SCAM Practical Session Introduction

Andrew Gettelman, John Truesdale, Brian Dobbins
Goals

- Get up and running with SCAM
- Explain how SCAM is run
  - Same methods / steps as CESM
  - Secret: SCAM is actual SCESM
    - (a Single Column Earth System Model)
- Basic model output and visualization
- Start some exercises with SCAM
What the heck is this?
Your Environment

Your Computer

Container: ‘Virtual Machine’

Choose Method for Control/Communication

Jupyter Lab IDE webserver

Link (‘Bind’)

container file system to local
Software Stack

- Docker = Virtualization layer
- Docker Container = Virtual Machine
  - Linux
  - Full CESM2 with libraries, compilers, etc
    - Configured for SCAM, with input data
  - Python (visualization)
  - Jupyter Lab = Integrated Development Environment (IDE)
    - Web server interface running in the container
    - GUI for controlling things
Workflow

- One time: Install Docker, ‘Load’ Container
- Run container (virtual linux machine)
  - Bind ‘work’ to local directory
  - Suggest launching Jupyter Lab IDE
- In container, through Jupyter
  - Terminal: Run SCAM script: build, compile, run
  - Terminal: run python plotting script
  - Notebook: interactive visualization
- Exercises: change the model, re-run, look at output
- ‘Stop’ container (or just leave it running)
Run SCAM...

- Get a terminal in Jupyter Lab
- Are you set up? (copy script to work directory)
- `./create_scam_iop`
  - Off you go: build, compile run
- Result will be a new ‘case’
  - cases directory
  - Output file:
    work/cases/tutorial.FSCAM.arm97/run/*\.nc
What does the SCAM script do?

- Paths: model code, ‘case’ and ‘run directories
- Set Case Name
- Location of ‘source mods’
- Run configuration (=settings), ‘compset’, IOP for SCAM
- Create case (create_newcase): sets up the case
- Changing case options: xmlchange
  - CAM_CONFIG options
- Setup case & Copy source mods.
- Namelist changes
- Build (compile) the model: case.build
- Run!
# Run SCAM with a single IOP
# Usage:
# ./create_scam6_iop <IOP>    # where IOP name is from list below
#   - or -
# ./create_scam6_iop          # IOP is specified in the script below
#******************************************************************************

# User sets options in this section

### Full path of cesm source code and case (output) directories (see examples)

### Case Name

Change case name every time you run: script adds compset and IOP to casename

### Set location of user source mods (if any)

setenv usrsr $this_dir/mods/$CASETITLE

### Standard Run Settings

set COMPSET=FSCAM

### Set Desired IOP

arm95 arm97 atex bomex cgilsS11 cgilsS12 cgilsS6 dycomsRF01 dycomsRF02
gateIII mpace rico sparticus togaII twp06

# create case

$CESMDIR/cime/scripts/create_newcase --compset $COMPSET --res $RES --compiler
$COMPILER --case $CASEDIR/$CASENAME --user-mods-dir $MODSDIR/$IOPNAME
--run-unsupported --mach ncar-scam-container

This says run SCAM. ‘BHIST’ will give you a fully coupled CESM2!

This specifies locations, times, input files. SCAM specific

This ‘sets up’ the model case. Note $COMPSET = SCAM
Also: ${IOPNAME} loads specific dates, times, etc.
Once the case is ‘set up’ with defaults (FSCAM) you can change some things this way.

An example of changing the model: altering CAM before compiling with ‘configure’

### Append to CAM configure options

```bash
# ./xmlchange --append CAM_CONFIG_OPTS=' '
```

Set up cesm configuration options

If you change code, this copies it to where the model can compile it

```bash
/bin/cp -f ${usrsrc}/*/SourceMods/src.cam/
```

This is where you can modify the post-compile run-time namelist. It controls output fields

Build & Compile code, make namelists

```bash
cat >> user_nl_cam << EOF
    fincl1= 'CDNUMC', 'AQSNOW','ANSNOW','FREQSL','LS_FLXPRC'
EOF
```

Run the model

```bash
../bld/cesm.exe
```
Suggested Workflow

- Make a `create_scams6_iop` script for each case.
- New case name each time
  - IOP is added to case name in the script.
- Option: copy the script each time you change something and call it `create_scams6_iop_${CASE}`
  - Then you remember what you did.
- Also, script is set up to have multiple directories for code modifications for each case in `./mods/${CASE}`
  - Important to track changes!
- Save cases with output for analysis in same location (`work/${CASE}/run`)
- Same workflow works for full CESM as well...
- Directory locations are different for full CESM
  - `run` directory is not under case directory.
Visualization with Jupyter

- Browser based Interactive Development Environment (IDE) = Web server running in container
- Runs a terminal and ‘jupyter notebooks’ (python)
- Currently points to test data, change paths for new runs

Filesystem tree
- Tabs for: Terminal window, Python Notebooks, Display images (PDF), Script editor

Tab shown: Jupyter Notebook interactive python visualization of 2 SCAM runs
How to modify CAM?

Four basic ways to modify SCAM runs.
Different changes require different methods
Goal is to show you all of them
1. Run Settings: Compset, IOP
2. CESM configuration changes
   a. CAM Configuration options (compile time)
3. Namelist settings: Output, input & ‘parameters’
4. Modified Source Code
Changing CAM

- Where you change something may not be logical
- Some things have to be done in order:
  - Configuration changes before model setup
  - Namelist changes before constructing namelists
  - Model code changes before compiling
- Careful with where to change things
  - Some parameters can be changed through the namelist, others require code modifications
  - Some parameterizations can be switched in the namelist, some cannot
- Sometimes things break
  - You can modify something that is overwritten!
  - Configuration changes can have ‘knock on effects’
Changing CAM: IOP

- Run a different location, time
- Different IOPs good for different questions

---

**Table 1**

*List of Single Column Atmosphere Model Intensive Observation Period Cases*

<table>
<thead>
<tr>
<th>Name</th>
<th>Long name</th>
<th>Lat</th>
<th>Lon</th>
<th>Date</th>
<th>Length</th>
<th>Reference</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>arm95</td>
<td>ARM Southern Great Plains</td>
<td>36</td>
<td>263</td>
<td>Jul 1995</td>
<td>18</td>
<td>M. Zhang et al. (2016)</td>
<td>Land convection</td>
</tr>
<tr>
<td>arm97</td>
<td>ARM Southern Great Plains</td>
<td>36</td>
<td>263</td>
<td>Jun 1995</td>
<td>30</td>
<td>M. Zhang et al. (2016)</td>
<td>Land convection</td>
</tr>
<tr>
<td>atex</td>
<td>Atlantic Trade Wind Exp</td>
<td>15</td>
<td>345</td>
<td>Feb 1969</td>
<td>2</td>
<td>Augstein et al. (1973)</td>
<td>Shallow cumulus</td>
</tr>
<tr>
<td>bhomex</td>
<td>Barbados Ocean and Met Exp</td>
<td>15</td>
<td>300</td>
<td>Jun 1969</td>
<td>5</td>
<td>Holland and Rasmusson (1973)</td>
<td>Shallow cumulus</td>
</tr>
<tr>
<td>cgilsS6</td>
<td>CFMIP-GASS SCM/LES Intercomp</td>
<td>17</td>
<td>211</td>
<td>Jul 1997</td>
<td>30</td>
<td>M. Zhang et al. (2013)</td>
<td>Shallow cumulus</td>
</tr>
<tr>
<td>dycomesRF02</td>
<td>Dynamics of Marine StratoCu</td>
<td>32</td>
<td>239</td>
<td>Jul 11 2001</td>
<td>2</td>
<td>Stevens et al. (2003)</td>
<td>Stratocumulus</td>
</tr>
<tr>
<td>dycomesRF01</td>
<td>Dynamics of Marine StratoCu</td>
<td>32</td>
<td>239</td>
<td>Jul 15 2001</td>
<td>2</td>
<td>Stevens et al. (2003)</td>
<td>Stratocumulus</td>
</tr>
<tr>
<td>gateIII</td>
<td>GATE Phase III</td>
<td>9</td>
<td>336</td>
<td>Aug 1974</td>
<td>20</td>
<td>Thompson et al. (1979)</td>
<td>Tropical convection</td>
</tr>
<tr>
<td>rico</td>
<td>Rain and Cumulus over Oceans</td>
<td>18</td>
<td>299</td>
<td>Dec 2004</td>
<td>3</td>
<td>Rauber et al. (2007)</td>
<td>Shallow cumulus</td>
</tr>
<tr>
<td>sparticus</td>
<td>Small Particles in Cirrus</td>
<td>37</td>
<td>263</td>
<td>Apr 2010</td>
<td>30</td>
<td>Mace et al. (2009)</td>
<td>Cirrus, convection</td>
</tr>
<tr>
<td>twp06</td>
<td>Tropical W. Pacific Convection</td>
<td>-12</td>
<td>131</td>
<td>Jan 2006</td>
<td>26</td>
<td>May et al. (2008)</td>
<td>Tropical convection</td>
</tr>
</tbody>
</table>

*Note.* Length is given in days. ARM = Atmospheric Radiation Measurement; GASS = Global Atmospheric System Studies; SCM = Single Column Model; LES = Large Eddy Simulation
Changing CAM: CESM Options

- CESM uses xml files to define configurations
- Includes fundamental cam configurations

```bash
### Append to CAM configure options
# ./xmlchange --append CAM_CONFIG_OPTS='-micropys mg1'

### DEBUG
./xmlchange DEBUG='TRUE'

### Use a different SST file (SST+4K)
./xmlchange SSTICE_DATA_FILENAME="/home/scam/work/sst_HadOIBl_bc_1x1_2000climoP4K_c180814.nc"

Warning, this is not in the v1.0 container: you will have to add the file
```
Changing CAM: Namelist Options

- Lots of control options here
- Complete List:
  http://www.cesm.ucar.edu/models/cesm2/settings/current/cam_nml.html
- Most common: output ‘history’ fields
A word about Output

Discussion of history fields and SCAM output
List of CAM6 history fields in the user guide section 7.6:


Master field list:

CAM Model I/O: History Fields

- Standard output of the coupled model
  - Allowable I/O is called a ‘history field’
  - Possible fields vary by component set

- CAM
  - Outputs a history file of fields determined by the Default Field List
    - Plus user additions. Default is h0 = monthly mean
    - Other history streams (h1-h9) are possible with different frequency (there is a standard, h1 = daily, h2= 6-hrly, etc)
  - Add fields in namelist namelist variable can add fields from the master list to any hist file:
    
    \texttt{finclN = fields to include} (in addition to defaults) to file \#(N-1), where N=1,10 (so fincl1=h0, fincl2=h1, etc)
    eg. \texttt{fincl1 = 'U850', 'U200'}
    adds zonal wind at 850 & 200 mb to the h0 file
Code modifications

- Script has multiple directories for code modifications for each case in ./mods/$CASE
  - Important to track changes!
- To change code, copy code from CESM code directories (/opt/ncar/cesm2) into ./mods/$CASE
- CAM physics code:
  ```bash
  ls /opt/ncar/cesm2/components/cam/src/physics/cam
  ```
Goals

- Play with SCAM using different methods
- **Session #1 (now)**
  - Basic modifications, different types
  - Basic visualization
- **Session #2 (tomorrow)**
  - Other parameterizations, combinations
- **Session #3 (wed AM)**
  - Design your own experiment (with help)
  - Report on what you learned
Okay, let’s do some exercises


Set 1

1. Run
2. Visualize
3. Different case
4. Change Output fields
5. Namelists: switch parameterizations
6. Modify code
7. Namelist ‘tuning’ parameters
Exercises

Set 2

1. Change physics with configure (MG1)
2. CLUBB Parameters (Optional)
3. Input data: SST forcing (Cloud Feedback)
4. MG2 parameters (optional)
Exercises

Set 3

1. Increase CO2 (namelist)
2. Stop the Earth (code modification)
3. Aerosol Radiative Forcing (namelist)
4. Explore your own
More about Output
History File Controls

• Time sample frequency
  \texttt{nhtfrq} - how frequently to write data to each history file
  If \texttt{nhtfrq}(i) > 0, frequency is specified as number of timesteps
  If \texttt{nhtfrq}(i) < 0, frequency is specified as number of hours.
  Only the first file series may be a monthly average [default], with \texttt{nhtfrq}(1) = 0

• Number of time samples per file
  \texttt{mfilt} - the maximum number of times to output into each file

• Example
  \texttt{fincl2} = 'T:I','Q:I','U:I','V:I'
  \texttt{fincl3} = 'T','Q','U','V'
  \texttt{nhtfrq} = 0,-24,-3
  \texttt{mfilt} = 1,31,8
  h1 file will have 31 timesamples (approx 1 month) of daily instantaneous fields T,Q,U,V
  h2 file will have 8 timesamples (1 day) of 3 hourly averaged fields T,Q,U,V
Other CAM History File Controls

User Guide (7. Model Output)


Provides settings/links to control output in a general way/for specific purposes:

- **empty_htapes** - turn off all default output and only write out the fields explicitly set via fincl settings
- **history_*** - ‘groups’ of variables. Add fields for specific purposes to the default output.
  - For the complete listing go to the namelist page and search for namelist variables with the history_ prefix (i.e. history_amwg, history_clubb, history_cosp, etc.)
- **finclNlatlon** = single point output (fincl1= '10e_15n')
  - Can also use this for regional output (fincl1='10e:20e_15n:20n')

See Namelist Variables for Full information:
http://www.cesm.ucar.edu/models/cesm2/settings/current/cam_nml.html
Advanced: Adding a variable for output

• This gets a little complicated.
• But you can output pretty much any array from CAM.
• Complication: fields in parameterizations need to be passed out to model ‘interface’ layer
• Best is to find something similar and copy the method.
  • Look in the *__intr.F90 modules....

1) [Fortran *__intr.F90] addfld: at model initialization, ‘registers’ the history field
2) [Fortran *__intr.F90] outfld: each model timestep, stores values for output
3) [Namelist] finclN: output something during a run