CESM tutorial, AGU, 9th Dec 2018

Useful Resources

- CESM2 main page: http://www.cesm.ucar.edu/models/cesm2/
- CIME users guide: http://esmci.github.io/cime/

  where to learn about the software infrastructure used in CESM (many of the basics taught here in the tutorial)

- NCAR's Climate Data Gateway: https://www.earthsystemgrid.org/

  where you can download CESM simulations provided by NCAR.

- Component users guides:

  CAM6: https://ncar.github.io/CAM/doc/build/html/users_guide/index.html#users-guide
  CLM5: http://www.cesm.ucar.edu/models/cesm2/land/
  POP2: http://www.cesm.ucar.edu/models/cesm2/ocean/
  CISM: http://www.cesm.ucar.edu/models/cesm2/land-ice/
  CICE: http://www.cesm.ucar.edu/models/cesm2/sea-ice/
  River Run Off: http://www.cesm.ucar.edu/models/cesm2/river/
  Wave Model: http://www.cesm.ucar.edu/models/cesm2/wave/

- CESM tutorial website: http://www.cesm.ucar.edu/events/tutorials/

  for more in depth lectures and tutorials on CESM.

- CESM2 namelist parameters:

  http://www.cesm.ucar.edu/models/cesm2/settings/current/

  Information on available namelist parameters can be found in the "Namelist Definitions" page for each component.

- XML parameters (i.e., parameters controlling your simulation)

  http://www.cesm.ucar.edu/models/cesm2/settings/current/drv_input.html

- CESM simpler models page: http://www.cesm.ucar.edu/models/simpler-models/

  for idealized configurations of CESM and information on source code modifications, namelist changes etc that may be of use with the comprehensive version of CESM.

- CESM experiments page: http://www.cesm.ucar.edu/experiments/

  for information about available CESM output.

- CESM working groups page: http://www.cesm.ucar.edu/working_groups/

  Contains links to the pages of each individual working group for useful information relative to a particular component or a particular aspect of Earth System Science.

- CESM bulletin board: https://bb.cgd.ucar.edu/about

  Where you can post questions that will be answered by NCAR staff and other users.

1 Introduction

1.1 CESM directory structure

Two things are necessary for running CESM: the CESM code itself and input datasets that are specific to the case you're running e.g., prescribed sea surface temperatures in the case of an uncoupled simulation or greenhouse gas/aerosol forcing datasets.

The location of the input data has been set up for you as the environment variable `$CESMDATAROOT`.

The location of the CESM root directory which contains all the CESM scripts and code that you'll be using has already been set up for you as an environment variable `$CESMROOT`. Go into the directory `$CESMROOT`:

```
cd $CESMROOT
```

This contains all the CESM code and scripts necessary to run CESM. The following two directories are likely to be of most interest:

`$CESMROOT/cime`: contains all the scripts necessary to set up and run a new case.

`$CESMROOT/components`: contains the model source code for each component.

As an example

```
cd $CESMROOT/components/cam/src/physics/cam
```

There you will see all the source code for physical parameterizations within CAM.

1.2 Getting started: setting up, compiling and submitting a case

The scripts that you'll use to set up a CESM case are located in `$CESMROOT/cime/scripts`

```
cd $CESMROOT/cime/scripts
```

The script "create_newcase" is used to set up a new case. This has three required inputs and other optional inputs. The required inputs are

- **case**: the name and directory path of your case e.g., setting case to `~/cases/case01` will make a simulation called `case01` and it will be located in `~/cases`
- **compset**: the CESM component set that you'd like to use. See

  http://www.cesm.ucar.edu/models/cesm2/config/compsets.html

  for a list of pre-defined compsets and

  http://esmci.github.io/cime/users_guide/compsets.html

  for documentation on how to create your own. Note that here we will be using the compset B1850-tutorial which has been specially developed for the purposes of this tutorial - it is not available within the CESM2 release.
- **res**: the resolution that you'd like to use.

In addition to the CESM web pages, you can find information on available compsets and resolutions using the "query_config" tool located in `$CESMROOT/cime/scripts` e.g.,

```
./query_config --help
```

will provide you with information on the arguments taken by `query_config` and then e.g.,

```
./query_config --grids
```

will list all the available grids. You can use this tool to find the arguments you want to provide to "create_newcase".

Exercise 1: Setting up and running your first case

Use the 4 CESM commands to create a new case. Do the setup, build and submit your job. Use *qstat* to check your job is running. Check that your run completes. Check the files are in the archive directory.

Decide on a location where you want to keep your CESM cases. We'll use a directory "cases" within your home directory as an example e.g.,

```
cd                  (Takes you to your home directory)
mkdir cases         (Makes the directory cases)
```

To start, we will create a coupled simulation under 1850's conditions using a 2 degree atmospheric resolution and the g37 ocean grid. We'll call it "case01" and put it in your "cases" directory.

```
./create_newcase --case ~/cases/case01 --compset B1850-tutorial --res
f19_g37
```

NOTE: B1850-tutorial is not an available compset in CESM2. It is a compset that has been specially designed for this tutorial. Also, the resolution f19_g37 is not supported within CESM2. It is used in this tutorial for efficiency

Now go in to your cases directory. You should see a sub-directory `case01`. This is your case directory and contains all the scripts necessary to compile and run CESM. From within your case directory, CESM can be set up, built and run, using the following three commands

```
./case.setup        (sets up the case)
./case.build        (builds the case)
./case.submit       (submits the case to the queue)
```

NOTE: for the purposes of this tutorial we are actually using a pre-compiled version of CESM. In reality, the build stage will take longer than you are finding here.

Once you have submitted the case, you can see its status in the queue with the following command

```
qstat
```

You can also see CESM outputting data as the simulation progresses in the run directory, which for our purposes, is located at ~`/scratch/case01/run`. This is where the data are output until the run is finished, at which point it is moved to the archive directory. For our purposes, this is located at ~`/scratch/archive/case01`.

Once the run has finished, you can go into ~`/scratch/archive/case01/atm/hist` and see the output CAM history files. However, for this first run, the default length of the run is 5 days and only monthly averages are output, so that directory is empty at the moment. But, if you go

into ~/scratch/archive/case01/rest/ you'll see a directory for the 6th January, year 1 that contains the restart files that would be necessary to continue this run further.

Exercise 1: Setting up and running your first case - *solution*

A B1850-tutorial case with the 2 degree atmosphere and g37 ocean grid and named "case01" can be set-up, built and submitted with the following commands

```
cd $CESMROOT/cime/sripts
./create_newcase --case ~/cases/case01 --compset B1850-tutorial --res f19_g37
cd ~/cases/case01
./case.setup
./case.build
./case.submit
```

Overall, there are 5 main directories that are being used by CESM:

- INPUT DATA Directory: `$CESMDATAROOT/inputdata` *(Contains all the input datasets for CESM)*
- CESM Code and scripts: `$CESMROOT` *(Contains the CESM source code and scripts for setting up cases)*
- CASE Directory: Here it is at `~/cases/case01` *(where you control aspects of your case, set up the case, compile and submit)*
- BUILD/RUN Directory: Here it is at `~/scratch/case01` *(contains the compiled executables for CESM and the run directory where CESM will output data as it runs)*
- ARCHIVE Directory: Here it is at `~/scratch/archive/case01` *(contains archived history and restart files once the run has completed)*

See Fig. 1 for a schematic depiction of these directories.

**********Wait Here**********

Spend the time familiarizing yourself with the above directories and their contents before we move on to the next exercise as a group. Take this time to ask any questions you might have to the tutorial instructors.

****

2 Resubmitting and changing the run length

Exercise 2: Continuing case01 for another month

Go to the case directory of case01 `~/cases/case01`. In here, there are many files that are used to control your case. Browse the file `env_run.xml`. This contains parameters that can be used to control the characteristics of your run such as the run length or whether it is a start up run or the continuation of a run. Any of the entries within this file can be edited with the "xmlchange" command, as you'll see below. Using xmlchange ensures that the changes you make propagate to all necessary locations.

Here you'll learn how to extend your case01 run for another month.

The length of the run is determined by the xml parameters "STOP_OPTION" and "STOP_N". Look at what they are currently set at in `env_run.xml`. You can do this either by simply looking in the file or by running the following xmlquery command from within the case directory

```
./xmlquery STOP_OPTION,STOP_N
```

You should find `STOP_OPTION=ndays` and `STOP_N=5` corresponding to the 5 day run you just performed.

The next segment of the simulation that we're going to perform is going to be 1 month long. This can be achieved by running the following xmlchange command from within your case directory

```
./xmlchange STOP_OPTION=nmonths, STOP_N=1
```

Run this command and take a look at the values of `STOP_OPTION` and `STOP_N` in `env_run.xml`. Have they changed?

Now, we need to ensure that we continue on from our first 5 day segment, rather than starting a new initial run. This is achieved using the `CONTINUE_RUN` parameter

```
./xmlchange CONTINUE_RUN=TRUE
```

ensures that this run continues on from the first one.

The above commands can all be performed at once, if you prefer

```
./xmlchange STOP_OPTION=nmonths, STOP_N=1, CONTINUE_RUN=TRUE
```

In order to perform a continuation of the run, you will need all the necessary restart files in your run directory. They are already there for the run we just performed, but, if that wasn't the case, you'd need to copy all the contents of the restart directory e.g., for our case

`~/scratch/archive/case01/rest/0001-01-06-00000`, into your run directory.

Now you can submit the run again.

```
./case.submit
```

Note that you don't have to set-up or re-build the run when performing a continuation like this

Check it is in the queue or running: `qstat -a`

Note that we performed the initial 5 day run and then resubmitted it for another month for demonstration purposes here. In practise, if you are outputting monthly data, you would really want to run continuously for an integer number of months. Otherwise, the monthly averages for the sea ice model will not be accurate. In this case, the January history file output for the sea ice model will actually be the average from days 5 to 31.

Exercise 2: Continuing case01 for another month- ***solution***

The run case01 can be continued for another month by executing the following commands from within `~/cases/case01`

```
cd ~/cases/case01
./xmlchange CONTINUE_RUN=TRUE, STOP_OPTION=nmonths, STOP_N=1
./case.submit
```

Another important parameter that we have not made use of here is `RESUBMIT`. This controls how many times you want to resubmit and run another segment of the simulation of length `STOP_N`. For example, if you were to set `RESUBMIT=2` then when you submit the job to the queue it will first run one month and then subsequently resubmit twice, running another month each time i.e., a total of 3 months. If you set resubmit to a non-zero value when you make your first submission, it will automatically set `CONTINUE_RUN=TRUE` after the first segment.

The above is one example of controlling your simulation using xml variables. You can find a listing of all the available xml variables along with their description by running the following command from your case directory

```
./xmlquery --listall --description
```

or for the variables in a specific file e.g., env_run.xml

```
./xmlquery --listall --file env_run.xml --description
```

You can also find the same information on the CESM website here

http://www.cesm.ucar.edu/models/cesm2/settings/current/drv_input.html

**********Wait Here**********

It's going to take some time for the 1 month simulation you've just submitted to complete. Spend the some time exploring the xml files in the case directory (`env_*.xml`), the run directory and the online documentation before we move on to the next exercise as a group. Take this time to ask the tutorial instructors any questions you might have.

****

3 Finding out timing statistics

Statistics of the timing of your simulation can be found in the directory "timing" within your case directory.

```
cd ~/case01/timing
```

In the file with the name that starts "cesm_timing.case01" you'll find the following information

```
Overall Metrics:
     Model Cost:                XXX pe-hrs/simulated_year
     Model Throughput:          XXX simulated_years/day
```

"Model Cost" is the cost of the simulation in processor element hours. "Model Throughput" is how many simulated years you can run in one day of real time.

Typically you will be running on a system that has a wall clock limit for each submission. For example, the cheyenne system only allows you to run continuously for a maximum of 12 hours. Normally you would want to choose the maximum length of a run that would fit within a 12 hour wall clock limit, while breaking the simulation into reasonably sized chunks e.g., an integer number of years.

Exercise 3: working out the optimum segment length

Consider the "Model Cost" for the simulation you've just run. If you wanted to run a 30 year simulation with each segment being an integer number of years long and you have a 12 hour wall

clock limit on your machines queue, what values of `STOP_OPTION`, `STOP_N` and `RESUBMIT` would you choose to minimize the number of individual re-submissions you would have to make?

Also browse the restart files and the log files. Ask any questions you may have about them.

Exercise 3: working out the optimum segment length - *solution*

Suppose you find that your model throughput is 12 simulated years per day. This means that per 11 hour wallclock limit, you can run roughly 5.5 years. However, it's best to run a little less than what might appear optimum from this run because the exact throughput can vary. Also, it's best to run an integer number of years, if possible. You can probably safely run 5 years in a 12 hour wallclock limit. So, set `STOP_OPTION=nyears` and `STOP_N=5`. Now, you want to run 30 years, that means you need to submit the model 6 times (5x6=30). That means that you need 5 re-submissions after your first one. So you would set `RESUBMIT=5`. This can be achieved with

```
./xmlchange STOP_OPTION=nyears, STOP_N=6, RESUBMIT=5
```

4 Model output

Exercise 4: Looking at the output from the 1 month simulation

Your one month simulation should have completed by now, go into

~/scratch/archive/case01/atm/hist.

Do you see a history file now? Take a look at what's inside. The following will list all the variables in the file and their attributes

```
ncdump -h case01.cam.h0.0001-01.nc | more
```

Press enter to continue scrolling down the entries.

To view fields, you can try the following

```
ncview case01.cam.h0.0001-01.nc &
```

If, for some reason your simulation hasn't completed. You can see example output at

/home/isimpson/scratch/archive/case01/atm/hist/

5 Namelist Modifications

Namelist parameters can be used to control various aspects of your simulation. In your case directory ~/cases/case01 you'll see a number of files of the form `user_nl_XXX` with `XXX` corresponding to the various CESM components e.g., `user_nl_cam` for CAM. It is in these namelist files that you can modify namelist parameters.

You can find information on all the available namelist parameters within each component here:

http://www.cesm.ucar.edu/models/cesm2/settings/current/

5.1 Doubling the CO_2 concentration

We'll start with a simple example: doubling the CO_2 concentration.

Exercise 5: Doubling the CO_2 concentration

Make a new case using the `B1850-tutorial` compset and `f19_g37` resolution. Put it in `~/cases` and call it `b1850_2x` i.e.,

```
./create_newcase --case ~/cases/b1850_2x --compset B1850-tutorial --res
f19_g37
```

Set up the case.

You can preview the namelists that the case is using with the following command in your case directory

```
./preview_namelists
```

This will generate files with names `XXX_in` where `XXX` corresponds to the model component e.g., `atm_in` for the atmosphere component. These files are generated both in the `CaseDocs` subdirectory of your case directory and the run directory. Take a look at `./CaseDocs/atm_in`.

Now for the `B1850-tutorial` compset, CO_2 concentrations are controlled by the namelist parameter `co2vmr` i.e., CO_2 volume mixing ratio. You should see in the atm_in file that it is currently set to 284.7e-6.

To double the CO2 concentration we need to modify `co2vmr` by setting it to the value we want in the `user_nl_cam` file. **DO NOT change it in the atm_in file. That will not affect the simulation. The atm_in file is generated simply for reference when the case is built or submitted**. So place the following entry in `user_nl_cam`

```
co2vmr=569.4e-6
```

preview the namelist. Has your change been propagated to the `atm_in` file?

We won't submit this case, this is just to get you familiar with the namelists and `preview_namelists`. The next example will also be a namelist change and you will actually submit that simulation.

Exercise 5: Doubling the CO2 concentration - *solution*

To create a new case using the B1850-tutorial compset with 2 degree atmospheric resolution and g37 ocean resolution you can do the following

```
cd $CESMROOT/cime/scripts
./create_newcase --case ~/cases/b1850_2x --compset B1850-tutorial --res f19_g37
cd ~/cases/b1850_2x
./case.setup
```

Now add the following line to user_nl_cam

```
co2vmr=569.4e-6
```

Now run

```
./case.setup
```

and check your changes have propagates to the namelist i.e., examine the files

~/cases/b1850_2x/CaseDocs/atm_in

and

~/scratch/b1850_2x/run/atm_in

We're not doing this here, but if you wanted to actually submit the run, you'd do

```
./case.build
./case.submit
```

5.2 Changing output variables and frequencies

The default output for CESM is monthly averages and, for CAM, these monthly average history fields are output in the ".h0." files as you had seen in Exercise 4. But you can output up to 10 CAM history files from ".h0." to ".h9." containing different variables or frequencies of output.

In CAM, the namelist parameter `nhtfrq` controls the output frequency for each history file.

If `nhtfrq=0` the file will contain monthly averages.
If `nhtfrq=X` where `X` is an integer value > 0, then output will be produced every `X` model timesteps.
If `nhtfrq=X` where `X` is an integer value < 0, then output will be produced every `X` hours.

For example, if you want the history field output to be at daily frequency, this can be achieved by setting `nhtfrq=-24`.

The number of time values in each history file is controlled by the parameter `mfilt`. For example, if we wanted to specify that each history file were to contain 10 time values, then we would set `mfilt=10`.

Finally, the list of history fields that are desired in each history file can be specified using the namelist parameter `finclX` where X ranges from 1 to 10. For example, for the first history file ".h0." which by default contains many monthly fields, you can add additional fields, say, `PRECT` (total precipitation) and `Z500` (geopotential at 500hPa) with the following

```
fincl1='PRECT','Z500'
```

Using a ":X" following a field specifies the averaging flag, where valid values of `X` are

I for instantaneous
A for average
M for minimum
X for maximum

For instance, the line

```
fincl1='PRECT:M'
```

is used to add the minimum value, over all model timesteps within the month, of `PRECT` to the ".h0." file.

Note that the value you use in `finclX` is one greater than the value that corresponds to the history file name e.g., `fincl1` corresponds to the ".h0." file, `fincl2` corresponds to the ".h1." file and so on.

In practise, you can specify `nhtfrq` and `mfilt` for each file in a sequential list or with `nhtfrq(X)` and `mfilt(X)` where `X` corresponds to the file number, `X`=1 for ".h0.", `X`=2 for ".h1." and so on.

For example, the following two sets of namelist parameters are equivalent

```
fincl2='T:I','Q:I','U:I','OMEGA:I'
nhtfrq = 0, -3
mfilt = 1, 8
```

```
fincl2='T:I','Q:I','U:I','OMEGA:I'
nhtfrq(1)=0
nhtfrq(2)=-3
mfilt(1)=1
mfilt(2)=8
```

Both ensure that the ‘‘.h0.’’ history file continues to contain monthly averages with one month per file, while additional output of 3 hourly instantaneous `T`, `Q`, `U` and `OMEGA` is added and will appear in the ‘‘.h1.’’ file with 8 values in each file i.e., 1 file per day.

Note that you cannot add or remove output variables part way through a simulation when running a continuation run. If you want to change the output frequency part way through a simulation, you have to run a branch or a hybrid run. See Exercise 8.

Namelist parameters controlling higher frequency output in other CESM components:

In CLM

`hist_nhtfrq`: output frequency of the history file
`hist_mfilt`: number of samples in each history file
`hist_finclX`: with X equal to an integer value between 1 and 5 can be used to add fields or auxiliary history files

The following example shows namelist parameters that can be specified in `user_nl_clm` to output 4 extra history files with daily, 6-hourly, hourly and every timestep values of `TG` and `TV`, leaving the primary history files as monthly averages:

```
hist_fincl2='TG', 'TV'
hist_fincl3='TG', 'TV'
hist_fincl4='TG', 'TV'
hist_fincl5='TG', 'TV'
hist_nhtfrq = 0, -24, -6, -1, 1
```

In CICE

`histfreq`: output frequency of the history file. Can take values `'1', 'd', 'm', 'y', ...`
`histfreq_n`: number of samples in each history file
`hist_avg`: logical. If False → instantaneous values. If True → time averages

The following example shows namelist parameters that can be specified in `user_nl_cice` to output an extra history file with daily values, leaving the primary history file as monthly.

```
histfreq = 'm','d','x','x','x'
histfreq_n = 1,1,1,1,1
```

In POP2

`tavg_freq`: output frequency of the history file
`tavg_freq_opt`: units that correspond to `tavg_freq` e.g., 'nmonth','nhour','once'
`tavg_file_freq`: frequency at which the model files are written
`tavg_file_freq_opt`: units of time for `tavg_file_freq` e.g., 'nmonth', 'nhour',...

For example, to output a timeseries of daily averages bundled into a monthly file, the following namelist parameters can be specified in `user_nl_pop`

```
tavg_freq_opt = 'nday'
tavg_freq = 1
tavg_file_freq_opt = 'nmonth'
tavg_file_freq = 1
```

Exercise 6: Adding additional, higher frequency, output via the namelist

Create a new case called "b1850_high_freq". Put it in your cases directory and use the B1850-tutorial compset at f19_g37 resolution.

Use the default 5 day run length

Set up the case

Now, in addition to the default monthly output, add the following output for CAM

- an "h1" file containing daily averages of `TREFHT` and `PRECT`
- an "h2" file containing instantaneous values of PSL and U10 every 6 hours

This can be achieved by adding the following into the namelist file `user_nl_cam`

```
fincl2='TREFHT:A','PRECT:A'
fincl3='PSL:I','U10:I'
nhtfrq=0,-24,-6
```

Set your namelist so that you output:

- a single h1 file with all the daily averaged output for the 5 day run
- an h2 file for each day of the month i.e., containing four 6 hourly time values.

i.e., set

```
mfilt=1,5,4
```

in the CAM namelist.

Build your run.

Check in the CaseDocs directory. Is your `atm_in` file correct? Does it contain the namelist additions you made? What about the file `atm_in` in your run directory? You should find that the changes you have made have propagated to all the `atm_in` files. If you would like to propagate the changes to the `atm_in` files without building you can also run the command `./preview_namelists`.

Submit it to the queue. Is it running successfully and producing the extra history files? One way to check is to go to the run directory as it is running and see whether the history files are arriving.

Once it has finished, do you see the history files in the archive directory?

How many time values are in each file? Does it make sense?

Exercise 6: Adding additional, higher frequency, output via the namelist - *solution*

The following is the solution to how to set up a new case and output daily averaged `TREFHT` and `PRECT` and 6 hourly instantaneous `PSL` and `U10`.

```
cd $CESMROOT/cime/scripts
./create_newcase --case ~/cases/b1850_high_freq --compset B1850-tutorial --res
f19_g37
cd ~/cases/b1850_high_freq
./case.setup
```

Now add the following into the file `user_nl_cam`

```
fincl2='TREFHT:A','PRECT:A'
fincl3='PSL:I','U10:I'
nhtfrq=0,-24,-6
mfilt=1,5,4
```

Then run

```
./case.build
./case.submit
```

You can check the nameslist changes have been made in

`~/cases/b1850_high_freq/CaseDocs/atm_in`

and

`~/scratch/b1850_high_freq/run/atm_in`

6 Troubleshooting and log files

Exercise 7: Another simulation with higher frequency output

Create another new case. Call it "b1850_high_freq_bugfixing". Again, put it in your cases directory and use the B1850-tutorial compset at f19_g37 resolution.

Use the default 5 day run length.

Now in addition to the default monthly output, add the following output

- an "h1" file containing daily averages of `T2M`
- set your namelist so that there is one file per day for this daily averaged output.

i.e., you would need the following in `user_nl_cam`

```
fincl2='T2M:A'
nhtfrq=0,-24
mfilt=1,1
```

Set up, build and submit your case. Is it running? When it says completed in the queue, can you see the history files in the archive directory? The answer should be no. Go to the run directory.

Is there any evidence of history files or restart files being created by the run? The answer, again, should be no. This is because we have tricked you, with a bug.

What you should find in your run directory is three log files. One for the coupler (`cpl.log.*`), one for CAM (`atm.log.*`) and one for CESM (`cesm.log.*`).

Somewhere in these log files is information about what has gone wrong, but it is often not entirely straightforward to find.

Often at the bottom of the log file, there are errors that are not relative to your problem because they are just demonstrating that individual processes are exiting. Often the relevant error lies above this and can sometimes be found by searching for the first occurrence of `ERROR` or `ABORT` or `cesm.exe`. In this case, searching for the first occurrence of `ERROR` in `cesm.log.*` gives us some relevant information. We find

```
ERROR: FLDLST: 1 errors found, see log
```

This tells us is that something has gone wrong with the list of output variables that we have asked for.

More information can then be found in the CAM log file (`atm.log.*`).

Look at the very end of that file and you should see

```
FLDLST: T2M in fincl(1, 2) not found
ERROR: FLDLST: 1 errors found, see log
```

This tells us that `T2M` is not a valid history variable for CAM. That's because the correct variable for near surface temperature is `TREFHT`, as we used in the previous example. `T2M` is not a CAM history field and this has caused CESM to crash on initialization.

Exercise 7: Another simulation with higher frequency output - *solution*

```
cd $CESMROOT/cime/scripts
./create_newcase --case ~/cases/b1850_high_freq_bugfixing --compset B1850-tutorial
--res f19_g37
cd ~/cases/b1850_high_freq_bugfixing
./case.setup
```

Now add the following into the file `user_nl_cam`

```
fincl2='T2M:A'
nhtfrq=0,-24
mfilt=1,1
```

Then run

```
./case.build
./case.submit
```

7 Source code modification

As discussed in section 1.1, the CESM source code resides in `$CESMROOT/components/`. If you want to make modifications to this source code, the best practise is to copy the relevant sub-routine into the relevant sub-directory within the "`SourceMods`" directory in your case directory. For example, if you wanted to modify a CAM subroutine, in the case `case01` that you ran at the beginning, you would copy it to the following location

```
~/cases/case01/SourceMods/src.cam
```

and modify it there.

So, the following steps should be used when making a source code modification:

(1) Find the subroutine you want to modify.

(2) Copy this subrouting to the relevant "SourceMods" directory within your case directory

(3) Make your modifications

(4) Compile and run the model

7.1 Example: outputting an extra variable

A common source code modification that you may want to do is to output a new variable. As an example, CAM has a history field that corresponds to the temperature at 500hPa and a number of other pressure levels, but not at 750hPa. Suppose you wanted to output the temperature at 750hPa. The following two calls are required to add an output variable:

```
call addfld('T750',...)      (Add a field to the master field list)
call outfld('T750',...)       (Collect values for this field and write to history file)
```

Each of these are now described in more detail:

addfld

The sub-routine `addfld` adds a field to the master list with the following syntax

```
addfld(fname,type,avgflag,units,long_name)
```

where

`fname`=field name
`type`=the type of field. The entry for a single level field would be "`horiz_only`" and the entry for a 3D field would be "`(/ 'lev' /)`".
`avgflag`=Averaging flag, A = average, I=instantaneous
`units`=the units of the field
`long_name`=Field full name

Appropriate values of these parameters for the output of `T750` are

```
call addfld('T750',horiz_only, 'A', 'K', 'Temperature at 750hPa pressure
surface')
```

outfld

The subroutine `outfld` accumulates (or takes the minimum or, maximum of etc as appropriate) the field into the history buffer for the appropriate history tape with the following syntax

```
outfld(fname, field, idim, c)
```

with

`fname` = Field name
`field` = array containing field values
`idim` = longitude dimension of field array
`c` = chunk (physics) or latitude (dynamics) index.

For example:

```
call outfld('T750',t750, pcols, lchnk)
```

Exercise 8: Add an additional output variable

Create a new case called "b1850_T750" using the B1850-tutorial compset at f19_g37 resolution. Put in in your cases directory.

Use the default 5 day run length

Now we are going to output daily values of T500 and T750 in the "h1" history file i.e., set the following in `user_nl_cam`

```
fincl2='T500','T750'
nhtfrq=0,-24
mfilt=1,5
```

But the problem here is that `T750` is not a history field that is available for CAM by default. We have to make source code modifications to output `T750`.

You can use the output variable `T500` as a template.

First, find the subroutine in which T500 is output in the CAM physics source code i.e., go to the directory that contains the cam physics source code

```
cd $CESMROOT/components/cam/src/physics/cam
```

Now, find the `.F90` module where `T500` is output. You could achieve this using the linux `grep` command e.g.,

```
grep 'T500' *
```

This should reveal that the only `.F90` file that contains `T500` is `cam_diagnostics.F90`. This will be the module that you'll want to modify.

Remember, you don't want to make modifications to the code directly in `$CESMROOT/components/cam/src/physics/cam/`. You want to copy `cam_diagnostics.F90` into the CAM "SourceMods" directory in your case directory i.e.,

```
cp $CESMROOT/components/cam/src/physics/cam/cam_diagnostics.F90
~/cases/b1850_T750/SourceMods/src.cam
```

The compiler will preferentially choose to compile any modules that are present in your SourceMods directory, over those in the original source code.

It is good practise to keep a copy of the original source code alongside the modified one so that you can easily see the changes that you've made and retrace your steps if things go wrong e.g., in your SourceMods directory

```
cp cam_diagnostics.F90 cam_diagnostics.F90-ORIG
```

Then make changes to `cam_diagnostics.F90` but you'll also have the original version to hand (`cam_diagnostics.F90-ORIG`), if you need to retrace your steps. The compiler won't compile `cam_diagnostics.F90-ORIG` because it doesn't end in `.F90`.

To output `T750` you'll want to replicate all the steps that are done for `T500` but do them for `T750` and obviously make sure that the interpolation to the pressure surface is modified to interpolate to 750hPa as opposed to 500hPa.

Make the necessary modifications. Set up and build your case. Did it successfully compile? If not, you have some debugging to do. If it compiled, then submit to the queue.

Once the simulation has completed, go and look at the output with `ncview`. Is there a variable `T750` there? Do the values look appropriate i.e., are they appropriate values for temperature at 750hPa? Are they warmer than the temperatures at 500hPa?

Exercise 8: Add an additional output variable - *solution*

To set up the case:

```
cd $CESMROOT/cime/scripts

./create_newcase --case ~/cases/b1850_T750 --compset B1850-tutorial --res f19_g37

cd ~/cases/b1850_T750

./case.setup
```

To find the source code that needs to be modified:

```
cd $CESMROOT/components/cam/src/physics/cam
grep 'T500' *
```

This will reveal that `T500` is only found in the file `cam_diagnostics.F90`.

Copy this into your SourceMods directory and retain the original copy there for reference

```
cp $CESMROOT/components/cam/src/physics/cam/cam_diagnostics.F90
~/cases/b1850_T750/SourceMods/src.cam

cd ~/cases/b1850_T750/SourceMods/src.cam

cp cam_diagnostics.F90 cam_diagnostics.F90-ORIG
```

There are three occurrences of `T500` in cam_diagnostics.F90 at lines 238, 1059 and 1062. The first of these occurs in the `addfld` command while the second two occur in the section where vertical interpolation and outputting of the variable is performed. These steps need to be replicated for `T750` in an equivalent point in the code and the pressure level interpolation has to be performed to 750hPa instead of 500hPa. This is achieved by placing the following below line 238 in `~/cases/b1850_T750/SourceModes/src.cam/cam_diagnostics.F90`

```
call addfld('T750, horiz_only, 'A', 'K', 'Temperature at 750 mbar pressure surface)
```

and the following below line 1063

```
if (hist_fld_active('T750')) then
call vertinterp(ncol, pcols, pver, state%pmid, 75000._r8, state%t, p_surf, &
extrapolate='T', ps=state%ps, phis=state%phis)
```

```
call outfld('T750', p_surf, pcold, lchnk)
end if
```

Now make the following additions to `user_nl_cam` to output `T750` as well as `T500` at daily frequency

```
fincl2='T500','T750'
nhtfrq = 0, -24
mfilt = 1, 5
```

Build and submit

```
./case.build
./case.submit
```

8 Hybrid and Branch runs

There are three ways in which you can start a simulation with CESM. Which way you use is determined by the xml variable `RUN_TYPE` in `env_run.xml`. The three options are "startup","branch" and "hybrid". These have the following characteristics:

- **startup**: all components are initialized using pre-defined baseline states that have been provided with CESM. So far we have only used this type of initialization
- **branch**: all components are initialized using the complete set of restart files from a previous run. You specify the name of the previous run you want to start from and the date you want to start from with the xml variables `RUN_REFCASE` and `RUN_REFDATE`. The run continues on from the date that you restart from, so you can't change the start date with a branch run. The advantage of a branch run over the hybrid run is that it reproduces exactly (bit-for-bit) the solution that would be obtained if the original simulation that is used for the initialization were simply continued (provided no other changes have been made to the model).
- **hybrid**: the simulation is initialized in a manner similar to a startup run, but instead of using the default initialization datasets, the initialization datasets are specified by the user, for example, they may be obtained from a previously performed simulation. Unlike for a branch run, the start date can be modified by the user and this method of initialization does not result in bit-for-bit similarity with the run that it is initialized from.

If you would like to change the output variables part way through a simulation, this can be done using a hybrid or a branch run.

Exercise 9: Initializing a simulation from a previous one.

Here you'll make use of restart files from a previously run simulation, to initialize your simulation using a hybrid initialization. A 1 year simulation, called b1850_initial, which uses the B1850-tutorial compset at f19_g37 resolution as already been performed and the archive directory is here

```
/home/isimpson/scratch/archive/b1850_initial/
```

Here you'll find all the restart files for 1st Jan for year 2 of this run

```
/home/isimpson/scratch/archive/b1850_initial/rest/0001-02-01-00000
```

We're going to use these restart files to initialize another simulation.

Create a new case, using the B1850-tutorial compset at f19_g37 resolution. Call it `b1850_hybrid` and put it in ∼`/cases`.

Set up the case.

The following changes to xml parameters in `env_run.xml` need to be performed to run the hybrid initialization.

```
./xmlchange RUN_TYPE=hybrid (changes to a hybrid initialization)
./xmlchange RUN_REFCASE=b1850_initial (the name of the run from which you are initializing)
./xmlchange RUN_REFDATE=0002-01-01 (sets the date of initialization - corresponds to the date of the initialization files being used)
```

Lets also output some high frequency fields so we can see the history fields in a short 5 day run. Add the following to `user_nl_cam` to output daily precipitation

```
fincl2='PRECT'
nhtfrq=0,-24
mfilt=1,1
```

Now you need to copy all the restart files from `b1850_initial` into the run directory i.e.,

```
cp /home/isimpson/scratch/archive/b1850_initial/rest/0002-01-01-00000/*
/scratch/b1850_hybrid/run
```

Build the case. Submit it. Is it running. Are the history files being created?

Note that the default is `RUN_STARTDATE=0001-01-01`. This means that even though you have initialized from January 1st year 2, the dates of your run will start from January 1st year 1. In a hybrid run, you can change `RUN_STARTDATE` to any date you like.

Exercise 9: Initializing a simulation from a previous one - *solution*

```
cd $CESMROOT/cime/scripts
./create_newcase --case ∼/cases/b1850_hybrid --compset B1850-tutorial --res f19_g37
cd ∼/cases/b1850_hybrid
./case.setup
./xmlchange RUN_TYPE=hybrid, RUN_REFCASE=b1850_initial, RUN_REFDATE=0002-01-01
```

Now copy the restart files into your run directory with the following

```
cp /home/isimpson/scratch/archive/b1850_initial/rest/0002-01-01-00000/*
/scratch/b1850_hybrid/run
```

Then build and submit

```
./case.build
./case.submit
```

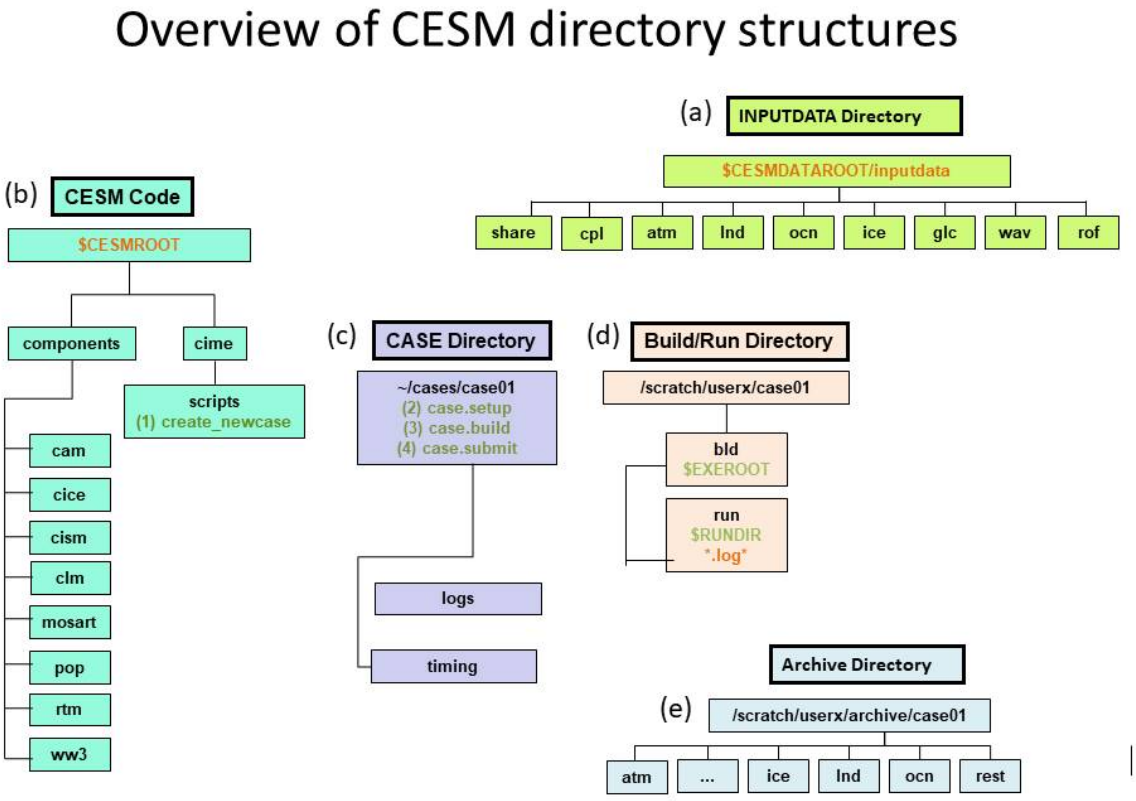

Figure 1: An overview of the CESM directory structures. (a) The input data directory. Contains all the input datasets, forcing files etc that are required to run a simulation. (b) The CESM code directory. Contains all the model source code (the components sub-directory) and the scripts needed to set up your case (the cime subdirectory). (c) The case directory. Where all the files that control your case will sit after you have run "create_newcase". (d) The Build/Run directory. Where all the compiled executable files sit (the bld subdirectory) and where the model output/log files will be put while the simulation is running (the run subdirectory). The run directory is also where any restart files that are needed for a continuation run will sit. (e) The archive directory. Where all the model history files and restart files will go when the run has completed.