

MEA 712: (An Introduction to) Mesoscale Atmospheric Modeling
Third mini computing assignment

Due at the start of the next class (Tuesday 1 September)

The next mini assignment is to use GrADS to make a plot of the gridded field from your existing FORTRAN code. **Note:** at this point, if you are very confident in your ability to take output from a FORTRAN code and make plots of it with some other piece of software, you are welcome to do so. However, *Microsoft Excel is not an acceptable solution, as you will be unable to make more sophisticated plots later on.* The software you choose needs to be able to perform things like animations, and to plot things like 2D fields and wind vectors. This exercise will get you started by showing you what you need to do to use GrADS. The handout is long because it is partly tutorial, but the procedure is actually pretty simple.

A. **If** you have some favorite piece of software that already does the above, and that you know you can use to display FORTRAN output:

1. Add write statements to your FORTRAN code and then read the data into your preferred piece of software.
2. Bring a print out of your code, and of the plot of Ψ that you make with your software.

B. **If** you do not fall under heading A, then follow this procedure instead.

1. The following are modifications you will need to make to your code.
 - a. Declare the variable IREC as an integer.
 - b. Add the following statement to open the output file. These should be the first lines below the declarations in your code. Note that there is a line continuation shown here in standard FORTRAN-90 form.

```
OPEN(10,FILE='output.gds',FORM='UNFORMATTED',ACCESS='DIRECT', &  
      RECL=4*NX)  
IREC=1
```

As a sidenote, the RECL record corresponds to the number of bytes needed to write out one horizontal level of data. In standard FORTRAN, there are 4 bytes per data point (this won't change). For our simple 1D model, we therefore use RECL=4*NX. If we were writing a 2D horizontal model, we would use RECL=4*NX*NY instead (where we would need to define NY as the number of y points).

(continued)

- c. Here is a **general** formulation for writing out 3D data in GrADS format (where J , K , and N are the y , z , and t indices; and, NZ and NT are the numbers of z and t points). In this example, `VARONE` and `VARTWO` are just the names of two different variables in whatever code we happen to be running.

```
DO N=1,NT
  DO K=1,NZ
    WRITE(10,REC=IREC) ((VARONE(I,J,K,N),I=1,NX),J=1,NY)
    IREC=IREC+1
  ENDDO
  DO K=1,NZ
    WRITE(10,REC=IREC) ((VARTWO(I,J,K,N),I=1,NX),J=1,NY)
    IREC=IREC+1
  ENDDO
ENDDO
```

This would write out both variables for all times and at all levels. The tricks to getting this to work properly are as follows:

- i. Write out all data for each time before moving on to the next time.
- ii. Write out all data for each variable before moving on to the next variable.
- iii. Write out all of a variable's horizontal data on one level before moving on to the next vertical level. The looping over I and J is accomplished as shown here, and the nesting needs to be in the order shown (I loop inside of the J loop).
- iv. Each horizontal level constitutes one "record". The variable `IREC` is our place-keeper (it keeps track of what record we're writing), so it need to be incremented by 1 after each level is written.

Use the above for future reference. **In our current 1D simple model**, we have only one time, only one vertical level, only one point in the y direction, and only one variable (`PSI`). So, for now all of the above simplifies down to:

```
WRITE(10,REC=IREC) (PSI(I),I=1,NX)
```

which belongs somewhere below the loop in which `PSI` is defined in the first place.

- d. End the executable part of your code (just above your `STOP` and `END` statements) with the command `CLOSE(10)` to close the output file.
2. Compile and run your code. You should now find a file called `output.gds` in your directory.
3. Download the GrADS data descriptor file (.ctl file) that I have made available at:

<http://www.meas.ncsu.edu/mdparker/courses/mea712/output.ctl>

 and put it in the same directory as `output.gds`. This .ctl file may look quite mysterious at first, but there is a logic to it. See the tutorial example on the accompanying handout (also available on the course web page).
4. You will need to have access to a working copy of GrADS. You have a few choices, here.

Instructions for downloading GrADS to your local machine

If you know what you are doing, then you can go to the GrADS homepage and download and install the full pre-compiled distribution from:

<http://www.iges.org/grads/downloads.html>

Once you have GrADS installed and working, proceed to step 5. It's actually pretty simple, and the instructions are generally good. However, if you're not sure how to do the download and install, the following alternative procedures should work.

(continued)

Instructions for working on grendel¹

login to the grendel machine using `local-prompt> ssh -X yourname@grendel.ece.ncsu.edu` and using your unity password.²

```
grendel% setenv GADDIR /afs/eos.ncsu.edu/courses/mea/mea712/common/grads/dat
```

```
grendel% setenv PATH /afs/eos.ncsu.edu/courses/mea/mea712/common/grads/bin:$PATH
```

Now, if you type `grendel% which gradsc` you should receive the reply:

```
/afs/eos.ncsu.edu/courses/mea/mea712/common/grads/bin/gradsc
```

This indicates that you now have GrADS in your path, and can proceed to step 5.

Note: you will need to perform these `setenv` commands every time you log in to grendel. If this is a pain and you know what you're doing, you can add the commands to the `.mycshrc` file in your home directory on grendel.

Instructions for working on bigdog³

login to the bigdog machine using `local-prompt> ssh -X yourname@bigdog.pams.ncsu.edu` and using your unity password.

```
[username@login-1]$ (enter the following as one command, with only a space at each line-end)
```

```
scp -r
```

```
yourname@grendel.ece.ncsu.edu:/afs/eos.ncsu.edu/courses/mea/mea712/common/forbd  
$HOME/grads
```

```
[username@login-1]$ setenv GADDIR $HOME/grads/dat
```

```
[username@login-1]$ setenv PATH /$HOME/grads/bin:$PATH
```

Now, if you type `[username@login-1]$ which gradsc` you should receive the reply:

```
/home/yourname/grads/bin/gradsc
```

This indicates that you now have GrADS in your path, and can proceed to step 5.

Note: you will need to perform these `setenv` commands (but *not* the `scp` command) every time you log in to bigdog. If this is a pain and you know what you're doing, you can add the commands to the `.cshrc` file in your home directory on bigdog.

(continued)

¹For all practical purposes, "grendel" is the CPU guts behind what is commonly called "unity". The main problem with relying on grendel is that most people have a pretty small disk quota on it. Therefore, you may not be able to make larger output files on grendel. But, it will enable you to at least get your feet wet as we go along.

²When logging on or transferring files to/from grendel, I find that maybe 1 time in 10 I fail the password challenge even when I am *sure* I typed it correctly. This appears to be a bug. If you encounter it, simply wait a moment and try again.

³There are a few more steps involved than for grendel, but bigdog is probably the better long-term option because most people have larger disk space and bigdog tends to be a faster machine.

5. Enter GrADS and make a plot. There is a great deal of information on GrADS commands available via the GrADS documentation website, at:

<http://www.iges.org/grads/gadoc/>

but for now I will walk you through the basic procedure of making a simple 1D plot. **In order for this to work** you need to have completed step 4 and you need to be in the directory that contains both the output.gds and output.ctl files. In other words, you either need to run your SIMPLEMOD code on grendel or bigdog, or you need to copy your files over to grendel/bigdog from wherever you created them.

```
grendel/bigdog-prompt> gradsc
      (hit return for Landscape mode)
ga-> open output.ctl
ga-> display psi
ga-> enable print psiplot
ga-> print
ga-> disable
ga-> quit
grendel/bigdog-prompt> gxeps psiplot
```

This final command will make a file psiplot.eps, which you can then print on any Postscript printer. If you don't know what to do with Postscript files, then instead try producing a .gif file using:

```
grendel/bigdog-prompt> gxgif psiplot
```

6. Bring a print out of your code, and of the plot of PSI that you make in GrADS.