A&OS C115/C228 – How to use DTDM and GrADS

Spring, 2007 – Fovell

The Dynamics and Thermodynamics Demonstration Model (DTDM) is a very simple, two-dimensional model to facilitate understanding of some fundamental phenomena. The DTDM package is script-driven and creates output that may be viewed using the GrADS package. I have run the model successfully on Mac OS X (PPC and Intel), Linux and Sun computers using commercial and free (g77/g95) compilers. I have not yet tried to use DTDM on Windows. More information regarding DTDM and GrADS may be found at these sites

```
http://www.atmos.ucla.edu/~fovell/DTDM
http://www.iges.org/grads/
```

Separate instructions are given below for getting DTDM up and running on the Department's Sun computers (in MS 7101) and on your own machines.

Obtaining the package

On the Sun: When you log into a Sun workstation and open a terminal window, you are in your "home" directory. First, from the command line, issue this instruction precisely as written (this font will be used to indicate command-line text):

```
cp /slow1/fovell/DTDM_package.tar .
```

This copies (`cp`) the file `DTDM_package.tar` from the filesystem `/slow1` to your present directory, preserving its name. That's what the period stands for. Make sure the period is included, and there is a space between the period and the filename.

On your machine: Download `DTDM_package.tar` from the DTDM web page. Move it to a location where it can be unpacked.

Unpacking the archive

The "tar" file is an archive. Next, we unpack the archive

```
tar -xvf DTDM_package.tar
```

This will create a new directory called `DTDM`. Next, we move into this directory

```
cd DTDM
```

`cd` stands for "change directory". Type `ls` to list the files in the directory. The result should look something like this (ignore the asterisks and forward slashes; they are not part of the filename):

```
Makefile                              input_strfcn_isolated_up4.txt
input_coldpool_hires.txt              input_strfcn_isolated_up8.txt
input_coolzone.txt                    input_strfcn_rolls.txt
input_generic.txt                     input_thermal.txt
input_hsrc.txt                        makesun.csh*
input_sbf_no_rolls.txt                scripts/
input_sbf_with_rolls.txt              src/
input_seabreeze.txt                   start.gs
input_strfcn_isolated_nowind.txt
```

There are two subdirectories, `scripts` and `src`. The `txt` files are sample model input scripts; we will use the `input_thermal.txt` script first. Files with the `gs` extension are scripts used with the visualization system, GrADS. There are more in the `scripts` directory.

Modifying the input scripts

Edit `input_thermal.txt` in a text editor. The Sun system has a fairly straightforward text editor called `nedit` which we will use to view and modify the input scripts. To view the primary input file we will use, issue the command

```
nedit input_thermal.txt
```

The input file is a Fortran namelist consisting of several sections, the first of which is

```
&experiment
        casename = 'thermal.anelastic',
$
```

This defines what the model output files will be called. The next section declares the number of horizontal (`nx`) and vertical (`nz`) grid points, the time step (`dt`), horizontal and vertical grid spacings (`dx` and `dz`) in meters, and integration length (`timend`) in seconds. The `plot` variable specifies the plotting interval, also in seconds. You should not change any of these at this time.

```
&grid_run
nx = 101,
nz = 84,
dx = 1000.,
dz = 250.,
dt = 1.0,
timend = 1200.,
plot = 120.,
$
```

Next comes the `&framework` section. The settings below tell the model to employ the anelastic approximation (`ianelastic = 1`) and provide pressure decomposition and acceleration data (`ipressure = 1`). If the model is not anelastic, `csnd` specifies the sound wave speed. Otherwise, it is ignored.

```
&framework
ipressure = 1,
ianelastic = 1,
csnd = 100.,
$
```

Skip down to the `&environ` section, which sets up the model base state. Here, you can set the stability of three layers of the atmosphere, the planetary boundary layer (PBL), the free troposphere and the stratosphere, by specifying the Brunt-Vaisalla frequencies. The PBL is pre-set to `pbld` = 2000 m deep, and the tropopause is at `tropo` = 12000 m above ground level. The script is pre-configured to make the atmosphere dry adiabatic, so (`bvpbl`), (`bvtropo`) and (`bvstrat`) are set to 0.0. There is no surface horizontal wind (`usurf`) and no shear, so the wind is initially calm everywhere. Up to three different shear layers can be configured in this section.

```
&environ
bvpbl = 0.00,
pbld = 2000.,
bvtropo = 0.00,
tropo = 12000.,
bvstrat = 0.00,
psurf = 965.,
usurf = 0.,
$
```

The next sections are used to set up a variety of physical processes and phenomena, including bubbles, heat fluxes, and momentum and heat sources. Our interest will be in the `&thermal` section.

```
&thermal
ithermal = 1,
delt = 3.,
radx = 8000.,
radz = 4000.,
zcnt = 3000.,
$
```

When `ithermal` = 1, the model starts with an initial, radially symmetric bubble. At the bubble center, located at `zcnt` meters above the ground, the bubble will be `delt` degrees Celsius warmer than its surroundings. The bubble's horizontal and vertical radii are `radx` and `radz` meters, respectively. Save your changes (if any) to the script and exit `nedit`.

Making the model executable

If you are on the Sun, execute this command to make the model: `./makesun.csh`. Otherwise, edit `Makefile` in a text editor, and select the proper lines applicable to your machine. The # sign indicates a comment line that is ignored. From the command line, type `make`. In any case, a new file called `dtdm` should appear in the directory. This is the model executable. You only have to do this step once, unless you alter the source code contained in the `src` subdirectory.

Running the model

The DTDM model is executed from the command line by invoking the executable and providing a script to it.

```
dtdm < input_thermal.txt
```

Make sure you use the less-than sign in the above; using the greater-than sign will destroy the input script file! The command above will run the model, periodically printing some information to the screen. The model run will take several minutes on these old, slow Suns. If the model runs successfully, you will see the "normal model stop" message. You should see two new files, based on the `casename` set in the input script, like these

```
thermal.anelastic.ctl
thermal.anelastic.dat
```

These are GrADS files. The "dat" file is a binary file containing fields generated by the model; the "ctl" file tells GrADS how to read the binary data.

Using GrADS

GrADS is an interactive visualization system that is also highly scriptable. Invoke GrADS at the command prompt with

```
gradsnc -l
```

A new graphics window should appear. The GrADS prompt is `ga->`. At this prompt, open the files created in your model run.

```
ga-> open thermal.anelastic
```

You do not normally need to include the file extension. This command tells GrADS to open the ctl file of that name; the ctl file then tells GrADS to open the dat file. To see what's in the file, issue the command `q file`, where "q" stands for "query". The result is

```
ga-> q file
File 1 : DTDM demo simulation
  Descriptor: thermal.anelastic.ctl
  Binary: thermal.anelastic.dat
  Type = Gridded
  Xsize = 99  Ysize = 1  Zsize = 82  Tsize = 11
  Number of Variables = 17
    u 82 0 horizontal velocity
    up 82 0 pert horizontal velocity
    w 82 0 vertical velocity
    th 82 0 potential temperature
    thp 82 0 pert potential temperature
    pi 82 0 ndim pressure
    pip 82 0 pert ndim pressure
    ppmb 82 0 pert pressure in millibars
    pbyc 82 0 buoyancy pressure in millibars
    pdyn 82 0 dynamic pressure in millibars
    ptot 82 0 total pressure in millibars
    dudtd 82 0 U acceleration - dynamic
    dudtb 82 0 U acceleration - buoyancy
    dudtt 82 0 U acceleration - total
    dwdtd 82 0 W acceleration - dynamic
    dwdtb 82 0 W acceleration - buoyancy
    dwdtt 82 0 W acceleration - total
```

This tells me my model domain is 99 points wide and 82 points high (both slightly smaller than the values of `nx` and `nz` set in the input script), that there are 11 times in the file, and 17 variables have been defined. The perturbation (pert) fields `up, thp, pip` are deviations from the initial environmental state. Thus, variable `th` plots the full potential temperature field, including the initial environmental variation with height, but `thp` plots only the departure of this field from the base state. If the initial winds are calm, the `u` and `up` fields will be identical.

The model actually prognoses a nondimensionalized pressure, but you may be more interested in `ppmb`, perturbation pressure in millibars. The variables `pbyc`, `pdyn` and `ptot` are the buoyancy, dynamic and total pressures decomposed from the pressure field, expressed in millibars, where `ptot` = `pbyc` + `pdyn`. If the model is anelastic, `ppmb` and `ptot` should be identical. If the model is compressible, then ptot is the anelastic approximation to ppmb; it may be interesting to compare those fields, especially to see how different they are as a function of csnd.

A quick look at the model fields: Execute these GrADS commands

```
ga-> start.gs
ga-> c
ga-> set t 1 11
ga-> d thp
```

By default, GrADS uses a black background. The first command executes a GrADS script called start.gs that, among other things, reverses the background color. The c command is short for clear. There are 11 times in this file; set t 1 11 asks GrADS to display all of them, in sequence. This makes for simple animations. The d thp command tells GrADS to display the thp field. GrADS defaults to contour plots using a rainbow color sequence. Hit return and watch the bubble rise and change shape as it does so.

The next set of commands select a single time and makes a shaded plot of thp. Then, a color bar is drawn. Finally, titles are added and an image file is saved (see Fig. 1). The default GrADS color table is very ugly, but this is easily changed.

```
ga-> set t 6
ga-> set gxout shaded
ga-> d thp
ga-> scripts/cbarn.gs
ga-> draw title Perturbation theta t = 6
ga-> printim theta_t6.gif gif
```

Now execute the GrADS script scripts/thermal.gs. This script tells GrADS to zoom in on part of the domain and plots vertical velocity using a new color table for the color shading. Then, we are switching back to contour plotting and superimposing the buoyancy pressure field (pbyc). See Fig. 2.

```
ga-> run scripts/thermal.gs
ga-> set gxout contour
ga-> d pbyc
ga-> draw title w (colored) and pbyc (contoured) at t = 6
ga-> printim w_pbyc_t6.gif gif
```

You can run the scripts/thermal.gs script repeatedly and superpose different fields on the shaded w field in the background. You can also superimpose wind vectors, though the imager may

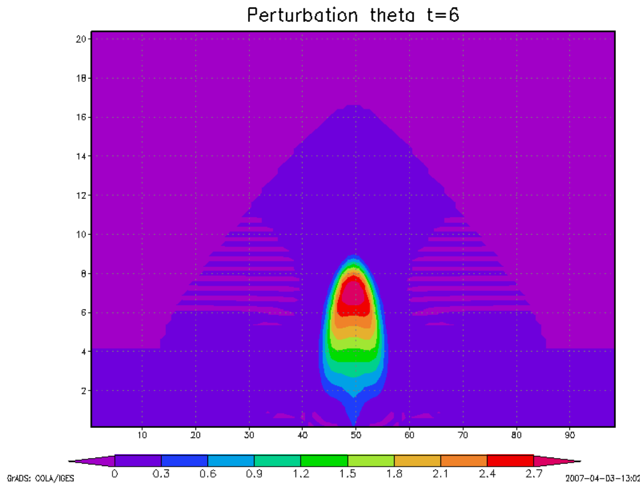

Figure 1: Example GrADS output.

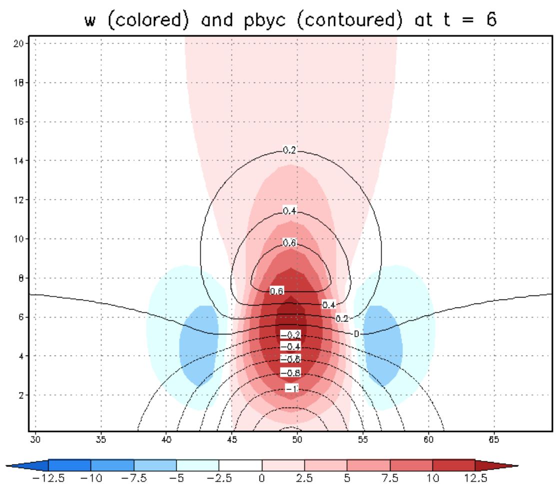

Figure 2: Output from thermal.gs script.

be messy if you try to plot every single grid point. Try out the commands below, to see what they look like.

```
ga-> c
ga-> d u;w
ga-> c
ga-> d skip(u,2,2);w
```

Finally, to end GrADS, type `quit`.