SUPERCOMPUTING WALES
UWCHGYFRIFIADURA CYMRU

Guide to Running ***GAMESS-US***

What is GAMESS-US?

GAMESS (US) is a program for ab initio molecular quantum chemistry. Briefly, GAMESS can compute SCF wavefunctions ranging from RHF, ROHF, UHF, GVB, and MCSCF. Correlation corrections to these SCF wavefunctions include configuration interaction, second order perturbation theory and coupled-cluster approaches, as well as the density functional theory approximation. Excited states can be computed by CI, EOM, or TD-DFT procedures. Nuclear gradients are available for automatic geometry optimisation, transition state searches, or reaction path following.

Computation of the energy Hessian permits prediction of vibrational frequencies, with IR or Raman intensities. Solvent effects may be modelled by the discrete effective fragment potentials, or continuum models such as the Polarizable Continuum Model. Numerous relativistic computations are available, including infinite order two component scalar relativity corrections, with various spin-orbit coupling options. The Fragment Molecular Orbital method permits use of many of these sophisticated treatments to be used on very large systems, by dividing the computation into small fragments. Nuclear wavefunctions can also be computed, in VSCF, or with explicit treatment of nuclear orbitals by the NEO code.

Step 1 - Log in

The examples used in this guide are configured to run on the Cardiff Skylake *Hawk* cluster. Connect to *hawklogin.cf.ac.uk* with your Supercomputing user credentials using your preferred method (e.g. PuTTY from a Windows machine or ssh from any Linux terminal, thus

```
ssh –l your_username hawklogin.cf.ac.uk
```

The steps below involve typing commands (**`in bold font`**) in the terminal window.

Step 2 - Load a GAMESS-US module

In common with most of the application guides in this series, we assume at the outset that the module of choice would be selected from those originally available on HPC Wales. Thus, a necessary first step would be to gain access to that module set by issuing the commands

```
module purge
module load hpcw
```

Note that the “module purge” command is required to prevent any unintentional collision with pre-existing modules. Subsequently issuing the module command

```
module avail
```

will provide visibility of and access to the entire set of HPCW modules

A number of GAMESS-US binary packages are available. Note that in common with most other software packages on the system, these are built with the Intel compiler.

- List pre-installed GAMESS_US versions:

```
module avail gamess
```

- Load the preferred version (20131206):

  ```
  module load gamess
  ```

- Confirm the loaded modules. Note that this is currently the latest version of the code. All dependencies are handled automatically via the module file:

  ```
  module list
  ```

Step 3 - Create a directory

From your home directory, create a directory to hold the GAMESS-US data:

```
cd ~
mkdir GAMESS
```

Step 4 - Obtain a test case

A number of test cases are provided with the installation at:

```
/app/chemistry/gamess/20131206/sb/intel-13.0/intel-4.1/example
```

This directory contains the required GAMESS input files, as well as SLURM job scripts. Copy the input file and job script to your user space, thus:

```
cd ~/GAMESS
cp /app/chemistry/gamess/20131206/sb/intel-13.0/intel-4.1/example/exam*.inp .
cp /app/chemistry/gamess/20131206/sb/intel-13.0/intel-4.1/example/gamessexam*.SLURM.q .
```

This 80-core example submitted using the SLURM job script

```
gamessexample.ppn=40.SLURM.q
```

performs an MP2 optimisation on the molecular species $C_2H_2S_2$ in a CCQ basis set.

The other two simple test cases that will have been copied to your user space feature input files `exam08.inp` *(*$^1A_1$ *H_2O RHF + MP2 gradient calculation) and* `exam15.inp` *(C_2 diatom, in the electronic state $^2\Pi_u$).*

Step 5 - Submit a parallel job

You are now ready to run these test cases with the supplied job scripts. Consider the largest case that uses the job script *gamessexample.ppn=40.SLURM.q*

- From your working directory, submit the job using:

  ```
  sbatch gamessexample.ppn=40.SLURM.q
  ```

- Check the job queue using: `squeue`
- Note that all the examples are executed under control of the `rungms` script,

  ```
  /app/chemistry/gamess/20131206/sb/intel-13.0/intel-4.1/bin/rungms
  ```

with four parameters to be specified, as shown in the example. Note also that the majority of the files created during the job are routed to the user's scratch directory, */scratch/$USER/GAMESS.<JobID>*, and may be deleted on job termination.

- The first thing to note is that this 80-core job will **fail in execution**, with the following error message captured in the log file - *example.Hawk.out.80.ppn=40.<JobID>* that is routed to the GAMESS directory.

```
DIRECT 4-INDEX TRANSFORMATION
 DDI Process 51: semop return an error performing 1 operation(s) on semid -
1882841088.
 semop errno=EINVAL.
 DDI Process 51: Semop error in DDI_Fence_check (acq); handle=2.
 DDI Process 15: semop return an error performing 1 operation(s) on semid
868622336.
 semop errno=EINVAL.
```

- This error arises from the DDI (Distributed Data Interface) used by GAMESS being unable to operate correctly when faced with fully populated 40-core nodes. While we expect to correct this error in due course, the job may be run successfully by only using 32 cores on the nodes.
- Thus the second script - *gamessexample.SLURM.q* – that uses 64-cores with two nodes and 32 cores per node through the SLURM directives

```
#SBATCH --ntasks=64                # number of parallel processes (tasks)
#SBATCH --ntasks-per-node=32       # tasks to run per node
```

- will contain all the output and the message "EXECUTION OF GAMESS TERMINATED NORMALLY". Inspect the log files *example.Hawk.out.64.ppn=32.<JobID>*, for more information. The 64-core job should complete in ca. four minutes.
- Compare your job output with the reference output file:

/app/chemistry/gamess/20131206/sb/intel-13.0/intel-4.1/example/example.Hawk.out.64.ppn=32.15504

Step 6 - More Test Cases

The other two simple test cases copied to your user space feature input files *exam08.inp* (*$^1A_1$ H_2O RHF + MP2 gradient calculation) and exam15.inp (C_2 diatom, in the electronic state $^2\Pi_u$).* Both use 32 cores and may be submitted using the scripts *gamessexam08.SLURM.q* and *gamessexam15.SLURM.q* respectively.

More test cases are distributed with GAMESS. They can be found at

`/app/chemistry/gamess/20131206/sb/intel-13.0/intel-4.1/tests/standard`

and http://www.msg.ameslab.gov/gamess/GAMESS_Manual/tests.pdf

Adapt the job script above to run these cases.

To create a new case, refer to the section on Input Examples of the GAMESS Manual, which is widely available online: http://www.msg.ameslab.gov/gamess/GAMESS_Manual/tests.pdf

A PDF version of all sections of the documentation is also available at: http://www.msg.ameslab.gov/gamess/documentation.html

References

- Official GAMESS website: http://www.msg.ameslab.gov/gamess/
- GAMESS Update Paper: http://www.msg.ameslab.gov/advances/advances.html
- “General atomic and molecular electronic structure system”, Schmidt, M.W., Baldridge, K.K., Boatz, J.A., Elbert, S.T., Gordon, M.S., Jensen, J., Koseki, S., Matsunaga, N., Nguyen, K. A., Su, S., Windus, T.L., Dupuis, M., Montgomery, J.A.:J. Comput. Chem. 14, 1347-1363 (1993).
- GAMESS User Documentation: http://www.msg.ameslab.gov/gamess/documentation.html