Set Up Guide

*Using Gaussian on your personal machine*

A windows version of Gaussian is already installed on all machines in the office

There is a graphical user interface called GaussView installed too

One can create molecules by hand in GaussView and then perform various operations on them, such as energy evaluations, geometry optimizations, etc.

Note, a geometry optimization will get you to a local minimum on the potential energy surface, not necessarily the global minimum

You are limited to the size of the machine (usually 4 cores) and can only run tasks in serial – computationally inefficient

But useful for small quick jobs like single geometry optimization of one conformer of a small molecule

*Using Gaussian on the CSF*

Linux version of Gaussian – accessed via the command line once logged into the CSF

Can use any command line environment to log in (Cygwin, ssh secure file transfer client, putty, etc)

You can run on many cores and tasks can be run in parallel – much more efficient

Useful for multiple jobs, eg geometry optimization of 100 conformers of a small molecule

How to setup:

Create a directory in your scratch folder on CSF

Need an input file (XXX.gjf) and GaussSub.py (get this of someone in the office)

Run GaussSub.py (command line: python GaussSub.py)

Choose a number of cores (must match those specified in input file)

This creates the submission script GaussSub.sh

To run this : qsub GaussSub.sh

To check job status : qstat

Running jobs will have a lower case ‘r’

Queued jobs will have a lower case ‘qw’

Sample input file format:

%nprocshared=4

# opt freq b3lyp/6-31g out=wfn

Title card here

0 1

O 1.234235 1.234345 2.237327

H 1.675832 1.098764 2.464646

H 1.095947 1.958574 2.090909

Water.wfn

%nprocshared - number of cores you want to use

# - everything after this is just keywords in an arbitrary order

Opt = do a geometry optimization

Freq = do a frequency analysis

B3lyp/6-31g = theory/basis set

Out=wfn = write a wfn file as output (must then specify the name of this at bottom of the input file)

Can have loads of other keywords – see Gaussian website

0 1 - Molecular charge and multiplicity

XYZ coordinates

ADD ‘ B3LYP\n’ TO END OF SECOND LINE OF ALL .wfn FILES

*Using AIMAll on the CSF*

Again, Linux version, takes .wfn files from Gaussian as input

Just need to put a copy of AIMAll in your home directory and give permission to all files in each folder (chmod 777 \*)

Put the wfn file of interest in a folder in scratch

Put AIMAllSub.py, InpAIMAllSub1.py, InpAIMAllSub2.py in there too (get these off someone in office)

AIMAllSub.py - Step one of AIMAll: creating input files from wfn info

Run this (python AIMAllSub.py), specifying number of cores

Run resulting AIMAllSub.sh to produce .inp files

InpAIMAllSub1.py - Step 2 of AIMAll: Creating individual atomic output files (.int)

Run this (python InpAIMAllSub1.py), specifying number of cores which must match those above

Run resulting InpAIMAllSub1.sh to produce atomic .int files (eg h34.int)

InpAIMAllSub2.py - Step 3 of AIMAll: Only if pairwise terms required

Run this (python InpAIMAllSub2.py), specifying number of cores which must match those above

Run resulting InpAIMAllSub2.sh to produce diatomic .int files (eg h34\_o35.int)