

Appendix

MTASpec program manual for installation and execution

1. Installation procedure

In order to execute the MTASpec program, firstly the user has to install the following software/libraries on Linux-based machine/s. It is also recommended that, the computer node/s in cluster must be recognized by unique machine name and their respective entries shall be made in the file “/etc/hosts” on machine from which MTASpec job is being executed.

1. The C and Fortran compilers preferably gcc and gfortran respectively
2. Gaussian quantum chemistry software
3. Python (version 2.7) with development libraries
4. C and bash shell
5. Passwordless ssh for network communication

after the fulfillment of the above prerequisites, the user may start proceeding with MTASpec software installation. At the outset, the user has to copy and paste the file *MTASpec.tar* on the desired machine. It is recommended that the machine should be competent enough (in terms of memory, no. of cores, hard dish etc.) to take up the Gaussian calculation on moderate-sized molecules.

Navigate to the folder MTASpec (this folder is generated by untarring the file *MTASpec.tar*) and open the file *Config.sh* for setting up the C and FORTRAN compiler names.

```
sahunity@pc15204:~/mtaplugin> cat Config.sh
#!/bin/csh
# Run as: csh compile

set COMPILER='gfortran'

cd fragmentation
make -f Makefile.64
mv mtapplugin.so ../
cd ../

cd extra-fs

$COMPILER -o grad_format_converter.exe grad_format_converter.f
$COMPILER -o IR.exe IR.F90
$COMPILER -o RAMAN.exe RAMAN.F90

mv grad_format_converter.exe IR.exe RAMAN.exe ../
```

Then execute the command `./Config.sh` [*cs****h*** *Config.sh* or *sh* *Config.sh*] to compile the MTASpec source code.

```
sahunity@pc15204:~/mtaplugin> csh Config.sh
gfortran -O2 -fdefault-integer-8 -shared -o mtaplugin.so cardinal.o collate.o
datatypes.o dmpick.o energy.o frag.o mcweeny.o misc.o modules.o preproc.o mta.
o
IR.F90:231:31:
                WRITE(INU,1210), (FREQ(J), J=I, I+2)
                1
Warning: Legacy Extension: Comma before i/o item list at (1)
IR.F90:232:31:
                WRITE(INU,1220), (REDF(J), J=I, I+2)
                1
Warning: Legacy Extension: Comma before i/o item list at (1)
IR.F90:236:31:
                WRITE(INU,1230), F1/deno, F2/deno, F3/deno
                1
Warning: Legacy Extension: Comma before i/o item list at (1)
IR.F90:237:31:
                WRITE(INU,1240), (FINT(J), J=I, I+2)
                1
Warning: Legacy Extension: Comma before i/o item list at (1)
```

After successful compilation, it will create the file *mtaplugin.so* and return the command prompt without any error message. With this step, the compilation part of MTASpec is done. Now the user may prompt for its use.

```
sahunity@pc15204:~/mtaplugin> ll mtaplugin.so
-rwxr-xr-x 1 sahunity users 198912 Sep  6 11:49 mtaplugin.so
```

2. MTASpec execution procedure

In order to use the MTASpec, the user has to move to the directory wherein the file containing the Cartesian coordinates (in Å) (say *sample.xyz*. cf. Appendix) of the parent molecule/cluster (closed shell) in standard xyz format, is available. In the same directory create a file with name *FragParam* which contains following details.

```
sahunity@pc15204:~/jobs> cat FragParam
radcut=3.2
maxsz=36
RDEXT=1/0
addum=1/0
MINSZ=18
```

Details of FragParam

- RADCUT=3.5** //number is angstrom
This defines R-Goodness in MTA and explains for the quality of a particular fragmentation scheme, typical value recommended within 3 - 4
- MAXSZ=51** //number (stands for the maximum size for the main fragments.
In general, this can vary from systems to systems, level of theory, type of calculations [energy, IR/Raman] and hardware availability. In case of weakly bonded molecular cluster, this number is a multiple of the number atoms in a monomer.
- RDEXT=1/0** //choice for reading the external fragments
1 (true) if fragments are already available
0 (false) for a new scheme i.e. **frgonl** run
- ADDUM=1/0** //choice for adding dummy atoms to fragments
1 if a covalent bond between atoms is cut during fragments, This option adds a dummy atom, mainly hydrogen at the terminal. Applicable for extended molecules
0 if no bond is cut in fragmentation. Suitable for weakly bonded molecular clusters. E.g. water clusters, CO₂ clusters, Acetylene clusters etc.
- MINSZ=30** //number
Minimum size of the main fragments

The terms in bold are the keywords (these are case insensitive) and thus have to be present in the file as is. The number 1 represent true whereas 0 represent the false. The terms RADCUT, MAXSZ and MINSZ are relevant only in the case when RDEXT is set to 0. In the case, user have own generated fragments at their disposal, the parameter RDEXT has to be set to 1 and the file containing the fragments the MTASpec readable format must be present in same directory. Such a file must have the prefix same as the parent molecule file name, with extension “-frag.list” (say sample-frag.list *cf.* Appendix).

At this stage, the user is ready to start the calculation after linking current working directory to MTASpec source code directory. For this purpose, navigate to the folder MTASpec and open a file *gamconstants.py* available in the sub-directory *gamserver*.

```
sahunity@pc15204:~/mtaplugin/gamserver> vi gamconstants.py
sahunity@pc15204:~/mtaplugin/gamserver> pwd
/home/sahunity/mtaplugin/gamserver
```

Set the variables:

- (1) PLUGIN_ROOT by providing the absolute path of folder MTASpec [Line# 10]
- (2) OUTPUT_PREFIX by providing the absolute path of folder wherein sample.xyz file is present [Line# 11]
- (3) Set the Gaussian compiler name [Line# 12].

```
1 import os
2 import socket
3 from getpass import *
4
5 # .....
6 # Setting Paths
7 # .....
8
9 #*****
10 PLUGIN_ROOT = "/home/sahunity/mtaplugin/"
11 OUTPUT_PREFIX = "/home/sahunity/jobs/"
12 GAUSSIANCOMAND = "g16"
13 #*****
14
15 # remote gamserver root
16 REMOTE_GAMSERVER_ROOT = PLUGIN_ROOT + "gamserver"
17
18 # output prefix
19
20 CODE = 'GAUSSIAN'
21 # GAUSSIAN COMMAND
```

At last, the user has to set the path for Gaussian software by setting up the variable in the file

```
sahunity@pc15204:~/mtaplugin> cat gauss.sh
#!/bin/csh

g16root="/home/$USER/Gaussian16"
GAUSS_SCRDIR="/home/$USER/Scratch"
export g16root GAUSS_SCRDIR
. $g16root/g16/bsd/g16.profile

g16 $1 $2
```

gauss.sh
available in

MTASpec folder.

This completes the installation and linking process. To perform the calculation, user may take the following steps.

1. Move to folder where in sample.xyz file is present (directory for the OUTPUT_PREFIX in gamconstants.py)
2. Call the python script *mta.py* (driver script) available in the folder MTASpec by providing its absolute path followed by three command line parameter viz. name of xyz file, MTASpec keyword, OUTPUT file name in sequence

```
python ~/mtaplugin/mta.py sample.xyz keyword sample.out
```

3. Provide the level of theory and basis set (LB/HB) on request.
4. Enjoy a coffee break!

Keywords: FRGONL [for a new fragmentation]

MTAGRAFT [for calculating single point energy]

IRGRAFT [for vibrational IR spectrum]

RAMANGRAFT [for vibrational Raman spectrum]

The sample, input and output files are provided in the folder Sample available in the MTASpec directory. In the case of Keywords IRGRAFT and RAMANGRAFT the spectrum will be written in the file with prefix same as the parent molecule file name, with extension “-VIS.log” (say sample-VIS.log)

Isotopes in Vibrational IR and Raman Spectrum

By default, MTASpec uses the masses for the most abundant element (as defined in the Gaussian software). For example, the default masses for Hydrogen and Oxygen (to name a few) are 1.00783 and 15.99491 amu respectively. In case of isotope labelling, other than default masses, the user has to specify proper masses for atoms.

- Create a file **sample-mass** in the directory wherein the file **sample.xyz** already present (OUTPUT_PREFIX directory, normally ~/jobs/ directory).
- Sample-mass contains data in two columns. Column 1 is the atomic index in the molecule and Column 2 is the specified mass for each of the atomic indices in Column 1.

For example, for a molecule containing any number of atoms, if someone uses *Newmass_ith*, *Newmass_jth* and *Newmass_kth* for the *ith*, *jth* and *kth* atom of the molecule, then the sample-mass file is as follows:

```
sahunity@pc15204:~/jobs> ls sample*
sample-mass  sample.xyz
sahunity@pc15204:~/jobs> cat sample-mass
ith  Newmass_ith
jth  Newmass_jth
kth  Newmass_kth
```

Running MTASpec job on single node with multiple cores

By default, MTASpec uses all the processors present on the head/firing/main node. The software automatically reads the number of present processors of the node from */etc/cpuinfo* and automatically runs the calculations on the total number of processors. The mandatory requirement of this step is to machine should be able to connect itself with passwordless ssh.

Running MTASpec job on different nodes with multiple cores

- Make a file called ***node.list*** in the directory wherein sample.xyz file is present and write the name of the current node. If the names of the nodes are pc15204, pc15205 then mention this in the file *node.list*
- All the compute nodes should have same computational specification; i.e. the scratch directory, username and Gaussian software path should be same in all the nodes. Each of the nodes has ~/jobs/ folder. Also, all the nodes have connected through passwordless *ssh* for network communication.
- The number of available processors on other nodes should be equal or less number of processors than the head node. In this way, all the nodes use same number of processors as that of the main node during the calculation.
- All the available nodes must have sufficient disk space for the calculations. The equal memory availability is recommended for all the nodes.
- All the compute node names (with IP addresses) should be included to the */etc/hosts* file of the main node.
- Depending upon the number of nodes to be used for the calculations, the node names (1st line corresponds to main node name) in ~/jobs/node.list is modified.
- The gamNodeList (...) in line # 188 of ~/mtaplugin/gamserver/gamconstants.py is changed to total number of nodes.

For instance, if one MTASpec calculations are to be performed on three nodes, viz. **pc15204**, **pc15205** and **pc15206** [with pc15204 as the main node], then the ~/jobs/node.list is

```
sahunity@pc15204:~/jobs> cat node.list
pc15204
pc15205
pc15206
```

and the gamNodeList in Line #188 of ~/mtaplugin/gamserver/gamconstants.py is

```
187 # See gamnodeallocator.py for available implementations.
188 gamNodeAllocator = gamnodeallocator.GroupClusteredNodeGamNodeAllocator(gamNodeList, [1, 1, 1])
189
```


Summary for setting MTASpec Calculations

1. Requirements

- The C and Fortran compilers preferably gcc and gfortran respectively
- Gaussian quantum chemistry software
- Python (version 2.7) with development libraries
- C and bash shell
- Passwordless ssh or rsh for network communication

2. `csh ~/mtaplugin/Config.sh` or `csh ~/mtaplugin/Config.sh` generates `~/mtaplugin/mtaplugin.so`

3. `~/jobs/` contains

`sample.xyz`

`sample-mass` (in case of different masses other than the default values for specific atoms)

`FragParam`

`node.list` (containing the node names to be used for calculations, 1st line is the main node)

4. in `~/mtaplugin/gamserver/gamconstants.py`, change the followings (if required)

Line #10: `PLUGIN_ROOT`

Line #11: `OUTPUT_PREFIX`

Line #12: `GAUSSIANCOMAND`

Line #188: `gamNodeList`

See `~/mtaplugin/examples` for test fragmentation schemes that includes molecules and clusters

Appendix

Program details

will run in an interactive mode with the following broad steps.

- Step 1. Read the essential input from the user i.e. the molecular geometry, Lower and Higher Basis sets, and MTASpec Keyword
- Step 2. Calculate the energy/hessian/dipole/polarizability using Lower basis set at the indicated level of theory by doing the full calculation
- Step 3. Calculate the energy/hessian/dipole/polarizability using Lower Basis set at the indicated level of theory by doing MTA calculation
- Step 4. Calculate the difference between energy/Hessian/dipole/ polarizability computed at steps 2 and 3.
- Step 5. Calculate the energy/hessian/dipole/polarizability using Higher Basis set at the indicated level of theory by doing MTA calculation
- Step 6. Graft the difference (calculated at step 4) to energy/Hessian/dipole/polarizability computed at step 5
- Step 7. Compute IR/Raman frequencies and IR intensities/Raman activities using the grafted-Hessian matrix, dipole/polarizability.
- Step 8. Write the IR/Raman spectrum in the Gaussian output format facilitating its normal mode visualization (as well as plotting) using most of the commonly used molecular visualizer.

MTASpec package comprises of FORTRAN and Python codes for enabling QC studies on Linux environment. The fragmentation module is written in FORTRAN while the parallelization and patching modules are in python. The present version of MTASpec supports *ab initio* calculations using the Gaussian suite of software. The driver script for MTASpec is mta.py. This script has different modules for different purposes. However, the fundamental modules along with their brief functionality are presented below.

1. *Check_Integrity_of_Input*: on receipt of the input from the user, this module will do a validation check. If anything is not in format, the user is prompted for the same and the execution is aborted. Else the program will proceed further.

2. *Generate_input_files*: On the successful exit of the above module, the input file in the Gaussian supported format will be generated. The number of cores and memory available with the machine is identified and appended to the file.
3. *Perform_FC_Run*: This module, invokes the Gaussian program and does FC molecular property calculation at LB. The output is stored on a hard disk for onward use.
4. *Fragmentor*: This module breaks the parent molecule into the main fragments, followed by the generation of the overlapping fragments, as per the keywords provided by the user through the file FileParam. If the keyword RDEXT is true, the program reads the fragment data from the file. Otherwise, it creates the fragments based on the keywords RAD CUT, MAXSZ and MINSZ. Further, dummy atoms are added to all the fragments to saturate the valences of cut atom/s, if the set the parameter ADDUM is set to 1.
5. *Perform_MTA_Run*: With the completion of the module fragmentor, all the fragments (main as well as overlap) are subjected to computation (at the prescribed level of theory and LB and HB) to yield fragment molecular properties. This module has a parallel execution possibility to make the optimal use of the computational resource. All the output files are stored on the hard-disk for further usage.
6. *Patcher*: This module extracts the molecular properties from all the fragments as per the keywords viz. MTAGRAFT, ITGRAFT, RAMANGRAFT, provided by the user. Furthermore, these fragment outcomes are patched together and stored in different files by employing the MTA equations. For example, the MTA grafted energy (vide Eq. 1) is printed in the output file (file name will be provided by the user).
7. *Grafting_correction*: This module calculates the difference between the molecular properties calculated by MTA and the actual run at LB. This difference is grafted to molecular properties calculated by MTA at HB [19].
8. *Calculate_Spectrum*: This is applicable only in the case of IRGRAFT and RAMANGRAFT. On successful patching and grafting of energy, the hessian matrix, dipole/polarizability tensors, this module first diagonalizes the mass-weighted Hessian matrix and estimates the frequencies followed by calculation of IR/Raman intensities.
9. *Generate_Log_File*: At this stage, the estimated molecular properties (frequencies, IR/Raman intensities and corresponding normal modes) are written in the file supporting

the Gaussian output file format. Due to this, the file can be directly fed to the GaussView for visualization. This file ends with an extension as “VIS.log” to the input file.

INPUT Section

Sample input file elaborating case study of Bz-(H₂O)₈ cluster named as sample.xyz is provided in **Table 1**. Explanatory notes are given with a # sign. **Table 2** represents the FragParam file for automatic fragmentation. **Table 3** lists the format for providing manually generated fragments through a file sample-frag.list. *Cartesian coordinates are to be given in Å units.*

TABLE 1 MTASpec sample input file. See text for details.

| | | | |
|---|--------------|----------|----------|
| 36 # total number of atoms in molecule/clusters | | | |
| Bz-(H ₂ O) ₈ Cluster | # title card | | |
| O | 1.41660 | 3.04641 | 1.46325 |
| O | -0.77570 | -0.39457 | 1.12601 |
| O | 1.32520 | 0.93477 | -1.95422 |
| O | -2.04883 | 2.95675 | -0.50754 |
| O | 0.66617 | 3.56163 | -1.05795 |
| O | -1.41508 | 0.33122 | -1.40667 |
| O | 1.88014 | 0.36999 | 0.59478 |
| O | -1.28059 | 2.23519 | 1.95198 |
| H | 1.97419 | 3.73066 | 1.85294 |
| H | -0.89376 | -1.34441 | 1.28041 |
| H | 2.02134 | 0.67509 | -2.56981 |
| H | -2.86042 | 3.47835 | -0.51775 |
| H | 1.20879 | 3.35334 | 0.53950 |
| H | -1.07219 | -0.25096 | 0.19026 |
| H | 1.63962 | 0.66558 | -1.04608 |
| H | -1.86943 | 2.74443 | 0.44954 |
| H | 1.02537 | -0.02102 | 0.87897 |
| H | -0.35971 | 2.55692 | 1.99820 |
| H | -1.76750 | 1.23193 | -1.26680 |
| H | 0.92954 | 2.75255 | -1.53846 |
| H | -0.30847 | 3.50464 | -1.01229 |
| H | -0.53266 | 0.47908 | -1.79701 |

| | | | |
|---|----------|----------|----------|
| H | -1.18669 | 1.26394 | 1.88999 |
| H | 1.88675 | 1.24571 | 1.02836 |
| C | -0.07171 | -3.69070 | 1.22758 |
| C | -1.19006 | -3.63252 | 0.36907 |
| C | -1.06961 | -3.04427 | -0.90573 |
| C | 0.16950 | -2.52193 | -1.32406 |
| C | 1.28655 | -2.58266 | -0.46929 |
| C | 1.16621 | -3.16370 | 0.80780 |
| H | -0.16572 | -4.14583 | 2.21833 |
| H | -2.15159 | -4.04057 | 0.69530 |
| H | -1.93966 | -2.97898 | -1.56491 |
| H | 0.25657 | -2.05636 | -2.30970 |
| H | 2.24562 | -2.16289 | -0.78443 |
| H | 2.03138 | -3.19768 | 1.47616 |

TABLE 2 FragParam file contents for automatic fragmentation. See text for details.

RDEXT=0
 addum=1
 radcut=2.8
 MAXSZ=24
 MINSZ=21

TABLE 3 MTASpec sample file providing manually generated fragments through a file sample-frag.list See text for details.

| | 3 | | | # Total number of fragments |
|---|----------|----------|----------|--------------------------------|
| | 24 | | | # Number of atoms in fragment1 |
| H | 2.03138 | -3.19768 | 1.47616 | |
| H | 2.24562 | -2.16289 | -0.78443 | |
| H | 0.25657 | -2.05636 | -2.30970 | |
| H | -1.93966 | -2.97898 | -1.56491 | |
| H | -2.15159 | -4.04057 | 0.69530 | |
| H | -0.16572 | -4.14583 | 2.21833 | |
| C | 1.16621 | -3.16370 | 0.80780 | |
| C | 1.28655 | -2.58266 | -0.46929 | |
| C | 0.16950 | -2.52193 | -1.32406 | |
| C | -1.06961 | -3.04427 | -0.90573 | |
| C | -1.19006 | -3.63252 | 0.36907 | |
| C | -0.07171 | -3.69070 | 1.22758 | |
| H | 1.88675 | 1.24571 | 1.02836 | |
| H | -0.53266 | 0.47908 | -1.79701 | |
| H | -1.76750 | 1.23193 | -1.26680 | |
| H | 1.02537 | -0.02102 | 0.87897 | |

| | | | | |
|----|----------|----------|----------|---------------------------------|
| H | 1.63962 | 0.66558 | -1.04608 | |
| H | -1.07219 | -0.25096 | 0.19026 | |
| H | 2.02134 | 0.67509 | -2.56981 | |
| H | -0.89376 | -1.34441 | 1.28041 | |
| O | 1.88014 | 0.36999 | 0.59478 | |
| O | -1.41508 | 0.33122 | -1.40667 | |
| O | 1.32520 | 0.93477 | -1.95422 | |
| O | -0.77570 | -0.39457 | 1.12601 | |
| 24 | | | | # Number of atoms in fragment2 |
| H | 1.88675 | 1.24571 | 1.02836 | |
| H | -1.18669 | 1.26394 | 1.88999 | |
| H | -0.53266 | 0.47908 | -1.79701 | |
| H | -0.30847 | 3.50464 | -1.01229 | |
| H | 0.92954 | 2.75255 | -1.53846 | |
| H | -1.76750 | 1.23193 | -1.26680 | |
| H | -0.35971 | 2.55692 | 1.99820 | |
| H | 1.02537 | -0.02102 | 0.87897 | |
| H | -1.86943 | 2.74443 | 0.44954 | |
| H | 1.63962 | 0.66558 | -1.04608 | |
| H | -1.07219 | -0.25096 | 0.19026 | |
| H | 1.20879 | 3.35334 | 0.53950 | |
| H | -2.86042 | 3.47835 | -0.51775 | |
| H | 2.02134 | 0.67509 | -2.56981 | |
| H | -0.89376 | -1.34441 | 1.28041 | |
| H | 1.97419 | 3.73066 | 1.85294 | |
| O | -1.28059 | 2.23519 | 1.95198 | |
| O | 1.88014 | 0.36999 | 0.59478 | |
| O | -1.41508 | 0.33122 | -1.40667 | |
| O | 0.66617 | 3.56163 | -1.05795 | |
| O | -2.04883 | 2.95675 | -0.50754 | |
| O | 1.32520 | 0.93477 | -1.95422 | |
| O | -0.77570 | -0.39457 | 1.12601 | |
| O | 1.41660 | 3.04641 | 1.46325 | |
| 24 | | | | # Number of atoms in fragment 3 |
| H | 2.03138 | -3.19768 | 1.47616 | |
| H | 2.24562 | -2.16289 | -0.78443 | |
| H | 0.25657 | -2.05636 | -2.30970 | |
| H | -1.93966 | -2.97898 | -1.56491 | |
| H | -2.15159 | -4.04057 | 0.69530 | |
| H | -0.16572 | -4.14583 | 2.21833 | |
| C | 1.16621 | -3.16370 | 0.80780 | |
| C | 1.28655 | -2.58266 | -0.46929 | |
| C | 0.16950 | -2.52193 | -1.32406 | |
| C | -1.06961 | -3.04427 | -0.90573 | |
| C | -1.19006 | -3.63252 | 0.36907 | |
| C | -0.07171 | -3.69070 | 1.22758 | |

| | | | |
|---|----------|---------|----------|
| H | -1.18669 | 1.26394 | 1.88999 |
| H | -0.30847 | 3.50464 | -1.01229 |
| H | 0.92954 | 2.75255 | -1.53846 |
| H | -0.35971 | 2.55692 | 1.99820 |
| H | -1.86943 | 2.74443 | 0.44954 |
| H | 1.20879 | 3.35334 | 0.53950 |
| H | -2.86042 | 3.47835 | -0.51775 |
| H | 1.97419 | 3.73066 | 1.85294 |
| O | -1.28059 | 2.23519 | 1.95198 |
| O | 0.66617 | 3.56163 | -1.05795 |
| O | -2.04883 | 2.95675 | -0.50754 |
| O | 1.41660 | 3.04641 | 1.46325 |