ECCO v4 development notes

Gaël Forget
Department of Earth, Atmospheric and Planetary Sciences
Massachusetts Institute of Technology

February 17, 2015

abstract

These notes pertain to the revamped and augmented estimation modules of MITgcm; to the global setups and files used in regression tests; and to the analysis of the state estimate solution, including its re-runs. The intention is to compile and condense practical information for users and developers of ECCO v4 setups. Some of the included material will finds its way in Forget and al. (2015), and some should be added to the MITgcm manual.

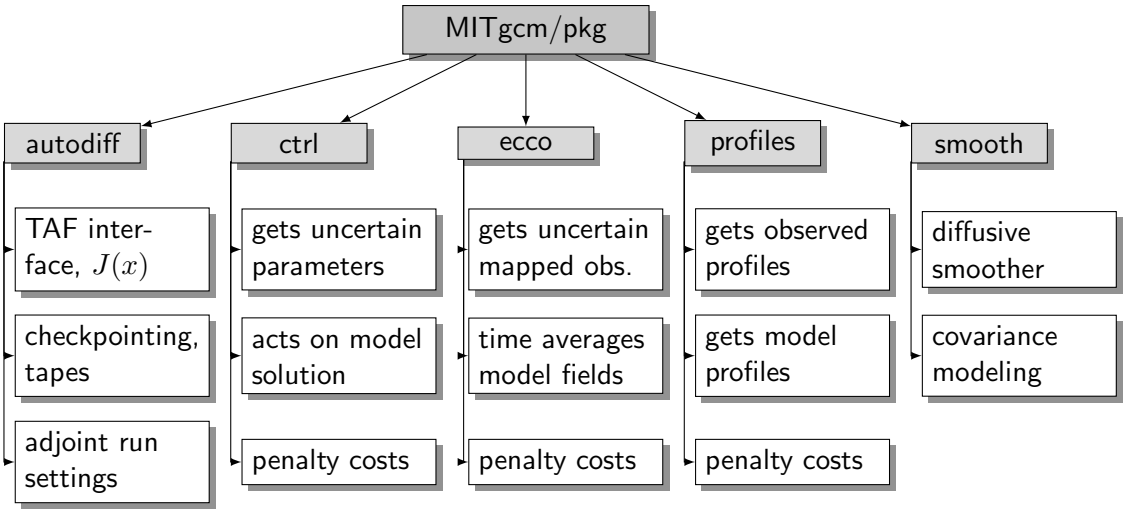


Figure 1: Organization and roles of MITgcm estimation modules.

Contents

1 downloads

1.1 MITgcm

Pre-requisites are cvs, gcc, gfortran (or alternatives), and mpi (only for parallel runs). Then :

- The MITgcm web-page is mitgcm.org
- Install MITgcm using cvs as explained @ cvs
- Run MITgcm using testreport (for one experiment) as explained @ manual, howto

For example, my laptop setup, including mpi and netcdf, involved the following mac ports :

- cvs @1.11.23_1 (active)
- wget @1.14_5+ssl (active)
- gcc48 @4.8.2_0 (active)
- mpich-default @3.0.4_9+gcc48 (active)
- mpich-gcc48 @3.0.4_9+fortran (active)
- netcdf @4.3.0_2+dap+netcdf4 (active)
- netcdf-fortran @4.2_10+gcc48 (active)

Side note – overridding the default mac gcc and mpich with the above, further requires

- sudo port select –set gcc mp-gcc48
- sudo port select –set mpich mpich-gcc48-fortran

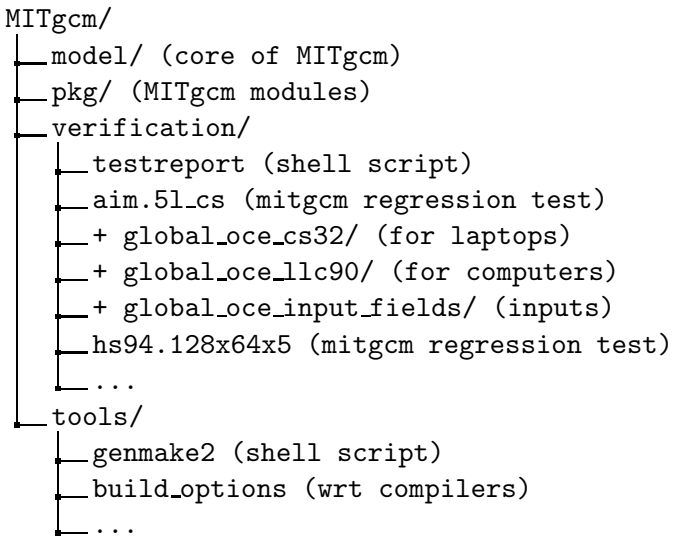
Side note – using mpi and netcdf within MITgcm requires two environment variables :

- export MPI_INC_DIR=/opt/local/include
- export NETCDF_ROOT=/opt/local

1.2 ECCO version 4 setup

Pre-requisites are MITgcm (see above) and mpi (except for small setup). User can then install the ECCO v4 setups, as explained @ README, using the setup_these_exps.csh shell script. This script downloads global_oce_cs32/ (small setup), global_oce_llc90/ (bigger setup) and global_oce_input_fields.tar.gz binary model inputs to global_oce_tmp_download/ (local subdirectory). User can then move these directories to MITgcm/verification/ to allow for automated execution by testreport using genmake2 (Fig.2).

Figure 2: MITgcm directory structure downloaded using cvs. The ECCO v4 directories indicated with "+" were downloaded separately using setup_these_exps.csh script and moved to MITgcm/verification/.



1.3 solution

The release 1 solution directory linked to ecco-group.org contains :

- 20 year solution output : readme, fields, profiles, grid
- additional input files required to run the full 20 year solution (coming soon...).

1.4 analysis

Tools (e.g. matlab scripts) available to analyze the release1 solution, and others, include :

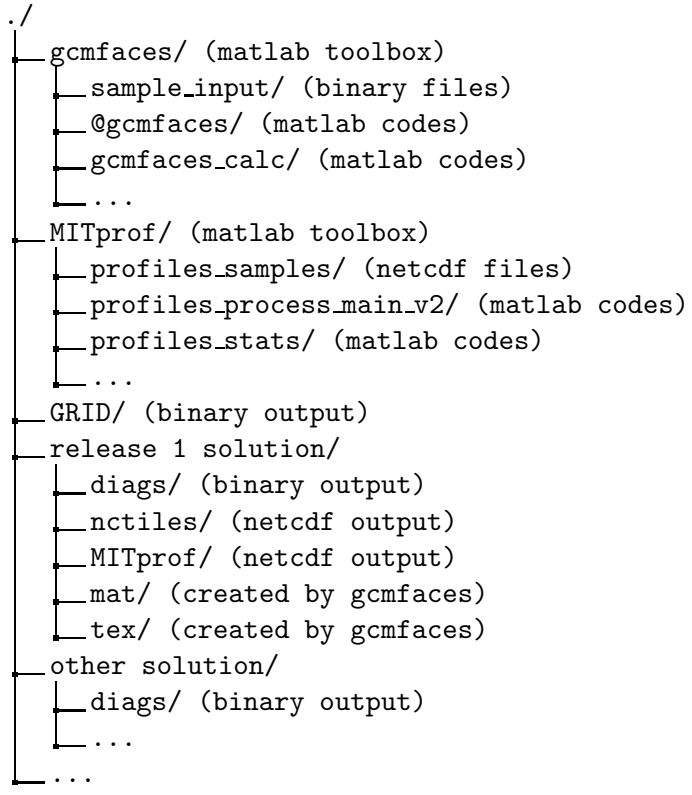
- download, set-up gcmfaces + MITprof using shell script or manually (see getting_started.m)
- download MITgcm/utils using cvs (basic functionalities only).

The so-called standard analysis.pdf is generated in matlab by means of diags_driver.m and diags_driver_tex.m in the following sequence :

```
diags_driver('release1/','release1/mat/',1992:2011);
diags_driver_tex('release1/mat/',{},'release1/tex/standardAnalysis');
```

assuming the conventional directory structure shown in Fig.3.

Figure 3: Directory structure as expected by gcmfaces and MITprof toolboxes. The toolboxes themselves can be relocated anywhere as long as their locations are included in the matlab path. Advanced analysis using diags_driver.m and diags_driver_tex.m will respectively generate the mat/ directory (for intermediate computational results) and the tex/ directory (for standard analysis). This diagnostic process relies on the depicted organization of GRID/ and solution/ for automation (user will otherwise be prompted to enter directory names) and depends on downloaded copies of fields to nctiles/ (local subdirectory).



2 MITgcm runs

Here I document a few procedures, commands and submission scripts that may be relevant to run the ECCO v4 MITgcm setup – either in short regression tests or for multi-decadal simulations such as the full 20 year state estimate. Downloading MITgcm and the ECCO v4 setups is a pre-requisite (section 1.2).

2.1 regression tests

MITgcm and ECCO v4 regression tests are run using testreport utility (see Fig.3; howto). Serial regression tests can always be executed simply with, e.g.

```
./testreport -t global_oce_cs32
```

```
./testreport -skipdir global_oce_llc90
```

If something goes wrong and/or interrupts the process it is often safer to clean up experiment directories (e.g., by executing ./testreport -clean -t global_oce_*) and start over. For example, the global_oce_llc90 experiments require 12 processors in forward (96 in adjoint), and may crash your laptop if you attempted to run them in serial mode.

Often in massive computing environments, however, mpi jobs can only be run within a queuing system. The, machine specific, submission script in Fig.4 provides an example. It contains 3 hard-coded switches : fwdORad = 1 (2 for adjoint); numExp = 1 (2 for llc90); excludeMpi = 0 (1 for serial). This script is located and submitted from MITgcm/verification. If compute nodes cannot access the remote adjoint compiler (TAF), then proceed in two steps :

1. compile outside of the queuing system using e.g.

```
./testreport -of ../tools/build_options/linux_amd64_ifort+mpi_ice_nas \
-j 4 -MPI 96 -command 'mpiexec -np TR_NPROC ./mitgcmuv' \
-t global_oce_llc90 -norun
```

2. Before submiting the Fig.4 script, add -q to the 'opt' variable to skip compilation.

Adjoint test require access to the TAF compiler. Then the call to testreport only needs to be altered by appending the '-ad' option and replacing 'mitgcmuv' with 'mitgcmuv_ad'. The ECCO v4 regression tests do not include the common, adjoint specific 'code_ad/' directory, which is generally un-necessary. Since testreport relies on the existence of 'code_ad/' for its adjoint option though, it is necessary to soft link 'code/' to 'code_ad/' in both global_oce_cs32/ and global_oce_llc90/ to run 'testreport -ad' on those experiments.

2.2 full ECCO v4 runs

There are three main differences between regression tests and full model runs (see howto) :

- compilation and run are done without testreport and with compiler optimization.
- additional forcing, control vectors and/or observational inputs is necessary.

- additional memory and/or disk space is often necessary.

The typical compilation sequence is shown in Fig.5. The tamc.h_itXX and profiles.h_itXX headers allow for additional time steps, and additional in situ data input, respectively. Also note that compiling the adjoint requires a TAF license. Once that is done, user creates and enters a run directory, links everything into place (see Fig.6), and finally submits a job to the queueing system (see Fig.7).

A mechanism, analogous to testreport but for long runs, has been introduced recently (Forget and al., 2015) that is testreport_ecco.m run within Matlab, which requires the downloaded 'MITgcm/verification/global_oce_llc90/results_itXX/' to be in the Matlab path. By itself is compares cost functions and global mean time series to the reference state estimate values. These can be extended to meridional transports, which requires gcmfaces. The typical call sequence is indicated in the help of testreport_ecco.m and Fig.8.

Note to self ... [1]

[1] Material for separate paragraph memory and disk space requirements, and additional input downloads.

Figure 4: Example script to run mpi testreport via a queueing system (machine dependent).

```
#PBS -S /bin/csh
#PBS -l select=1:ncpus=16:model=ivy+4:ncpus=20:model=ivy
#PBS -l walltime=02:00:00
#PBS -q devel
#PBS -m n

#environment variables and libraries
#-------------------------------------------------
limit stacksize unlimited
module purge
module load modules comp-intel/2013.1.117 mpi-sgi/mpt.2.10r6 netcdf/4.0
#
setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${HOME}/lib
setenv MPI_IB_TIMEOUT 20
setenv MPI_IB_RAILS 2
setenv MPI_IB_FAILOVER 1
setenv  MPI_CONNECTIONS_THRESHOLD  2049

#local variables and commands
#--------------------------------------
set fwdORad = 1
set numExp  = 1
set excludeMpi = 0
#
if ( ${numExp} == 1) then
  set nameExp = global_oce_cs32
  set NBproc  =  6
else
  set nameExp = global_oce_llc90
  set NBproc  = 96
endif
#
if ( ${excludeMpi} == 1 ) then
  set opt = '-of ../tools/build_options/linux_amd64_ifort -j 4'
else
  set opt = '-of ../tools/build_options/linux_amd64_ifort+mpi_ice_nas -j 4'
endif
#
if ( ${fwdORad} == 1 && ${excludeMpi} == 0 ) then
  ./testreport ${opt} -MPI \
  ${NBproc} -command 'mpiexec -np TR_NPROC ./mitgcmuv' -t ${nameExp}
else if ( ${fwdORad} == 2 && ${excludeMpi} == 0 ) then
  ./testreport ${opt} -MPI \
  ${NBproc} -command 'mpiexec -np TR_NPROC ./mitgcmuv_ad' -ad -t ${nameExp}
else if ( ${fwdORad} == 1 && ${excludeMpi} == 1 ) then
  ./testreport ${opt} -t ${nameExp}
else if ( ${fwdORad} == 2 && ${excludeMpi} == 1 ) then
  ./testreport ${opt} -ad -t ${nameExp}
endif

exit
```

Figure 5: Compilation directives, outside testreport, for intensive model runs. On a different machine (computer) another build option file such as linux_amd64_gfortran or linux_amd64_ifort11 should be used. To compile the adjoint, users need a TAF license and to replace 'make -j 4' with 'make adall -j 4'. Note : the '-mods=../code' specification can be omitted if the build directory contains the 'genmake_local' file).

```
cd verification/global_oce_llc90/build
../../../tools/genmake2 -optfile=\\
../../../tools/build_options/linux_amd64_ifort+mpi_ice_nas -mpi -mods=../code

make depend

\rm tamc.h profiles.h
cp ../code/tamc.h_itXX tamc.h
cp ../code/profiles.h_itXX profiles.h

make -j 4
```

Figure 6: Example script to setup the 20 year ECCO v4 state estimate. It is implied that user has filled directories /bla, /blaa, /blaaa and /blaaa with appropriate forcing, observational, control vector, and pickup files.

```
#!/bin/csh -f

set forcingDir  = ~/bla
set obsDir      = ~/blaa
set ctrlDir     = ~/blaaa
set pickDir     = ~/blaaaa

source ../input_itXX/prepare_run
cp ../build/mitgcmuv .
\rm pick*ta EIG*
ln -s ${forcingDir}/EIG* .
ln -s ${obsDir}/* .
ln -s ${ctrlDir}/xx* .
ln -s ${pickDir}/pick* .

exit
```

Figure 7: Example script to run the 20 year ECCO v4 state estimate on 96 processors (machine dependent).

```
PBS -S /bin/csh
#PBS -l select=1:ncpus=16:model=ivy+4:ncpus=20:model=ivy
#PBS -l walltime=12:00:00
#PBS -q long

#environment variables and libraries
#-------------------------------------------
limit stacksize unlimited
module purge
module load modules comp-intel/2013.1.117 mpi-sgi/mpt.2.10r6 netcdf/4.0
#
setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${HOME}/lib
setenv MPI_IB_TIMEOUT 20
setenv MPI_IB_RAILS 2
setenv MPI_IB_FAILOVER 1
setenv  MPI_CONNECTIONS_THRESHOLD  2049

#run MITgcm
#----------------
mpiexec -np 96 dplace -s1 ./mitgcmuv
exit
```

Figure 8: Calling sequence to be executed form within matlab to verify that their re-run of the 20 year ECO v4 state estimate is acceptably close to the released state estimate.

```
addpath ../results_itXX;%necessary .m and .mat files
mytest0=testreport_ecco([],'release1'); mytest0.info.interactive=0;%initialization
mytest=testreport_ecco(mytest0,'release1',[-1:4],'./',1);%compute the tests
testreport_ecco(mytest,'release1');%display the results
%testreport_write(mytest,'myRun');%save the results to a mat file
```

References

Forget, G. and al., 2015: ECCO version 4: an integrated framework for non-linear inverse modeling and global ocean state estimation. *Geoscientific Model Development*, **((to be submitted))**.