Script for Snowstorm test case SNOW2023
ATM419/563 Fall 2024

*** ------------- preliminaries -- ***
* make a directory in your lab space called **SNOWSTORM**, and move into it
* copy **$LAB/SNOWSTORM/SETUP.TAR** and
* unpack it: **tar -xvf SETUP.TAR**
* execute **sh make_all_links.sh**

*** ------------- visualize domain and run geogrid ----------------------------------***
Launch ARCC Jupyterlab: https://jupyterlab.its.albany.edu/hub/login
One node on batch suffices. Execute `plot_WRF_domain.ipynb` notebook

srun -n 4 -p burst-daes geogrid.exe
{Look for: "*Successful completion of geogrid.*"}

On Jupyterlab, execute `plot_WRF_terrain.ipynb` notebook to visualize terrain

csh max.csh MAPFAC_M geo_em.d01.nc
ncview geo_em.d01.nc
[select 2D variable MAPFAC_M from drop down menu to view map factors]
[select 2D variable HGT_M from drop down menu to view terrain height]

*** ------------- ungrib ---***
link_grib.csh $LAB/DATA/GFS_2023012212/gfs.* .
ls -al GRIBFILE* [make sure everything is OK]
/bin/wgrib2 GRIBFILE.AAA | more [looking at contents]

cp Vtable.GFS Vtable [select correct Vtable!]

UNGRIB CAN BE TIME-CONSUMING AND CAUSE RESOURCE CONTENTION
Listen for which option we will use for this demonstration
Option (A): Run ungrib using srun
srun -p burst-daes ungrib.exe ***(output goes to screen)***

Option (B): Submit ungrib as a batch job
sbatch -p burst-daes submit_ungrib
tail -f ug.srun.out *Break out of tail with ctrl-C*

Option (C): Link to prepared ungrib outputs [that space and dot are important]
ln -s $LAB/SNOWSTORM/UNGRIB/FILE* .

Ungrib is done when you see: *"Successful completion of ungrib."*
Ungrib makes 10 gigabytes worth of outputs…
ls FILE*

*** -------------- metgrid ---***
srun -n 4 -p burst-daes metgrid.exe
[look for Successful completion.... If issues, check metgrid.log.0000 file]
ls met_em*

ncdump -h met_em.d01.2023-01-22_12:00:00.nc | more [TAB COMPLETION!]

[Notice is says `num_metgrid_levels = 34` in the header information]
[Note in `namelist.input`, we specify `num_metgrid_levels = 34`]

*** -------------- TOUR of batch scripts ---***
(see PPT)

*** -------------- real.exe ---***
sbatch -p burst-daes submit_real

[NOTE JOB NUMBER ASSIGNED. Example: Submitted batch job 774952]
[check job status as directed]
myjobs

[when job is finished, check ‘tail’ of rsl.out.0000 file with ‘trsl’ command.
Make sure it says “SUCCESS COMPLETE REAL_EM INIT”]
trsl *Break out of tail with ctrl-c*

ls -al wrfbdy* wrfin*

*** -------------- wrf.exe -- ***
sbatch -p burst-daes submit_wrf

[check job status as directed. WRF runs should take about 2 minutes.]
myjobs

* monitor WRF run
trsl *(ctrl-c to break out)*

[check for successful completion with ‘trsl’]

ls -1 wrfout_d01* (Verify you have wrfout_d01_2023-01-22_12:00:00)

*** -------------- analyze WRF simulation ------------------------------ ***

• Launch WRF_plot_SNOW2023_V3.ipynb

Cell #1 = openers
Cell #2 = useful functions

Cell #3 = define and open WRF output. Should not require editing.
Cell #4 = Extract some fields from WRF output
Cell #5 = Plot model topography
Cell #6 = Prepare for a plot of 10m winds, microphysics total precipitation, and cumulus total precipitation at final forecast time
Cell #7 = Plot 10m winds, microphysics total precipitation, and cumulus total precipitation at final forecast time
Cell #8 = Extract snow depth for a single location, convert to inches, and plot as a time series

*** -------------- TOUR of namelist.input settings -------------------- ***
(see PPT, slides 43-end)

• examine model vertical grid (see slide 50)
dopython
python read_wrfinput.py wrfinput_d01