LETKF computer practical with the BV model

PA Browne
Email: p.browne@reading.ac.uk
Department of Meteorology, University of Reading,
RG6 6BB, UK

14 March 2016

0 Getting started

Log into windows (somehow).

Once you are finally in - load an xterminal window from the following menus:

Start → All Programs → MobaXterm Personal Edition → MobaXterm Personal Edition

Click the button "Start local terminal" in the middle of the screen.

Log into ARCHER. Replace `USERNAME` below with the one you were given on a separate piece of paper.

```
:~> ssh -X USERNAME@login.archer.ac.uk
```

Enter your password. If a prompt box opens up, you can ignore it if you wish.

Move to your /work directory.

```
:~> cd /work/n02/n02/$USER
```

Copy the files you will need for the course.

```
:~> cp /work/n02/n02/pbrowne/2016_dacourse.tar .
```

Unpack the files into your /work directory

```
:~> tar -xvf 2016_dacourse.tar
```

Setup some modules for ARCHER (NOTE: ONLY EVER NEED TO DO THIS ONCE):

```
:~> cat 2016_dacourse/bashrc >> ~/.bashrc && . ~/.bashrc
```

1 Barotropic vorticity model

Now play with the barotropic vorticity model. This is a much bigger model and requires it to be sent to the queue on ARCHER to be processed.

Move to the appropriate directory:

```
:~> cd 2016_dacourse
```

1.1 Truth run

Now we must run *one* instance of the model to act as the truth.

Generate a submission script for ARCHER for this by running the command

```
:~> ensemble obs
```

This will generate a submission file `pbs_jobscript`. Submit this to the queue on ARCHER using the command

```
:~> qsub -q course1 pbs_jobscript
```

You can watch the status of the queue with the command

```
:~> qstat
```

more specifically, just your own jobs in the queue can be shown with the command

```
:~> qstat -u $USER
```

Note the letter in the penultimate column. Q means queuing, R means running, E means ending. When nothing appears as a result of running qstat, this means the job has left the queue. Hopefully it was successful.

When the job has finished, we can now look at the results.

There are two scripts that can let you look at the output: `python_field.py` and `python_trajectories.py`

Look at the trajectories of the truth with the command

```
:~> python python_trajectories.py
```

If you wish, save the image produced using the interactive buttons.

Now look at the full fields and observations. The command to use here is

```
:~> python python_field.py M
```

Where M is the observation timestep, i.e. $0, 1, 2, \dots, 24$. M=0 corresponds to the initial ensemble where we have no observation.

Again, if you wish, you can save these plots.

1.2 Stochastic ensemble

It is useful to have a stochastic control run with which to compare the assimilation results.

Generate a submission script for ARCHER for this by running the command

```
:~> ensemble stoch N
```

Where N is the number of ensemble members to run. I suggest using 24 or 48 ensemble members.

As before, submit this to the queue one ARCHER with the command

```
:~> qsub -q course1 pbs_jobscript
```

As before, you may use the scripts `python_field.py` and `python_trajectories.py` to look at the output.

We also have a further output diagnostic tool: `python_rmse.py`.

Run this with

```
:~> python python_rmse.py
```

Try and understand the output.

1.3 Assimilation with the LETKF

Generate a submission script for ARCHER for this by running the command

```
:~> ensemble letkf N
```

Where N is the number of ensemble members to run.

Now with the LETKF there are 2 parameters to choose: the inflation factor and the localisation radius.

These are defined for the codes in `empire.nml`.

Inspect this file, and when you are happy with the input parameters, submit the job to the queue. Once the job has finished, try and understand the outputs. Use the tools you used before to look at the outputs.

Experiment with changing the inflation `rho` and the localisation length scale `len`, as well as the number of ensemble members.