Instructions for reproducing microplastic distribution: Distribution of plastics of various sizes and densities in the global ocean from a 3D Eulerian model

Zih-En Tseng

June 13, 2025

1 Introduction

This document serves as instructions for reproducing the simulation results presented in "Distribution of plastics of various sizes and densities in the global ocean from a 3D Eulerian model", and is largely modified from a documented instruction for reproducing the ECCOv4r4 dataset [Wan20]. The complete walk through is divided into five parts:

1. Download the MITgcm source code.
2. Download the ECCOv4r4 dataset (to be more specific, the forcing and initial conditions for MITgcm to reproduce the ECCOv4r4 dataset).
3. Download additional code files (used in step 4) and run-time input files (used in step 5) at https://github.com/zizien1019/reproduce_eccov4r4_online_68o that involve treatments on the passive tracer advection-diffusion equation.
4. Compile both the original MITgcm code and additional code.
5. Conduct simulations with different inputs on particle properties.

2 Download code & data

2.1 MITgcm source code

The backbone of our numerical work is the MITgcm, an ocean model that enables 4D-Var data assimilation. In that way, ECCOv4r4 data were produced with the MITgcm version checkpoint66g, which is available at https://github.com/MITgcm/MITgcm.git. However, we work with MITgcm version checkpoint68o that allows more flexible manipulations on the "ptracers" package compared to earlier versions.

To keep the home directory neat, first, create a working directory named freely, say, WORKDIR. All directories created and files downloaded will be under WORKDIR.

```
$ mkdir WORKDIR
$ cd WORKDIR
```

Then, git clone the MITgcm version checkpoint68o into the working directory:

```
$ git clone https://github.com/MITgcm/MITgcm.git
$ git checkout checkpoint68o
```

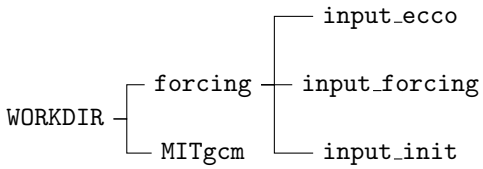
2.2 ECCOv4r4 forcing and other input data

To access the ECCOv4r4 data on a WebDAV server called ECCO Drive at https://ecco.jpl.nasa.gov/drive/files, each user is required to register for an Earthdata account at https://urs.earthdata.nasa.gov/. A detailed help page can be found at https://ecco.jpl.nasa.gov/drive/help. The forcing files required to reproduce ECCOv4r4 state estimate can be downloaded using the following commands:

```
$ wget -P forcing -r --no-parent --user USERNAME --ask-password -nH --cut-dirs=4 \
https://ecco.jpl.nasa.gov/drive/files/Version4/Release4/input_forcing
$ wget -P forcing -r --no-parent --user USERNAME --ask-password -nH --cut-dirs=4 \
https://ecco.jpl.nasa.gov/drive/files/Version4/Release4/input_init
$ wget -P forcing -r --no-parent --user USERNAME --ask-password -nH --cut-dirs=4 \
https://ecco.jpl.nasa.gov/drive/files/Version4/Release4/input_ecco
```

Here USERNAME should be replaced with your Earthdata username. Note that when prompted for password, you should provide the ECCO Drive's WebDAV password, instead of your Earthdata account password. The WebDAV password can be located after logging in ECCO Drive at https://ecco.jpl.nasa.gov/drive/ with an Earthdata account. Once log in, you will be directed to a page reading "Your WebDAV/Programmatic API credentials:", and the ECCO Drive's WebDAV password will be shown in the box following "password:".

Once the downloads are complete, the `WORKDIR` should be structured as follows:

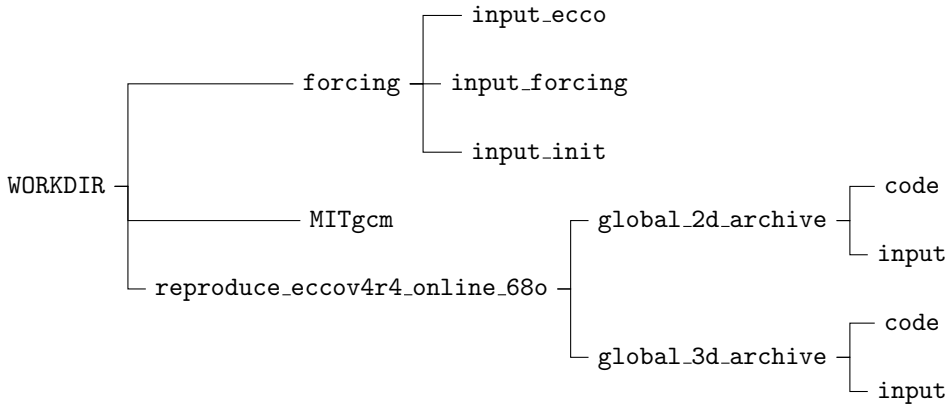


2.3 Additional code files and input setting files

At this point we have acquired the source code of MITgcm and the ECCOv4r4 dataset. The last piece we need are additional modified code files (under "`code`" directory) and model behavior setting files (under "`input`" directory) for the 2D surface and 3D simulations are available at my github repository:

```
$ git clone https://github.com/zizien1019/reproduce_eccov4r4_online_68o
```

Now your working directory should be structured as follows:



Under `global_2d_archive` and `global_3d_archive`, each of them contain one "`code`" directory and one "`input`" directory, and exactly one pair of "`code`" and "`input`" directories are required in one run of simulation. Reference outputs of particle distributions are also available in the repository.

3 Compile & Run

We herein provide commands to compile and run the software. The commands provided in this section were successfully tested on the Great Lakes HPC Cluster in the University of Michigan. Before compiling and running, create another directory, say, `build` and `run`, to contain the make files and

simulation outputs.

```
$ mkdir experiment
$ cd experiment
$ mkdir build
$ mkdir run
```

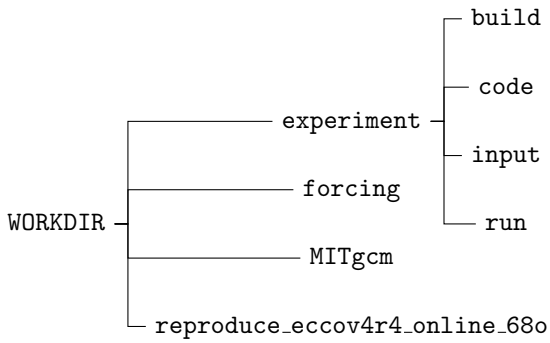
Next, copy the `code` and `input` directories from our github repository that contains all modified files. For example, if we wish to re-run all 3D cases, do:

```
$ cp ../../reproduce_eccov4r4_online_68o/global_3d_archive/code .
$ cp ../../reproduce_eccov4r4_online_68o/global_3d_archive/input .
```

Or, if one wishes to re-run the 2D surface case, replace `global_3d_archive` with `global_2d_archive`, and do, instead:

```
$ cp ../../reproduce_eccov4r4_online_68o/global_2d_archive/code .
$ cp ../../reproduce_eccov4r4_online_68o/global_2d_archive/input .
```

Now, the WORKDIR should be structured as:



3.1 Compile

To build the code, first, make sure the machine follows the big-endian number format, then load the required compiling modules. In this case, we use intel and impi compilers for compiling parallel codes, and the build options are specified in the "optfile" available in
`../../MITgcm/tools/build_options/linux_amd64_ifort+impi`.
Note that a different build option file should be used for running on a different machine or with a different compiler. Modifications in the option file may be necessary to correctly direct to the path where modules are installed on the machine.

```
$ export LANG=en_US.utf8
$ export LC_ALL=en_US.utf8
$ module purge
$ module load intel impi
$ cd build
$ make CLEAN
$ ../../MITgcm/tools/genmake2 -mods=../code -rd=../../MITgcm \
-optfile=../../MITgcm/tools/build_options/linux_amd64_ifort+impi -mpi
$ make depend
$ make all
$ cd ..
```

Once MITgcm is successfully built, one can find the executable "`mitgcmuv`" in the "`build`" directory.

3.2 Run

Finally, the simulation is run under the "`run`" directory. Below is an example script file on the Great Lakes cluster to run a simulation.

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=36
```

```
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=3g
#SBATCH --time=2-00:00:00
$ module purge
$ module load intel
$ module load impi
$ module load hdf5
$ module load netcdf-fortran
$ unset I_MPI_PMI_LIBRARY
$ export I_MPI_JOB_RESPECT_PROCESS_PLACEMENT=0
$ cd experiment/run
$ rm -rf ../run/*
$ ln -s ../../forcing/input_init/error_weight/ctrl_weight/* .
$ ln -s ../../forcing/input_init/error_weight/data_error/* .
$ ln -s ../../forcing/input_init/* .
$ ln -s ../../forcing/input_ecco/input_other/* .
$ ln -s ../../forcing/input_forcing/* .
$ ln -s ../input/* .
$ cp -p ../build/mitgcmuv .
$ mpiexec -np 36 ./mitgcmuv > a.log
```

References

[Wan20] Ou Wang. Instructions for reproducing ecco version 4 release 4. https://ecco-group.org/docs/v4r4_reproduction_howto.pdf, 2020. Accessed: 2024-11-16.