A Quick Introduction to TMM

Tatsuro Tanioka

December 17, 2020

Abstract

This document goes through how to use the Transport Matrix Model (TMM) using computational resources at the University of Minnesota's Minnesota Supercomputing Institution (MSI). All the source codes and documents (including this tutorial) that Tanioka made to TMM is available at my github website (https://github.com/tanio003/tmm/tree/TT_Release).

Contents

1 Setting up

1.1 Flow Chart

Before you do anything, read "README.txt" by Samar Khatiwala at the following website: https://github.com/samarkhatiwala/tmm. I will go over each of these steps specifically aimed at audiences using the computational cluster *Mesabi*.

1. Installing and configuring PETSc
2. Downloading all the scripts and transport matrices into your own local directory
3. Compiling the model (we use the BGC model `MOPS2` for this example)
4. Running the model
5. Processing the model outputs
6. Displaying the model outputs

1.2 Steps

1.2.1 Step 0: Logging into MSI and Mesabi

Open the terminal (assuming that you have a MAC or Linux environment) on your computer and log in to MSI with your x500 account:

```
$ ssh -Yt youremail@umn.edu
```

Log in to Mesabi:

```
$ ssh -X mesabi
```

Make a new directory called TMM2 in your home directory and enter into this directory. Everything related to TMM will go into this directory.

```
$ mkdir TMM2
$ cd TMM2
```

1.2.2 Step 1: Installing and configuring PETSc

Download the latest version of PETSc and save it in your TMM2 directory and unzip this package. If opened properly, you should see the new directory petsc-3.13.5.

```
$ wget http://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-lite-3.13.5.tar.gz
$ tar -xvf petsc-lite-3.13.5.tar.gz
$ ls
petsc-3.13.5
```

Import the required modules: (1) impi, (2) impi/intel, and (3) cmake. Also make sure that you are using python 3, not python 2 (= default for MSI).

```
$ module purge
$ module load intel
$ module load impi/intel
$ module load cmake
$ module load python3
$ module list
Currently Loaded Modulefiles:
 1) intel/2018.release(default)   4) cmake/3.10.2(default)
 2) intel/2018/release            5) python3/3.7.1_anaconda
 3) impi/intel(default)
```

Set up `$PETSC_DIR` to your petsc-3.13.5 directory. For your future uses, I would advise you to set `$PETSC_DIR` in your `.bashrc` as well.

```
$ export PETSC_DIR=$HOME/TMM2/petsc-3.13.5
$ echo $PETSC_DIR
/.../TMM2/petsc-3.13.5
```

Configure PETSc. Although this part is quite tricky you can copy and use my config file "`reconfigure-arch-linux-c-opt.py`". If the config file does not work properly, let me know and I can show you a way to compile without using this .py file.

```
$ cd petsc-3.13.5
$ cp ~/../tanio003/TMM2/petsc-3.13.5/config/reconfigure-arch-linux-c-opt.py config/
$ config/reconfigure-arch-linux-c-opt.py
```

Don't worry about some warning signs. It takes few minutes to compile. If it's compiled properly you should see the notice "Conifgure stage complete." Then build PETSc library:

```
$ make all
```

Building process takes about 15-30 minutes. If you're very lucky it will go through in a single shot. But in most cases, it fails during the middle of the process. Don't worry if it fails the first time. Simply type "`$ make all`" again and hopefully it will finish building from where it left off. If built properly, you should see the message "Now to check if the libraries are working do:...". Then type,

```
$ make check
...
Completed test examples
```

If you get this far, you've managed to build the PETSc successfully and you're ready to go to the next step. If it failed, read the error messages, debug, and try again. **Building PETSc is harder than it looks** so you need to be patient.

As more of a technical note, the procedure above uses the Intel compilers and Intel MPI library. By loading the cmake, the PETSc build system can learn more about the host machine. In addition to taking advantage of compiler optimizations and vectorization, the procedure above builds PETSc against the Intel Math Kernel Library (MKL)

for BLAS, LAPACK and ScaLAPACK which gives a performance gain over the reference implementations. For the FORTRAN compiler, we specifically need to use `mpiifort`, and not `mpif90` (the default compiler), because TMM codes are written in both F77 and F90. Also, since we don't require C++ for TMM we put the flag in the config file, `--with-cxx=0`. The reason we need to use MPI compilers, not regular gcc compilers, is because we want to run PETSc in a parallel mode (i.e., by using the command `mpiexec` in the runscript). For more details about building PETSc please check out https://www.mcs.anl.gov/petsc/documentation/installation.html.

1.2.3 Step 2: Downloading all the scripts and transport matrices

1. First download Matlab scripts from http://kelvin.earth.ox.ac.uk/spk/Research/TMM/tmm_matlab_code.tar.gz and put into the first level of your TMM2 folder Path. You could also get my copy.

```
$ cp -r ~/../tanio003/TMM2/tmm_matlab_code $HOME/TMM2/
```

2. Download transport matrices and related data for the model of your choice: http://kelvin.earth.ox.ac.uk/spk/Research/TMM/TransportMatrixConfigs/ and put into he first level of your TMM2 folder Path. You can download all 6 configurations but I warn you that `MITgcm_ECCO_v4` and `UVicKielIncrIsopycDiffTransient` take a very long time. For the ones that I have you could also grab my copy (e.g., to copy `MITgcm_ECCO`):

```
$ cp -r ~/../tanio003/TMM2/MITgcm_ECCO $HOME/TMM2/
```

   Alternatively, you can get download these TMM files from the web directly (e.g., to download `MITgcm_ECCO`):

```
$ wget kelvin.earth.ox.ac.uk/spk/Research/TMM/TransportMatrixConfigs/MITgcm_ECCO.tar
$ tar -xvf MITgcm_ECCO.tar
```

3. Download miscellaneous data called OceanCarbon from http://kelvin.earth.ox.ac.uk/spk/Research/TMM/MiscData/. You can get my copy by:

```
$ cp -r ~/../tanio003/TMM2/OceanCarbon $HOME/TMM2/
```

4. Download source codes for TMM and models:

```
$ git clone https://github.com/tanio003/tmm
```

   This directory (`/TMM2/tmm`) contains the source codes from Khatiwala's master branch ("`master`") and my public release branch ("`TT_Release`"). For our exercise, we will be using some of my new codes so you need to switch from the master branch to my branch in the newly created tmm directory:

```
$ cd tmm
(master) $ ls
driver  HOWTO.txt  LICENSE.txt  models  README.txt
(master) $ git checkout TT_Release
(TT_Release) $ ls
driver  HOWTO.txt  LICENSE.txt  models  README.txt  Tutorial_MSI
```

   Notice that in the branch `TT_Release`, there is a new directory `Tutorial_MSI`, which was not present in the master branch.

   (Optional) If you want to make start making your own changes to the source codes, I would suggest making a new branch in your local computer (e.g.., `yournewrepo`), and leave `master` and `TT_Release` untouched.

```
(TT_Release) $ git checkout -b yournewrepo
(yournewrepo) $ git branch --show-current
yournewrepo
```

5. Set the environment variable TMMROOT to point to the top level of the TMM directory.

```
(TT_Release) $ export TMMROOT=$HOME/TMM2/tmm
(TT_Release) $ echo $TMMROOT
/home/.../TMM2/tmm
```

   For your future convenience, I would advise you to set `$TMMROOT` in your `.bashrc` as well so you don't need to set the variable every time you log in to MSI.

6. To synchronize your `tmm` folder with the remote repository (e.g., syncing `TT_Release` with my updates), use the git pull command your terminal. I suggest you do this regularly to keep your files up to date.

```
$ cd $TMMROOT
$ git checkout TT_Release
$ git pull origin TT_Release
```

1.2.4 Step 3: Compiling the model

Here, let's try compiling the biogeochemical MOPS. If you want to learn about the basic architecture of MOPS, read the model description paper by Kriest and Oschlies (2015) at https://gmd.copernicus.org/articles/8/2929/2015/.

1. For each model there are model-specific source codes in the directory (`$TMMROOT/models/current/mops2/`) and we need to import some of them to the run directory. First, we create a new run directory. I make a new base directory called "Runs" and in that directory, I make subdirectories for specific experiments. I call it `Runs/MOPS/Test_spinup` and copy here all the files needed.

```
$ cd ~/TMM2
$ mkdir -p Runs/MOPS/Test_spinup
$ cd Runs/MOPS/Test_spinup
$ cp -p $TMMROOT/models/current/mops2.0/src/Makefile .
$ cp -p -R $TMMROOT/models/current/mops2.0/matlab/* .
$ cp -p $TMMROOT/models/current/mops2.0/runscripts/* .
$ cp -p $TMMROOT/models/current/mops2.0/parameters/* .
```

2. Compile mops. Make sure that all the modules are loaded and `$PETSC_DIR` is set correctly before you compile mops.

```
$ module load intel
$ module load impi/intel
$ module load cmake
$ make cleanall
$ make mops
```

   If compiled properly, you'd find a new executable "mops" created along with a bunch of objective .o files.

```
$ ls
BGC_INI.o                          n7fluxes28.m
BGC_MODEL.o                        n7physics.m
CAR_CHEM.o                         n7tracers28.m
CAR_INI.o                          n7tracersavg28.m
external_forcing_mops_biogeochem.o perry1996-runoff-noarctic_noname.txt
insolation.o                       perry1996-runoff_noname.txt
load_output.m                      petsc_matvec_utils.o
load_output_time_avg.m             petsc_signal_utils.o
load_pco2.m                        process_output.m
Makefile                           runscript
make_input_files_for_mops_model.m  runscript_msi
make_rivers.m                      tmm_external_bc.o
misfit_mops_biogeochem.o           tmm_forcing_utils.o
mops                               tmm_forward_step.o
mops_biogeochem_copy_data.o        tmm_main.o
mops_biogeochem_diagnostics.o      tmm_monitor.o
mops_biogeochem_ini.o              tmm_profile_utils.o
mops_biogeochem_misfit.o           tmm_timer.o
mops_biogeochem_model.o            tmm_write.o
mops_biogeochem_set_params.o       biogem_params.txt
```

3. Edit the file `make_input_files_for_mops_model.m`. First thing to do is to make sure that variable `base_path` point to the right directory for the TMM configuration.

```
% make_input_files_for_mops_model.m

% Set toplevel path to GCMs configuration
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_2.8deg';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_ECCO';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_ECCO_v4';
base_path='~/TMM2/MITgcm_2.8deg';
 %base_path='~/TMM2/MITgcm_ECCO';

addpath(genpath('~/TMM2/tmm_matlab_code'));% add tmm_matlab_code to the search path
oceanCarbonBasePath='~/TMM2/OceanCarbon';  % add OceanCarbon to the search path
atmosDataPath=fullfile(oceanCarbonBasePath,'AtmosphericCarbonData');
```

   In the same matlab file, there are different switches (0 = no and 1 = yes). For this spin-up exercise, we **couple MOPS to a simple OCMIP-like carbon model** and **fix atmospheric** pCO_2 **at 280 ppm**. So set the switches as following:

```
% make_input_files_for_mops_model.m
...
periodicForcing=1
periodicMatrix=1

dt=43200; % time step to use (43200s for ECCO and MIT2.8; 28800s for any other TMMs)

rearrangeProfiles=1
```

```
bigMat=0
writeFiles=1
writeTMs=1
useCoarseGrainedMatrix=0
writePCFiles=0

READ_SWRAD=0                        % Read short-wave radiation?
useCarbon=1                         % Use simple inorganic carbon model?
useOrgCarbon=0                      % Use organic carbon module?
useAtmModel=0                       % Use prognostic 1-box atmosphere?
pCO2atm_ini=280.0                   % Initial pco2?
useTimeVaryingPrescribedCO2=0       % Use prescribed pco2 pathway?
useVirtualFlux=1                    % Use DIC and Alk to calculate E-P?
empScaleFactor=1.0                  % Scaling factor for E-P (default = 1)
%-----------------------------------
% Modified by Tatsuro Tanioka 200907 to allow for Atmospheric CO2 option
% For a prescribed pCO2 run, useTimeVaryingPrescribedCO2=1 and choose a scenario

% Available options: 'historical', 'RCP3PD', 'RCP45', 'RCP6' and 'RCP85'
co2Scenario='RCP85';
%-----------------------------------
```

Then open MATLAB and run `make_input_files_for_mops_model.m`

```
$ module load matlab
$ matlab -nodesktop
                                  < M A T L A B (R) >
                        Copyright 1984-2019 The MathWorks, Inc.
                    R2019a Update 5 (9.6.0.1174912) 64-bit (glnxa64)
                                    July 31, 2019
To get started, type doc.
For product information, visit www.mathworks.com.

>> make_input_files_for_mops_model
```

This creates a bunch of periodic forcing files (`xxx_01, xxx_02,...`), initial tracer concentrations (`po4ini.petsc, no3ini.petsc,...`), and binary files (.bin and .petsc) related to model geometry and forcing.

4. Edit the parameter file. `biogem_params.txt`. We only need to set the **first 7 parameters** because we are not using organic carbon module for this exercise. We will use the calibrated values for MOPS/MIT2.8 follwiing Kriest et al. (2020).

```
  1.39     #01. detmartin = Martin Curve exponent of Detritus [1.39 for MIT2.8, 1.46 for ECCO]
  173.7    #02. ro2ut = Refield -O2:P                         [173.7 for MIT2.8, 151.1 for ECCO]
  0.00119  #03. nfix = Max growth rate (1/d) of N-fixers      [0.00119 for MIT2.8, 0.00229 for ECCO]
  15.8     #04. subdin = No denitrification below this level of DIN (mmol/m3)          [15.8, 16.0]
  1.01     #05. ACkbaco2 = Half sat-constant for oxic degradation (mmol/m3)            [1.01, 1.07]
  32.0     #06. ACkbacdin = Half sat-constant for suboxic degradation                  [32.0, 23.1]
  0.001    #07. alimit = Min. value for the degradation of PHY and DOP [0.001 for mops, 0.01 for mops_cp]
```

1.2.5 Step 4: Running the model

MSI systems use job queues to efficiently and fairly manage when computations are executed. The queuing system at MSI is called Slurm and to submit a job to a queue users create Slurm job scripts. The script contains information on the resources requested for calculation, as well as the commands for executing the calculation.

Below is the custom Slurm script for submitting a new job to run MOPS2 using Mesabi. It's called `runscript_msi` and should be in the current directory already.

```
#!/bin/bash -l
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=24
#SBATCH --mem-per-cpu=2gb
#SBATCH --time=06:00:00
#SBATCH --mail-type=ALL
#SBATCH --mail-user=tatsurobkkuk@gmail.com
#SBATCH -p small
#SBATCH --output=%j.out

cd $SLURM_SUBMIT_DIR

module load intel
module load impi/intel
module load cmake
```

The first line defines which type of shell the script will be read. Here we will use the `bash`. Commands for the Slurm queuing system begin with `#SBATCH`. Lines 2 to 4 in the above sample script contain the Slurm resource

request. The current job will require about 6 hours, 1 node each with 24 processor cores, and 2GB of memory per core . The two lines containing `#SBATCH --mail` are both commands having to do with sending message emails to the user. The first of these lines instructs the Slurm system to send a message email when the job aborts, begins, or ends. The second command specifies the email address to be used. Using the message emails is recommended because the reason for a job failure can often be determined using information in the emails. A Slurm script should contain the appropriate change directory commands to get to the job execution location. A Slurm script also needs to contain module load commands for any software modules that the calculation might need.

The follwing lines (17~) in `runscript_msi` contain the commands to execute and start a specific program. Different flags need to be change accordingly depending on the nature of experiments. For more information on different options available, read "HOWTO.txt" by S. Khatiwala located in `$TMMROOT`. Do not put these # comments in the actual file.

```
# 360 days per year with a time step of 2 steps per day:
mpirun -np 24 ./mops \ # number of cores, models
        -numtracers 9 \  # number of tracers (i.e. state variables)
        -i po4ini.petsc,dopini.petsc,oxyini.petsc,phyini.petsc,zooini.petsc,detini.petsc,no3ini.petsc,
        dicini.petsc,alkini.petsc \  # files for initialization of BGC state variables
        -me Ae \   # the name of the explicit transport matrix
        -mi Ai \   # the name of the implicit transport matrix
        -t0 0.0 -iter0 0 \
        -deltat_clock 0.0013888888888889 \  # starting time[years] starting time[timesteps]: for initial run
        -max_steps 2160000 \   # total number of timesteps to be evaluated (here 3000 yrs)
        -write_time_steps 72000 \   # output frequency(in timesteps, here every 100 yrs)
        -o po4out.petsc,dopout.petsc,oxyout.petsc,phyout.petsc,zooout.petsc,detout.petsc,no3out.petsc,
        dicout.petsc,alkout.petsc \   # files for output of 9 BGC state variables
        -external_forcing \   # calculate BGC explicitly
        -use_profiles \
        -nzeuph 2 \   # number of layers in euphotic zone (2 for MIT2.8, 6 for ECCO)
        -biogeochem_deltat 43200.0 -days_per_year 360.0 \ # ocean timestep[seconds]
        -burial_sum_steps 720 \  # sum burial over a period of 720 timesteps = 1 yr
        -pco2atm 280.0 \ # use fixed pCO2 of 280 ppm
        -use_virtual_flux \ # use the global surface mean DIC and Alk to calculate E-P
        -periodic_matrix \ # use of periodic transport matrix # the unit of time is year and
 # circulation has a periodicity of 1 year; monthly mean tranport matrix (12 TMs/year)
        -matrix_cycle_period 1.0 -matrix_num_per_period 12 \
        -periodic_biogeochem_forcing \
        -periodic_biogeochem_cycle_period 1.0 -periodic_biogeochem_num_per_period 12 \
        -num_biogeochem_steps_per_ocean_step 8 \  # the number of BGC timestep per ocean step
        -separate_biogeochem_time_stepping \ # timestep BGC model separately from ocean step
        -time_avg -avg_start_time_step 2159281 -avg_time_steps 60 \
        # initial timestep for diagnostic fluxes (for final year); avg over 60 timesteps = 1 month
        -avg_files po4avg.petsc,dopavg.petsc,oxyavg.petsc,phyavg.petsc,zooavg.petsc,detavg.petsc,no3avg.petsc,
        dicavg.petsc,alkavg.petsc \ # avg file names
        -bgc_params_file biogem_params.txt \   # parameter files to be read
        -num_bgc_params 7 \   # number of parameters read
        -calc_diagnostics -diag_start_time_step 2159281 -diag_time_steps 60 \
        # initial timestep for diagnostic fluxes (for final year); avg over 60 timesteps = 1 month
        -diag_files fbgc1.petsc,fbgc2.petsc,fbgc3.petsc,fbgc4.petsc,fbgc5.petsc,fbgc6.petsc,fbgc7.petsc,
        fbgc8.petsc \ # diagnsotic flux files names
        > log # outputting to logfile
```

Currently BGC model writes the following 8 diagnostics into the different files:

1. `fbgc1.petsc`: primary production in each box [mmolP/m^3/oceantimestep]
2. `fbgc2.petsc`: zooplankton grazing in each box [mmolP/m^3/oceantimestep]
3. `fbgc3.petsc`: detritus sedimentation through upper boundary of each box [mmolP/m^2/oceantimestep]
4. `fbgc4.petsc`: remineralization of detrius and DOP in each box [mmolP/m^3/oceantimestep]
5. `fbgc5.petsc`: river runoff [mmolP/m^3/oceantimestep]
6. `fbgc6.petsc`: nitrogen fixation [mmolN/m^3/oceantimestep]
7. `fbgc7.petsc`: denitrification [mmolN/m^3/oceantimestep]
8. `fbgc8.petsc`: Irradiance at the top of every layer in euphotic zone [W/m^2], added by T.Tanioka (2020/11)

To submit the job to the queue, type on the command line:

```
$ chmod u+x runscript_msi
$ sbatch runscript_msi
```

To check your job status at MSI, type:

```
$ squeue -u yourusername
```

1.2.6 Step 5: Processing the model outputs

When the run is completed, you should receive an email from MSI (if you set it so in the runscript). You should also check you log file and the bottom of the log file should say "Wall clock time xxx seconds". As the output files are binary and cannot be opened on its own, we have to convert from the binary format into either netcdf (.nc) or Matlab (.mat) files using Matlab scripts.

(Option 1, recommmended): To convert to **.nc** files you are going to use the following 5 scripts:

1. `n7tracers28.m`: This file converts each tracer snapshot .petsc file (e.g., po4put.petsc) into a single .nc file. The MATLAB syntax is:

   ```
   >> n7tracers28('filename1.nc')
   ```

   Make sure to set the basepath correctly and set useCarbon=1 (when carbon model is used) and useOrgCarbon=0. Also make sure that there is no .nc file with the same name already in the directory.

2. `n7tracersavg28.m`: This file converts each tracer time-averaged .petsc file (e.g., po4avg.petsc) into a single .nc file. The MATLAB syntax is:

   ```
   >> n7tracersavg28('filename2.nc')
   ```

   Make sure to set the basepath correctly and set useCarbon=1 and useOrgCarbon=0.

3. `n7fluxes28.m`: This file converts each diagnostic flux .petsc file (e.g., fbgc1.petsc) into a single .nc file. The MATLAB syntax is:

   ```
   >> n7fluxes28('filename3.nc')
   ```

   Make sure to set the basepath correctly and and set useOrgCarbon=0.

4. `n7physics.m`: This file makes single .c files with monthly mean temperature and salinity. The MATLAB syntax is:

   ```
   >> n7physics('filename4.nc')
   ```

   Make sure to set the basepath correctly and make sure that there is no .nc file with the same name already in the directory.

5. `load_pco2.m`: This file makes a global mean atmospheric pCO_2 file ("pco2.nc") and surface CO_2 air-sea flux file ("co2airseaflux.nc"). CO_2 is in ppm and CO_2 air-sea flux is in [mmolC/m^2/timestep] (positive flux means CO_2 is going into the sea from air). The MATLAB syntax is

   ```
   >> load_pco2
   ```

   Make sure to set the basepath and the CO_2 run options correctly.

6. `process_output.m` (optinal): To run all 5 scripts at once, edit and use this file. The MATLAB syntax is

   ```
   >> process_output
   ```

(Option 2): To convert to **.mat** files you are going to run 2 scripts:

1. `load_output.m`: This file converts each tracer snapshot .petsc file (e.g., po4put.petsc) to .mat file. Make sure to edit the `base_path` and add `tmm_matlab_code` to search path in lines 2 and 3.

2. `load_output_time_avg.m`: This file converts each tracer average concentration .petsc file (e.g., po4avg.petsc) to .mat file. Make sure to edit the `base_path` and search path correctly.

3. Unfortunately I don't have scripts for creating .mat diagnostic flux files, physics files, and CO_2 files. But this should not be too hard to do and all you have to do is to modify other .m files.

1.2.7 Step 6: Displaying the model outputs

(Option 1): To view .nc files, I recommend the software *Ferret* developed by NOAA. It is already installed in MSI, and to launch Ferret, type on the command line:

```
$ ls *.nc
filename1.nc filename2.nc filename3.nc filename4.nc co2airseaflux.nc pco2.nc
$ module load ferret/7.6.0
$ ferret
       NOAA/PMEL TMAP
       PyFerret v7.6 (optimized)
       Linux 4.15.0-1089-azure - 06/25/20
       11-Nov-20 16:51

yes? load filename1.nc
yes? sho d
     currently SET data sets:
    1> ./filename1.nc  (default)
 name     title                          I         J         K         L
 PO4                                   1:128     1:64      1:15      1:31
 DOP                                   1:128     1:64      1:15      1:31
 OXYGEN                                1:128     1:64      1:15      1:31
 PHYTO                                 1:128     1:64      1:15      1:31
 ZOO                                   1:128     1:64      1:15      1:31
 DET                                   1:128     1:64      1:15      1:31
 NO3                                   1:128     1:64      1:15      1:31
 DIC                                   1:128     1:64      1:15      1:31
 ALK                                   1:128     1:64      1:15      1:31
```

I won't go into too much detail here but to learn more about Ferret, visit NOAA's website and take a tutorial at https://ferret.pmel.noaa.gov/Ferret/documentation/ferret-tutorials. You can make all kinds of graphs and figures with Ferret.

(Option 2): If you want to visualize .mat files, I recommend the MATLAB package `m_map`. Again, I would not go into much detail here but if you want to learn about it, download the package and take the tutorial from the developer's website: https://www.eoas.ubc.ca/~rich/map.html. Compared to Ferret, you have more freedom for customizing graphs. The downside is that you are going to have to write much longer codes.

2 Case Studies

2.1 Study 1: Postindustrial CO_2 uptake with MOPS+MIT2.8

In the first case study, we will simulate postindustrial CO_2 uptake using MOPS and MITgcm2.8 under IPCC's RCP8.5 scenario from 1765 to 2265. We are going to do the continuous run from the spinup run (Section 1) so make sure that you have completed that run before you proceed.

1. Update source codes in your `tmm` directly by doing `git pull` from the remote repository.

```
$ cd $TMMROOT
$ git checkout TT_Release
$ git pull origin TT_Release
```

2. Make a new run directory. Let's call it `Test_co2historyRCP85` and copy all the files needed into this directory.

```
$ cd ~/TMM2
$ mkdir -p Runs/MOPS/Test_co2historyRCP85
$ cd Runs/MOPS/Test_co2historyRCP85
$ cp -p $TMMROOT/models/current/mops2.0/src/Makefile .
$ cp -p -R $TMMROOT/models/current/mops2.0/matlab/* .
$ cp -p $TMMROOT/models/current/mops2.0/runscripts/* .
 $ cp -p $TMMROOT/models/current/mops2.0/parameters/* .
```

3. Compile mops as we have done in the spinup run. Don't forget to load modules and make sure that environmental variables `$TMMROOT` and `$PETSC_DIR` are set up properly.

```
$ make cleanall
$ make mops
```

4. Copy the file `pickup.petsc` from the spinup run directory into the current working directory and give it a new name (restart.petsc). This petsc files contains the tracer concentrations from the final year of the run and is required for the continuous run.

```
$ cp ../Test_spinup/pickup.petsc restart.petsc
```

Also, copy the file `pickup_runoff.bin` from the spinup run directory. This binary file contains the global sedimentation from the final year of the previous run.

```
$ cp ../Test_spinup/pickup_runoff.bin restart_runoff.bin
```

(Optional) Instead of copying files, you can also create symbolink links so that you would know exactly where these restart files actually come when you do `$ls -l`.

```
$ ln -s ../Test_spinup/pickup.petsc restart.petsc
$ ln -s ../Test_spinup/pickup_runoff.bin restart_runoff.bin
$ ls -l restart.petsc
lrwxrwxrwx. 1 tanio003 matsumot 26 Sep 28 12:33 restart.petsc -> ../Test_spinup/pickup.petsc
```

5. Edit the input generation file (`make_input_files_for_mops_model.m`). The key here is to set `useTimeVaryingPrescribedCO2=1` and choosing the appropriate `co2Scenario`, in this case RCP85.

```
% make_input_files_for_mops_model.m

% Set toplevel path to GCMs configuration
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_2.8deg';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_ECCO';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_ECCO_v4';
base_path='~/TMM2/MITgcm_2.8deg';
%base_path='~/TMM2/MITgcm_ECCO';

addpath(genpath('~/TMM2/tmm_matlab_code'));% add tmm_matlab_code to the search path
oceanCarbonBasePath='~/TMM2/OceanCarbon';  % add OceanCarbon to the search path
atmosDataPath=fullfile(oceanCarbonBasePath,'AtmosphericCarbonData');
...

READ_SWRAD=0                    % Read short-wave radiation?
useCarbon=1                     % Use simple inorganic carbon model?
useOrgCarbon=0                  % Use organic carbon module?
useAtmModel=0                   % Use prognostic 1-box atmosphere?
pCO2atm_ini=280.0               % Initial pco2? (will be ignored as useAtmModel=0)
useTimeVaryingPrescribedCO2=1   % Use prescribed pco2 pathway?
useVirtualFlux=1                % Use DIC and Alk to calculate E-P?
empScaleFactor=1.0              % Scaling factor for E-P (default = 1)
%-----------------------------------
% Modified by Tatsuro Tanioka 200907 to allow for Atmospheric CO2 option
% For a prescribed pCO2 run, useTimeVaryingPrescribedCO2=1 and choose a scenario

% Available options: 'historical', 'RCP3PD', 'RCP45', 'RCP6' and 'RCP85'
co2Scenario='RCP85';
%-----------------------------------
```

6. Open Matlab and make input files:

```
$ matlab -nodesktop
...
>> make_input_files_for_mops_model
```

7. Edit `runscript_msi`. Important changes are commented with blue (do not actually put these comments after line 13 on the script). You can manually edit the script or copy my script from `~/../tanio003/TMM2/Runs/MOPS/Testruns/Slurm/Test_co2historyRCP85_slurm`.

```
 1 #!/bin/bash -l
 2 #SBATCH --nodes=1
 3 #SBATCH --ntasks-per-node=24
 4 #SBATCH --mem-per-cpu=2gb
 5 #SBATCH --time=00:60:00
 6 #SBATCH --mail-type=ALL
 7 #SBATCH --mail-user=tatsurobkkuk@gmail.com
 8 #SBATCH -p small
 9 #SBATCH --output=%j.out
10
11 cd $SLURM_SUBMIT_DIR
12
13 module load intel
14 module load impi/intel
15 module load cmake
16
17 # 360 days per year with a time step of 2 steps per day:
18 mpirun -np 24 ./mops \
19   -numtracers 9 \
20   -pickup restart.petsc \   # read initial values from file
21   -me Ae \
22   -mi Ai \
```

```
23    -t0 1765.0 -iter0 0 \    # start from year 1765
24    -deltat_clock 0.0013888888888889 \
25    -max_steps 360000 \  # run for 720*500 timesteps = 500 years
26    -write_time_steps 7200 \    # output every 720*10 timestep = 10 years
27    -o po4out.petsc,dopout.petsc,oxyout.petsc,phyout.petsc,zooout.petsc,
detout.petsc,no3out.petsc,dicout.petsc,alkout.petsc \
28    -external_forcing \
29    -use_profiles \
30    -nzeuph 2 \
31    -biogeochem_deltat 43200.0 -days_per_year 360.0 \
32    -burial_sum_steps 720 \
33    -runoff_ini_file restart_runoff.bin \ # read global sedimentation of previous run
34    -periodic_matrix \
35    -pco2atm_history TpCO2.bin,pCO2atm.bin \ # read prescribed pCO2 and time
36    -use_virtual_flux \
37    -matrix_cycle_period 1.0 -matrix_num_per_period 12 \
38    -periodic_biogeochem_forcing \
39    -periodic_biogeochem_cycle_period 1.0 -periodic_biogeochem_num_per_period 12 \
40    -num_biogeochem_steps_per_ocean_step 8 \
41    -separate_biogeochem_time_stepping \
42    -time_avg -avg_start_time_step 169201 -avg_time_steps 720 \
# output annual avg. from year 2000 (235 years after 1765)
43    -avg_files po4avg.petsc,dopavg.petsc,oxyavg.petsc,phyavg.petsc,zooavg.petsc,
detavg.petsc,no3avg.petsc,dicavg.petsc,alkavg.petsc \
44     -bgc_params_file biogem_params.txt \
45      -num_bgc_params 7 \
46    -calc_diagnostics -diag_start_time_step 169201 -diag_time_steps 720 \
# output annual mean fluxes from 2000 (235 years after 1765)
47    -diag_files fbgc1.petsc,fbgc2.petsc,fbgc3.petsc,fbgc4.petsc,fbgc5.petsc,
fbgc6.petsc,fbgc7.petsc,fbgc8.petsc \
48    > log
```

8. Make sure that paramter file (`biogem_params.txt`) is same as the one used in the spinup run. You can simply copy that file into the current directory.

```
$ cp ../Test_spinup/biogem_params.txt .
```

9. Submit the job to MSI queue.

```
$ sbatch runscript_msi
```

10. Processing and outputting results. Use 5 Matlab scripts (`n7tracers28.m,n7tracersavg28.m,n7fluxes28.m,n7physics.m,load_pco2.m`) to create .nc files. Here, I will show you how to plot CO_2 timeseries and global annual mean air-sea flux of CO_2 using Ferret. The script is also at `~/../tanio003/TMM2/Runs/MOPS/Testruns/Slurm/Test_co2historyRCP85_slurm/analyzeoutput.jnl`.

```
$ module load ferret/7.6.0
$ ferret
       NOAA/PMEL TMAP
       PyFerret v7.6 (optimized)
       Linux 4.15.0-1089-azure - 06/25/20
       16-Sep-20 14:49
yes? use pco2.nc
yes? use co2airseaflux.nc
yes? set w 1;plot/d=1/hlimits=1800:2120/vlimits=200:1200/title="pCO2␣RCP8.5" pco2
yes? ppl XLAB Year
yes? ppl YLAB pCO2 (ppm)
yes? ppl PLOT
yes? set w 2 ; sha/d=2/x=-180:180/lev=(-Inf)(-4,4,0.8)(Inf)/palette=cmocean_balance/
title="Annual␣CO2␣flux␣(mol␣C␣m-2␣yr-1)" co2air_to_sea_flux[l=10]*2*360/1000*(-1);go fland
yes? set w 3 ; sha/d=2/x=-180:180/lev=(-Inf)(-4,4,0.8)(Inf)/palette=cmocean_balance/
title="Annual␣CO2␣flux␣(mol␣C␣m-2␣yr-1)" co2air_to_sea_flux[l=100]*2*360/1000*(-1);go fland
yes? set w 4; plot/d=2/hlimits=2000:2100/vlimits=-7:0/
title="Ocean-atmosphere␣CO2␣flux␣(PgC␣yr-1)" co2air_to_sea_flux[x=@din,y=@din]*2*360/1000*12/1e15*(-1
```

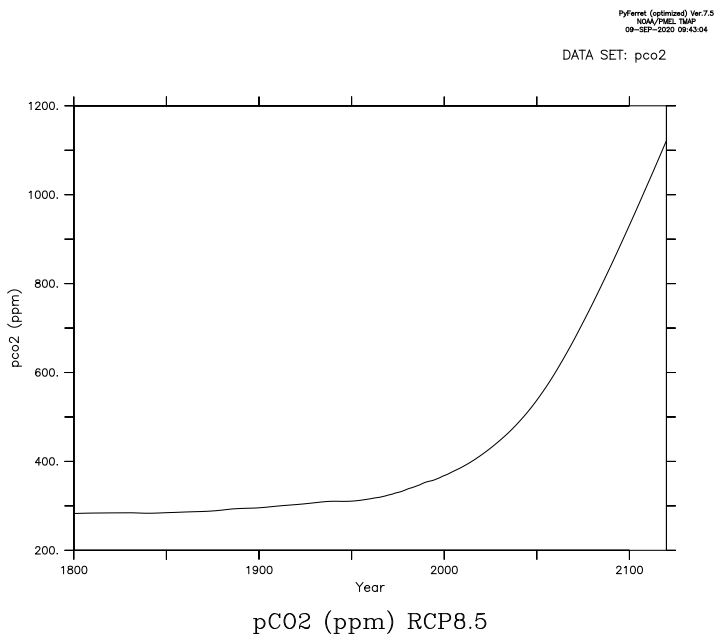

Figure 1: CO_2 concentration pathway under RCP8.5 scenario (Meinshausen et al., 2011).

Figure 1 (*Ferret:w 1*) shows the change in CO_2 under RCP8.5 scenario. There is a rapid increase in CO_2 concenctration from the late 1990s.

Figure 2 compares annual mean CO_2 flux in 2010 (*Ferret:w 2*) and 2100 (*Ferret:w 3*). Notice that in 2010, large parts of the equatorial regions are net source of CO_2 but in 2100, larger parts are becoming sink. Also, greater part of the Southern Ocean is also expected to absorb more CO_2 in 2100 compared to 2010.

Figure 3 is a CO_2 flux in 2010 from a study by Woolf et al. (2019) who used Surface CO_2 atlas and wind data. If you compare with the left panel of Figure 2, you can see that our MOPS model does a quite good job in reproducing the large scale pattern.

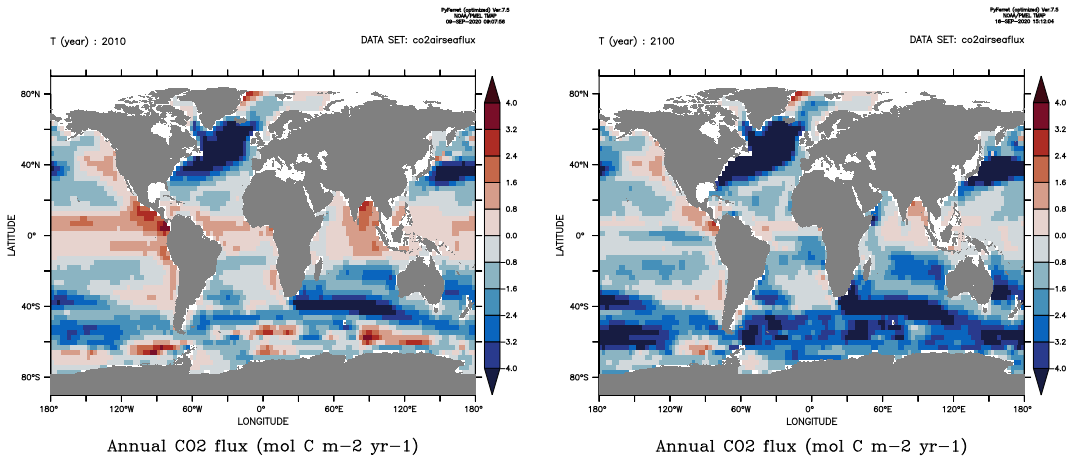

Figure 2: Modeled map of annual CO_2 flux for 2010 (Left) and for 2100 (Right). Upward fluxes are defined as positive.

Figure 4 (*Ferret:w 4*) is a globally integrated sea-air CO_2 flux projections under RCP8.5 scenario.The negative value means net uptake of CO_2 by ocean. For a comparison, we show the similar projection using the CESM and CMIP5 (Figure 5a-b.) from Lovenduski et al. (2016). Again, notice that MOPS's result is largely consistent in terms of the magnitude.

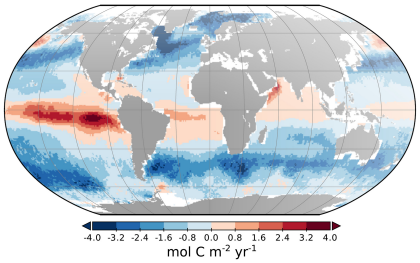

Figure 3: Map of annual CO_2 flux in 2010 from CO_2 and wind data (from Woolf et al., 2019). Compare this with Figure 2.

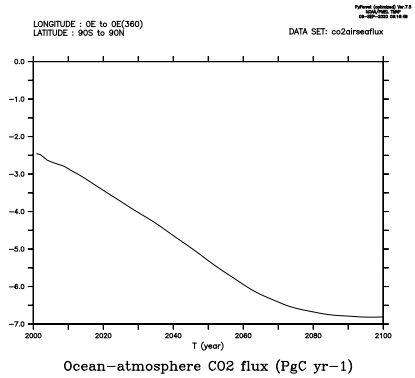

Figure 4: Modeled annual-mean globally integrated sea-air CO_2 flux projections under RCP8.5 scenario.

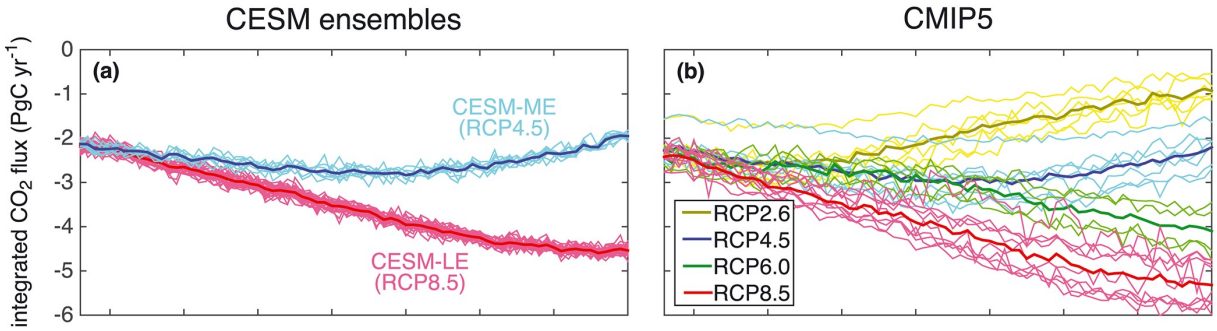

Figure 5: Modeled annual-mean globally integrated sea-air CO_2 flux projections (from Lovenduski et al., 2016) with CESM (a) and CMIP5 models (b). Compare this with Figure 4.

2.2 Study 2: Using MOPS+ECCO to simulate flexible C:P

In the second case study, we will use the modified MOPS code to study the spatiotemporal variability in organic matter C:P ratio under steady-state scenario. We are going to do a 1000-yr spinup run where C:P of phytoplankton is flexible but C:P of zooplankton is homeostatic (i.e. less flexible). We will use the TMM model 12 monthly mean TMs derived from the Estimating the Circulation and Climate of the Ocean (ECCO) project, which provides circulation fields that yield a best fit to hydrographic and remote sensing observations over a 10-year period, on a spatial grid of 1-by-1 degree horizontal resolution with 23 vertical levels

We model P:C uptake ratio of phytoplankton using the power-law as a function of nutrients, temperature, and light following Tanioka and Matsumoto (2020):

$$(\mathrm{P:C})_{\mathrm{uptake,phyto}} = (\mathrm{P:C})_{\mathrm{ref}} \left(\frac{[\mathrm{PO_4}]}{[\mathrm{PO_4}]_0}\right)^{s^{\mathrm{P:C}}_{\mathrm{PO_4}}} \left(\frac{[\mathrm{NO_3}]}{[\mathrm{NO_3}]_0}\right)^{s^{\mathrm{P:C}}_{\mathrm{NO_3}}} \left(\frac{[\mathrm{T}]}{[\mathrm{T}]_0}\right)^{s^{\mathrm{P:C}}_{\mathrm{T}}} \left(\frac{[\mathrm{I}]}{[\mathrm{I}]_0}\right)^{s^{\mathrm{P:C}}_{\mathrm{I}}} \quad (1)$$

Exponent $s^{\mathrm{P:C}}_{\mathrm{driver}}$ is constant and describe the strength of a given driver on P:C.

Zooplankton P:C uptake ratio is also computed using the power-law as a function of phytoplankton P:C of biomass:

$$(\mathrm{P:C})_{\mathrm{uptake,zoo}} = (\mathrm{P:C})^{1-\mathrm{H}}_{\mathrm{ref,zoo}} \cdot (\mathrm{P:C})^{\mathrm{H}}_{\mathrm{biomass,phyto}} \quad (2)$$

Parameter H is a zooplankton homeostatic parameter that describes how variable zooplankton P:C is affected by

phytoplankton P:C. When $H = 0$, P:C uptale equals a fixed reference P:C and is independent of phytoplankton P:C. When $H = 1$, uptake P:C equals cellular P:C of phytoplankton ("you are what you eat"). Here we will use the value of $H = 0.08$ following Persson et al. (2010), who conducted meta-analysis on zooplankton C:P.

In order to cope with the mismatch between required P:C and P:C of prey, zooplankton can excrete excess P or C as dissolved and particulate organic matter. In teh default setting, all the excess material will be routed towards particulate phase. This can be changed by adjusting the parameter `par_excrtodoc` (default value = 0.0). At steady state, uptake P:C ratio should equal the biomass P:C ratio for both phytoplankton and zooplankton.

In implementing flexible C:P, I added 4 new tracers: (1) Phytoplankton C biomass, (2) Zooplankton C biomass. (3) DOC, and (4) POC. The total number of tracers is now 13.

1. Update source codes in your `tmm` directly by doing `git pull` from the remote repository.

```
$ cd $TMMROOT
$ git checkout TT_Release
$ git pull origin TT_Release
```

2. Make a new run directory. Here, we call it `Test_ECCO_CP` and copy all the files needed into this directory. I also created a new directory called `paramters` which contains an ASCII text file (`biogem_params.txt`) with biogeochemical parameters.

```
$ cd ~/TMM2
$ mkdir -p Runs/MOPS/Test_ECCO_CP
$ cd Runs/MOPS/Test_ECCO_CP
$ cp -p $TMMROOT/models/current/mops2.0/src/Makefile .
$ cp -p -R $TMMROOT/models/current/mops2.0/matlab/* .
$ cp -p $TMMROOT/models/current/mops2.0/runscripts/* .
$ cp -p $TMMROOT/models/current/mops2.0/parameters/* .
```

3. Compile mops. The new model with flexible C:P is called `mops_cp`.

```
$ make cleanall
$ make mops_cp
```

4. Edit the file `make_input_files_for_mops_model.m` and run it on Matlab. First thing to do is to make sure that variable `base_path` point to the right directory for the TMM ECCO configuration. Also make sure that option `useORGCARBON` is set to 1. The options `useORGCARBON` creates new initial concentrations arrays for DOC, POC, ZooC, and PhytoC. This initizalization process takes much longer than that of the first case study.

   (Optional) Instead of creating input matrix files each time, **you can create symbolic links** to the TMM matrices using the script "`link_inputs.sh`" in the directory `~/TMM2/tmm/models/current/mops2.0/runscripts`. This way it saves a lot of time and disc storage.

5. Edit the parameter input file `biogem_params.txt`. In the first run, rows 1-6 should be the default values for MOPS + ECCO follwiing Kriest et al. (2020). Parameters 12-22 correspond to the P:C power-law formulations of phytoplankton and zooplankton from Equations (1)-(2). Refernce P:C for both phytoplankton and zooplankton is $1/117 = 8.547$ permil (line 12).

```
 1 1.46     #01. detmartin = Martin Curve exponent of Detritus [1.39 for MIT2.8, 1.46 for ECCO]
 2 151.1    #02. ro2ut = Refield -O2:P                         [173.7 for MIT2.8, 151.1 for ECCO]
 3 0.00229  #03. nfix = Max growth rate (1/d) of N-fixers      [0.00119 for MIT2.8, 0.00229 for ECCO]
 4 16.0     #04. subdin = No denitrification below this level of DIN (mmol/m3)          [15.8, 16.0]
 5 1.07     #05. ACkbaco2 = Half sat-constant for oxic degradation (mmol/m3)            [1.01, 1.07]
 6 23.1     #06. ACkbacdin = Half sat-constant for suboxic degradation                  [32.0, 23.1]
 7 0.01     #07. alimit = Min. value for the degradation of PHY and DOP [0.001 for mops, 0.01 for mops_cp]
 8 1.46     #08. detmartin_c = Martin Curve Exponent of POC. For pref. remin, change from #01 [1.39, 1.46]
 9 1        #09. cp_option (1 = Power-law, 2: Galbraith and Martiny, 0: Redfield)
10 250.0    #10. maxcp = upper bound C:P                                                [250.0]
11 50.0     #11. mincp = lower bound C:P                                                [50.0]
12 8.547    #12. par_bio_pc0 = Reference P:C (permil) for power-law                     [8.547]
13 0.57     #13. par_bio_po4_ref = Reference PO4 (mmolP/m3) for power-law               [0.57]
14 5.7      #14. par_bio_no3_ref = Reference NO3 (mmolN/m3) for power-law               [5.7]
15 18.0     #15. par_bio_temp_ref = Reference Temp (deg C) for power-law                [18.0]
16 70.0     #16. par_bio_light_ref = Reference Irradiance (W/m2) for power-law          [70.0]
17 0.21     #17. par_bio_spc_p = s_P:C^PO4 for power-law                                [0.21]
18 0.0      #18. par_bio_spc_n = s_P:C^NO3 for power-law                                [0.0]
19 0.0      #19. par_bio_spc_i = s_P:C^I for power-law                                  [0.0]
20 -3.6     #20. par_bio_spc_t = s_P:C^T for power-law                                  [-3.6]
21 0.08     #21. par_zoo_cp_hom = Homeostatic parameter for Zooplankton (0-1)           [0.08]
22 0.0      #22. par_excrtodoc = Fraction of zooplankton adujstment flux going to DOC   [0.0]
```

6. Edit `runscript_msi`. Here we are going to run the model for 15 hours with 10 nodes, 24 processors per core. Each processor is allocated with 2GB of memory. As we are using more than 9 nodes, we submit to the queue *large*. If you want the job to proceed quicker, you can increase the number of nodes from 10. (Note: Asking for more nodes will mean longer time in the queue. So this will require some trial and error to find the optimal node request.)

   Here is the first 12 lines of the runscript. The script is also at `~/../tanio003/TMM2/Runs/MOPS/Testruns_ECCO/201108b/runscript_msi`.

```
1 #!/bin/bash -l
2 #SBATCH --nodes=10
3 #SBATCH --ntasks-per-node=24
4 #SBATCH --mem-per-cpu=2gb
5 #SBATCH --time=15:00:00
6 #SBATCH --mail-type=ALL
7 #SBATCH --mail-user=tatsurobkkuk@gmail.com
8 #SBATCH -p large
9 #SBATCH --output=%j.out
10
11 cd $SLURM_SUBMIT_DIR
12
```

   The follwing lines (13~) contain the commands to execute and start the program `mops_cp`.

```
13 module load intel
14 module load impi/intel
15 module load cmake
16
17 # 360 days per year with a time step of 2 steps per day:
18 mpirun -np 240 ./mops_cp \  # use 240=24*10 cores, program = mops_cp
19          -numtracers 13 \   # use 13 tracers (9 tracers in mops + 4 new tracers)
20          -i po4ini.petsc,dopini.petsc,oxyini.petsc,phyini.petsc,zooini.petsc,detini.petsc,
            no3ini.petsc,dicini.petsc,alkini.petsc,docini.petsc,pocini.petsc,phycini.petsc,
            zoocini.petsc \    # 13 initial tracer concentrations
21          -me Ae \
22          -mi Ai \
23          -t0 0.0 -iter0 0 \
24          -deltat_clock 0.0013888888888888889 \
25          -max_steps 2160000 \   # Run for 720 steps/year * 3000 years
26          -write_time_steps 720000 \  # Output snapshot every 1000 years
27          -o po4out.petsc,dopout.petsc,oxyout.petsc,phyout.petsc,zooout.petsc,detout.petsc,
            no3out.petsc,dicout.petsc,alkout.petsc,docout.petsc,pocout.petsc,phycout.petsc,
            zoocout.petsc \    # 13 output tracers
28          -external_forcing \
29          -use_profiles \
30          -nzeuph 6 \        # the default number of euphotic zones = 6 layers in ECCO
31          -biogeochem_deltat 43200.0 -days_per_year 360.0 \
32          -burial_sum_steps 720 \
33          -pco2atm 280.0 \   # Constant pCO2 at 280 ppm
34          -use_virtual_flux \
35          -periodic_matrix \
36          -matrix_cycle_period 1.0 -matrix_num_per_period 12 \
37          -periodic_biogeochem_forcing \
38          -periodic_biogeochem_cycle_period 1.0 -periodic_biogeochem_num_per_period 12 \
39          -num_biogeochem_steps_per_ocean_step 8 \
40          -separate_biogeochem_time_stepping \
41          -time_avg -avg_start_time_step 2159281 -avg_time_steps 60 \ # monthly avg in final year
42          -avg_files po4avg.petsc,dopavg.petsc,oxyavg.petsc,phyavg.petsc,zooavg.petsc,detavg.petsc,
            no3avg.petsc,dicavg.petsc,alkavg.petsc,docavg.petsc,pocavg.petsc,phycavg.petsc,
            zoocavg.petsc \    # 4 new time-average tracers added
43          -bgc_params_file biogem_params.txt \   # telling to read the parameter file
44          -num_bgc_params 22 \                   # the number of parameters to be read = 22
45          -calc_diagnostics -diag_start_time_step 2159281 -diag_time_steps 60 \ # monthly diagnostics
46          -diag_files fbgc1.petsc,fbgc2.petsc,fbgc3.petsc,fbgc4.petsc,fbgc5.petsc,fbgc6.petsc,
            fbgc7.petsc,fbgc8.petsc,fbgc9.petsc,fbgc10.petsc,fbgc11.petsc \  # 11 diagnostics
47          > log
```

   The important notes are commented with blue. The three new diagnostics related to organic carbon and flexible C:P module I added are:

   (a) `fbgc9.petsc`: POC sedimentation through upper boundary of each box [mmolC/m^2/oceantimestep]

   (b) `fbgc10.petsc`: Phytoplankton C:P uptake ratio in molar units.

   (c) `fbgc11.petsc`: Zooplankton C:P uptake ratio in molar units.

   Monthly-averaged disgnostics as well as motnhly-averaged tracers will be computed in the final year (Year 2999) of the run.

7. To submit the job to the queue, type on the command line:

```
$ chmod u+x runscript_msi
$ sbatch runscript_msi
```

To check your job status at MSI, type:

```
$ squeue -u yourusername
```

8. Process the outputs using the Matlab Scripts as outlined previously. For each .m files (`n7tracers28.m,n7tracersavg28.m,n7fluxes28.m,n7physics.m,load_pco2.m`), make sure that variable `base_path` point to the right directory for the TMM ECCO configuration. Also make sure that option `useORGCARBON` is set to 1.

9. Analyze outputs. Here we will look at seasonal cycles of productivity at 4 different locations. These 4 locations are: (1) the high latitude bloomforming North Atlantic Ocean (NAT: 25W-35W, 45N-50N, black line), (2) the North Atlantic subtropical gyre (NASG: 25W-70W, 25N-35N, red line), (3) the South Pacific subtropical gyre (SPSG:90W-150W, 15S-40S, green line), and (4) the Equatorial upwelling region (EQU: 5S-5N, blue line). The code for making figrues is in my directory `~/../tanio003/TMM2/Runs/MOPS/Testruns_ECCO/201108b/analyze_201108b.jnl`.

   Figure 6 shows the seasonal cycle of productivity (left) and the tracers/forcings (right). Notice that phytoplankton biomass exceeds zooplankton biomass throughout the year in all regions. The only exception is NAT where zooplankton biomass catches up with the phytoplankton biomass following the spring bloom. NPP peaks in early spring and export production lags NPP about a month or two.

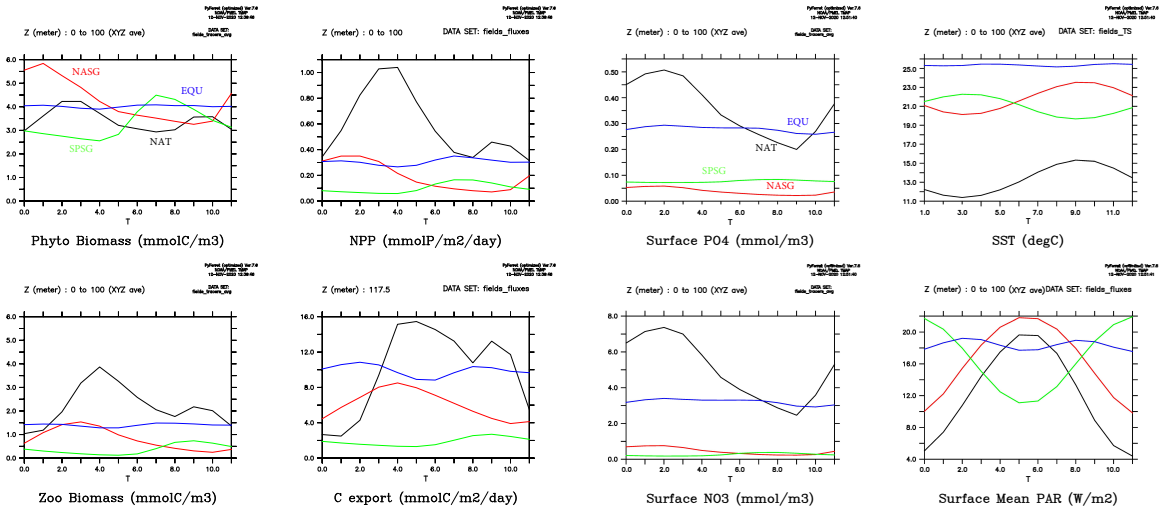

Figure 6: Modeled seasonal cycle of productivity (Left) and tracers (Right) at four geographical regions. North Atlantic Ocean (NAT: 25W-35W, 45N-50N, black line), (2) the North Atlantic subtropical gyre (NASG: 25W-70W, 25N-35N, red line), (3) the South Pacific subtropical gyre (SPSG:90W-150W, 15S-40S, green line)

   Figure 7 shows the seasonal cycle of C:P. Phytoplankton C:P at NAT and NASG peaks in fall (~September) and is lowest during spring. This pattern generally follows that of SST and PO4, where low PO4/high SST leads to higher C:P. Seasonal variability of phytoplankton C:P at SPSG and EQU are smaller as the change in SST and nutrients are small. C:P of zooplankton remains around 117 even in gyres. As phytoplankton biomass is greater than that of zooplankton, C:P of bulk POM is similar in magnitude to that of phytoplankton. What's interesting is that C:P export is greater than C:P of phytoplankton and bulk POM at NASG, SPSG, and EQU. This is because excess carbon that is excreted by zooplankton is added to detrital POC pool leading to higher C:P of exported organic mater. At NAT, the different is relatively smaller because the stoichioemtric mismatch between phytoplankton C:P and zooplankton C:P is small.

10. (Optional) We can also calculate export of DOC and DOP. The code for making figrues is in my directory `~/../tanio003/TMM2/Runs/MOPS/Testruns_ECCO/201108b/analyze_dom_export.jnl`.

    Figure 8 shows that the total DOC export is around 3.0 PgC/yr which is factor of 2 greater than the previous modeling studies. This is mainly due to the fact that the DOC surface concentration at the moment is too high. This is something that need to be worked on. Majority of export occurs in the subtropical gyres by advection. Export due to diapycnal diffusion occurs in equatorial upwelling regions but the maginude is utmost 10%

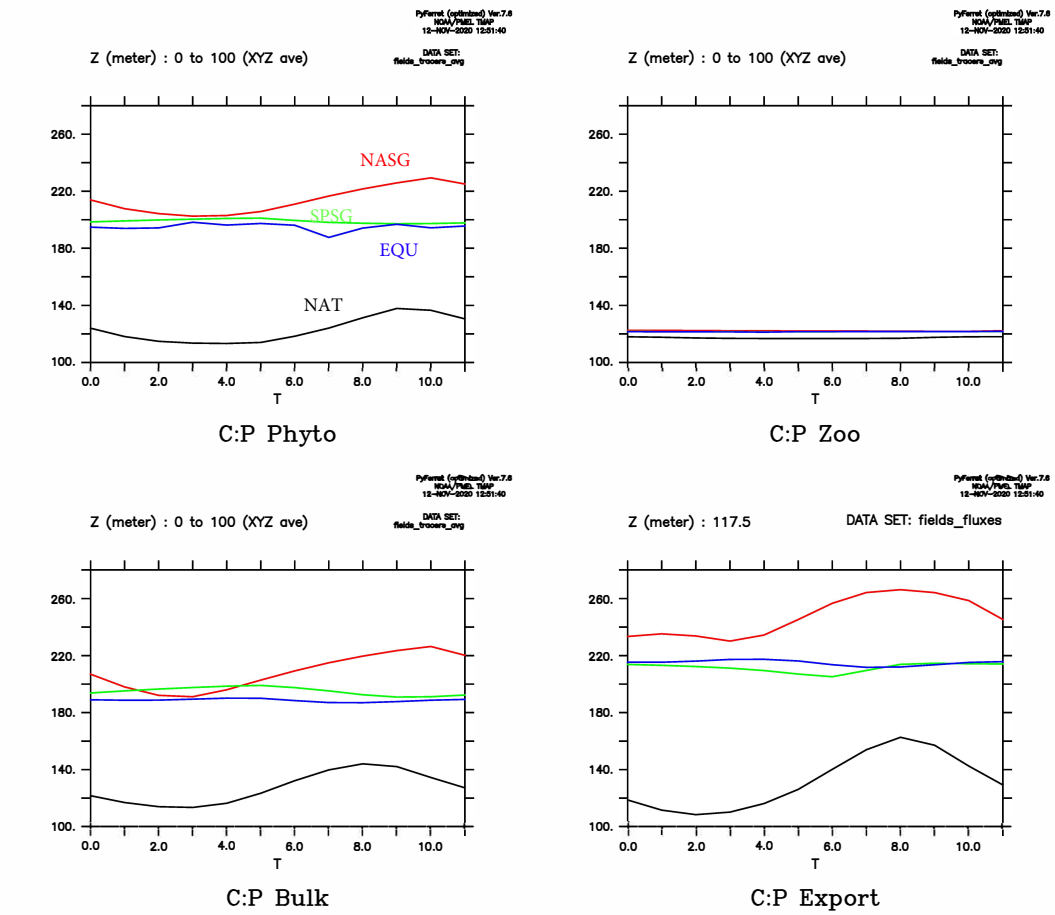

Figure 7: Modeled seasonal C:P of (a) Phytoplankton, (b) Zooplankton, (c) Bulk POM (= Detritus + Phytoplankton + Zooplankton), and (d) Export ratio at 100 m

The seasonal pattern of DOC:DOP export in Figure 9 is similar to that of POC:POP export. Compared to POC:POP export (Fig. 7(d)), temporal shift in DOC:DOP export is smaller. Another interesting thing is that DOC concentration (Fig. 9(d)) and DOC export (Fig. 9(c)) are not correlated in a predictable way. This shows the DOC export is much more nuanced as a result of interaction between physics (flow field) and biogeochemistry. DOC dynamics is something that has to be explored in more detail.

2.3 Study 3: Using MOBI+UVic to simulate isotope dynamics

For this study, we will explore ocean biogeochemical dynamics using the second version of MOBI (Model of Ocean Biogeochemistry and Isotopes). MOBI includes:

- NO_3, PO_4, and DFe as interactive nutrients.
- 3 Phytoplankton classes (diazotrophs, coccolithophores, and other phytoplankton).
- Zooplankton.
- Detritus (dead POM).
- $CaCO_3$ is a prognostic variable and its production is calculated as a fraction of the detrirus produced by coccolithophores and zooplankton and it is sinking into the deep ocean where it dissolves.
- Nitrogen (^{15}N) and carbon (^{13}C) isotopes are included in all compartments.
- Radiocarbon (^{14}C) is included only in the DIC compartment.
- Dissolved O_2 controls the water column denitrification. Benthic denitrification is calculated considerting subgridscale bathymetry.

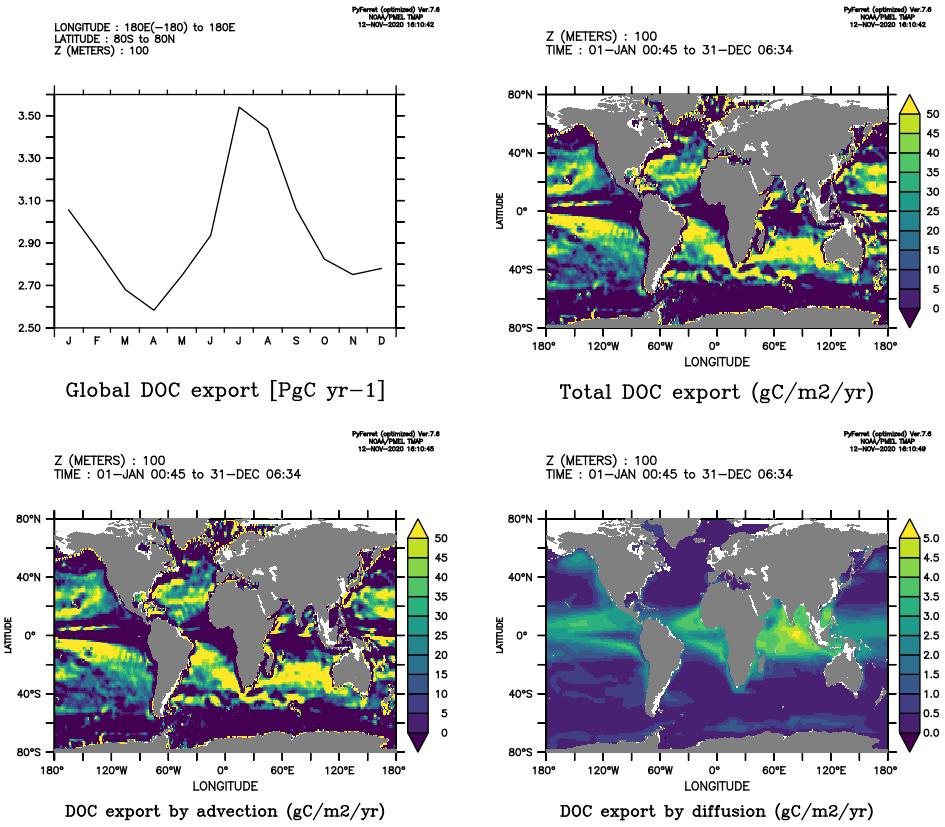

Figure 8: Modeled DOC export. (a) Monthly global DOC export at 100 m (b) Annual mean global DOC export (c) Annual mean global DOC export by advection, and (d) Annual mean global DOC export by diffusion.

- DOM is calculated separately for C, N, and P (DOC, DON, DOP).
- The most complex configuration has a total of 30 prognostic tracers.

To learn different configurations of MOBI, refer to the primer by A. Schmittner (https://github.com/tanio003/tmm/blob/TT_Release/Tutorial_MSI/MOBI2.0_Schmittner.pdf).

In this exercise, we will try conducting a spin-up run for 1000 yrs using 28 prognostic tracers. Instructions will mostly follow that of S. Khatiwala. MOBI is coupled to the UVicgit p model which consists of ocean general circulation at coarse resolution (1.8×3.6 degrees, 19 vertical levels), a one-layer energy moisture balance atmaospheric component, a dynamic/thermodynamic sea ice model, and a land surface/dynamic vegetation module.

1. Download the compatible versions of UVic ESCM 2.9 in the first level of your TMM directory

```
$ cd ~/TMM2
$ git clone https://github.com/tanio003/UVic2.9
```

2. Download the TMM called `UVicOSUpicdefault`. To copy TMM in the shared directory,

```
$ cp -r ~/../shared/tmm/UVicOSUpicdefault $HOME/TMM2/
```

   Currently, MOBI can only be coupled to this TMM configuration (`UVicOSUpicdefault`) because other TMMs like MITgcm models do not have bathymetry profile and hydrothermal Fe input masks.

3. Make a Run directory for TMM-MOBI. For this run, I call it `Runs/MOBI/Test_spinup`. Copy `mobi2.0/src/Makefile` and `mobi2.0/src/MOBI_TMM_OPTIONS.h` to this directory.

```
$ cd ~/TMM2
$ mkdir -p Runs/MOBI/Test_spinup
$ cd Runs/MOBI/Test_spinup
$ cp -p $TMMROOT/models/current/mobi2.0/src/Makefile .
$ cp -p $TMMROOT/models/current/mobi2.0/src/MOBI_TMM_OPTIONS.h .
```

   Also, following the instructions at the top of Makefile, set the environmental variable `$UVICESCMROOT` to point to the top level of the UVic ESCM 2.9. Add this to your `.bashrc` for your future convenience.

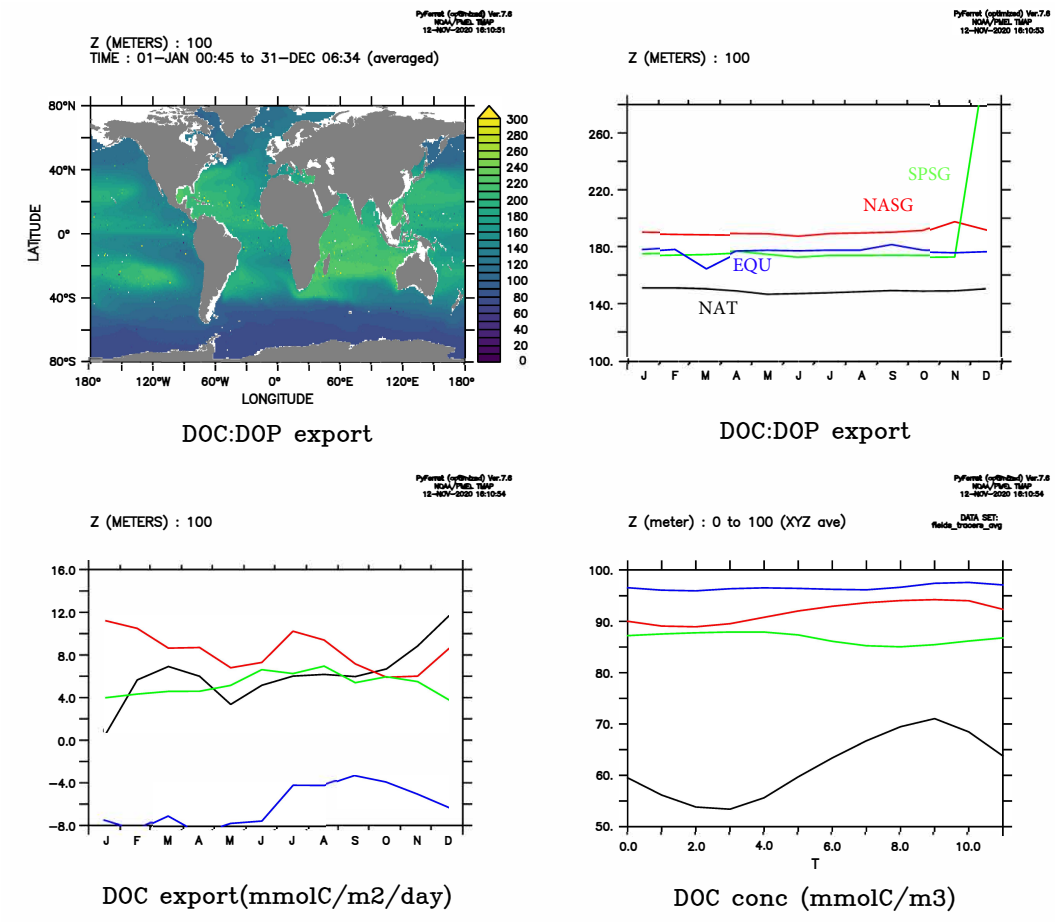

Figure 9: Modeled DOC:DOP and regional DOC export. (a) Annual mean DOC:DOP export ratio at 100 m, (b) Annual mean DOC:DOP at 4 different regions, (c) Annual mean DOC export, and (d) Annual mean surface DOC concentration.

```
$ export UVICESCMROOT=$HOME/TMM2/UVic2.9
echo $UVICESCMROOT
/home/.../TMM2/UVic2.9
```

4. Copy the contents (including the subdirectories) of `mobi2.0/matlab` and `mobi2.0/runscripts` to the test directory.

```
$ cp -p -R $TMMROOT/models/current/mobi2.0/matlab/* .
$ cp -p $TMMROOT/models/current/mobi2.0/runscripts/* .
$ ls
control.in                Makefile
get_mobi_diagnostic.m     make_input_files_for_mobi_model.m
InitialConditionProfiles  make_input_files_for_mobi_model_write_ic.m
load_diagnostics.m        MOBI_TMM_OPTIONS.h
load_output.m             MOBI_tracer_names.txt
load_output_sed.m         runscript
load_output_time_avg.m    runscript_msi
```

5. Edit the C preprocessor options in `MOBI_TMM_OPTION.h` as necessary. In particular, you have to define the directive `O_co2_user` (i.e., by `# define O_co2_ccn_user`). See Appendix 3.3 for what each of these options mean.

6. Build the model with:

```
$ make cleanall
$ make tmmmobi
```

   If compiled properly, you would see an executable file `tmmmobi` created along with a bunch of objective .o files. Don't worry about various warnings as long as they are not fatal.

7. Generate the names tracers being simulated (as per the options set in `MOBI_TMM_OPTIONS.h`) and their initial consitions. To do this, first set at the top of `make_input_files_for_mobi_model_write_ic.m`.

```
% make_input_files_for_mobi_model_write_ic.m
% Set toplevel path to GCMs configuration
% base_path='/data2/spk/UVic_OSU_Matrix/LGM_WindPerturbation_Experiments/no_embm_awind2/picdefault';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_2.8deg';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_ECCO';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_ECCO_v4';
base_path='~/TMM2/UVicOSUpicdefault';
addpath(genpath('~/TMM2/tmm_matlab_code'));
```

Then execute in MATLAB.

```
>> make_input_files_for_mobi_model_write_ic
```

8. Compile and execute `tmmmobiwrite`.

```
$ make tmmmobiwrite
$ ./tmmmobiwrite
```

This will write out `MOBI_tracer_names.txt` containing the tracer names, and a corresponding set of *.dat text files (e.g., dic.dat) containing the vertical profile of tracer concentration. These names and initial profiles are used by `make_input_files_for_mobi_model.m` to generate initial condition files (e.g., dicini.petsc).

28 tracers created are:

(1) `imobipo4` = PO_4 [mmolP m^{-3}]
(2) `imobiphyt` = Phytoplankton biomass [mmolN m^{-3}]
(3) `imobizoop` = Zooplankton biomass [mmolN m^{-3}]
(4) `imobidetr` = Detritus [mmolN m^{-3}]
(5) `imobidic` = DIC [mmol m^{-3}]
(6) `imobidic13` = DI^{13}C [mmol m^{-3}]
(7) `imobiphytc13` = ^{13}C of Phytoplankton [mmol m^{-3}]
(8) `imobizoopc13` = ^{13}C of Zooplankton [mmol m^{-3}]
(9) `imobidetrc13` = ^{13}C of Detritus [mmol m^{-3}]
(10) `imobidoc13` = DO^{13}C [mmol m^{-3}]
(11) `imobidiazc13` = ^{13}C of Diazotrophs [mmol m^{-3}]
(12) `imobicoccc13` = ^{13}C of Coccolithophores [mmol m^{-3}]
(13) `imobicaco3c13` = Ca$^{13}CO_3$ [mmol m^{-3}]
(14) `imobidop` = DOP [mmolP m^{-3}]
(15) `imobino3` = NO_3 [mmolN m^{-3}]
(16) `imobidon` = DON [mmolN m^{-3}]
(17) `imobidiaz` = Diazotroph biomass [mmolN m^{-3}]
(18) `imobidin15` = DI^{15}N [mmol m^{-3}]
(19) `imobidon15` = DO^{15}N [mmol m^{-3}]
(20) `imobiphytn15` = ^{15}N of Phytoplankton [mmol m^{-3}]
(21) `imobizoopn15` = ^{15}N of Zooplankton [mmol m^{-3}]
(22) `imobidetrn15` = ^{15}N of Detritus [mmol m^{-3}]
(23) `imobidiazn15` = ^{15}N of Diazotrophs [mmol m^{-3}]
(24) `imobicoccn15` = ^{15}N of Coccolithophores [mmol m^{-3}]
(25) `imobicocc` = Coccolithophores biomass [mmolN m^{-3}]
(26) `imobicaco3` = Calcium carbonate [mmol m^{-3}]
(27) `imobidfe` = Dissolved iron [nmol m^{-3}]
(28) `imobidetrfe` = Detrital iron [nmol m^{-3}]

9. Make a directory `InitialConditionProfiles` and copy the *.dat files into it. By default `make_input_files_for_mobi_model.m` (mentioned later) will look in this directory for the *.dat files.

```
$ cp -p *.dat InitialConditionProfiles/
```

(Note-1): There are already copies of `MOBI_tracer_names.txt` and `InitialConditionProfiles/` in matlab/ but S.Khawtiwala recommends you still go through this step to generate tracer names consistent with your version of `UVic/MOBI` and `MOBI_TMM_OPTIONS.h`.

(Note-2): The order of tracers in `MOBI_tracer_names.txt` is the same as that in which you specify initial condition, output and time average files in the TMM-MOBI run script. You can reduce the tedium (and chances of making a mistake) of generating the correct sequence to specify in the run script by executing this on the command line:

```
$ printf "%sini.petsc," `tr '\n' ' ' < MOBI_tracer_names.txt` | sed 's/,*$//g'
```

(The tr replaces newlines with spaces to put all tracer names on a single line, the printf appends "ini.petsc," to each tracer name, and the sed strips out the final comma.)

(Note-3): `tmmmobiwrite` uses UVic routines where initial condition profiles are set. The code for this is hardwired for the standard UVic grid with 19 levels. If you're using TMs from a different ocean model you will need to modify `make_input_files_for_mobi_model.m` accordingly to specify initial conditions corresponding to your grid. You will also need to change the number of levels (in kilometers) in `size.h`.

10. Set the path at the top of `make_input_files_for_mobi_model.m` and execute in MATLAB. Make sure to set the correct timestep of 28800 seconds.

```
% make_input_files_for_mobi_model.m
% Set toplevel path to GCMs configuration
% base_path='/data2/spk/UVic_OSU_Matrix/LGM_WindPerturbation_Experiments/no_embm_awind2/picdefault';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_2.8deg';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_ECCO';
% base_path='/data2/spk/TransportMatrixConfigs/MITgcm_ECCO_v4';
base_path='~/TMM2/UVicOSUpicdefault';
addpath(genpath('~/TMM2/tmm_matlab_code'));

periodicForcing=1
periodicMatrix=1

dt=28800; % time step to use
....
```

```
>> make_input_files_for_mobi_model
```

11. Edit the namelists in `control.in` as necessary. The important variables you must set are:

    - `dtts` in the `&tsteps` namelist: this MUST match the time step argument to the run time option called `-biogeochem_deltat` in the runscript. This sets the tracer advection-diffusion timestep (28800 [seconds] for UVIc model).
    - `co2ccn` and `dc14ccn` in the `&carbon` namelist.
    - `dtnpzd` in the `&npzd` namelist. Set this so that `dtts` is a multiple of `dtnpzd`. For stability, `dtnpzd` (in seconds) should not exceed 1/3rd of a day (in seconds). The default is `dtpnzd`=14400 = 1/2 of ocean timestep.

    See Appendix 3.4 for what each of these other options mean in `control.in`.

12. Edit runscript as necessary. Also see (NOTE-2 in #9). It takes about 8 hours to complete 1000 yr run with 2 nodes (48 cores). In this run, we will output monthly tracer averages as well as diagnostic fluxes in the final year of the run. Tracer snapshot concentrations will be outputted every 100 years.

```
# runscript_msi
```

    (NOTE-1): If turning on sediments and running on a smaller number of CPUs you should unlimit stacksize. Otherwise you may get a segmentation fault. Don't worry about this too much for MSI users but you can check just in case by:

```
$ ulimit -H -a
...
stack size              (kbytes, -s) unlimited
```

13. Submit your job to MSI queue.

```
$ sbatch runscript_msi
```

14. Whilte you wait for the run to be completed, set the path at the top of the load*.m scripts to read in the output in Matlab. To make all the output netcdf files in matlab at once, run `process_output.m`. For this exercise, the sedimentary module, hence the sedimentary output is suppressed.

    Each .nc file is spewed out one by one. There are different kinds of output files that are being produced:

    - F_**.nc (e.g., F_dic.nc): Surface Downward Fluxes [mol $m^{-2}s^{-1}$]. Edit "available_diagnostics.txt" to select which fluxes to create.
    - O_**.nc (e.g., O_phytnpp.nc): Various biogeochemical fluxes and concentrations that are calculated in the model. Units are in [mol $m^{-2}s^{-1}$] or [mol m^{-2}]. Edit "available_diagnostics.txt" to select which fluxes to create.
    - **.nc (e.g., PO4.nc): Snapshot concentrations of tracers. Edit "MOBI_tracer_names.txt" to select which tracers you want to convert to .nc files.
    - **mmavg.nc (e.g., PO4mmavg.nc): Time-averaged concetrations of tracers. Edit "MOBI_tracer_names.txt" to select which tracers you want to convert to .nc files.

15. Display outputs. Here, we will make figures for $\Delta^{14}C$ [permil] and $\delta^{13}C$ [permil] at the final year of the run and compare with observations. The code for making the following two figures is my directory `~/../tanio003/TMM2/Runs/MOBI/Test_spinup/analyze_mobi_spinup.jnl`.

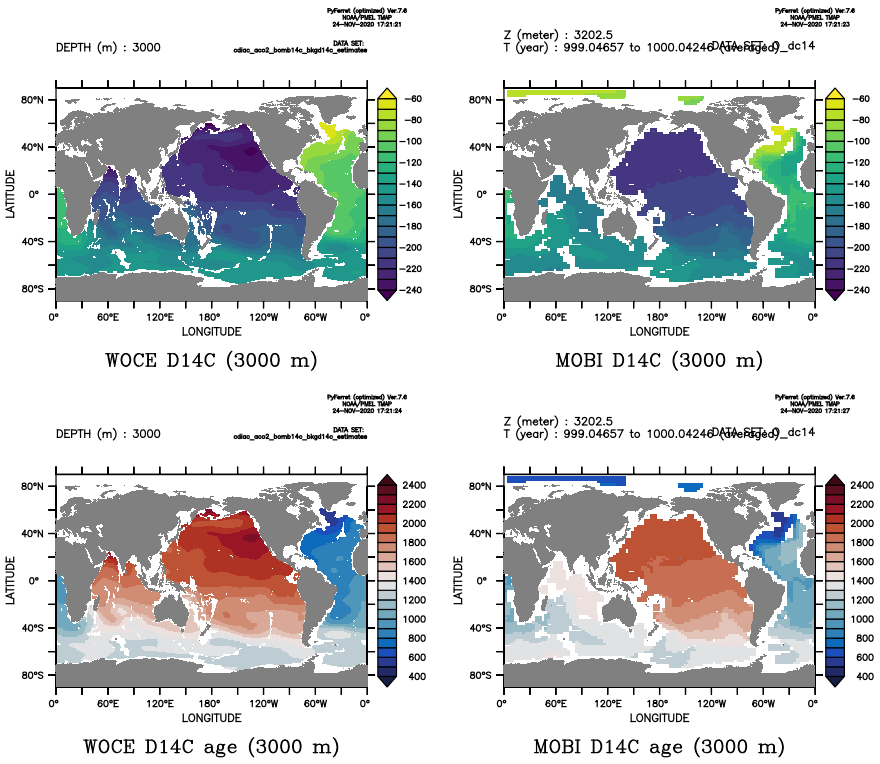

Figure 10: Observed and modeled natural $\Delta^{14}C$ and ^{14}C age at 3000 m. (a) Data-derived estimtate of $\Delta^{14}C$ from WOCE (Key et al., 2004) at 3000 m (b) Modeled $\Delta^{14}C$ by MOBI at Year 1000 (c) Observed ^{14}C age at 3000 m, and (d) Modeled ^{14}C age at 3000 m.

Figure 10 compares observed and modeled radiocarbon ages. Although ^{14}C has not fully reached steady-state after 1000 yrs (figure not shown), the MOBI seems to reproduce the observed pattern of deep-sea natural radiocarbon age quite well.

Figure 11 compares observed and modeled $\delta^{13}C$. In general, modeleded $\delta^{13}C$ overestimates the observed value in the surface ocean, but it captures the large-scale pattern. Biological fractionation and the sinking of isotopically light $\delta^{13}C$ organic matter from the surface into the interior ocean leads to low $\delta^{13}C$ DIC values at depths and in high latitude surface waters and high values in the upper ocean at low latitudes with maxima in the subtropics. Below 1000 m, model reproduces the observed pattern fairly well.

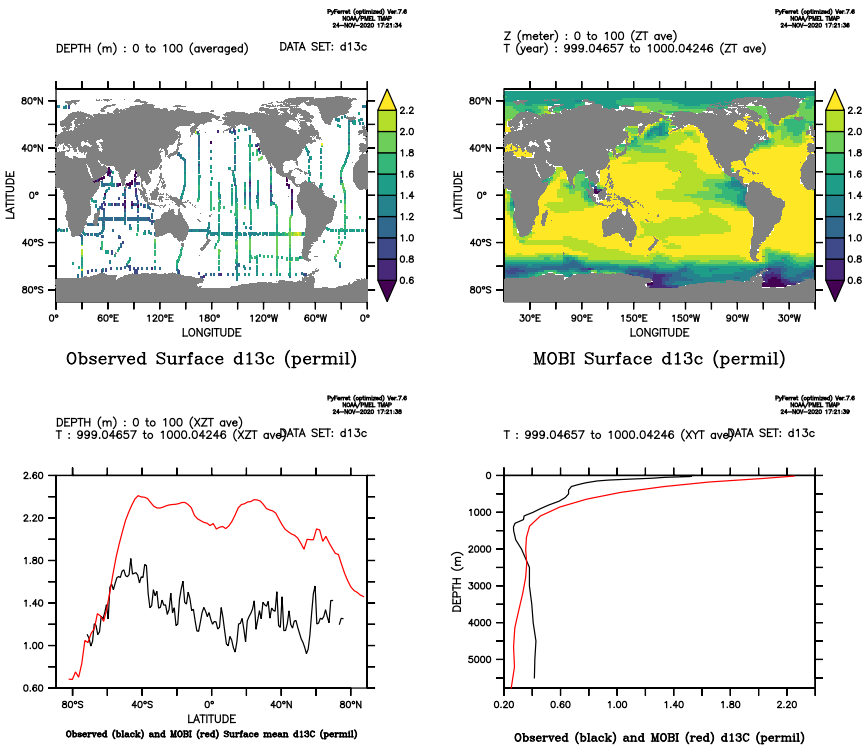

Figure 11: Observed and modeled δ^{13}C at the surface and the depth profile. (a) Observed δ^{13}C of DIC at 0-100 m from years 1990 to 2005 (Schmittner et al., 2013) (b) Modeled surface δ^{13}C by MOBI at Year 1000 (c) Observed (black) and Modeled (red) zonal average surface δ^{13}C (d) Global mean depth-profile of observed (black) and modeled (red) δ^{13}C.

2.4 Study 4: Using Transport Matrix to Caluclate Steady State Model Circulation Age

Here, we will breifly go over how to calculate steady state model circulation age and radiocarbon distributions. For full mathematical details, refer to Khatiwala et al. (2005) and Khatiwala (2007).

In TMM, tracer vector $\mathbf{c}$ at timestep $n+1$ can be calculated from the concentration and the transport matrix $\mathbf{A}$ at time step n:

$$\mathbf{c}^{n+1} = \mathbf{A}_i^n(\mathbf{A}_e^n\mathbf{c}^n + \mathbf{q}^n) \tag{3}$$

where $\mathbf{A}_e$ is the TM for the explicit-in-time component of advection-diffusion, and $\mathbf{A}_i$ the matrix for implicit transport. $\mathbf{q}$ represents the source and sinks. The TMs $\mathbf{A}_e$ and $\mathbf{A}_i$ are both extremely sparse. Physically, this is because of the finite speed of advection and diffusion (i.e., in a single timestep tracer can only communicate with their nearest neighbors). In the vertical, convection can spread the tracer further but this is restricted to small regions in high latitudes and is captured entirely by $\mathbf{A}_i$.

In some cases, tracers are subject to a prescribed concentration (BC) at the surface. To solve such equations, we convert Equation (3) into an equation for the time evolution of the interior tracer field as follows:

$$\mathbf{c}_I^{n+1} = \mathbf{A}_i^I(\mathbf{A}_e^I\mathbf{c}_I^n + \mathbf{B}_e\mathbf{c}_B^n + \mathbf{q}_I^n) + \mathbf{B}_i\mathbf{c}_B^{n+1} \tag{4}$$

where $\mathbf{c}_I$ is the vector of interior tracer concentrations, $\mathbf{c}_B$ the vector of prescribed, possibly time-dependent surface boundary values, and $\mathbf{q}_I$ the interior source/sink term. Notice that we split $\mathbf{A}_e$ into an "interior" matrix $\mathbf{A}_e^I$ and a "boundary" matrix $\mathbf{B}_e$. Implicit matrix $\mathbf{A}_i$ is split into an interior and a boundary component in the same manner. The square interor matrix describes the transport of tracer between interior grid points. The rectangular boundary matrix describes the exchange of tracer between the surface boundary points and the inteior.

In many cases, we are interested in the equilibrium solutions. Using Equations (3) and (4), we can directly compute stead-state distributions of tracers without the need for long transient integrations, for example if the source/sink and boundary terms are simple linear terms. Steady-state solutions can be analytically obtained by setting $\mathbf{c} \equiv \mathbf{c}^{n+1} = \mathbf{c}^n$ in Equation (3) and solving for $\mathbf{c}$. For an illustration, we will compute using this method: (1) the steady state ideal age (mean age), and (2) radiocarbon Δ^{14}C.

(1) Ideal age $\boldsymbol{\tau}_m$: In a conventional GCM, calculation of mean age requires integrating a tracer with a unit source in the interior and a surface BC of zero concentration to steady state, an expensive computation even at coarse

resolution. In our notation of Equation (4), $\mathbf{c}_B = \mathbf{0}$ and $\mathbf{q}_I = dt\mathbf{1}$, where $\mathbf{1}$ is a vector of 1's and dt is the time step. In ECCO, dt is 1/16 d = 5400 seconds, which is same as that of MOPS-ECCO. (In MIT2.8, I recommend using dt of 1/2 d.). Substituting these in Equation (4), we get

$$(\mathbf{A}_i^I \mathbf{A}_e^I - \mathbf{I})\boldsymbol{\tau}_m = -dt\mathbf{A}_i^I \mathbf{1} \tag{5}$$

which can be readily solved for $\boldsymbol{\tau}_m$ by,

$$\boldsymbol{\tau}_m = (\mathbf{A}_i^I \mathbf{A}_e^I - \mathbf{I})^{-1}(-dt\mathbf{A}_i^I \mathbf{1}) \tag{6}$$

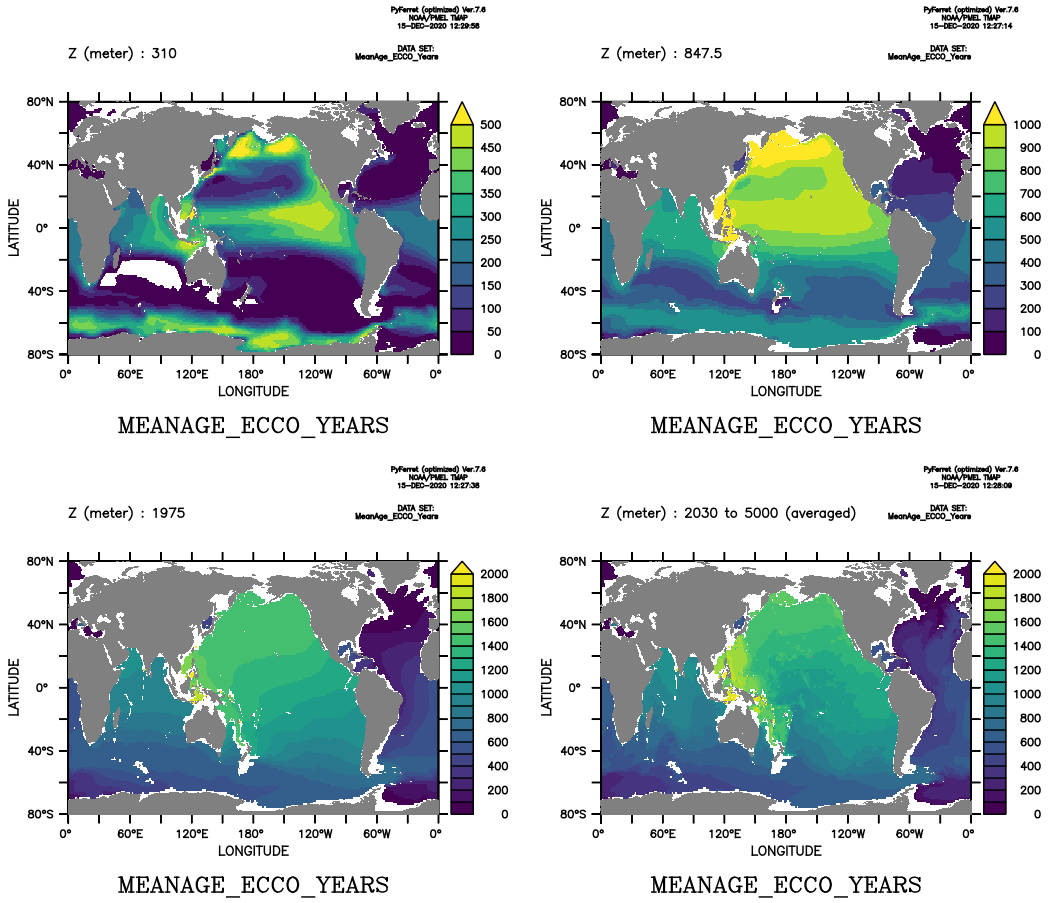

Figure 12: Modeled annual mean circulation age (in years) for the ECCO model at 4 different depths.

Figure 12 shows the mean age at four different depths obtained with the annual mean TM for ECCO model. MATLAB codes used in calculating mean age are available in the directory "`/home/matsumot/tanio003/TMM2/steady_state/ecco_meanage.m`". To run this file, you need to use the shell script `ecco_meanage.sh` in the same directory and takes 10 minutes or so on MSI. Note that I used the function `calc_steadystate_tracer` in the directory "`~/TMM2/tmm_matlab_code/TMM`" created by S. Khatiwala. See that file for more details on the syntax.

(2) Radiocarbon Δ^{14}C [Method 1]: In the first method of calculating Δ^{14}C, we prescribe the surface value using the gridded GLODAP data by Key et al. (2004) and compute steady state interior concentration. In our notation of Equation (4), $\mathbf{q}_I = -\lambda\mathbf{c}$, where λ is the decay constant for ^{14}C (half-lie of 5370 years) and $\mathbf{c}_B$ is the surface boundary concentration from GLODAP. Following Toggweiler et al. (1989), $\mathbf{c}$ is related to conventional Δ^{14}C units by,

$$\Delta^{14}\text{C (permil)} = (c - 100) \times 10 \tag{7}$$

In other words, we compute $\mathbf{c}$ first then convert to Δ^{14}C via Equation (7)).

At steady state, we can write Equation (4) as,

$$(\mathbf{A}_i^I \mathbf{A}_e^I - \lambda\mathbf{A}_i^I - \mathbf{I})\mathbf{c} = -\mathbf{A}_i^I \mathbf{B}_e \mathbf{c}_B - \mathbf{B}_i \mathbf{c}_B = -\mathbf{A}_i^I (\mathbf{B}_e \mathbf{c}_B + (\mathbf{A}_i^I)^{-1} \mathbf{B}_i \mathbf{c}_B) \tag{8}$$

The second term on the right hand side of the equation is collected by $\mathbf{A}_i^I$ for the later convenience. In the MATLAB code, the second terms needs to be divided by dt because $\mathbf{B}_i\mathbf{c}_B$ is measured on the discrete time, while $\mathbf{B}_e\mathbf{c}_B$ is on the continuous time. We solve for $\mathbf{c}$ in Equation (8) by,

$$\mathbf{c} = -(\mathbf{A}_i^I\mathbf{A}_e^I - \lambda\mathbf{A}_i^I - \mathbf{I})^{-1}\mathbf{A}_i^I(\mathbf{B}_e\mathbf{c}_B + (\mathbf{A}_i^I)^{-1}\mathbf{B}_i\mathbf{c}_B) \tag{9}$$

then convert to Δ^{14}C by Equation (7).

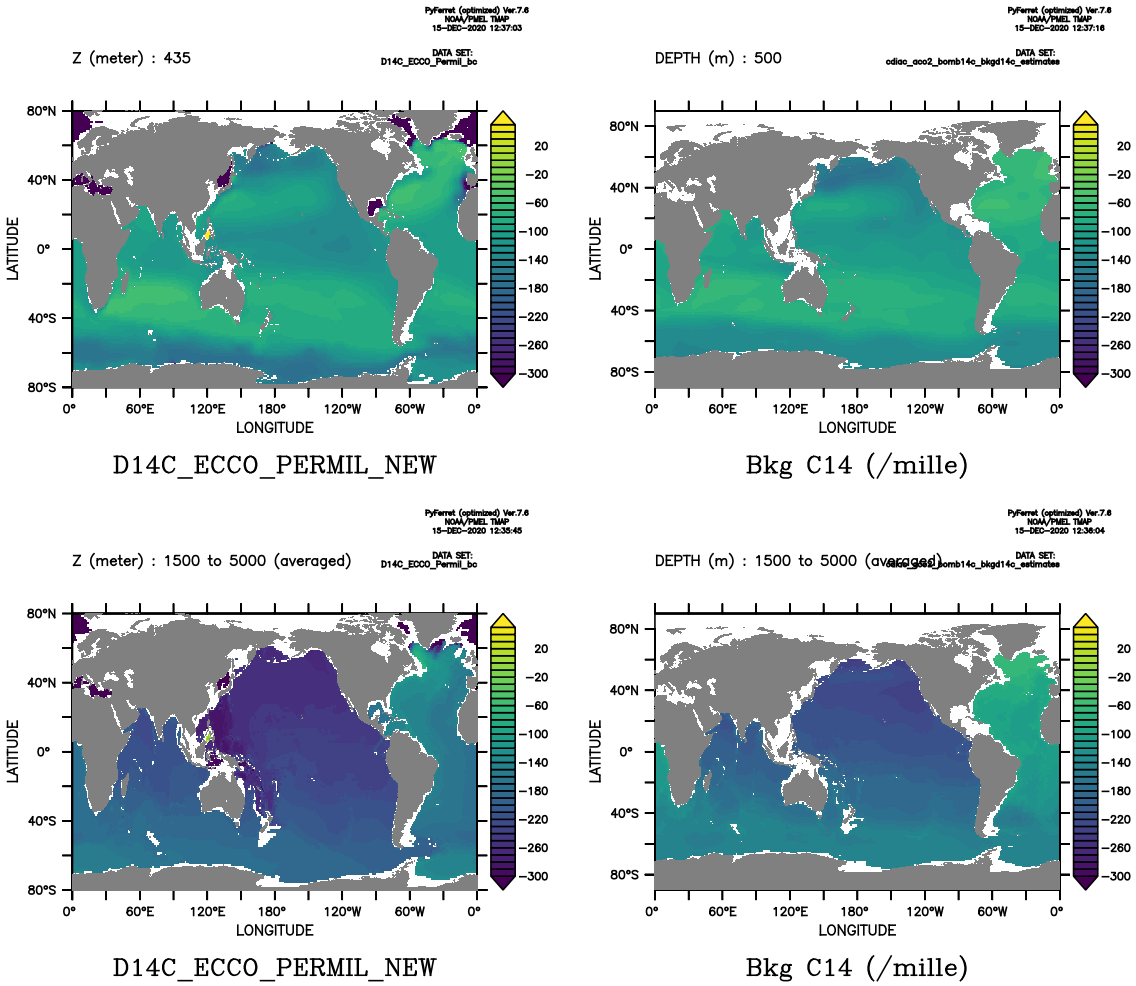

Figure 13: Modeled (Left) and GLODAP-derived (right) natural annual mean Δ^{14}C (in permil) at 2 different depths. Model used is ECCO and GLODAP map is from Key et al. (2004).

Figure 13 shows the modeled (left panel) and GLODAP-based Δ^{14}C (right panel) at 500 m and at 1500-5000 m. We can see that modeled Δ^{14}C does a good job in reproducing the observed radiocarbon pattern. In the deep ocean, modeled radiocarbon is slightly older than that of observation. MATLAB codes used in calculating radiocarbon and its 14 C age (figures not shown) are available in the directory "`/home/matsumot/tanio003/TMM2/steady_state/ecco_radiocarbon_bc.m`". To run this file, you need to use the shell script `ecco_radiocarbon_bc.sh` in the same directory and takes 10 minutes or so on MSI.

For this exercise, you would also need to copy the GLODAP surface boundary condition file "`bkgc14_cdiac_regird.mat`" in the same directory, which is a surface backgroud radiocarbon concentration regridded to the ECCO grid. I made the script "`regrid_BC.m`" in the directory "`/~/../tanio003/TMM2/steady_state/BC_regrid`" that can convert .nc to .mat format. Make sure to use Ferret to regrid Δ^{14}C data into ECCO grid before use MATLAB to convert from netcdf to matlab file. (There maybe a way to regrid the file in MATLAB but I don't know how.)

(3) Radiocarbon Δ^{14}C [Method 2]: Alternatively, we can impose the surface boundary condition where Δ^{14}C is restored to 0 ($\mathbf{c} = \mathbf{100}$) with a wind speed-dependent timescale, as done in Khatiwala et al. (2005) and

Khatiwala (2007). In this case, we can write Equation (3) as,

$$\mathbf{c}^{n+1} = \mathbf{A}_i^n(\mathbf{A}_e^n\mathbf{c}^n - k(\mathbf{c}^n - \mathbf{100}) - \lambda\mathbf{c}^n) \tag{10}$$

where k is the wind speed depdendent timescale following Equation (11) of Toggweiler et al. (1989):

$$k\ [\text{days}^{-1}] = \frac{3.5 \times (w-2)\ [\text{mol m}^{-2}\ \text{yr}^{-1}]}{2.0\ [\text{mol m}^{-3}] \times DZ_1 \times 365\ [\text{days yr}^{-1}]} \times 1.2 \tag{11}$$

where w is the spatially-variable wind speed in m s^{-1} and DZ_1 is the depth of the surface layer in meters (5 m in ECCO). Windspeed $w = 0$ when the grid box is completely covered by ice. At steady state, Equation (10) can be rearranged and solved for $\mathbf{c}$

$$\mathbf{c} = -(\mathbf{A}_i^I\mathbf{A}_e^I - \lambda\mathbf{A}_i^I - k\mathbf{A}_i^I - \mathbf{I})^{-1}\mathbf{A}_i^I(\mathbf{100}k) \tag{12}$$

There is a MATLAB script called in my directory "`/home/matsumot/tanio003/TMM2/steady_state/ecco_radiocarbon.m`" that computes Δ^{14}C and age this way. The result is very similar to that of the first method (Figure not shown). All the netcdf files are in my output directory "`/home/matsumot/tanio003/TMM2/steady_state/output`".

3 Appendix

3.1 Downloading PyFerret on MSI

1. Install miniconda on your home directory. This is a free cross-platform for Python distribution from Continuum Analytics.

```
$ cd ~
$ module purge
$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
$ bash Miniconda3-latest-Linux-x86_64.sh
```

   Follow the prompts on the installer screens. If you are unsure about any setting, accept the defaults. To make the changes take effect, close and then re-open your terminal window. Once installed you should find new directory called `miniconda3` on your home directory.

2. Execute the following command to install `pyferret` as well as `ferret_datasets` (the default Ferret datasets) into conda.

```
$ conda create -n PyFERRET -c conda-forge pyferret ferret_datasets --yes
```

   `PyFERRET` is the environment name where `pyferret` is installed. You can change that to any name you like.

3. To start using PyFerret, execute the following commands (replace `PyFERRET` with whatever environment name you used):

```
(base) $ conda activate PyFerret
(PyFERRET) $ pyferret
        NOAA/PMEL TMAP
        PyFerret v7.6 (optimized)
        Linux 4.15.0-1089-azure - 06/25/20
        18-Sep-20 15:07

yes?
```

4. (Optional) You may with to add directories to `FER_DATA` and `FER_GO`:

```
(PyFERRET) $ export FER_DATA="$FER_DATA␣/my/path/to/big/data/dir"
(PyFERRET) $ export FER_GO="$FER_GO␣/my/path/to/custom/ferret/scripts"
```

5. Once you are done working with PyFerret you can leave this environment, if you wish, with the command:

```
(PyFERRET) $ conda deactivate
(base) $
```

3.2 General reference for MOPS and MOBI

Key references for MOPS:

- Kriest, I. and Oschlies, A. (2015). Mops-1.0: modelling the regulation of the global oceanic nitrogen budget by marine biogeochemical processes. *Geoscientific Model Development*, 8:2929–2957
- Kriest, I., Kähler, P., Koeve, W., Kvale, K. F., Sauerland, V., and Oschlies, A. (2020). One size fits all? Calibrating an ocean biogeochemistry model for different circulations. *Biogeosciences*, 17(12):3057–3082
- Niemeyer, D., Kriest, I., and Oschlies, A. (2019). The effect of marine aggregate parameterisations on nutrients and oxygen minimum zones in a global biogeochemical model. *Biogeosciences*, 16(15):3095–3111

Key references for MOBI:

- Khatiwala, S., Schmittner, A., and Muglia, J. (2019). Air-sea disequilibrium enhances ocean carbon storage during glacial periods. *Science advances*, 5(6):eaaw4981
- Muglia, J., Somes, C. J., Nickelsen, L., and Schmittner, A. (2017). Combined effects of atmospheric and seafloor iron fluxes to the glacial ocean. *Paleoceanography*, 32(11):1204–1218
- Somes, C. J., Schmittner, A., Muglia, J., and Oschlies, A. (2017). A three-dimensional model of the marine nitrogen cycle during the last glacial maximum constrained by sedimentary isotopes. *Frontiers in Marine Science*, 4:108
- Kvale, K. F., Meissner, K., Keller, D., Eby, M., and Schmittner, A. (2015). Explicit planktic calcifiers in the university of victoria earth system climate model, version 2.9. *Atmosphere-Ocean*, 53(3):332–350
- Yamamoto, A., Abe-Ouchi, A., Shigemitsu, M., Oka, A., Takahashi, K., Ohgaito, R., and Yamanaka, Y. (2015). Global deep ocean oxygenation by enhanced ventilation in the southern ocean under long-term global warming. *Global Biogeochemical Cycles*, 29(10):1801–1815

3.3 TMM options for MOBI (listed at MOBI_TMM_OPTIONS.h)

The default setting is shown in the [define/undef]. To run MOBI2.0 with 30 tracers, these switches must be turned on:

1. O_npzd — NPZD model
2. O_npzd_o2 — O2 gas-exchange and aerobic remineralization
3. O_npzd_nitrogen — N fixation, Denitirication and NO3
4. O_npzd_nitrogen_15 — N15 of Phyto, Zoo, Detritus, DON, NO3
5. O_npzd_iron — dFe, pFe and Fe from sediment and dust, scavenging, hydrothermal
6. O_carbon — CO2 air-sea gas exchange, DIC speciation
7. O_npzd_alk — Alkalinity
8. O_npzd_caco3 — Carbonate chemistry
9. O_carbon_13 — C13 of DIC,DOC,POC,CaCO3 and plankton

```
O_TMM                          # use TMM?                          [define]
O_co2ccn_user                  # you must define thi               [define]
O_TMM_interactive_atmosphere   # interactive atmosphere?           [undef]
O_npzd_iron_diagnostics        # NPZD with iron?                   [define]
O_even_fluxes                  #                                   [define]
O_read_my_kmt                  #                                   [undef]
O_read_my_grid                 #                                   [define]
O_cyclic                       # cyclic grid?                      [define]
O_time_averages                # accumulate or "average and write" time mean data?   [undef]
O_time_step_monitor            # accumulate or "average and write" time mean integrated data? [undef]
O_sbc_in_memory                #                                   [define]
O_fourfil                      # Fourier filtering?                [define]
O_constant_flux_reference      #                                   [define]
O_embm                         # Atmospheric energy moisture balance model? [define]
O_embm_mgrid                   # EMBM on multigrid solver? [define]
```

```
O_embm_awind                    #                               [undef]
O_embm_adiff                    #                               [define]
O_save_embm_diff                #                               [define]
O_save_embm_wind                #                               [define]
O_ice                           # ice model?                    [define]
O_ice_evp                       # ice model with elastic-viscous-plastic rheology? [define]
O_ice_fourfil                   # Fourier filtering on ice model? [define]
O_mtlm                          # land surface and vegetation model? [undef]
O_mtlm_segday                   #                               [undef]
O_gthflx                        #                               [undef]
O_mom                           # Modular ocean model?          [define]
O_ramdrive                      # mom related                   [define]
O_conjugate_gradient            # mom related                   [define]
O_sf_5_point                    # mom related                   [define]
O_stream_function               # mom related                   [define]
O_consthmix                     # mom related                   [define]
O_constvmix                     # mom related                   [define]
O_fullconvect                   # mom related                   [define]
O_save_convection               # mom related                   [define]
O_stability_tests               # mom related                   [define]
O_gyre_components               # mom related                   [undef]
O_term_balances                 # mom related                   [define]
O_energy_analysis               # mom related                   [define]
O_meridional_overturning        # mom related                   [define]
O_tracer_averages               # accumulate tracers for averages under horizontal regions [define]
O_gent_mcwilliams               # Gent-McWilliams parameterization? [define]
O_isopycmix                     # Isopycnal mixing?             [define]
O_fct                           # mom related                   [define]
O_npzd                          # NPZD model?                   [define]
O_npzd_alk                      # NPZD with DIC and ALK?        [define]
O_npzd_caco3                    # NPZD with CaCO3?              [define]
O_kk_ballast                    # Ballasting?                   [undef]
O_npzd_nitrogen                 # NPZD with N cycle?            [define]
O_npzd_nitrogen_15              # NPZD wih N15?                 [define]
O_npzd_no_vflux                 # No vertical flux?             [undef]
O_npzd_o2                       # NPZD with O2?                 [define]
O_save_npzd                     #                               [define]
O_save_kv                       #                               [define]
O_tidal_kv                      #                               [define]
O_anisotropic_viscosity         #                               [define]
O_save_anisotropic_viscosity    #                               [define]
O_npzd_iron                     # NPZD with iron cycle?         [define]
O_npzd_iron_diagnostics         # Iron Diagnostics?             [undef]
O_npzd_extra_diagnostics        #                               [undef]

O_carbon                        # carbon chemistry?             [define]
O_carbon_13                     # 13C with C chemistry?         [define]
O_carbon_13_coupled             #                               [undef]
O_c13ccn_data                   # 13C forcing data?             [undef]
O_carbon_14                     # 14C of DIC?                   [define]
O_save_carbon_carbonate_chem    #                               [define]
O_co2ccn_data                   # CO2 forcing data?             [undef]
O_agric_data                    #                               [define]
O_landice_data                  #                               [define]
O_solar_data                    # Insolation forcing data?      [undef]
O_tai_otsf                      #                               [define]
O_tai_ns                        #                               [define]
O_tai_lo                        #                               [define]
O_tai_slh                       #                               [define]
O_tai_rad                       #                               [define]
O_units_temperature_Celsius     # temp in degC?                 [define]
O_units_time_years              # time in years?                [define]
O_save_time_relyear0            # save time from year 0         [define]

O_embm_annual                   # update EMBM annually?         [define]
O_volcano_data                  #                               [undef]
O_volcano_data_transient        #                               [undef]
O_co2emit_data_transient        # CO2 transient emission forcing? [undef]
O_co2ccn_data_transient         # CO2 transient concentration forcing? [undef]
O_co2emit_track_co2             # Track CO2 emissions           [undef]
O_co2emit_track_sat             # Tracks surface air temp?      [undef]
O_embm_vcs                      # CO2 from tracking air temp?   [undef]
O_c14ccn_data                   # 14C forcing?                  [undef]
O_c14ccn_data_transient         # 14C concentration transient?  [undef]
O_aggfor_data                   # Aggregating greenhouse gas forcing? [undef]
O_aggfor_data_transient         # Greenhouse gas forcing transient? [undef]
O_sealev_data_transient         # Sealevel transient?           [undef]
O_ice_cpts                      # cpts option for ice model?    [undef]
O_crop_data_transient           # Transient crop data?          [undef]
O_pasture_data_transient        # Transient pasture data?       [undef]
O_agric_data_transient          # Transient agricultural data?  [undef]
O_carbon_fnpzd                  # Read fixed fluxes for cabon and alkalinity? [undef]

O_sed                           # Dynamic sediment model?       [undef]
O_sed_weath_diag                # Diagostic sedimentary weathering? [define]
O_sed_constrain_rainr           # Set limits on rain ratio by limiting effective organic rain? [define]
```

3.4 Paramters for MOBI (listed at control.in)

- **contrl**:Variables related to general control. Declared in `$UVICESCMROOT/source/common/switch.h`.

Num	Parameters	Description	Unit	Default
1	*init*	This is a initial run (true) or from a restart	[true,false]	.false.
2	*runlen*	Integration period in "rununits"	[rununits]	36500.
3	*rununits*	Units of "runlen". Can be days, months, or years	[Days/Month/Years]	'days'
4	*restrt*	Write a restart at the end of the run	[true,false]	.true.
5	*runstep*	Integration period in timesteps	[timesteps]	2550000

- **tsteps**: Variables related to timestepping.

Num	Parameters	Description	Unit	Default
1	*dtts*	Timestep for density and tracers	[sec]	28800.
2	*dtuv*	Timestep of baroclininc velocity (internal mode)	[sec]	1125.
3	*dtsf*	Timestep of barotropic velocity	[sec]	1125.
4	*dtatm*	Timestep of atmospher	[sec]	54000.
5	*namix*	Timesteps between mixing	[timesteps]	10
6	*segtim*	Integration time is broken into number of segments each of length "segtim" days	[Days]	5.

- **riglid**: Variables related to numerical analysis thresholds.

Num	Parameters	Description	Unit	Default
1	*mxscan*	Max number of allowable scans for Poisson solvers	[-]	200
2	*sor*	Over-relaxation multiplier	[-]	1.60
3	*tolrsf*	Tolerance for stream function calculation	[-]	5.e8
4	*tolrsp*	Tolerance for surface pressure calcualtion	[-]	1.e-4
5	*tolrfs*	Tolerance for implicit free surface calculation	[-]	1.e-4

- **mixing**: Variables related to mixing. Most of these declared in `$UVICESCMROOT/source/mom/UVic_ESCM.F` but no descriptions given for some.

Num	Parameters	Description	Unit	Default
1	*am*	?	?	1.5e9
2	*ah*	?	?	8.e6
3	*ahbkg*	?	?	0.
4	*ambi*	Constant lateral biharminc viscosity coefficient for momentum	[-]	1.e23
5	*ahbi*	Constant lateral biharminc diffusion coefficient for tracers	[-]	5.e22
6	*kappa_m*	Constant vertical viscosity coefficient	[cm^2/sec]	10.
7	*kappa_h*	Constant vertical dissusion coefficient	[cm^2/sec]	0.35
8	*aidif*	Coefficient for implicit time differencing for vertical diffusion	[-]	0.5
9	*nmix*	Number of timesteps between mixing timesteps	[timesteps]	16
10	*eb*	Conifgures for the use of a Euler Backward mixing timestep	[true,false]	.false.
11	*ncon*	Number of passes through convection routine	[-]	1
12	*cdbot*	Bottom drag coefficient	[-]	1.3e-3
13	*acor*	Implicit Coriolis Factor (0.0 => 1.0)	[-]	0.
14	*dampts*	Timescale for damping surface tracers to data	[Days]	30.,60.
15	*dampdz*	Ocean level thicknesss for converting Newtonian damping	[cm]	50.e2,50.e2

- **diagn**: Variables related to time integral. Descriptions are found in `$UVICESCMROOT/source/common/switch.h`.

Num	Parameters	Description	Unit	Default
1	*tsiint*	Number of days between printing of time step integrals	[Days]	365.
2	*tsiper*	Averaging period for "`time_step_monitor`"	[Days]	365
3	*timavgint*	Interval for writing time mean data from the "averaging" grid	[Days]	36500.
4	*timavgper*	Averaging period for "`time_averages`"	[Days]	36500.
5	*restint*	Days between saving restarts	[Days]	36500.
6	*tavgint*	Days between regional tracer averages (under horizontal regions)	[Days]	-365000.
7	*itavg*	Write regional mask info the tracer average diagnostic	[true,false]	.true.
8	*tmbint*	Number of days over which tracer equation in averaged in depth and longitude	[Days]	-365000.
9	*itmb*	Write "msktmb" for tracer meridional balance diagnostic.True for the new runs.	[true,false]	.true.
10	*trmbint*	Days between momentum and tracer term balances	[Days]	-365000.
11	*itrmb*	Write regional mask info for the term balance diagnostic	[true,false]	.true.
12	*tbtint*	Averaging period for writing term balances	[Days]	-365000.
13	*tbtper*	Averaging period for tracer term balances	[Days]	-365.
14	*glenint*	Number of days between global energetics integrals	[Days]	-365000.
15	*vmsfint*	Number of days between calculation of vertical and meridional stream function	[Days]	-365000.

16	*stabint*	Number of days between sampling for various stability criteria	[Days]	-365000.
17	*zmbcint*	Number of days between calculation of zonal mean surface boundary conditions	[Days]	-365000.
18	*gyreint*	Number of days between calculation of traccer northward transport	[Days]	-365000.
19	*extint*	Number of days between printouts of external mode	[Days]	-365000.
20	*prxzint*	Number of days between printouts of x-z data	[Days]	-365000.
21	*dspint*	Number of days between surface pressure calculation	[Days]	-365000.
22	*trajint*	?	[Days]	-365000.
23	*xbtint*	Averaging period for xriting XBT data	[Days]	-365000.

- **io**: Variables related to ocean circulation module. Descriptions are found in `$UVICESCMROOT/source/common/iounit.h` and `$UVICESCMROOT/source/common/diag.h`.

Num	Parameters	Description	Unit	Default
1	*expnam*	60 character text string for experiment name	['Name']	' '
2	*iotavg*	Control # for tracer averages	[-]	6
3	*iotmb*	Control # for writing meridional tracer budget	[-]	6
4	*iotrmb*	Control # for term balances for tracer and momentum	[-]	6
5	*iozmbc*	Control # for writing zonal mean surf boundary conditions	[-]	6
6	*ioglen*	Control # for writing global energetics integrals	[-]	6
7	*iovmsf*	Control # for writing meridional stream function	[-]	6
8	*iogyre*	Control # for writing gyre transport	[-]	6
9	*ioprxz*	Control # for writing x-z sections from latitudes	[-]	6
10	*ioext*	Control # for writing external mode (stream function)	[-]	6
11	*iodsp*	Control # for writing diagnostic surface pressure	[-]	6
12	*iotsi*	Control # for writing time step integrals	[-]	6
13	*iotraj*	?	[?]	6
14	*ioxbt*	Control # for writing time averaged xbt data	[-]	6
15	*mrot*	Regional mask region for max/min overturning	[-]	1
16	*jsot*	Starting j index for max/min overturning	[-]	65
17	*jeot*	Ending j index for max/min overturning	[-]	90
18	*ksot*	Starting k index for max/min overturning	[-]	3
19	*keot*	Ending k index for max/min overturning	[-]	17

- **ictime**: Variables related to time management. Descriptions are found in `$UVICESCMROOT/source/common/tmngr.h`, `$UVICESCMROOT/source/common/tmngr.F`, and `$UVICESCMROOT/source/common/switch.h`.

Num	Parameters	Description	Unit	Default
1	*eqyear*	Use constant length year calendar?	[true,false]	.true.
2	*eqmon*	Use 12 equal months of monlen days each?	[true,false]	.false.
3	*refinit*	reference to initial condition time	[true,false]	.true.
4	*refrun*	referenced to the start of each run	[true,false]	.false.
5	*init_time_in*	sets input restart to initial time	[true,false]	.false.
6	*init_time_out*	sets output restart to initial time	[true,false]	.false.
7	*year*0	year of initial conditions	[year]	0
8	*month*0	month of initial conditions	[month]	1
9	*day*0	day of initial conditions	[day]	1
10	*hour*0	hour of initial conditions	[hour]	0
11	*min*0	minute of initial conditions	[minute]	0
12	*sec*0	second of initial conditions	[second]	0

- **blmix**: Variables related to Bryan-Lewis mixing model.

Num	Parameters	Description	Unit	Default
1	*AHV*	Vertical diffusion coefficient	[-]	0.6

- **hlmix**: Variables related to horizontal mixing. Default = not given.

- **isopyc**: Variables related to the isopycnal mixing scheme. Descriptions are found in `$UVICESCMROOT/source/mom/isopycnal.F`.

Num	Parameters	Description	Unit	Default
1	*slmx*	max slope of isopycnals	[-]	0.01
2	*ahisop*	isopycnal tracer diffusivity	[cm^2/sec]	1.2e7
3	*athkdf*	isopycnal thickness diffusivity	[cm^2/sec]	8.e6
4	*del_dm*	transition for scaling diffusion coefficient	[cm^2/sec]	0.4e-2
5	*s_dm*	half width scaling for diffusion coefficient	[cm^2/sec]	0.1e-2

- **ppmix**: Variables related to the Pacanowski/Philander vertical mixing scheme. Default = not given.
- **smagnl**: Variables related to the Smagorinsky nonlinear horizontal viscosity. Default = not given.
- **embm**: Variables related to the energy balance atmosphere model.

Num	Parameters	Description	Unit	Default
1	$adiff$	atmospheric diffusivity?	[?]	0.03

- **carbon**:Variables related to carbon cycle in the energy balance atmosphere model. Descriptions are found in `$UVICESCMROOT/source/common/cembm.h`.

Num	Parameters	Description	Unit	Default
1	$co2ccn$	atmospheric CO2 concentration	[ppmv]	277.4412
2	$dc14ccn$	atmospheric dC14 concentration	[permil]	0.0

- **paleo**: Variable related to paleoclimatology and paleoceanography. Descriptions are found in `$UVICESCMROOT/source/common/cembm.h`.

Num	Parameters	Description	Unit	Default
1	$pyear$	default paleo calendar year (-/+ = BC/AD)	[year]	1765.

- **ice**: Variables related to ice module. Default = not given.
- **veg**: Variables related to land/vegetation module. Default = not given.
- **mtlm**:Variables related to land-surface and vegetation model.Descriptions are found in `$UVICESCMROOT/source/common/mtlm.h`.

Num	Parameters	Description	Unit	Default
1	BF	Burn fraction	[-]	0.0

- **npzd**: Parameters related to biogeochemical NPZD model. Descriptions are from https://github.com/OSU-CEOAS-Schmittner/UVicMOBI_fct_npzd_o2_n_fe_ca_caco3_c13_n15/blob/master/updates/npzd_src.F

Num	Parameters	Description	Unit	Default
1	$k1n$	Half saturation constant for N uptake	$[mmol/m^3]$	0.7
2	$capr$	carbonate to carbon production ratio	$[molar]$	0.055
3	$abio\_P$	a; Maximum growth rate parameter	$[1/day]$	0.6
4	$alpha$	Initial slope P-I curve	$[(W/m^2)^{-1}/day]$	0.16
5	nup	Specific mortality rate (Phytoplankton)	$[1/day]$	0.03
6	$nupt0$	Fast-recycling mortality rate (Phytoplankton)	$[1/day]$	0.001
7	$redctn$	C/N Redfield ratio	$[molar]$	7
8	geZ	Zooplankton growth efficiency	[-]	0.54
9	$redotn$	O2/N Redfield ratio	$[molar]$	11.0
10	$kfemin$	Minimum half saturation constant for Fe limitation	$[mmolFe/m^3]$	0.04e-3
11	$kfemax$	Maximum half saturation constant for Fe limitation	$[mmolFe/m^3]$	0.4e-3
12	$kfemin\_C$	Minimum half saturation constant for Fe limitation (Cocco)	$[mmolFe/m^3]$	0.04e-3
13	$kfemax\_C$	Maximum half saturation constant for Fe limitation (Cocco)	$[mmolFe/m^3]$	0.4e-3
14	kfe_D	Half saturation constant for Diaz Fe limitation	$[mmolFe/m^3]$	0.1e-3
15	$zprefC$	Zooplankton preference for C	[-]	0.21
16	$zprefP$	Zooplankton preference for P	[-]	0.24
17	$dissk0$	initial dissolution rate parameter	[-]	0.013
18	$gbio$	Maximum grazing rate	$[1/day]$	0.37
19	$wd0$	Sinking speed of detritus at surface	$[m/day]$	15
20	nuz	Quadratic mortality (zpk)	$[(mmol/m^3)^{-2}day^{-1}]$	0.06
21	$jdiar$	factor reducing the growth rate of diazotrophs	[-]	0.2
22	$nud0$	detritus remineralization rate	$[1/day]$	0.07
23	$sgbdfac$	sub-grid benthic denitrification rate factor	[-]	2.0
24	$diazntp$	diazotroph N:P ratio	[molar]	40.0
25	nup_D	Specific mortaility rate (Diazotrophs)	$[1/day]$	0.0001
26	$abio\_C$	a; Maximum growth rate parameter (Cocco)	$[1/day]$	0.3
27	$k1n\_C$	Half sat const for N uptake (Cocco)	$[mmol/m^3]$	0.35
28	$alpha\_C$	Initial slope P-I curve (Cocco)	$[(W/m^2)^{-1}/day]$	0.1
29	$nuct0$	Fast-recycling mortality rate (Cocco)	$[1/day]$	0.013
30	$nupt0\_D$	Fast-recyling mortality rate (Diazotrophs)	$[1/day]$	0.001
31	dfr	phyt mortality refractory/semi-labile DOM fraction	[-]	0.1
32	$dfrt$	phyt fast-recy refracotyr/semi-labile DOM fraction	[-]	0.08

33	$dtnpzd$	time step of biology	[s]	14400.0
34	$hdop$	DOP growth rate handicap	[-]	0.4
35	$nudop0$	DOP remineralization rate	$[1/day]$	2.0e-5
35	$nudon0$	DON remineralization rate	$[1/day]$	1.0e-5
36	eps_assim	Scaling factor for assimilation	[-]	6.0
37	eps_excr	Scaling factor for excretion	[-]	4.0
38	eps_recy	Scaling factor for recycling	[-]	1.3
39	eps_wcdeni	Scaling factor for water column denitrification	[-]	25.0
40	eps_bdeni0	Scaling factor for benthic denitrification	[-]	4.0
41	eps_nfix	Scaling factor for benthic N fixation	[-]	1.0
42	mwz	Depth where sinking below remains constant	$[cm]$	100000
43	mw	Sinking speed increase with depth	$[s^{-1}]$	0.055

To find out what these paramters actually do in the model, you may need to refer to several description papers:

- Original NPZD model (Oschlies and Garçon, 1999; Giraud et al., 2000; Schmittner et al., 2005, 2008).
- Zooplankton grazing formulation (Keller et al., 2012)
- Nitrogen isotope model (Somes et al., 2010a,b, 2013)
- Carbon isotope (^{13}C) model (Schmittner et al., 2013)
- Prognostic $CaCO_3$, Coccolithophores, and balast (Kvale et al., 2015)
- Iron Model (Nickelsen et al., 2015)
- Benthic denitrification and sedimentary iron release (Somes and Oschlies, 2015; Muglia et al., 2017)

- **sed**: Variables related to sediment module. Default = not given.

References

Giraud, X., Bertrand, P., Dadou, I., et al. (2000). Modeling δ15n evolution: First palaeoceanographic applications in a coastal upwelling system. *Journal of marine research*, 58(4):609–630.

Keller, D., Oschlies, A., and Eby, M. (2012). A new marine ecosystem model for the university of victoria earth system climate model. *Geoscientific Model Development*, 5:1195–1220.

Key, R. M., Kozyr, A., Sabine, C. L., Lee, K., Wanninkhof, R., Bullister, J. L., Feely, R. A., Millero, F. J., Mordy, C., and Peng, T. H. (2004). A global ocean carbon climatology: Results from Global Data Analysis Project (GLODAP). *Global Biogeochemical Cycles*, 18(4):1–23.

Khatiwala, S. (2007). A computational framework for simulation of biogeochemical tracers in the ocean. *Global Biogeochemical Cycles*, 21(3).

Khatiwala, S., Schmittner, A., and Muglia, J. (2019). Air-sea disequilibrium enhances ocean carbon storage during glacial periods. *Science advances*, 5(6):eaaw4981.

Khatiwala, S., Visbeck, M., and Cane, M. A. (2005). Accelerated simulation of passive tracers in ocean circulation models. *Ocean Modelling*, 9(1):51–69.

Kriest, I., Kähler, P., Koeve, W., Kvale, K. F., Sauerland, V., and Oschlies, A. (2020). One size fits all? Calibrating an ocean biogeochemistry model for different circulations. *Biogeosciences*, 17(12):3057–3082.

Kriest, I. and Oschlies, A. (2015). Mops-1.0: modelling the regulation of the global oceanic nitrogen budget by marine biogeochemical processes. *Geoscientific Model Development*, 8:2929–2957.

Kvale, K. F., Meissner, K., Keller, D., Eby, M., and Schmittner, A. (2015). Explicit planktic calcifiers in the university of victoria earth system climate model, version 2.9. *Atmosphere-Ocean*, 53(3):332–350.

Lovenduski, N. S., McKinley, G. A., Fay, A. R., Lindsay, K., and Long, M. C. (2016). Partitioning uncertainty in ocean carbon uptake projections: Internal variability, emission scenario, and model structure. *Global Biogeochemical Cycles*, 30(9):1276–1287.

Meinshausen, M., Smith, S. J., Calvin, K., Daniel, J. S., Kainuma, M., Lamarque, J.-F., Matsumoto, K., Montzka, S., Raper, S., Riahi, K., et al. (2011). The rcp greenhouse gas concentrations and their extensions from 1765 to 2300. *Climatic change*, 109(1-2):213.

Muglia, J., Somes, C. J., Nickelsen, L., and Schmittner, A. (2017). Combined effects of atmospheric and seafloor iron fluxes to the glacial ocean. *Paleoceanography*, 32(11):1204–1218.

Nickelsen, L., Keller, D., and Oschlies, A. (2015). A dynamic marine iron cycle module coupled to the university of victoria earth system model: the kiel marine biogeochemical model 2 for uvic 2.9. *Geoscientific Model Development*, 8:1357–1381.

Niemeyer, D., Kriest, I., and Oschlies, A. (2019). The effect of marine aggregate parameterisations on nutrients and oxygen minimum zones in a global biogeochemical model. *Biogeosciences*, 16(15):3095–3111.

Oschlies, A. and Garçon, V. (1999). An eddy-permitting coupled physical-biological model of the north atlantic: 1. sensitivity to advection numerics and mixed layer physics. *Global Biogeochemical Cycles*, 13(1):135–160.

Persson, J., Fink, P., Goto, A., Hood, J. M., Jonas, J., and Kato, S. (2010). To be or not to be what you eat: Regulation of stoichiometric homeostasis among autotrophs and heterotrophs. *Oikos*, 119(5):741–751.

Schmittner, A., Gruber, N., Mix, A. C., Key, R. M., Tagliabue, A., and Westberry, T. K. (2013). Biology and air-sea gas exchange controls on the distribution of carbon isotope ratios (delta c-13) in the ocean. *Biogeosciences*, 10(9):5793–5816.

Schmittner, A., Oschlies, A., Giraud, X., Eby, M., and Simmons, H. (2005). A global model of the marine ecosystem for long-term simulations: Sensitivity to ocean mixing, buoyancy forcing, particle sinking, and dissolved organic matter cycling. *Global Biogeochemical Cycles*, 19(3).

Schmittner, A., Oschlies, A., Matthews, H. D., and Galbraith, E. D. (2008). Future changes in climate, ocean circulation, ecosystems, and biogeochemical cycling simulated for a business-as-usual co2 emission scenario until year 4000 ad. *Global biogeochemical cycles*, 22(1).

Somes, C. J. and Oschlies, A. (2015). On the influence of non-redfield dissolved organic nutrient dynamics on the spatial distribution of n2 fixation and the size of the marine fixed nitrogen inventory. *Global Biogeochemical Cycles*, 29(7):973–993.

Somes, C. J., Oschlies, A., and Schmittner, A. (2013). Isotopic constraints on the pre-industrial oceanic nitrogen budget. *Biogeosciences*, 10(9):5889–5910.

Somes, C. J., Schmittner, A., and Altabet, M. A. (2010a). Nitrogen isotope simulations show the importance of atmospheric iron deposition for nitrogen fixation across the pacific ocean. *Geophysical research letters*, 37(23).

Somes, C. J., Schmittner, A., Galbraith, E. D., Lehmann, M. F., Altabet, M. A., Montoya, J. P., Letelier, R. M., Mix, A. C., Bourbonnais, A., and Eby, M. (2010b). Simulating the global distribution of nitrogen isotopes in the ocean. *Global Biogeochemical Cycles*, 24(4).

Somes, C. J., Schmittner, A., Muglia, J., and Oschlies, A. (2017). A three-dimensional model of the marine nitrogen cycle during the last glacial maximum constrained by sedimentary isotopes. *Frontiers in Marine Science*, 4:108.

Tanioka, T. and Matsumoto, K. (2020). A meta-analysis on environmental drivers of marine phytoplankton C : N : P. *Biogeosciences*, 17(11):2939–2954.

Toggweiler, J. R., Dixon, K., and Bryan, K. (1989). Simulations of radiocarbon in a coarse-resolution world ocean model: 1. Steady state prebomb distributions. *Journal of Geophysical Research*, 94(C6):8217.

Woolf, D. K., Shutler, J. D., Goddijn-Murphy, L., Watson, A., Chapron, B., Nightingale, P. D., Donlon, C. J., Piskozub, J., Yelland, M., Ashton, I., et al. (2019). Key uncertainties in the recent air-sea flux of co2. *Global Biogeochemical Cycles*, 33(12):1548–1563.

Yamamoto, A., Abe-Ouchi, A., Shigemitsu, M., Oka, A., Takahashi, K., Ohgaito, R., and Yamanaka, Y. (2015). Global deep ocean oxygenation by enhanced ventilation in the southern ocean under long-term global warming. *Global Biogeochemical Cycles*, 29(10):1801–1815.