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 to obtain patch code for Release 3. Section 4 deals with the forcing, initial condition, and other auxiliary files that are needed to reproduce Release 3. Section 5 talks about how to compile and run the model. Normally, the model should be run non-stop for the whole time period. See more details about how to stop and then restart the model in Section 5.2.1.

2 Download source code

One could download the source code from MITgcm cvs server. First create a directory, say, called WORKINGDIR. All directories and files downloaded thereof will be under WORKINGDIR.

```
mkdir WORKINGDIR
cd WORKINGDIR
```

For bash or sh shell:

```
$ 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
```

tcsh or csh shell:

```
$ setenv 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
```

Alternatively, one could download the code from the MITgcm's github repository by either

```
$ 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`
```

3 Release 3 specific code

The Release 3 specific code is archived in MITgcm CVS repository and can be retrieved as follows:

```
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
```

4 Forcing and other input files

The forcing and other input files are stored on Release 3's ftp server at ftp://ecco.jpl.nasa.gov/Version4/Release3/ and can be downloaded using the following commands:

```
wget --recursive ftp://ecco.jpl.nasa.gov/Version4/Release3/input_forcing
wget --recursive ftp://ecco.jpl.nasa.gov/Version4/Release3/input_init
wget --recursive ftp://ecco.jpl.nasa.gov/Version4/Release3/input_ecco
```

The total size of the forcing and other input files are about 150Gb. All forcing and input files are necessary to reproduce Release 3.

4.1 Diagnostic output

Release 3 specifies a list of core diagnostic products in

```
ftp://ecco.jpl.nasa.gov/Version4/Release3/input_init/NAMELIST/data.diagnostics
```

that one can use to close property budgets for heat, salt, mass, volume, and momentum. A full list of pre-defined diagnostics variables is available at

`ftp://ecco.jpl.nasa.gov/Version4/Release3/doc/available_diagnostics.log`

Chapter 7 of the MITgcm manual, titled "Diagnostics and I/O - Packages II, and Post-Processing Utilities", available at

`http://mitgcm.org/public/r2_manual/latest/online_documents/node264.html`

gives instructions about how to output different diagnostic variables, change the time-averaging period and vertical levels, and even define a new diagnostic variable.

5 Compile and run

The compilation and run of the model are compiler and platform specific. Given below is an example to compile and run the code on NASA Ames' Pleiades.

5.1 Compile

The example given here is to compile the code on NASA Ames' Pleiades. A different set of build options may be required for other machines. Sets of build options for various platforms and compilers have been provided in MITgcm/-tools/build_options. Users may need to create their own build options that are deemed suitable to their unique platform and/or compilers.

```
cd MITgcm/verification/release3
mkdir build
cd build
../../../tools/genmake2 -mods=../code -optfile=../../../tools/
   build_options/linux_amd64_ifort+mpi_ice_nas -mpi
make depend
make all
cd ..
```

If compiled successfully, the executable will be ./build/mitgcmuv.

5.2 Run

Below is an example of the run script for Pleiades. The run directory will be MITgcm/verification/release3/run/. This run script will reprouce v4r3's 24-yr results from 1992 through 2015 without stopping.

```
#PBS -S /bin/csh
#PBS -l select=4:ncpus=24:model=has
#PBS -l walltime=14:00:00
```

```
#PBS -j oe
#PBS -o ./
#PBS -m bea

limit stacksize unlimited
module purge
module load comp-intel/2015.0.090
module load mpi-sgi/mpt.2.12r26
module load math/intel_mkl_64_10.0.011
module load netcdf/4.1.2
module list

setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:${HOME}/lib
setenv FORT_BUFFERED 1
setenv MPI_BUFS_PER_PROC 128
setenv MPI_DISPLAY_SETTINGS

set nprocs  = 96
set basedir   = ./
set inputdir  = ./ecco.jpl.nasa.gov/Version4/Release3/
if ( -d ${basedir}/run) then
echo 'Directory ' ${basedir} '/run exists.'
echo 'Please remove it and re-submit the job.'
exit 1
endif
mkdir ${basedir}/run
cd ${basedir}/run
mkdir diags

ln -s ${inputdir}/input_init/NAMELIST/* .
ln -s ${inputdir}/input_init/error_weight/ctrl_weight/* .
ln -s ${inputdir}/input_init/error_weight/data_error/* .
ln -s ${inputdir}/input_init/* .
ln -s ${inputdir}/input_ecco/*/* .
ln -s ${inputdir}/input_forcing/eccov4r3* .

cp -p ../build/mitgcmuv .
mpiexec -np ${nprocs} dplace ./mitgcmuv
```

5.2.1 Stop and restart a run

Normally, the model should be run non-stop for the whole time period. If one has to stop and then restart the model for any reason, for each restart one has to set the correct time step and to disable the initial UVTS and ssh control adjustments, which essentially involves modification of the two namelist files: data and data.ctrl. Note that the pickup file where the model stops should have

been generated with v4r3's configuration. When one restarts the model, the run directory should contain a copy of or a link to this pickup file.

To set the correct time step number for the restart, one has to set the variable "nIter0" in the namelist file "data" to the number of time step at which the model stop previously.

For users' convenience, a version of the namelist file "data.ctrl" that disables the initial control adjustments is provided in ftp://ecco.jpl.nasa.gov/Version4/Release3/input_init/NAMELIST/data.ctrl.restart. When to restart the model, one can unlink data.ctrl and rename/copy data.ctrl.restart to data.ctrl.

6 Tiled netCDF files

Release 3's diagnostic outputs available at

`ftp://ecco.jpl.nasa.gov/Version4/Release3/`

are generally in tiled netCDF format. The global fields are saved into 13 netCDF files, with each file accounting for one of the 13 regional tiles (cf. Wang et al., 2017). The tiled netCDF files were converted offline from flat binary model outputs by using a utility in the Matlab toolbox gcmfaces (Gael, 2017).

6.1 How to convert binary to tiled netCDF files

One needs to first install gcmfaces package following the instruction of the gcmfaces' user guide (Gael, 2017).

The following MATLAB script converts binary model diagnostic outputs to tiled netCDF format.

```
clear all, close all, clc
cd /my_gcmfaces_dir/

%declare global variables and initiate some variables
gcmfaces global;
global mygrid; mygrid=[];

%load grid
grid_load;

%go to the directory where the binary model files reside
rundir = '/my_model_binaryfiles_dir/';
cd rundir;

%assume that the binary model diagnostic files are named
% as "state_2d_set1*" and stored in
% /my_model_binaryfiles_dir/diags/STATE/.
```

```
% note that although one may use a different file name
% other than "state_2d_set1*" for the diagnostic files,
% the directory structure should be as follows:
% /my_model_binaryfiles_dir/diags/STATE/
% /my_model_binaryfiles_dir/diags/BUDG/
% /my_model_binaryfiles_dir/diags/TRSP/.
% The subdirecories are the default directories that
% gcmfaces look for the binary model diagnostics.
% we also assume that each of the "state_2d_set1*" files
% contains the following three variables
% 'ETAN', 'PHIBOT', and 'SIarea' for model
% sea-level, model ocean bottom pressure and sea-ice
% concentration.

%to convert all three variables to 13 tiled netCDF files
% with each tile a size of 90x90 model grids.
process2nctiles(rundir,'state_2d_set1',{}, [90 90]);

%to convert the variable 'ETAN' only
process2nctiles(rundir,'state_2d_set1',{'ETAN'}, [90 90]);

%output directories of tiled netCDF files
% are /my_model_binaryfiles_dir/nctiles_tmp/ETAN, PHIBOT,
% and SIarea.
```

7 Concluding remarks

If there are any questions, please contact us at ecco-support@mit.edu (please subscribe via http://mailman.mit.edu/mailman/listinfo/ecco-support)

8 Acknowledgement

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

9 References

Forget, G., 2017: gcmfaces, A generic treatment of gridded earth variables in Matlab,

ftp://mit.ecco-group.org/ecco_for_las/version_4/release2/doc/ECCOV4R2_gcmfaces.pdf

Wang, O., I. Fukumori, and I. Fenty, 2017: An Overview of ECCO Version 4 Release 3's ftp site ftp://ecco.jpl.nasa.gov/Version4/Release3/. (Available at

`ftp://ecco.jpl.nasa.gov/Version4/Release3/doc/v4r3_overview.pdf)`