

# Reproducing ECCO v4r3 on Texas Advanced Computing Center machines

Ou Wang

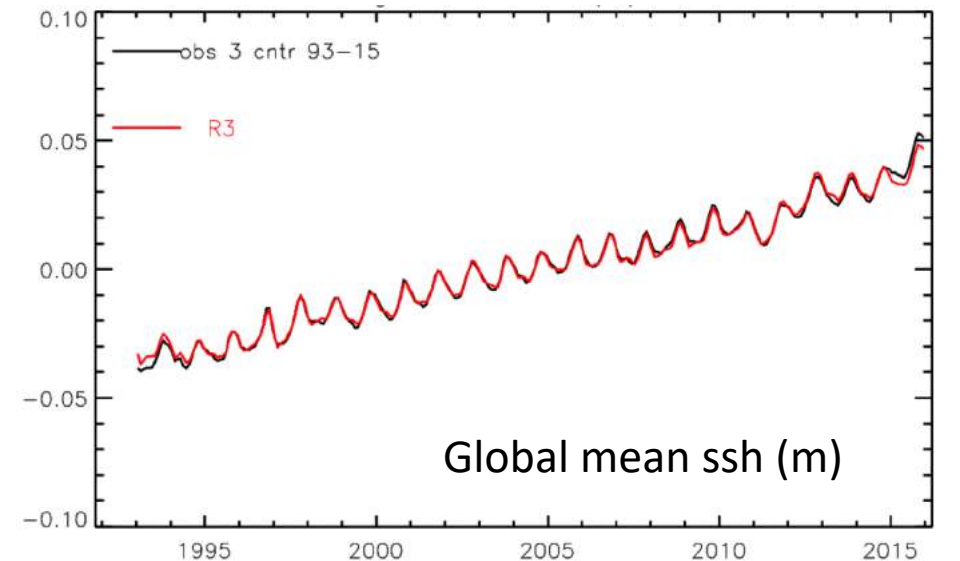
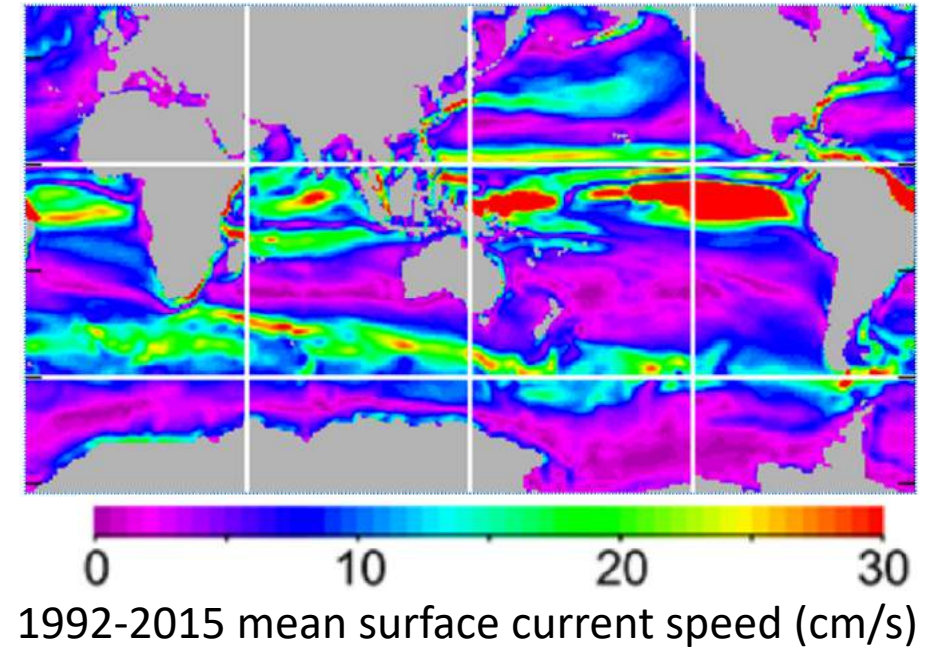
Jet Propulsion Laboratory

An T. Nguyen, Tim Smith

The University of Texas at Austin

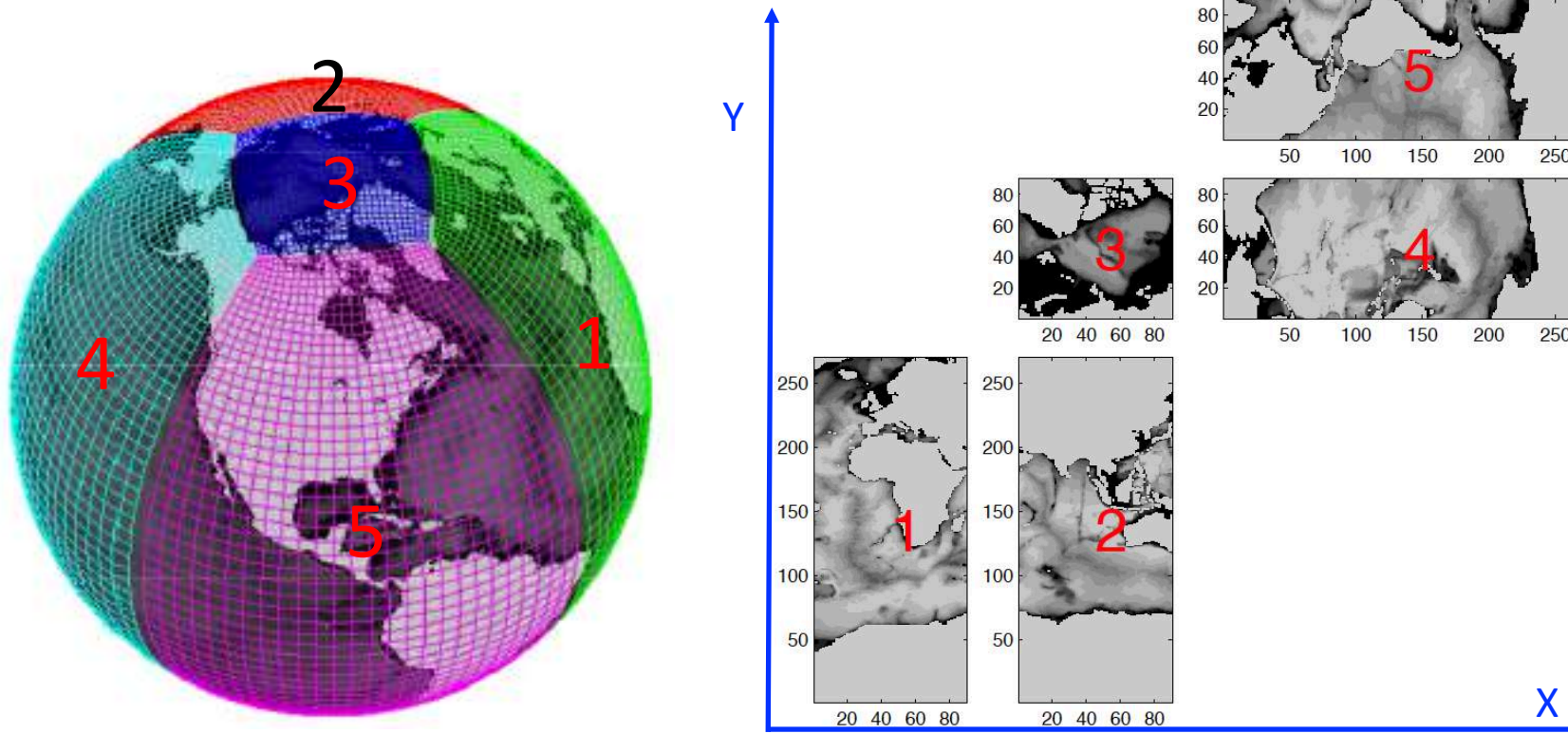
# ECCO Version 4

- Multi-decadal (1992-2015), global ocean state estimate;
- Constrained by satellite altimetry, GRACE, Aquarius, AVHRR, ARGO, CTD, XBT...;
- Adjusting atmospheric forcing, mixing parameters, and initial conditions (controls);
- Model: non-linear free surface boundary and real fresh water boundary conditions;
- A physically consistent solution.



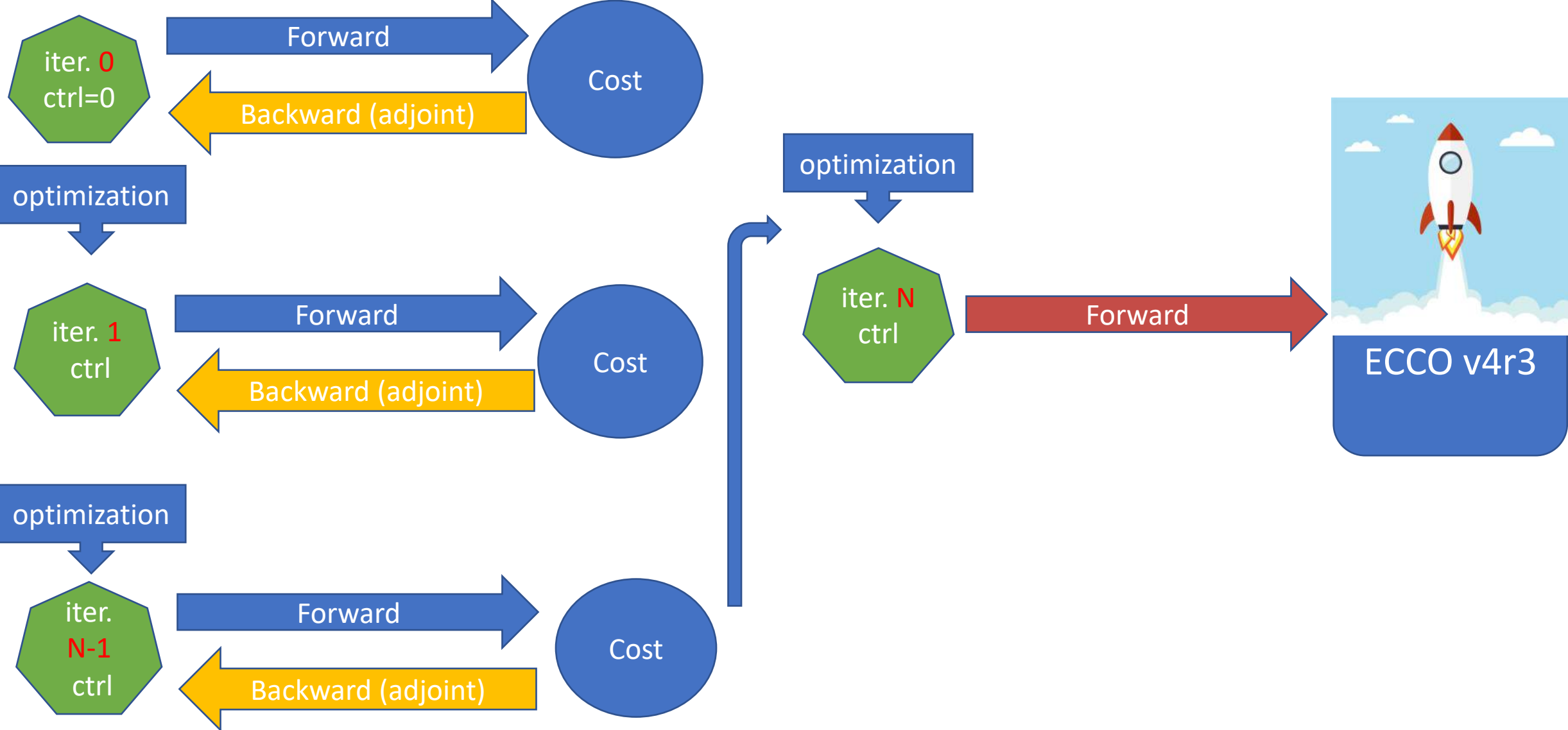
# ECCO v4r3

## Grid: Lat-Lon-Cap90 (LLC90)



Horizontal resolution	22km to 111km
Vertical resolution	10m to 457m from surface to bottom @ 6145m

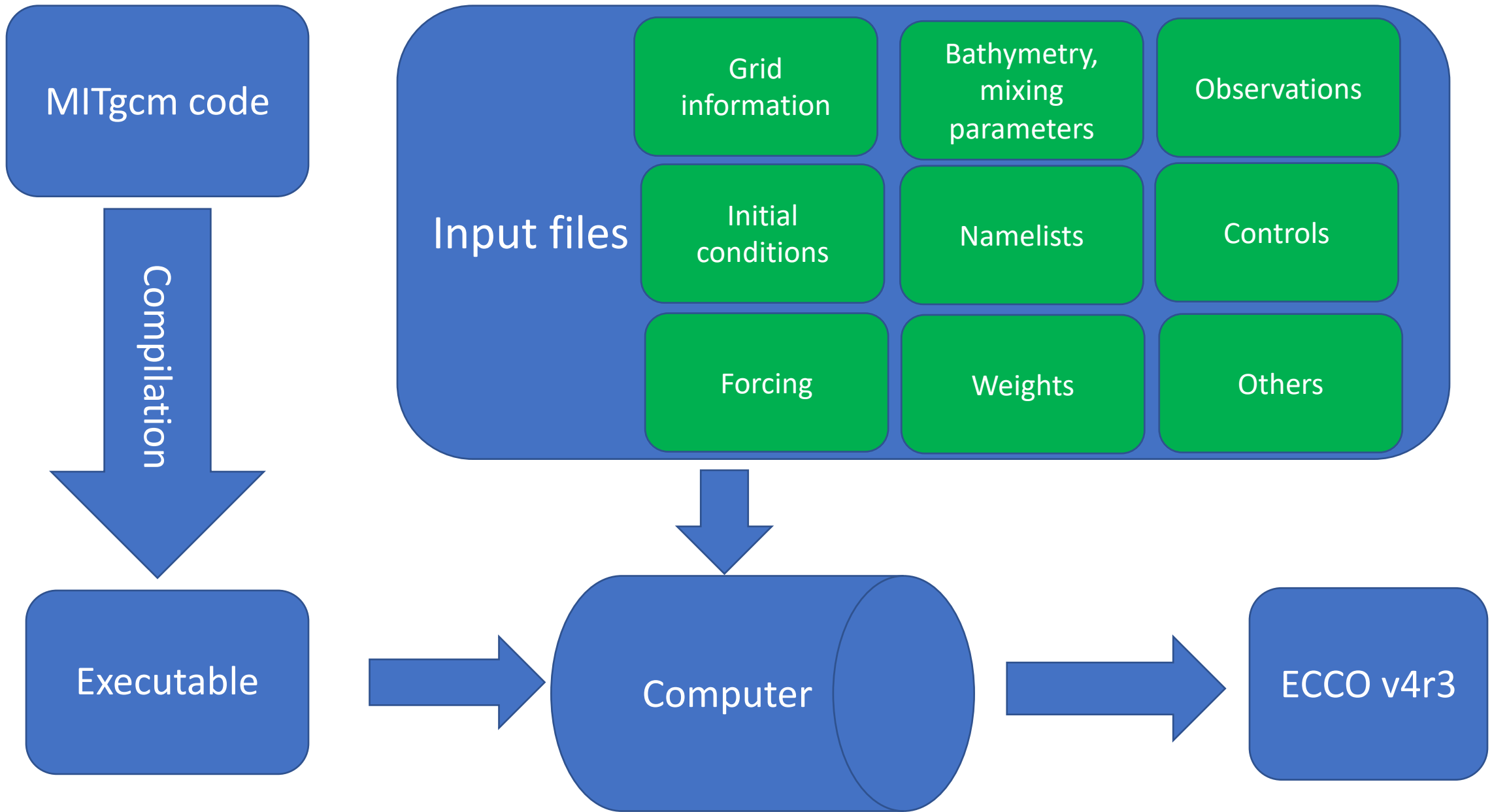
# An iterative process



# Why to reproduce ECCO v4r3?

- Generate different model outputs
  - Sampling frequency from monthly to weekly
  - Other model fields
- Forward sensitivity experiments using different forcing, mixing, etc.
  - Climatological forcing
  - Increased/reduced mixing parameters

- Code
- Input files
- Compilation
- Submit forward run
- Results



# Instructions

## 1 Reproducing ECCOv4r3 on TACC machines

2 An T. Nguyen, Tim Smith, Ou Wang

3 May 18, 2019

### 4 1 File System

#### 5 1.1 community directory

6 The directory [/work/projects/aci/ECCO/community/](#)  
7 should be accessible to ALL members of the ECCO School 2019 (those who have registered for TACC ac-  
8 cess). This is where we will host all the ECCOv4r3 data required to rerun the set up. In the following we  
9 will refer to this directory as **communitydir**.

10  
11 There is an example directory set up by An Nguyen, along with example bash scripts and extra files  
12 needed for the TACC machine stampede2, so that users can browse through and compare while trying to  
13 reproduce the ECCOv4r3 run:

14 \$communitydir/**atnguyen\_example**/ (extra\_namelists\_tiles/ , \*.bash)

[/work/projects/aci/ECCO/community/ECCO/ECCOv4/Release3/input.ec  
co\\_v4r3/reproducing\\_ECCOv4r3\\_TACC.pdf](#)

Also available at <https://www.eccosummerschool.org/resources>



# Instructions: A more general version

[ftp://ecco.jpl.nasa.gov/Version4/Release3/doc/ECCOv4r3\\_reproduction.pdf](ftp://ecco.jpl.nasa.gov/Version4/Release3/doc/ECCOv4r3_reproduction.pdf)

## Instructions for reproducing ECCO Version 4 Release 3

Ou Wang

Jet Propulsion Laboratory, California Institute of Technology

May 9, 2019

### **1 Introduction**

This instruction describes how users can reproduce the ECCO Version 4, Release 3 results that are archived at <ftp://ecco.jpl.nasa.gov/Version4/Release3/>.

The document is organized as follows. Section 2 describes how to download MITgcm code from MITgcm main CVS repository. Section 3 describes how

# Log on to stampede2.tacc.utexas.edu

```
[owang@pike ~]$ ssh -X owang@stampede2.tacc.utexas.edu
```

To access the system:

- 1) If not using ssh-keys, please enter your TACC password at the password prompt
- 2) At the TACC Token prompt, enter your 6-digit code followed by <return>.

Password:

TACC Token Code:

If you log on to this computer system, you acknowledge your awareness of and concurrence with the UT Austin Acceptable Use Policy. The University will prosecute violators to the full extent of the law.

TACC Usage Policies:

<http://www.tacc.utexas.edu/user-services/usage-policies/>

---

Welcome to Stampede2, \*please\* read these important system notes:

--> Stampede2 user documentation is available at:

<https://portal.tacc.utexas.edu/user-guides/stampede2>

----- Project balances for user owang -----

Name	Avail SUs	Expires
Polar-Project	1921	2020-03-31

----- Disk quotas for user owang -----

Disk	Usage (GB)	Limit	%Used	File Usage	Limit	%Used
/home1	0.0	10.0	0.03	47	200000	0.02
/work	0.4	1024.0	0.04	15263	3000000	0.51
/scratch	180.8	0.0	0.00	8115	0	0.00

Tip 35 (See "module help tacc\_tips" for features or how to disable)

\$OLDPWD is the name of the previous directory in bash and zsh. This can be abbreviated by "--".

\$ ls --

or with the cd command

\$ cd -

login4(1001)\$

# File systems

\$HOME: e.g. `/home/12345/jsmith`;

\$WORK: e.g. `/work/12345/jsmith/stampede2` top level directory for your code & analysis;

\$SCRATCH: e.g. `/scratch/12345/jsmith` where your run directory would be.

## Community directory for the summer school and others

```
>export communitydir=/work/projects/aci/ECC0/community/
```

```
>export binarydir=$communitydir/ECC0/ECC0v4/Release3/input.ecco_v4r3/
```

```
>export WORKINGDIR =$WORK/directory_for_your_code
```

```
>export basedir=$WORKINGDIR/MITgcm/verification/release3
```

# Download MITgcm code

1. `>cd $WORKINGDIR`
2. Download the code using git or bash

## From github

```
$ git clone https://github.com/MITgcm/MITgcm.git
```

or if ssh keys are set up

```
$ git clone git@github.com:MITgcm/MITgcm.git
```

Then to check out the MITgcm at the time specified for release 3:

```
$ git checkout 'git rev-list -n 1 --first-parent --before="2017-04-27 08:00" master'
```

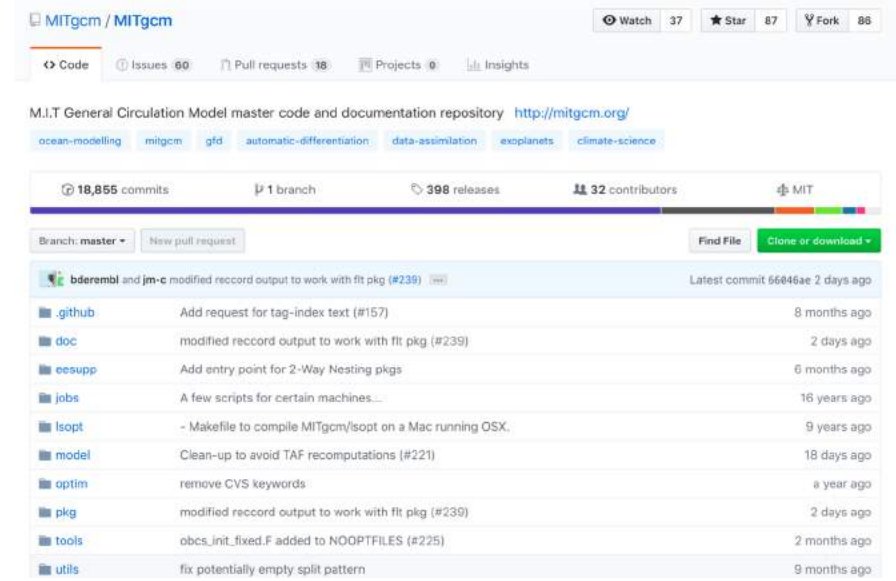
## From CVS using bash

```
$ export CVSROOT=':pserver:cvsanon@mitgcm.org:/u/gcmpack'
```

```
$ cvs login
```

```
( enter the CVS password: "cvsanon" )
```

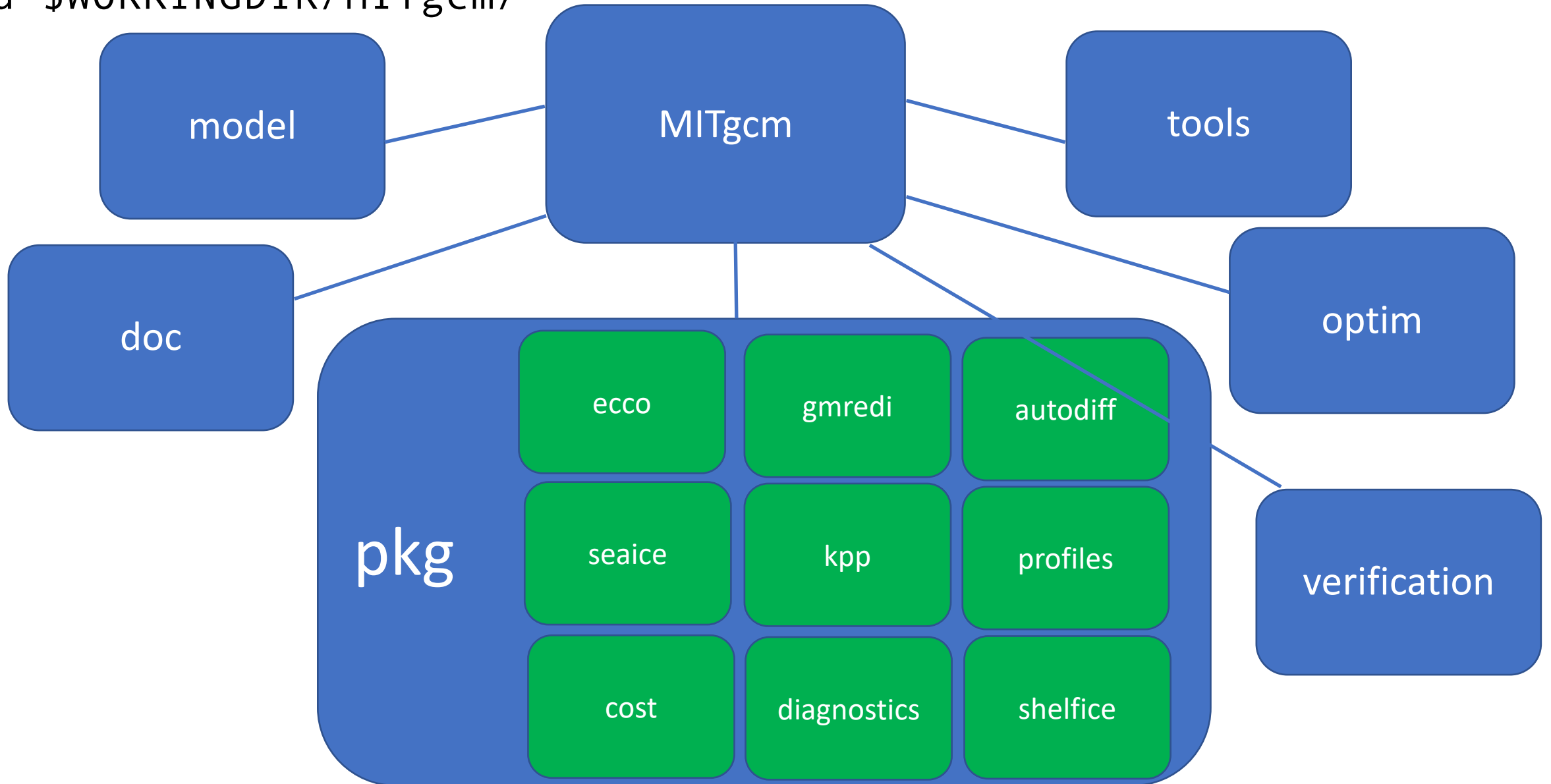
```
$ cvs co -P -D "2017-04-27 8:00" MITgcm_code
```



The screenshot shows the GitHub repository page for MITgcm. The repository name is MITgcm / MITgcm. It has 37 watchers, 87 stars, and 86 forks. The repository is described as 'M.I.T General Circulation Model master code and documentation repository' with the URL <http://mitgcm.org/>. There are 18,855 commits, 1 branch, 398 releases, and 32 contributors. The current branch is master. A list of recent commits is shown, including one by bderembi and jm-c modifying record output to work with flt pkg (#239) 2 days ago. Other commits include adding a request for tag-index text (#157) 8 months ago, adding an entry point for 2-Way Nesting pkgs 6 months ago, and adding a few scripts for certain machines 16 years ago.

# Directory structure

```
>cd $WORKINGDIR/MITgcm/
```



# V4r3 specific code

Why do we need this? CPP options and SIZE.h

```
cd MITgcm
mkdir -p verification/release3
cd verification/release3
cvs co -P MITgcm_contrib/ecco_utils/ecco_v4_release3_devel/code
mv MITgcm_contrib/ecco_utils/ecco_v4_release3_devel/code .
\rm -rf MITgcm_contrib
```

[\\$WORKINGDIR/MITgcm/verification/release3/code](#)

[/\[MITgcm\]/MITgcm\\_contrib/ecco\\_utils/ecco\\_v4\\_release3\\_devel/code](#)

# Index of /MITgcm\_contrib/ecco\_utils/ecco\_v4\_release3\_devel/code



Files shown: 27

Sticky Tag:

File	Rev.	Age	Author	Last log entry
<a href="#">Parent Directory</a>				
<a href="#">AUTODIFF_OPTIONS.h</a>	1.1	2 years	ou.wang	update code and input directories to be consistent with the code (checkpoint66g)...
<a href="#">CPP_OPTIONS.h</a>	1.1	2 years	ou.wang	update code and input directories to be consistent with the code (checkpoint66g)...
<a href="#">CTRL_OPTIONS.h</a>	1.2	21 months	ou.wang	Updating etaN, but not etaH, when initial SSH is part of control variables.
<a href="#">CTRL_SIZE.h</a>	1.1	2 years	ou.wang	update code and input directories to be consistent with the code (checkpoint66g)...



# Input files (runtime parameters and model inputs)

`$communitydir/ECCO/ECCOv4/Release3/input.ecco_v4r3`

directory	contents
input_init	Grid, initial conditions, bathymetry, etc.
input_forcing	Adjusted forcing (ERA + atm. control adjustments)
input_ecco	Observations
profiles, profiles_30x30, profiles_15x30, profiles_45x45	in situ data for various tile configurations
xx*	Control adjustments
reproducing_ECCOv4r3_TACC.pdf	Instructions to reproduce ECCOv4r3 on TACC

# Copy runtime parameters

Location: \$basedir/my\_namelists

Why? Modify your own copy as needed

```
>cp -r $binarydir/input_init/NAMELIST $basedir/my_namelists
>chmod -R u+w $basedir/my_namelists
>cp $communitydir/atnguyen_example/extra_namelists_tiles/data* \
  $basedir/my_namelists
>cd $basedir/my_namelists/
>cp data_exch2_30x30x96 data.exch2 (if SIZE.h_30x30x96 was chosen)
```

# Modules

```
[[stampede2 ~]$ module list
```

```
Currently Loaded Modules:
```

- 1) intel/18.0.2
- 2) libfabric/1.7.0
- 3) impi/18.0.2
- 4) git/2.9.0
- 5) autotools/1.1
- 6) python2/2.7.15
- 7) cmake/3.10.2
- 8) xalt/2.6.5
- 9) TACC
- 10) netcdf/4.3.3.1

An Intel compiler

Parallel computing

TACC core module

Needed it for handling netCDF files

```
>cat $communitydir/computing/env/bashrc_stampede2 >> ~/.bashrc  
>source ~/.bashrc
```

# Domain decomposition for parallel computing

- Global horizontal grid dimensions: 1170x90
- Users can choose the following tile configuration:
  - 30x30 tiles using 96 CPUs (default)
  - 45x45 tiles using 48 CPUs
  - 15x15 tiles using 187 CPUs

# Compilation

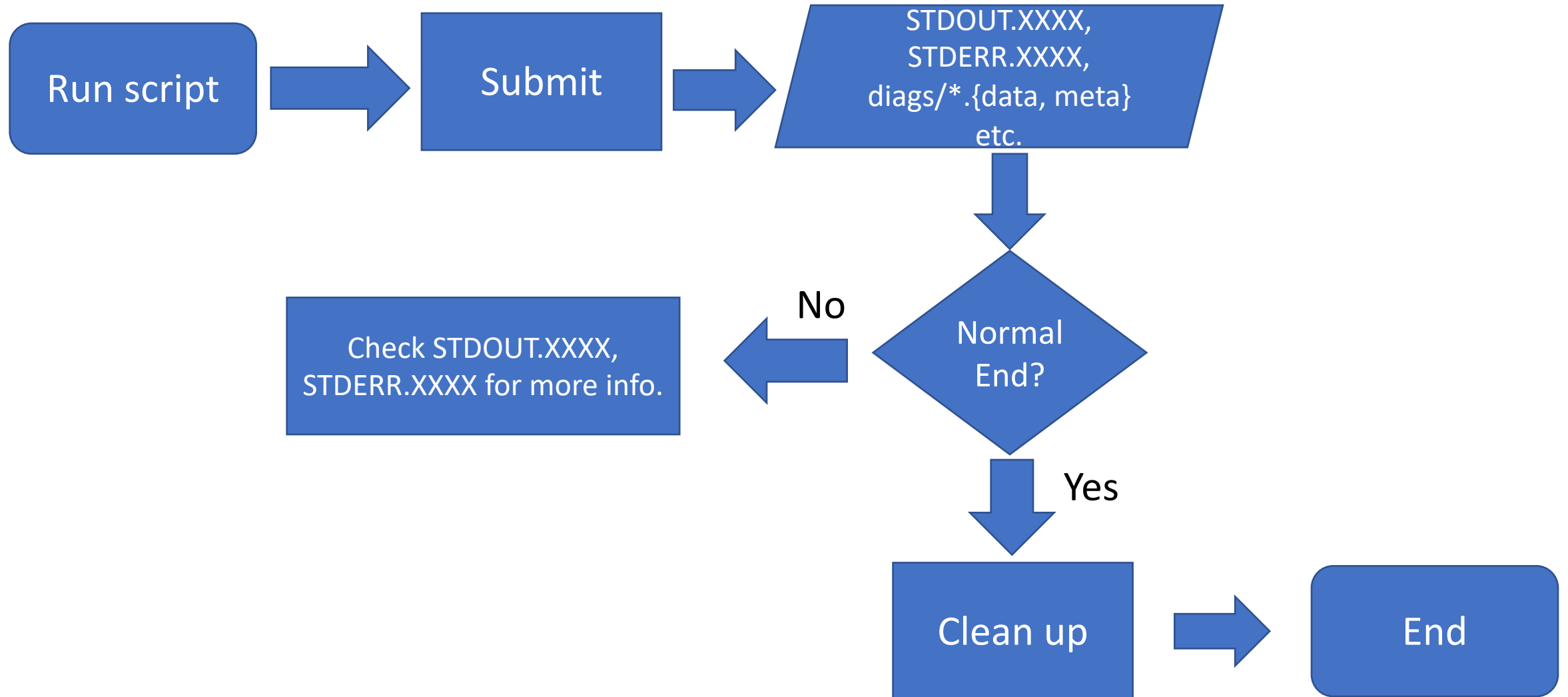
```
cd $basedir          change to $basedir ($WORKINGDIR/MITgcm/verification/release3/)
mkdir build         where to "build" the executable
cd build
../../../../tools/genmake2 -mods=../code -mpi \
-optfile=$communitydir/computing/optfiles/linux_amd64_ifort+mpi_stampede2_skx
```

genmake2: a bash script to generate Makefile

optile: an option file specifying compiler flags and options

```
make depend        Link header files
make all           To generate an executable called mitgcmuv
mv mitgcmuv mitgcmuv_30x30x96 (if SIZE.h_30x30x96 was chosen)
cd ../
```

# Conduct the run



# Sample run script

`$communitydir/atnguyen_example/script_r3_stampede2.bash`

```
#!/bin/bash
```

```
#SBATCH -J eccov4r3
```

```
#SBATCH -o eccov4r3.%j.out
```

```
#SBATCH -e eccov4r3.%j.err
```

```
##=====
```

```
##### stampede2
```

```
#####option to use the devel queue for something quick, limit
```

```
#SBATCH -p skx-dev
```

```
#SBATCH -t 2:00:00
```

```
#####option for using 4 nodes and 96cpus:
```

```
##SBATCH -N 4
```

```
##SBATCH -n 96
```

```
#SBATCH --mail-user=atnguyen@oden.utexas.edu
```

```
#SBATCH --mail-type=begin
```

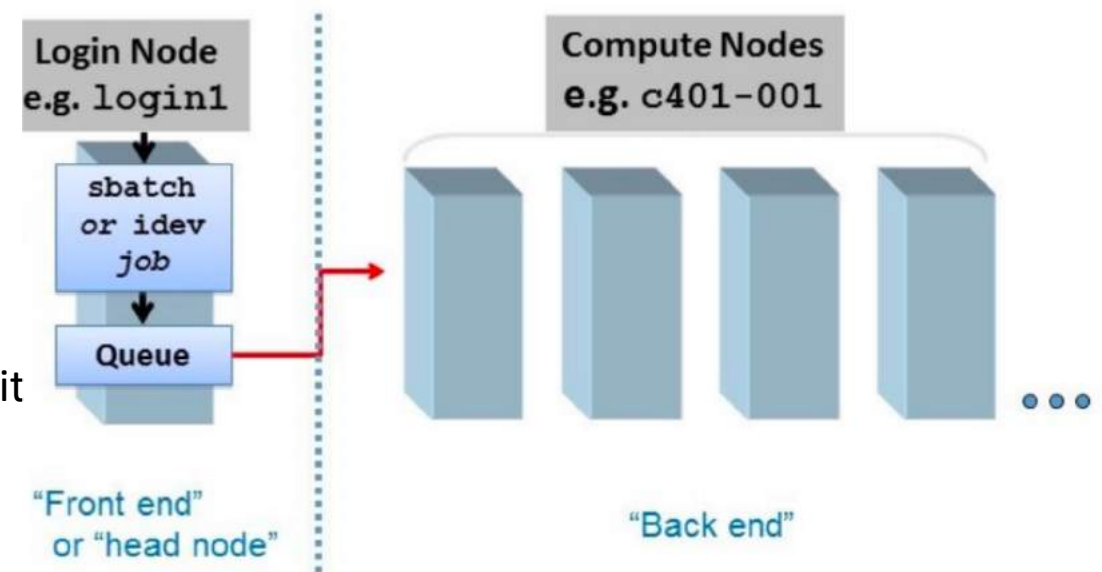
```
#SBATCH --mail-type=end
```

← Job name, log files

← Queue name and run wall-clock time

← 4 (N) nodes and 96 (n) CPUs

← Email notification; change it.



From TACC user portal

# Sample run script (continued)

```
module purge  
module load intel/18.0.2
```

Load the same modules when generating executable

...

```
nprocs=96  
snx=30  
sny=30
```

Number of CPUs and dimensions of each tile

...

```
builddir=$basedir/build${forwadj}
```

Set directories, e.g. build, input

...

```
ln -s ${binarydir}/input_forcing/eccov4r3*
```

Link forcing and other input files

...

```
remora ibrun -n $nprocs ./mitgcmuv${forwadj}
```

Run command



# Submit job to Skylake nodes (48-CPU per node, 4GB/CPU)

```
>cp $communitydir/atnguyen_example/script_r3_stampede2.bash $basedir
```

make necessary changes

```
>SBATCH script_r3_stampede2.bash
```

Other useful commands for batch jobs.

command	
>queue -u yourusername	check the status of your job
>scancel yourjobid	kill your job

See more on the TACC user guide page at  
<https://portal.tacc.utexas.edu/user-guides/stampede2#slurm-job-scheduler>

# Results

`$$SCRATCH/run_r3_stampede2_it0059_np96`

## Expected files

files	
STDOUT.XXXX	model configuration, monitored statistics of model state variables
STDERR.XXXX	any warnings
diags/*.{data,meta}	outputs of the model state in binary format
profiles/*	model equivalents of profile
m_*. {data,meta}	outputs from pkg/ecco for cost calculation
xx*	control adjustments

**Successful** if the last line of STDOUT.0000 is:

**PROGRAM MAIN: Execution ended Normally**

# Clean-up script

\$communitydir/atnguyen\_example/script\_cleanup\_rundir

name	
run_cleanup	reorganize files into separate directories, list binary and forcing inputs for book-keeping
run_cleanup_tar	tar the large input and output binaries
run_cleanup_rm	<b>CAUTION!</b> remove files that have already be tarred and zipped.

```
>chmod u+x run_cleanup
>chmod u+x run_cleanup_tar
>./run_cleanup
>./run_cleanup_tar
```

**CAUTION!** Make sure the files have been tarred successfully prior to executing the following  
(chmod u+x run\_cleanup\_rm)  
(./run\_cleanup\_rm)

# Sample directory structure after clean-up

```
$commuitydir/atnguyen_example/MITgcm/verification/release3/run_r3_stampede2_it0059_np96/
```

```
ADJfiles_empty      list_costfunction  remora_3511686.tgz
ADXXfiles           list_pickup        remora_memory_tail.txt
available_diagnostics.log list_tapes         run.MITGCM.timing
barfiles.tgz        mitgcmuv.gz       SBO_global.0000000001.data.gz
code                NAMELISTS         SBO_global.0000000001.meta
diags               OUTPUT_dump2.tgz  script_r3_stampede2.bash
GRID.tgz            OUTPUT_dump.tgz   STDERR.0000
impi_data.txt       PICKUP.tgz        STDOUT.0000.gz
list_barfiles       profiles.tgz       STDs.tgz
```

```
diags/
```

```
budg2d_hflux_set1.tgz  budg2d_zflux_set1.tgz  exf_zflux_set1.tgz  state_3d_set1.tgz
budg2d_hflux_set2.tgz  budg2d_zflux_set2.tgz  list_BUDG           state_3d_set2.tgz
budg2d_hflux_set3_11.tgz budg2d_zflux_set3_11.tgz list_STATE          tar_diags1
budg2d_kpptend_set1.tgz budg3d_hflux_set2.tgz  list_TRSP          trsp_2d_set1.tgz
budg2d_snap_set1.tgz   budg3d_kpptend_set1.tgz rm_diags1          trsp_3d_set1.tgz
budg2d_snap_set2.tgz   budg3d_snap_set2.tgz  state_2d_set1.tgz  trsp_3d_set2.tgz
budg2d_snap_set3_11.tgz budg3d_zflux_set2.tgz  state_2d_set2.tgz  trsp_3d_set3.tgz
```

# Namelist (runtime parameters)

filename	
data	Core runtime parameters
data.cal	Specify model start time
data.pkg	Individual package switch
data.diagnostic	Diagnostics variables and output frequencies
data.exf	Forcing files and formats
data.profiles	In situ files
data.ecco	Cost terms
data.gmredi	GM-Redi parameters
data.seaice	Sea-ice parameters
data.ctrl	Control variables, weights
data.exch2	Tile exchange and blank tile parameters

# Namelist (runtime parameters)

- data  
24-year

```
# Time stepping parameters
&PARM03
nIter0=1, #start time-step number
#24y:
nTimeSteps=210359, #number of time-steps
```



3-day

```
# Time stepping parameters
&PARM03
nIter0=1,
#3day:
nTimeSteps=59,
```

- data.diagnostics  
Monthly mean

```
#---
frequency(2) = 2635200.0,
fields(1,2) = 'ETAN',
filename(2) = 'diags/ETAN_mon_mean',
#---
```



Daily mean

```
#---
frequency(2) = 86400.0,
fields(1,2) = 'ETAN',
filename(2) = 'diags/ETAN_day_mean',
#---
```

# Things to know to run it on another computer

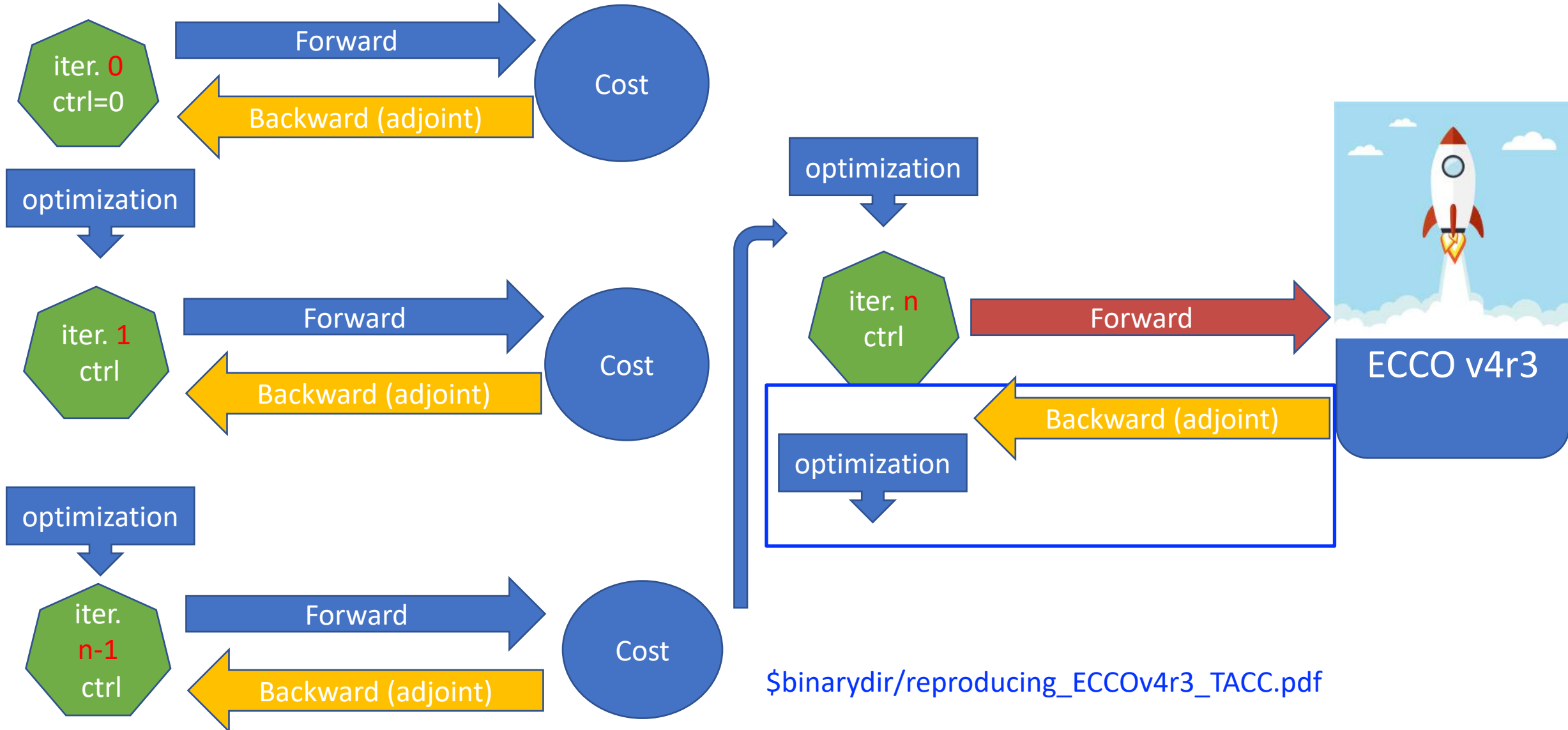
- Have necessary modules installed
  - Fortran compiler
  - MPI
  - netCDF
- Compilation option file
  - linux\_amd64\_ifort+mpi\_stampede2\_sgx may not work
  - Start from one from the list in MITgcm/tools/build\_options/. Pick up one having the same operating system, machine name and compiler in the filename as yours.
- Run out memory?
  - Request more CPUs than the number of tiles

# To run v4r3 for 3-day

- \$communitydir/Hands\_on/reproducingv4r3
  - README
  - steps\_rerun\_eccov4r3\_commandline.txt
  - script\_r3\_stampede2.3day.bash
  - reorganize\_diags.py



# To be presented: Adjoint and optimization



***Thank you!***

**Questions:**

**`ecco-support@mit.edu`**

**(please subscribe via**

**`http://mailman.mit.edu/mailman/listinfo/ecco-support`)**