



Guide to Running GAMESS-UK II. MPI/ScaLAPACK

What is GAMESS-UK?

The General Atomic and Molecular Electronic Structure System (GAMESS) is a general purpose *ab initio* quantum chemistry package. The original code split in 1981 into GAMESS (US) and GAMESS (UK) variants, which now differ significantly. Many of the early developments in the UK version arose from the earlier UK based ATMOL program which, unlike GAMESS, lacked analytical gradients for geometry optimisation.

GAMESS (UK) can perform a number of general computational chemistry calculations, including Hartree–Fock, Møller–Plesset perturbation theory (MP2 & MP3), coupled cluster (CCSD & CCSD(T)), density functional theory (DFT), configuration interaction (CI), and other advanced electronic structure methods. Calculation of valence bond wave functions is possible by the TURTLE code, due to J. H. van Lenthe.

Note there are two versions of the parallel code, one based on the Global Array (GA) tools from Pacific Northwest National Laboratory, and the second based on MPI and ScaLAPACK. The current note is restricted to a consideration of the MPI/ScaLAPACK code that focuses on the performance of efficient, large-scale DFT calculations.

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-UK 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-UK binary packages may be available. Note that in common with most other software packages on the system, these are built with the Intel compiler.

- List pre-installed GAMESS_UK versions:

```
module avail gamess-uk
```

- Load the preferred MPI version (8.0) and Intel compiler:

```
module load compiler/intel/15.0/1.133
module load gamess-uk/8.0-MPI
```
- 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-UK data:

```
cd ~
mkdir GAMESS-UK-mpi
```

Step 4 - Obtain a test case

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

```
/app/chemistry/gamess-uk/8.0-MPI/sb/intel-15.0/intel-4.1/examples
```

This directory contains the required GAMESS-UK input files, as well as job scripts. Copy the two job scripts to your user space:

```
cd ~/GAMESS-UK-mpi
cp /app/chemistry/gamess-uk/8.0-MPI/sb/intel-15.0/intel-4.1/examples/GAMESS-UK.mpi.example*.SLURM.q .
```

Note that the job scripts point to the location of the required input files. There are two test cases:

- *GAMESS-UK.mpi.example1.SLURM.q* (Density Functional Theory (DFT) B3LYP calculation of cyclosporin in a 6-31G** basis – 1855 GTOs),
- *GAMESS-UK.mpi.example2.SLURM.q* (DFT B3LYP calculation of the SiOSi₇ zeolite fragment in a DZVP (O and Si) and DZVP2 (H) basis – 3975 GTOs).

The test case used in this guide is *GAMESS-UK.mpi.example1.SLURM.q*.

Step 5 - Submit a parallel job

You are now ready to run this test case with the supplied job script *GAMESS-UK.mpi.example1.SLURM.q*

- From your working directory, submit the job using:

```
sbatch GAMESS-UK.mpi.example1.SLURM.q
```
- Check the job queue using: `squeue`

When execution commences, a file called

```
DFT.cyclo.6-31G-dp.Hawk.n120.PPN=40.out.<Job_ID>
```

is created in the *GAMESS-UK-mpi* directory that holds the job output, where *<Job_ID>* is the ID generated by the system. Many temporary files are generated and

routed to the user's scratch directory, `/scratch/$USER/GAMESS-UK.<Job_ID>`, created by the job.

- The job should take around 2.5 minutes using 120 cores if the case runs successfully. The file `DFT.cyclo.6-31G-dp.Hawk.n120.PPN=40.out.<Job_ID>` should contain all the output and point to successful completion of the job.
- Compare your job output with the reference output file

`/app/chemistry/gamess-uk/8.0-MPI/sb/intel-15.0/intel-4.1/examples/DFT.cyclo.6-31G-dp.Hawk.n120.PPN=40.out.14831`

Step 6 - More test cases

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

`/app/chemistry/gamess-uk/8.0-MPI/sb/intel-15.0/intel-4.1/examples/parallel_MPI/input_files_benchmarks`

Run the 2nd example above – `GAMESS-UK.mpi.example2.SLURM.q`, and compare your job output with the reference output file

`/app/chemistry/gamess-uk/8.0-MPI/sb/intel-15.0/intel-4.1/examples/DFT.siosi7.3975.Hawk.n120.PPN=40.out.14833.`

The job should take around 20 minutes using 120 cores if the case runs successfully.

Adapt the job script above to run these other cases.

To create a new case, refer to Chapters 3 and 4 of the GAMESS-UK Manual, which is widely available online

<http://www.cfs.dl.ac.uk/docs/>

PDF versions of all sections of the manual are also available at the URL above.

References

- Official GAMESS-UK website: <http://www.cfs.dl.ac.uk/>
- "The GAMESS-UK electronic structure package: algorithms, developments and applications", M.F. Guest, I. J. Bush, H.J.J. van Dam, P. Sherwood, J.M.H. Thomas, J.H. van Lenthe, R.W.A Havenith, J. Kendrick, *Molecular Physics*, Vol. 103, No. 6-8, 20 March-20 April 2005, 719-747, doi:[10.1080/00268970512331340592](https://doi.org/10.1080/00268970512331340592).
- GAMESS-UK User Documentation: <http://www.cfs.dl.ac.uk/docs/index.shtml>