

What is SWAN

SWAN is a wave model to compute random, short-crested wind-generated waves in coastal regions and inland waters. This user guide provides instructions on how to run a standard SWAN test case on the HPC Wales systems.

Step 1 - Log in

The example used in this guide is configured to run on the Swansea Sandy Bridge cluster. Connect to *login.hpcwales.co.uk* with your HPC Wales user credentials using your preferred method (e.g. PuTTY from a Windows machine or ssh from any Linux terminal), then **ssh sw-sb-log-001** to connect to the Swansea system.

The steps below involve typing commands in the terminal window.

Step 2 - Load a SWAN module

A number of SWAN binary packages are available.

- List pre-installed SWAN versions:

```
module avail SWAN
```

Note this command also returns SWAN-unstructured modules, which will be discussed in step 6.

- Load your preferred version (41.01 is used in this tutorial):

```
module load SWAN/41.01
```

- Confirm the loaded modules (note the dependent compiler, MPI, and I/O library are loaded automatically):

```
module list
```

Step 3 - Create a directory

From your home directory, create a directory to hold SWAN user files:

```
cd ~
```

```
mkdir -p SWAN
```

Step 4 - Obtain a test case

SWAN has a number of predefined test cases. For this example, we are using the academic test case 'a11refr', a copy of which is at:

```
/app/environment/SWAN/41.01/sb/intel-13.0/intel-4.1/example
```

Now copy files to your local directory by:

```
cd ~/SWAN
```

```
cp -rip /app/environment/SWAN/41.01/sb/intel-13.0/intel-4.1/example/* .
```

Step 5 - Submit a job

Now the current directory should contain all required files to run the SWAN 'a11refr' test case. Inspect the jobscript *SWAN.SLURM.q* – parallel SWAN jobs are launched using the *swanrun* script supplied by the SWAN installation (rather than using *mpirun* directly). Note the use of *\$SLURM_NTASKS* and *srun*, the former an environmental variable set by the SLURM job scheduling software. This helps set the number of cores used by the job while *srun* helps generate the *machinefile* containing the list of computing nodes running the job, both required by *swanrun* to launch the job.

- Submit the job using: `sbatch SWAN.SLURM.q`
- Check the job queue using: `squeue`
- When completed, screen output can be found in a file called `SWAN.o.<JOBID>` (where `<JOBID>` is the unique job ID generated by the queuing software). A number of other files should be created containing the results of the run.
- For more information, check the SWAN user manual in the reference section below.

Step 6 - Using SWAN-unstructured

SWAN can better represent real-world complex geometries when using unstructured grid. These variants are provided as separate SWAN-unstructured modules on the HPC Wales systems. Note that SWAN-unstructured is not available by default for all users. Please contact the helpdesk to ask about getting access to the software.

Once you have been registered to use SWAN-unstructured please follow the steps here to run an unstructured test case called F32:

- Repeat step 2, but load a SWAN-unstructured module instead:

```
module purge
```

```
module load SWAN-unstructured/41.01
```

```
module list
```

- Create a working directory and copy the F32 test case:

```
cd ~
```

```
mkdir -p SWAN-unstructured
```

```
cp -r /app/environment/SWAN-unstructured/41.01/sb/intel-13.0/intel-4.1/example/* .
```

- Repeat step 5 to run the case. This time the test case is serial.

- This job will take a couple of minutes to complete. Examine the output to confirm everything is OK.
- For more information, check the unstructured SWAN tutorial in the reference section below.

References

- SWAN website: <http://swanmodel.sourceforge.net/>
- SWAN user manual:
http://swanmodel.sourceforge.net/online_doc/swanuse/swanuse.html
- Unstructured SWAN, a tutorial:
<http://swanmodel.sourceforge.net/unswan/unswan.htm>