**Building Offline Transport Operators for CESM POP version 2.2.0, X3**

**On GreenPlanet**

These instructions are for building the tracer transport operators for a CESM POP2 offline model from the CESM POP2 parent.

This version of the instructions is for version 2.2.0, nominal 3° horizontal resolution, on UCI’s GreenPlanet computer complex. However, the instructions can easily be adapted for another location or model resolution.

This document incorporates information from Keith Lindsay, Keith Moore and Ann Bardin. References used in this document are listed at the end of this document.

It is assumed that the POP2 parent model has been spun-up dynamically, and that a restart file exists. For execution on GreenPlanet, it is assumed that the CESM library is in Keith Moore’s (JKM’s) directories:

/DFS-L/DATA/moore/jkmoore/cesm2.2.0

The process can be seen as consisting of several steps:

1. Run the dynamic model with the Impulse-Response-Function (IRF) functionality turned on. Output files with the average IRF responses are generated.
2. Transform the IRF responses into monthly operators.
3. Make a set of climatologically-averaged monthly operators, with an adjustment for SSH in the surface layer.

It is suggested that the user read through the detailed directions for each major step before implementing them.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**STEP 1**: Make a branch run of the POP ocean model, with the IRF module installed. Reference documents for setting up, modifying, and running the dynamic model are online. For the CESM User’s Guide see

https://www.cesm.ucar.edu/models/?ref=hp

Start with CESM2.2 Quickstart Guide, which describes such details as component namelists and .xml files.

The directory structure assumed in the build scripts, and as implemented on GreenPlanet, for the IRF generation run is as shown in the figure below. The dotted lines show the parts of the directory structure that are added as part of the build.



The \*.notes files contain the script to be executed to implement the generation of the IRF output files. The \*.notes files for running the IRF-takeoff was adapted from Keith Moore’s \*.notes file and from Keith Lindsay’s \*.notes files.

**Step 1a**: An IRF-definition file gives the locations for the Impulses for each of the tracers. For the X3, v2.2.0 configuration, this is included with the CESM release. If you have a different configuration, see the instructions in Appendix C for generating this file.

**Step 1b**: If this is a branch from someone else’s run, copy the /archive/[run-name]/restart/[restart\_time/\* to your own /archive/. (This is necessary because of a flaw in the restart processing in the CESM release v2.2.0).

**Step 1b**: Copy to your own directory set an example of the \*.notes file, which contains the job setup commands to build the dynamic model including the IRF tracers. As shown in the figure above, this is in directory /CESM\_run/.

Modify the \*.notes to be your own case and parent CESM model.

RUN\_REFCASE is the dynamic model run from which the IRF run is branched.

RUN\_REFDATE is date of branching (in model time)

Scripts assume that restart files for $RUN\_REFCASE at $RUN\_REFDATE are in the directory:

/$RUN\_REFCASE/$RUN\_REFDATE-00000.

These definitions are set in the \*.notes file.

Note that the IRF tracer must be the last one listed under passive tracers. Note that POP\_TAVG\_R8=”TRUE”, which calls for double-precision output is essential for obtaining sufficiently accurate transport matrices.

If there are other mods to the standard model, they will need to be placed in the …/SourceMods/src.pop/ directory before the executable is built, similarly to the KM\_MODs in the example .notes script.

Note that the IRF output variables are hardcoded to tavg stream 1 (ocn.IRF.setup.csh). For information on how to change tavg variables and data stream assignments, refer to the CESM User’s Guide.

The IRF script is set up to generate monthly-average files for the IRF data. The mechanisms for changes to the namelist and tavg names and frequencies are described in the CESM Users Guide.

The \*.notes file is scripted to be run interactively – without the comments.

**Step 1c:** Build the branch run case, with the IRF passive tracer set on.

The IRF-modifications to the POP source code have been incorporated into the standard release CESM v2.2.0. A description of the modifications is in Appendix E.

An example script for obtaining output of IRF variables with 5-day means, is also included in /CESM\_run/.

**Step 1d:** Run the dynamic model to generate the IRF output.

Execute the ./case.submit command at the end of the job script.

To monitor the job: ‘squeue –u {username}’ gives the status for all of your jobs.

In the example, the IRF output from the run is in

/DFS-L/SCRATCH/moore/{username}/archive/{CASEname}/ocn/hist/.

The processing of the IRF output to make the offline operators begins with Step 2.

**STEP 2**: Transform the IRF responses into monthly operators.

Appendix A gives the build code file organization.

**STEP 2a:**

For a new configuration, the IRF stencils, used to transform the IRF responses into operators, will need to be built. The directions for building the IRF stencils are given in Appendix B.

For the X3 v2.2.0 configuration the stencils are saved in

/DFS-L/DATA/moore/abardin/IRF\_masks/CESM2.2.0\_X3/

**STEP 2b: Build the “noc” monthly operators from the IRF output.**

The IRF monthly output in the CESM history files is converted into monthly operators. The raw monthly operators are kept in the “noc”\_directory as building blocks for building a climatologically averaged set of operators. The set of operators that is built in the “noc” directory is extracted from the IRF output, and does not have any SSH adjustment factor. Each monthly operator set contains the operators A (advection), H (horizontal diffusion), D (vertical diffusion), and T (sum of A, H, and D).

In the “noc”\_directory, the month number starts with the first month of the first model year to be processed, and the month number continues consecutively. For example, if there are 10 years worth of output files, they are numbered 1 through 120.

**STEP 2b1**: Edit [case].slurm job file

The .slurm file specifies the location and quantity of IRF-output files to process. It executes Build\_monthly\_ops.m for each year’s worth of IRF-output files.

Modify the reference file names. These are the IRF-takeoff output files that are in the /archive/. Modify the location of the “noc\_dir”, where the unadjusted monthly operators are to go. Modify the “zeroyr”, which is the year before the first year of the IRF-takeoff output.

**STEP 2b2: IFF this is a new configuration,** edit the buildMET.m files.

See Appendix D, Editing buildMET.m files.

**STEP 2b3:** Execute [case].slurm to generate the ‘noc’ operators

“sbatch [case].slurm”

The control flow is as follows:

buildMET; builds the geometric data that is needed for processing the operators and/or running with them. MET is saved in the noc\_ops directory.

make\_ops\_from\_IRF\_output: builds the ocean-sized matrix operators (excludes land points) from the OGCM netcdf IRF output files. The operators have not yet been adjusted to make them annually-periodic, and are thus in an intermediate state. The files are output to the noc\_dir directory.

Build\_state\_data: collects selected monthly data from the dynamic model’s output files, and saves it in the noc directory.

The monthly operator sets are in files MTM[n].mat Each MTM file contains the operators A (advection), H (horizontal diffusion), D (vertical diffusion), and T (total A + H + D).

The state data sets are in files model\_state\_data\_[n].mat. Each file contains SSH, and other data useful for biogeochemical processing.

**STEP 3**: **Generate the climatologically averaged operator set**

The end-product are operators A, H, D, T, plus the periodic adjustment term dxidt. These are organized into monthly files plus an annual average file in the final directory.

The final climatologically averaged operator set generated in the "ops\_dir" directory contains a set of monthly operators, numbered 1 through 12, and the annual average, numbered 0.

The code for this is in the /multi\_yr\_build/ directory. In the control file, [case][first\_yr][nyears].slurm, are defined the noc directory, the final ops directory, which years and how-many years to use for the climatological average. The final operator directory contains only monthly files that are included in the annualized average, numbered according the calendar month, and the annualized average (MTM0.mat). The periodic adjustment has been applied to the sets of operators. The state-data files are also climatologically averaged and included in the ops directory.

The major logic for generating the climatologically averaged operator set is in the following files:

make\_A\_periodic: Computes the periodic adjustment for the surface layer from the divergence of the annual advection operator. This is applied to the A and T operators for all the months in the annual set, in addition to the annual average A and T.

For each month, the effect of the change in sea-surface height (SSH) is calculated from the row-divergence in the surface layer grid-cells, to compensate for the SSH changes in volume during the month. The term is dxidt, which is the correction (dxi) per unit time (dt). It is used when stepping the offline model through a seasonal cycle per equation (12) of Bardin et al.(2014).

Mass-balance and row-divergence are checked on the resulting operators. Checks are also made that each IRF pulse occurred once, and only once; and that operator fields are not on land.

Each monthly output file contains A, H, D, T, dxidt in the final ops\_dir directory.

build\_model\_state\_data: Builds .mat files containing SSH, TEMP (potential temperature), PD (potential density), SALT (salinity), RHO (in-situ density), IFRAC (sea-ice fraction, and other variables that are convenient to have for additional BGC processing.

**Enjoy using the offline operators.**

An example of using the operators is given for radiocarbon in Bardin et al. (2014).

**Appendix A: File Organization for building transport operators:**

The code directory structure is shown in the figure below.

Under the top-level /X3\_v2.2.0/ is a text case log, which describes each CESM run case, and lists transport operator sets derived from the run.

In /CESM\_run/ can be found the scripts for running CESM for IRF-takeoff; for collecting MARBL forcings; and for various cases of parameter changes for MARBL.

Subdirectory /IRF\_mods/ contains no mods as of v2.2.0. (Thank you Keith Lindsay!) It does contain an NCL script for generating the impluse map used in the IRF-takeoff, and the output file containing the impulse map. These are only needed as models if using a non-standard ocean configuration.

Subdirectory /KM\_mods/ contains modified versions of MARBL and POP parameter files that define the initial set-up of the v2.2.0 CESM for execution with MARBL.

Subdirectory /MARBL\_mods/ contains modified versions of user\_nl\_marbl, user\_nl\_pop, and several tavg\_content files that are for making variations of the MARBL parameters and output. Also present are any post-run processing scripts, such as for changing the source of the tracers from an initial value file to the restart file.

Subdirectory /POP\_mods/ contains a modified namelist\_defaults file, with intent to limit output.

Other subdirectories may contain special-purpose POP or MARBL modifications; the intent is to segregate short-lived modifications.

In /build\_ops/ can be found the MATLAB routines to build a set of transport matrices from the output of an IRF-takeoff run.

The files are listed by top-level-script or function, then by the functions called. The file ending of X1 or X3 means they contain configuration details for the nominal 1-degree and 3-degree configurations of the POP model.

Subdirectory /noc\_ops\_build/ contains the software to take the IRF-output and make it into a set of non-corrected (for sea-surface height changes) monthly transport operators.

BldStencils.slurm: job control to build the IRF stencils (aka IRF masks).

Build\_stencils\_X3.m builds the IRF set of stencils needed to pick up the output from the IRF dynamic parent model runs.

Uses internal functions:

make\_stencils\_61

constructs the regular IRF masks, and those for the source

and entrainment regions.

Uses internal functions:

setup\_stencil\_bands\_61

makes the X, Y, and Z bands for the POP stencils from the dimensions specified

make\_prod\_stencil\_61

constructs the extended masks for the product areas

At the end of the Build\_stencils\_X3.m file, are the definitions for the formatting of the product regions definition file.

{case}.slurm: is the control job that builds the noc operator set and state data. A new version of this should be generated for each case run.

Build\_monthly\_ops.m builds the advection/diffusion operators for the offline model from the history files resulting from running the POP ocean model with the "IRF tracers" turned on. This script invokes functions which output files containing a sparse matrix with each operator (A (advection), H (horizontal diffusion, D (vertical diffusion), for each month specified in the control job. The monthly files are saved in the noc operator directory.

Uses functions:

buildMET.m

gathers the geometric data from a netcdf history file

make\_ops\_from\_IRF\_output.m

collects the output from the netcdf history files, and

makes the operators

Uses functions:

get\_g\_data.m

form A and H operators from the regular IRF output

get\_s\_data.m

form A and H operators from overflow source IRF output

get\_e\_data.m

form A and H operators from overflow entrainment IRF output

get\_VDC\_data.m

get vertical diffusion parameters from IRF output

Vdiff.m

make D operators from VDC data

pulse\_ck.m

check IRF pulse locations are in all ocean grid cells, and only once for each grid cell

Subdirectory /multi\_yr\_build/ contains the code to make the climatologically averaged set of operators from the un-adjusted monthly operators.

Build\_multi\_period.m makes the adjustment for annual periodicity to the operators, and creates the climatologically averaged operators.

Uses functions:

make\_annual\_ops.m makes annualized average operators with the adjustment, and saves them in the ops directory

make\_A\_periodic.m calculates the 2-D modification to the surface grid cells for the advection operator (dxidt), to make it cyclic on an annualized basis, without divergence. Each file contains A, H, D, T, dxidt.

Uses internal functions

mkAtilde2d makes the annual periodic-adjustment operator for advection.

grad makes the gradient operator from the grid-cell geometry.

div makes the divergence operator from the grid-cell geometry.

load\_iocn\_ops loads the un-adjusted operators from the noc directory.

ck\_ops\_massb\_div.m provides mass-balance and row-sum checks.

Build\_mo\_climate\_state.m builds a set of climatological monthly data files containing SSH, and other model data frequently needed for biogeochemical models.

The other directories contain code for specific tasks related to the use of the transport operators.

/build\_forcings/ contains the code to gather the forcings needed to run MARBL from the dynamic model, and format it for MARBL’s use.

/exam\_hist/ contains code to examine the history output from the dynamic model for changes in value with time.

/tracers/ contains the code to run a few tracers that help characterize the ocean circulation to an equilibrium solution, and make graphs of the results.

**Appendix B: Building IRF stencils for a new configuration**

This is the procedure for building the IRF stencils for transforming the IRF output into transport operators. (Step 2a)

The definitions for the overflow source, entrainment, and product regions have been adapted from the gx3v7\_overflow file, which is part of the standard release, and can be found in [cesm\_version]/models/ocn/pop2/input\_templates/

The definition for the product region is from an edited version of the gx3v7\_overflow file.

The source and entrainment regions are hard-coded in build\_stencils\_X3.m. The product regions are read in from a text file that must be created as described in STEP 2a1.

**STEP 2a1:**

Edit POP source file \*\_overflow, to create \*\_overflow\_data.txt, formatted so that the Matlab script can read it. There are formatting and editing definitions at the end of build\_stencils\_X3.m

**STEP 2a2**:

Edit the parameters at the beginning of build\_stencils\_X3.m to define the file names and directories to access, and the "expected configuration dimensions", which is used as a check for consistency with the netcdf output data from the model run. For a new configuration, the definitions for the source and entrainment regions, which are hard-coded in build\_stencils\_X3.m, will need to be modified.

**STEP 2a3:**

Execute build\_stencils\_X3.m. The IRF masks are built, and saved in the specified directory.

The remainder of this Appendix are notes giving additional details of the code that makes the stencils (masks).

There are three types of masks used. Each mask deals with the possible results of one IRF source grid-cell, and the possible grid-cells that may be impacted after one timestep. The g\_irf\_mask is for the general case, where the impulse may have an impact on any cell within 2 grid-cells in each of the 3 dimensions. These masks define a 3D box with a total of 125 grid-cells, with the impulse grid-cell in the middle. The s\_irf\_mask defines the locations impacted from an impulse in the overflow source region. This includes anywhere in the source region itself, plus 2 grid-cells bordering the source region; specified grid-cells in the associated overflow entrainment region; and specified grid-cells in the associated overflow product region. The e\_irf\_mask is defined for the entrainment region, since it acts as an additional source. The e\_irf\_mask is for an impulse within the entrainment area, and includes the entrainment area, and the associated product grid-cells further down-slope.

In build\_stencils\_X3.m, the parameters are defined for the general IRF masks. The numbers of masks are defined by PARAM.g\_imin (1) to PARAM.g\_imax (number of cells in the x –dimension), etc. for each dimension. The z-dimension has an extra layer compared to the OGCM to ensure that there is land (or air) on all boundaries. PARAM.g\_X\_band defines how many cells there are on each side of the subject Impulse cell in the x dimension (=2). This is similar for the other dimensions. PARAM.g\_Y\_limit (= 0) is a constraint on the Y dimension, which has no effect on the general IRF mask, but is provided for compatibility with this parameter for the source and entrainment definitions.

The source and entrainment definitions are given in a format that matches that in the Fortran source code. Each column defines one source or associated entrainment region’s location in terms of the xyz cell indices. The PARAM.Y\_limit is a constraint in the y direction to assure that there is no overlap. All of the parameters are passed to the function make\_irf\_masks, which is in the same file.

Function make\_irf\_masks first makes the regular masks, in a 2-step process. It uses function setup\_irf\_bands to set up a band of matrices in each dimension. The bands are then propagated in each direction using the kron function, which makes a matrix of matrices (the Kronecker tensor product).

Function setup\_irf\_bands builds an nx by nx matrix, with bands on the diagonal that are + and – X\_band (defined in the PARAMs) wide. The band is extended to deal with the wrap-around of the globe in the x-dimension.

The band in the y-dimension is built in a similar manner, but with a limitation on the dimension that keeps regions from overlapping. This has no effect for the general IRF mask, and actually only has an effect if the definitions of the overflow regions overlap. A similar banded matrix is made for the z-dimension.

Within function make\_irf\_masks, the overflow “source” region masks are made in a similar manner, but within the dimensions of each source region. The masks are extended with the “product” regions, which include areas within the associated “entrainment” region and further down-slope. The overflow entrainment masks are defined in a similar manner.

Within function make\_irf\_masks, the product region masks are created in function make\_prod\_mask. This is a separate function because the product-area masks are defined in a different way. The locations of the product regions are determined by reading the product\_txt\_file. The way that the product areas are specified is somewhat convoluted; there are some detailed notes on the format of the file at the end of file build\_stencils\_X3.m. There may be several product areas associated with one overflow region, including parts of the entrainment area, and further downslope. A mask is created that includes all of the possible product area grid-cells for the overflow region.

**Appendix C: Building the IRF definition file**

IFF it does not exist for the model configuration to be used, generate the IRF-definition file:

IRF\_offline\_transport\_tracers\_’grid\_name’.nc

The script to generate the IRF definitions for offline transport in a netcdf file is in directory /IRF\_mods/:

gen\_IRF\_offline\_transport.ncl

(This was copied from /glade/p/cgd/oce/people/klindsay/cesm12\_cases/IRF/ on Cheyenne.)

Executing the script will generate 125 standard IRF tracers (with 1s and 0s), and additional IRF tracers for the overflow “source” and “entrainment” regions. The generated IRF masks are stored in a netcdf file: IRF\_offline\_transport\_tracers\_’grid\_name’.nc.

The script should be modified for the following:

- output directory (“basis\_function\_dir”)

- grid\_name (“grid\_name”), valid values are "gx1v6" or "gx3v7"

* For other versions of POP, check also for new names for “grid\_dir” and “grid\_fname”, for the input grid definitions.
* Note that the locations for the overflow special handling, “source” and “entrainment” regions, are hard-coded in this routine.

**Appendix D: Edit the buildMET files for a different configuration**

buildMET.m contains the definitions of the mask-outs for the marginal seas that are not included in the offline model. It also contains the expected dimensions of the model, in order to check for a consistent definition. BuildMET gets the geometry parameters from the referenced history file.

ms\_buildMET.m, which generates the geometric definitions without the marginal seas masked out, also contains the expected dimensions of the model, and must be made consistent with buildMET.m.

Note that for v2.2.0, some of the dimensions on land bordering the ocean have a “no-data” flag value in the history file. Because the geometry configuration is the same as for v1.2, those missing values were copied from a v1.2 configuration. (Otherwise, some of the bulk 3-D processing done on matrices does not work properly.)

**Appendix E: IRF mods to POP**

The IRF modifications that were added to POP are described below. These have been incorporated in the CESM v2.2.0 release, and are included here for information only.

./build-namelist

This is a perl script that constructs POP's namelist file, pop2\_in. Modifications are to include namelist variables for IRF\_nml, the namelist.

For IRF mods, the IRF tracer module namelist variables are:

irf\_tracer\_file

IRF\_MODE

The generated pop2\_in file is in $CASEROOT/CaseDocs/pop2\_in.

./namelist\_definition\_pop2.xml

This is an xml file with documentation of all namelist variables. Modifications are to include IRF\_nml variables.

./namelist\_defaults\_pop2.xml

This is an xml file with default values for all namelist variables. Modifications are to include defaults for IRF\_nml variables.

./pop2.buildexe.csh

This is the script to build the POP2 executable. Modifications are to add IRF tracers to POP's full tracer count.

./ocn.IRF.tavg.csh

This is a new file that will in future releases be placed in directory:

$CCSMROOT/models/ocn/pop2/input\_templates.

This script generates tavg output variable names:

HDIF\_EXPLICIT\_3D\_$var\_name

ADV\_3D\_$var\_name

VDC\_GM: on the bottom face of a cell.

./advection.F90

Modifications are to add ADV\_3D\_ vars and positive and negative parts of velocity as variables that can be recorded using the tavg mechanism.

./baroclinic.F90

Modifications are to add a new argument to reset\_passive\_tracers.

./hmix\_gm.F90

Modifications are to add VDC\_GM tavg variable.

./horizontal\_mix.F90

Modifications are to add HDIF\_EXPLICIT\_3D\_ variables.

./passive\_tracers.F90

Modifications are to add a call to the IRF\_mod subroutines.

./IRF\_mod.F90

This is a new file that in future releases will be placed in directory:

$CCSMROOT/models/ocn/pop2/source

This is the module that implements the IRF tracers in the dynamic model.

(vertical\_mix.F90 is not modified

(VDC\_KPP\_{1,2} in the previous IRF mod set now exists in standard code now as VDC\_{T,S}.)

VDC\_{T,S} are on the bottom face of a cell.)

**APPENDIX F: Notes** on changes going from v1.0 to v1.2

The interface to the prognostic POP model is organized differently. The output variables in the history files have the same names as much as possible, however, there are some changes:

- For the VDC data, version 1.2 name VDC\_T (for temperature) replaces v1.0 name VDC\_KPP\_1 (not used for the standard operators).

- Version 1.2 name VDC\_S (for salinity) replaces v1.0 name VDC\_KPP\_2.

- For both VDC\_S and VDC\_GM, the values given are on the BOTTOM of the cell, rather than the top. The two variables need to be added together to get the total coefficient for vertical diffusion, as before.

Another significant change is that the pulse locations are not output in the history file. Instead, they can be read from a file that was created from the definition of the pulse locations.

This file is copied into the output /case\_name/ directory for access when building the offline operators. It was found that this file has in addition to the pulses in the ocean, pulses in all the land grid-cells, which must be masked out to make the file usable.

The variable KPP\_NONLOC\_KERN in version 1.0 is not recorded in the history file; the processing for the corresponding off-line operator has been deleted.

All 178 of the tracers were captured in a single 1-year run; as opposed to the previous 7 runs required in version 1.0. Changes were made in BUILD\_OPS.m and functions it uses to eliminate the number of cases required, and the file handling to extract the data.

**References:**

Bardin, Ann, Francois Primeau, Keith Lindsay, An offline implicit solver for simulating prebomb radiocarbon, Ocean Modelling (2014)

Briegleb, B.P., Danabasoglu, G., Large, W.G., 2010. An Overflow Parameterization for the Ocean Component of the Community Climate System Model. NCAR Technical Note. NCAR.

Smith et al., Parallel ocean program (POP) reference manual (2010)

Online users guide:

CESM Quickstart Guide (CESM2.2)

https://escomp.github.io/CESM/versions/cesm2.2/html/