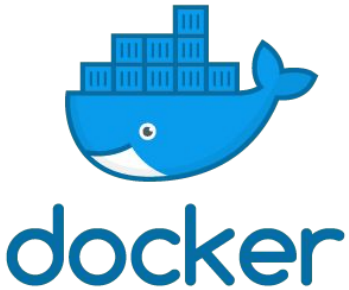
SCAM Practical Session Introduction

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

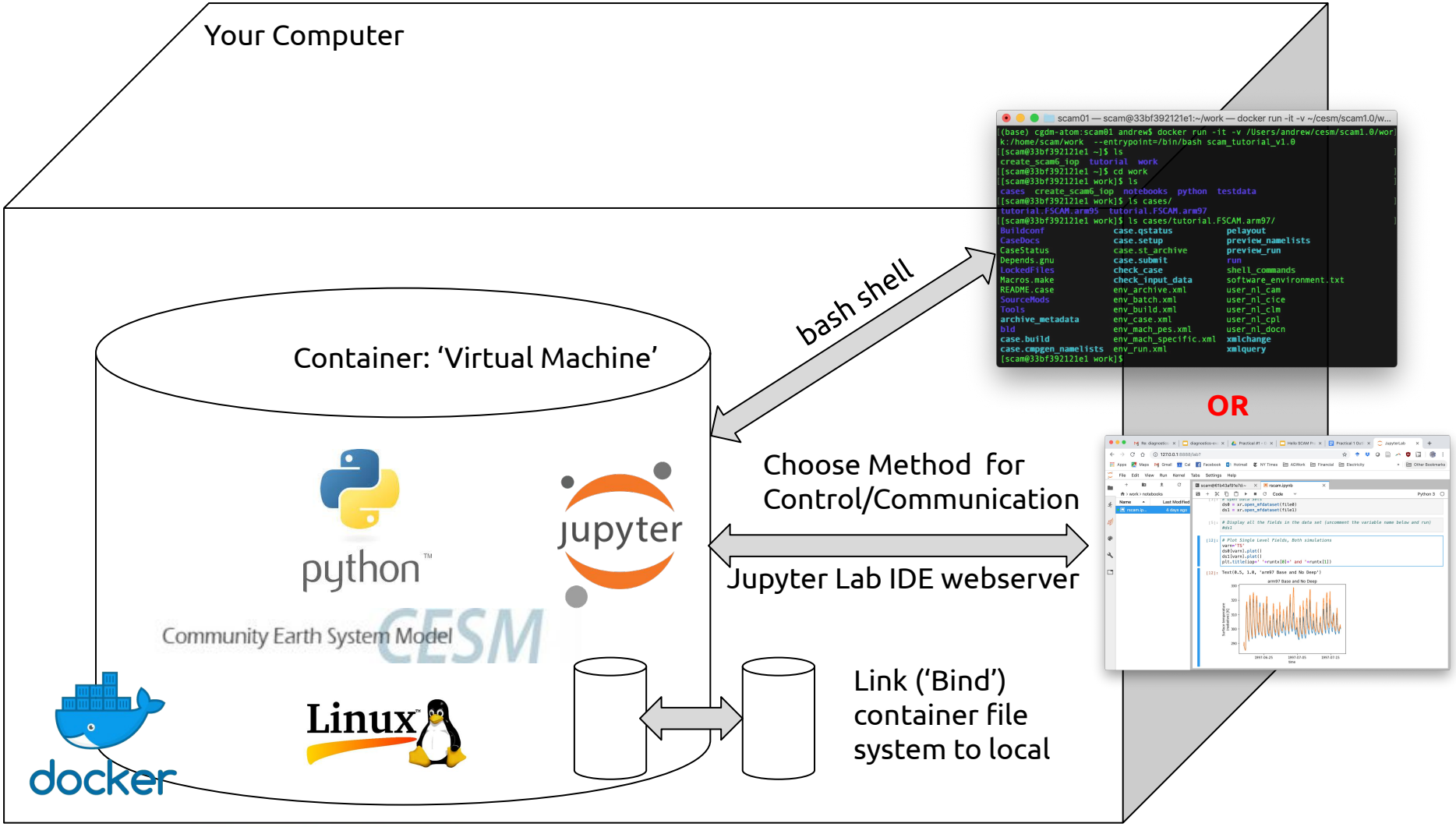
What the heck is this?



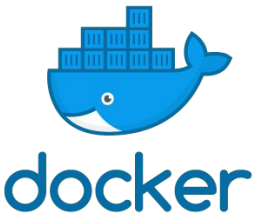
Community Earth System Model CESM

python™

Your Environment



Software Stack



- Docker = Virtualization layer
- Docker Container = Virtual Machine

  - Linux
  - Full CESM2 with libraries, compilers, etc
    - Configured for SCAM, with input data

  - Python (visualization)

Community Earth System Model CESM

  - 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!

`create_scam6_iop` walkthrough Part 1

```
#*********************************************************************
# 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

### Set location of user source mods (if any)
setenv usrsrc  ${this_dir}/mods/$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
```

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

If you change code, this guides where it will go.

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.

`create_scam6_iop` walkthrough Part 2

```
#------------------
# XMLCHANGE OPTIONS HERE

### Append to CAM configure options
# ./xmlchange --append CAM_CONFIG_OPTS=' '

#------------------
# Setup Case
./case.setup

#------------------
#  source mods: copy them into case directory

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

#------------------
# Add all user specific cam namelist changes here

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

#------------------
# Build

./case.build

#------------------
# Run

../bld/cesm.exe
```

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'

Set up cesm configuration options

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

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

Build & Compile code, make namelists

Run the model

Suggested Workflow

- Make a `create_scam6_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_scam6_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

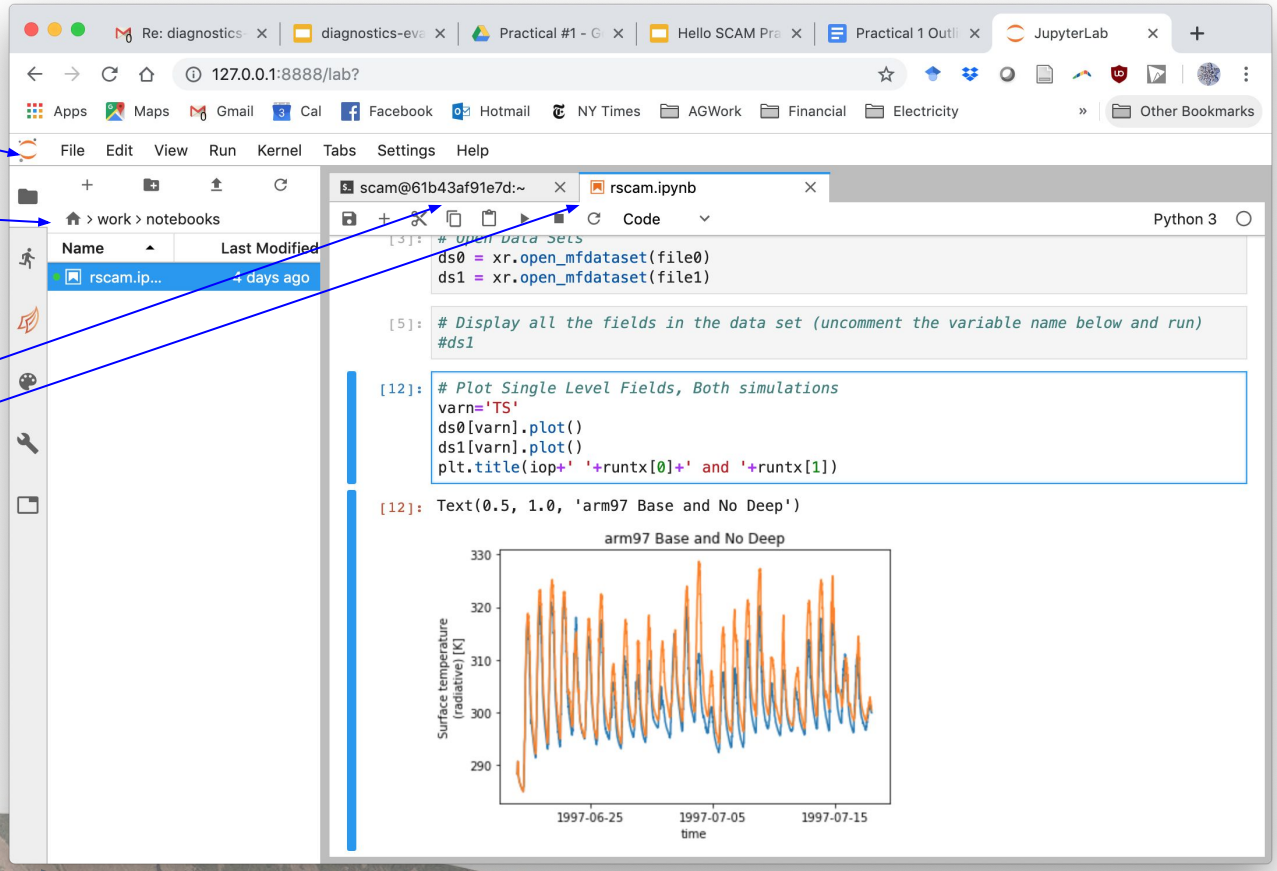
Jupyter Menus

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 thing 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

Name	Long name	Lat	Lon	Date	Length	Reference	Type
arm95	ARM Southern Great Plains	36	263	Jul 1995	18	M. Zhang et al. (2016)	Land convection
arm97	ARM Southern Great Plains	36	263	Jun 1997	30	M. Zhang et al. (2016)	Land convection
atex	Atlantic Trade Wind Exp	15	345	Feb 1969	2	Augstein et al. (1973)	Shallow cumulus
bomex	Barbados Ocean and Met Exp	15	300	Jun 1969	5	Holland and Rasmusson (1973)	Shallow cumulus
cgilsS12	CFMIP-GASS SCM/LES Intercomp	35	235	Jul 1997	30	M. Zhang et al. (2013)	Stratus
cgilsS11	CFMIP-GASS SCM/LES Intercomp	32	231	Jul 1997	30	M. Zhang et al. (2013)	Stratocumulus
cgilsS6	CFMIP-GASS SCM/LES Intercomp	17	211	Jul 1997	30	M. Zhang et al. (2013)	Shallow cumulus
dycomsRF02	Dynamics of Marine StratoCu	32	239	Jul 11 2001	2	Stevens et al. (2003)	Stratocumulus
dycomsRF01	Dynamics of Marine StratoCu	32	239	Jul 15 2001	2	Stevens et al. (2003)	Stratocumulus
gateIII	GATE Phase III	9	336	Aug 1974	20	Thompson et al. (1979)	Tropical convection
mpace	Mixed Phase Arctic Clouds Exp	71	206	Oct 2004	17	Verlinde et al. (2007)	Arctic
rico	Rain and Cumulus over Oceans	18	299	Dec 2004	3	Rauber et al. (2007)	Shallow cumulus
sparticus	Small Particles in Cirrus	37	263	Apr 2010	30	Mace et al. (2009)	Cirrus, convection
twp06	Tropical W. Pacific Convection	-12	131	Jan 2006	26	May et al. (2008)	Tropical convection
togaII	Tropical Ocean Global Atmosphere	-2	154	Dec 1992	21	Webster and Lukas (1992)	Tropical convection

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

```
### 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:

https://ncar.github.io/CAM/doc/build/html/users_guide/model-output.html#example-default-history-fields-and-master-field-lists

Master field list:

http://www.cesm.ucar.edu/models/cesm2/atmosphere/docs/ug6/hist_flds_f2000.html

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:

    `finclN` = **f**ields to **incl**ude (in addition to defaults)

    to file #(N-1), where N=1,10 (so fincl1=h0, fincl2=h1, etc)

    eg. **`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:

```
>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

ftp://ftp.cgd.ucar.edu/archive/cam-tutorial/SCAM_Practicals.pdf

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

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

- Number of time samples per file

  mfilt - the maximum number of times to output into each file

- Example

```
fincl2 = 'T:I','Q:I','U:I','V:I'
fincl3 = 'T','Q','U','V'
nhtfrq =  0,-24,-3
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)

https://ncar.github.io/CAM/doc/build/html/users_guide/model-output.html?highlight=history#

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