

Compiling the dynamical core on NCAR's bluefire

This is possible due to the work of Seok-Woo Son. I've following what he did to get the code running (and modifying it to work on later machines). If you started by copying my directory, steps 1 through 3 are already completed, and you can go right to the end. (An exception may be the udunits-1.12.4 directories, which you will need to copy as well to compile the model.)

1) I started from GFDL's memphis release tarfile, which is available here:
`/blhome/gerber/models/code/memphis/backup_source_code/atm_dycores_patch1.tar.gz`

Once you unzip and un-tar this, you'll find everything within the directory `atm_dycores/` I'll refer to things relative to this point.

2) Update the `mkmf.template` file. Template files are located in

`atm_dycores/bin/`

As bluefire is an IBM machine, I used `mkmf.template.ibm`, but modified it slightly to work at NCAR. (You can see the original `.ibm` template in this directory as well.) My modified version is located here; you can copy this script into your directory to get started.

`/blhome/gerber/models/code/memphis/atm_dycores/bin/mkmf.template.ibm`

If you take a look at this script, you'll see a reference to something in my own directory. I apologize that this is not a very elegant way to work with the model. We need these libraries for the netcdf files, and I naively installed them locally. There may be more up-to-date version, but these work:

`/blhome/gerber/epg_library/libraries/udunits-1.12.4`

It might be safest to copy this directory of libraries to a place on your home system, and then switch the reference in your `mkmf.template.ibm` to where you put the files.

3) To get it to compile you need to make a couple key changes to the fortran files.

`atm_dycores/src/shared/mpp/mpp_pset.F90`

Here, you need to add one line: `sequence`. See the difference between `mpp_pset.F90` and `mpp_pset.F90.original` in
`/blhome/gerber/models/code/memphis/atm_dycores/src/shared/mpp/`

You must also change:
`atm_dycores/src/shared/memutils/memutils.F90`

Here the change is moving an `ifdef` statement back a few lines, so that it

comes after all the variable definitions. (I assume this was the problem.)

Again, compare it with the .original file in

```
/blhome/gerber/models/code/memphis/atm_dycores/src/shared/memutils/
```

4) Next, get the compile scripts set for NCAR. I split it up into two, one to compile the mppcombine routines, and one to compile the code. They can be run locally -- that is, don't need to be submitted as jobs.

```
/blhome/gerber/models/code/memphis/atm_dycores/exp/spectral/ncar_dycore_compile_script
```

```
/blhome/gerber/models/code/memphis/atm_dycores/exp/spectral/ncar_mppcombine_compile_script
```

The scripts should be run from your directory:

```
atm_dycores/exp/spectral/
```

The latter script gets you the mppncombine utility, which combines the output files from the various processors into one file at the end. You only need to compile this once, unless the computer system gets upgraded significantly.

When you want to compile the model, you run the first script. Unfortunately it won't work all the way. The problem is that something to do with netcdf.inc. The paths that I'm giving it in the mkmf.template.ibm seem to be necessary, but also include things the model doesn't need, and the model is not compiling properly. (I had a similar problem on the nyu system, but was able to fix it by specifying only the libraries I need. I don't know the ncar system as well. The problem may be that it's trying to compile additional netcdf related files. The errors I got more recently seem to be related to mpi issues, too.)

This was my earlier explanation (but note that as of December, 2010, I seem to have other errors with this script). As far as I can tell, the code does NOT need to compile netcdf.f90, but it somehow got brought into our "Makefile" by accident. This file is the actual instructions that are used to compile the code, located here:

```
/blhome/gerber/memphis/atm_dycores/exp/spectral/exec.ibm/Makefile
```

I looked carefully for the file, and netcdf.f90 would be compiled to make netcdf.o, but this .o file is never used. Another hint that this is the problem can be found in the old makefile that worked on bluevista:

```
/blhome/gerber/memphis/atm_dycores/exp/spectral/exec.ibm.bluevista/Makefile
```

I found no reference to netcdf.o in this file. Somehow it got picked up on bluefire, though. Perhaps the problem is that /usr/local/include/ on bluefire has a file: netcdf.f90, and the code automatically included it in the Makefile.

The solution that I have is to eliminate the offending lines from the Makefile, and then run it again. I'm sure there's a more elegant solution, but I really don't understand the Makefile commands very well. At any rate, I've made one Makefile that works:

```
/blhome/gerber/models/code/memphis/atm_dycores/exp/spectral/Makefile_fixed
```

Move this file to the executable directory,

```
atm_dycores/exp/spectral/exec.ibm/
```

and from this directory, execute:

```
make -f Makefile_fixed
```

The code should now compile all the way.