

# Using ECCO v4

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

May 23, 2016

## abstract

These notes pertain to the ECCO v4 state estimate, model setup, and associated codes (Forget et al., 2015). Section 1 summarizes download procedures and links to additional documentation<sup>1</sup>. Section 2 explains how ECCO v4 solutions, or corresponding short regression tests, can be re-run.

## Contents

<b>1</b>	<b>Downloading ECCO v4</b>	<b>2</b>
1.1	Released ECCO v4 Solution . . . . .	2
1.2	Diagnostic Tools . . . . .	2
1.3	ECCO v4 setup . . . . .	2
<b>2</b>	<b>Running ECCO v4</b>	<b>3</b>
2.1	Baseline ECCO v4 solution . . . . .	4
2.2	Other ECCO v4 Solutions . . . . .	6
2.3	Short Regression Tests . . . . .	6
2.4	Iterative Optimization Test Case . . . . .	7

## References

Forget, G., J.-M. Campin, P. Heimbach, C. N. Hill, R. M. Ponte, and C. Wunsch, 2015: ECCO version 4: an integrated framework for non-linear inverse modeling and global ocean state estimation. *Geoscientific Model Development*, **8** (10), 3071–3104, doi:10.5194/gmd-8-3071-2015, URL <http://www.geosci-model-dev.net/8/3071/2015/>.

Forget, G., J.-M. Campin, P. Heimbach, C. N. Hill, R. M. Ponte, and C. Wunsch, 2016: ECCO version 4: Second release. URL <http://hdl.handle.net/1721.1/102062>.

---

<sup>1</sup>Throughout this document links are indicated by blue colored font.

# 1 Downloading ECCO v4

This section first provides direction to download the ECCO v4 state estimate output (section 1.1) and associated matlab analysis tools (section 1.2). It then explains download procedures for the ECCO v4 model setup and MITgcm (section 1.3).

## 1.1 Released ECCO v4 Solution

The model output for the ECCO v4-release 2 state estimate (Forget et al., 2016) is available via [this opendap server](#) and [this ftp server](#) from [ecco-group.org](#). The servers provide the [grid files](#) and [monthly output fields](#) in ‘nctiles’ format, as well as [collocated in situ and state estimate profiles](#) in ‘MITprof’ format. The ‘nctiles’ and ‘MITprof’ format are described in Forget et al. (2015). The files can be downloaded at the command line, e.g. within a linux environment, by typing

```
wget --recursive ftp://mit.ecco-group.org/ecco_for_las/version_4/release2/nctiles_grid
wget --recursive ftp://mit.ecco-group.org/ecco_for_las/version_4/release2/nctiles_monthly
wget --recursive ftp://mit.ecco-group.org/ecco_for_las/version_4/release2/profiles
```

## 1.2 Diagnostic Tools

To analyze model output from section 1.1 or section 2.1, two sets of Matlab tools are available:

- The [gcmfaces+MITprof](#) framework (Forget et al., 2015) gets installed as explained in [the gcmfaces.pdf](#) documentation. This software can be used, for example, to generate the diagnostics provided in Forget et al. (2016) for ECCO v4-release 2 (its ‘standard analysis’ as defined in Forget et al. 2015) from the ECCO v4 output.
- Basic MITgcm [tools](#) can also be downloaded via [cvs](#).

## 1.3 ECCO v4 setup

First, install the MITgcm using cvs as explained at [this site](#). Second, install the ECCO v4 model setup on the LLC90 and CS32 grids (see Forget et al., 2015) also via [the MITgcm cvs server](#):

```
cd MITgcm/verification
cvs co -P -d global_oce_llc90 MITgcm_contrib/gael/verification/global_oce_llc90
cvs co -P -d global_oce_cs32 MITgcm_contrib/gael/verification/global_oce_cs32
cd global_oce_llc90/input_fields/
./gunzip_files
cd ../../
```

As an alternative to using [the MITgcm cvs server](#), frozen versions are available at [this site](#) (c65w\_verif.tar.gz is currently the latest). [global\\_oce\\_cs32/](#) (614M) is a small setup used only for testing, whereas [global\\_oce\\_llc90/](#) (595M) is the production setup that typically runs on 96 processors. It is advised to always download or update the two setups together since they are tied to each other for the purpose of section 2.3 even though only [global\\_oce\\_llc90/](#) is needed for the purpose of section 2.1. Running and verifying the section 2.1 solutions furthermore requires downloading the three-hourly forcing fields (96G) and observational data inputs (25G) from:

```

37 cd MITgcm/verification
38 wget --recursive ftp://mit.ecco-group.org/ecco_for_las/version_4/release2/input_forcing/
39 wget --recursive ftp://mit.ecco-group.org/ecco_for_las/version_4/release2/input_ecco/
40 mv mit.ecco-group.org/ecco_for_las/version_4/release2/input_forcing forcing_baseline2
41 mv mit.ecco-group.org/ecco_for_las/version_4/release2/input_ecco inputs_baseline2

```

42 Running the section 2.3 regression tests instead requires ‘global\_oce\_input\_fields/’ (1.6G):

```

43 cd MITgcm/verification
44 wget http://mitgcm.org/~gforget/global_oce_input_fields.tar.gz
45 gunzip global_oce_input_fields.tar.gz
46 tar xf global_oce_input_fields.tar
47 \rm -f global_oce_input_fields.tar

```

## 48 2 Running ECCO v4

49 This section explains how the ECCO v4 setup is used to re-run the 20-year state estimate (section  
50 2.1) or short regression tests (section 2.3). As a pre-requisite, one must have downloaded the  
51 MITgcm as well as the ECCO v4 model setup and inputs (section 1.3). Based upon the section  
52 1.3 directions, the various downloaded directories should be organized as illustrated in Fig.1  
53 within the MITgcm directory. Running MITgcm furthermore requires the following software:  
54 gcc, gfortran (or alternatives), mpi (only for parallel runs) and netcdf (only if ‘pkg/profiles’ is  
55 used). Additional information can be found in [the MITgcm howto](#) and in [the MITgcm manual](#).

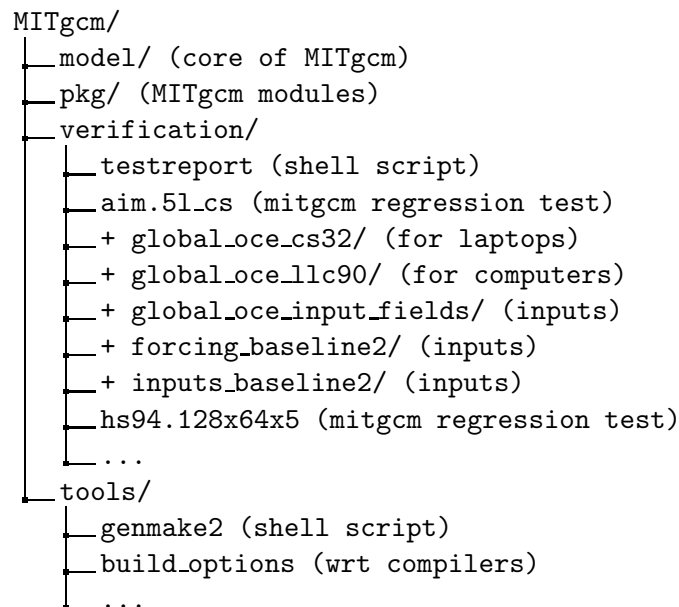


Figure 1: MITgcm directory structure including the ECCO v4 directories (indicated with “+”) downloaded according to the section 1.3 directions.

## 56 2.1 Baseline ECCO v4 solution

57 The ‘ECCO v4-release 2’ state estimate for 1992-2011 (documented in Forget et al., 2016) is a  
58 minor update of ‘ECCO v4-release 1’ (Forget et al., 2015) that is easiest for outside users to  
59 re-run and further benefits from a few additional corrections (listed in Forget et al., 2016). Users  
60 can re-run ‘ECCO v4-release 2’ by following the directions in Fig. 2.

61 A 20-year ECCO v4 model run typically takes between 6 to 12 hours on 96 cores (depending  
62 on the computing environment). To verify the re-run results one proceeds according to Fig. 3.  
63 The expected level of accuracy for 20-year re-runs (with an up to date MITgcm; on any given  
64 computer) is reached when the displayed values are all  $\leq -3$  (see Forget et al., 2015, for details).

65 The number of cores (96 by default and in Fig. 2) can be reduced to, e.g., 24 by copying  
66 ‘global\_oce\_llc90/code/SIZE.h\_24cores’ over ‘global\_oce\_llc90/code/SIZE.h’ before compiling the  
67 model and then running it with ‘mpiexec -np 24 ./mitgcmuv’. Different compiler options (alter-  
68 natives to ‘linux\_amd64\_gfortran’ in Fig. 2) are available in ‘MITgcm/tools/build\_options’.

```
#1) compile the model
cd MITgcm/verification/global_oce_llc90/build
../../../../tools/genmake2 -mods=../code -optfile \
    ../../../../tools/build_options/linux_amd64_gfortran -mpi
make depend
make -j 4

#2) link files into run directory
cd ../run
ln -s ../build/mitgcmuv .
ln -s ../input.ecco_v4/* .
ln -s ../input_fields/* .
ln -s ../../inputs_baseline2/input*/* .
ln -s ../../forcing_baseline2 .

#3) run model
mpiexec -np 96 ./mitgcmuv
```

Figure 2: Procedure to re-run the baseline ECCO v4 state estimate (‘ECCO v4-release 2’; Forget et al. 2016). Pre-requisites: (1) installation of gcc, gfortran (or alternatives), and mpi (only for parallel runs); (2) installation of the MITgcm and ECCO v4 setup installation according to section 1.3. The contents of ‘input.ecco\_v4’ (short text files) and ‘input\_fields’ (grid and other binary input) should match [this site](#). The contents of ‘forcing\_baseline2’ directory should match [this site](#). The contents of ‘inputs\_baseline2’ should match [this site](#). These files can be downloaded freely, e.g., as explained in section 1.3.

Figure 3: Top: instructions to verify (using ‘testreport\_ecco.m’ within Matlab) that a re-run of the ECCO v4 state estimate is acceptably close to the reference result (‘baseline2’). Bottom: example output from testreport\_ecco.m where the re-run agrees up to 6 digits with the reference result. To activate additional tests (of meridional transports) one needs to have installed [gcmfaces](#) (see section 1.2) and uncommented the ‘addpath’ and ‘gcmfaces\_global’ commands below (where ‘ /Documents/MATLAB/gcmfaces’ is a user specific path).

```
cd MITgcm/verification/global_oce_llc90
matlab -nodesktop -nodisplay

%addpath ~/Documents/MATLAB/gcmfaces;
%gcmfaces_global;

addpath results_itXX;%necessary .m and .mat files
mytest=testreport_ecco('run/');%compute the tests and display result
```

```
-----
      & jT & jS &      ... & (reference is)
run/  & (-6) & (-6) &      ... & baseline2
-----
```

69 **2.2 Other ECCO v4 Solutions**

70 Users can also easily re-run ‘baseline1’ that most closely matches the ‘release1’ from section 1.1.

71 To re-run ‘baseline1’ instead of ‘release2’ a few modifications to the setup are needed:

72

73 (a) get the corresponding forcing fields per

74 `wget --recursive ftp://mit.ecco-group.org/ecco_for_las/version_4/release1/forcing_baseline1/`

75 (b) before compiling the model: define ‘ALLOW\_KAPGM\_CONTROL\_OLD’ and

76 ‘ALLOW\_KAPREDI\_CONTROL\_OLD’ in ‘global\_oce\_llc90/code/GMREDI\_OPTIONS.h’;

77 define ‘ALLOW\_AUTODIFF\_INIT\_OLD’ in ‘global\_oce\_llc90/code/AUTODIFF\_OPTIONS.h’;

78 (c) before running the model: copy ‘global\_oce\_llc90/input\_itXX/data’ and ‘data.exf’ over

79 ‘global\_oce\_llc90/input.ecco\_v4/data’ and ‘data.exf’.

80

81 Users who may want to reproduce ‘release1’ even more precisely than ‘baseline1’ does should  
82 contact ecco-support@mit.edu to obtain additional model inputs. Users holding a TAF license  
83 can also: (a) compile the adjoint by replacing ‘make -j 4’ with ‘make adall -j 4’ in Fig. 2; (b)  
84 activate the adjoint by setting ‘useAUTODIFF=.TRUE.’ in data.pkg; (c) run the adjoint by  
85 replacing ‘mitgcmuv’ with ‘mitgcmuv\_ad’ in Fig. 2.

86 **2.3 Short Regression Tests**

87 To ensure continued compatibility with the up to date MITgcm, the ECCO v4 model setup  
88 is also tested on a daily basis using the ‘testreport’ command line utility (indicated in Fig.1)  
89 that compares re-runs with reference results over a few time steps. The reader is referred to  
90 ‘testreport -help’ and the MITgcm howto for details. The short regression test of the smaller  
91 ECCO v4 setup (global\_oce\_cs32/) is thus executed by typing:

92 `./testreport -t global_oce_cs32`

93 If everything proceeds as expected then the regression test results are reported to screen  
94 as shown in Fig. 4. The daily results of the regression tests (forward & adjoint, cs32 & llc90,  
95 gfortran & 24 processors, on ‘glacier’ cluster) are reported on this site. On other machines the  
96 degree of agreement (16 digits in the Fig. 4 example) may vary and ‘testreport’ may indicate  
97 ‘FAIL’. Despite the dramatic character of such message, it may not prevent reproducing 20-year  
98 solutions (see section 2.1) with acceptable accuracy.

```
default 10 ----T----- ----S-----
G D M   c      m s      m s
e p a R g m m e . m m e .
n n k u 2 i a a d i a a d
2 d e n d n x n . n x n .

Y Y Y Y>14<16 16 16 16 16 16 16 16 pass global_oce_cs32
```

Figure 4: Abbreviated output of testreport to screen.

99 It should be stressed that the bigger ECCO v4 setup ([global\\_oce\\_llc90/](#)) requires at least 12  
100 cores in forward mode (96 in adjoint mode) and therefore should not be run using the above  
101 command (or on a laptop). Instead the [global\\_oce\\_llc90/](#) regression tests use mpi:

```
102 ./testreport -of ../tools/build_options/linux_amd64_gfortran \  
103 -j 4 -MPI 24 -command 'mpiexec -np TR_NPROC ./mitgcmuv' \  
104 -t global_oce_llc90
```

105 with 24 processors and gfortran (these settings may differ on another machine).

106 To prevent users from inadvertently running the llc90 tests in serial mode (e.g. via a './testre-  
107 port' call) the results were moved from their expected location to 'global\_oce\_llc90/results/hidden/'.  
108 To activate the llc90 tests, the 'output\*' files contained in this directory must therefore be soft  
109 linked to 'global\_oce\_llc90/results/'. To further activate the adjoint tests (which require a [TAF](#)  
110 license) one needs to soft link 'code' to 'code\_ad' in [global\\_oce\\_cs32/](#) and [global\\_oce\\_llc90/](#).

111 To slightly reduce memory and storage needs in these short regression tests, one can copy  
112 'global\_oce\_llc90/code/tamc.h\_short' and 'global\_oce\_llc90/code/PROFILES\_SIZE.h\_short' over  
113 'global\_oce\_llc90/code/tamc.h' and 'global\_oce\_llc90/code/PROFILES\_SIZE.h'. It should also  
114 be noted that the llc90 input files in 'global\_oce\_llc90/input\_fields/' need to be un-compressed  
115 (using 'gunzip\_files'; see section [1.3](#)) for the regression tests to proceed as expected.

## 116 2.4 Iterative Optimization Test Case

117 The [global\\_oce\\_cs32/input\\_OI](#) directory implements an iterative optimization test case. It boils  
118 down to optimal interpolation solved by a variational method using the MITgcm adjoint (the  
119 ocean model being replaced with a simple diffusion equation here). The pre-requisites are:

- 120 1. run the adjoint benchmark in [global\\_oce\\_cs32/](#) via testreport (see section [2.3](#)).
- 121 2. Go to MITgcm/lsopt and compile (see section 3.18 of [manual](#)).
- 122 3. Go to MITgcm/optim, replace 'natl\_box\_adjoint' with 'global\_oce\_cs32' in [this Makefile](#),  
123 and compile as explained in section 3.18 of [manual](#). An executable named 'optim.x' should  
124 get created in MITgcm/optim. If otherwise, please contact ecco-support@mit.edu
- 125 4. go to MITgcm/verification/global\_oce\_cs32/input\_OI and type 'source ./prepare\_run'

126 To match the reference results reported in [this file](#), the user should proceed as follows

- 127 1. ./mitgcmuv\_ad > output.txt
- 128 2. ./optim.x > op.txt
- 129 3. increment optimcycle by 1 in data.optim
- 130 4. go back to step #1, to run the next iteration
- 131 5. type 'grep fc costfunction000\*' to display results