you will see

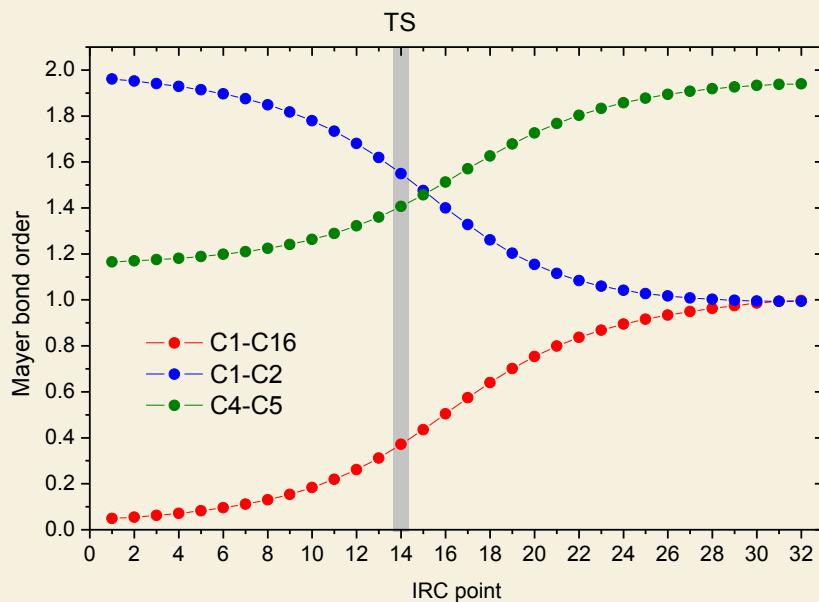


Clearly, C1-C16 become stronger and stronger as the reaction proceeds, its Mayer bond order gradually increases to 1.0 (typical single bond).

With the same method, we also calculate the Mayer bond order of C1-C2 and C4-C5, namely run below commands

```
grep "1(C )      2(C )" * > out2.txt
grep "4(C )      5(C )" * > out3.txt
```

Plot the data in **out.txt**, **out2.txt** and **out3.txt** together, you will see



This graph vividly shows that the C1-C2 smoothly becomes to a single bond from a double bond during the DA adduction, and the reaction increases the double-bond character of C4-C5 significantly.

5 Make animation of ELF isosurface

Next we make animation to study how the ELF isosurface varies during the DA adduction.

Create a plain text file ELFbatch.txt in Multiwfn folder with below content

```
5 // Generate grid data
9 // ELF
2 // Medium quality grid
2 // Export the grid data to ELF.cub in current folder
```

Create a script file named ELFbatchrun.bat, whose content is

```
for /f %%i in ('dir C:\DA_IRCchk\*.fch /b') do (
Multiwfn C:\DA_IRCchk\%%i < ELFbatch.txt
rename ELF.cub %%~ni.cub
)
```

Run **ELFbatchrun.bat**, Multiwfn will sequentially load the .fch files in "C:\DA_IRCchk" and export the corresponding ELF grid data to DA0001.cub, DA0002.cub ... DA0032.cub in current folder.

We use VMD 1.9.1 program (freely available at <http://www.ks.uiuc.edu/Research/vmd/>) to render isosurface for these cube files. Move all of the cube files to VMD folder, and create a plain text file named **isoall.tcl** in the VMD folder, the content is

```
set isoval 0.88
axes location off
color Display Background white
for {set i 1} {$i<=32} {incr i} {
set name DA[format %04d $i]
puts "Processing $name.cub..."
mol default style CPK
mol new $name.cub
#translate by -0.100000 0.20000 0.000000
#scale to 0.30
rotate y by 50
rotate z by 90
rotate x by -30
rotate y by -20
mol addrep top
mol modstyle 1 top Isosurface $isoval 0 0 0 1 1
mol modcolor 1 top ColorID 3
render snapshot $name.bmp
mol delete top
}
```

This file essentially is a VMD script, in which the command *set isoval 0.88* means the isosurface of 0.88 will be plotted, the default view point is adjusted by *scale*, *rotate* and *translate* commands. *for {set i 1} {\$i<=32} {incr i}* means the file from DA0001.cub to DA0032.cub will be processed.

Now boot up VMD, and input the command *source isoall.tcl* in its command line window, then

you will have DA0001.bmp, DA0002.bmp ... DA0032.bmp.

There are numerous programs that can convert single-frame graphic files to animation, such as Atani, Premiere, Vegas, Ulead Video Studio, Videomach, etc. Here we use ImageMagick tool in Linux to do this, and we choose gif as the animation format, since gif animation can be directly embedded into webpages.

Copy all of the .bmp files to Linux system, and run below command in the corresponding folder:

```
convert -delay 12 -colors 100 -monitor *.bmp ELF_IRC.gif
```

in which *-delay* controls the time interval between each frame in the animation, and *-colors* determines the number of colors used, the larger the value, the more smoothly the color changes, but the larger the animation file. You can run *convert --help* to study more arguments of this tool

If the the resultant ELF_IRC.gif cannot be properly displayed on your system, use your webpage explorer or advanced image explorers (e.g. IrfanView) to open it. The deformation of ELF isosurface in this animation very intuitively exhibits how the new bonds are formed and how the characteristic of existing bonds changed.

4.A.2 Calculation of spin population

As there are many ways to calculate atomic charges (see Section 3.9 for introduction and 4.7 for examples), there are various ways to calculate spin population. Spin population is defined as the population number of alpha electrons minus that of beta electrons. Spin population is a key quantity for characterizing electronic structure of open-shell systems, *i.e.* radicals and antiferromagnetic systems. From spin population we can clearly know where the spin electrons are mainly distributed. Moreoever, we can discuss contribution from different regions (atomic orbitals, atoms or fragments) to the total magnetic dipole moment **m** due to electron spin. If spin population of a region is *x*, then its contribution to **m** will be $x\mu_B$, where the Bohr magneton $\mu_B=e\hbar/(2m_e)$ (*e*: electron charge, m_e : mass of electron) represents the magnetic moment produced by a single electron. Note that in chemical systems the movement of electron in orbitals and nuclear spins also have contributions to **m**, but the magnitude is evidently weaker and thus can often be neglected.

In Multiwfn, the spin population defined in many different ways can be calculated by three modules, they are briefly discussed below.

(1) Population analysis module (main function 7). In this module, if you select Mulliken or Löwdin population analysis, alpha, beta and spin population of each basis function, shell and angular moment orbitals will be outputted. If you select modified Mulliken population analysis (e.g. SCPA), only the alpha/beta/spin population of each atom will be shown. If you first define a fragment via option -1 in main function 7, then spin population of the fragment will be printed together. Do not use these methods when diffuse basis functions are presented in your basis set, otherwise the result may or may not be reliable.

(2) Fuzzy atomic spaces analysis module (main function 15). After you entered this module, select option 1 and choose electron spin density, the spin population of each atom will be shown. They are calculated by integrating electron spin density in fuzzy space of each atom. If you want to obtain spin population of a fragment, you should first use option -5 to define the atoms to be calculated.

By default the fuzzy atomic space defined by Becke is employed, so the result can be called Becke spin population. If before calculation you selected option -1 to switch to Hirshfeld or

Hirshfeld-I fuzzy atomic space, then the result will correspond to Hirshfeld or Hirshfeld-I spin population. All the Becke, Hirshfeld and Hirshfeld-I methods are reliable in all cases. For more detail you can consult Section 3.18.

(3) Basin analysis module (main function 17). In this module, you can use AIM method to calculate spin population. Please consult Section 4.17.1 on how to perform integration of real space function in AIM atomic basins. If electron spin density is chosen to be the integrand, then the result will correspond to AIM spin population. In general I do not suggest using this method, because the computational cost is evidently higher than using population analysis module and fuzzy atomic spaces analysis module.

Overall, if you only need to calculate atomic spin population, using fuzzy atomic spaces analysis module is recommended, while SCPA is also a good choice when no basis function is employed. However, if more detailed information are requested, such as spin population in different angular moment orbitals, please use or Löwdin Mulliken population analysis.

4.A.3 Overview of methods for studying aromaticity

Aromaticity is a fundamental concept in organic chemistry and wavefunction analysis realm. Previously I wrote a post to thoroughly discuss the methods for studying aromaticity, see "The methods for measuring aromaticity and their calculations in Multiwfn" (in Chinese, <http://sobereva.com/176>). Multiwfn supports a very large number of methods for investigating aromaticity, they are summarized in below table and will be briefly introduced in turn. There are also many other methods, such as induced ring current, ARCS, magnetic susceptibility exaltation, aromatic stabilization energy (ASE), CiLC; they will not be mentioned since they are not directly related to the capacities of Multiwfn.

	Method	Principle	Year	Pop.	Reliab.	Univ.	Ref.	Anti-arom.	Sep. σ/π	Cost	Value
1	Molecular orbital	Hückel	1951	++	0	0	N	Y	Y	0	+
2	AdNDP	Hückel	2008	+	0	+	N	Y	Y	0	+
3	NICS	Magnet.	1996	+++	++	++	N	Y	Y	+	+++
4	ICSS	Magnet.	2001	0	++	++	N	Y	Y	+++	+
5	HOMA	Geom.	1972	+	+	0	Y	Y	N	--	+
6	Bird	Geom.	1985	--	0	-	Y	?	N	--	-
7	Multi-center BO	Delocal.	1990	+	+++	++	N	N	Y	0	+++
8	ELF-σ/π	Delocal.	2004	+	0	-	N	Y	Y	0	+
9	PDI	Delocal.	2003	0	+	0	N	N	Y	0	+
10	ATI	Delocal.	2005	--	+	0	N	N	Y	-	0
11	PLR	Delocal.	2012	--	+	0	N	N	Y	0	0
12	ΔDI	Delocal.	2003	--	-	-	N	N	Y	0	--
13	FLU, FLU-π	Delocal.	2005	0	+	0	Y/N	N	Y	0	+
14	RCP properties	ρ	1997	-	0	0	N	N	?	-	0
15	Shannon aromaticity	ρ	2010				N	Y	N	-	
16	EL index	ρ	2012	--	-	-	Y	Y	N	-	--

In the table, "+++", "++", "+", "0", "-" and "--" correspond to very high, high, relatively high, normal, relatively low and low, respectively. "Y" and "N" stand for "Yes" and "No", respectively. The

meaning of each column are given below.

Principle: The principle behind the method. "Hückel" = Hückel rule; "Magnet." = Magnetic properties; "Geom." = Molecular geometry; "Delocal." = Electron delocalization character; " ρ " = Electron density distribution.

Year: The year that the method was first time proposed.

Pop.: Popularity in recent years.

Reliab.: Reliability, measuring if the method is able to faithfully reveal aromaticity.

Univ.: Universality. A method with high universality must be able to be applied to a wide variety of kinds of systems and situations, such as the rings containing heteroatoms and transition metals, non-equilibrium geometry (e.g. transition state of Diels-Alder adduction), excited state, etc.

Ref.: If the method relies on reference systems. A universal method must avoid this feature.

Anti-arom.: If the method is also able to measure anti-aromaticity.

Sep. σ/π : If the method can be used to separately discuss σ and π aromaticity.

Cost: The computational cost to apply the method.

Value: The overall value. This is the most important descriptor.

Next, the methods presented in above table will be briefed sequentially, and how to realize them in Multiwfn will also be mentioned.

1. Molecular orbital (MO): The famous Hückel $4n+2$ and $4n$ rule for determining aromaticity character was first explicitly presented in *J. Am. Chem. Soc.*, **73**, 876 (1951). For a molecule, if there are totally $4n+2$ electrons in π (σ) MOs, and this set of MOs share the similar delocalization pattern, then the ring involved in these MOs will show π (σ) aromaticity. If there are $4n$ electrons, then the ring should possess anti-aromaticity. Note that for Möbius type of molecule, the $4n+2$ and $4n$ rule are inverted.

In order to use the Hückel rule to determine aromaticity, one should first pick out proper MOs by visualizing MO isosurfaces, you can use main function 0 for this purpose. If the system is exactly planar, you can directly make use the function introduced in Section 3.100.22 to find out the indices of all π MOs.

2. AdNDP (Adaptive natural density partitioning): The MO method shown above commonly is only applicable to the molecule containing only one ring. When there are multiple rings, such as phenanthrene, the MOs are useless, since MOs in general delocalize over the whole molecule and thus cannot be used to study local aromaticity of different rings. The AdNDP method, which was proposed in *Phys. Chem. Chem. Phys.*, **10**, 5207 (2008), is able to overcome this difficulty. AdNDP has been carefully introduced in Section 3.17, and many examples are given in Section 4.14.

3. NICS (Nucleus-independent chemical shift): NICS uses the negative value of magnetic shielding value at ring center to measure its aromaticity. This is the most popular aromaticity index nowadays, it was originally proposed in *J. Am. Chem. Soc.*, **118**, 6317 (1996) and reviewed in *Chem. Rev.*, **105**, 3842-3888. There are also a few variants, among them the best one is NICS(1)_{ZZ}, see *Org. Lett.*, **8**, 863 (2006). For non-planar system, it is often difficult to calculate NICS(1)_{ZZ}, in this case you will find the function introduced in Section 3.100.24 quite useful.

4. ICSS (Iso-chemical shielding surface): The original paper of ICSS is *J. Chem. Soc., Perkin Trans.*, **2**, 1893 (2001). This method analyzes aromaticity by visualizing isosurface of magnetic shielding value around the molecule. See Section 3.200.4 for introduction and Section 4.200.4 for

example. The main drawback of this method is that calculating grid data of magnetic shielding values is fairly time-consuming.

5. HOMA (Harmonic oscillator measure of aromaticity): HOMA measures aromaticity based on bond lengths in the ring of interest. See Section 3.100.13 for introduction and Section 4.100.13 for example.

6. Bird index: The same as above.

7. Multi-center bond order (MCBO): MCBO is an indicator of electron delocalization ability over a ring and is the aromaticity index I most strongly recommended. Larger MCBO value corresponds to stronger aromaticity. See Section 3.11.2 for introduction. Some applications of MCBO in aromaticity studies can be found in *J. Phys. Org. Chem.*, **26**, 473 (2013), *Phys. Chem. Chem. Phys.*, **2**, 3381 (2000) and *J. Phys. Chem. A*, **109**, 6606 (2005). It is straightforward to discuss π and σ aromaticities separately by MCBO, that is before calculating MCBO value, first set occupation number of all σ and π MOs to zero respectively by subfunction 22 of main function 100.

Note that the definition of MCBO in many literatures differ with that in Multiwfn by a constant coefficient. The calculation cost of MCBO for six-membered ring can be ignored, however the cost increases exponentially with the number of atoms in the ring.

8. ELF- σ/π : The ELF calculated solely based on π orbitals and all other orbitals are referred to as ELF- π and ELF- σ , respectively. It was argued that the value of bifurcation point of ELF- π (ELF- σ) is an indicator of π (σ) aromaticity, some applications can be found in *J. Chem. Phys.*, **120**, 1670 (2004), *J. Chem. Theory Comput.*, **1**, 83 (2005) and *Chem. Rev.*, **105**, 3911 (2005). An example of calculating ELF- σ/π is given in Section 4.5.3. In addition, Section 4.4.9 presented an example of studying LOL- π (which is very similar to ELF- π) by plotting plane map. I do not think ELF- σ/π is a very ideal method for measuring aromaticity, mostly because this method often suffers from ambiguity (you will recognize this point if you have used this method to study many practical systems). Also note that the bifurcation values of ELF- σ/π in a lot of literatures are incorrect; if you try, you will find it is impossible to reproduce their results at all. (So do not always believe literatures but believe in yourself!)

9. PDI (Para-delocalization index): This aromaticity index is only applicable to six-membered rings. PDI was first proposed in *Chem. Eur. J.*, **9**, 400 (2003) and reviewed in *Chem. Rev.*, **105**, 3911 (2005). Please check Section 3.18.6 for introduction of PDI and Section 4.15.2 for example of using PDI.

10. ATI (Average two-center indices): ATI was first proposed in *J. Phys. Org. Chem.*, **18**, 706 (2005). In fact ATI does not contain any new idea, it simply replaces the delocalization indices involved in PDI formula with corresponding Mayer bond orders, and according to the discussions in *J. Phys. Chem. A*, **109**, 9904 (2005), there is no essential difference between Mayer bond order and delocalization index in physical nature. If you would like to use ATI, you can directly calculate Mayer bond order by Multiwfn and then manually calculate ATI according to its formula.

11. PLR (Para linear response index): As ATI, PLR is also very akin to PDI. The only difference between PLR and PDI is that the delocalization indices in PDI are replaced with corresponding condensed linear response kernels. Original paper of PLR is *Phys. Chem. Chem. Phys.*, **14**, 3960 (2012). You can check Section 3.18.9 for introduction of PLR and Section 4.15.2 for example of using PLR.

12. ADI: This method was proposed in *Chem. Eur. J.*, **9**, 400 (2003) for measuring aromaticity of 5-membered systems. Consider below case



The ΔDI is simply defined as the difference of delocalization index (DI) between the formal C=C bond and the C-C bond. The DI can be either calculated by fuzzy atomic space analysis module or by basin analysis module (though the definition of atomic spaces are different in these two module, the results are similar in common). In fact, you can also use Mayer bond order instead of DI. I do not believe ΔDI is reliable, since aromaticity is an overall property of a system, while the delocalization over C-X bond is completely ignored in ΔDI .

13. FLU and FLU- π (Aromatic fluctuation index): They were proposed in *J. Chem. Phys.*, **122**, 014109 (2005). See Section 3.18.7 for introduction and Section 4.15.2 for example.

14. RCP properties: In *Can. J. Chem.*, **75**, 1174 (1997) it was shown that the density and the curvature of density perpendicular to the ring plane at ring critical point (RCP) closely relate to aromaticity of the ring. The larger the density, or the more negative the curvature, the larger the aromaticity. You can use topology analysis module of Multiwfn to apply this method. Detail introduction can be found in Section 3.14.6, an example is given in Section 4.2.1.

15. Shannon aromaticity: This method was proposed in *Phys. Chem. Chem. Phys.*, **12**, 4742 (2010), which measures aromaticity based on electron density at bond critical points (BCP) in the ring. See Section 3.14.6 for introduction and the example given in Section 4.2.1.

16. EL index: The idea of EL index is quite similar to HOMA, the most prominent difference is that the bond lengths in HOMA formula are replaced with electron density ellipticity at BCPs in the ring. For more detail see the original paper *Struct. Chem.*, **23**, 1173 (2012). Electron density ellipticity at BCPs can be directly calculated by topology analysis module of Multiwfn. Since the ellipticity at BCP is usually unclear for strongly polar bonds, EL index may be unreliable for the ring containing heteroatoms. In addition, EL index shares the same drawback of HOMA, that is reference system is need. If reference system cannot be obtained, such as the case of metal clusters, this method does not work.

17. Aromaticity indices defined based on information-theoretic quantities: It was demonstrated in *ACS Omega*, **3**, 18370 (2018) that the average of information-theoretic quantities of the atoms constituting a ring is closely related to aromaticity. This method is supported as subfunction 12 of main function 15, see Section 3.18.11 for details.

4.A.4 Overview of methods for predicting reactive sites

There are numerous methods able to predict reactive site of electrophilic, nucleophilic and radical reactions, and almost all of them are supported by Multiwfn. In this section, I will summary and briefly introduce the methods available in Multiwfn. The interested reader is highly recommended to take a look at [*Acta Phys.-Chim. Sinica*, 30, 628 \(2014\)](#), in which various methods for predicting electrophilic sites are carefully introduced and thoroughly compared. You may also find the slideshow "Predicting reactive sites" in "Related resources and posts" Section of Multiwfn website useful.

1 Electrostatic potential (ESP). If you are not familiar with ESP, please consult corresponding introduction in Section 2.6. Since electrophile (nucleophile) locally carries negative (positive)

charge, and thus tends to be attracted to the region where ESP is negative (positive), the position and value of minima (maxima) of ESP on molecular vdW surface is often used to reveal favorable site of electrophilic (nucleophilic) attack. ESP analysis can be realized via quantitative molecular surface analysis module, see Section 4.12 for detailed introduction and Section 4.12.1 for example. There are also alternative ways to study ESP; as illustrated in Section 4.12.3, the average of ESP on local vdW surface corresponding to each atom is also very useful, and this approach is more reliable and robust than analyzing ESP extrema on vdW surface. For planar system, one can also calculate and compare the ESP value above 1.6 Å (approximately equal to vdW radius of carbon) of molecular plane from different atoms to examine their reactivities; to do this, you need to use main function 1, which directly outputs various real space function values at given points.

However, as shown in my paper [Acta Phys.-Chim. Sin., 30, 628 \(2014\)](#), ESP is usually not a reliable property for predicting reactive sites.

2 Average local ionization energy (ALIE). If you are not familiar with ALIE, please consult corresponding introduction in Section 2.6. ALIE can be studied in analogous ways to ESP. The most common way to predict reactive sites in terms of ALIE is analyzing minima of ALIE on vdW surface, see Section 4.12.2 for example. Also, you can study average of ALIE on local vdW surface or evaluate ALIE above 1.6 Å of molecular plane for planar system.

Note that ALIE analysis is applicable to electrophilic and radical attacks, but it is useless for nucleophilic attack. However, another similar real space function named local electron affinity (LEA) may be useful for this purpose, see *J. Mol. Model.*, **9**, 342 (2003). LEA can be studied in Multiwfn via user-defined function, see Section 2.7 for detail. According to my experience, LEA is not well defined and usually does not work well.

3 Atomic charges. It is easy to understand that favorable electrophilic and nucleophilic reactive sites should carry negative and positive atomic charges respectively, so that they can attract electrophile and nucleophile to attack them. Multiwfn supports a lot of methods to calculate atomic charges, see Section 3.9 for introduction and Section 4.7 for some instances. Among the available atomic charges, the best one for predicting reactive sites purpose may be Hirshfeld, interested readers are suggested to consult *J. Phys. Chem. A*, **118**, 3698 (2014). Do not use Mulliken charges, which may be the worst one, though it is the most popular charge model.

4 Frontier molecular orbital (FMO) theory. Atom with larger contribution to HOMO (LUMO) is more likely to be the preferential site of electrophilic (nucleophilic) attack. Multiwfn supports many kinds of methods to calculate molecular orbital composition, see Section 3.10 for introduction and Section 4.8 for examples. Commonly I suggest using Becke or Hirshfeld method. Mulliken method works equally well if no diffuse functions are presented. NAO method is also a good choice, but not suitable for analyzing virtual MOs. Besides, you can also directly visualize the isosurface of MOs by main function 0 to discuss their compositions.

5 Fukui function and condensed Fukui function. The Fukui function proposed in *J. Am. Chem. Soc.*, **106**, 4049 (1984) by Parr is the most prevalently used method for predicting reactive sites nowadays. Please consult Section 4.5.4 for introduction and illustration. Fukui function is a real space function, which is commonly studied by means of visualization of isosurface. In order to facilitate quantitative comparison between difference sites, one can calculate condensed Fukui function based on atomic charges, please consult Section 4.7.3. In addition, as demonstrated in Section 4.12.4, the Fukui function can also be investigated in combination with local quantitative molecular surface analysis technique.

6 Dual descriptor and condensed dual descriptor. As demonstrated in *Acta Phys.-Chim. Sinica*, **30**, 628 (2014), the dual descriptor proposed in *J. Phys. Chem. A*, **109**, 205 (2005) may be the most robust method for predicting reactive sites, at least for electrophilic reaction. Like Fukui function, dual descriptor also has condensed version. Dual descriptor and the condensed version are introduced in Section 4.5.4 and 4.7.3, respectively.

7 Orbital overlap distance function. Analysis of this function may be useful for revealing favorable reactive site, see Section 4.12.8 for example.

4.A.5 Overview of methods for studying weak interactions

There are a lots of ways to characterize weak interactions, and most of them are supported by Multiwfn, here I give you a brief summary. If you can read Chinese, I suggest reading my blog article "An overview of the weak interaction analysis methods supported by Multiwfn" (<http://sobereva.com/252>), in which this topic is discussed more deeply and extensively.

(1) AIM topology analysis is a very popular method for studying both strong and weak interactions. Its use in weak interaction analysis is partially illustrated in Section 4.2.1.

(2) NCI analysis proposed in 2010 may be viewed as an visualization extension of AIM analysis, this method rapidly became quite popular after it was proposed. The example of using NCI analysis is given in Sections 3.23.1, 4.20.1 and 4.20.2. NCI analysis is also able to be employed to study weak interaction in dynamic environment such as molecular dynamic simulation, see the introduction in Section 3.23.2 and the accompanied example in Section 4.20.3. Integrating domain of NCI is a useful way to discuss weak interactions quantitatively, examples are provided in Section 4.200.14.

DORI and IGM analyses are closely related to NCI method. Advantage of DORI is that all kinds of interactions can be simultaneously visualized, as illustrated in Section 4.20.5. The strong point of IGM analysis is that this method is able to individually visualize intrafragment and interfragment interaction regions, and contribution of atomic pairs and atoms can be quantified and rendered, see Section 3.23.5 for introduction and 4.20.10 for example.

(3) Electrostatic potential (ESP) analysis. ESP has been introduced in Section 2.6, this is a extremely important real space function for studying electrostatic dominated weak interactions. There are many different ways to carry out ESP analysis:

- Visually studying ESP color-mapped molecular vdW surface, this analysis can be used to quickly figure out potential electrostatic interaction sites and qualitatively study interaction strength. See the end of Section 4.12.1 and *J. Mol. Model.*, **13**, 291 (2007) for example.
- Studying ESP minima and maxima on molecular vdW surface. This can done by quantitative molecular surface analysis module, see Section 4.12.1 for example and Section 3.15 for more details. The value of these ESP extrema on vdW surface strongly correlate with electrostatic interaction energies, and you can find many papers have used this method, for example *Phys. Chem. Chem. Phys.*, **15**, 14377 (2013), *J. Mol. Model.*, **13**, 305 (2007), *Int. J. Quantum. Chem.*, **107**, 3046 (2007), *Phys. Chem. Chem. Phys.*, **12**, 7748 (2010), *J. Mol. Model.*, **14**, 659 (2008), *J. Mol. Model.*, **18**, 541 (2012), *J. Mol. Model.*, **15**, 723 (2009), Chapter 6 of book *Practical Aspects of Computational Chemistry* (2009).
- Superposition analysis of ESP contour map. This method was proposed by Tian Lu in *J. Mol. Model.*, **19**, 5387 (2013), it is quite vivid, easy-to-use and powerful. It was demonstrated that

stability of complex configurations can be fairly well predicted by this method. Section 4.4.4 showed how to plot ESP contour map.

- In *J. Phys. Chem. A*, **118**, 1697 (2014), the authors showed that by making use of ESP at nuclear positions the electrostatic dominated intermolecular interaction energies can be very accurately predicted. See Section 4.1.2 for introduction and example.

(4) Atomic charge analysis. Atomic charge is a very simple and intuitive model for describing charge distribution and can be used to analyze the strength of electrostatic interaction between different sites. The functions for calculating atomic charges are introduced in Section 3.9, and some practical examples are given in Section 3.7.

(5) Bond order and delocalization index (DI) analysis. Commonly weak interactions are dominated by electrostatic and/or vdW interactions, so bond order and DI analysis, which mainly reflect covalent character are often not useful in these cases. However, for "strong" weak interactions, such as low-barrier hydrogen bonds (LBHB) and charge-assisted halogen bonds, covalent contribution may be not negligible, and thus bond order and DI analysis can be applied. Bond order calculations are illustrated in Section 4.9. In Multiwfn, DI can be calculated based on fuzzy atomic space or AIM basin, the former is equivalent to fuzzy bond order, while the latter can be evaluated in basin analysis module, see example in Section 4.17.1.

(6) ELF analysis. In *Theor. Chem. Acc.*, **104**, 13 (2000), Fuster and Silvi defined CVB index based on ELF to distinguish strength of H-bonds. *J. Phys. Chem. A*, **115**, 10078 (2011) employed this method to study a large amount of resonance-assisted hydrogen bonds and find this index is in good correlation with other H-bond strength indices. CVB index can be easily calculated in Multiwfn, see Section 3.200.1 for detail. There are also other papers using ELF to study H-bonds, e.g. *Chem. Rev.*, **111**, 2597 (2011).

(7) Charge variation analysis. Weak interactions often accompanied by charge transfer and polarization, therefore studying how the electrons are transferred between or within molecules, as well as how the electron density is polarized due to the presence another molecule are important. There are many available ways to investigate these points:

- Plotting difference map of electron density between complex and monomers. This is the most straightforward and intuitive way to study variation of electron density. The procedure is illustrated in Section 4.5.5.

• Plotting charge displacement curve. After generating grid data of density difference, in order to quantitatively study the charge variation in a direction, you can plot charge displacement curve, see Section 3.16.14 for introduction and Section 4.13.6 for example.

• Variation of atomic charges of monomers in their isolated states and in complex state can quantitatively and clearly show how the electrons are transferred between different atoms/fragments due to the interaction.

• After generating grid data of electron density difference between complex and monomers, you can use basin analysis module to integrate basin of density difference to study amount of electron variation in various characteristic regions (e.g. the region corresponding to σ -hole). You can consult the example in Section 4.17.4.

• Charge decomposition analysis (CDA). CDA is used to reveal underlying details of charge transfer, the amount of donation and back-donation of electrons between two fragments due to various complex MOs can be studied. In addition, the CDA module of Multiwfn can tell you how the fragment MOs are mixed and hence yield complex MOs. CDA is commonly applied to strong

interaction, but it may be also useful for exploring weak interactions. The theory of CDA is introduced in Section 3.19, practical examples are given in Section 4.16.

- Multiwfn has a function dedicated to analyze charge transfer in electron excitation based on electron density difference, many important quantities characterizing the transfer can be obtained, see Section 3.21.3 for introduction and 4.18.3 for example. Based on the grid data of electron density difference between complex and monomers, this function may be also useful for studying charge transfer due to weak interaction.

(8) Hirshfeld and Becke surface analysis. This kind of analysis is extremely useful for revealing weak interaction in molecular crystals, but can also be applied to molecular clusters, see examples in Sections 4.12.5 and 4.12.6 as well as theory introduction in Section 3.15.5.

(9) LOLIPOP index is useful for measuring π - π stacking ability, see the introduction in Section 3.100.14 and the example in Section 4.100.14.

(10) Energy decomposition analysis is a kind of important approaches for characterizing the nature of weak interactions, physical components of total interaction energy can be separately obtained. Multiwfn is capable of performing simple energy decomposition analysis in combination with Gaussian, see Section 4.100.8 for example. Multiwfn can also perform energy decomposition analysis based on molecular forcefield, this function is very useful, flexible and easy-to-use, see Section 3.24.1 for introduction and Section 4.21.1 for example.

(11) Source function analysis is defined in the framework of AIM theory. Gatti et al. suggested using source function to study both strong and weak interactions. Introduction of source function can be found in Section 2.6, and tutorial of performing source function analysis is given in Section 4.17.5. A thorough review is *Struct. & Bond.*, **147**, 193 (2010), in which H-bond analysis is involved.

(12) Mutual penetration distance of vdW surfaces. For the same kind of weak interaction, generally the larger the penetration of vdW surface, the stronger the interaction strength. For a noncovalently interacting atom pair AB, the difference between the distance of A-B and the sum of their non-bonded radii is termed as mutual penetration distance. The non-bonded atomic radius is the closest distance between a nucleus and the molecular vdW surface, and can be obtained by option 10 in post-process interface of quantitative molecular surface analysis module of Multiwfn.

(13) Atomic multipole moment analysis. The definition of atomic multipole moment can be found in Section 3.18.3. Atomic multipole moment measures the anisotropy distribution of electron density around an atom, which has important impact on interatomic electrostatic interactions. See Section 7.4.3 of the Bader's book *Atoms in molecules-A quantum theory* for illustrative examples. In Multiwfn, atomic multipole moment can be calculated by both fuzzy space analysis module and basin analysis module, for the latter case see Section 4.17.1 for example.

(14) Orbital overlap. For weak interactions involving orbital interaction, you can use Multiwfn to study orbital overlap, which is closely related to orbital interaction strength. The example Section in 4.100.15 illustrated how to calculate intermolecular orbital overlap integral. Section 4.0.2 exemplified how to visualize overlap degree of two NBO orbitals, high (low) overlap degree commonly implies large (small) second-order perturbation energy E(2) between the two NBOs.

(15) As demonstrated in *J. Mol. Model.*, **19**, 2035 (2013), interaction energy of halogen-bond complexes is well correlated with the properties of (3,-1) critical point of Laplacian of electron density at σ -hole location. The topology analysis of Laplacian of electron density can be conveniently realized in main function 2. Section 4.2.2 showed how to perform topology analysis for LOL, however you can use the same method to analyze Laplacian of electron density.

There are also other possibly ways to study weak interactions, but they are not directly relevant to Multiwfn. These methods include: NBO E(2) and NBO deletion analyses (see the tutorials in NBO official website), rehybridization analysis (specific for H-bond, based on natural population analysis), variation of bond length and vibrational frequency, Mayer energy decomposition analysis (*Phys. Chem. Chem. Phys.*, **8**, 4630 (2006)), magnetically induced current (*Phys. Chem. Chem. Phys.*, **13**, 20500 (2011)), interacting quantum atoms (IQA, see *J. Phys. Chem. A*, **117**, 8969 (2013) for example), SAPT analysis (supported by PSI4, Molpro etc. see *WIREs Comput. Mol. Sci.*, **2**, 254 (2012)), ETS-NOCV (supported by ORCA, see *J. Chem. Theory Comput.*, **5**, 962 (2009)).

4.A.6 Calculate odd electron density and local electron correlation function

Part 1: Odd electron density

Some papers calculate the so-called odd electron density (OED), its was originally defined in *Chem. Phys. Lett.*, **372**, 508 (2003), and employed in many papers, e.g. *Theor. Chem. Acc.*, **130**, 711 (2011) and *J. Phys. Chem. C*, **116**, 19729 (2012). Here I introduce how to plot it by using Multiwfn in combination with .wfn file produced by Gaussian.

Spatial natural orbitals are yielded by diagonalizing total density matrix and have occupation number within 0 and 2.0. The OED of the k th natural orbital is defined as

$$\rho_k^{\text{odd}}(\mathbf{r}) = \min(2 - n_k, n_k) \rho_k(\mathbf{r})$$

where $\rho_k(\mathbf{r})$ and n_k are its electron density and occupation number, respectively. Thus for $n_k \leq 1$, the prefactor corresponds to occupation number, while for $n_k \geq 1$, the prefactor corresponds to the complement to achieve a closed shell. In other words, the $\min(2-n_k, n_k)$ term can be regarded as the probability of the electron to be unpaired in this orbital.

The total OED is defined as the summation of OED for all natural orbitals:

$$\rho^{\text{odd}}(\mathbf{r}) = \sum_k \rho_k^{\text{odd}}(\mathbf{r})$$

Now we calculate OED for a closed-shell system OC-BH₃ at CCSD level (while at HF/DFT level, this quantity is obviously zero everywhere). The Gaussian input file can be found in *examples\COBH3_CCSD.gjf*, note that *density out=wfn* keywords are used. The resulting file *examples\COBH3_CCSD.wfn* contains all CCSD natural orbitals.

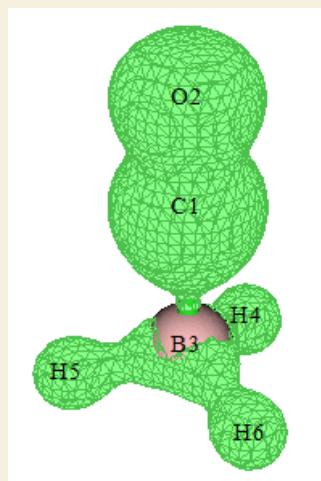
We first calculate total OED. Boot up Multiwfn and input

```
examples\COBH3_CCSD.wfn
6
26 // Modify occupation number
0 // Select all orbitals
odd // Take min(2-nk, nk) as occupation number for all orbitals
q // Return
-1
```

Then we plot isosurface map of electron density as usual

5
I
2
-1

Then we set isovalue to 0.005, you will see the total OED, as shown below. Although the use of OED in above mentioned papers aims to study biradical character, *in my personal viewpoint, this function is also useful to reveal where electron correlation is significant*. As you can see in the below graph, electron correlation is very strong in the multiple-bonds region of CO.

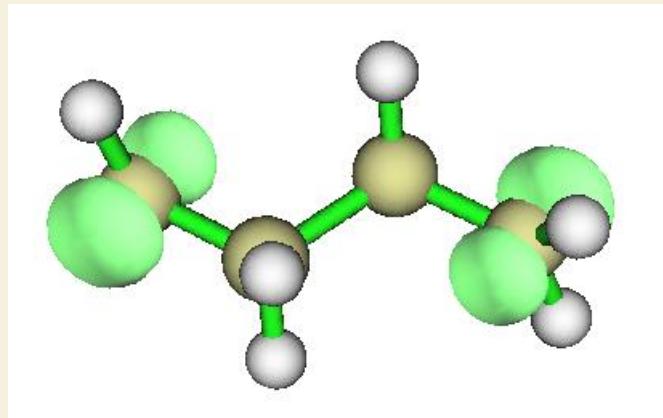


Next, we plot OED for natural orbital 11. What we need to do first is cleaning the occupation number for all other orbitals. Close the GUI and input

0
6
26
1-10
0 // Clean the occupation number for orbitals 1 to 10
12-57
0 // Clean the occupation number for orbitals 12 to 57

Then return to main menu and plot electron density as usual. That's all.

Below we plot total OED for a typical biradical system C₄H₈ at M06-2X level. In this case, unrestricted open-shell calculation is needed and *guess=mix* keyword should be used to achieve symmetry-broken state. In addition, *pop=no out=wfn* must be specified so that spatial natural orbital will be generated by mixing alpha and beta density matrix, diagonalization and then be outputted to .wfn file. The natural orbitals obtained by such a UHF/UDFT calculation are sometimes referred to as unrestricted natural orbitals (UNO). The Gaussian input file for producing the .wfn file is *examples\c4h8-uno.gif*, and the resulting .wfn file is *examples\c4h8-uno.wfn*. Please use this file to plot total OED like above example, the isosurface map with isovalue of 0.02 should look like below (its character is quite similar with spin density map)



Part 2: Local electron correlation functions

The local electron correlation functions proposed in *J. Chem. Theory Comput.*, **13**, 2705 (2017) are very useful real space functions to reveal total, dynamic and nondynamic electron correlation in various molecular regions. The definition of these function can be found in entry 87, 88 and 89 of Section 2.7. As you can see, their forms are closely related to OED. Now let us plot local total electron correlation functions I_T for OC-BH₃.

Set "iuserfunc" in settings.ini to 87, then boot up Multiwfn and input

examples\COBH3_CCSD.wfn

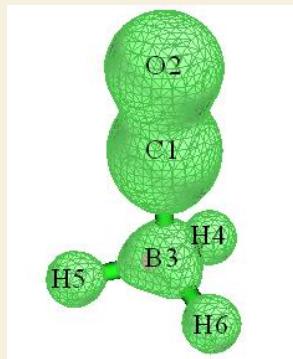
5

100 // User-defined function, now it corresponds to I_T

2 // Medium quality grid

-1 // Visualize isosurface

Set isovalue to 0.013, then you will see



This graph is rather similar with the isosurface map of OED. Since I_T is a real space function specific for revealing electron correlation, our observation clearly demonstrates that OED is also capable of exhibit electron correlation.

Local electron correlation function has another two forms, namely I_D and I_{ND} , which aim for representing dynamic and nondynamic electron correlation, respectively. They can be plotted with the same way as shown above, but you need to set "iuserfunc" to 88 and 89, respectively.

Plotting individual contribution to local electron correlation functions is also possible, you can screen uninteresting natural orbitals by setting their occupation numbers to zero via suboption 26 of main function 6.

Using subfunction 4 of main function 100 you can integrate local electron correlation functions

over the whole space, the result indicates the magnitude of electron correlation for the whole system. For example, we return to main menu and input

```
100
4 // Integrate a real space function over the whole space
100 // User-defined function
```

The result, which is referred to as total electron correlation index, is 1.576. Repeat this calculation for dynamic and nondynamic electron correlation functions, you will find the resulting indices are 1.267 and 0.309, respectively. Evidently, dynamic correlation governs the total electron correlation effect for present system.

In fact, the total, dynamic and nondynamic electron correlation indices can also be calculated by subfunction 15 of main function 200, which is much faster and more convenient, see Section 3.200.15 for introduction. Now we have a try. Return to main menu and input 200 then 15, you will immediately see

```
Nondynamic correlation index: 0.30880061
Dynamic correlation index: 1.26746042
Total correlation index: 1.57626103
```

4.A.7 Study (hyper)polarizability density

Chinese version of this section is my blog article " Using Multiwfn to calculate (hyper)polarizability density" <http://sobereva.com/305>.

By means of custom operation of main functions 4 and 5, (hyper)polarizability density can be easily plotted by Multiwfn as plane map and isosurface map. This is quite useful for discussing nature of (hyper)polarizability of a given molecule. If this feature is used in your work, citing my paper *J. Comput. Chem.*, **38**, 1574 (2017) is recommended, in which the (hyper)polarizability density analysis is involved. Other illustrative applications can be found in *e.g. Phys. Chem. Chem. Phys.*, **19**, 23951 (2017).

Part 1: Basic theory of (hyper)polarizability density

There is a well-known Taylor expansion relationship for dipole moment

$$\begin{aligned}\boldsymbol{\mu}(\mathbf{F}) &= -\frac{\partial E}{\partial \mathbf{F}} = \boldsymbol{\mu}_0 + \boldsymbol{\alpha} \mathbf{F} + (1/2)\boldsymbol{\beta} \mathbf{F}^2 + (1/6)\boldsymbol{\gamma} \mathbf{F}^3 + \dots \\ \boldsymbol{\mu}_0 &= -\left.\frac{\partial E}{\partial \mathbf{F}}\right|_{\mathbf{F}=0} \quad \boldsymbol{\alpha} = -\left.\frac{\partial^2 E}{\partial \mathbf{F}^2}\right|_{\mathbf{F}=0} \quad \boldsymbol{\beta} = -\left.\frac{\partial^3 E}{\partial \mathbf{F}^3}\right|_{\mathbf{F}=0} \quad \boldsymbol{\gamma} = -\left.\frac{\partial^4 E}{\partial \mathbf{F}^4}\right|_{\mathbf{F}=0}\end{aligned}$$

where \mathbf{F} is external electric field vector, E is system total energy, $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_0$ are current electric dipole moment and permanent dipole moment, respectively. $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are polarizability, the first and second hyperpolarizability tensors, respectively.

Similarly, Taylor expansion with respect to \mathbf{F} can be applied to electron density

$$\begin{aligned}\rho(\mathbf{r}, \mathbf{F}) &= \rho^{(0)}(\mathbf{r}) + \boldsymbol{\rho}^{(1)}(\mathbf{r}) \mathbf{F} + (1/2)\boldsymbol{\rho}^{(2)}(\mathbf{r}) \mathbf{F}^2 + (1/6)\boldsymbol{\rho}^{(3)}(\mathbf{r}) \mathbf{F}^3 + \dots \\ \boldsymbol{\rho}^{(1)}(\mathbf{r}) &= \left.\frac{\partial \rho(\mathbf{r})}{\partial \mathbf{F}}\right|_{\mathbf{F}=0} \quad \boldsymbol{\rho}^{(2)}(\mathbf{r}) = \left.\frac{\partial^2 \rho(\mathbf{r})}{\partial \mathbf{F}^2}\right|_{\mathbf{F}=0} \quad \boldsymbol{\rho}^{(3)}(\mathbf{r}) = \left.\frac{\partial^3 \rho(\mathbf{r})}{\partial \mathbf{F}^3}\right|_{\mathbf{F}=0}\end{aligned}$$

Since $\mu(\mathbf{F}) = \int -\rho(\mathbf{r}, \mathbf{F}) \mathbf{r} d\mathbf{r}$, by comparing above equations, we find

$$\begin{aligned}\mu_0 &= \int -\rho^{(0)}(\mathbf{r}) \mathbf{r} d\mathbf{r} & \alpha &= \int -\rho^{(1)}(\mathbf{r}) \mathbf{r} d\mathbf{r} \\ \beta &= \int -\rho^{(2)}(\mathbf{r}) \mathbf{r} d\mathbf{r} & \gamma &= \int -\rho^{(3)}(\mathbf{r}) \mathbf{r} d\mathbf{r}\end{aligned}$$

where $\rho^{(1)}$ is known as polarizability density, while $\rho^{(2)}$ and $\rho^{(3)}$ are known as the first and second hyperpolarizability densities, respectively. Using (hyper)polarizability densities, we can easily investigate contribution of various regions to total molecular (hyper)polarizabilities. In current example we will only focus on $\rho^{(3)}$. Of course, by analogous treatment we can also easily study $\rho^{(1)}$ and $\rho^{(2)}$.

The second hyperpolarizability density $\rho^{(3)}$ is a third-order tensor function, it can be explicitly represented as

$$\rho_{ijk}^{(3)}(\mathbf{r}) = \left. \frac{\partial^3 \rho(\mathbf{r})}{\partial F_i \partial F_j \partial F_k} \right|_{\mathbf{F}=0}$$

It is impossible to discuss all of its components, since there are as many as $3*3*3=27$ components. Assume that the γ_{zzz} is the most crucial component of γ , we can simply study $\rho_{zzz}(\mathbf{r})$:

$$\gamma_{zzz} = \int -z \rho_{zzz}^{(3)}(\mathbf{r}) d\mathbf{r} \quad \rho_{zzz}^{(3)}(\mathbf{r}) = \left. \frac{\partial^3 \rho(\mathbf{r})}{\partial F_z^3} \right|_{F_z=0}$$

Clearly, the $-z \rho_{zzz}^{(3)}(\mathbf{r})$ is simply the contribution of point \mathbf{r} to the γ_{zzz} . If it is plotted as isosurface map or plane map, the source of γ_{zzz} can be vividly revealed.

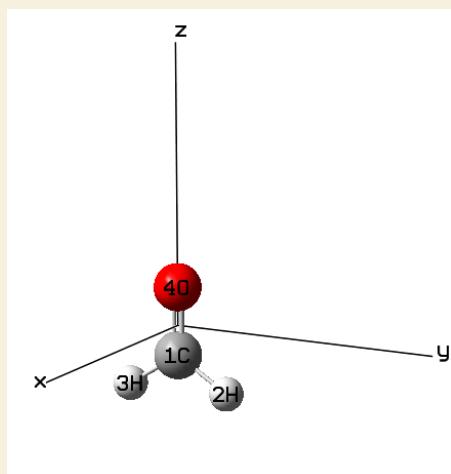
The easiest way of obtaining the $\rho_{zzz}^{(3)}$ is using finite difference method (see my article <http://sobereva.com/305> on how to derive it)

$$\rho_{zzz}^{(3)} = \frac{\rho(2F_z) - 2\rho(F_z) + 2\rho(-F_z) - \rho(-2F_z)}{2(F_z)^3}$$

where F_z is strength of the external electric field applied along Z axis. The functions such as $\rho(F_z)$ and $\rho(-F_z)$ denote the electron density distribution yielded when F_z is applied along positive and negative directions of Z-axis, respectively. The F_z in this case corresponds to finite difference step size, it should not be too large or too small, otherwise numerical error will be significant. According to my experience, 0.003 a.u. is a good choice of F_z .

Part 2: Preparation work

Below we will study the second hyperpolarizability density for a typical small molecule H₂CO. The orientation of the molecule must be clarified, in current case the C=O bond is parallel to the Z-axis, as shown below



Before performing hyperpolarizability density analysis, we first carry out a regular static γ calculation using Gaussian, so that we can then judge whether our hyperpolarizability density analysis is reasonable. The Gaussian input and output files have been provided as *polar=gamma.gif* and *polar=gamma.out* in "examples\hyperpolar" folder. As can be seen at the end of the output file, the γ_{zzz} is 0.145571D+04 a.u. (1455.71 a.u.). The theoretical method we used here is Hartree-Fock, evidently this is not a good choice for practical studies, but it is acceptable for illustrating purpose.

Next, we need to obtain files containing wavefunctions yielded with $-2F^z$, $-F^z$, $+F^z$, $+2F^z$ external electric fields, respectively. The corresponding Gaussian input files have been provided in "examples\hyperpolar" folder, the generated .wfx files are also provided. Note that for Gaussian users, using .wfx and .fch formats are recommended for present study, while using .wfn format is highly deprecated, since recording accuracy of wavefunction in this format is relatively poor.

NOTE 1: In general, when applying external electric field, you should add *nosymm* keyword in Gaussian input file so that Gaussian will not automatically put the system in standard orientation. However, since I found the standard orientation of H₂CO just meet our expectation, namely C=O bond is parallel to Z-axis, therefore *nosymm* keyword is ignored.

NOTE 2: The direction of external electric field applied via *field* keyword in Gaussian is contrary to common convention. If 0.003 a.u. field is applied along positive direction of Z-axis, the keyword should be *field=Z-30*.

Part 3: Draw isosurface map of $\rho^{(3)}_{ZZZ}$ and $-z\rho^{(3)}_{ZZZ}$

Please first familiarize yourself with custom operation feature of Multiwfn by reading such as Section 4.5.4. Here we will calculate $\rho^{(3)}_{ZZZ}$ according to the finite difference equation shown earlier by using custom operation in main function 5, so that we can obtain its grid data and directly it as isosurface map.

Boot up Multiwfn and input below commands

```
examples\hyperpolar\H2CO+2FZ.wfx // The first term in above mentioned equation, namely
 $\rho(2F^z)$ 
```

```
5 // Calculate grid data
```

```
0 // Custom operation
```

```
5 // Five files will be operated on the firstly loaded file
```

```
-,examples\hyperpolar\H2CO+FZ.wfx // The  $-2\rho(F^z)$  term in the equation. The coefficient
of 2 is taken into account by employing subtraction operation twice
```

```
-,examples\hyperpolar\H2CO+FZ.wfx
```

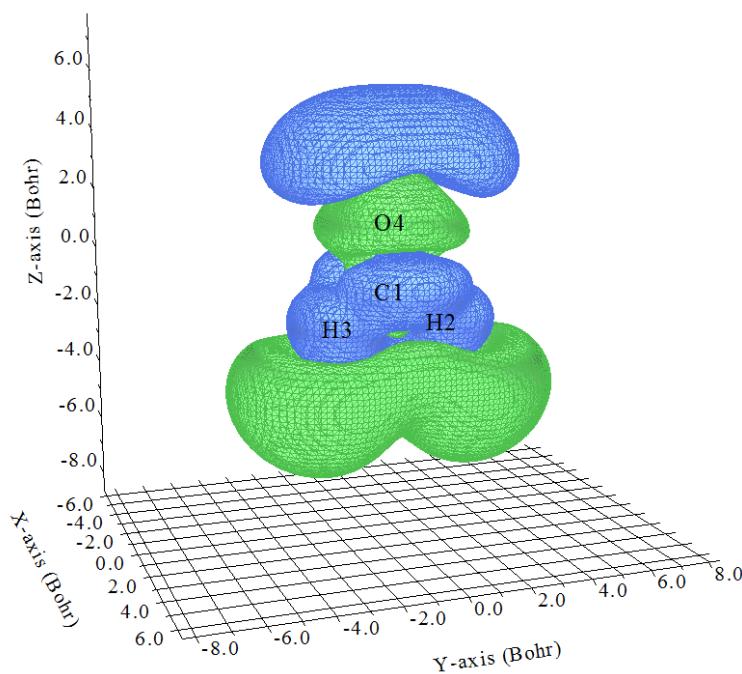
```
+,examples\hyperpolar\H2CO-FZ.wfx // The  $2\rho(-F^z)$  term
```

```
+,examples\hyperpolar\H2CO-FZ.wfx
```

```
-,examples\hyperpolar\H2CO-2FZ.wfx // The  $-\rho(-2F^z)$  term
```

```

1 // Electron density
2 // Medium quality grid. This is adequate for small system
After calculation, we input
6 // Divide the grid data by a value
0.000000054 // The denominator of the equation, namely  $2(F^z)^3=2\times0.003^3$ 
-1 // Visualize isosurface
Now you will see the isosurface map of  $\rho^{(3)}_{ZZZ}$ :
```

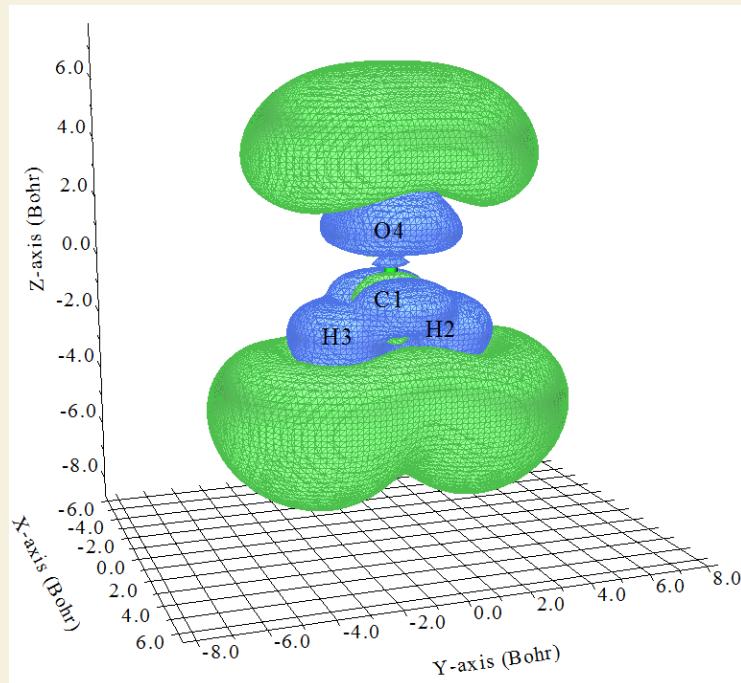


However, from this map it is still somewhat difficult to discuss contribution of various regions to the $\rho^{(3)}_{ZZZ}$, because the $-z$ prefactor has not been taken into account. To obtain $-z\rho^{(3)}_{ZZZ}$, we input below commands

```

0 // Return to main menu
13 // Process grid data
11 // Performing mathematical operation on grid data
20 // Multiply the grid data by a coordinate variable
Z // Multiply Z coordinate
11 // Performing mathematical operation on grid data
5 // Multiply the grid data by a constant
-1 // Multiplying -1. The grid data in memory now has corresponded to  $-z\rho^{(3)}_{ZZZ}$ 
-2 // Visualize isosurface
```

Now you will see



In the figure, the regions enclosed by green (blue) isosurfaces have positive (negative) contribution to the γ_{zzzz} . It can be seen that in the molecular valence region, the contribution is basically negative, however at the two ends of the molecule the contribution is remarkably positive, and the magnitude obviously exceeds the negative region, this is why the current system has an evidently positive value of γ_{zzzz} (the value calculated by Gaussian is 1455.71 a.u.). For a large system, through such a picture, you can immediately explain why γ is big or small, or why it is positive or negative. Undoubtedly this kind of analysis is extremely useful for study underlying characteristics of (hyper)polarizability.

It is noteworthy that one can also investigate how basis set and theoretical method affect the result of (hyper)polarizability by plotting difference between grid data of (hyper)polarizabilities generated by two different levels as isosurface map, namely first calculating the two sets of grid data, and then use mathematical operation in main function 13 to obtain grid data of their difference.

One can verify if the γ_{zzzz} evaluated by integrating the grid data of $-z\rho^{(3)}_{zzz}$ is close to the γ_{zzzz} directly evaluated by Gaussian. We close the GUI of main function 13, enter its option 17 and then input 1, the statistical data of the grid data over the whole space will be printed, from it you can find below line:

Integral of all data: 1429.5132718689

This is the γ_{zzzz} obtained by integrating the cubic grids of the $-z\rho^{(3)}_{zzz}$, it is quite close to the 1455.71 a.u. outputted by Gaussian, showing that our calculation steps are correct and the result is meaningful. (If we use high quality grid and increase the extension distance of the grid data from the default value to 10 Bohr, then the result will be 1459.43 a.u., which is closer to the γ_{zzzz} outputted by Gaussian)

Part 4: Plot plane map of $\rho^{(3)}_{zzz}$ and $-z\rho^{(3)}_{zzz}$

Using custom operation feature in main function 4 of Multiwfn, the $\rho^{(3)}_{zzz}$ and $-z\rho^{(3)}_{zzz}$ can also easily plotted as various kinds of plane map. We first plot contour line map for $\rho^{(3)}_{zzz}$, input below commands:

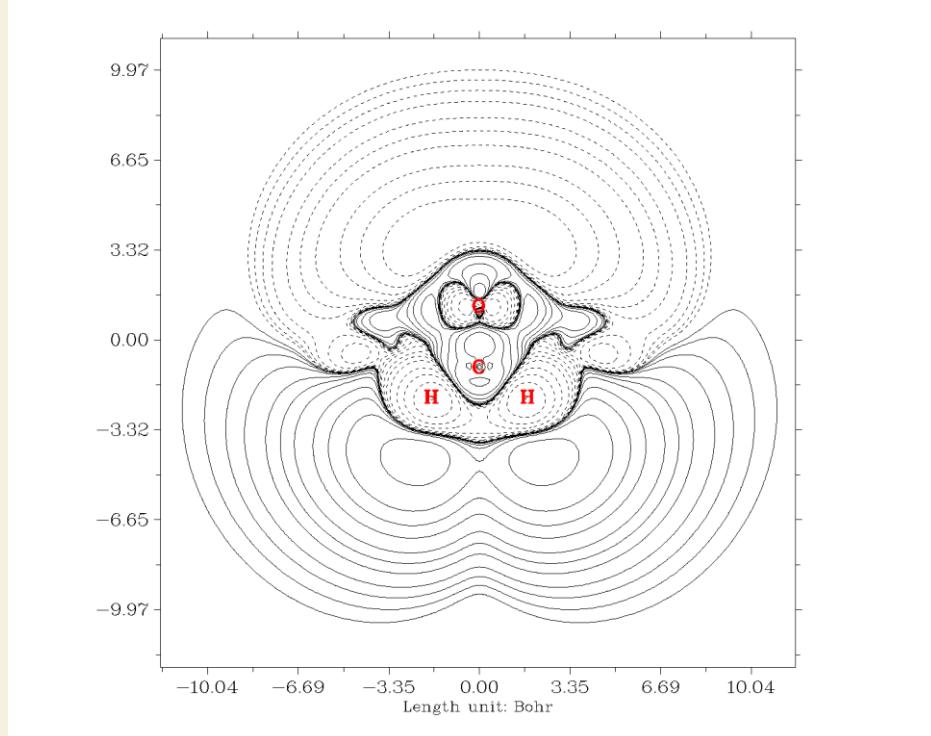
`examples\hyperpolar\H2CO+2FZ.wfx`

```

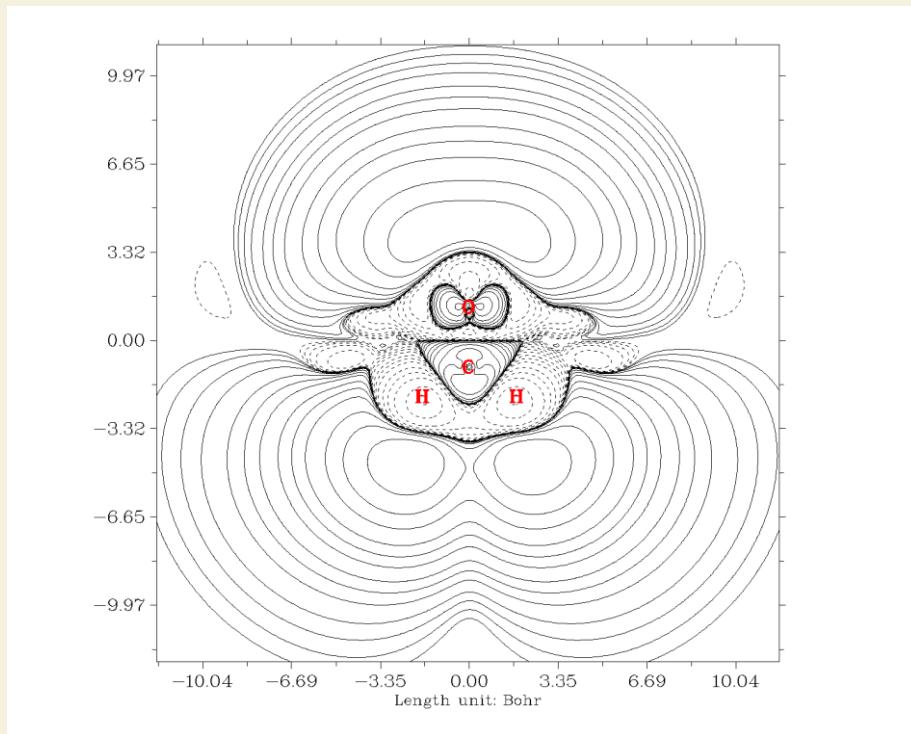
4 // Plot plane map
0 // Custom operation
5 // Five files will be operated on the firstly loaded file
-,examples\hyperpolar\H2CO+FZ.wfx
-,examples\hyperpolar\H2CO+FZ.wfx
+,examples\hyperpolar\H2CO-FZ.wfx
+,examples\hyperpolar\H2CO-FZ.wfx
-,examples\hyperpolar\H2CO-2FZ.wfx
1 // Electron density
2 // Contour line map
Press ENTER button to employ default grid setting
0 // Modify extension distance
10 // Increase to 10 Bohr
3 // YZ plane
0 // X=0
Close the graph that pops up and then input
-7 // Multiply the data by a value
18518518.5185185 // Namely the inverse of the denominator 0.000000054
-1 // Replot

```

Then you will see below map, which corresponds to a slice of the $\rho^{(3)}_{ZZZ}$ isosurface map shown earlier. The solid and dashed lines correspond to positive and negative regions, respectively.



Next, we plot plane map for $-z\rho^{(3)}_{ZZZ}$. Quit Multiwfn, set "iuserfunc" in settings.ini to 23, we do this because in this case the user-defined function will correspond to the $-z\rho$ function, if we repeat above steps but replace the function 1 (electron density) with function 100 (user-defined function), then what we plotted will just correspond to $-z\rho^{(3)}_{ZZZ}$. The resulting map is shown below. You can try to further improve the graph quality by using the lots of options provided in Multiwfn.



Part 5: Rapidly export grid data of $\rho^{(3)}_{ZZZ}$ and $-z\rho^{(3)}_{ZZZ}$

In main function 13, you can find corresponding option used to export grid data in memory as cube file. In addition, as described in Section 5.2, Multiwfn can be run in silent mode. Therefore, it is possible to rapidly export cube file of $-z\rho^{(3)}_{ZZZ}$ and $\rho^{(3)}_{ZZZ}$. The input stream file for this purpose has been provided as *examples\hyperpolar\hyperdens.txt*, please open it to check its content. Assume that all relevant .wfx files have been placed in proper folder, then after running *Multiwfn < hyperdens.txt*, you will find *density.cub* and *-Zrho3zzz.cub* in current folder, they correspond to cube file of $\rho^{(3)}_{ZZZ}$ and $-z\rho^{(3)}_{ZZZ}$, respectively.

Part 6: Obtain atomic contribution to (hyper)polarizability

Thanks to the flexibility of Multiwfn, one can easily obtain atomic contribution to (hyper)polarizability by integrating corresponding (hyper)polarizability density in each atomic space based on its grid data. Commonly, employing fuzzy atomic space is recommended, because the cost is very low. Here we will calculate atomic contributions to γ_{ZZZ} by integrating grid data of $-z\rho^{(3)}_{ZZZ}$.

Set "iuserfunc" in settings.ini to -1, so that the user-defined function will correspond to the interpolated function based on the loaded grid data. Then boot up Multiwfn, load the cube file corresponding to $-z\rho^{(3)}_{ZZZ}$ into it, then input below commands:

```
15 // Fuzzy atomic space analysis
1 // Integrate a real space function in every atomic space
100 // User-defined function
```

The result is

Atomic space	value	% of sum	% of sum abs
1(C)	80.42330427	5.643648	5.643648
2(H)	428.30264522	30.055833	30.055833
3(H)	428.19625953	30.048367	30.048367
4(O)	488.10117815	34.252152	34.252152

Summing up above values: 1425.02338717

The values under "Value" label are atomic contributions to γ_{zzzz} , and the values under the "% of sum" are percentage contributions. From the data it is clear that the two hydrogens and the oxygen have major contribution to γ_{zzzz} . The sum of all values is 1425.02 a.u., which is also quite close to the γ_{zzzz} value (1455.71 a.u.) outputted by Gaussian *polar=gamma* task.

Appendix

Below expressions are given so that you can also easily evaluate polarizability density and the first hyperpolarizability density by following above mentioned steps.

Polarizability density

$$\rho_z^{(1)} = \frac{\rho(F^z) - \rho(-F^z)}{2F^z}$$

$$\gamma_{zz} = \int -\rho_z^{(1)}(\mathbf{r}) z d\mathbf{r}$$

The first hyperpolarizability density

$$\rho_{zz}^{(2)} = \frac{\rho(F^z) - 2\rho(0) + \rho(-F^z)}{(F^z)^2}$$

$$\beta_{zzz} = \int -\rho_{zz}^{(2)}(\mathbf{r}) z d\mathbf{r}$$

4.A.8 Analyze wavefunction higher than CCSD level

In the most commonly used program Gaussian, the highest level of wavefunction is CCSD. Although CCSD wavefunction is absolutely sufficient for almost all cases, due to some special reasons, one may want to study wavefunction produced at higher level. Below I describe how to make Multiwfn able to analyze CCSD(T) wavefunction generated by PSI4 program (<http://www.psicode.org>), and arbitrary order of coupled cluster and CI wavefunction (including Full CI) yielded by Kallay's MRCC program (<http://www.mrcct.hu>).

(1) PSI4

The version of PSI4 I currently use is 1.2.1. Below is an example of input file, which calculate hydrogen fluoride at CCSD(T)/cc-pVTZ level, and produce *HF_CCSDpT.fchk* in current folder.

```
molecule HF {
    H      0.0      0.0     -0.831975
    F      0.0      0.0      0.092442
}
```

```
set basis cc-pVTZ
grad, wfn = gradient('CCSD(T)', return_wfn=True)
fchk_writer = psi4.FCHKWriter(wfn)
fchk_writer.write('HF_CCSDpT.fchk')
```

The resulting *HF_CCSDpT.fchk* records Hartree-Fock MOs and CCSD(T) density matrix. If you directly feed this file into Multiwfn, because Multiwfn never utilizes density matrix but only load orbitals from the file, the result of following analyses will correspond to Hartree-Fock level. In

order to make Multiwfn analyze CCSD(T) wavefunction, you should do below steps:

1. Boot up Multiwfn and load *HF_CCSDpT.fchk* as usual
2. Enter main function 200 and select subfunction 16. This function is used to transform density matrix in the .fch/.fchk file into natural orbitals, see Section 3.200.16 for more detail.
3. Input *CCSD*, then “Total CCSD Density” field in the .fchk file will be loaded, and you will immediately see occupation numbers of natural orbitals (NOs) yielded by diagonalization of CCSD(T) density matrix.
4. Input *y*. Then *new.molden* is generated in current folder, which records NOs at CCSD(T) level. This file is automatically loaded into Multiwfn, therefore the orbitals in memory now correspond to NOs of CCSD(T) wavefunction, and thus all following analyses will correspond to CCSD(T) wavefunction.

Note that if this is an open-shell system, you can choose the type of NOs that to be generated, including, spatial NOs, alpha/beta NOs and spin NOs. See Section 3.200.16 for more detail.

(2) MRCC

The version of MRCC I currently use is Sept 25, 2017. Below is an example of input file, which calculate hydrogen fluoride at CCSDT/cc-pVTZ level.

```

basis=cc-pvtz
calc=CCSDT
mem=2500MB
dens=1

geom=xyz
2

H      0.0      0.0     -0.831975
F      0.0      0.0      0.092442

```

After run it by MRCC, you will find a file named MOLDEN in current folder, which is a Molden input file and records Hartree-Fock MOs. In current folder you can also find a file named CCDENSITIES, which records the 2nd and 1st reduced density matrix (2RDM and 1RDM). In order to make Multiwfn analyze CCSDT wavefunction, we must convert the 1RDM into natural orbitals and save them to .molden file.

Change the name of MOLDEN to *e.g. mrcc.molden*, then boot up Multiwfn and load it. Enter main function 1000 and select subfunction 97, input path of the CCDENSITIES file. Then input the number of frozen core orbitals. By default, MRCC freezes core MOs in electron correlation calculation. Current system has two core electrons (in the output file it can be seen that " Number of core electrons: 2"), and this is a closed-shell system, each occupied MO has two electrons, therefore only one core MO is frozen, so we input 1. Soon, generation of natural orbitals by diagonalizing CCSDT density matrix is finished and occupation numbers are printed on screen. We use subfunction 2 of main function 100 to export a .molden file. then reboot Multiwfn and load it, now both the basis function information and GTF information in memory correspond to CCSDT wavefunction, therefore all the following analyses will be carried out at CCSDT level.

The procedure of analyzing CI wavefunction produced by MRCC is exactly the same as those shown above. Below is an example input file of calculating elongated LiH at FCI/aug-cc-pVDZ

level without frozen core treatment.

```

basis=aug-cc-pvdz
calc=fci
mem=2500MB
dens=1
core=0

geom=xyz
2

H      0.0      0.0      0.0
Li     0.0      0.0      3.0

```

4.A.9 Calculate TrEsp (transition charge from electrostatic potential) charges and analyze excitonic coupling

1 Theory about TrEsp

General form of electrostatic potential (ESP) of a molecule, say A , can be written as

$$\phi_{aa'}^A(\mathbf{r}) = \delta_{a,a'} \sum_I \frac{Z_I}{|\mathbf{r} - \mathbf{R}_I|} - \int \frac{\rho_{aa'}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}'$$

where Z_I and \mathbf{R}_I are nuclear charge and coordinate of atom I , respectively. δ is Kronecker function. $\rho_{a,a'}$ is transition density between state a and a' .

The ESP we commonly studied is ESP of a single state, i.e. $a=a'$. When a and a' correspond to different states, the potential may be referred to as "transition electrostatic potential", which measures ESP exerted by excitation corresponding to $a-a'$ transition.

It is known as exact ESP of a single state can often be well approximately represented as the potential evaluated based ESP fitting charges (e.g. CHELPG and MK charges, see Sections 3.9.10 and 3.9.11),

$$\phi_a^A(\mathbf{r}) = \sum_I \frac{Z_I}{|\mathbf{r} - \mathbf{R}_I|} - \int \frac{\rho_a(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \approx \sum_I \frac{q_a^I}{|\mathbf{r} - \mathbf{R}_I|}$$

where q_a^I is ESP fitting charge of atom I at electronic state a .

In light of this, *J. Phys. Chem. B*, **110**, 17268 (2006) puts forward the concept of TrEsp (transition charge from electrostatic potential), and shows that exact transition ESP can be well approximated as below

$$\phi_{aa'}^A(\mathbf{r}) = - \int \frac{\rho_{aa'}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \approx \sum_I \frac{q_{aa'}^I}{|\mathbf{r} - \mathbf{R}_I|}$$

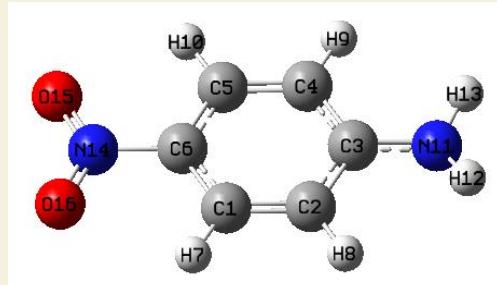
where $q_{aa'}^I$ is TrEsp of atom I derived from transition density of $a-a'$.

The way of calculation of TrEsp charges is almost exact the same as evaluation of common

ESP fitting charges, the only differences are that the nuclear contribution should be ignored, and density of a single state should be replaced with transition density between two states.

2 Example of calculating TrEsp charges

Now I use a simple molecule 4-nitroaniline to illustrate how to calculate TrEsp charges for its S0-S2 transition. Here I assume you are a Gaussian user (If you prefer to use ORCA program, the steps will be slightly lengthy, but can be significantly simplified via writing shell script. Please check #2 of this post for detail: <http://sobereva.com/wfnbbs/viewtopic.php?pid=389>).



First, run the Gaussian input file *examples\4-Nitroaniline_TrESP.gjf*, the keywords *PBE1PBE/6-31g(d) TD density=transition=2 out=wfn* mean transition density between ground state (S0) to S2 will be generated at TD-PBE0/6-31G(d) level, and then it will be automatically diagonalized to yield corresponding natural orbitals, which are finally saved to specified .wfn file. If you are confused or do not have Gaussian in hand, you can directly download related files from <http://sobereva.com/extrafiles/TrEsp.zip>

Boot up Multiwfn and input

```
S0S2.wfn // The .wfn file generated in above process
7 // Population analysis
12 // CHELPG fitting method (you can also use MK or RESP method instead)
5 // Choose form of ESP
3 // Transition electronic (i.e. the ESP specific for evaluating TrEsp)
1 // Start calculation
```

Calculation of ESP for even medium-sized systems is time-consuming, you need to wait patiently. Finally, the TrEsp charges are shown on the screen:

center	x	y	z	charge
1C	-0.006197	0.042893	2.284588	0.213411
2C	-0.006197	-2.564075	2.283674	-0.167947
3C	-0.004195	-3.920061	0.000000	0.184157
...[ignored]				
14N	0.001111	4.086240	0.000000	-0.064630
15O	0.003902	5.154581	-2.036539	-0.094910
16O	0.003902	5.154581	2.036539	-0.094899
Sum of charges: -0.000000				
RMSE: 0.000970 RRMSE: 0.044908				

The sum of charges is exactly zero, which is what we expected, because electronic transition process does not alter total number of electrons.

Beware that, the above outputted TrEsp charges **must then be manually divided by $\sqrt{2}$!** This is because the natural orbitals in the exported .wfn file were generated based on symmetrized form

of transition density matrix (TDM), however the symmetrization was done via a strange way by Gaussian, namely $TDM_{i,j} = (TDM_{i,j} + TDM_{j,i})/\sqrt{2}$ rather than $TDM_{i,j} = (TDM_{i,j} + TDM_{j,i})/2$ as expected, therefore this problem should be manually fixed via dividing the resulting charges by $\sqrt{2}$.

In fact, in Multiwfn the transition charge can also be calculated by Mulliken method via hole-electron analysis module, see Section 3.21.1.3, and the computational cost is almost negligible. However, Mulliken transition charges must not be as good as TrEsp charges for approximately representing transition electrostatic potential and analyzing intermolecular excitonic coupling purposes.

Skill 1: Accelerating calculation of TrEsp by making use of cubegen utility

Making use of cubegen utility in Gaussian package can significantly decrease cost of ESP relevant analyses in Multiwfn and is thus highly recommended, please read Section 5.7 for detail. The cubegen can also be used to significantly reduce computational time of TrEsp charges, as will be shown below.

Since cubegen calculate ESP based on density matrix information in .fch file, we must first generate TDM and store it into a .fch file, the function mentioned in Section 3.21.9 can do this. We first use *PBE1PBE/6-31g(d) TD IOp(9/40=4)* keywords to carry out electron excitation calculation and meantime keep the corresponding .fch, the corresponding files for 4-Nitroaniline are *4-Nitroaniline.IOp.gjf*, *4-Nitroaniline.IOp.out* and *4-Nitroaniline.fchk* in the aforementioned *TrEsp.zip* package.

Boot up Multiwfn and input

4-Nitroaniline.fchk

18 // Electron excitation analysis

9 // Generate and export TDM

1 // Generate TDM between ground state and excited state

4-Nitroaniline.IOp.out

2 // Generate TDM between S0 and S2

y // Symmetrize the resulting TDM in usual way, namely $TDM_{i,j} = (TDM_{i,j} + TDM_{j,i})/2$

*y // Export *TDM.fch*, whose density matrix field corresponds to the just generated TDM*

Please make sure that *cubegenpath* parameter in settings.ini has been set to actual path of cubegen utility in Gaussian folder, then reboot Multiwfn and input

TDM.fch

7 // Population analysis

12 // CHELPG fitting method

5 // Choose form of ESP

3 // ESP specific for evaluating TrEsp

1 // Start calculation

Since cubegen calculates ESP quite fast, immediately the TrESP charges are shown on screen. You do not need to manually divide the result by $\sqrt{2}$, because the TDM generated by Multiwfn has been symmetrized in correct way.

It is worth to note that if you want to verify whether the fitted TrEsp charges are reasonable, you can compare the electric dipole moment computed via these charges and the transition electric dipole moment printed by Gaussian (or other quantum chemistry codes). As it is well known that ESP fitting charges are able to well reproduce electric dipole moment, commonly the TrEsp charges

are also able to well reproduce actual electric transition dipole moment.

After calculation of TrEsp, we choose *y* to let Multiwfn export the charges to *TDM.chg* file in current folder. Then boot up and load this file, you will find below information on screen

Component of electric dipole moment:

```
X= -0.011607 a.u. ( -0.029501 Debye )
Y= -1.781254 a.u. ( -4.527495 Debye )
Z= 0.000011 a.u. ( 0.000029 Debye )
```

In *4-Nitroaniline_IOP.out* you can find below information

Ground to excited state transition electric dipole moments (Au):

state	X	Y	Z	Dip. S.	Osc.
1	-0.0000	-0.0000	0.0001	0.0000	0.0000
2	-0.0165	-1.7911	0.0000	3.2083	0.3408
3	0.0210	0.0188	0.0000	0.0008	0.0001

Since the electric dipole moment evaluated based on our TrEsp charges is very close to the exact transition electric dipole moment, it is clear that our TrEsp charges must be reasonable.

Skill 2: Imposing customized charge constraint during TrEsp fitting process

The restrained electrostatic potential (RESP) module of Multiwfn has been detailedly introduced in Section 3.9.16. As you can see, this module is more general and more powerful than the MK or CHELPG module, because you can arbitrarily impose customized constraints on the resulting ESP fitting charges, for example, you can request some atoms must have exactly the same charge, or request sum of a batch of charges must equal to a predefined value.

Here, I present an example to illustrate how to use the RESP module to calculate TrEsp based on MK fitting grid with additional constraint that atomic charge of all hydrogens must be zero.

Firstly, write a plain text file (*e.g. chgcons.txt*) with below content:

```
7 0
8 0
9 0
10 0
```

This file will be used in the RESP module. The 7~10 are atom indices of the hydrogens, the 0 means their charges will be constraint to zero during fitting.

Boot up Multiwfn and input

```
SOS2.wfn // The .wfn file we previously used
7 // Population analysis
18 // RESP module
11 // Choose form of ESP
3 // Transition electronic
6 // Set charge constraint in one-stage fitting
1 // Load charge constraint setting from external plain text file
chgcons.txt // The file containing charge constraint
2 // Start one-stage ESP fitting calculation with customized constraint. The default fitting grid
is MK (you can also change to CHELPG by option 3)
```

The result is

```
Center Charge
```

```

1(C )  0.158613
...
7(H )  0.000000
8(H ) -0.000000
9(H )  0.000000
10(H ) -0.000000
11(N )  0.178112
...

```

Clearly, our charge constraints have been in effect, and all other atoms still have reasonable TrEsp charges. You can learn more about the RESP module by reading corresponding example in Section 4.7.7. It is worth to note that when you select “Transition electric” in option 3, the default atom equivalence constraint is automatically removed and the restraint strength in one-stage fitting is automatically set to zero, since these treatments are not useful in current case.

3 Evaluating excitonic coupling energy based on TrEsp

General form of intermolecular Coulomb interaction energy can be expressed as

$$V_{aa',bb'}^{A,B} = \delta_{a,a'}\delta_{b,b'} \sum_{I \in A} \sum_{J \in B} \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} + \int \int \frac{\rho_{a,a}^A(\mathbf{r}) \rho_{b,b}^B(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r} d\mathbf{r}' \\ - \delta_{a,a'} \sum_{I \in A} \int \frac{Z_I \rho_{b,b}^B(\mathbf{r})}{|\mathbf{R}_I - \mathbf{r}|} d\mathbf{r} - \delta_{b,b'} \sum_{J \in B} \int \frac{Z_J \rho_{a,a}^A(\mathbf{r})}{|\mathbf{R}_J - \mathbf{r}|} d\mathbf{r}$$

This quantity may have different physical meaning. For example

$V_{00,00}^{A,B}$: Coulomb interaction energy between A and B in their ground states

$V_{00,11}^{A,B}$: Coulomb interaction energy between A in ground state and B in the first excited state

$V_{01,10}^{A,B} = V_{10,01}^{A,B}$: Excitation energy transfer couplings between transition of the two molecules

Calculation of the integrals in $V_{aa',bb'}^{A,B}$ is difficult, there is a method known as transition density cube (TDC), which calculates the integrals by numerical integration based on evenly distributed grids, its cost is extremely high for large system. Fortunately, it was shown that by using TrEsp charges calculated for two molecules, their excitonic coupling energy $V_{aa',bb'}^{A,B}$ can be readily evaluated using below formula at commonly satisfactory accuracy:

$$V_{aa',bb'}^{A,B} \approx \sum_{I \in A} \sum_{J \in B} \frac{q_{a,a}^I q_{b,b'}^J}{|\mathbf{R}_I - \mathbf{R}_J|}$$

In Multiwfn, you can easily calculate excitonic coupling energy based on TrEsp charges of two molecules. The steps are briefly outlined below:

(1) Optimize dimer structure

(2) Extract coordinate of each monomer and write it into Gaussian input file, properly change keywords and perform electron excitation calculation to yield .wfn file containing natural orbitals derived from transition density. Note that *nosymm* keyword must be used to avoid Gaussian

automatically translating and rotating the overall monomer coordinate.

(3) Generate TrEsp charges for each monomer using respective .wfn file generated at last step, then export TrEsp charges as .chg file.

(4) Manually combine content of the two monomer .chg file as a single .chg file. The monomer coordinate in this file should be consistent with optimized dimer coordinate.

(5) Load the dimer .chg file into Multiwfn, enter main function 7 and select option -2, then input atom list of the two monomers in turn, the excitonic coupling energy will be printed immediately.

4.A.10 Intuitively exhibiting atomic properties by coloring atoms

There is a very useful way of intuitively exhibiting atomic properties calculated by Multiwfn, namely coloring atoms in VMD program (<http://www.ks.uiuc.edu/Research/vmd/>), here I illustrate how to do that via two examples. More detailed discussions and examples can be found in my blog article "Using Multiwfn+VMD to exhibit atomic charges, spin populations, charge transfer and condensed Fukui function via coloring atoms" (<http://sobereva.com/425>).

(1) Coloring atoms according to atomic charges

First, I illustrate how to use this manner to vividly represent atomic charges of polyyne. This system was also involved in Section 4.13.6.

The first step is calculating atomic charges. Boot up Multiwfn and input below commands:

examples\polyyne.wfn

7 // Population analysis

11 // ADCH charge (this type of charge is generally recommended)

1

y // Export atomic coordinates and atomic charges to polyyne.chg in current folder

Now reboot Multiwfn, then input

polyyne.chg

100 // Other function (part 1)

2 // Export new file

I // The format of the new file is .pqr

polyyne.pqr

Now we have polyyne.pqr in current folder. The .pqr format is very similar to the popular .pdb format, the major difference is that in the .pqr format the last two columns are specific for recording atomic charges and atomic radii, respectively. In current file, the atomic charges correspond to the polyyne ADCH charges, while the atomic radii correspond to Bondi van der Waals radii.

The .pqr file can be recognized by VMD. We boot up VMD, then drag the polyyne.pqr into VMD main window to load it. After that, we modify plotting settings:

(a) Use white background: Inputting *color Display Background white* in VMD console window

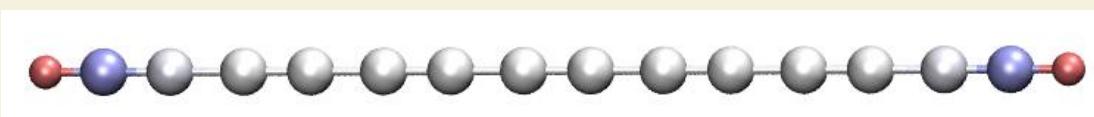
(b) Modifying drawing style and coloring setting: Enter "Graphics" - "Representation", set the drawing method as "CPK", set the coloring method as "Charge". Then choose "Trajectory" tab, input -0.4 and 0.4 in the "Color Scale Data Range" text boxes and press ENTER button.

(c) Changing color transition style: Enter "Graphics" - "Colors", choose "Color Scale" tab,

change the default RWB to BWR (Blue-White-Red)

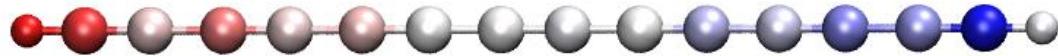
(e) Choose "Display" - "Orthographic" to use orthographic perspective.

Now you will see below graph in VMD OpenGL window. The atoms at both ends are hydrogen, all the other atoms are carbon.



In above graph, the red and blue colors reflect that the atom has positive and negative charge, respectively. The deeper red (blue) the more positive (negative) the charge. As can be seen, since carbon has larger electronegativity than hydrogen, the two hydrogens have evident positive charge and the carbons bonded to them have evident negative charge. The white color indicates that the charge of the carbons in the middle region of the molecule is close to zero.

The polyyne is a highly conjugated system, it is expected that external field could significantly polarize its charge distribution. To study this problem, we use the same procedure to plot the map based on *examples\polyyne_field.wfn*, which was generated under 0.03 a.u. external electric field along molecular axis. The resulting graph is shown below, the direction of the external electric field is from the right side to the left side.



It can be seen that the atomic charge distribution is no longer symmetric. Since the source of the field is at right side, large amounts of electrons transferred from left to right, as a result, the net charge of the carbon atoms at left side become positive, while the ones at right side become negative.

(2) Coloring atoms according to atomic contribution to molecular orbitals

The atomic coloring method is not only able to be employed to exhibit atomic charges, but can also be used to exhibit other atomic properties. As an example, I illustrate how to represent atomic contributions to molecular orbitals by coloring atoms, *examples\N-phenylpyrrole.fch* is taken as example molecule.

First, we calculate orbital composition of an orbital. Boot up Multiwfn and input

examples\N-phenylpyrrole.fch

8 // Orbital composition

3 // SCPA method

36 // Select MO 36 as example

Then we copy all atomic contributions from the Multiwfn window to a text file using the method described in Section 5.4.

0 // Return

-10 // Return to main menu

100 // Other function (part 1)

2 // Export new file

2 // The format of the new file is xyz, because .xyz is very similar to .chg

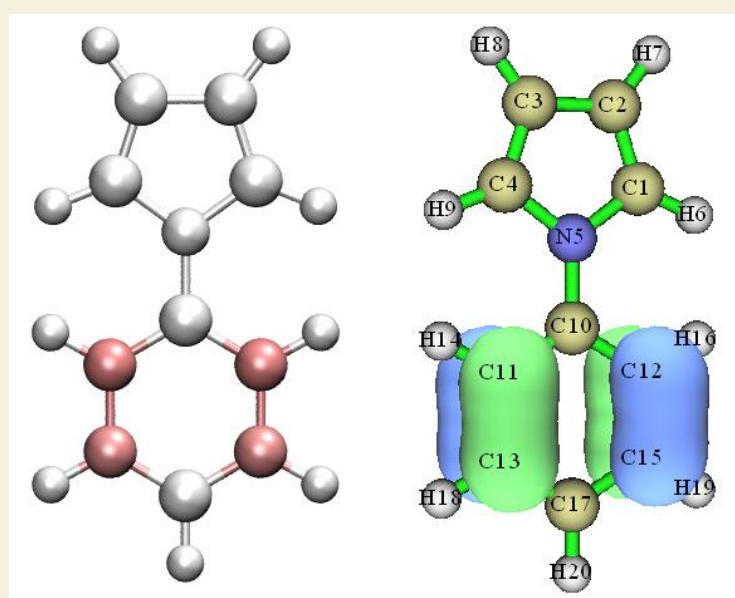
N-phenylpyrrole.chg // Name of the new file

Now manually modify the *N-phenylpyrrole.chg* by your favorable text editor (Ultraedit is recommended), delete the first two lines, and copy the orbital composition to the last column using

column mode, then save the file. Finally, the content of the *N-phenylpyrrole.chg* should be

```
C      -0.00000000   1.12162908   1.82507914  0.396870
C      -0.00000000   0.71310006   3.13457424  0.099898
C      -0.00000000  -0.71310006   3.13457424  0.099898
...[ignored]
H      0.00000000   2.14896316   -3.05414223  0.000000
H      0.00000000  -0.00000000   -4.31975533  0.000000
```

We use Multiwfn to load this .chg file and convert it to *N-phenylpyrrole.pqr*, then use VMD to visualize it using exactly the same procedure described in last example. However, this time the lower and upper limit of color scale should be set to -50 and 50, respectively. The resulting graph is shown as left part of below map; as a comparison, the corresponding isosurface map of MO36 is shown at right part.



The more red the atom, the greater its contribution to the orbital. As can be seen, the atomic coloring introduced in this section well reflects actual orbital distribution. For very large molecules, the isosurface map may become quite complicated, while the atomic coloring map should be much clearer.

Of course, the atomic coloring method is also applicable to other kinds of atomic properties calculated by Multiwfn, such as condensed Fukui function, atomic spin population, atomic transition charge, source function of atoms, integral of electron energy in atomic space, variation of atomic charge during electronic transition or intermolecular interaction. More examples can be found in my blog article <http://sobereva.com/425> (in Chinese).

4.A.11 Overview of methods for studying chemical bonds

PS: Chinese version of this Section is <http://sobereva.com/471>, which also contains extended discussion.

In this section, I present an overview of all methods that may be used to study chemical bonds. You will find Multiwfn is indispensably useful in characterizing and unraveling nature of the bonds.

Most analyses can be applied to both ground state and excited states (see Section 4.18.13 for more information about this point)

1 AIM (Atoms-in-molecules) analysis

In the framework of AIM, the bond critical point (BCP) is the most representative point of a bond, hence character of chemical bonds can be characterized by various properties at corresponding BCPs, for example:

- Electron density and potential energy at BCP, namely $\rho(\text{BCP})$ and $V(\text{BCP})$, are often used to discuss bonding strength. For the same kind of bond, they are usually positively and negatively correlated to bonding strength, respectively.
- Laplacian of electron density at BCP, namely $\nabla^2\rho(\text{BCP})$, is often used to judge whether or not a bond mainly shows covalent character. Negative and positive values imply that the major nature of the bond is covalent and non-covalent, respectively. But notice that this criterion is often wrong (e.g. CO has positive $\nabla^2\rho(\text{BCP})$ but it is evidently a polar covalent bond)
- In *Angew. Chem. Int. Ed. Engl.*, **23**, 627 (1984) it was argued that negative and positive values of energy density at BCP, *i.e.* $H(\text{BCP})$, implying the bond has covalent and non-covalent nature, respectively. But this criterion is not always true; for example, the Ca-O in CaO is typical ionic bond, but its $H(\text{BCP})$ is negative.
- The $V(\text{BCP})/G(\text{BCP})$ was proposed in *J. Chem. Phys.*, **117**, 5529 (2002), where $G(\text{BCP})$ denotes Lagrangian kinetic energy density at BCP. It was argued that $<0, >1$ but $<2, >2$ of this quantity respectively imply that the bonding mainly belongs to close-shell interaction, intermediate (mixed) interaction and covalent interaction.
- The eta index was proposed in *J. Phys. Chem. A*, **114**, 552 (2010) and further studied in *Angew. Chem. Int. Ed.*, **53**, 2766 (2014), it is defined as $|\lambda_1(\mathbf{r})|/\lambda_3(\mathbf{r})$, where λ_1 and λ_3 are the smallest and largest eigenvalues of Hessian matrix of electron density, respectively. It was argued that if eta index at BCP is smaller than 1, then the bonding should be closed-shell interaction; while if it is larger than 1, the interaction should has covalent nature, and the more positive the value, the stronger the covalent character. However I found this argument is not always true, for example this quantity of both Ni-C and C-O bonds in Ni(CO)₄ is less than 1, but undoubtedly they should be attributed to polar covalent bonds.
- Bond degree (BD) was proposed in *J. Chem. Phys.*, **117**, 5529 (2002) and defined as $H(\text{BCP})/\rho(\text{BCP})$. The physical meaning of BD is energy density of unit electron at BCP. For covalent interaction (usually $H(\text{BCP})<0$), the more negative the BD, the stronger the bonding; while for non-covalent interaction (usually $H(\text{BCP})>0$), the more positive the BD, the weaker the interaction.
- Bond ellipticity was proposed in *J. Am. Chem. Soc.*, **105**, 5061 (1983) and defined as $\varepsilon(\mathbf{r})=[\lambda_1(\mathbf{r})/\lambda_2(\mathbf{r})]-1$. At BCP the λ_1 and λ_2 must be negative and exhibit the curvature of electron density perpendicular to the bond. The larger the value deviates to 0, the stronger the tendency that the electron density has unsymmetric distribution in the plane perpendicular to the bond at the BCP.
- The source function with BCP as reference point has been employed to study chemical bonds, see *Struct. & Bond.*, **147**, 193 (2010) for comprehensive review and Section 4.17.5 for analysis example.

There are some other research papers utilized BCP properties to discuss chemical bonds, for example *J. Am. Chem. Soc.*, **120**, 13429 (1998) and *J. Comput. Chem.*, **39**, 1697 (2018). It is worth to note that properties at certain critical points can also be used to estimate metallicity of crystals,

see *J. Am. Chem. Soc.*, **124**, 14721 (2002), *Chem. Phys. Lett.*, **471**, 174 (2009) and *J. Phys.: Condens. Matter*, **14**, 10251 (2002).

The bond path is also a very important concept in the AIM framework, it rigorously reveals the main interaction path connecting various atoms. It is important to understand the fact that chemical bond must be accompanied by a bond path and BCP, while present of bond path and BCP does not imply existence of chemcal bond.

AIM topology analysis has been systematically introduced in Section 3.14 and illustrated in Section 4.12.1, all above mentioned quantities can be easily and rapidly evaluated by Multiwfn. It is noteworthy that in Multiwfn many real space functions at critical point (or specific point) can be decomposed as contributions from various orbitals (usually MOs or LMOs), see Section 4.2.4 for example; moreover, any real space function can be plotted along bond paths, see Section 4.2.3 for example. These useful features can often provide much deeper insight into the bonding.

The above mentioned real space functions can also be plotted as curve map, plane map or isosurface map so that one can visually study their distribution, see Section 4.3, 4.4 and 4.5 for practical example, respectively. The contour line map and isosurface map of $\nabla^2\rho(\mathbf{r})$ is particularly useful and frequently employed.

2 Bond order and delocalization index analysis

Bond order is a very useful and straightforward way of characterizing chemical bonds. Multiwfn supports a lot of bond order definitions, please check Section 3.11 for detailed introduction. Different bond orders have different characters and physical meanings. For example, Laplacian bond order (LBO) measures covalent component of a bond and usually has good relationship with bond dissociation energy (BDE), while Mayer bond order essentially reflects the number of electrons shared by two interacting atoms. The bond analysis module of Multiwfn is also able to do more things than just calculating the value of bond order. For example, Multiwfn can decompose some bond orders as contributions from various orbitals, the Wiberg bond order can be decomposed as contributions from atomic orbital pairs. Many detailed analysis examples of bond order are given in Section 4.8.

It is worth to mention that the delocalization index (DI) is essentially equivalent to Mayer bond order and fuzzy bond order in physical nature. The difference arises from how the atomic spaces are defined. For non-polar bonds, the DI is usually very close to Mayer and Fuzzy bond orders, but they quantitatively differ for polar bonds. For DI, the AIM atomic basins are employed as atomic spaces. The DI can be calculated via basin analysis module, see Section 3.18.5 for detailed introduction of DI and Section 4.17.1 for DI analysis example. Commonly I do not suggest employing DI, because its cost is by far higher than evaluation of Mayer and Fuzzy bond orders.

For the same kind of chemical bond, e.g. C-O bond in different transition metal coordinates, the Mayer bond order and LBO are positively correlated with bonding strength. For different kinds of chemical bonds, Mayer bond order should not be used to compare bonding strength, for example, the bond in N₂ and that in P₂ has evidently different BDEs, but they have basically the same Mayer bond order because both of them are typical triple bonds. In contrast, the LBO is able to faithfully reflect that the bond in P₂ is much weaker than N₂. Please check original paper of LBO (*J. Phys. Chem. A*, **117**, 3100 (2013)) for more discussions, comparisons and examples.

Plotting variation of bond order versus reaction coordinate is an absolutely very good idea to shed light on the underlying change of electronic structure in a chemical reaction, see Section 4.A.1 on how to easily realize this.

3 Orbital localization analysis

Molecular orbitals (MOs) commonly are unable to be used to study bonding character because they are highly delocalized and do not directly correspond to chemical bonds. The orbital localization is a very powerful technique, it can transform the MOs to localized molecular orbitals (LMOs), which are highly localized and have very close relationship with bonding. Via LMOs, numerous useful information about chemical bonds can be extracted, such as bond polarity, bond multiplicity, bond type, the atomic orbitals that participate in the bonding and so on. Please check Section 3.22 for detailed introduction of LMOs and follow the LMO analysis examples in Section 4.19.

4 AdNDP analysis

The purpose of the adaptive natural density partitioning (AdNDP) method is somewhat similar to orbital localization method, the advantage of AdNDP is that it is also able to derive orbitals with semi-delocalization character from the complicated multi-electron wavefunction. If the AdNDP analysis has been properly performed, then the resulting orbitals will faithfully reveal all multi-center bonds in current systems. The drawback of AdNDP analysis is that user must manually pick out orbitals from candidate list, this process is slightly cumbersome and requires the user has adequate chemical intuition. When there is no multi-center bond, using orbital localization is much preferred over AdNDP because it is fully automatic, fast and free of subjectivity; while if you suspect that present system may have evident mulit-center bonds and you want to study them, commonly AdNDP is the only choice. The AdNDP method is detailedly introduced in Section 3.17, related examples are provided in Section 4.14.

5 Analysis of ELF and relevant real space functions

ELF is a very important real space function, it is able to reveal localization and delocalization of electrons in chemical systems. Brief introduction of ELF can be found in Section 2.6. In Multiwfn ELF can be analyzed in many different ways, as shown below

- Visualizing study. In Multiwfn, the ELF can be drawn as curve map by main function 3, drawn as plane map by main function 4 and plotted as isosurface map via main function 5, see Sections 4.3, 4.4 and 4.5 for examples, respectively. From the graphs one can easily identify which region contains evident covalent interaction (*i.e.* evident share of electrons) and judge the nature of a given chemical bond. In addition, the bond multiplicity can be inferred from the shape of ELF isosurface around the bond. Multiwfn is also capable of studying ELF- π and ELF- σ , so that π interaction and σ interaction can be studied separately, see Sections 4.5.3 and 4.100.22 for example.

Note that there are a lot of real functions having analogous distribution feature as ELF, though their underlying ideas may be not very similar to ELF. Multiwfn supports most of them and they can also be plotted in the exactly the same way as ELF. These real space functions include LOL, SCI, SEDD, RoSE, PS-FID and DORI. The LOL is introduced in Section 2.6 and sometimes preferred over ELF; introduction of other real space functions can be found in Section 2.7. Section 4.20.5 presented an example of analyzing the DORI function, the special feature of DORI compared to ELF is that DORI is able to reveal all kinds of interactions, including both covalent and non-covalent ones.

The negative part between two atoms in $\nabla^2\rho$ map is able to reveal the region where electrons concentrate due to formation of covalent bond, this point is similar with ELF. In *J. Phys. Chem.*, **100**, 15398 (1996), Bader believes that $\nabla^2\rho$ and ELF are homeomorphic and their similarities and differences are able to provide complementary information in understanding chemical bonds.

However, notice that for bonds involving very heavy atom, $\nabla^2\rho$ map often fails to reveal covalent character. For example, $\nabla^2\rho$ in the interacting region of Re-Re bond is entirely positive.

By using Multiwfn and shell script as well as third-part software, anime of ELF or other functions during a chemical process (often represented as trajectory resulting from intrinsic reaction coordinate or rigid scan tasks) can be easily generated, such an anime is able to very vividly exhibit variation of characters of chemical bonds, see Section 4.A.1 on how to make the anime.

- Basin analysis of ELF (or similar functions): This kind of analysis can be carried out via basin analysis module (main function 17), see Section 4.17.2 for example. All ELF basins collectively make up the whole space, each ELF basin corresponds to a region with featured electronic structure. For example, the ELF basins may correspond to covalent bond, lone pair, core region, etc. By analyzing character of bond basins, one can acquire many information about the bonds, such as average number of electrons that occurs in the bonding region, degree of electron localization in the bonding region, dipole moment of the bonding region. Contribution of each atom to the electron population in the bonding region can also be obtained, as illustrated in Section 4.17.7.

- Topology analysis of ELF (or similar functions): This kind of analysis allows one to obtain accurate position of ELF maximum (also known as ELF attractor) and (3,-1) type of ELF critical point (also known as ELF bifurcation point), the former displays the most representative point of a ELF basin, while the value of the latter somewhat reflects the extent of electron sharing (degree of delocalization) between two ELF basins. Topology analysis of ELF can be realized via main function 2, see Section 4.2.2 for example. For practical studies based on topology analysis of ELF and LOL, see *Nature*, **371**, 683 (1994) and *J. Comput. Chem.*, **30**, 1093 (2009). Tracing variation of ELF attractors is particularly useful for understanding the change in electronic structure and bonding character during a chemical process, illustrative examples of such analysis are *RSC Adv.*, **5**, 62248 (2015), *Chem. Phys.*, **501**, 128 (2018) and *Comput. Theor. Chem.*, **1154**, 17 (2019).

Note that basin analysis is also able to give positions of ELF attractors, the procedure is even more simple than using topology analysis module, however the accuracy of the positions given by basin analysis module is not as good as topology analysis module, since basin analysis is carried out based on even-distributed grids.

6 Analysis of valence electron density

As clearly illustrated in my article [*Acta Phys. -Chim. Sin.*, **34**, 503 \(2018\)](#), visualizing electron density of valence electrons is a very useful way of revealing electron structure and studying character of chemical bonds, please carefully read this paper. Moreover, basin analysis can be applied to valence electron density to unveil more information of chemical interest. See Section 4.6.2 for example on how to carry out this kind of analysis.

7 Electron density difference analysis

Formation of chemical bond always leads to significant electron reorganization (polarization and charge transfer), in particular, formation of covalent bond must be accompanied with the phenomenon that electrons concentrate to the bonding region. Plotting electron density difference (EDD) map is one of the best ways to reveal this point, EDD can be very easily plotted as curve map, plane map and isosurface map in Multiwfn via mainfunctions 3, 4 and 5, respectively. EDD can be defined in different ways, if you want to study the bond formed between two fragments, you should study the EDD between the whole system and the two fragments, see Section 4.5.5 for example; if you want to study the reorganization of electron density due to forming bonds between various atoms in the system, you should study deformation density, which is defined as the difference between the electron density of the whole system and all atom in their isolated states, see Section 4.4.7 for example.

Do not forget that Multiwfn also provides advanced technique for analyzing EDD, for example, basin analysis can be applied to EDD, see Section 4.17.4 for example. Also one can plot charge displacement curve to better quantitatively study electron reorganization along specific direction, see Section 4.13.6 for example.

It is worth to note that plotting difference map of ELF between whole system and its fragments is also valuable, see illustration in Section 4.4.8.

8 Analysis of δg function

The real space function δg is defined in the framework of IGM theory, see Section 3.23.5 for introduction. δg is capable of revealing all kinds of interactions, including both chemical bonding and weak interaction, as well as both covalent and non-covalent ones. Moreover, the magnitude of δg in the bonding region is often positively correlated with bonding strength, therefore one can easily examine bonding strength in different regions by inspecting colors in color-filled map or by properly adjusting isovalue in isosurface map. In addition, the isosurface of δg can be mapped by $\text{sign}(\lambda_2)\rho$ function via various colors, this makes the isosurface map informative. Please check Section 4.20.10 for IGM examples, although the examples focus on studying weak interactions, the same procedure can also be migrated to chemical bond analysis.

9 Quantifying amount of charge transfer due to bonding

Formation of chemical bond between two different fragments must result in detectable charge transfer (CT) between the two fragments. The amount of CT can be obtained as difference between the fragment charge in actual system and the net charge of the fragment in its isolated state. The fragment charge is defined as sum of charges of the atoms in the fragment. In the population analysis module of Multiwfn, if a fragment has been defined, the fragment charge will be directly outputted when calculating atomic charges, see Section 4.7.1 for example.

10 Charge decomposition analysis (CDA)

Using fragment charge we can easily discuss the total amount of CT, however, in order to examine details of charge transfer due to bonding, the CDA must be employed. CDA is able to explicitly show electron donation and back-donation between each pair of user-defined fragments at resolution of orbital interactions, also it provides clear information about how the MOs of the entire system is composed of MOs of individual fragments. See Section 3.19 for introduction and Section 4.16 for example.

11 Density-of-states (DOS) analysis

The partial DOS (PDOS) curve map is useful for intuitively exhibiting bonding and anti-bonding due to interaction between user-defined fragments (may be defined as a batch of atoms, shells or atomic orbitals) at various energy range, see Section 3.12 for introduction and Section 4.10.1 for example.

12 Energy decomposition analysis

Energy decomposition analysis is used to decompose the bond energy to different physical components to provide deeper insight into the bonding nature. The “simple energy decomposition” supported by Multiwfn can be applied to chemical bonds, please check Section 4.100.8 for example. You need Gaussian to use this function.

13 Studying bond polarity

It is often interesting to study polarity of a bond, there are several possible ways, as shown

below. The results often differ significantly, since the concept itself cannot be uniquely defined.

- Calculating respective contribution from the two bonding atoms to the LMO corresponding to the bond (Θ_A and Θ_B), then the ionicity of the bond can be evaluated as $|\Theta_A - \Theta_B|$. Clearly the larger this value, the higher the bond polarity. To obtain Θ_A and Θ_B , you should first perform orbital localization, then find the LMO corresponding to the bond in main function 0, and finally use main function 8 to evaluate composition of the LMO via proper method.
- Firstly evaluating respective contribution from the two bonding atoms to the population number of the ELF basin corresponding to the bond, as illustrated in Section 4.17.7, then take the difference of the two contribution values to estimate the bond polarity.
- Calculating bond polarity index. See Section 3.200.12 for introduction and 4.200.12 for example.
- It is worth to note that Laplacian bond order only reflects covalent component of a bond, while Mayer bond order may be regarded as total bond order. Therefore, in certain cases, the difference between Laplacian and Mayer bond orders may be used to reveal bond polarity.

14 Studying bond dipole moment

There are three possible ways in Multiwfn:

- Calculate bond dipole moment based on two-center localized molecular orbitals, please check introduction in Section 3.22 and example in Section 4.19.4.
- Perform ELF basin analysis and check dipole moment of the basin corresponding to the bond of interest. See Section 4.17.2 for illustration. At the same time, quadrupole moment of the bond basin can also be obtained.
- Calculating bond dipole moment in Hilbert space. See Section 3.200.2 for introduction

15 PAEM (potential acting on one electron in a molecule) analysis

The PAEM refers to the total potential acting on an electron at a point. By analyzing PAEM at proper positions between two atoms, the nature (covalent or non-covalent) can be determined. See Section 4.3.3 for illustration.

4.A.12 Overview of methods for analyzing electron excitation

In this section, I present a systematic overview of all methods supported by Multiwfn that can be used for analyzing electron excitation problems.

Note: Chinese version of this section corresponds to my blog article <http://sobereva.com/437>.

1 Hole-electron analysis

All kinds of excitations can be essentially described as "hole-to-electron" transition, that is, "hole" is the region where the excited electron leaves, and "electron" is the region where the excited electron eventually goes. Hole-electron analysis corresponds to subfunction 1 of main function 18, see Section 3.21.1 for introduction and 4.18.1 for illustration. This analysis is very powerful and universal and is an almost indispensable analysis method for all kinds of electron excitation problems. Specifically, the hole-electron analysis has below capacities:

- Displaying isosurfaces of hole and electron. From this picture, one can intuitively understand how electrons are excited
- Transforming the hole and electron distributions to a form described by Gaussian function, making them significantly easier to examine visually

- Calculating quantitative indices that measure characteristics of electron excitation, including the S_r index, which measures the degree of overlap of hole and electron; the D index, which measures the distance between hole and electron centroids; the σ index, which measures the breadth of hole and electron distributions; the t -index, which measures degree of separation of hole and electron, and so on.
- Plotting density difference map, which corresponds to subtracting hole from electron
- Calculating contribution of basis functions, atomic orbitals, atoms, molecular fragments and molecular orbitals to hole and electron, so that the nature of hole and electron can be thoroughly analyzed. Moreover, amount of hole and electron as well as hole-electron overlapping extent on various atoms and fragments can be directly displayed as heat map (color-filled matrix map), which is very convenient for visual horizontal comparison.
- Calculating Coulomb attraction between hole and electron, which is a common definition of exciton binding energy.

2 Nature transition orbital (NTO) analysis

When doing electron excitation calculations, it is often found that many orbital transitions have negligible contribution to electron excitation, this phenomenon makes viewing orbitals to discuss electron excitation characteristics difficult, and in this case it is necessary to examine multiple orbitals simultaneously. After transforming the molecular orbitals to NTOs using subfunction 6 of main function 18, for most cases the electron excitation can be solely described by only one pair of NTO transition, thus making the discussion much simpler. See Section 3.21.6 for introduction of NTO analysis and Section 4.18.6 for practical example.

3 Λ index and Δr index

The Λ index proposed in 2008 may be the earliest index to quantitatively examine characteristic of electron excitations, its intrinsic physical meaning is a measure of degree of overlap between electron and hole. The Δr proposed in 2013 is another index for characterizing electron excitation based on the idea of Λ index. The Δr essentially measures the centroid distance between electron and hole. Λ and Δr are described in detail in Sections 3.21.14 and 3.21.4, and they can be computed via subfunctions 14 and 4 of main function 18, respectively.

In fact, with the S_r and D indices defined in the hole-electron analysis framework, it is no longer necessary to use the Δr and Λ indices, since S_r and D are in principle more significant in physical meaning. However, since Multiwfn is able to calculate Δr and Λ for a large number of selected excited states simultaneously, if you simply want to roughly examine electron excitation characteristics for a batch of excited states at once, employing Δr and Λ is still a good choice.

4 IFCT analysis

The full name of IFCT is "interfragment charge transfer", which is a method proposed by me to estimate amount of electron transfer between atoms or fragments in the process of electron excitation. The calculation cost is extremely low. This method has been detailed described in Section 3.21.8 and illustrated in Section 4.18.8. Although using difference between fragment charge of excited state and that of ground state can also study variation of electron population during electron excitation, one cannot understand details of charge transfer at "who transferred to whom" level, therefore IFCT analysis has important and irreplaceable practical value for investigating problems of electron excitation. In particular, when studying transition metal coordinates, exact amount of MC, LC, LLCT, MLCT and LMCT can be separately evaluated by the IFCT analysis.

5 Analysis based on density difference between excited state and ground state

Density difference analysis is a prevalently used and widely accepted method for studying the difference in charge distribution between two electron states of a system. Multiwfn supports a variety of analysis methods based on the density difference between excited state and ground state, as shown below:

- Plotting density difference map

First of all, Multiwfn can easily calculate density difference between excited state and ground state and plot it as curve map, plane map and isosurface map through main functions 3, 4, 5, see examples in Section 4.3, 4.4 and 4.5, respectively. Moreover, not only the density difference between excited state and ground state can be drawn, but also the density difference between two excited states can be easily drawn via Multiwfn, see illustration in Section 4.18.13.

- Smoothing density difference and calculating statistical data of density difference

The original density difference map between excited state and ground state is not easy to examine, because its positive and negative regions are interlaced and appear to be messy. After calculating the density difference grid data, one can use subfunction 3 of main function 18 to transform it to replace the positive and negative parts of the density difference with very smooth Gaussian functions, then the image will become much more intuitive and easier to analyze. At the same time, the program outputs various statistical data about the density difference, such as centroid coordinates of positive and negative parts, charge transfer distance, degree of separation between positive and negative parts. See introduction in Section 3.21.3 and example in Section 4.18.3.

- Local integral curve and charge displacement curve

If the system under studying is linear or an interface system (such as a dye molecule attached to TiO₂ surface), the local integral curve and charge displacement curve can be plotted along the direction of the molecular chain or perpendicular to the interface. The local integral curve shows the integral value of the density difference on each section perpendicular to the chosen direction, while the charge displacement curve shows the integral of density difference from the beginning side to the current position. These two kinds of maps are useful to quantitatively study electron transfer feature along a certain direction. It is easy to draw these two kinds of graphs in Multiwfn, please check Section 3.16.14 of the manual for introduction and Section 4.13.6 for example.

- Basin integration for density difference

Multiwfn is able to perform basin integration for density difference, so that one can study variation of number of electron in some featured local regions, see Section 4.17.4 for example.

6 Analyzing difference between excited state and ground state in electron population or atom/fragment charges

Main function 7 is used to perform population analysis or atomic charge calculation, and if a fragment is defined by subfunction -1 before evaluating atomic charges, fragment charge will also be given in the output. See corresponding examples in Section 4.7. After calculating the fragment charges of excited state and ground state separately, the difference between them can be used to understand how many electrons were lost or gained at different fragments during the electron excitation, and thus the influence of electron excitation on the charge distribution can be investigated at quantitative level.

Although IFCT analysis is able to realize the same purpose, the advantage of using atomic/fragment charge to discuss this problem is that there is a large room of choice of the method for evaluating atomic charges, and the charge distribution of excited state can correspond to relaxed

density.

7 Draw transition density isosurface map, plotting heat map of transition density matrix

The transition density matrix (TDM) is very useful for unveiling the underlying nature of an electron excitation. TDM has two forms: (1) The three-dimensional real space form, which can be expressed by drawing isosurface map. Large value at a point corresponds to large overlap of hole and electron at this place, see Section 3.21.1.1 for detailed introduction and Section 4.18.2.1 for analysis example. (2) The matrix form in the common sense. This form of TDM can be exhibited as heat map (namely color-filled matrix map), which may be atom-based or fragment-based. Its diagonal elements vividly show which atoms or fragments are simultaneously occupied by hole and electron, while the non-diagonal elements directly reflect the direction and extent of electron transfer between corresponding atoms or fragments. See Section 3.21.2 for introduction of the TDM heat map and Section 4.18.2.2 for analysis example.

8 Analyzing heat map of charge transfer matrix

If each atom is defined as a fragment in the aforementioned IFCT analysis, the amount of charge transfer between various atoms and the amount of charge redistribution within each atom will constitute a matrix, which is referred to as "atom-atom charge transfer matrix" by me and may be further contracted to fragment-fragment charge transfer matrix. Both of the matrices can be drawn as heat maps by subfunction 2 of main function 18, see Section 3.21.2 and 3.21.8 for introduction as well as Section 4.18.2.2 for practical example. The information carried by the charge transfer matrix heat map is very similar to the TDM heat map, and the way of analysis is exactly the same, but the charge transfer matrix is more strictly defined and has clearer physical meaning. Moreover, the charge transfer matrix is completely in agreement with the hole and electron distributions given by the hole-electron analysis module, therefore I believe the charge transfer matrix map analysis is a better method than the popular TDM heat map analysis.

9 Analyses on transition dipole moment

For absorption process, the larger the oscillator strength of an electron excitation, the stronger the corresponding absorption peak. The transition probability between two excited states is mainly determined by oscillator strength, which is proportional to square of corresponding transition electric dipole moment. Therefore, it is very meaningful to conduct an in-depth analysis on intrinsic factors affecting the transition electric dipole moment. Multiwfn provides a number of functions for decomposing transition dipole moment (including both the electric one and magnetic one), as described below.

- Drawing transition dipole moment density

The transition dipole moment density is a function that measures the contribution of a point in the three-dimensional space to the transition dipole moment, and its integral over the whole space is exactly equal to the transition dipole moment. Obviously, if the transition dipole moment density is plotted as isosurface map or plane map, contribution of each region to the transition dipole moment can be vividly exhibited. See Section 3.21.1.1 for introduction and Section 4.18.2.1 for example.

- Drawing heat map of transition dipole moment matrix

Subfunction 2 of main function 18 can draw heat map of transition dipole moment matrix, either atom-based or fragment-based. The sum of all matrix elements is exactly the transition dipole moment of the system, so the diagonal elements in the map show the contribution of atoms or

fragments to the transition dipole moment solely by themselves, while the non-diagonal elements reflect the atom-atom or fragment-fragment coupling contribution to transition dipole moment. Clearly, the internal structure of transition dipole moment can be clearly understood via this kind of heat map. See Section 4.18.2.2 for analysis example.

- Decomposing transition dipole moment to basis function contributions and atom contributions

Subfunction 11 of main function 18 can decompose transition dipole moment into contribution of each atom and each basis function, see Section 3.21.11 for details. Moreover, based on the data outputted by Multiwfn, via a VMD script one can draw arrows to exhibit contribution vector of custom fragments to transition dipole moment, so that the contribution of each part of the system to the transition dipole moment can be intuitively understood, see Section 4.18.11 for example.

- Decomposing transition dipole moment into contributions of orbital transitions

Subfunction 10 of main function 18 can decompose transition dipole moment into contribution of each orbital transition, and at the same time the program outputs the oscillator strength evaluated based on the current electron excitation information. Therefore, when many orbitals have significant participation in electron excitation, this function can be used to immediately identify which orbital transitions have critical influence on the oscillator strength, so that further discussion can be made. In addition, one can set configuration coefficients of some orbital transitions to zero in subfunction -1 of main function 18, and then enter this function again to check influence on the oscillator strength due to ignoring those orbital transitions. See corresponding introduction in Section 3.21.10 and example in Section 4.18.10.

- Calculating transition dipole moments between excited states and dipole moment of each excited state

Transition dipole moments between excited states are important for some studies. For example, they are needed by the sum-over-states (SOS) method, which can be used to calculate (hyper)polarizability (see Section 3.200.8); in addition, simulating transient absorption spectrum needs oscillator strength (f) between excited states, while evaluation of f requires transition dipole moment between corresponding two excited states. In Multiwfn, subfunction 5 of main function 18 can evaluate transition dipole moments between excited states, and dipole moments of each excited state can also be directly outputted. See Section 3.21.5 for details about this function.

10 Analyzing excited state wavefunction

Multiwfn is extremely powerful on electronic structure analysis, the analyses can not only be applied to ground state, but can also be applied to excited state, as long as the input file contains excited state wavefunction. Note that if the excited states were calculated by CIS/TDHF/TDDFT/TDA-DFT methods, the input file must record natural orbitals (NOs) of corresponding excited state. By using Multiwfn, the NOs can be generated based on the excited state density matrix in .fch file, see Section 3.200.16 for detail; the NOs can also be generated based on configuration coefficients, as shown in Section 3.21.13.

After loading excited state wavefunction into Multiwfn, one can carry out a variety of electronic structure analyses. For example, main function 9 can be used to calculate various kinds of bond orders for excited state, main function 7 can perform population analysis and calculate atomic charges for excited state, main functions 3,4,5 are able to plot more than one hundred of real space functions for excited state, AIM analysis can be applied to excited state by main functions 2 and 17, weak interaction of excited state can be visually studied via main function 20, excited state

aromaticity can be investigated via a bunch of methods in Multiwfn (see Section 4.A.3). By comparing analysis result of excited state and ground state, the impact on electronic structure caused by electron excitation can be fully shed light on.

11 Orbital composition analysis

Multiwfn has a very powerful orbital composition analysis module (main function 8), which supports all orbital composition analysis methods. Via this function, one can study the MOs or NTOs that mainly involved in electron excitation to make clear the role that played by various atomic orbitals, atoms and fragments.

12 Examining overlapping extent and centroid distance between orbitals

Subfunction 11 of main function 100 is used to calculate overlapping extent and centroid distance between two selected orbitals. Evidently, this function is useful for studying electron excitation. For example, using this function to analyze the MO pair or NTO pair that dominates the electron excitation, one can investigate charge displacement and separation degree during the electron excitation.

13 Evaluating atomic transition charges

The atomic charge we commonly say is for a single electronic state, it is essentially determined by density matrix of this state. It is also possible to calculate charge for each atom using transition density matrix between two states, these charges are known as atomic transition charges. Just as the method of calculating atomic charges is not unique, there are many different methods for calculating the atomic transition charges. Multiwfn can calculate Mulliken atomic transition charges, see corresponding description in Section 3.21.12. Multiwfn can also calculate atomic transition charges via electrostatic potential fitting method, *J. Phys. Chem. B*, **110**, 17268 (2006) and some other literatures called this kind of charges as TrEsp (transition charge from electrostatic potential). Basic theory and calculation example of TrEsp can be found in Section 4.A.9 of the manual. The main use of atomic transition charges is quickly calculating electrostatic potential corresponding to transition density, thereby examining the exciton coupling between molecules, this point is also described in detail in Section 4.A.9.

14 Investigating contributions of orbital transitions to electron excitation

Computing contribution of an orbital transition to electron excitation is rather simple, see beginning of Section 3.21 for introduction. For facilitating analysis, when you enter subfunction -1 of main function 18, contribution of ten orbital transitions that have largest contributions to the selected electron excitation will be directly listed, see Section 3.21.0 for more information.

15 Identifying ghost states

Asymptotic behavior of exchange potential of pure or hybrid DFT functionals with low HF exchange composition is obviously incorrect. When TDDFT with such exchange-correlation functional is used to calculate excited states of large conjugate systems, a batch of artificial charge transfer excited states with low energy tend to occur. The ghost states have no any physical meaning, their existences not only wastes computation time, but may also cause beginners to mistake a ghost state as an emission state. The ghost-hunter index proposed in *J. Comput. Chem.*, **38**, 2151 (2017) can be used to diagnose whether an excited state produced by TDDFT calculation is a ghost state. This index is automatically outputted after performing hole-electron analysis analysis, see Section 3.21.7 for detailed introduction and 4.18.1 for example. If a ghost state is found, the researchers can

avoid these states in their discussions, or try to eliminate these states by using a DFT functional with higher HF exchange composition or long-range corrected functionals.

Others

Do not forget Multiwfn has main function 11, which can plot UV-Vis and ECD spectrum based on the oscillator/rotatory strengths and excitation energies outputted by quantum chemistry codes. This module is by far more powerful and flexible than any other plotting tools and able to provide detailed information about the spectrum. Please check Section 3.13 for introduction and Section 4.11 for abundant examples.

It is also worth mentioning the biorthogonalization method, which may be also useful in studying the nature of triplet excited state, that is this method can usually describe the triplet excited state calculated by UKS or UHF method in terms of orbital transition model, so that discussion the nature of the excitation could be simplified. See Section 3.100.12 for introduction and 4.100.12 for example.

Finally, note that only for the aforementioned entry 5 (density difference analysis), entry 6 (atom/fragment charge analysis) and entry 10 (excited state wavefunction analysis), arbitrary electron excitation calculation method could be used as long as they can produce excited state wavefunction. For example, for the density difference analysis, the difference can be made between the electron density of the lowest triplet excited state and singlet state calculated by KS-DFT, the difference can also be made between excited state density produced by EOM-CCSD and the ground state density yielded by CCSD. While for other kinds of analyses, such as hole-electron analysis, IFCT analysis, only CIS, TDHF, TDDFT and TDA-DFT can be employed for calculating excited states.

4.A.13 Plot electrostatic potential colored van der Waals surface map and penetration graph of van der Waals surfaces

Note: Chinese version of this tutorial is <http://sobereva.com/443>, it contains more discussion and examples than this section.

Foreword

In the tutorial "Plotting electrostatic potential colored molecular surface map with ESP surface extrema via Multiwfn and VMD" (<http://sobereva.com/multiwfn/res/plotESPsurf.pdf>), I detailedly described how to plot electrostatic potential (ESP) colored molecular van der Waals (vdW) surface, this kind of map is very important and frequently occur in literatures. However there are lots of steps in the tutorial. In order to make plotting this kind of map easier, here I introduce a script-based method to draw similar graph, and meantime I will introduce how to plot penetration graph of vdW surfaces, which is very useful for discussing intermolecular interactions. However, I still strongly suggest you also read the tutorial "plotESPsurf.pdf" after reading the content in present section, so that you can understand more details and are able to properly improve the obtained graph.

Preparation

VMD program is needed in the plotting, the version I used here is 1.9.3, it can be freely downloaded from <http://www.ks.uiuc.edu/Research/vmd/>. In this section I assume that you are using Windows system, however the method introduced in this section is also suitable for Linux system,

but some steps may need to be properly changed.

All files utilized below are given in "examples\drawESP" folder, they are briefly introduced here:

- .bat files: Batch process files of Windows system. They are used to invoke Multiwfn to calculate needed data for plotting graphs in VMD. The content of the files is very easy to understand and can be easily modified. If you do not know how to run Multiwfn in silent mode, please check Section 5.2

- .txt files: Input stream files of Multiwfn that involved in the .bat files.
- .vmd files: VMD plotting scripts.

Before plotting, you should do below things:

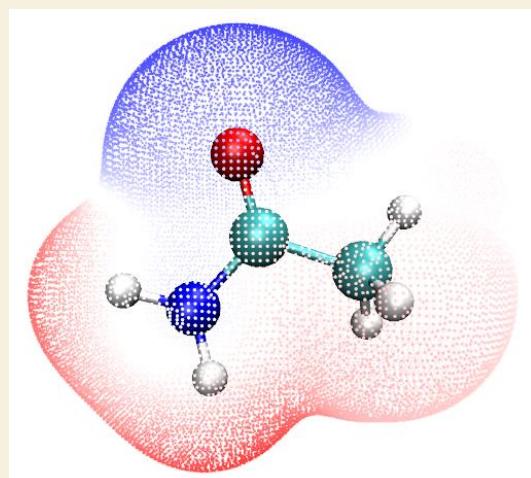
- (1) Move all .bat and .txt files to the folder containing Multiwfn executable file
- (2) Modify the VMD path in the .bat files to actual path of VMD in your machine
- (3) Copy all .vmd files to VMD folder
- (4) Add below content to the end of the *vmd.rc* file in VMD folder:

```
proc iso {} {source ESPiso.vmd}
proc iso2 {} {source ESPiso2.vmd}
proc pt {} {source ESPpt.vmd}
proc pt2 {} {source ESPpt2.vmd}
proc ext {} {source ESPext.vmd}
```

Do not forget that when .fch file is used as input file, the computational cost of ESP can be significantly reduced by borrowing cubegen utility in Gaussian package, see Section 5.7 for detail.

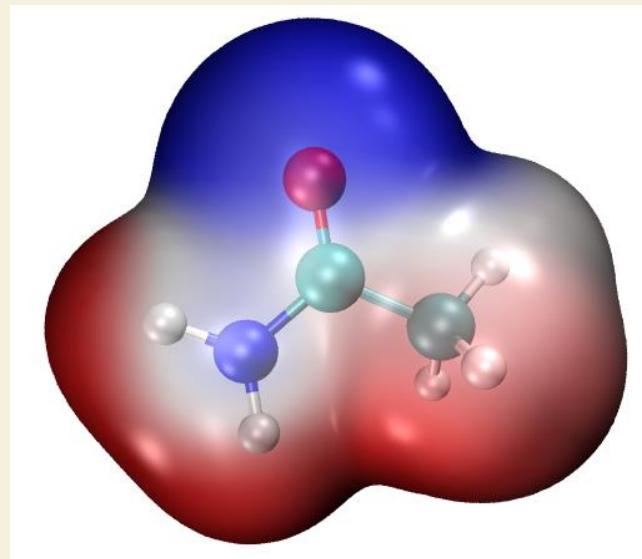
Plotting ESP colored vdW surface for single molecule

Here we take acetamide as example. Move the *CH3CONH2.fch* in "examples" folder to the folder containing Multiwfn executable file, modify the file name to *1.fch*. Double click the *ESPpt.bat*, Multiwfn will be invoked to carry out quantitative molecular surface analysis (main function 12) for the *1.fch*, once the calculation has done, the exported *moll.pdb* and *vtx1.pdb* will be automatically moved to the VMD folder. Then boot up VMD and input command *pt* in VMD console window, then the *ESPpt.vmd* will be activated to load the *moll.pdb* and *vtx1.pdb* to draw below map:



The default lower and upper limits of color scale is -50 and 50 kcal/mol, respectively, and the default color transition is BWR (Blue-White-Red), therefore in above map the white area corresponds to the region having almost vanished ESP value, while the red and blue points have evidently positive and negative ESP, respectively. You can manually change the default setting by modifying the *ESPpt.vmd*, the settings can also be changed in VMD GUI interface, see the *plotESPSurf.pdf* tutorial for detail.

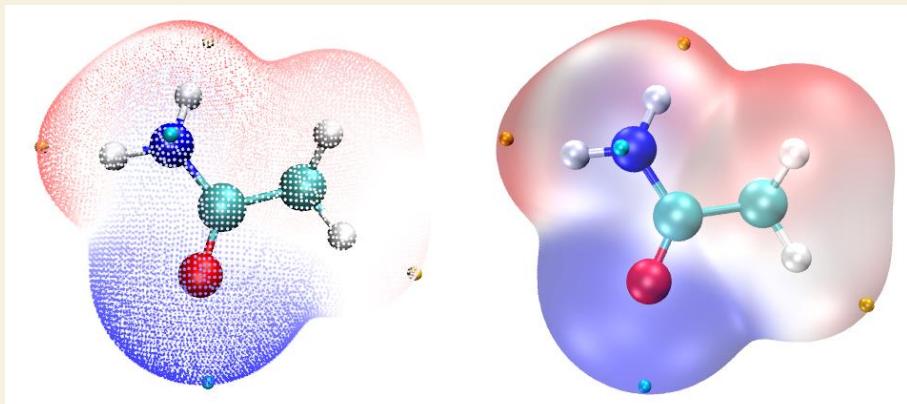
In above map, the ESP colored vdW surface is represented in terms of surface vertices, the graph can also be drawn in an alternative way, namely mapping ESP on electron density isosurface, we do this now. Double click the *ESPIso.bat*, then Multiwfn will be invoked to calculate and export cube file of electron density and ESP, the resulting *density1.cub* and *ESPI.cub* will be automatically moved to the VMD folder. Then boot up VMD and input command *iso* in VMD console window, then the *ESPIso.vmd* will be activated to load the two cube files to draw below map. Notice that in order to gain slightly better effect, I used the bulit-in Tachyon render to obtain below graph, namely selecting "File" - "Render", change to "Tachyon (internal, in-memory rendering)" and click "Start Rendering" button (The resulting file is in .tga format, you need to use advanced image viewer to view it, such as IrfanView, which is freely available at <https://www.irfanview.com>).



This graph looks more pretty than the former one. Unfortunately, calculation of ESP grid data is much more time-consuming than calculation of ESP on surface vertices, since the number of surface vertices is by far less than the number of grid points.

It is possible to append the ESP surface extrema on the graph. To do this, double click *ESPext.bat*, it will do all things that *ESPpt.bat* do, but it also outputs *surfanalysis.pdb* and move it to the VMD folder, this file records surface extrema. Then boot up VMD and input command *pt* or *iso* to draw corresponding map first, and then input *ext*, then *ESPext.vmd* will be activated to load the *surfanalysis.pdb* and render the surface extrema as points. The combination of *pt+ext* and *iso+ext* are shown at left and right sides of below graph, respectively. Note that in order to make ESP extrema at backside visible, I have changed the material of electron density isosurface to "Transparent" (namely enter "Graphics" - "Representation", switch to "density1.cub", change "Material" to "Transparent"). If you want to make this as default setting, modify the *ESPIso.vmd* and

change the "\$id EdgyGlass" to " id Transparent")

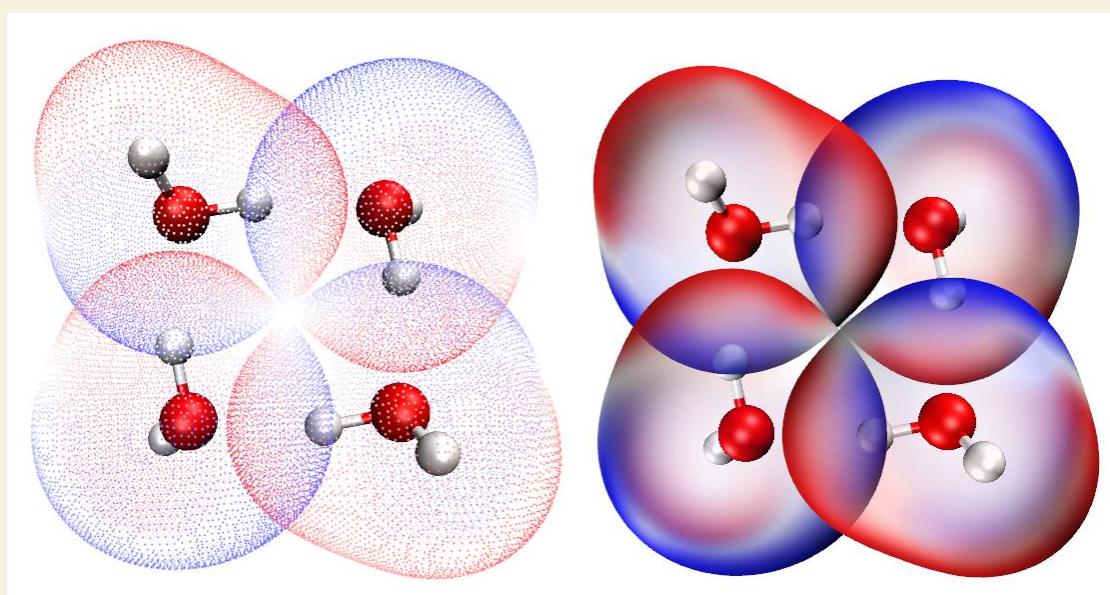


In above graph, the orange and cyan spheres correspond to the positions of maxima and minima of ESP on the vdW surface. You can also manually mark ESP values on the extrema by image editor, see the *plotESPsurf.pdf* tutorial on how to do this.

Plotting ESP colored penetration map of vdW surface of monomers

Here I use water tetramer to illustrate how to plot this kind of map. The files used in this instance are provided in "examples\water_tetramer\fch" folder. The Gaussian input files of the four water molecules are *1/2/3/4.gjf*, respectively, their coordinates were directly extracted from the optimized tetramer coordinate, which can be found in *complex.gjf*. Run these .gjf files by Gaussian, you will obtain the *1/2/3/4.fch*. Notice that *nosymm* keyword has been employed, otherwise the Cartesian coordinates of the monomers will be no longer consistent with those in complex, because without this keyword Gaussian will automatically put the systems to standard orientation.

Copy the *1/2/3/4.fch* files to the folder containing Multiwfn executable file, run *ESPpt.bat*, then Multiwfn will be invoked to calculate the four .fch files in turn, the resulting *mol1/2/3/4.pdb* and *vtx1/2/3/4.pdb* will be automatically moved to the VMD folder. Then boot up VMD and input *pt2* to activate the *ESPpt2.vmd* script, you will immediately see left part of below graph. If you run *ESPiso.bat* and then input *iso2* in VMD instead, then the *ESPiso2.vmd* will be activated to draw the right part of below graph based on the exported *density1/2/3/4.cub* and *ESP1/2/3/4.cub*.



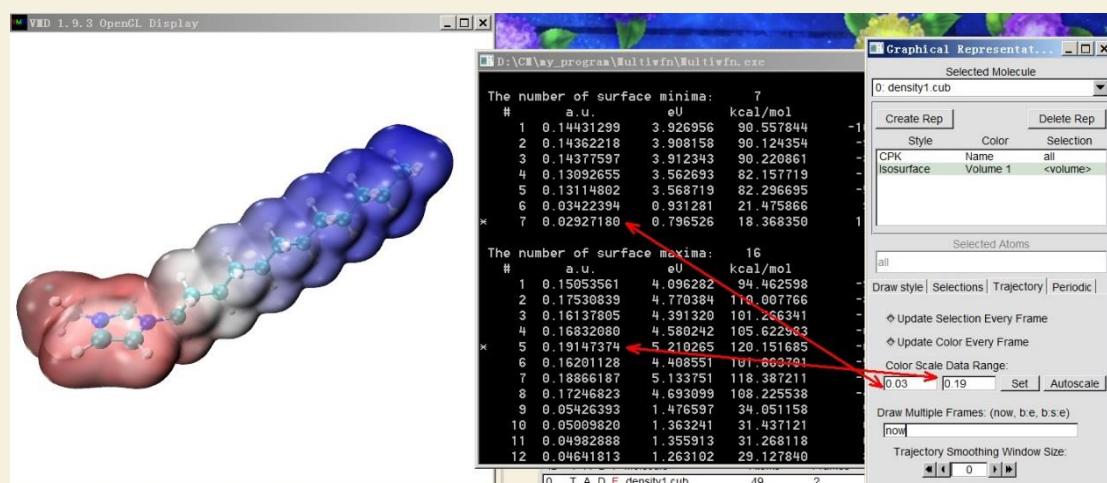
From above map, the inter-penetration between the vdW surfaces of the four monomers due to formation of hydrogen bonds can be clearly seen. In addition, the mapped colors show that the tetramer was formed in ESP positive-negative complementary way, revealing the electrostatic nature of the hydrogen bonds.

As an exercise, please try to plot the ESP colored vdW surface penetration map of Guanine-Cytosine dimer via above two ways, the .fch file of the two monomers can be downloaded at http://sobereva.com/multiwfn/extrafiles/GC_fch.rar. Notice that before plotting, you should manually delete the .pdb and .cub files generated for previous systems in VMD folder.

I strongly suggest readers to check the content of the .bat, .txt and .vmd files to make clear how they work. As you can see, the *ESPpt.bat* and *ESPIso.bat* can at most deal with four .fch file (1/2/3/4.fch), you can also extend them to more molecules. In the *ESPpt2.vmd* and *ESPIso2.vmd*, the variable "nSystem" was set to 4, namely at most mol4.pdb&vtx4.pdb and density4.cub&ESP4.cub will be loaded and plotted, clearly if you want to use the plotting scripts to simultaneously plot more monomers, the "nSystem" should be enlarged.

It is noteworthy that the value 0.15 in the *ESPpt.txt* is the grid spacing in quantitative molecular surface analysis; if you increase it, the surface vertices will become more sparse, and calculation cost will be reduced. The default commands in the *ESPIso.txt* correspond to using medium quality grid for electron density while use low quality grid for ESP (for saving computational cost purpose), this combination is suitable for small and medium sized systems, but for large systems, you may need to modify the file so that high and medium quality grids are employed for electron density and ESP, respectively, otherwise the resulting isosurface may be unsMOOTH and the mapped color may be blurred.

For better graphical effect, users are suggested to manually change lower and upper limits of color scale, so that the variation of ESP on vdW surface can be represented as clear as possible. For ionic systems, the default color scale always should be changed, otherwise the vdW surface will be monochromatic. For these systems, you should load the input file, enter main function 12, select option 1 to carry out quantitative molecular surface analysis for ESP, copy the global minimal and maximal values of ESP to the text box for setting color scale, as shown below, and then press ENTER button to make the setting take effect.



5 Skills

5.1 Make Multiwfn support more quantum chemistry programs

Although currently Multiwfn is able to directly accept Molden input file (.molden) as input file, only the file generated by a few programs is formally supported (see the related description in Section 2.5). If the Molden input file is generated by other programs then the analysis result may be incorrect. For these cases, you can use Molden2aim program written by Wenli Zou (<https://github.com/zorkzou/Molden2AIM>) to produce standardized Molden input file.

Using Molden2aim is easy. First move the Molden input file (e.g. ltwd.molden) into the directory where molden2aim.exe is placed, then properly modify its setting file m2a.ini, then boot up Molden2aim and follow its prompts to input commands, finally you will get ltwd_new.molden (standardized Molden input file) and maybe other files (.wfn, .wfx, etc.).

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 originally 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 outputted 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, do not forget to modify "charge" field in the wfn file outputted by Molden2aim to effective nuclear charges.

5.2 Running Multiwfn in silent mode

Multiwfn aims for ease of use, so it is designed as an interactive program. Nevertheless Multiwfn can also run in silent mode, by which you do not need to press any buttons during Multiwfn running. Here I introduce how to do that.

For Windows users

For example, you want to obtain the graph in example 4.4.1 silently, you need to write an inputstream file first, the content is (red texts are comments):

```
4 ← Main function 4
1 ← Real space function 1
1 ← Color-filled map
```

```

← Space line, corresponding to press ENTER button once (use default grid setting)
2 ← XZ plane
0 ← Y=0
0 ← Option 0: save graph to current directory

```

I suppose that the inputstream file is named as *4.4.1.txt*, I have already provided this file in “examples” directory. Now change “silent” parameter in “settings.ini” from 0 to 1, this step tells Multiwfn to forbid popping up of any graph or GUI automatically during running (otherwise you have to close the window by clicking mouse button). Then enter command-line environment of Windows (click “Start”-“run” and type “cmd”) and run:

```
Multiwfn HCN.wfn < 4.4.1.txt > medinfo.txt
```

Here I assume that *Multiwfn.exe*, *4.4.1.txt* and *HCN.wfn* are in current directory. After a few seconds, you will find that image file has appeared in current directory. From *medinfo.txt* you can find all intermediate information outputted by Multiwfn (the same as in interactive mode).

What does the content in inputstream file mean? The answer is: the texts in each line of inputstream file are just what you need to input in interactive mode. It is very easy to write a new inputstream file by following the prompts on screen in interactive mode. The symbol “<” and “>” are called as redirection operator, they respectively tell Multiwfn that the content in *4.4.1.txt* is inputstream, while the outputstream should be stored to *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:

```

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  

.....
```

Actually they are not errors, and hence you can safely ignore them. However, if you really want to exit Multiwfn gracefully to avoid printing the “errors”, you should properly write the inputstream file so that the hidden option -10 in main function menu is selected at the final step.

Another example, assume that you want to save detail compositions of orbitals 1 to 3 of *COCl2.fch*, just create an inputstream file *orbana_1_3.in* with below content:

```

8 ← Orbital composition analysis
1 ← Mulliken method
1 ← Orbital 1
2 ← Orbital 2
3 ← Orbital 3

```

Then run the command: *Multiwfn COCl2.fch < orbana_1_3.in > orbana_1_3.txt*.

Notice that if you are using Powershell in Windows environment, since the “<” redirection operator has not been supported, you should use “Get-Content” command and pipeline feature instead. For example, above command should be written as (assume that *Multiwfn.exe* is in current folder):
Get-Content orbana_1_3.in | ./Multiwfn.exe COCl2.fch > orbana_1_3.txt

For Linux / MacOS users

If you are a Linux or Mac OS user, you can not only run Multiwfn silently as introduced above, but also make use of "echo" command to avoid explicitly writting an inputstream file. The last example can be equivalently realized by running this command:

```
echo -e 8\n1\n1\n2\n3" Multiwfn COCl2.fch > orbana_1_3.txt
```

Each \n means pressing ENTER button once.

If you prefer to use shell script, you can also add below lines into your shell script file:

```
Multiwfn COCl2.fch > orbana_1_3.txt << EOF
8
1
1
2
3
EOF
```

5.3 Running Multiwfn in batch mode

If you are familiar with writing shell scripts and you have carefully read last section, you must have already know how to use Multiwfn to process a batch of files, it is indeed very easy. I will briefly introduce this point in this section.

For Windows users

Assume that you want to generate .cub files of ELF for these input files: ultravox.wfn, chinaski.fch, strawberry_egg.wfn, you should create a plain text file named *batchrun.bat* (the suffix must be .bat, while file name is arbitrary) with below content:

```
Multiwfn ultravox.wfn < genELFcub.txt > null
move ELF.cub ultravox.cub
Multiwfn chinaski.fch < genELFcub.txt > null
move ELF.cub chinaski.cub
Multiwfn strawberry_egg.wfn < genELFcub.txt > null
move ELF.cub strawberry_egg.cub
del null
```

where *genELFcub.txt* is inputstream file for generating ELF cube file, it is a plain text file having below content is

```
5 ← Main function 5, calculate grid data
9 ← Real space function 9, namely ELF
2 ← Option 2: Medium quality grid
2 ← Option 2: Exporting cube file in current directory
```

Now, double click the icon "batchrun.bat" or input command *batchrun* in command line window of Windows system, the work will start and then the three ELF cube files will be generated in current folder sequentially.

Shell script is very useful and powerful, it can automate a lot of repetitive work. As a simple example, you want to generate ELF .cub file for all .wfn files in current folder, and hope that the name of the resulting .cub files are *[Input_file_name]_ELF.cub*, then you can write a .bat file with below content

```
for /f %%i in ('dir *.wfn /b') do (
    Multiwfn %%i < genELFcub.txt > null
    rename ELF.cub %%~ni_ELF.cub
)
```

Then run the .bat file, the .cub files will be yielded in turn. Assume that one of input file is *yoshiko.wfn*, the corresponding resulting .cub file will be *yoshiko_ELF.cub*.

For Linux / MacOS users

Similarly, you can run Multiwfn under Linux/MacOS environment in batch mode and write script to significantly make your research easier. To run the first example in this section, you can create a file *runthree.sh* with below content,

```
./Multiwfn ultravox.wfn < genELFcub.txt > null
mv ELF.cub ultravox.cub
./Multiwfn chinaski.fch < genELFcub.txt > null
mv ELF.cub chinaski.cub
./Multiwfn strawberry_egg.wfn < genELFcub.txt > null
mv ELF.cub strawberry_egg.cub
rm null
```

Assume that executable file of Multiwfn, *runthree.sh* and all input files are in current, then run this command: *chmod +x ./runthree.sh;./runthree.sh* (The *chmod +x* command is used to add executable permission, it may be not necessary for certain systems)

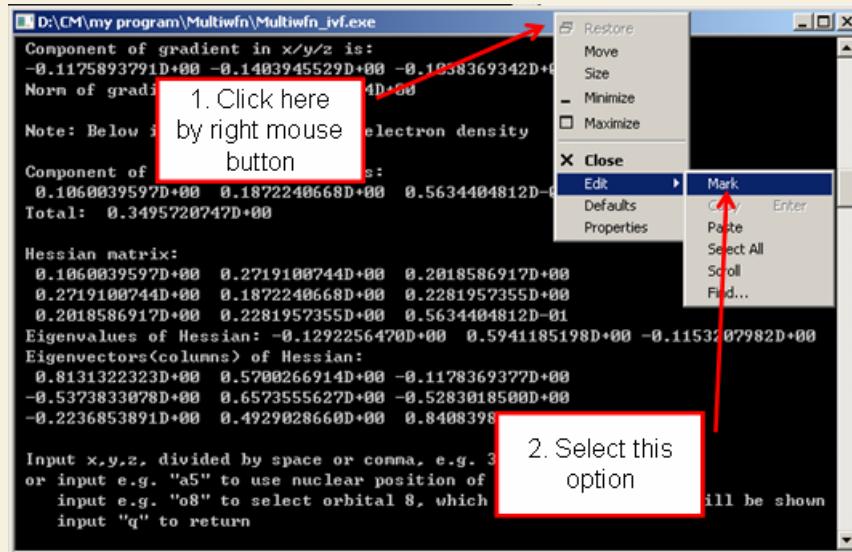
To realize the second example of this section under Linux (assume that you are using the most popular "bash" shell), you should create a shell script file with below content

```
#!/bin/bash
for inf in *.wfn
do
    echo Running ${inf} ...
    ./Multiwfn ${inf} < genELFcub.txt > null
    mv ELF.cub ${inf//.wfn}_ELF.cub
done
rm null -f
```

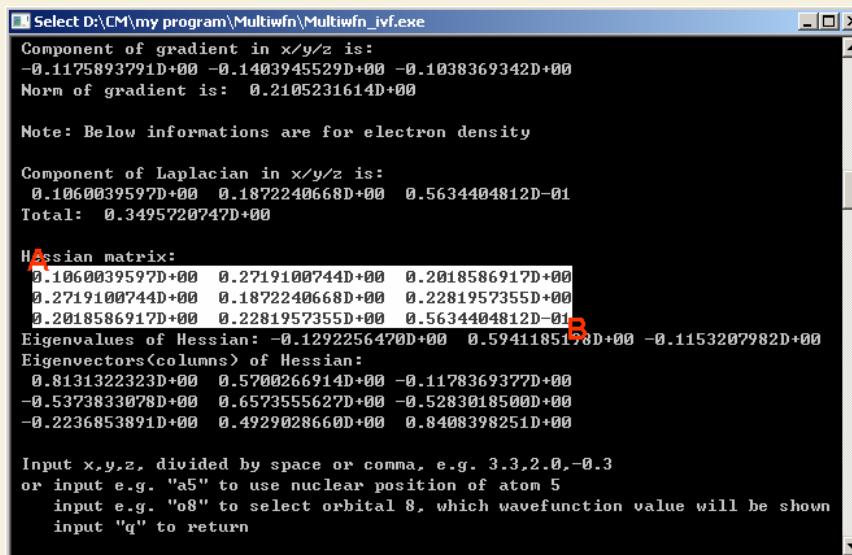
Substantially and flexibly employing shell scripts can automate much more kinds of analyses than that shown above. For example, in Section 4.18.6 I illustrated that with a simple script, natural transition orbitals (NTOs) of all selected excited states can be yielded and exported to various files in one run.

5.4 Copy outputs from command-line window to clipboard

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



Then press ENTER button, the information highlighted by white rectangle will be stored to clipboard, you can paste them to anywhere, such as plain text file.

For Mac OS or Linux system running in graphical environment, you can also copy the output of Multiwfn from console to plain text file by similar manner.

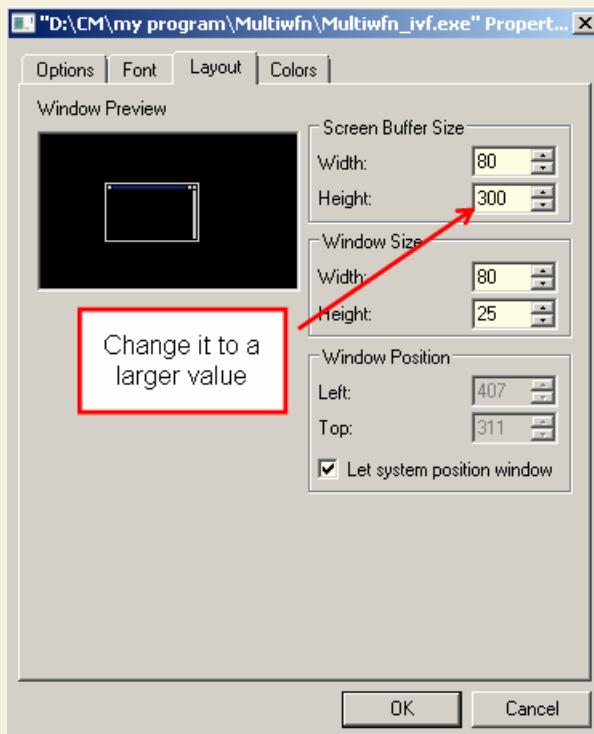
5.5 Make command-line window capable to record more outputs

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 9999, 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 and Mac OS, you can also find similar option used to set buffer size of terminal.



5.6 Rapidly load a file into Multiwfn

Probably sometimes you feel inputting the path of input file is cumbersome, especially when the path is very long. Below I provide you some tricks, which make this step much more easier.

If you want to rapidly load a file into Multiwfn without inputting its path, you can boot up

Multiwfn and then directly drag the icon of the file into the Multiwfn command-line window.

In Windows platform, an even more simple method is directly dragging the file onto the icon of "Multiwfn.exe", then the file will be automatically loaded into Multiwfn. Notice that in this situation, the "current folder" is the position of the input file.

If directly inputting letter *o*, the file last time loaded will be loaded again, whose path is recorded as *lastfile* variable in settings.ini file.

Assume that the file you last time loaded is *C:\sob\lover\K-ON\Mio.wfn*, and this time you want to load *C:\sob\lover\K-ON\Azusa.wfn*, you can simply input *?azusa.wfn*, namely the path of the folder last time involved can be replace with a question mark.

5.7 Make use of cubegen utility in Gaussian package to reduce computational time of electrostatic potential analyses

Frankly speaking, calculation speed of ESP in Multiwfn is not as fast as the cubegen utility in Gaussian package. For very large system, such as carbon nanotube, if you directly use Multiwfn to carry out analyses about ESP, you will find the computational cost is sometimes prohibitively high. In this circumstance, you can make use of cubegen to calculate the ESP data instead of Multiwfn internal code to significantly reduce the overall time-consuming.

The way of making use of cubegen in analyses about ESP is rather easy: Set *cubegenpath* parameter in settings.ini file to actual path of cubegen executable file (For example, "D:\study\G16W\cubegen.exe" in Windows platform and "/sob/g16/cubegen" in Linux platform). Then if you use .fch/fchk type of file as input file of Multiwfn, cubegen will be automatically invoked by Multiwfn at the right time to calculate all needed ESP data. Below cases or functions currently support this feature:

- Plotting curve map of ESP (main function 3)
- Plotting plane map of ESP (main function 4)
- All functions that request grid data of ESP (*e.g.* using main function 5 to evaluate grid data of ESP, using main function 17 to perform basin analysis for ESP, using subfunction 14 of main function 200 to perform domain analysis for ESP)
- Calculating ESP fitting atomic charges such as CHELPG, MK and RESP (via corresponding subfunctions in main function 7)
 - Calculating TrEsp charges (see Section 4.A.9 on how to do this)
 - Quantitative molecular surface analysis with ESP as mapped function (main function 12)

Although many other functions in Multiwfn also need ESP information, they do not support utilizing cubegen because only very few number of points are needed to be calculated.

Currently, this feature is fully compatible with cubegen in Gaussian 09 and 16 packages. For cubegen in Gaussian 16, the number of CPU cores assigned for cubegen is identical to the *nthreads* parameter in settings.ini file. However, since parallel implementation of cubegen in Gaussian 09 seems buggy, only one CPU core will be employed for it. Multiwfn judges whether it is G09 or G16 cubegen by checking if there is "g16" or "G16" string in the *cubegenpath*.

Even if you are not a Gaussian user, as long as the quantum chemistry program you used can

produce .molden file, or you are using GAMESS-US/Firefly, you can also benefit from cubegen, because by using subfunction 2 of main function 100, Multiwfn can convert the loaded .molden/.gms file to .fch file.

- NOTICE 1: The wavefunction invoked by cubegen used for evaluating ESP comes from density matrix in .fch file. This file may have more than one density matrices, by default the SCF density matrix is used. The type of density matrix employed could be chosen by *cubegendenstype* parameter in settings.ini. For example, if a .fch file was produced via "# MP2/cc-pVTZ density" keyword, then the file will contain both SCF density matrix and MP2 density matrix. If you want to calculate ESP at MP2 level by means of cubegen, you should change the value of *cubegendenstype* from the default "SCF" to "MP2".
- NOTICE 2: Even though you have modified current wavefunction via some functions of Multiwfn, e.g. using subfunction 26 of main function 6 to manually change orbital occupation numbers, the ESP calculated by cubegen still corresponds to the original wavefunction, since the density matrix in the input file was not modified. If you indeed want to calculate ESP based on modified wavefunction via cubegen, you should first use subfunction 2 of main function 100 to export .fch file, the SCF density matrix in this file will correspond to the density matrix constructed by current wavefunction, and thus then you can use this .fch file as input file to carry out ESP analyses.

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 properly, therefore Gaussian does not know where to find 11.exe. Here I show you how to set it.

- (1) Windows XP users: Enter “Control panel”-“System properties”-“Advanced”
- (2) Windows 7 users: Enter “Control panel”-“System”-“Advanced system setting”-“Advanced”
- (3) Windows 10 users: Click right mouse button on the start button, enter “Control panel”-“System”-“Advanced system setting”-“Advanced”

Then 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 that 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 The routines for evaluating real space functions

Below are the routines in function.f90 file. You can make use of them to construct new real space function by yourself. For more details please check the comments in the code of corresponding routine.

Calculate real space functions:

function calcfuncall: A warpper for calculating any supported real space function at a given point

function userfunc: User defined real space function

function linintp3d: Function value obtained by trilinear interpolation from grid data in memory

function fmo: Orbital wavefunction value

function fdens: Electron density

function fspindens: Spin or Alpha or Beta electron density

function fgrad: Gradient (x,y,z components or its norm) of density, or reduced density gradient (RDG)

function flapl: Laplacian of electron density (xx or yy or zz part or total)

function Lagkin: Lagrangian kinetic $G(\mathbf{r})$, or its components

function Hamkin: Hamiltonian kinetic $K(\mathbf{r})$, or its components

function calcprodens: Promolecular density

function signlambda2rho: $\text{Sign}[\lambda_2(\mathbf{r})]\rho(\mathbf{r})$

subroutine signlambda2rho_RDG: Calculate $\text{Sign}[\lambda_2(\mathbf{r})]\rho(\mathbf{r})$ and RDG at the same time

function signlambda2rho_prodens: $\text{Sign}[\lambda_2(\mathbf{r})]\rho(\mathbf{r})$ with promolecular approximation
function RDGprodens: RDG with promolecular approximation
subroutine signlambda2rho_RDG_prodens: Calculate $\text{Sign}[\lambda_2(\mathbf{r})]\rho(\mathbf{r})$ and RDG at the same time with promolecular approximation
subroutine IGMprodens: Calculate promolecular density gradient of usual type or Independent Gradient Model (IGM) type
function ELF_LOL: ELF or LOL or SCI (strong covalent interaction index)
function avglocion: Average local ionization energy
function loceleaff: Local electron affinity
function edr: Electron delocalization range EDR($\mathbf{r};d$)
function edrdmax: Orbital overlap distance function D(\mathbf{r})
function delta_g_IGM: $\delta g(\mathbf{r})$ defined in IGM method
function linrespkernel: Approximate form of DFT linear response kernel for closed-shell
function pairfunc: Exchange-correlation density, correlation hole and correlation factor, on-top pair density
function srfunc: Source function
function infoentro: Shannon information entropy function or Shannon entropy density
function totesp: Total ESP
function nucesp: ESP from nuclear or atomic charges
function eleesp: ESP from electrons
function totespskip: ESP without contribution of nuclues defined by *iskipnuc* parameter
subroutine planeesp: Calculate ESP in a plane
subroutine espcub: Calculate grid data of ESP from electrons
function twoorbnorm: Product of norm of two orbitals
function beckewei: Generate Becke weighting function
function densellip: Ellipticity of electron density and η index
function xLSDA: Integrand of LSDA exchange functional
function xBecke88: Integrand of Becke88 exchange functional
function cLYP: Integrand of LYP corelation functional
function DFTxcfunc: Various kinds of integrand of DFT exchange-correlation functionals
function DFTxcpot: Various kinds of DFT exchange-correlation potentials
function weizsacker: Integrand of Weizsacker functional (steric energy)
function KED: Various kinds of kinetic energy density of electrons
function stericpot: Steric potential, which negative value is one-electron potential
function stericcharge: Steric charge
function stericforce: Magnitude of steric force
function paulipot: Pauli potential
function pauliforce: Magnitude of Pauli force
function paulicharge: Pauli charge
function Fisherinfo: Fisher information density
function calcatmdens: Promolecular density calculated based on Lagrange interpolation of built-in atomic radial density
function PAEM: Potential acting on one electron in a molecule
function SEDD: Single exponential decay detector (SEDD)

function DORI: Density Overlap Regions Indicator (DORI)
function localcorr: Local electron correlation function
function elemomdens: Electron linear momentum density
function magmomdens: magnetic dipole moment density
function energydens_grdn: Gradient norm of energy density
function energydens_lapl: Laplacian of energy density

Calculate derivatives of real space functions:

subroutine gencalchessmat: A general routine used to calculate value, gradient and Hessian matrix at a given point for some real space functions
subroutine orbderiv: Calculate wavefunction value of a range of orbitals and their derivatives at given point, up to third-order
subroutine EDFrho: Calculate contribution from EDFs to density and corresponding derivatives (up to third-order)
subroutine calchessmat_dens: Calculate electron density, its gradient and Hessian matrix
subroutine calchessmat_prodens: Calculate electron density, its gradient and Hessian matrix with promolecular approximation
subroutine gendensgradab: Generate electron density and gradient norm for alpha and beta electrons at the same time
subroutine calchessmat_lapl: Calculate Laplacian of electron density, its gradient and Hessian matrix (Hessian is currently not available)
subroutine calchessmat_ELF_LOL: Calculate ELF/LOL, its gradient and Hessian matrix (Hessian is currently not available)
subroutine calchessmat_mo: Calculate gradient and Hessian matrix for an orbital wavefunction
subroutine stericderv: Calculate the first-order derivative of steric potential

3 Detail of built-in atomic densities

Some analyses, such as Hirshfeld/ADCH population analyses and Hirshfeld orbital composition analysis request atomic densities. Though, as shown in Section 3.7.3, the atomic densities can be evaluated based on atomic .wfn files, the process is slightly complicated, namely the element .wfn files needed must be prepared and sphericalized first. In order to simplify these analysis tasks, a set of built-in atomic densities (available from H to Lr) was provided in Multiwfn and one can directly choose to use it.

These built-in atomic densities were evaluated at highly accurate computational level at atom ground-states, and have been sphericalized (density distribution of many atoms in their ground states are not spherically symmetric). Main group elements with index ≤ 18 were calculated at B3LYP/cc-pVQZ level, those > 18 were calculated at B3LYP/ANO-RCC level (except for Ca, UGBS is used since the ANO-RCC of Ca in EMSL website is wrong). Transition metals were calculated at HF/UGBS level. Lanthanides and Actinides were calculated at B3LYP/SARC-DKH level (except for U and Np, for which ROHF was used instead of B3LYP because DFT cannot reproduce their correct ground state configurations). For all elements heavier than Ar, DKH2 method was employed to take scalar relativistic effect into account. Unless otherwise specified, open-shell systems was treated by unrestricted open-shell formalism.

The atomic densities are recorded as radial points in "atmraddens.f90", the second kind of Gauss-Chebyshev method was used to generate the position of radial points, and the point distribution is identical for all elements. The atomic density at any point is evaluated in terms of Lagrange interpolation method based on the points.

If you want to replace built-in atomic density of an element with that calculated by yourself, after booting up Multiwfn and loading corresponding atom wavefunction file, select subfunction 10 in main function 100 (a hidden option), then Multiwfn will calculate radial electron density and output the result to *sphavgval.txt* in current folder. You can directly copy the Fortran codes in this file into corresponding field of atmreddens.f90.

4 Details about supplying inner-core electron density for the wavefunctions involving pseudo-potential

In Section 2.5, the characteristic and the meaning of electron density function (EDF) have been introduced. EDF information is used to represent the inner-core density replaced by pseudo-potential, so that for the wavefunctions involving pseudo-potential, the result of the wavefunction analyses that purely based on electron density can be almost exactly identical to full-electron wavefunction.

When the input file you used contains GTF information, and meantime some atoms utilized pseudo-potential, Multiwfn automatically finds proper GTF information from built-in EDF library for these atoms. The built-in EDF library was developed by Wenli Zou and coworkers, it was originally released as a part of Molden2aim program (<https://github.com/zorkzou/Molden2AIM>). The quality of this EDF library is quite good, and is better than the EDF field contained in the .wfx file produced by Gaussian program. This EDF library covers the entire periodic table, up to index of 120. For most elements, it contains both EDF information for large and small core pseudo-potentials. Some descriptions about this library posted by its developer can be found in <http://bbs.keinsci.com/thread-5354-1-1.html> as well as *J. Comput. Chem.*, **39**, 1697 (2018). If you do not want Multiwfn to read EDF information from this library automatically, set "isupplyEDF" in settings.ini to 0.

As mentioned in Section 2.5, the .wfx file produced by Gaussian directly carries EDF field when pseudo-potential is used. When this kind of file is used as input, Multiwfn by default reads EDF information from EDF field of this .wfx file rather than from the built-in EDF library. If you do not want Multiwfn read EDF information from this file but from the built-in EDF library, you can change "readEDF" in settings.ini from 1 to 0.

Worhnotingly, Multiwfn also allows reading EDF information from atomic .wfx file produced by Gaussian rather than from the built-in EDF library of Multiwfn (This feature is rarely useful, since as mentioned above, the quality of the built-in EDF library in Multiwfn is better than the EDF library embedded in Gaussian). An example is provided as below:

"examples\Pt(NH₃)₂Cl₂.wfn" is a file corresponding to Pt(NH₃)₂Cl₂, Lanl2 pseudo-potential accompanied by Lanl2DZ basis set is used for Pt and Cl, while 6-31G* is used for other atoms. "examples\Pt_lanl2.wfx" and "examples\Cl_lanl2.wfx" are the Pt and Cl atomic .wfx files produced by G09, in which Lanl2 is also used, therefore their EDF fields represent the inner-core electron

density of Pt and Cl replaced by Lanl2.

In order to borrow EDF information from the atomic .wfx files in the analysis of Pt(NH₃)₂Cl₂.wfn, first we need to set "isupplyEDF" parameter in settings.ini file to 1. Then boot up Multiwfn and input following commands

```
examples\Pt(NH3)2Cl2.wfn
Pt    // Load EDF information for element Pt
examples\Pt_lanl2.wfx // Take EDF information of Pt from this file
Cl    // Load EDF information for element Cl
examples\Cl_lanl2.wfx // Take EDF information of Cl from this file
q    // We have finished, exit
```

Now we can perform wavefunction analysis as usual. But it is better to first carry out some tests to check if inner-core electron density has been properly represented, for example, we integrate electron density over the whole space

```
100  // 100 Other functions (Part1)
4    // Integrate a function in whole space
I    // Electron density
```

The result is 132.00, which is the expected total number of electron of Pt(NH₃)₂Cl₂. Assume that we did not load the EDF information, then the result will be 52.00, which is just the number of valence electrons of Pt(NH₃)₂Cl₂.

Note that you can also directly input atomic indices instead of element name, for example, inputting 4,8-10,11 means selecting atoms 4,8,9,10,11 in present system. Of course, the atoms you selected each time must corresponds to the same element and the same pseudo-potential.

It is users' responsibility to prepare the atomic .wfx files. Since there are too many kinds of pseudo-potentials and elements, evidently I am uncapable of providing all of them for you.

5 Check sanity of wavefunction

The input files of Multiwfn generated by various quantum chemistry codes are not always standard. For example, .molden file generated by many programs are problematic in content or format. After you feeding them to Multiwfn and you want to make sure if the wavefunction has been correctly loaded, there are two useful ways to check sanity:

(1) Enter main function 100 and select subfunction 4, then choose electron density. If the integral of electron density over the whole space is very close to the actual number of electrons, then the wavefunction is ready for use.

(2) Enter main function 1000 (a hidden function) and select subfunction 100. This function will check satisfaction of normalization condition of all orbitals, and show you the maximum deviation to unity and integer. If both the two maximum deviations are evidently larger than zero, then there must be some severe problems in the input file; if any one of them is very close to zero, the input file should be fully compatible with Multiwfn.

6 Special functions

There are some special functions in Multiwfn, they are mainly used for debugging, and some

of them are requested by some Multiwfn users. Here a few of them are mentioned.

(1) Calculate nuclear attractive energy between a fragment and an orbital

You can use this function by entering subfunction 90 of main function 1000 (hidden in main menu). This function calculates attractive energy between all nuclei in a user-defined fragment and an orbital, namely:

$$E_{\text{MO}i-\text{frag}} = \left\langle \varphi_i \left| \sum_{A \in \text{frag}} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \right| \varphi_i \right\rangle \equiv \int |\varphi_i(\mathbf{r})|^2 \sum_{A \in \text{frag}} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} d\mathbf{r}$$

During the calculation, the atomic contributions to the result are outputted in turn, for example, below output

```
Processing center    2(H) /      3
Accumulated value:   -8.3269642826  Current center:      -0.0629848533
```

means

$$E_{\text{MO}i-\text{frag}}^{2\text{H}} = \int w_{2\text{H}}(\mathbf{r}) |\varphi_i(\mathbf{r})|^2 \sum_{A \in \text{frag}} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} d\mathbf{r} = -0.06298$$

where $w_{2\text{H}}(\mathbf{r})$ is atomic weighting function of 2H atom defined by Becke's partition.

The integral is evaluated by Becke's multi-center integration method, therefore the *radpot* and *sphpot* in settings.ini affect integral accuracy. Commonly the default value is accurate enough. Note that, as shown in above formula, the occupation number of orbital is not taken into account in the calculation.

Information needed: GTFs, atom coordinates