CAM Breakout Session Exercises

CESM Tutorial

Aug 12, 2016

Presented by Dani Coleman

with contributions from Rich Neale

AMP, CGD, NCAR

Summary

- First run the control case
- Then choose one (or more) exercises to try

1. Turn off 'freeze-drying' approximation
   (*namelist change, Arctic*)
2. Increase orographic height over the western US by 50%
   (*dataset change, mid-latitude)*

2b. Modify sea surface temperature
   (*dataset change, tropical)*

3. Increase the triggering threshold for deep convection just over land (*code change--simple, tropical*)
4. Add a (fake) physics parameterization (*code change--advanced*)

- Compare your test exercise to your control

Control Case: Atmosphere-only

To run with prescribed ocean (observed data) and prescribed sea-ice (thickness, area), use compset type F.

Compset Type	Atmosphere	Land	Ocean	Ice
F *(these exercises)*	*Interactive*	*Interactive*	Data	Thermodynamic
B *(previous exercises)*	*Interactive*	*Interactive*	*Interactive*	*Interactive*

Create a new F-case for your control (Do this now)

```
cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/scripts
./create_newcase -case ~/cases/cesm1_2_2.cntl \
      -res T31_T31 -compset F_2000_CAM5 -mach yellowstone
```

*The casename is **your choice**; this is an example of how I like to name my cases, with as much information as possible. Feel free to use the tutorial standard but start thinking about what information you want in your case names in the future.*

Note the RES & COMPSET difference with the B-case we've been running(DON'T run this command now)

```
./create_newcase -case case01 -res T31_gx3v7 -compset B_1850_CN -mach yellowstone
```

Control Case (continued)

- **Run length options (suggested run time of 2 months will take about 10 mins wallclock)**

```
setenv CASENAME cesm1_2_2.cntl; setenv CASEROOT ~/cases/$CASENAME
cd ~/cases/$CASENAME
./xmlchange STOP_N=2,STOP_OPTION=nmonths
```

These c-shell short cuts will make your work easier. For bash, use export VAR='my value' or export VAR=$MY_OTHER_VAR

- **Configure**

```
./cesm_setup
```

- **Build**

```
./cesm1_2_2.cntl.build
```

(Continue below; edit run files in a new window while build completes)

- **Required namelist variables (for this atm resolution to use data ice model)**

Add two variables to $CASEROOT/user_nl_cice with an editor (*vi* or *emacs*).

```
grid_file = '/glade/p/cesm/cseg/inputdata/share/domains/domain.ocn.48x96_gx3v7_100114.nc'
kmt_file = '/glade/p/cesm/cseg/inputdata/share/domains/domain.ocn.48x96_gx3v7_100114.nc'
```

Note that the file names are the same; do a mouse-copy 'ls' to double check the complicated names.

The 'single quotes' are absolutely necessary.

Note that, although the configuration you've been running so far does so straight 'out of the box,' that is NOT the typical scenerio. It is often necessary to make a few modifications to get exactly what you want and it is good for you to learn how to do so.

- **The rest of the exercises are configured the same up to this point**

Do the *create_newcase*, *cesm_setup* and build just like this page with **new casenames** for each exercise. These slide titles are in gray boxes so you can search quickly through the document for them.

Control Case (continued)

- **Check completion of build**

```
cat CaseStatus
```
(should show ‘build complete’)

```
ls /glade/scratch/$LOGNAME/$CASENAME/bld/cesm.exe
```
(should exist)

- **Submit**

```
cesm1_2_2.cntl.submit
```

- **Checking jobs**

```
bjobs
```

Shows output similar to:

```
JOBID  USER    STAT QUEUE FROM_HOST   EXEC_HOST    JOB_NAME   SUBMIT_TIME
992612 bundy   PEND small yslogin1-ib              *2_0_cntl Aug 11 11:10
```

This should take about 10 minutes once it is running.

Meanwhile, continue to the following pages to choose and set up your next exercise.

You may want to read through the descriptions of all the exercises to find one that will both interest and challenge you. The namelist change is similar to the day2 practical; if you found that challenging then this is recommended, as it is the most common thing you will do with the model . The dataset exercises shows tools of interest for modifying netcdf files. The code changes are important if you'll be modifying code, and the second provides hints and tools to help you find the relevent code to change intead of just telling you what to do.

Optional: Use a script

You may skip this page

But if you are interested in using a script instead of the command line for all of this:

- It is possible to put most of the commands necessary to run CESM in a shell script instead of calling them from the command line
- Advantages of a script
  - An exact record of the commands used to make each case
  - Easy reproducibility and ability to make small changes for another case
- Disadvantages of a script
  - Necessary to understand a scripting language (example is in c-shell)
  - When the construction of a case requires editing or adding a file, you must have the script stop before the edit and resume afterward
- ***If you want to use scripts***: copy the instructor's script for the control case to your directory, open it in an editor, read and understand it and make the necessary modifications to use it for your case (including changing the casename and paths)

```
cd ~/cases
cp /glade/u/home/bundy/cases/cesm1_2_2.test.csh your_casename.csh
```

This will create a case directory as specified in the script. I recommend naming the script with the casename it will create, and placing it in the directory containing all your cases.

Since this is a more advanced method, there are no more instructions for doing it this way. But you can compare to scripts for each exercise available in `~bundy/my_scripts`.

Comparing results to control case

Once you have completed an exercise, come back to this page for a guide to a quick comparison. Again, the titles in gray boxes are to return to for each case.

You might need to load modules: `module load ncview; module load netcdf`

- **A quick comparison of two (finished!) cases:** create an ncfile containing the diffs

```
cd /glade/scratch/$LOGNAME/archive/$CASENAME/atm/hist
```

If you want to make it easier, use your case names in the following setenvs so you can copy and use the ncdiff lines)

```
setenv CNTL cesm1_2_2.cntl; setenv CASE1 cesm1_2_2.nofrzdry
```

```
ncdiff $CASE1/atm/hist/$CASE1.cam.h0.0001-01.nc
       $CNTL/atm/hist/$CNTL.cam.h0.0001-01.nc
        diff_${CASE1}_${CNTL}.nc
```

Use ncview to browse variables in this file that contains only the diffs

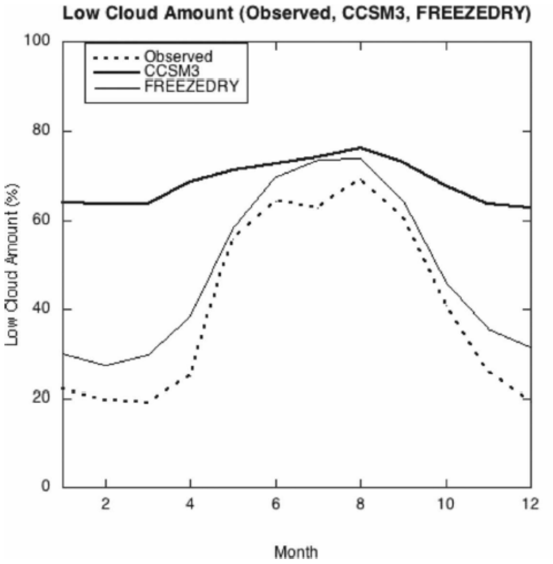
```
ncview diff_${CASE1}_${CNTL}.nc
```

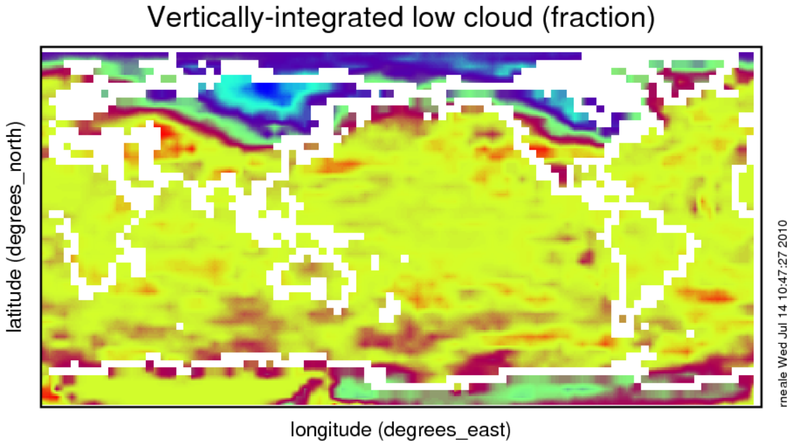
- **Use ncl scripts from the diagnostics tutorial to compare single fields**
- **Run the AMWG mean diagnostics for a more comprehensive comparison (not time to do this today, needs at least 14 months of data)**
- **Is two months enough model run time to see changes? To see meaningful changes? Is T31 good enough resolution to start with?**

Exercise 1: *Namelist Change*

(a review from previous day)

- **ACTION**: Switch off an approximation that reduces the cloud fraction in very dry atmospheric conditions (Vavrus and Walliser, 2008), which improved results from CCSM3.




Vertically-integrated low cloud (fraction)
latitude (degrees_north)
longitude (degrees_east)
meale Wed Jul 14 10:47:27 2010

```
if (cldfrc_freeze_dry) then
        f = f x max(0.15,min(1.0,q/0.003))
endif
```

Recall from day2: for a complete list of namelist, please see the on-line documentation for each component model.
CAM4: http://www.cesm.ucar.edu/cgi-bin/eaton/namelist/nldef2html-pub
CAM5: http://www.cesm.ucar.edu/cgi-bin/eaton/namelist/nldef2html-cam5_2

(Exercise 1): *Namelist Change*

- Follow **control case** example (slides 3-4) through the build, but use a new casename
- **Set namelist variable that controls 'freeze drying'** `cldfrc_freeze_dry= .false.`

Recall from day 2 you can choose one of two methods to change the namelist:

1. **env_run.xml** using xmlchange (do *before* ./cesm_setup)

```
xmlchange -file env_run.xml -id CAM_NAMELIST_OPTS -val "cldfrc_freeze_dry=.false."
```

2. (Or) add setting to **user_nl_cam** file (do *after* ./cesm_setup) using emacs/vi:

```
! Users should add all user specific namelist changes below in the form of
! namelist_var = new_namelist_value
 cldfrc_freeze_dry                 = .false.
```

- Check that it worked: is the namelist really changed?

if using method (2), call `./preview_namelist` first

```
grep freeze /glade/scratch/$LOGNAME/$CASENAME/run/atm_in
```

or compare to control namelist:

```
diff /glade/scratch/$LOGNAME/$CASENAME/run/atm_in /glade/scratch/$LOGNAME/cesm1_2_2.cntl/run/atm_in
```

You should see the difference

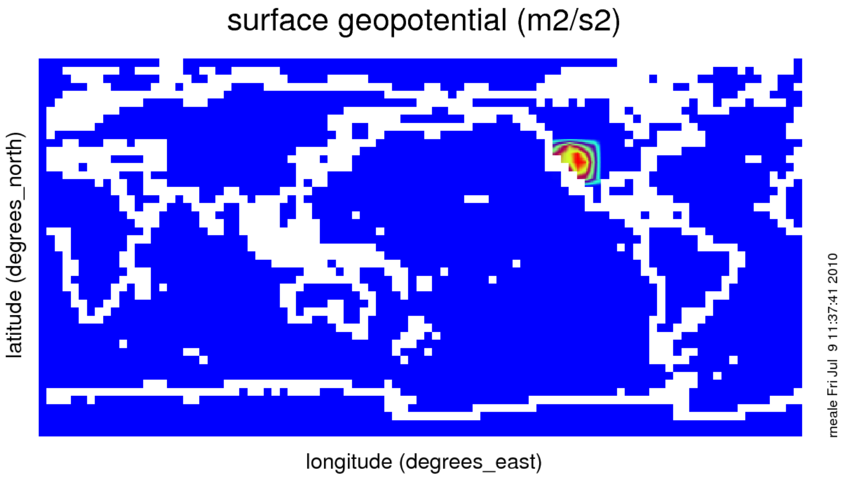
If so, submit run ***(did you change kmt file & project number)?***. While waiting, you might want to start another exercise.

What effects to look for? (See slide 7: Comparing to Control)

✓ *cloud (CLDLOW) ,Surface temperature (TS), surface energy budget (FSDS,FLNS), surface pressure (PS),...*

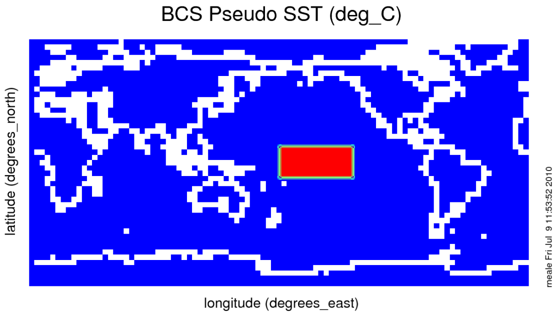
Exercise 2: *Dataset change*

- Change input boundary datasets (orography) by increasing surface geopotential height by 50% in the western USA



Range of surface geopotential: 0 to 31521.7 m2/s2

See alternative case 2b to add a Sea Surface Temperature (SST) anomaly instead



Range of BCS Pseudo SST: 0 to 1 deg_C

(Exercise 2) : *Dataset change*

1) Follow **control case** Exercise (slides 3-4) through the build but with a new casename. I'll use cesm1_2_2.oro

2) **Find Orography surface boundary dataset**

Orography file (specifies surface geopotential height **PHIS; `ncview`** if you want to look)

```
% grep -i topo ~/cases/cesm1_2_2.oro/CaseDocs/*_in
CaseDocs/atm_in: bnd_topo = '/glade/p/cesmdata/cseg/inputdata/atm/cam/topo/USGS-gtopo30_48x96_c050520.nc'
```

3) Make a local copy of the orography file in your $CASEROOT (created in step1)

```
cd $CASEROOT
cp /glade/p/cesmdata/cseg/inputdata/atm/cam/topo/USGS-gtopo30_48x96_c050520.nc .
```

(Exercise 2) : *Dataset change*

Change orography surface boundary data

3) Use nco utilities to edit values on the file (http://nco.sourgeforce.net)

We will use a function called ncap2 (netCDF Arithmetic Averager)

```
ncap2 -O -s 'lat2d[lat,lon]=lat ; lon2d[lat,lon]=lon' \   1
-s 'omask=(lat2d >= 30. && lat2d <= 50.) && (lon2d >= 235. && lon2d <= 260.)'\   2
-s 'PHIS=(PHIS*(1.+omask*0.5))'\   3
USGS-gtopo30_48x96_c050520.nc USGS-gtopo30_48x96_us_oro_x50.nc   4
```

1. *Define 2D latitude and longitude arrays*
2. *Create a mask – setting = 1 for the desired lat/lon range; elsewhere = 0*
3. *Apply mask to 2D field (PHIS = surface geopotential = gZ) – 1.5x PHIS in region of interest*
4. *Infile Outfile*

Mouse-copy all of the FOUR lines of this command into ONE line in your shell

Quickly check your changes by making a netcdf file containing the diffs (new – old)

```
ncdiff -v PHIS USGS-gtopo30_48x96_us_oro_x50.nc \
     USGS-gtopo30_48x96_c050520.nc PHIS_diff.nc
ncview PHIS_diff.nc
```

You have to click ok to make ncview check ALL of the data subsampling only finds the zeros

(Exercise 2) : *Dataset change*

Edit user namelist to point to modified dataset

```
echo "bnd_topo = './USGS-gtopo30_48x96_us_oro_x50.nc'" >>& user_nl_cam
```

Get the namelist change into the actual namelists (2nd line checks)

```
./preview_namelists
grep topo /glade/scratch/$LOGNAME/$CASENAME/run/atm_in
```

Copy changed orography boundary data to run directory

```
cp USGS-gtopo30_48x96_us_oro_x50.nc /glade/scratch/$LOGNAME/$CASENAME/run
```

Build and run. ***(Did you remember to change kmt, grid files, project # and stop_option?)***

Once it starts running, check that the correct file is being used.

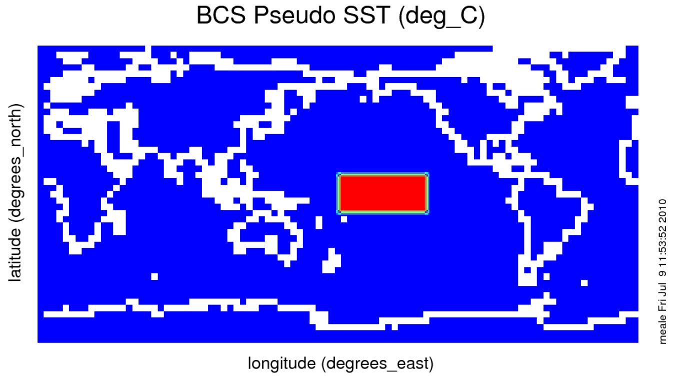
(Hint: grep logfiles for name of file)

What to look for when experiment is complete? (See slide 7: Comparing to Control)

✓ *Surface temperature (TS), surface pressure (PS), cloud (CLDLOW), rainfall (PRECT), winds (OMEGA,U,W)*

Exercise 2b: *Dataset change*

- ACTION: Change input boundary datasets (Sea Surface Temperature) by increasing it's value by 2K in the tropical Central Pacific



Range of BCS Pseudo SST: 0 to 1 deg_C

Follow **control case** (slides 3-4) through build with a new casename (eg. cesm1_2_2.sst)

Modify the SST file while the case is building:

1) Find it in the namelist

```
% grep -i sst ~/cases/cesm1_2_2.sst/CaseDocs/*_in
/glade/p/cesmdata/cseg/inputdata/atm/cam/sst/sst_HadOIBl_bc_48x96_clim_c050526.nc
```

2) Copy SST file to your case directory and follow instructions on the next page

```
setenv CASEROOT ~/cases/cesm1_2_2.sst
cd $CASEROOT
cp /glade/p/cesmdata/cseg/inputdata/atm/cam/sst/sst_HadOIBl_bc_48x96_clim_c050526.nc .
```

(Exercise 2b) : *Dataset change*

Change SST surface boundary data

Use nco utilities to edit values on the file (**http://nco.sourgeforce.net**)

We will use a function called **ncap2 – (netCDF Arithmetic Averager)** single line command below

```
ncap2 -O -s 'lat2d[lat,lon]=lat ; lon2d[lat,lon]=lon' \[1]
-s 'omask=(lat2d >= -10. && lat2d <= 10.) && (lon2d >= 180. && lon2d <= 240.)'\[2]
-s 'SST_cpl=(SST_cpl+omask*2.)'\[3]
sst_HadOIBl_bc_48x96_clim_c050526.nc sst_HadOIBl_bc_48x96_clim_warmtcp.nc[4]
```

1. *Define 2D latitude and longitude arrays*
2. *Create a mask – setting = 1 for the desired lat/lon range; elsewhere = 0*
3. *Apply mask to 2D field (SST_cpl): +2K in region of interest (equatorial Pacific)*
4. *Infile/Outfile*

Mouse-copy all of the FOUR lines of this command into ONE line in your shell

Quickly check your changes by making a netcdf file containing the diffs (new – old)

```
ncdiff -v SST_cpl sst_HadOIBl_bc_48x96_clim_warmtcp.nc \
   sst_HadOIBl_bc_48x96_clim_c050526.nc SST_cpl_diff.nc
ncview SST_cpl_diff.nc
```

You should see a difference!

(Exercise 2b) : *Dataset change*

Apply changed SST using `env_run.xml`

(recommended over user_nl* since also present in* dococn.streams.txt.prescribed *)

- ***figure out which namelist variable to change*** ***('grep' is how you would have found the sst file to change)***

```
      cd $CASEROOT
grep sst_ *.xml
    env_run.xml:<entry id="SSTICE_DATA_FILENAME"       value="$DIN_LOC_ROOT/atm/cam/sst/
    sst_HadOIBl_bc_48x96_clim_c050526.nc"  />
```

- ***change it* in env_run.xml**

```
./xmlchange -file env_run.xml -id SSTICE_DATA_FILENAME \
  -val "./sst_HadOIBl_bc_48x96_clim_warmtcp.nc"
```

- ***Process* env_run.xml to make namelist files**

```
./preview_namelists
```

Check* *your changes went into namelists *Hint:* grep sst_ $CASEROOT/CaseDocs/*

- ***Copy* changed SST data to run directory**

```
cp sst_HadOIBl_bc_48x96_clim_warmtcp.nc /glade/scratch/$LOGNAME/$CASENAME/run
```

- ***Submit* your run**

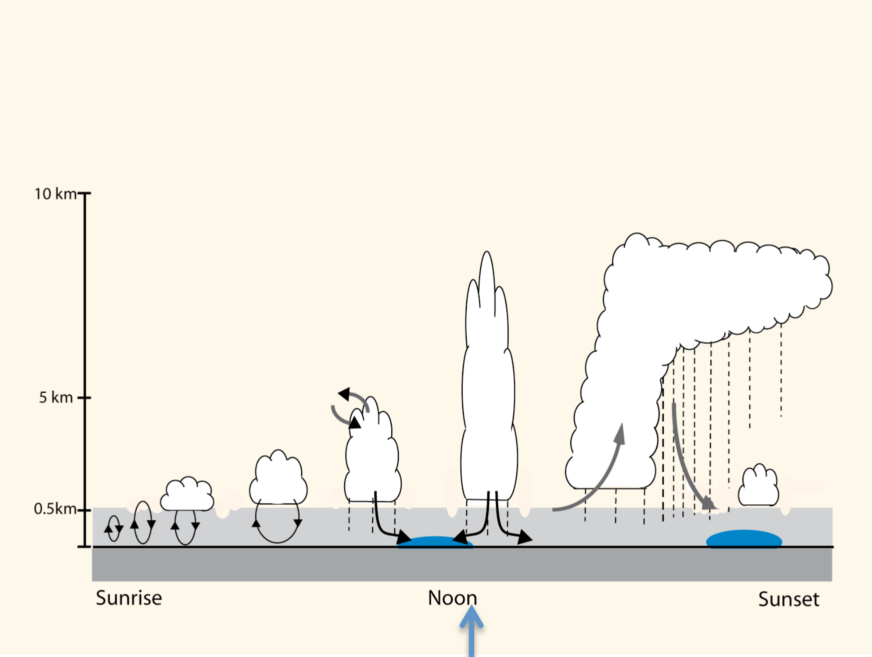
(While running, *check* *it's really using the file.* *Hint*: cd $RUNDIR; grep sst_Had cesm.log.*

What to look for in results? (See slide 7: Comparing to Control)

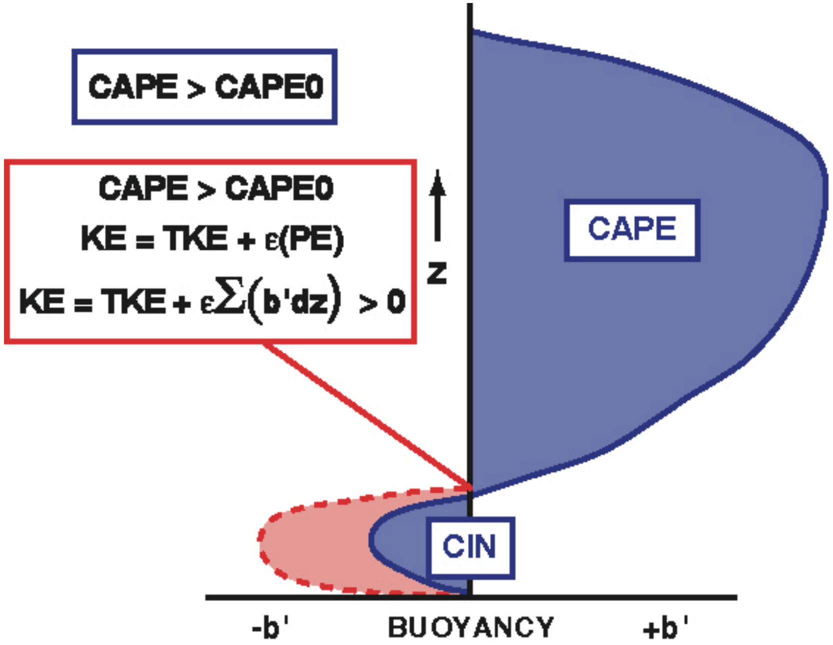
✓ *Surface temperature (TS), surface pressure (PS), cloud (CLDLOW), rainfall (PRECT), winds (OMEGA,U,W)*

Exercise 3: *Code Change CAPE*

- Examine the mean effects of delaying the initiation of convection by increasing the minimum required convective available potential energy (CAPE) to initiate convection over land;



Models peak around noon
Too early



(Exercise 3) : *Code Change CAPE*

- **Follow guide for control case (slides 3-4) through build with new casename**
- **Copy fortran code file that controls deep convection calculation (Zhang and McFarlane, 1995) to local code modification directory for the atmosphere**

```
cd $CASEROOT/SourceMods/src.cam
cp /glade/p/cesm/tutorial/cesm1_2_2.tutorial/models/atm/cam/src/physics/cam/zm_conv.F90 .
```

- **Edit** `zm_conv.F90:`

Search for the following lines (around Line 558 in file)

```
    if (cape(i) > capelmt) then
     lengath = lengath + 1
     index(lengath) = i
    end if
```

The minimum CAPE (capelmt) for triggering convection is = 70 J/kg everywhere.

Let's increase the value *over land* x10 (to 700 K/kg) to get a big signal in order to analyze what the mean effect of delaying convection is.

(Exercise 3) : *Code Change CAPE*

Change the following in sub-routine ***zm_convr***

```
    if (cape(i) > capelmt) then
     lengath = lengath + 1
     index(lengath) = i
    end if
```

To

```
    if (landfrac(i) > 0.5_r8) then
     capelmt_mask = 10._r8*capelmt
    else
     capelmt_mask = capelmt
    end if

    if (cape(i) > capelmt_mask) then
     lengath = lengath + 1
     index(lengath) = i
    end if
```

Where land fraction is >50%,

Scale the existing trigger value by 10x

Create a local variable called capelmt_mask to use as the convection trigger

And, near the top of subroutine zm_convr, right after:

```
     real(r8) pblt(pcols)
```

Add

```
     real(r8) :: capelmt_mask
```

Declare the capelmt_mask variable

Check your changes by 'diff new old' files, Start build

(Exercise 3) : *Code Change CAPE*

To check that build was succesful:

- stdout/err from build command
- CaseStatus file in CASEROOT should show 'build complete'

To check that **new code was incorporated into build**, check time stamp on object files. Assuming you started the build before, the modified routine (and those affected by it) should be newer than most of the object files *and* the executable should show the newest time stamp, too.

```
ls -ltr /glade/scratch/$LOGNAME/$CASENAME/bld/atm/obj
... [more files above] ...
-rw-r--r-- 1 bundy ncar  10504 Jul 25 14:47 clubb_intr.o
-rw-r--r-- 1 bundy ncar 269592 Jul 25 14:47 micro_mg_cam.o
-rw-r--r-- 1 bundy ncar 107487 Jul 25 14:47 microp_driver.mod
-rw-r--r-- 1 bundy ncar  10416 Jul 25 14:47 microp_driver.o
-rw-r--r-- 1 bundy ncar    883 Jul 25 16:08 tmp_filepath
-rw-r--r-- 1 bundy ncar  31004 Jul 25 16:08 zm_conv.mod
-rw-r--r-- 1 bundy ncar 230648 Jul 25 16:08 zm_conv.o
-rw-r--r-- 1 bundy ncar  47994 Jul 25 16:08 convect_deep.mod
-rw-r--r-- 1 bundy ncar  84350 Jul 25 16:08 zm_conv_intr.mod
-rw-r--r-- 1 bundy ncar 141424 Jul 25 16:08 zm_conv_intr.o
... [a few files below] ...
```

timestamp on files shows that uwshcu are newer than most of the other files.

```
ls -l /glade/scratch/$LOGNAME/$CASENAME/bld/cesm.exe
-rwxr-xr-x 1 bundy ncar 84383206 Jul 25 16:09 cesm.exe
```

executable is newer than uwshcu files

Never assume that it's working– always find a way to check!

(Exercise 3): *Code Change CAPE*

- **Submit job. When done:**

What to look for ? (Slide 7: Comparing to control)

✓ *Surface temperature (TS), surface fluxes (LHFLX,SHFLX), cloud (CLDLOW), rainfall (PRECT) over land*

✓ *How does it affect the diurnal cycle? (if feeling brave output hourly rainfall (PRECT) and surface temperature (TS) over a limited land region*

✓ *Is 2 months long enough to get a good signal?*

Exercise 4 *Code Change*

Add a parameterization

Exercise 4 is a more general, free-form exercise to prepare you to do major modifications or replace an existing parameterization.

This exercise will provide guidance if you want to **add a parameterization** to CAM, including these topics:

a) The requirements for a parameterization and interface

b) Finding source code (reading documentation, browsing and smart searching)

c) A 'stub' parameterization to add to CAM (the subroutines & calls without content code)

d) References to CAM physics code details

First: Create, setup and build a new case following the control case instructions (slides 3-4) but with a unique name, eg. *cesm1_2_2.param*
We will re-build after modifying the source code, meanwhile using some files created by the build to navigate through the source code.

(Exercise 4) *Code Change*

Parameterization requirements

A physics parameterization

1. Must calculate a tendency (rate of change)
2. Must not change the model state
3. Must conserve vertical integrals of
   - mass
   - momentum
   - total energy
   - dry static energy

The **tendencies** from each physics parameterizations are used in `physpkg.F90` (subroutines `tphysbc` and `tphysac)` to calculate the new model **state**, along with checks that energy and water balance.

eg. in /glade/p/cesm/tutorial/cesm1_2_2.tutorial/models/atm/cam/src/physics/cam/physpkg.F90

`call convect_shallow_tend(…)` is followed by the update to the state:

```
call physics_update(state, ptend, ztodt, tend)
```

(Exercise 4) *Code Change*

Interfacing recommendations

Any parameterization should consist of two parts:

1. An **interface** layer to communicate between CAM and the parameterization
2. The **parameterization** package, with as little of CAM structures in it as possible

This helps make a parameterization which is portable between different models or model versions.

(Exercise 4): *Code Change*

Routines that every parameterization interface might need

The public methods of a CAM interface

(eg. `PARAM` means the name of the parameterization, like `convect_shallow`)

`PARAM_register`

Register fields in the physics buffer, register constituents in the constituent arrays.

`PARAM_init`

Package specific initialization at beginning of run, including setting time-invariant variables

`PARAM_timestep_init`

Per-timestep initialization, (e.g. time interpolation from a boundary dataset.

`PARAM_timestep_tend`

Calls the package run method which computes the tendencies for each model timestep.

And methods for parameterizations that introduce constituents:

`PARAM_init_cnst , PARAM_implements_cnst`

See online document for details about interfaces, along with other useful CAM physics info
http://www.cesm.ucar.edu/models/atm-cam/docs/phys-interface/ (Utility Modules section)
or browse code for examples.

(Exercise 4): *Code Change*

CAM example of interface & parameterization: ***initialization***

Interface = models/atm/cam/src/physics/cam/convect_shallow.F90 methods call:
Parameterization/ package: uwshcu.F90 (University of Washington shallow convection scheme)

1. Registration (allocates memory at beginning of model run)

```
physpkg.F9:phys_register()
    call convect_shallow_register
    convect_shallow.F90:convect_shallow_register()
         call uwshcu_register
```

2. Initialization (done once at the beginning of model run or restart)

```
physpkg.F90:phys_init()
      call convect_shallow_init
      convect_shallow.F90:convect_shallow_init()
         call uwshcu_init
```

`physpkg` doesn't know anything about the shallow convection package, so a new parameterization could be swapped in with modifications only to the **interface** `convect_shallow.F90`. Also, changes in CAM model shouldn't touch `uwshcu.F90`; just `convect_shallow.F90`.

(Exercise 4): *Code Change*

CAM example of interface & parameterization:

time-stepping

3. Time-stepping (as the model is running).
`tphysbc` gets a tendency `ptend` from an interface method (1), updates the model `state` with that tendency (2), and checks conservation (3).

```
physpkg.F90:phys_run1()
1. call convect_shallow_tend(state, ptend,…)
     convect_shallow.F90:subroutine convect_shallow_tend(state,
ptend,…)
       intent(in) state
       intent(out) ptend

       call compute_uwshcu_inv(… state   ,ptend …

2. call physics_update(state, ptend,…)
3. call check_energy_chng
```

`Intents` show that the interface can only modify ptend, not state. The `ptend` is returned from the parameterization method

(Exercise 4): *Code Change*

Finding source code

Now that you have been introduced to the requirements for a parameterization and recommendations for an interface how do you find an existing parameterization or know where to put a new one?

- ➢ If replacing an existing parameterization
  - **Read documentation** to find names of existing routines
  - **Browse** the code
  - Use **grep/find** or grepccm/findccm (described later)

  to find all the source code references

- ➢ If adding a new parameterization
  - **Browse** or use grep/find tools to find a parameterization to use as a model

- ➢ The existing parameterizations can serve as **examples to follow**, but some have better code than others!

(Exercise 4): *Code Change*

Reading documentation

If you want to replace an existing parameterization, you may need to first **learn what is already in CAM**, in order to find the correct search terms.

For instance, search the CAM5 Scientific Description for “`shallow convection`.”

This doesn't tell you much more than the name of the new scheme in CAM5 and what it replaces, but it's a start.
http://www.cesm.ucar.edu/models/cesm1.0/cam/docs/description/cam5_desc.pdf

4.3 Shallow Convection Scheme

Shallow convection scheme in CAM5 is from Park and Bretherton [2009] that is a replacemnt of Hack [1994b]…

(Exercise 4): *Code Change*

Browsing the source code

The source code is under the `$rootdir` (recall where you go to call `create_newcase`).

The source for CAM is in `$rootdir/models/atm/cam/src`
With subdirectories

```
advection/  chemistry/  control/  cpl_esmf/  cpl_mct/  cpl_share/
dynamics/   physics/    utils/
```

This exercise mostly makes changes to code in the physics/cam subdirectory but I'll point out `control/` as a useful directory, holding physical constants, interpolation routines and, indeed, the top level module for the CAM model component, `cam_comp.F90`

You can find a lot by looking through source code files. For our shallow convection example, notice there is a file called `models/atm/cam/src/physics/cam/convect_shallow.F90`
Looking at, you can see there are different convection options (

```
 ! The following namelist variable (shallow_scheme) controls which shallow conv package is used.
 !        'Hack'    = Hack shallow convection (default)
 !        'UW'      = UW shallow convection by Sungsu Park and Christopher S. Bretherton
 !        'off'     = No shallow convection
```

(Exercise 4): *Code Change-*

Optional tools to search CAM code

- UNIX commands `find` and `grep` can do the job
- However, CAM source code is
  - in a complicated directory structure *and*
  - any particular model build uses only a subset of the source code (eg. dycores)

So we have tools that use `find` and `grep` **within** this directory structure

- These tools are available with this tutorial although their use is optional;

If you want to use them in the future, copy them to your home machine before you go

`grepccm string` *searches for string in all files in the directories used by this particular model build*

`findccm filename` *searches for a file named filename in all the directories used by this model build*

- For now, copy `/glade/u/home/bundy/bin/grepccm` and make executable

```
cp ~bundy/bin/grepccm ~; chmod 755 ~/grepccm
cp ~bundy/bin/findccm ~; chmod 755 ~/findccm
```

- Change to directory containing CAM's Filepath in order to use!

```
cd /glade/scratch/$LOGNAME/$CASENAME/bld/atm/obj
ls Filepath
```

(Exercise 4): *Code Change*

Optional tools to search CAM code (details)

Filepath is a text file listing all the directories containing source code that are *actually* used in the model build.

For example (yours may look different):

```
   % cat /glade/scratch/$LOGNAME/$CASENAME/bld/atm/obj/Filepath
/glade/u/home/bundy/cam/case/tutorial/cesm1_2_2.cntl/SourceMods/src.cam
/glade/p/cesm/tutorial/cesm1_2_2.tutorial/models/atm/cam/src/chemistry/mozart
…
/glade/p/cesm/tutorial/cesm1_2_2.tutorial/models/atm/cam/src/physics/cam
/glade/p/cesm/tutorial/cesm1_2_2.tutorial/models/atm/cam/src/dynamics/eul
/glade/p/cesm/tutorial/cesm1_2_2.tutorial/models/atm/cam/src/control
/glade/p/cesm/tutorial/cesm1_2_2.tutorial/models/atm/cam/src/utils
```

You must be in a directory containing Filepath to use `grepccm` and `findccm`

- `grepccm` searches for ***a string*** in each file in each directory listed in Filepath

```
cd /glade/scratch/$LOGNAME/$CASENAME/bld/atm/obj/
~/grepccm zhang
~/grepccm "calculates cape"
```

- `findccm` searches for ***a file*** in each directory listed in Filepath

```
cd /glade/scratch/$LOGNAME/$CASENAME/bld/atm/obj/
~/findccm convect_shallow.F90
```

(Exercise 4): *Add a parameterization- Do it!*

This exercise has two steps:

1. Add a new parameterization: copy a stub parameterization to your SourceMods dir
2. Interface the new param with a stub.
3. Call the interface methods from CAM physics code
4. Build and run to test your set up.

Once you've completed these 'toy' exercises, you will be free to expand upon them or start using your own work.

(Exercise 4)

Add a new (fake) parameterization

To create a new file, you need to place valid source code in SourceMods/src.cam
You could copy existing files; to make it easier I created a ***'stub' parameterization*** and interfaced to give you a simple place to start from.

1) copy Dani's stub files to your SourceMods directories

```
% setenv $CASEROOT ~/cases/<your-case-directory-here>
% cp ~bundy/tutorial/param/{param.F90,param_interface.F90} $CASEROOT/SourceMods/src.cam
```

Look at the files to see what they contain; which subroutines will the model call?
Notice the write statements with a debug string that you will need to `grep` the logfiles for to make sure it worked.

2) Add calls to new subroutine

```
% cp ~bundy/tutorial/param/physpkg.F90 $CASEROOT/SourceMods/src.cam
```

Look at the file to see how it calls param_interface routines. Notice I just mirrored the convect_shallow calls…

3) Remove files used by the build to force the build process to notice new file

```
% cd /glade/scratch/$USER/$CASENAME/bld/atm/obj
% rm Depends
% rm Srcfiles
% cd $CASEROOT
% $CASENAME.build
```

4) Check that the new file was compiled by looking at timestamp of .mod/.o files in bld dir

```
% ls -ltr /glade/scratch/$USER/$CASENAME/bld/atm/obj/
```

You should **see** param, param_interface files and timestamp of physpkg.* should be **later** than original build.

(Exercise 4): *Add a parameterization*

Interface the new parameterization

3) Build and run (did you remember the grid/kmt commands? See slide 4)

4) Check that run was successful

```
% cd $CASEROOT
% cat CaseStatus
```

- if successful, you're still not done!

Check that the new parameterization is being called by grepping for the debug string in the log files (Hint `%gzip -d *.gz` unzips them all)

Never assume it is working, Always find a way to test!

- if failed: problem solve. `ls -ltr` in rundir, logfiles in rundir or casedir?

(Exercise 4): *Add a parameterization*

What’s next?

After completing these ‘toy’ exercises, you may want to:

- change the name of param.F90 to something more descriptive. get the model to rebuild and run successfully without starting over from scratch.
- have the new param add a field to the history file (addfld/outfld)
- do something with the physics buffer (pbuf) in the new parameterization
- write code in param.F90 to do something more. Check that it's working.
- make a modification of your choice to an existing parameterizations
- … your choice

Due to the open-ended nature of this set of exercises,
please don’t use the helper resources for your own project…
if you’ve advanced to this point,
you should start the practice of finding answers yourself. ;)

See following slides for references about the model code

References

Learn more about coding in CAM

- CAM reference manual

http://www.cesm.ucar.edu/models/cesm1.2/cam/docs/rm5_3/

It provides details of how the CAM routines are called, the data structures (state, ptend), the array dimensions (chunks and columns), and descriptions of subcolumns and radiative constituents.

- CAM coding standards

https://wiki.ucar.edu/display/ccsm/Draft+of+Coding+Standards+for+CAM

- Unit testing

http://www.cesm.ucar.edu/events/ws.2014/Presentations/SEWG/santos.pdf

- Physics interface design

http://www.cesm.ucar.edu/models/atm-cam/docs/phys-interface/

References

Stand-alone , Single-Column CAM

•Users who are developing CAM might be interested in using the stand-alone CAM configure/run instead of dealing with the entire CESM structure

```
   models/atm/cam/bld/run-yellowstone.csh  (among others)
```

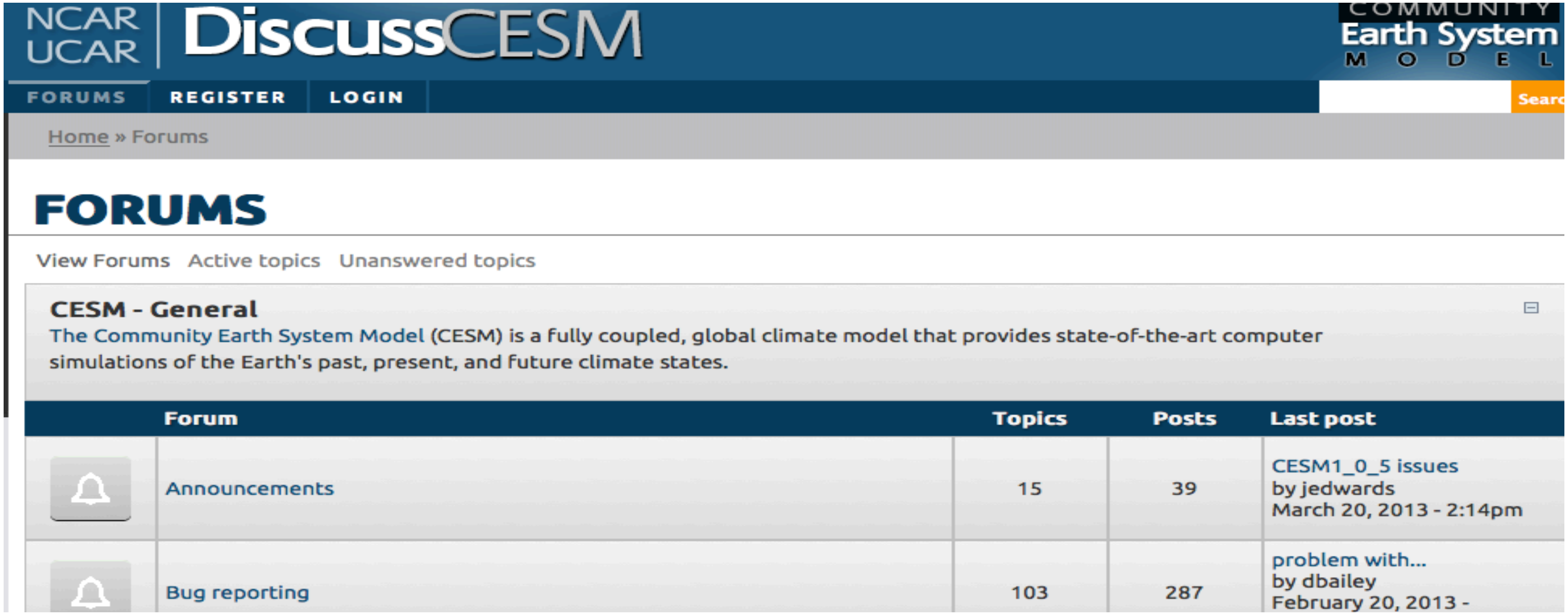
•Single-column CAM is a good tool for developing physics parameterizations

- no dynamics
- runs with field-experiment data

```
     models/atm/cam/bld/run-scam.csh
```

References

CESM bulletin board



Home » Forums

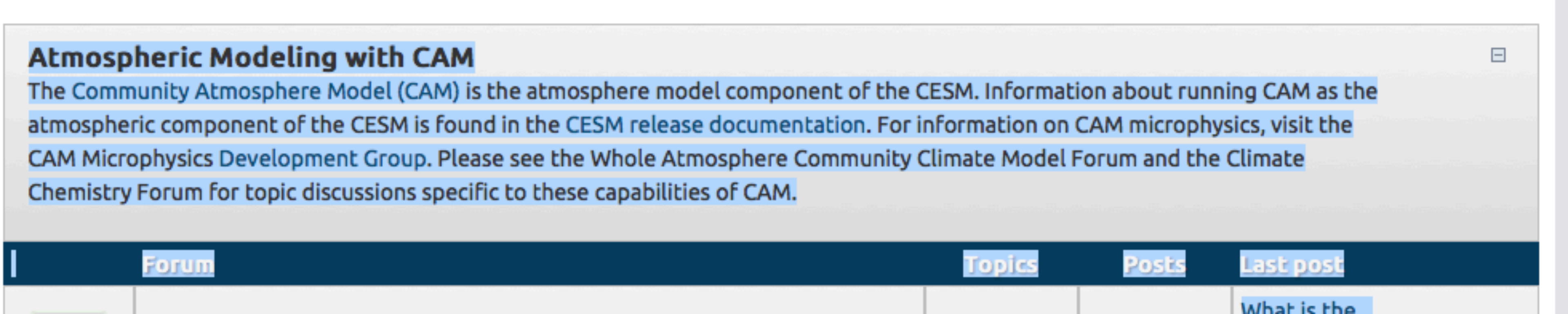
FORUMS

View Forums Active topics Unanswered topics

CESM - General

The Community Earth System Model (CESM) is a fully coupled, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states.

Forum	Topics	Posts	Last post
Announcements	15	39	CESM1_0_5 issues by jedwards March 20, 2013 - 2:14pm
Bug reporting	103	287	problem with... by dbailey February 20, 2013 -



Atmospheric Modeling with CAM

The Community Atmosphere Model (CAM) is the atmosphere model component of the CESM. Information about running CAM as the atmospheric component of the CESM is found in the CESM release documentation. For information on CAM microphysics, visit the CAM Microphysics Development Group. Please see the Whole Atmosphere Community Climate Model Forum and the Climate Chemistry Forum for topic discussions specific to these capabilities of CAM.

Forum	Topics	Posts	Last post
			What is the

Answers

Basic commands for all cases (using my casename cesm1_2_2.cntl)

```
cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/scripts
./create_newcase -case ~/cases/cesm1_2_2.cntl -mach yellowstone -res T31_T31 -compset F_2000_CAM5

cd ~/cases/cesm1_2_2.cntl
./cesm_setup
./cesm1_2_2.cntl.build  >& ! out.build

./xmlchange -file env_run.xml -id STOP_N -val 2
./xmlchange -file env_run.xml -id STOP_OPTION -val nmonths

echo "grid_file = '/glade/p/cesm/cseg//inputdata/share/domains/domain.ocn.48x96_gx3v7_100114.nc'" >>&user_nl_cice
echo "kmt_file = '/glade/p/cesm/cseg//inputdata/share/domains/domain.ocn.48x96_gx3v7_100114.nc'" >>&user_nl_cice
```

Make any dataset changes or code changes here.

```
./cesm1_2_2.cntl.submit
```

Likely run-time errors

- No hist files in archive directory and only 5 days in rundir ->**stop_n , stop_option not changed**
- Run quit with an error relating to ice model grid (or no description)-> **forget to set grid_file, kmt_file in user_nl_cice**

Exercise 4 errors

May need to delete /glade/scratch/$USER/$CASENAME/bld/atm/* to get it to work, especially if there were previous build errors