anual for Aroma – 2.0

OS: Linux, Mac, Windows

Contents

This program should be cited as Rahalkar, A.; Stanger, A. "Aroma", Amnon Stanger - הפקולטה לכימיה בטכניון : הפקולטה לכימיה בטכניון (technion.ac.il). The program contains several NICS-scan based methods that are to be cited according to their uses.

(a) NICS-scan: Stanger, A. *J. Org. Chem.* **2006**, *71*, 883-893.

(b) Sigma-only model: Stanger, A. *J. Org. Chem.* **2010**, *75*, 2281-2288.

(c) NICS-XY-scan: Gershoni-Poranne R; Stanger, A. *Chem. Eur. J.* **2014**, *20*, 5673-5688.

(d) ∫NICS: Stanger, A. *J. Phys. Chem. A.* **2019**, *123*, 3922-3927.

(e) NICS-scan at larger distances: Stanger, A. *ChemPhysChem*, **2023**, *24*, e202300080.

1. Aroma: Introduction

Aroma is a utility package for evaluating aromatic properties via NICS methods, preferably by $NICS_{\pi}$,zz. It is designed as a "Plug-In" utility for the computational chemistry packages; Gaussian 09, Gaussian 16, Orca 5 and Orca 6. NBO 6 and NBO 7 are also supported for the purpose of CMO-NICS calculations (using the NCS procedure). It offers automated building of input files, calculations, and analysis of output files for the following methods:

NICS-scan (reference a). May be applied in conjunction with NCS and/or the σ-only model (reference b) for $NICS(r)_{\pi}$,zz.

NICS-XY-scan (reference c). May be applied in conjunction with NCS and/or the σ-only model (reference b) for $NICS_{\pi}$,zz values.

∫NICS (reference d). Should be applied with NCS and/or the σ-only model (reference b).

NICS-scan at larger distances (reference e): Should be applied with NCS and/or the σ-only model (reference b).

In addition, Aroma contains utilities that allow computations of other molecular properties that may be needed, such as calculations of the area of ring(s) in a system, further analysis of the data, etc. For details see utility scripts in the appendix.

Aroma is capable of handling multiple centers within a given molecule within a single run and execute such multiple runs corresponding to different tasks by a single command.

Requirements:

The QM package that you are using must be installed on the same machine.

2. Getting Started:

For most users, it is best to download the binary version (for Linux/Mac and/or for Windows). This does not require any further installation. For Python programmers the source code is also available.

(a) Download the version you need for your operating system.

▪ For Linux/Mac: aroma-2.0.tar.gz, Put it in a directory that you wish (for example, /home/username) and untar it using "tar zxvf aroma-2.0.tar.gz". This will create a new directory (for example, /home/username/Aroma-linux) with all the Aroma files.
▪ For Windows: download aroma-Win-2.0.zip, and extract the folder "aroma-win" on the appropriate destination on your machine.
▪ The Python source code + a text file listing all the required libraries that must be installed before running the program.

It is noted that the explanations in this manual are given for the binary versions. It is expected that those using the source code will be able to run, edit and/or augment it without further instructions. One may refer to the community forum on github: link

(b) Editing the user_aroma_constants.py file – see appendix 1.

(c) Creating an arm file.

The arm file is the primary input file that Aroma runs. It contains all the information required by Aroma to execute the requested job. The full list of keywords is given in appendix 2. Here we describe only the necessary and the most popular keywords. Please note that options which were not available in Aroma 1 are marked in blue.

Please note that the keywords for the .arm file are case-insensitive and the order of keywords does not matter. Aroma also ignores any extra empty lines or any

extra text that it does not identify as a valid keyword. Hence, feel free to write your poem in the file and it will still run, but make sure that your keywords are spelled correctly.

A keyword GEOMFILE tells Aroma where to find the geometry for the calculations, with the full path.

Geomfile=<full path>/<file name> **required**
Any standard input or output file (Gaussian or ORCA) can be used as a geomfile. For example:

Geomfile=/home/username/output/benzene.log

A keyword AROMARUN tells Aroma what you want to do. Its format is
Aromarun= <list of options> **required**

For example, for getting CMO-NICS$_{\pi,zz}$ one would use

Aromarun=nicsscan, ncs

For getting a NICS-XY-scan with NICS$_{\pi,zz}$ values (reference c) using the σ-only model (reference b) for one would use
Aromarun=nicsscan, xy, sigma

A full list of the aromarun options with possible combinations between the options is given in Appendix 2.

The third line is center=<A1,A2,A3.....An> where A are the number of atoms at the center of which (and above) you want the BQs to be placed. **At least one center is required**. For example, the "center=" for the D_3h-benzene shown in Figure 1 is:

Center=1,4,2,5,3,6 Note: number are separated by a comma – no space.

The order does not matter but the connectivity has to describe the system. Thus, equivalent representations are, for example,

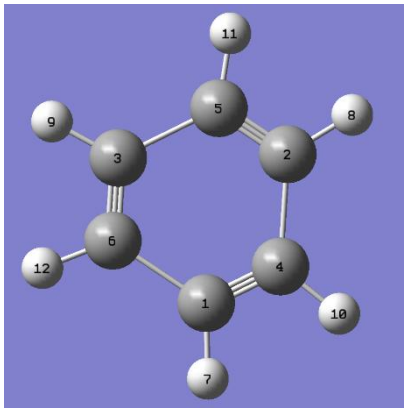


Figure 1

Center=6,3,5,2,4,1

Or

Center=2,4,1,6,3,5

Etc.

For XY scan, the "center=" defines the points along the trajectory of the BQs. Here, a bond and/or an atom may serve as a "center=", however, at least one complete cycle has to be listed (according to which Aroma decides the perpendicular direction where the BQs will be added). Figure 2 shows naphthalene with the desired trajectory.

Going from left to right, the definition of the trajectory will be:

Center=1,6

Center=1,2,3,4,5,6

Center=3,4

Center=3,4,11,14,15,10

Center=14,15

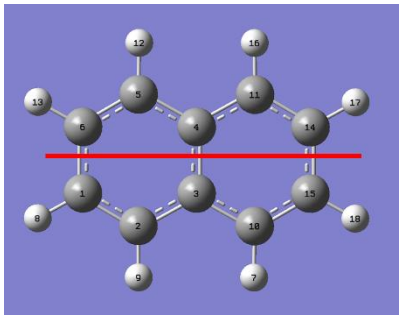


Figure 2

An alternative definition may be:

Center=1,6

Center=1,6,5,4,11,14,15,10,3,2

Center=14,15

In this case, the two trajectories are identical due to the symmetry of the system. However, the first alternative gives more defined points on the trajectory, therefore fits also less symmetric systems. It is recommended to define as many as possible "centers" along the trajectory.

Definitions related to the σ-only model.

When "sigma" is asked for in the "aromarun=" command, the (anti)aromatic rings on which the hydrogens have to be bound are specified here. The format is

Aromatic ring
A1,A2,A3…An
B1, B2, B3….Bn
.
.
.
N1,N2,N3…..Nn
End

Were A1-An, B1-Bn,….N1-Nn are the (anti)aromatic rings in the systems.
Note: "Aromatic ring….end" is needed only if the σ model is asked for (in Aromarun=). It is not needed for $NICS_{zz}$ or from CMO-NICS.

For example, for the benzene case (figure 1) the definition is:

Aromatic ring
1,4,2,5,3,6
End

For the Naphthalene case (figure 2) the definition is:

Aromatic ring
1,2,3,4,5,6

3,4,11,14,15,10
End

Other options for aromarun= Note: for details look in Appendix 2.
Zintegral: scanning at higher elevations (default 2-5 Å). This produces ∫NICS (reference d) NICS(1) and NICS(1.7) (reference e).

Inpuonly and outonly: inponly creates the necessary input files and stops. These input files may be edited and saved under the same name and executed as "regular" Gaussian or ORCA files. Inponly also copies the arm file (name.arm) to arm.org (name.arm.org) file and changes the inponly command in the arm file to outonly. When the manual execution of the input files is over, aroma uses the arm file to take the output files, filter and collate the relevant data and further analyze the results. The inponly file is useful when Aroma cannot produce the correct or desired input file(s). outonly command is useful for further analysis of the output file. For example, if the CMOs where not correctly identified in the original run, one can use the outonly option with pimos=A1, A2…An (A1, A2…An are the π-MOs) to get the correct CMO-NICS values.

Other additional commands:

Sonly charge=. Aroma determines automatically the charge of the sigma model. However, this can be overridden by the sonly charge command. Please be careful when using this option, since a wrong determination of charge may cause an error in the QM program (for example, the request of a closed-shell system with odd number of electrons).

Sonly mult=. Aroma determines the multiplicity of the sigma model. Sometimes it is necessary to have a different multiplicity. For example, in triplet state systems with extended π system, the electronic state results only from the π system. Therefore, its sigma model should be a singlet. Specifying sonly mult=1 results in a singlet sigma model even if the system under study is a triplet.

Pimos=A1,A2…An. When aromarun=ncs is used, Aroma tries to identify the π MOs for obtaining CMO-NICS values. This procedure may result in wrong assignments, i.e., either not including all the π CMOs or including non-π CMOs. If you know (from previous run) which are the π-MOs you can use this command. For example, for benzene it would be pimos=17,20,21.

3. Usage:

a. line Version:

Linux: ./aroma <full path to the arm file without extension> &

b. GUI Version

Linux: double click on the Aroma_GUI file or type ./Aroma_GUI &

Click the "browse" button to select the arm file and click on the "run" button.

Windows: double click on the Aroma_GUI icon. Click the "browse" button to select the arm file and click on the "run" button.

4. During Running Aroma

Aroma prepares the input files and some run-related (<name>.json) files. The QM package (Gaussian or ORCA) is activated and runs these input files sequentially, so that the user can run more than one Aroma job parallelly. At the end of a successful run the temporary files may be deleted, unless the inponly keyword was used or further analysis is wished. In this case it is important to keep this json files for running the continuation (i.e., the same arm file, now with the outonly keyword).

Please note: Depending on the maxAtomsInInputFile parameter in the user_aroma_constant.py file (see appendix 1) Aroma will produce multiple input files for the same "center". The json files are necessary for combining these results for the analyses and the output files (see below).

5. After the run

Aroma produces numerous files. These are:

(1) A graphic (jpg) file. This file shows the results of the scan(s). It is not meant to be for quantitative uses. Rather, it indicates if the run produced reasonable results. This is very useful for making sure that there was no error in the arm file.

(2) armlog (text) file. For z-scans (i.e., nicscan or zintegral) it contains the analyses results and the final $NICS(r)_{zz}$, $NICS(r)_{\pi,zz}$ (r=1.0, 1.7) and ∫NICS values. For XY-scan there are two armlog files. One containing the coordinates of the "centers" (as defined in the arm file) along the scan trajectory. If "sigma" was asked for, the second armlog file is named <name.-alldiff>.armlog and contains the values of $NICS_{zz}$ of the system and of the sigma model and the difference between the two (i.e., $NICS_{\pi,zz}$) for each of the points along the trajectory.

(3) Picmo (text) files. These files are created whenever NCS keyword is used. It contains the contribution of each π-CMO to the $NICS_{zz}$ of each BQ and their sum which is the CMO-$NICS_{\pi,zz}$ for each BQ. For Z-scan it contains the data that is used for the analyses (see (2) above). For XY scan it contains the $NICS(1.7)_{\pi,zz}$ value for each point along the trajectory, as defined in the arm file.

(4) Armdat (text) files. Contains the crude NICS data for each BQ.

(5) A zipped library, containing all the input, output, armdat and (in case NCS is asked for) the picmo files.

The GUI for Aroma can be launched by double clicking on the application named as **aroma_gui.exe** or by issuing **./aroma_gui** from command line when in aroma directory. The GUI allows user to select the .arm file and initiate a run by pressing appropriate button. Once the execution of Aroma job starts, the status and runtime output of the Aroma is displayed on the GUI.

4. Input:

Aroma needs two input files.

1. An arm file which contains Aroma keywords. The keywords for Aroma are explained in detail in Appendix 2. For further details, please refer to the sample .arm files in Sample directory of Aroma.
2. The molecular geometry in one of the following options: (1) a standard Gaussian or Orca input file (as Z-matrix or as Cartesian Coordinates). A Gaussian or ORCA output file. (3) A checkpoint file. The path for this file is specified through the .arm file. Since Gaussian software allows use of various extensions to its standard input and output files, Aroma supports all the extensions, but the user has to configure the desired options in the user_aroma_constants.py file.

5. Illustrations

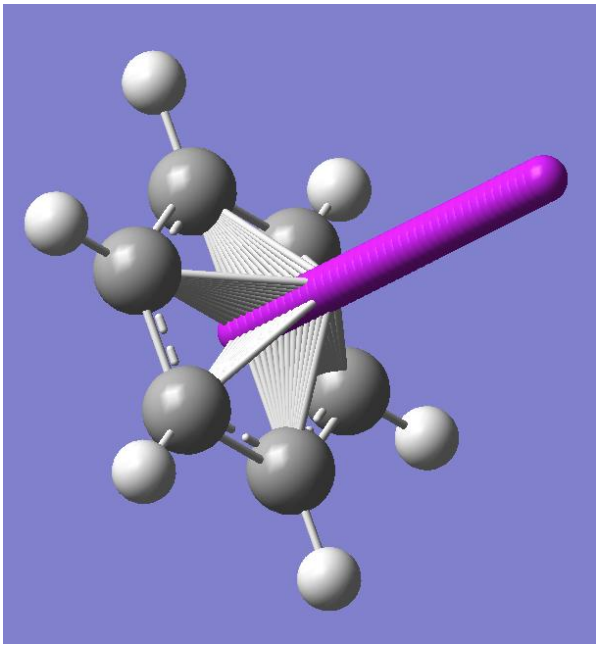
This Section explains the necessary details of the illustrative calculations which are included in Sample folder of Aroma.

1. Benzene, Z-scan, CMO and σ-only model

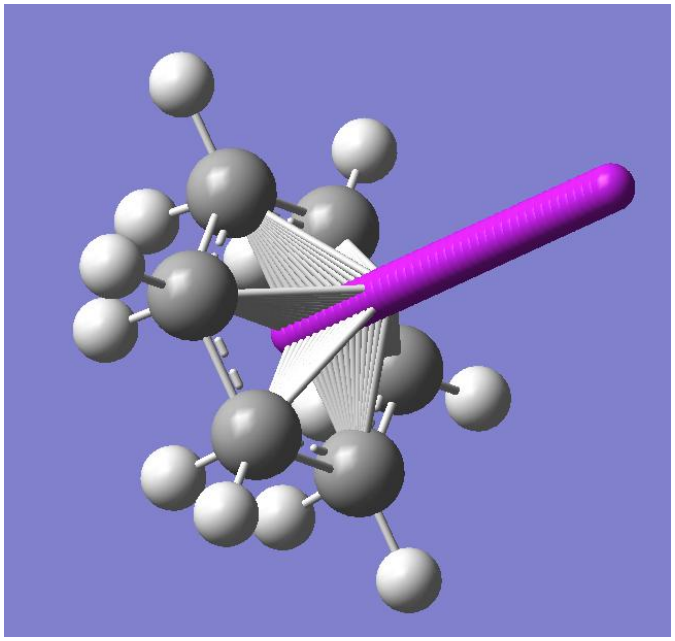
Input: benzene-z-scan-seminar.arm, benzene-4-seminar.log

Output: benzene-z-scan-seminar.armlog, benzene-z-scan-seminar-center1(.armdat & .picmo), benzene-z-scan-seminar-sigma-center1.armdat, benzene-z-scan-seminar-sigma-plot1.PNG and an archive (.zip) containing all the input (.in), output (.log), armdat and picmo files.

Description: A NICS-Scan in Z-direction with CMO and the σ-only model.



Benzene with BQs



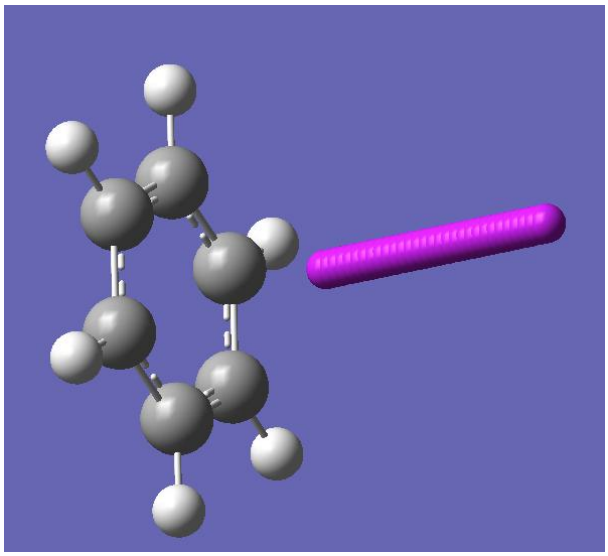
Benzene's σ-only model with BQs.

2. Benzene, Z-integral, CMO and σ-only model

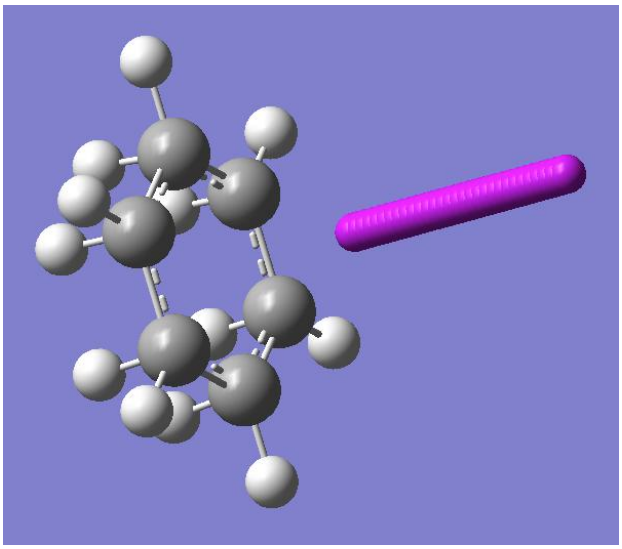
Input: benzene-z-int-seminar.arm, benzene-4-seminar.log

Output: benzene-z-int-seminar.armlog, benzene-z-int-seminar-center1(.armdat & .picmo), benzene-z-int-seminar-sigma-center1.armdat, benzene-z-int-seminar-sigma-plot1.PNG and an archive (.zip) file containing all the input (.in), output (.log), armdat and picmo files.

Description: A NICS-Scan in Z-direction at 2-5 Å distances for ∫NICS, $NICS(1)_{\pi,zz}$ and $NICS(1.7)_{\pi,zz}$ using CMO and the σ-only model.



Benzene with BQs



Benzene's σ-only model with BQs.

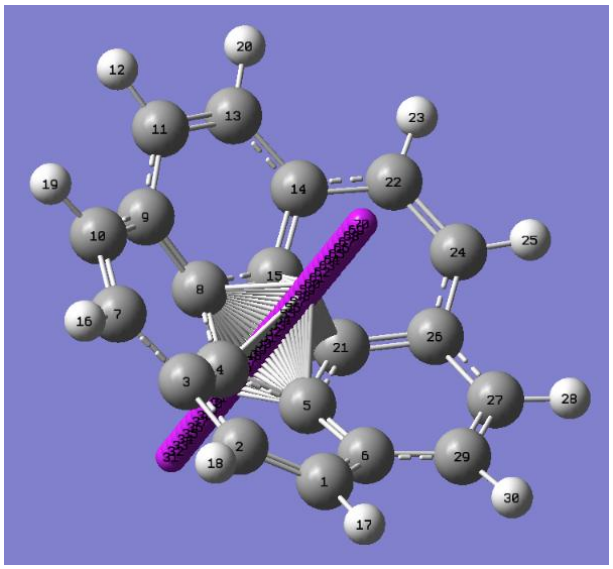
Please note: The armlog contains fitting of several NICS parameters. The important results are Integral-NICS(CMO) = -82.1928, Integral-NICS(delta-ZZ) = -82.2819, for CMO NICS(1) = -33.8509 NICS(1.7) = -17.4168 and for delta-zz NICS(1) = -34.387 NICS(1.7) = -17.315.

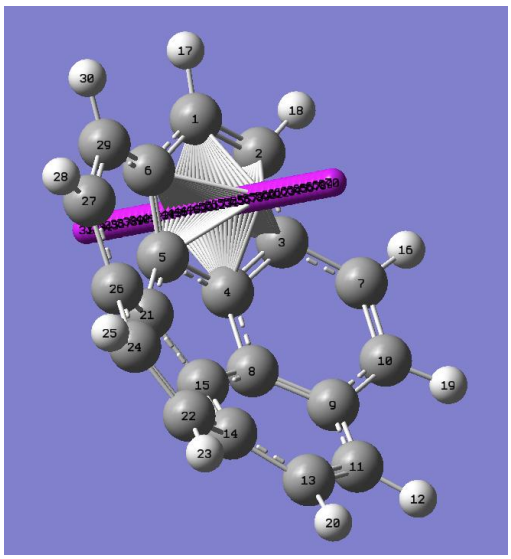
3. Corannulene, Z-scan.

Input: corannulene.log, corannulene-z-scan-range-step.arm

Output: corannulene-z-scan-range-step.armlog, corannulene-z-scan-range-step-plot1.PNG and a library (zip) containing the input (.in), output (.log) and armdat files for each center.

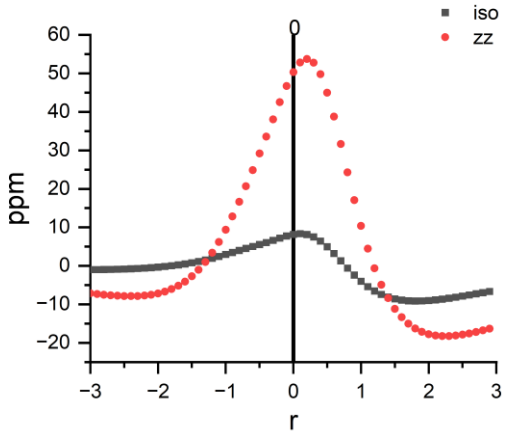
Description: Utilization of the bqrange and bqstep in a z-scan. BQRANGE (default: 0 to 3.9) has been set to -3,3 for showing the different tropicity on each side of the system. The BQSTEP (default 0.1) has been set to 0.15.

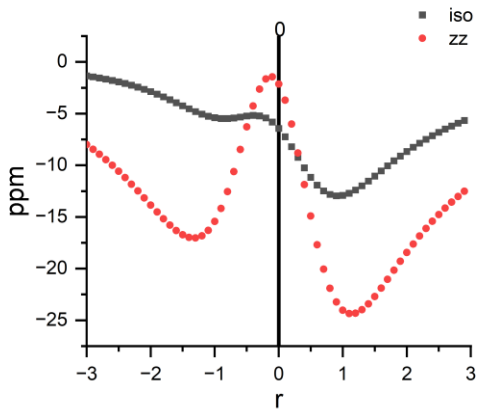




Z-scan, central ring

Z-scan, side ring.





Z-scan, central ring

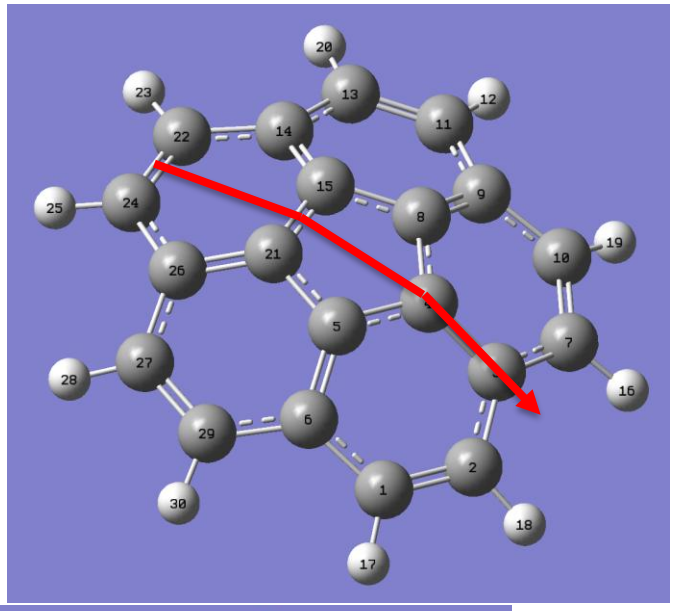
Z-scan, side ring.

4. Corannulene, XY-scan, sigma model, direction

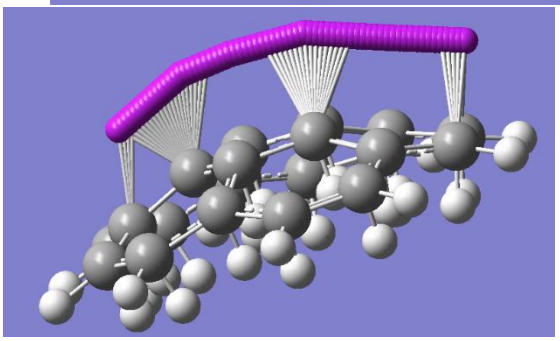
Input: corannulene.log, corannulene-x-scan-direction-1.arm, corannulene-x-scan-direction-2.arm.

Output: the two directions ae separate Aroma jobs. All the files named *direction-X are direction-1 and direction-2. corannulene-x-scan-direction-X-alldiff.armlog, corannulene-x-scan-direction-X-center1.armdat, corannulene-x-scan-direction-X-sigma-center1.armdat, corannulene-x-scan-direction-X-sigma-plot1.PNG and the two libraries (corannulene-x-scan-direction-X-archive.zip) containing all the input (.in), output (.log) and armdat files.

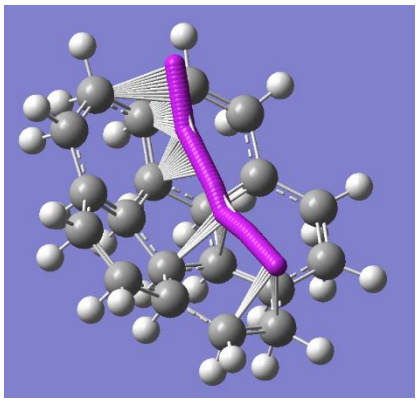
Description: In some cases (for example, non-planar systems where the tropicity on each side is different) it is wished to specify the side on which the scan (i.e., Z scan and/or XY scan) is performed, and, if the σ only model is used, at which side the hydrogen atoms are bound. The command forcedir determines these directions. This example shows an X-scan with the same definitions of "center"s but on opposite faces of the corannulene.



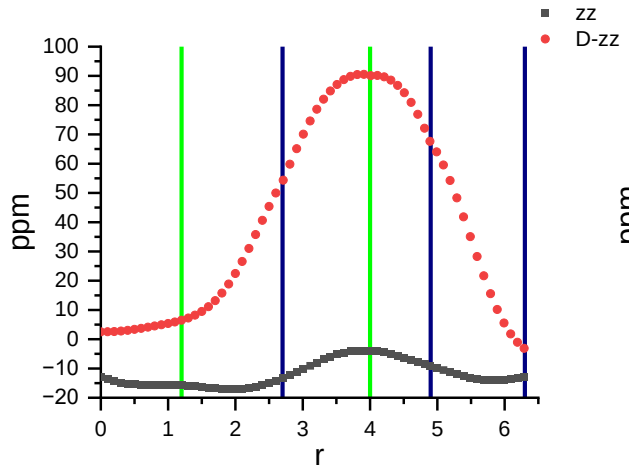
X-scan trajectory



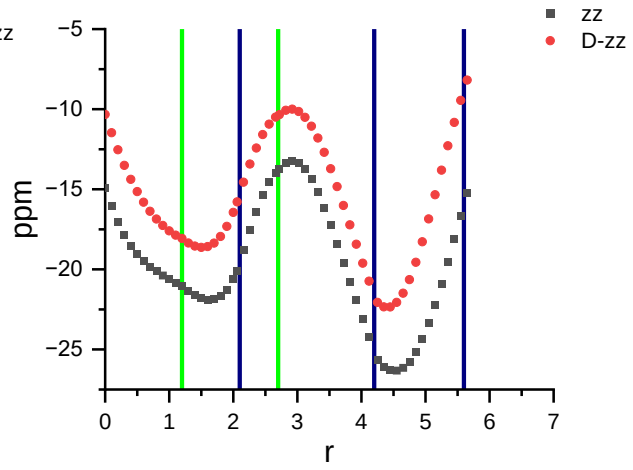
σ only model with forcedir=22,24,26,21,15,14



σ only model with forcedir=14,15,21,26,24,22



NICS-X-scan with forcedir=22,24,26,21,15,14 (“outside”). Green line – center of a ring. Blue line – atom or center of a bond.



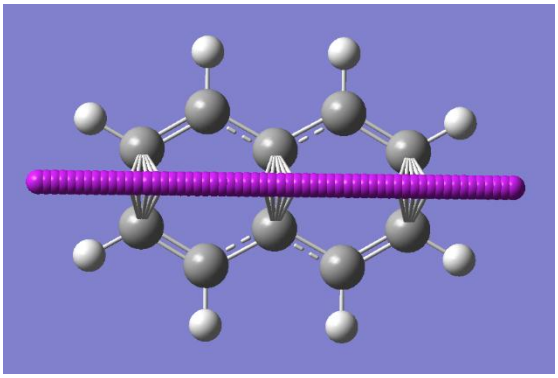
NICS-X-scan with forcedir=14,15,21,26,24,22 (inside) Green line – center of a ring. Blue line – atom or center of a bond.

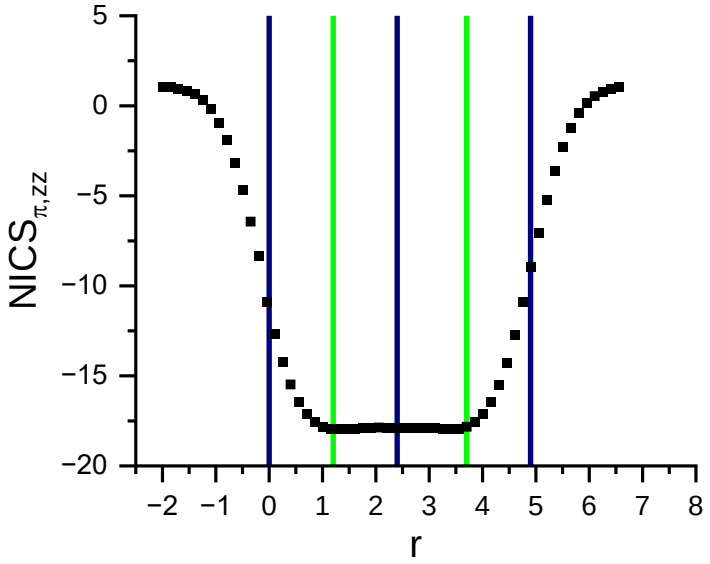
5. Naphthalene, XY-scan, BQSTEP, CMO, EXTEND.

Input: naphthalene.log, naphthalene-xy-extend-2-bqstep.arm

Output: naphthalene-xy-extend-2-bqstep.armlog, naphthalene-xy-extend-2-bqstep-center1(.armdat, .picmo), naphthalene-xy-extend-2-bqstep-plot1.PNG and an archive (zip) containing the input file (.in), otput file (.log) and the armdat and picmo files.

Description: an X-scan of naphthalene, starting 2 Å before the fist “center” (bond), anding 2 Å after the last “center” (bond), using xyextend=2.0 and asking for a BQ each 0.15 Å (overriding the 0.1 Å default) using bqstep=0.15.





NICS-X-scan of naphthalene with xyextend=2.0 and bqstep=1.5. Green line – center of a ring. Blue line –center of a bond.

Appendix 1: Editing the user_aroma_constants.py file

This file contains all the settings and controls and background data that is needed to running Aroma. It should be tuned to the specific systems that you are using. It also contains all the atoms parameters, so that here you can add atoms that are not listed in the default data. It also contains the scan parameters; The distance between the NICS probes (BQ in Gaussian, H: in Orca) and the scan regions for the different procedures. Below is a detailed description of the different parts of the file which can or should be edited.

(1) Atom information. Here all the information about the atoms is given. Elements that are not included may be added. For example, if calculations of systems that include Ge are wished, the parameters of Ge must be added. In the AtmSym line add 'GE':32. In the AtmMass add 32:72.64, in the AtmCovalentRadii add 32:1.20 and in the Max_Conn line add 32:4. Please keep the format of the included atoms. Please note: (1) The covalent radii and the atomic mass are taken from the literature. (2) The Max-Conn (maximal connectivity) of an atom is mainly used for the σ-only model. If the user wishes to extend this (for example, allow Si to be six-coordinated) change 14:4 to 14:6.

(2) Setting of the QM package: Gaussian

(a) Locations of the files; where do you want the input files, output files and checkpoint files. These are called inpdir, outdir and chkdir, respectively. For example "inpdir": "/home/username/input/",

(b) Inputs and outputs extension. For example, com for input, out for output. These are given in the inpExt and outExt. For example, "inpExt": ".gjf", and "outExt": ".log",. Please note: The user can use and input and output extensions wished (even if not standard Gaussian or Orca extensions) as long as they are added here.

(c) Command for running the Gaussian and its utilities. For example, if your Gaussian 16 is installed at /use/local/, then "extCmd": "/usr/local/g16/g16 ", and "chkCmd": "/usr/local/g16/formchk ",

(d) Some installations of the QM package have size limitations as per the local clusters or computational capacity of the computers. This is the reason for having the "maxAtomsInInputFile". Suppose that you have a limitation of 100 atoms in the input file, your molecule contains 60 atoms, and your scan requires 120 BQ. Aroma will create 3 input files, each with the

molecule and the appropriate number of BQs. It will run the three jobs, collect the data and you will obtain one file with all the NICS data. If there is no limitations in your installations just enter a number that exceeds you usage, for example, "maxAtomsInInputFile": 1000,. Please note: Each of these input files runs from the beginning, including the guess and all the SCF cycles. Thus, it is a good idea to find out the maximum numbers of atoms that your installation allows to minimize the number of input files and the Aroma run time.

(e) Default routes: These are the routes that will be printed in the Aroma-generated input files. There are three types of routes:

1. Optimization route. Aroma can be used to optimize geometry. Suppose you wish to optimized your molecule with the M06-2X functional, using def2pvdz basis set. You would like to use 4 cores with 6144 MB shared memory and run frequencies calculations. This will appear as "defaultOptimizationKeyline": "%nproc=4\n%mem=6144MB\n# M06-2X/def2pvdz OPT freq \n",
2. NICS calculations (for the system and the σ-only model). Suppose you wish to use here the B3LYP functional with the 6-311+G(d) basis set, using 8 cores and 12288 MB memory. The NMR method is GIAO, and you wish to have accurate NICS values. In some cases, the convergence of the σ-only model I slow, so more than the defaults SCF cycles are needed. The rout will then be "defaultNicsKeyline": "%nproc=8\n%mem=12288MB\n# B3LYP/6-311+G* 6D scf=maxcycle=2000 NMR=GIAO INTEGRAL=(GRID=ULTRAFINE) CPHF=(GRID=FINE)\n",
3. CMO-NICS calculations: This calculations requires activation of NBO6 or NBO7. Thus, at the end of the input file there is an NBO definition of what NBO is required to do. The example here is running a NICS job at the same computational level as is (2), but asking NBO 7. "defaultNcsKeyline": "%nproc=8\n%mem=12288MB\n# B3LYP/6-311+G* 6D NMR=GIAO IOP(10/46=1) POP(NBO7READ, FULL) INTEGRAL=(GRID=ULTRAFINE) CPHF=(GRID=FINE)\n",. The line that tells NBO what to do is "defaultNboKeyline": "$NBO NCS=0.1,I,MO,XYZ $END\n", and it will appear at the end of the Aroma-generated input.

The following section describes the same actions with ORCA. It is completely equivalent, except that ORCA keywords are used. Please note: If you are using only one QM package (i.e., Gaussian or Orca) you have to update these parameters only for the relevant QM package used.

(3) Defaults for the different NICS-scan methods. The current values are the defaults, as appear in the scientific papers (references a-e). However, the user is able to changes these defaults. For example, in some systems the scans at short distances deviates from regularity, so that the the point for analysis should be taken from a larger distance from the molecular plane, for example, afrom 1.5 Å. This can be changed by converting the DEFAULT_DISTANCE_FOR_ANALYSIS = 1.0 to DEFAULT_DISTANCE_FOR_ANALYSIS = 1.5.

(4) Next section is the defaults for generating the σ-only model. It is recommended not to tamper with these values. Please note that if you added an atom (section 1 above) a default bond length between the new atom and hydrogen atom must be added. Taking the above example of adding Ge, the value , 32:'1.50' should be added to the ATM_H_BL = line.

(5) The last section is activation of an e-mail msg. to be sent to your e-mail when the Aroma job is finished. This is possible only if the server that you are using is configured as a mail server. The user should change the e-mail address to his/her own in the line "to_user": "user-mail ",. It is noted that under some Linux versions this section causes a fatal error in aroma. Thus, it is recommended to add # at the beginning of each line (this converts the line to a comment) and run Aroma without it. When successful, delete the # and test it again.

Appendix 2

Full list of Aroma 2 keywords

Notes:

1. All the keywords are **case-insensitive**.

2. The order of keywords and extra blank lines between keywords do not matter. Insert a blank line at the end of the arm file

3. If the molecule does not possess overall planarity, then for each ring, the entire molecule will be re-oriented in such a way that the ring under consideration lies in XY plane. This allows the program to clearly define the in-plane and out-of-plane directions.

GEOMFILE

Description:

Compulsory.

This is a path for a file containing molecular geometry. This file can be a Gaussian/ORCA input, output or checkpoint file.

Notes:

1. The extension of input, output or checkpoint file should match with those defined in user_aroma_constants.py

2. Along with the geometry, the charge and multiplicity of the system are also taken from this file.

3. The geometries are considered to be in angstroms (and degrees in case that a z-matrix format is used). Even if the geometry is provided through checkpoint file, it is converted to angstrom units by Aroma.

```
Example:
GEOMFILE=/home/anuja/input/benz.in
# when GaussInpExt (in user_aroma_constants.py) is set as '.in'
Or GEOMFILE=/home/anuja/output/benz-opt.out
# GaussOutExt (in user_aroma_constants.py) is set as '.out'
```

AROMARUN

Description:

Optional, default is always NICSSCAN.

Possible values and their meanings are described below.

NICSSCAN: Default, performs NICS-Z-scan at the range that is defined in DEFAULT_BQ_RANGE line in the user_aroma_constants.py. the default is 0-4 Å range.

The following keywords are optional and should be given in addition to NICSSCAN if the user wants.

ZINTEGRAL: performs NICS-Z-scan at the range that is defined in DEFAULT_INTEGRALNICS_RANGE line in the user_aroma_constants.py. the default is 2-5 Å range.

XY: Performs NICS-Scan in XY-direction above the molecular plane at height specified by DEFAULT_XY_DISTANCE in user_aroma_constants.py file. The default is set to 1.7 Å (see reference c).

SIGMA: Performs NICS-scan calculations on a σ-only model of the original molecule (reference b)

NCS: Performs a NICS-scan and NCS analysis (included in NBO) for obtaining CMO-NICS

OPT: Performs geometry optimization (prior to NICS calculations). Please note: "opt" option can be combined with all the other options. However, it is recommended to optimize the structure before running Aroma. In some cases, optimization needs more than one cycle or leads to a different isomer.

8 Possible combinations:

(a) **Aromarun=nicsscan,sigma**
(b) **Aromarun=nicsscan,ncs**
(c) **Aromarun=nicsscan,sigma,ncs**
(d) **Aromarun=zintegral,sigma**
(e) **Aromarun=zintegral,ncs**
(f) **Aromarun=zintegral,sigma,ncs**
(g) **Aromarun=nicsscan,xy,sigma**
(h) **Aromarun=nicsscan,xy,ncs**
(i) **Aromarun=nicsscan,xy,sigma,ncs**

All the above may be combined with the inponly option.

Resulting outputs:

(a) **NICS(1)$_{\pi,zz}$ based on the σ-only model (reference (b)), scanning between 0 and 4 Å, analyzed by 3rd polynomial fitting.**
(b) **NICS(1)$_{\pi,zz}$ based on the CMO, scanning between 0 and 4 Å, analyzed by 3rd polynomial fitting.**
(c) **A combination of (a) and (b) above.**
(d) **NICS(1)$_{\pi,zz}$, NICS(1.7)$_{\pi,zz}$ and ∫NICS$_{\pi,zz}$ based on the σ-only model (reference (b)), scanning between 2 and 5 Å, analyzed by two parameters fitting (∫NICS$_{\pi,zz}$, reference d) and three parameters fitting (NICS(r)$_{\pi,zz}$, reference e).**
(e) **NICS(1)$_{\pi,zz}$, NICS(1.7)$_{\pi,zz}$ and ∫NICS$_{\pi,zz}$ based on CMO, scanning between 2 and 5 Å, analyzed by two parameters fitting**

($\int NICS_{\pi,zz}$, reference d) and three parameters fitting ($NICS(r)_{\pi,zz}$, reference e).

(f) The combination of (d) and (e) above.

(g) $NICS(1.7)_{\pi,zz}$-X-scan (reference c) based on the σ-only model (reference (b)).

(h) $NICS(1.7)_{\pi,zz}$-X-scan (reference c) based on CMO.

(i) Combination of (g) and (h) above.

Example:

RUN=NICSSCAN,SIGMA # NICSSCAN calculations for original molecule and its σ-only model in Z-direction.

Or

RUN=XY,NICSSCAN,SIGMA # NICS-XY-SCAN calculations for original molecule and its σ-only model. Please note that XY,NICSSCAN is actually one keyword asking for the NICS-XY-procedure.[d]

Or

RUN=OPT,NICSSCAN,NCS # Geometry optimization followed by NICSSCAN and CMO-NICS calculation for the final optimized geometry.

OPT_EXTERNAL

Description:

Optional

If AROMARUN = OPT, then user may give path and filename for the input file for optimization directly using this keyword instead of automatic generation of this file by Aroma.

This is especially useful when user wants to perform constrained optimization using Z-matrix or modredundant or using an input file which does not contain a geometry (geom=CheckPoint) or use previous results (e.g., frequencies, geometry) from a checkpoint file.

Note: This file will be directly used for geometry optimization as it is. In other words, all the Gaussian keywords including method, basis set, chk, memory, processors etc. will be used from this file and not from Aroma.

Example:

OPT_EXTERNAL=/home/anuja/benz-opt.in # Geometry optimization will be performed using this file directly and the final geometry from the output will be obtained for usual proceedings of Aroma.

CENTER

Description:

Compulsory, at least one CENTER must be defined.

List of atoms making up the rings or bonds for which center(s) a NICS scan is to be performed.

Notes: 1. Each ring/bond should be defined in a new line starting with the keyword Center.

2. In case of XY-Scan, the geometrical centers of bonds and rings define the direction in which BQs are generated. Therefore, it is vital that the centers are specified in proper sequence.

3. A single atom or a bond is accepted as Center only in case of XY-Scan. Nevertheless, at least one "center" has to be a ring.

4. When the input geometry is in Z-matrix format the dummy atoms are not counted for the "Center" data.

Example:

```
CENTER=1,2,3,4,5  # BQs will be generated starting from the geometrical center of the ring created by atoms with indices 1 to 5.
Or in case of XY Scan
CENTER=1,6
CENTER=1,2,3,4,5,6
CENTER=3,4 # The BQs are generated on lines joining the perpendiculars of GMs of the consecutive centers.
```

NORMAL

Description:

Optional, relevant only for NICS Scan in Z-direction

A vector along which the NICS Scan in Z-direction is to be performed.

Notes: 1. This vector is specified as a list of x,y,z coordinates of two points which defines the vector.

2. This is useful if the position for NICS Scan in Z-direction cannot be defined as center of any ring.

3. The BQs are generated starting from the first point in the list.

4. This is irrelevant for XY-Scan.

Example:

```
NORMAL=0.917,1.896,0.000,0.917,1.896,-1.000 # BQs will be generated starting from the first point (0.917,1.896,0.000) along this vector.
```

FORCEDIR

Description:

Optional.

In non-planar systems (e.g., corannulene) the NICS values are different for each size of the system. Forcedir determines on which side of the system the scan (whether Z or XY) will be performed. Aroma needs only one ring to

determine the direction. Clockwise count results in the "up" direction and anticlockwise count results in the "down" direction.

Example:

For the naphthalene example (Figure 2 in page 5 of the manual)

```
forcedir=3,4,11,14,15,10 # will result in a scan trajectory above the molecule.
forcedir=1,2,3,4,5,6 # will result in a scan trajectory below the molecule.
```

POINT

Description:

Optional, relevant for NICS Scan in XY-plane.

Cartesian coordinates of a "point" in the sequence of XY centers which is included in the trajectory of XY-Scan.

Notes: 1. This is useful if certain point which cannot be defined as an atom, center of a bond or center of a ring, is to be included in the trajectory.

2. Since there is no ring-plane to define the perpendicular direction on this point, a normal vector for this point is aligned according to those on the previous and next ring-centers. In other words, the trajectory between the previous "center" and the point and/or the point and the next center will be raised to a distance above the systems, as defined in the user_aroma_constants.py file (e.g., 1.7 Å).

3. "Point" is irrelevant for scan in Z-direction.

Example:

```
aromarun = xy, nicsscan, sigma
center:3,20,21,22,23
point:-1.713,-1.680,0.000
center:1,5,6,7,8
# The trajectory is defined as vectors joining first center to point to the second center. The perpendicular for the point is average of those for the first and the second centers.
```

AROMATIC RING – END

Description:

Compulsory, for AROMRUN = SIGMA.

If σ-only model calculations are requested, the list of all aromatic (and/or antiaromatic) rings is to be provided. The σ-only model will be generated by adding H-atoms above the atoms forming all of these rings.

Example:

```
AROMATIC RING
1,2,3,4,5,6
3,4,7,8,9
END
```

 # This means there are two aromatic/antiaromatic rings in the molecule under consideration. Even if the NICS-scan calculations are to be performed only on one ring (for example, 1,2,3,4,5,6) which is defined via keyword CENTER, the keyword AROMATIC RING should list all of the rings, in order to generate the correct model. See example 3 in Illustrations.

EXOCYCLIC - END

Description:

Optional, if AROMARUN=SIGMA

If there is an exocyclic atom attached to a ring-atom, where the user would like to add dummy H-atom while generating the σ-only model, this keyword is to be used.

Example:

```
AROMATIC RING
1,2,3,4,5,6
2,3,15,11,13
END
EXOCYCLIC
11,12
END
```

 # This means there is an exocyclic atom indexed 12 attached to ring atom 11, for which dummy H-atom is to be added. See example 4 in Illustrations. This option should be used if an exocyclic double bond (e.g., $=CH_2$, =O) that is part of the conjugation in the system exists.

BQSTEP

Description:

Optional. The default distance between the BQs is defined in DEFAULT_BQ_STEP = in the user_aroma_constants.py (default value is 0.1 Å). BQSTEP overrides this value.

Step size or distance between BQs in Å.

Please note that this value is applicable for all the types of scans

Example:

```
BQSTEP = 0.2 # BQs will be generated at 0.2 Å distance.
```

XYEXTEND

Description:

Optional. Relevancy: XY-scan. The default XY trajectory starts at the first "center" (usually an atom or a center of a bond). In some cases one wishes to know what are the NICS values outside the system. This option extend the XY scan outside the first and the last "center".

Example:

```
XYEXTEND = 1.5 # BQs will be generated from 1.5 Å before and after the first and last "center"s, respectively, continuing the trajectory's line.
```

BQRANGE

Description:

Optional. Default range of the scan is defined in the DEFAULT_BQ_RANGE = in the user_aroma_constants.py file (the default is [0,4]). This range can be changed with the BQRANGE command, assuming that the geometrical center (GC) of the ring is the Origin. This option is applicable only for the AROMARUN=NICSSCAN with/without SIGMA and/or NCS.

Example:

```
BQRANGE=-4,4 # The BQs are generated on both sides of the ring through its center. This is useful for non-planar ring systems.

BQRANGE=0,5 # The BQs are generated on one side of the ring starting at the ring plane up to 5 Å. This is useful for obtaining the data for NICSSCAN and ZINTEGRAL in the same run of Aroma, but will require manual analysis.
```

ANALYSE

Description:

Optional.

If AROMARUN = NICSSCAN, SIGMA, Aroma processes the data and determines the chemical shifts at distance of 1 Å above the molecular plane based on 3rd degree polynomial fittng of Δoopc (Δ_{zz}) and 3Δiso (see reference b).

The height from which values are considered for the analysis is given in the DEFAULT_DISTANCE_FOR_ANALYSIS in user_aroma_constants.py file (default value is 1.1 Å). There are cases that the scan lines start behaving regularly only further away from the molecular plane (for example, in some 2ed and 3rd row elements). The user can instruct the program to consider the values from a larger distance for the analysis.

Example:

```
ANALYSE = 1.5

# For polynomial fitting, data of BQs from distance 1.5 Å and above is considered

ANALYSE AREA
```

 # In addition to analysis, also calculates the area for all the given aromatic rings as defined in the "aromatic rings" section.

OUTFILE

Description:

Optional.

If AROMARUN = SIGMA, the results of analysis are stored in this file. By default the results are stored in a file with the same filename as that of ".arm" file which has extension ".armlog" in the output directory "outdir".

Example:

OUTFILE=/home/anuja/output/benz-aroma.log

 # The chemical shifts for each center will be stored in this file instead of "benz.armlog".

SONLY CHARGE

Description:

Optional.

User defined charge on σ-only model, if user does not want to use the automatically calculated charge.

Example:

SONLY CHARGE = -1 # Charge of -1 will be used for σ-only model

SONLY MULT

Description:

Optional.

The multiplicity of the σ model is assumed to be the same as the multiplicity of the molecule (taken from geomfile). In some cases, for example, when the molecule is in the triplet state, it is desired to assign different multiplicity to the σ model. This keyword overrides the automatic assignment of multiplicity and allows the user to choose it.

Example:

SONLY MULT = 1 # the multiplicity of the σ-only model is assigned to be a singlet

KEYLINE – END KEYLINE

Description:

Optional.

The lines of keywords for running Gaussian.

These have further sub-keywords :

NICSSCAN_TEMPLATE

Description:

Optional, with following defaults in user_aroma_constants.py.

DEFAULT_NICS_KEYLINE = "%nproc=1\n%mem=1024MB\n# B3LYP/6-311+G* NMR=GIAO\n"

The % keywords such as chk, nproc, mem etc. and the route command starting with #, specifying the level and basis set etc. These lines are attached to the NICS-SCAN input for each “Center” as specified.

NCS_TEMPLATE

Description:

Optional, with following default parameters in user_aroma_constants.py.

DEFAULT_NCS_KEYLINE = "%nproc=1\n%mem=1024MB\n# B3LYP/6-311+G* NMR=GIAO IOP(10/46=1) POP(NBOREAD, FULL)\n"

Template for NICS-scan run with Gaussian with NCS run by NBO package.

Note: If AROMARUN = NICSSCAN, NCS then the keylines should be specified under **NCS_TEMPLATE** and not under **NICSSCAN_TEMPLATE**, or Aroma will use the defaults for NCS keylines.

OPT_TEMPLATE

Description:

Optional, with following default parameters in user_aroma_constants.py.

DEFAULT_OPTIMIZATION_KEYLINE = "%nproc=1\n%mem=1024MB\n# B3LYP/6-311G* OPT \n"

Template of keywords for optimization if AROMARUN = OPT

NBO_TEMPLATE

Description:

Optional, with following default parameters in user_aroma_constants.py.

DEFAULT_NBO_KEYLINE = "$NBO NCS=0.1 <I MO XYZ> $END\n"

As per requirement of NBO package, these keywords are attached at the end of the Gaussian input file.

Note:

1. Since **KEYLINE – END KEYLINE** is a complete set of keywords, there should not be extra blank lines in this section.

2. The order of sub-keywords does not matter.

3. If your **TEMPLATE** contains "%chk" keyword, then Aroma will add an appropriate filename for each "Center". Please refer to the example below.

4. This option is useful if you want to use parameters which are different from those defined in the user_aroma_constant.py file and override them. If the change is going to be constant it is a better practice to change the respective keylines in the user_aroma_constant.py file avoiding the use of the **KEYLINE – END KEYLINE** option.

Example:

```
KEYLINES
OPT_TEMPLATE
%chk=/home/anuja/chk/benz-opt.chk
%nproc=8
%mem=4000mb
# M06-2x/aug-cc-PVDZ opt
NCS_TEMPLATE
%chk=/home/anuja/chk/benz.chk
%nproc=16
%mem=4000mb
# pb0pb0/def2-pvdz nmr=giao iop(10/46=1) pop(nbo7read, full)
END KEYLINE
```

 # Instead of the defaults the Gaussian calculations will be performed at these specified levels of theory with specified hardware controls.

As "chk" is specified in the keylines:

1. The specified chkfile will be generated for the optimization.

2. For NICS-scan and CMO runs, %chk=/home/anuja/chk/benz-center1.chk will be used for center no. 1 and so on for the rest of the centers, assuming chkdir="/home/anuja/chk/" as in user_aroma_constants.py.

CLEAR

Description:

Optional.

During Aroma run, a lot of input and output files are generated. This keyword removes all the intermediate files including inputs and outputs corresponding to centers.

Example:

```
CLEAR # This should be in new line anywhere in the .arm file.
```