

Instructions for reproducing ECCO Version 4 Release 4

v1.3 2023-04-17

Ou Wang and Ian Fenty

NASA Jet Propulsion Laboratory/California Institute of Technology

© 2023 California Institute of Technology. Government sponsorship acknowledged.

1. Introduction

These instructions describe how to reproduce the ECCO Version 4, Release 4 (V4r4) ocean and sea-ice state estimate (ECCO Consortium, et al., 2021)

The daily and monthly-mean fields from the V4r4 solution are available from NASA's Physical Oceanography Distributed Active Archive Center (PO.DAAC), <https://podaac.jpl.nasa.gov/ECCO>. Instructions for downloading the solution from PO.DAAC can be found on the ECCO website, <https://ecco-group.org/products-ECCO-V4r4.htm>.

The document is organized as follows. Section 2 describes how to obtain the core source code of the [MIT general circulation model](#) (MITgcm). Section 3 describes how to download source code and model configuration files that are specific to the V4r4 solution. Section 4 deals with downloading the model's atmospheric forcing, initial conditions, and other needed files. In section 5, we give an example of how to compile and run the model. Section 6 explains how to stop and restart the simulation. Finally, Section 7 describes how to customize which fields are saved during the model simulation.

2. Download the MITgcm source code

The ECCO V4r4 state estimate is a free-running solution of the MIT general circulation model. Reproducing the state estimate requires compiling and running the model source code version *checkpoint 66g*, which is the same code version used to produce the solution.

Warning: The MITgcm code is always evolving and therefore you must use checkpoint 66g to exactly reproduce the solution.

The MITgcm source code *checkpoint 66g* is available from the MITgcm GitHub repository. Begin by first creating a working directory, here labelled **WORKINGDIR**. All directories and files downloaded hereafter will be inside your **WORKINGDIR**.

```
mkdir WORKINGDIR
cd WORKINGDIR
```

Clone the MITgcm *checkpoint66g* repository using *git*. The *git clone* command will create a subdirectory called **MITgcm**.

```
git clone https://github.com/MITgcm/MITgcm.git -b checkpoint66g
```

Reproducing ECCO V4r4

Alternatively, you can download the MITgcm source code using *wget* (or equivalent), unzip, and rename as follows:

```
wget https://github.com/MITgcm/MITgcm/archive/refs/tags/checkpoint66g.zip
unzip checkpoint66g.zip
mv MITgcm-checkpoint66g MITgcm
```

Your directory structure should now look like this:

```
WORKINGDIR
└─ MITgcm
```

3. Download specific code and run-time parameters

In addition to the core MITgcm *checkpoint 66g* code, some additional MITgcm code files and the model's run-time parameter files (also referred to as "namelist" files) are needed. These files are available from the ECCO-GROUP GitHub repository:

<https://github.com/ECCO-GROUP/ECCO-v4-Configurations/tree/master/ECCOv4%20Release%204>

From **WORKINGDIR**, clone the **ECCO-v4-Configurations** repository, then make a directory for running the model, here labelled **ECCOv4/release4**. Finally, copy the **code/** and **namelist/** directories to the **ECCOv4/release4** directory:

```
cd WORKINGDIR
git clone https://github.com/ECCO-GROUP/ECCO-v4-Configurations.git
mkdir -p ECCOv4/release4
cp -r "ECCO-v4-Configurations/ECCOv4 Release 4/code/" ECCOv4/release4/code
cp -r "ECCO-v4-Configurations/ECCOv4 Release 4/namelist/" ECCOv4/release4/namelist
```

Your directory structure should now look like this:

```
WORKINGDIR
├─ ECCO-v4-Configurations
├─ ECCOv4
│   └─ release4
│       ├── code
│       └─ namelist
└─ MITgcm
```

4. Download atmospheric forcing, initial conditions, and other input files

ECCO V4r4's atmospheric forcing, model initial conditions, and other required input files are available in a dataset, called **ECCO Ancillary Data (Version 4 Release 4)**, hosted on PO.DAAC. The ancillary dataset has 8 file archives (gzipped tarballs), 6 of which are required (see *Table 1*). Combined file size of these 6 files is approximately 200 GB. The dataset "short name" is **ECCO_L4 Ancillary Data V4R4** and its PO.DAAC url is:

https://podaac.jpl.nasa.gov/dataset/ECCO_L4 Ancillary Data V4R4

Table 1: The 6 files from the ECCO V4r4 ancillary dataset required to reproduce the solution

| FILENAME | DESCRIPTION |
|----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ancillary_data_doc_ECCO_V4r4.tar.gz</code> | documentation |
| <code>ancillary_data_input_init_ECCO_V4r4.tar.gz</code> | files required to initialize the model |
| <code>ancillary_data_input_forcing_ECCO_V4r4.tar.gz</code> | atmospheric boundary conditions and other fields used to force the model |
| <code>ancillary_data_native_grid_files_ECCO_V4r4.tar.gz</code> | files defining the model grid geometry |
| <code>ancillary_data_data_constraints_ECCO_V4r4.tar.gz</code> | observational data used to constrain the model. Except for in situ data, data are mapped to the model grid as single precision flat binary format. The in-situ data are provided in NetCDF format. |
| <code>ancillary_data_misc_ECCO_V4r4.tar.gz</code> | miscellaneous files |

An account with NASA Earthdata system is required to download these files. New Earthdata accounts can be made here: <https://urs.earthdata.nasa.gov/home>

There are several different ways to get these files. One way is to navigate to the Earthdata webpage for the dataset and simply click on each file one at a time to initiate transfers, <https://cmr.earthdata.nasa.gov/virtual-directory/collections/C2096684707-POCLOUD/temporal/1992/01/01>

Another way is to navigate to the “Earthdata Search” page associated with the dataset, and then click on the download icons associated with each of the 6 files: <https://search.earthdata.nasa.gov/search/granules?p=C2096684707-POCLOUD>

Detailed instructions for **batch** downloading these files from the command line (using *wget*) or from within a Jupyter Python notebook are provided on the ECCO-GROUP GitHub repository: https://github.com/ECCO-GROUP/ECCO-ACCESS/blob/master/PODAAC/Downloading_ECCO_datasets_from_PODAAC/README.md

Alternatively, because there are only 6 files, you can download them one at a time using *wget* by following the instructions below. Note: replace **YOURUSERNAME** with *your* Earthdata login and use your Earthdata password when prompted.

```
cd WORKINGDIR/ECCOV4/release4
mkdir input
cd input
wget --user YOURUSERNAME --ask-password https://archive.podaac.earthdata.nasa.gov/podaac-ops-cumulus-protected/ECCO_L4_ANCILLARY_DATA_V4R4/ancillary_data_misc_ECCO_V4r4.tar.gz
wget --user YOURUSERNAME --ask-password https://archive.podaac.earthdata.nasa.gov/podaac-ops-cumulus-protected/ECCO_L4_ANCILLARY_DATA_V4R4/ancillary_data_doc_ECCO_V4r4.tar.gz
wget --user YOURUSERNAME --ask-password https://archive.podaac.earthdata.nasa.gov/podaac-ops-cumulus-protected/ECCO_L4_ANCILLARY_DATA_V4R4/ancillary_data_native_grid_files_ECCO_V4r4.tar.gz
wget --user YOURUSERNAME --ask-password https://archive.podaac.earthdata.nasa.gov/podaac-ops-cumulus-protected/ECCO_L4_ANCILLARY_DATA_V4R4/ancillary_data_input_forcing_ECCO_V4r4.tar.gz
wget --user YOURUSERNAME --ask-password https://archive.podaac.earthdata.nasa.gov/podaac-ops-cumulus-protected/ECCO_L4_ANCILLARY_DATA_V4R4/ancillary_data_data_constraints_ECCO_V4r4.tar.gz
wget --user YOURUSERNAME --ask-password https://archive.podaac.earthdata.nasa.gov/podaac-ops-cumulus-protected/ECCO_L4_ANCILLARY_DATA_V4R4/ancillary_data_input_init_ECCO_V4r4.tar.gz
```

Reproducing ECCO V4r4

Once you have the 6 archives, use the following command to extract the files, and delete the original archives (optional).

```
tar xzvf ancillary_data_misc_ECCO_V4r4.tar.gz
tar xzvf ancillary_data_doc_ECCO_V4r4.tar.gz
tar xzvf ancillary_data_native_grid_files_ECCO_V4r4.tar.gz
tar xzvf ancillary_data_input_forcing_ECCO_V4r4.tar.gz
tar xzvf ancillary_data_data_constraints_ECCO_V4r4.tar.gz
tar xzvf ancillary_data_input_init_ECCO_V4r4.tar.gz
(optional) rm ancillary_data* ECCO_V4r4.tar.gz
```

Your directory structure should now look like this:

```
WORKINGDIR
├── ECCO-v4-Configurations
├── ECCOV4
│   ├── release4
│   │   ├── code
│   │   └── input
│   │       ├── data_constraints
│   │       ├── doc
│   │       ├── input_forcing
│   │       ├── input_init
│   │       ├── misc
│   │       └── native_grid_files
│   └── namelist
└── MITgcm
```

5. Compile and run

The compilation and run of the model are compiler and platform specific. In this section we show how to compile and run the model on the NASA's Pleiades Supercomputer as of Jan 2022.

5.1 Compile

Before compiling the code and running the model, one has to load the required modules, including compiler and NetCDF packages, with the following commands:

```
module purge
module load comp-intel/2020.4.304
module load mpi-hpe/mpt.2.25
module load hdf4/4.2.12
module load hdf5/1.8.18_mpt
module load netcdf/4.4.1.1_mpt
```

The modules listed above are the current versions of the available modules on Pleiades as of January 14, 2022. It is generally recommended to use newer versions of modules as available.

The commands to compile are listed below. A set of system-specific "build options" are specified in `WORKINGDIR/ECCOV4/release4/code/linux_amd64_ifort+mpi_ice_nas`

Reproducing ECCO V4r4

```
cd WORKINGDIR/ECCOV4/release4
mkdir build
cd build
export ROOTDIR=../../MITgcm
../../MITgcm/tools/genmake2 -mods=../code
    -optfile=../code/linux_amd64_ifort+mpi_ice_nas -mpi
make -j16 depend
make -j16 all
cd ..
```

Note: use the full path of **WORKINGDIR**. If compiled successfully, the model executable will be **WORKINGDIR/ECCOV4/release4/build/mitgcmuv**.

Warning: *Different systems require different build options! Example build option files for several platforms and compilers can be found in MITgcm/tools/build_options/ You may need to make a build options file suitable for your own system in order for the Makefile to use appropriate compiler options and to find required libraries.*

The model needs NetCDF installed to read some constraining in situ data that are in NetCDF format. The compiler (the genmake2 command above) will check if NetCDF is correctly installed and linked, and print out a warning message (see below) if otherwise.

WARNING: the "profiles" package was enabled but tests failed to compile NetCDF applications. Please check that:

- 1) NetCDF is correctly installed for this compiler and*
- 2) the LIBS variable (within the "optfile") specifies the correct NetCDF library to link against.*

Due to this failure, the "profiles" package is now DISABLED.

5.2 Run

An example of the run script for Pleiades (called `pbs_eccoV4r4.csh`) is included at the end of this document. The example PBS script will reproduce V4r4's 26-yr results from 1992 through 2017 in one pass (without stopping). The requested wall clock time is 24 hours (`#PBS -l walltime=24:00:00`). While 24 hours is usually sufficient to finish the 26-year model integration period on Pleiades, one may have to increase the job run time on a different machine or when increasing the amount of model diagnostic output.

To submit the job to the PBS scheduler, run the command:

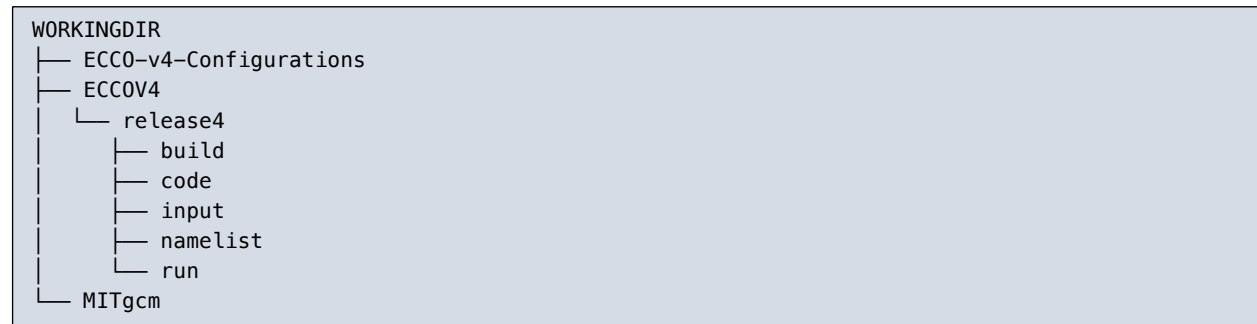
```
qsub pbs_eccoV4r4.csh
```

The job may wait for some time in the queue before executing. The run directory will be created by the `pbs_eccoV4r4.csh` script (once after it is out of the queue). The name of the run directory

Reproducing ECCO V4r4

will be `WORKINGDIR/ECCOV4/release4/run`. To change the name of the run directory, simply modify the csh script.

After running, your directory structure should look like this:



6. Stopping and restarting a run

Normally, the model simulation should be run non-stop for the entire time period. However, the simulation can be interrupted and then restarted partway through by making use of a “pickup” file. MITgcm pickup files are snapshots of the model state which can serve as model initial conditions. To learn more about how to use pickup files in the MITgcm, see: https://mitgcm.readthedocs.io/en/latest/getting_started/getting_started.html?highlight=pickup#restart-pickup-files

The normal ECCO model simulation starts from a pickup file corresponding with time step 1: `WORKINGDIR/ECCOV4/release4/input/input_init/pickup.0000000001.data`

By default, the model will write a new pickup file at the calendar end of each simulated year. The integer number associated with the pickup file (e.g., `0000000001`) corresponds with the model time step at the moment of the snapshot. Each new pickup file will increment by approximately 8760 (there are approximately 8760 1-hour model time steps per year).

To restart the model from a pickup file other than `pickup.0000000001.data`, you must (1) tell the model to use the time step corresponding with the integer of the pickup file and (2) disable the control adjustments to the model’s initial conditions of `U`, `V`, `T`, `S`, and `ssh`. Both steps require modifying run-time namelist files in the `WORKINGDIR/ECCOV4/release4/namelist/` directory.

To update the model’s initial time step, change the parameter `nIter0` in the `data` namelist file. To disable the control adjustments to the model’s initial conditions modify the `data.ctrl` namelist file. For users’ convenience, we provide an example `data.ctrl` specifically for restarting from a different pickup file: `data.ctrl.restart`. When restarting the model, unlink the default `data.ctrl` and rename/copy `data.ctrl.restart` to `data.ctrl`.

7. Configuring custom diagnostic output

Various model fields (e.g., temperatures, salinities, velocities, fluxes) can be output by the model when re-running the simulation using the MITgcm’s *diagnostic* package. The full list of pre-defined diagnostic fields that one can output is provided in here:

`WORKINGDIR/ECCO-v4-Configurations/ECCOV4 Release 4/doc/available_diagnostics.log`

Users can specify which diagnostic fields to output in the `data.diagnostics` runtime parameter file: `WORKINGDIR/ECCOV4/release4/namelist/data.diagnostics`

Reproducing ECCO V4r4

The diagnostic package can average fields over different time periods, provide snapshots of the model state, and even perform some statistical calculations. Interested users can even define their own diagnostics. More information about how to use the package can be found in *Chapter 7* of the MITgcm user manual, “Diagnostics and I/O - Packages II, and Post-Processing Utilities”: https://mitgcm.readthedocs.io/en/latest/outp_pkgs/outp_pkgs.html

8. Concluding remarks

If you have any questions, please contact us at ecco-support@mit.edu.

You can also subscribe to the mailing list <http://mailman.mit.edu/mailman/listinfo/ecco-support>

9. Acknowledgements

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

10. References

ECCO Consortium, Fukumori, I., Wang, O., Fenty, I., Forget, G., Heimbach, P., & Ponte, R. M. (2021, February 10). Synopsis of the ECCO Central Production Global Ocean and Sea-Ice State Estimate (Version 4 Release 4). <https://doi.org/10.5281/zenodo.4533349>

Wang, O. (2017). Instructions for reproducing ECCO Version 4 Release 3. Available at https://ecco.jpl.nasa.gov/drive/files/Version4/Release3/doc/v4r3_reproduction_howto.pdf.

Reproducing ECCO V4r4

Example PBS script (pbs_eccoV4r4.csh) for submission to the NASA Electra Supercomputer

```
#PBS -S /bin/csh
#PBS -l select=3:ncpus=40:model=sky_ele
#PBS -l walltime=24:00:00
#PBS -j oe
#PBS -o ./
#PBS -m bea

limit stacksize unlimited

module purge
module load comp-intel/2020.4.304
module load mpi-hpe/mpt.2.25
module load hdf4/4.2.12
module load hdf5/1.8.18_mpt
module load netcdf/4.4.1.1_mpt
module list

setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}
setenv FORT_BUFFERED 1
setenv MPI_BUFS_PER_PROC 128
setenv MPI_DISPLAY_SETTINGS

set nprocs = 96
set basedir = ./
set inputdir = ../input/

if ( -d ${basedir}/run) then
echo 'Directory "run" exists.'
echo 'Please rename/remove it and re-submit the job.'
exit 1
endif
mkdir ${basedir}/run
cd ${basedir}/run

ln -s ../namelist/* .
ln -s ${inputdir}/input_init/error_weight/data_error/* .
ln -s ${inputdir}/input_init/* .
ln -s ${inputdir}/misc/tools/mkdir_subdir_diags.py .
ln -s ${inputdir}/data_constraints/data_error/** .
ln -s ${inputdir}/data_constraints/** .
ln -s ${inputdir}/input_forcing/adjusted/eccov4r4* .
ln -s ${inputdir}/input_forcing/other/*.bin .
ln -s ${inputdir}/input_forcing/control_weights/* .
ln -s ${inputdir}/native_grid_files/tile*.mitgrid .

python mkdir_subdir_diags.py

cp -p ../build/mitgcmuv .
mpiexec -np ${nprocs} /u/scicon/tools/bin/mbind.x ./mitgcmuv
```