CESM Application Homework

Ben Kravitz ([bkravitz@iu.edu](mailto:bkravitz@iu.edu) or Slack)

Please post all questions in the Google Classroom so everyone can see them

**Problem 1 (10 points)**

Find a paper using CESM (scholar.google.com can help you out a lot). Answer the following questions:

1. What was the goal of the paper? (2-3 sentences at most)
2. How did they use CESM? What sorts of simulations did they run?
3. What kinds of output does CESM produce? You can think of this in terms of variables (temperature, precipitation, etc.) or types of information (time series, maps, etc.).
4. Why did the authors use a climate model as opposed to some other source of data?

**Problem 2 (50 points)**

Let’s install and run CESM!

1. Follow the instructions for installing CESM (chameleon\_instructions.txt and runcesm\_instructions.txt, the basic running instructions). When it completes, it should output a bunch of log files in the BLD directory (specified as BLDDIR). Look for the file that has “cesm.log” in it and upload that here.
2. To install CESM, you had to load in the NetCDF library. What is a NetCDF file? Why are they useful?
3. The basic running instructions will compile and run CESM. When it completes, it should output a timing file in the CASE directory (specified as CASEROOT) and some NetCDF files in the OUTPUT directory (specified as RUNDIR or DOUT\_S\_ROOT, depending on whether the short-term archiver ran). Specifically, there should be a file that has “cam.h1” in the name. Follow the instructions for ncdump and output the results of ncdump into a text file. Upload that text file here.
4. Now postprocess the CESM output so that you have three files: TREFHT, PRECT (which is the sum of PRECC and PRECL), and U10. Each of these files should contain all times output by the model. Note that you can use some of the instructions in runcesm\_instructions.txt, but you’ll need to change a few things.
5. Now run the analysis scripts (make\_map\_plot.py and make\_line\_plot.py) on the TREFHT, PRECT, and U10 NetCDF files and upload the figures it creates. Note that you will likely need to modify the scripts somewhat. Describe the figures that you see: what are you looking at, and what features does it have?
6. The run should output two timing files. Upload them here. How many model years per day do you get with your current setup? What is the slowest model component?

**Problem 3 (20 points)**

Let’s change the processor layout!

1. Follow the instructions for changing the processor layout (sometimes abbreviated pe layout). This should look very similar to homework #2a and #2c. This will produce a new run of the model. Make sure you call it something different so it doesn’t overwrite the previous run.
2. Postprocess and plot the output as you did in homework #2e. (You may want to use the provided Python script that allows you to compare two different runs.) Do you notice any differences between the two model runs? Why or why not?
3. Now compare the timing files between this run and the previous run (homework #2f). Do you notice any differences between the two model runs? Why or why not?
4. Let’s say you were running CESM in the actual competition, and your objective was to get the longest run possible. Do you think it would be best to run for a short time but on as many processors as possible, or do you think it would be better to use a low number of processors and run the model on them the entire time? Remember that you have other applications that you need to complete in the competition, so you cannot only run the CESM application and expect to win.

**Problem 4 (30 points)**

Now let’s work with some other parameters you might want to adjust in CESM.

1. Let’s change the resolution of the model (how big the grid boxes are). Note: for the competition you will only be running with one resolution. But you might want to use coarser resolution for testing. In the run script (testrun1), line 7 defines RES. Change f09\_f09\_mg17 to f19\_f19\_mg17. What exactly are you doing by changing this line? Look at the timing file after your run completes. Do you notice any differences from your run in Problem 2?
2. Let’s have the model output restart files. In your run script (testrun1), line 29 defines REST\_OPTION. Change nmonths to ndays. What is this doing? Do you notice any differences in how long your run takes? Why?
3. Let’s do a continuation run. Set RESUBMIT=’2’ so that the model will run for 15 days total, but in three blocks. It should create a new timing file for each block. Compare them and describe any notable differences. Notice how much time model initialization takes; there are drawbacks to doing many short runs.