CRESM : The Community Regional Earth System Model

Version 1.0.0

User Guide

by

JAISON KURIAN

(*jaisonk@tamu.edu*)

Department of Oceanography
Texas A&M University
3416 TAMU
College Station, Texas 77843-3146

September 2017

Copyright

See *Montuoro* (2017) for original copyright details about the CRESM model and first documentation (January 31, 2017).

Preface

This is a user manual to help CRESM (version 1.0.0 assembled by *Montuoro* (2017)) users. Typically, model user guides describe model features and provide few examples. A more practical hands-on approach is used here focusing on learning phases. This documentation is developed based on authors experience with running this model in a fully coupled mode for a Gulf of Mexico configuration. Therefore some of the information provided here is directly from the practical experience rather than solid knowledge about the model. This deficiency will be fixed in the future versions of the user manual. In its present form, this documentation may be better suited for users with some prior numerical modelling experience. Future version of the documentation may include additional sections to help novice modellers. Users are strongly encouraged to give feedback to update this documentation and its contents.

Contents

Chapter 1

Quick Start Guide

This is a quick start guide to CRESM version 1.0.0 and a Gulf of Mexico test case. The test case can be used without any modification on TAMU super computer Ada but may need changes to port on to other computers or to adapt the test case for new experiments.

CRESM source code and test case bundle comes with 3 tar files:

1. cresm-1.0.0.tar.gz (source code)
2. cresm-1.0.0_gom_input.tar (input files for test case, 4.5 GB)
3. E01_run1_output.tar (output files, test case run 1, 20 GB) (optional)
4. E01_run2_output.tar (output files, test case run 2, 20 GB) (optional)

Please contact Prof. Ping Chang (ping@tamu.edu) for the source code (1) and Jaison Kurian (jaisonk@tamu.edu) for input and output files (2-3).

Please use full path to the directory or file represented with preceeding dots like ".../CRESM/lib" in this user guide.

1.1 Compiling CRESM

1.1.1 Requirements

Make sure the availability of following modules/libraries.

1. Fortran and C compilers (eg. ifort, icc)
2. MPI Libraries (mpiifort, mpicc)
3. MCT (see Section 1.1.3 for installation instructions)
4. PIO (see Section 1.1.3 for installation instructions)
5. netCDF

6. pnetCDF

On TAMU Ada, the following worked fine (August/2017), for Fortran, C, MPI, netCDF and pnetcdf libraries.

```
[user@comp]$ module purge
[user@comp]$ module load wrf-deps/intel-2015B
[user@comp]$ module load pnetcdf/1.6.1-intel-2015B
```

pnetCDF version 1.6.1-intel-2015B (from /software/tamusc/pnetcdf/1.6.1-intel-2015B) has been used in this example here. Please see Section 1.1.3 for installation instructions for MCT and PIO. Please see Section 3.2.1 for running without pnetcdf library.

1.1.2 Source code and directory tree

Please obtain the CRESM source code from Prof. Ping Chang (*ping@tamu.edu*). Make a directory tree to place CRESM related files as follows.

```
[user@comp]$ mkdir /scratch/user/jaisonk/Models/CRESM
[user@comp]$ cd    /scratch/user/jaisonk/Models/CRESM
[user@comp]$ mkdir src run lib
```

Unzip the source code to a subdirectory with proper name:

```
[user@comp]$ mkdir -p src/cresm-1.0.0/E01_test
[user@comp]$ cd       src/cresm-1.0.0/E01_test
[user@comp]$ tar -xf  /path/to/this/file/cresm-1.0.0.tar.gz
```

1.1.3 Install mct and pio

This section describes the method to install mct (Model Coupling Toolkit) and pio (Parallel Input Output) using the packages and tools provided along with the CRESM source code.

Load proper modules for fortran compiler with mpi libaries and for the netCDF libraries. On TAMU Ada, the following worked fine (August/2017):

```
[user@comp]$ module purge; module load wrf-deps/intel-2015B
[user@comp]$ module load pnetcdf/1.6.1-intel-2015B
```

To install mct, edit E01_test/tools/build_mct.sh with the "−−prefix" pointing to ..../CRESM/lib/mct and run this script from E01_test/models/utils/mct

```
[user@comp]$ pwd   #  E01_test/models/utils/mct
[user@comp]$ ../../../tools/build_mct.sh  # relative path
[user@comp]$ make
[user@comp]$ make install
[user@comp]$ make clean
[user@comp]$ ls ..../Models/CRESM/lib/mct/ # check installation
```

To install pio, edit E01_test/tools/build_pio.sh with the "−−prefix" pointing to ..../CRESM/lib/pio. Also edit MCT_PATH, NETCDF_PATH and PNETCDF_PATH as appropriate. On Ada pnetCDF version 1.6.1-intel-2015B (path /software/tamusc/pnetcdf/1.6.1-intel-2015B) worked fine. Then run this script from E01_test/models/utils/pio. The "make install" step may fail and if it fails, manually copy relevant files to ..../CRESM/lib/pio (29 files of *.mod and *.h and 1 file of .a type).

```
[user@comp]$ pwd   #  E01_test/models/utils/pio
[user@comp]$ ../../../tools/build_pio.sh  # relative path
[user@comp]$ make
[user@comp]$ make install
[user@comp]$ mkdir -p /scratch/user/jaisonk/Models/CRESM/lib/pio/lib
[user@comp]$ mkdir /scratch/user/jaisonk/Models/CRESM/lib/pio/include
[user@comp]$ cp *.mod *.h /scratch/..../CRESM/lib/pio/include
[user@comp]$ cp *.a /scratch/..../CRESM/lib/pio/lib
[user@comp]$ ls /scratch/..../CRESM/lib/pio/ # check installation
```

1.1.4 Compile source code

Make a directory to place the final CRESM related files (preferably within E01_test subdirectory). Then edit the "−−prefix" in run_config.sh to reflect above "exec" directory. Also edit the mct, pio, pnetcdf and netcdf library paths as appropriate.

```
[user@comp]$ pwd  # /scratch/...../src/cresm-1.0.0/E01_test
[user@comp]$ mkdir  exec
[user@comp]$ vi     run_config.sh   # edit "--prefix" & lib paths
```

Now, compile and install CRESM as follows:

```
[user@comp]$ ./run_config.sh
[user@comp]$ make
[user@comp]$ make install
[user@comp]$ ls ./exec/bin  # location of cresm executable
```

Upon successful completion of compilation, the cresm executable will be in exec/bin directory (filename 'cresm'). Files in exec/lib directory (lib*.a files for atm, csm_share, esmf_time, glc, tptl, ice, lnd, ocn, & rof) are not needed to run CRESM. Files in exec/include (config.h, ocn.h, & txglo_3dnudg.h) will serve as a reference for model configuration for future use.

1.2 Input Files

The standard test case for CRESM is a Gulf of Mexico configuration with 9 km WRF and 3 km ROMS. Please obtain the input data files (cresm-1.0.0_gom_input.tar.gz) from Jaison Kurian (jaisonk@tamu.edu).

These files are for a run initialized on 2010-01-01_00:00:00. Time varying data is available till 2010-01-11_00:00:0.

There are several different input files for the CRESM, which are discssed in detail in Chapter 4. See this section before trying to adapt the test case discussed below for a new application.

Please note that this test case have 3D-nuding option for ROMS (nudging 3D temperature and salinity to HYCOM data). This is not normally used for ROMS runs. See Section 3.4.2 for tuning off the 3D-nudging.

1.3 Running CRESM

Now, set up the run directory and copy input files.

```
[user@comp]$ mkdir -p /scratch/..../CRESM/run/E01_run1
[user@comp]$ cd /scratch/..../CRESM/run/E01_run1
[user@comp]$ tar -xf /path/to/file/cresm-1.0.0_gom_input.tar.gz
```

Also make links of WRF table files from src directory
(/scratch/.../CRESM/src/cresm-1.0.0/E01_test/models/atm/wrf/3.5.1/WRFV3/run/*)

```
[user@comp]$ ln -s /scratch/...../3.5.1/WRFV3/run/* ./
```

Copy or link CRESM executable to run directory
(from /scratch/.../CRESM/src/cresm-1.0.0/E01_test/exec/bin)

```
[user@comp]$ ln -s /scratch/...../E01_test/exec/bin/cresm ./
```

Now edit the run_cresm.job file as appropriate. Please keep the number of cores the same. If using pnetcdf on Ada, the job file should have correct pnetcdf files as follows (just before the mpiexec command):

```
export PNETCDF_HINTS="striping_unit=16777216"
export PNETCDF_PATH=/software/tamusc/pnetcdf/1.6.1-intel-2015B
export PNETCDF=/software/tamusc/pnetcdf/1.6.1-intel-2015B
```

Then submit the job file.

```
[user@comp]$ bsub < run_cresm.job
```

In order to monitor a run, please look at the log files.

```
[user@comp]$ tail ocn.log
[user@comp]$ tail cpl.log
[user@comp]$ tail atm.log
[user@comp]$ tail rsl.out.0000
```

Log files will be the best place to check and make sure the run has completed (Section 5.4.2). In case the test run encounter any run time issues, please see Chapter 2 for known issues and fixes.

1.4 Checking Output

Please obtain E01_run1_output.tar from Jaison Kurian (jaisonk@tamu.edu) for verifying the output from above test run and E01_run2_output.tar for verifying the restart run discussed in Section 1.5.

Make a directory and untar the output.tar file.

```
[user@comp]$ mkdir /scratch/user/..../CRESM/test_case
[user@comp]$ cd /scratch/user/..../CRESM/test_case
[user@comp]$ tar -xvf /path/to/this/file/E01_run1_output.tar
```

Directories E01_run1_output and E01_run2_output correspond to the run described in Section 1.3 and in Section 1.5 respectively. The output netCDF file string has the following meaning:

TXGLO:	name of the domain/experiment
atm/ocn/cpl:	component model (atm=atmosphere, ocn=ocean, cpl=coupler)
hi/r:	file type (hi=history, r=restart)
YYYY-MM-DD_hh:mm:ss:	date and time for first record in the file

For More details about the CRESM output files, please see Section 4.5. Compare the output netCDF files in this directory with that from the test run (described in Section 1.3). Make sure it matches perfectly. If the test case is made on a different machine than TAMU Ada, you may notice slight missmatch in values (eg. difference around 3rd or 4th decimal place for K=50 temp in ocean history files) which is quite normal.

Also compare the log files to see they have similar integrated values (eg. ocn.log) and run times.

1.5 Restarting CRESM

Assume E01_run1 is the existing run directory, with restart files (from the run described inSection 1.3). In order to make a restart run, make a new run directory E01_run2.

```
[user@comp]$ mkdir -p /scratch/..../CRESM/run/E01_run2
[user@comp]$ cd /scratch/..../CRESM/run/E01_run2
```

Now use the mkcopy.csh tool (see Section 5.4.1) to copy all the input files from previous run directory (/scratch/..../CRESM/run/E01_run1) to current restart run directory. Here gom03 is the prefix for ROMS files, "../E01_run1" is the source run directory and "./" is the destination run directory.

```
[user@comp]$ ~/bin/mkcopy.csh gom03 ../E01_run1 ./
```

Manually copy the map files from E01_run01

```
[user@comp]$ cp -a ../E01_run1/map_???_????.nc .
```

Please note that "rpointer.*" files in E01_run1 will have the most recent restart dates (here 2010-01-04_00:00:00). If the restart date is different, please edit the "rpointer.*" files.

From E01_run1 directory, link ROMS, WRF, CPL restart file to E01_run2. Use the restart date string to copy just the required restart files.

```
[user@comp]$ ln -s ../E01_run1/TXGLO.*.r.2010-01-04*.nc ./
```

ROMS model can handle multiple boundary (BRYNAME) and nudging (CLMNAME) netCDF files (eg. one file for each month and 12 such files for a 1-year run). For the restart example here, the BRYNAME and CLMNAME for ROMS can be the same but for real-life applications, it can be different. Also note that the CLMNAME is required only if ROMS is running with 3D-nudging (see Section 3.4.2).

For the 3 day run from the restart file for 2010-01-04_00:00:00, the model start date/time is 2010-01-04_00:00:00 and the end time is 2010-01-07_00:00:00. Edit the following files as suggested (for details on the input files discussed below, please see Chapter 4).

1. ocean.in
   (a) change NTIMES from 8640 to 17280 (note that ROMS need cumulative number of timesteps from the very first run).
   (b) change NRREC from 0 to 1 (check ROMS restart file to make sure)
   (c) change ININAME from ./gom03_N050_md15m_ini.....201001.nc to ./TXGLO.ocn.r.2010-01-04_00:00:00.nc
   (d) Keep BRYNAME the same
   (e) Keep CLMNAME the same
2. drv_in (&seq_timemgr_inparm)
   (a) start date: change start_ymd from 20100101 to 20100104
   (b) run length: Keep stop_n the same, 3 (if run length is different change this)
   (c) restart interval: Keep restart_n the same, 3 (if run length is different change this)
3. ocn_in (&ocn_timemgr)

   (a) change start_day from 01 to 04 (in general, edit the start year, month, day, hour, minute, and second according to the restart time)

4. docn_ocn_in
   (a) keep strems the same (the numeric entries represent yrAlign, yrFirst & yrLast, please see Section 4.3.2.2 for details on editing this for a multi-year run)

5. namelist.input (&time_control)
   (a) change start_day from 01 to 04 (in general, edit the start year, month, day, hour, minute, and second according to the restart time)
   (b) change end_day from 04 to 07 (in general, edit the end year, month, day, hour, minute, and second according to the run end time)
   (c) change restart from .false. to .true. (this is a restart run).
   (d) keep restart_interval_d same at 3 (restart interval in days, change if needed)

6. rpointer.
   (a) Make sure the entries in every rpointer file refers to the model restart date and time (2010-01-04_00:00:00).
   (b) Please mind the syntax difference between rpointer files.
   (c) Edit if required (see Section 4.3.2.1 for details).

With these changes the restart run is ready to be submitted. After submitting the job, monitor its progress and completion as described in Section 1.3. It is highly recommended to check the output as described in Section 1.4.

1.6 Adapting Test Case for Other Applications

The guidelines to adapt the Gulf of Mexico test case and cresm_1.0.0 source code discussed in previous sections for a new experiment are provided here.

1.6.1 Changes to source code and default configuration

The source code for WRF in cresm_1.0.0 is identical to *Montuoro* (2017) with default choices. However, there are slight changes for the ROMS source code. The original version of those edited files are provided in respective directories with "_org" suffix. To find those files, and see what has changed, please do the following:

```
[user@comp]$ cd ......./cresm-1.0.0
[user@comp]$ find ./ -name '*_org' -print
```

```
[user@comp]$ diff -EbwB ./models/csm_share/shr/shr_stream_mod.F90 \
        ./models/csm_share/shr/shr_stream_mod.F90_org
```

Here is a complete list of edited files compared to *Montuoro* (2017):

1. models/csm_share/shr/shr_stream_mod.F90
   Minor editing to print out model year, data year, yrFirst, yrLast, yrAlign etc.
2. models/ocn/roms/ROMS/Modules/mod_scalars.F
   lmd_Cv (Ratio of interior Brunt-Vaisala frequency to that at entrainment depth "he") changed from 1.25 to 1.8 and lmd_Ric (Critical bulk Richardson number) changed from 0.3 to 0.15. Without this change (and KPP scheme) the Gulf of Mexico test case was blowing up few months into the run.
3. models/ocn/roms/ROMS/Nonlinear/diag.F
   Edited for printing out few ROMS internal variables to better debug NaN blowups.

Other than these modifications, this source code is identical to *Montuoro* (2017). If default ROMS settings are preffered, please make the following changes:

1. ROMS 3D-nudging: The default method or running ROMS is without 3D-nudging. To adapt the CRESM Gulf of Mexico test case for an experiment without ROMS 3D-nudging, please see the instructions in Section 3.4.2.
2. ROMS LMD vertical mixing: The Gulf of Mexico test case had blow up issues with standard LMD (KPP) set up. In order to fix this, the mod_scalars.F program (models/ocn/roms/ROMS/Modules/mod_scalars.F) has been edited. If default settings are preferred, please change lmd_Cv (Ratio of interior Brunt-Vaisala frequency to that at entrainment depth "he") to 1.25 and lmd_Ric (Critical bulk Richardson number) to 0.3 before compiling the source code. See Section 1.6.1 for details.

1.6.2 Changes to Input Files

Please take a look at Chapter 3 to know the basic details and requirements of CRESM modeling framework. Based on the specific options selected for the application with CRESM, WRF and ROMS, prepare input files. Input files are discussed in detail in Chapter 4. Use the input files from the Gulf of Mexico test case as a reference. Tools and scripts to create input netCDF files are provided in Chapter 5.

As with any models, all CRESM, WRF and ROMS files are model grid dependent. Please recreate all input files for any experiments new model grid. Also note that the xroms files and mapping weight files (Section 4.3) are also grid dependend and needs to be recreated with change in any component model grids.

Chapter 2

Porting and Fixes to Known Issues

2.1 Porting CRESM

The default CRESM code and test cases are tuned to work on TAMU Ada. For porting it on to other machines, some changes may be required as detailed in this section.

2.1.1 TAMU Ada

On TAMU Ada, the following settings worked fine (last test Sep/2017) for Fortran, C, MPI, netCDF and pnetcdf libraries.

```
[user@comp]$ module purge
[user@comp]$ module load wrf-deps/intel-2015B
[user@comp]$ module load pnetcdf/1.6.1-intel-2015B
```

The default CRESM source code and test cases may be used without any modification on TAMU Ada.

A note about pnetcdf on Ada. On Ada “filesystem-hints=gpfs" (while installing pio), and “stripingunit=16777216" (run_cresm.job). For reference, the complete environmental variables for run_cresm.job on Ada is provided below:

```
ulimit -s unlimited
module purge
module load wrf-deps/intel-2015B
module load pnetcdf/1.6.1-intel-2015B
export I_MPI_HYDRA_BOOTSTRAP=lsf
export I_MPI_LSF_USE_COLLECTIVE_LAUNCH=1
```

```
# set following to the number of hosts ie. (-n value)/(ptile value)
export I_MPI_HYDRA_BRANCH_COUNT=18
export PNETCDF_HINTS="striping_unit=16777216"
```

2.1.2 UCAR Cheyenne

Compared to the default version (tuned for TAMU Ada), NCAR Cheyenne needs some modification (as of Oct/10/2017) for the following reasons:

1. On Cheyenne, mpiifort and mpiicc are present in proper location (path set using “module load" command) but they do not work. So, some of the files need to be edited to set mpif90 and mpicc as default compilers.
2. pnetcdf library is available on Cheyenne (module load pnetcdf/1.8.0) but there is absolutely no documentation online about the “filesystem-hints" (while installing pio) and the environmental variables (needed at runtime, like “striping_unit"). Hence use CRESM without pnetcdf on Cheyenne.
3. makefile for ROMS includes a check for the version of “make" but it do not have the version 4.0 (which is the one available on Cheyenne) in the check list.

The default modules worked fine on Cheyenne. A list of loaded modules are given below.

```
[user@comp]$ module list
1) ncarenv/1.2    3) ncarcompilers/0.4.1   5) netcdf/4.4.1.1
2) intel/17.0.1   4) mpt/2.15f
```

The complete list of modifications to port default CRESM on to NCAR Cheyenne is given below.

1. Before installing libraries (Section 1.1.3)
   (a) Edit build_mct.sh etry for “MPIFC" as follows (before mct installation) export MPIFC=mpif90
   (b) Edit build_pio.sh with correct entries for MPIFC and MPICC and remove any reference to pnetcdf (before installing pio). The edited build_pio.sh is given below.

```
#!/bin/bash
export CC=icc
export FC=ifort
export MPIFC=mpif90
export MPICC=mpicc
export MCT_PATH=/glade/p/work/...../Models/CRESM/lib/mct
export NETCDF_PATH=/glade/u/apps/ch/opt/netcdf/...../17.0.1
./configure --prefix=/glade/p/work/...../Models/CRESM/lib/pio
```

2. Before compilation (Section 1.1.4)

   (a) Edit the "configure" file (cresm-1.0.0/configure) as follows:

   ```
   change following line from:
       for ac_prog in mpiifort mpif90 mpfort
   to
       for ac_prog in mpif90 mpiifort mpfort
   and following line from
       for ac_prog in mpiicc mpicc mpcc
   to
       for ac_prog in mpicc mpiicc mpcc
   ```

   (b) Edit ROMS makefile (cresm-1.0.0/models/ocn/roms/makefile) as follows:

   ```
   change following line from:
       NEED_VERSION := 3.80 3.81 3.82
   to
       NEED_VERSION := 3.80 3.81 3.82 4.0
   ```

   (c) Edit "run_config.sh" (cresm-1.0.0/run_config.sh) to remove the entry for pnetcdf ("−−with-pnetcdf=/path/to/pnetcdf")(also see Section 3.2.1)

3. Before running (Section 1.3)

   (a) Edit WRF input file (namelist.input in run directory) entries for io_form_history & restart as follows (value 2 is for non-pnetcdf case and value 11 is for pnetcdf case (see Section 3.2.1)

   ```
       io_form_history = 2
       io_form_restart = 2
   ```

(b) Edit “run_cresm.job" (run directory) to remove any entries specific to pnetcdf (like striping_unit, see Section 2.1.1).

2.2 Known Issues and Fixes

2.2.1 Cannot Find Compiler

Issue: On UCAR Cheyenne, with default modules loaded (ncarenv, ncarcompilers, intel & mpt), the "which mpiifort" command returns the complete path to mpiifort. However, while compiling the model (Section 1.1.4) or trying to install mct or pio (see Section 1.1.3), the following error pops up:

```
which: no mpiifort in (/glade/u/apps/ch/opt/vim/8.0.0273/........
```

Same error message appear with mpiicc too.

Fix: If mpif90 and mpicc are working, then use them with cresm-1.0.0/tools/build_mct.sh, cresm-1.0.0/tools/build_pio.sh, & cresm-1.0.0/configure (see Section 2.1.2) instead of mpiifort and mpiicc.

2.2.2 Make version Error with ROMS

Issue: CRESM compilation (Section 1.1.4) fails for ROMS component with the following error:

```
Compilation Status : Error : makefile:32: *** This makefile
      requires one of GNU make version 3.80 3.81 3.82..  Stop.
```

Fix: ROMS makefile (cresm-1.0.0/models/ocn/roms/makefile) has a check for the version of “make" command against a predefined list, which includes 3.80, 3.81 & 3.82. On machines with a different version of make, this check will fail. For make versions closer to 3.8 (say 4.0) simply add the version number to the existing list to make it work (Section 2.1.2):

```
change following line from:
    NEED_VERSION := 3.80 3.81 3.82
to
    NEED_VERSION := 3.80 3.81 3.82 4.0
```

More careful editing of the makefile may be required if the version of make command is very different from 3.8.

2.2.3 CRESM SIGSEGV Error

Issue: CRESM fails during run time with the following error:

```
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image              PC                Routine         Line     Source
cresm              0000000002F4E4A1  Unknown         Unknown  Unknown
.............................................................
libpthread-2.19.s  00002AAAAAEE3870  Unknown         Unknown  Unknown
libmpi_mt.so       00002AAAAB57C9ED  MPI_SGI_bcast   Unknown  Unknown
```

Fix: This error happens when pio and CRESM were compiled with pnetcdf options but runtime pnetcdf options are not set up correctly. So, either compile without pnetcdf (Sections 3.2.1 and 2.1.2) or set pnetcdf environment properly (Section 2.1.1).

2.2.4 ROMS Blow-up Issue

Issue: The ROMS component of CRESM sometimes blows up with NaN values in the Potential Energy and Total Energy fields. If this happens, please search for the string "NaN" in ocn.log file in run directory. Check whether you see lines like:

```
 2881    1 00:00:30 1.150065E-02 1.543130E+04 1.543131E+04 3.563578E+15
      (282,410,25) 2.585091E-03 4.415171E-04 9.561279E-02 1.917248E+00
 2882    1 00:01:00 1.150041E-02 1.543130E+04 1.543131E+04 3.563578E+15
      (282,410,25) 2.587113E-03 4.457352E-04 9.571663E-02 1.917091E+00
 2883    1 00:01:30 1.150017E-02          NaN          NaN 3.563578E+15
      (282,410,25) 2.588798E-03 4.498424E-04 9.583168E-02 1.916931E+00
```

As shown above, typically the blow-up happens at 2883'rd time step from the start. It could happen for the restart runs too (eg. restart time step 8640 and blows up at time step 11523). This blow-up is related to some model bug (but not from model inputs) and the exact reason has not been figured out yet.

Fix: Resubmitting the run/job again works fine during most of the time. Please consider cleaning up the run directory (see Section 5.4.4) before resubmitting the job.

2.2.5 WRF SIGINT Error

Issue: For some reason, with the intel/MPI libraries on Ada (see Section 1.1.1), the CRESM run do not exit gracefully. With respect to the model run and model output, restart and log files, everything will be complete but still the job do not exit the queue. The CRESM log file (cresm.log) do report some memory related issues as shown below.

```
forrtl: error (69): process interrupted (SIGINT)
Image        PC                Routine     Line     Source
cresm        0000000002B24EA1  Unknown     Unknown  Unknown
```

Fix: Could not find a clean solution yet. Please see Section 5.4.2 to decide whether a CRESM run is complete in terms of log files. If the job is complete but still showing up in the queue, please kill the job. As a work around, please make sure you only request for reasonable wall clock time limits for your job.

2.2.6 MCT error with mapping files

Issue: CRESM fails/hangs with the errors in rsl.err.0000 similar to:

```
SOIL TEXTURE CLASSIFICATION = STAS FOUND  19 CATEGORIES
  MCT::m_SparseMatrixPlus:: FATAL--length of vector y different
           from row count of sMat.Length of y =   320295 Number
           of rows in sMat =   694216
  000.MCT(MPEU)::die.: from MCT::m_SparseMatrixPlus::initDistributed_()
           application called MPI_Abort(MPI_COMM_WORLD, 2) - process 0
```

Fix: This happens when mapping weight files (Section 4.3.2.5) specified in seq_maps.rc (Table 4.4) do not match with the CRESM configuration (whether using ROMS or xROMS). When using xROMS, the mapping is from WRF to xROMS (extented ROMS). But with ROMS (configured with -enable-atm-sst) the mapping is from WRF to ROMS grid. For details about mapping weight files, please see Section 4.3.2.5. Also see Section 3.1 to know the difference between ROMS and xROMS configuration.

2.2.7 Error with Data Ocean Year

Issue: A run starting in 2010 and extending to 2011 (or longer) may fail with error messages in data.ocn.log like:

```
(docn_comp_run) ocn: model date 20101231   64800s
(shr_stream_findBounds) ERROR: limit on and rDateIn gt rDategvd
   20101231.7604167        20101231.7500000
```

```
(shr_sys_abort) ERROR: (shr_stream_findBounds) ERROR: rDateIn gt
                     rDategvd limit true
(shr_sys_abort) WARNING: calling shr_mpi_abort() and stopping
```

or like:

```
(shr_sys_abort) ERROR: (shr_stream_findBounds) ERROR: LVD not found,
                    all data is after yearLast
(shr_sys_abort) WARNING: calling shr_mpi_abort() and stopping
```

Fix: Please edit "docn_ocn_in" (Section 4.3.2.2) as

```
streams = "docn.streams.txt.prescribed yrAlign yrFirst yrLast"
```

So, for a run starting anywhere in 2010 and ending in 2011, yrAlign and yrFirst are 2010 and yrLast is 2011. Please note that yrAlign should be same as yrFirst always!!!!!

Chapter 3

CRESM Modeling Framework

CRESM version 1.0.0 is based on Cummunity Earth System Model (CESM) coupling framework (CPL7, *Craig et al.* (2012)) distributed with CESM version 1.1 (*CESM1.1*). It includes active and data regional ocean, atmosphere, land and ice components (Table 3.1). The Gulf of Mexico test case utilize WRF for atmospheric model and xROMS for ocean component. Important aspects of the component models are discussed in this chapter.

3.1 Ocean Component

Typically the WRF domain is designed to cover slightly bigger area than the ROMS grid so that the artifacts from WRF boundary region will not influence the ROMS fields. In addition WRF and ROMS can have different horizontal resolutions and hence different grid point locations. CRESM provides two optios, using ROMS and xROMS, to handle these differences.

3.1.1 ROMS

To use ROMS ocean component (reffered to as “embedded ROMS" in *Montuoro* (2017)), compile CRESM (see Section 1.1.4) with

```
--enable-ocn=roms
--enable-atm-sst=yes
--enable-ocn-sfc=wrf
```

Please note that this configuration can only use ocean surface scheme from WRF (−−enable-ocn-sfc=wrf). With this configuration, CRESM will automatically fill out the missing

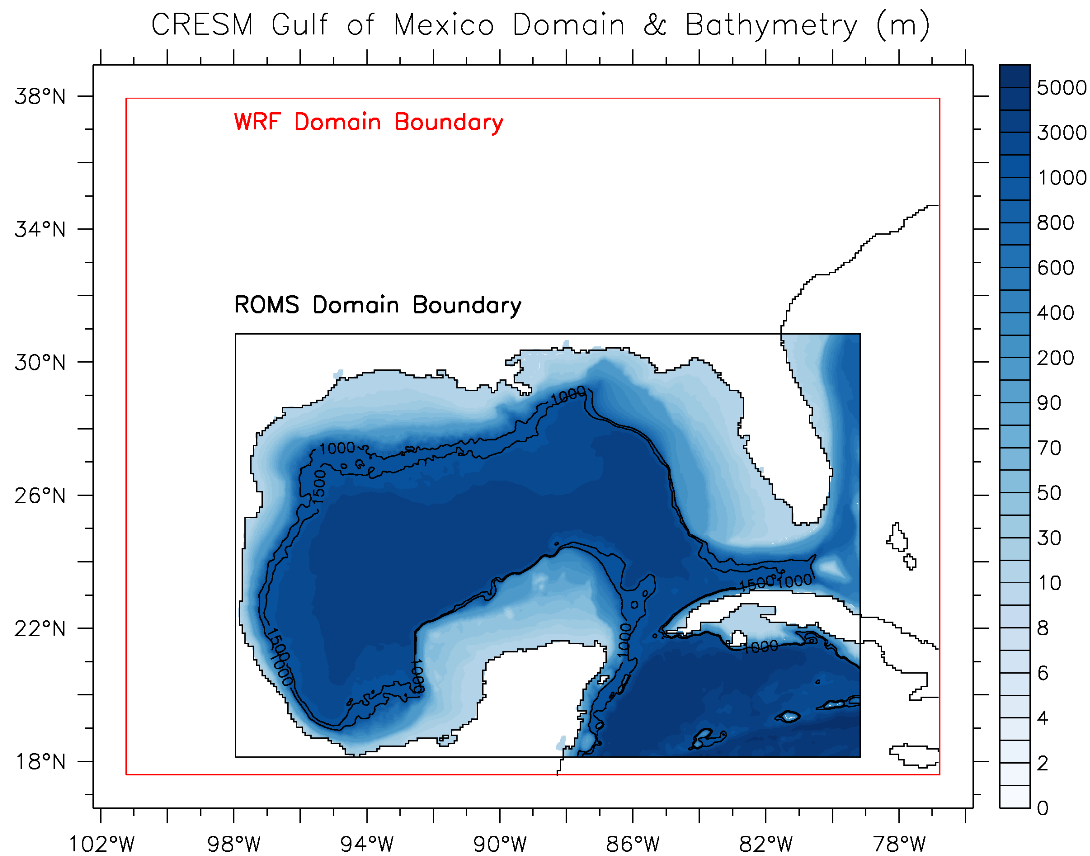

Figure 3.1: An example of typical CRESM configuration. The CRESM domain for the Gulf of Mexico test case is shown here. Red line shows the WRF domain and black line shows the ROMS domain. The blue shading indicates bathymetry for the ROMS grid with colorbar shown to the right. Black contours show the 1000 and 1500 m bathymetry. WRF domain is slightly bigger than the ROMS domain so that the ROMS solution will be free of any boundary effects from WRF. The buffer zone between ROMS and WRF boundaries are handled through SST and ice data specified through either WRF lower boundary input files or through dedicated xROMS files. ROMS and WRF can have different horizontal resolution and pre-compiled interpolation weight files are used to map surface variables between component models.

Component	Model	Status	Description
Atmosphere	WRF 3.5.1	Stable	Adavanced Research Weather Reasearch and Forecasting model (WRF-ARW) (*Skamarock and Klemp*, 2008).
Ocean	ROMS 3.5	Stable	Regional Ocean Modelling System, rev 568 (*Shchepetkin and McWilliams*, 2005).
	Data/Slab	Stable	Data/slab ocean model, as provided with CESM 1.1
	xROMS	Stable	New regional component, featuring ROMS nested in data ocean
Land	CLM 4.0	Stable	Community Land Model, version 4.0 (*Lawrence et al.*, 2011)
	WRF built-in	Stable	WRF-provided land models: Unified Noah, Noah-MP, CLM 4
Sea Ice	CICE 4.0 beta	Experim.	Los Alamos Sea Ice Model, version 4.0 (beta) (*Bailey et al.*, 2010)
Land Ice	stub	Inactive	Unavailable. Code for CISM included but inactive.
Runoff	stub	Inactive	Unavailable. Experimental code for data, MOSART, RTM, and RVIC included but inactive.

Table 3.1: Model components available in CRESM (from *Montuoro* (2017)).

portios of the ocean surface with SST data from WRF's original lower boundary input files (eg. wrflowinp_d01). Please note that the mapping files for this configuration should be created using WRF grid and ROMS grid (see Section 4.3.2.5 for details). Section 2.2.6 describes error from the use of wrong mapping files and methods to fix it. This configuration do not need any data ocean (xROMS) input files (like ∗_xroms_sstice.nc, see Section 3.1.2).

3.1.2 xROMS

To use xROMS (ROMS within a data ocean model), compile CRESM (see Section 1.1.4) with

```
--enable-ocn=xroms
--enable-atm-sst=no
```

Please note that this configuration can use any ocean surface scheme. With this configuration, the ROMS domain is inside of a data model. The data model will have same horizontal resolution as ROMS but extends outward to cover the WRF domain too. In the common region both data model and ROMS will have exactly matching grid points. Please see Section 2.2.3 in *Montuoro* (2017) for more details. The mapping files for this configuration should be created using WRF and xROMS grid (see Section 4.3.2.5 for details). Section 2.2.6 describes error from the use of wrong mapping files and methods to fix it. This configuration require data ocean (xROMS) input files (like ∗_xroms_sstice.nc, see Section 4.3.2.6).

3.2 Machine dependant Options

3.2.1 CRESM with or without pnetCDF

Here is how to run the GOM test case without pnetcdf library.

- Before installing pio, edit build_pio.sh (cresm-1.0.0/tools) to remove "PNETCDF_PATH" environment variable export line and "−−enable-pnetcdf" and "−−enable-filesystem-hints=gpfs" from configure options (configure just with "−−prefix=/path/to/install" option).
- Before compiling, edit "run_config.sh" to remove "−−with-pnetcdf=/path/to/pnetcdf" and compile CRESM.

- Before submitting job, edit namelist.input (&time_control) entries for io_form for history and restart as follows:

```
io_form_history                    = 2
io_form_restart                    = 2
```

- Before submitting job, edit run_cresm.job to remove following three lines:

```
export PNETCDF_HINTS="striping_unit=16777216"
export PNETCDF_PATH=/software/tamusc/pnetcdf/1.6.1-intel-2015B
export PNETCDF=/software/tamusc/pnetcdf/1.6.1-intel-2015B
```

Do the exact opposite for running with pnetCDF (set io_form values for history and restart in namelist.input to 11).

3.3 CRESM Options

This section provides common CRESM related options.

3.3.1 Restart File Frequency

Please note that the restart file writing frequency is not determined by rpointer files (Section 4.3.2.1) but by the value of "restart_n" in drv_in. It seems like the value of "restart_n" in drv_in overrides the restart frequency settings for ROMS in ocean.in (field NRREC) and WRF in namelist.input (field restart_interval_d).

3.3.2 Fine-tuning Land-Sea Mask

See cresm-1.0.0/mapping/check_maps to see how to compile and make ESMF_RegridWeightGen. Then use cresm-1.0.0/tools/esmf_remap.sh to make remapping files. Using these remapping files make a short CRESM model run and look at the outputs (eg. SST from WRF file). Check for anomoulus patterns of values. If there are bad values, then edit the land mask of desired component (WRF or ROMS grid), then recompute the mapping files. Make a test run and repeat this process untill the fields in both WRF and ROMS output files appear without any anomouls patterns near land-sea boundary.

3.4 ROMS Options

This section provides ROMS ocean component related options.

3.4.1 ROMS Configuration

The configuration options for ROMS model are selected using through two files

3.4.2 Turn off ROMS 3D-Nudging

The default method or running ROMS is without 3D-nudging. To adapt the CRESM Gulf of Mexico test case for an experiment without ROMS 3D-nudging, edit roms header file (txglo_3dnudg.h in cresm-1.0.0 directory, see 4.1.1 for details) before compilation and roms input file (ocean.in) before model run as follows.

- roms header file (txglo_3dnudg.h): undef following fields

```
# undef TCLIMATOLOGY
# undef TCLM_NUDGING
```

- roms input file (ocean.in): set CLMNAME to /dev/null

```
CLMNAME ==  /dev/null
```

With these changes, compile the CRESM source code and run the experiment.

Chapter 4

Input/Output Files

All files required during CRESM running phase are called input files and files created by CRESM during a run are called output files. Please note that for models like ROMS configurable options are chosen during both compilation and running phases. Since the source code always contain files required during compilation stage, their list is not presented here. But the important files for the compilation phase are discussed in detail. Since the files needed during model running phase (for a new experiment) are not available as a package like the CRESM source code, their list is presented for each component model. Again, most important among these are discussed in detail. Please use the input files for the Gulf of Mexico test case (Chapter 1) as a reference for setting up a new experiment.

TIP: If the model run involves many restart runs, keep the input files (for the running phase) which are invariant across the runs in a separate directory and make symbolic links in the run directory as required.

4.1 ROMS Input Files

4.1.1 Compilation Phase

For the ROMS model, some of the scheme/parametrization choices are made pre-compilation using header (file with '.h' extension) files. File txgla_3dnudg.h in src/cresm-1.0.0/ directory is the ROMS header file for the Gulf of Mexico test case. A copy of the header file used during compilation will be available in src 'exec/include' directory (Section 1.1.4).

4.1.2 Running Phase

For ROMS, the 3D nudging file is required only if the 3D nudging is turned on (see Section 3.4.2). All other files are required for realistic configurations (ROMS do have the options to provide grid, boundary and initial files through analytical functions for idealized configurations). A list of ROMS input files are given in Table 4.1.

Sl. No.	Filename	Component Model	File Type	File Description
1	∗_roms_grd.nc	ROMS	netCDF	Grid file
2	∗_roms_bry.nc	ROMS	netCDF	Boundary condition file
3	∗_roms_ini/rst.nc	ROMS	netCDF	Initial condition or restart file
4	#∗_roms_nudg.nc	ROMS	netCDF	3D nudging file (temp/salt/u/v)
5	ocean.in	ROMS	txt	Namelist/options file
6	varinfo.dat	ROMS	txt	Input & Output variable details

Table 4.1: List of ROMS input files. Optional file is marked with #.

4.1.2.1 Grid File

ROMS grid file (netCDF) contains all the grid related information needed by ROMS model. In addition model grid cell latitude and logitude, grid file contains several parameters like grid cell area, model bathymetry and land-sea mask.

4.1.2.2 Boundary Condition File

For open lateral boundaries, ROMS require time varying data for temperature, salinity, u and v velocities and sea surface height. This is provided through the boundary condition netCDF file.

4.1.2.3 Inidial Condition or Restart File

For the very first run, ROMS need initial condition file with full 3D model state (temperature, salinity, u nad v velocities and sea surface height). For a restart run, this is provided through model generated restart files.

4.1.2.4 3D Nudging File

ROMS can be configured with 3-dimentional nudging to specified data to make sure model is not drifting too far away from the specified data (typically from ocean reanalysis products). The data can be climatology or inter-annually varying depending on the

Sl. No.	Filename	Component Model	File Type	File Description
1	wrfinput_d01	WRF	netCDF	Grid file
2	wrflowinp_d01	WRF	netCDF	Lower boundary conditions
3	wrfbdy_d01	WRF	netCDF	Lateral boundary conditions
4	#atm.r.2010-01-04.*.nc	WRF	netCDF	Restart file
5	namelist.input	WRF	txt	Namelist/options file

Table 4.2: List of WRF input netCDF and namelist files. Item 4 is required only for restart runs.

application. Nudging can be along few grid points along the open boundaries or throughout the entire ROMS domain (like in the Gulf of Mexico test case). Irrespective of this choice, the 3D nudging data should always be for the entire 3D model domain (a ROMS requirement) and time varying.

4.1.2.5 ocean.in

ROMS model options, including domain decomposition are specified in the input file ocean.in. The total PE in ocean.in, computed as the product of NtileI and NtileJ (eg. 16*24=384) should be same as ocn_tasks in drv_in (&ccsm_pes namelist) (see Section 4.3.2.3).

4.2 WRF Input Files

4.2.1 Compilation Phase

Files during compilation WRF

4.2.2 Running Phase

WRF input netCDF and namelist files are listed in Table 4.2, which are mandatory for all cases.

For WRF there are serveral table files in addition to input netCDF and namelist files. All files in a WRF directory (cresm-1.0.0/models/atm/wrf/3.5.1/WRFV3/run/*) are linked to run directory as table files and no check has been done to see which all of these files are mandatory for running CRESM and which all reduntant. A list of table files are given in Table 4.3.

Sl. No.	Filename	Sl. No.	Filename
1	aerosol.formatted	24	ETAMPNEW_....._rain
2	aerosol_lat.formatted	25	ETAMPNEW_....._rain_DBL
3	aerosol_lon.formatted	26	GENPARM.TBL
4	aerosol_plev.formatted	27	grib2map.tbl
5	CAM_ABS_DATA	28	gribmap.txt
6	CAM_AEROPT_DATA	29	LANDUSE.TBL
7	CAMtr_volume_mixing_ratio.A1B	30	MPTABLE.TBL
8	CAMtr_volume_mixing_ratio.A2	31	ozone.formatted
9	CAMtr_volume_mixing_ratio.RCP4.5	32	ozone_lat.formatted
10	CAMtr_volume_mixing_ratio.RCP6	33	ozone_plev.formatted
11	CAMtr_volume_mixing_ratio.RCP8.5	34	RRTM_DATA
12	CLM_ALB_ICE_DFS_DATA	35	RRTM_DATA_DBL
13	CLM_ALB_ICE_DRC_DATA	36	RRTMG_LW_DATA
14	CLM_ASM_ICE_DFS_DATA	37	RRTMG_LW_DATA_DBL
15	CLM_ASM_ICE_DRC_DATA	38	RRTMG_SW_DATA
16	CLM_DRDSDT0_DATA	39	RRTMG_SW_DATA_DBL
17	CLM_EXT_ICE_DFS_DATA	40	SOILPARM.TBL
18	CLM_EXT_ICE_DRC_DATA	41	tr49t67
19	CLM_KAPPA_DATA	42	tr49t85
20	CLM_TAU_DATA	43	tr67t85
21	co2_trans	44	URBPARM.TBL
22	ETAMPNEW_DATA	45	URBPARM_UZE.TBL
23	ETAMPNEW_DATA_DBL	46	VEGPARM.TBL

Table 4.3: List of WRF table input files.

4.2.2.1 namelist.input

Please adapt the namelist.input file provided with Gulf of Mexico test case for a new application rather than using namelist.input file available with independant WRF distribution. Please note that the domain decomposition (processor tiling) is automatically determined by CRESM and there is no field in namelist.input to control this aspect for the WRF component. Please see Section 4.3.2.3 for details about processor specifying options for CRESM.

4.3 CRESM Input Files

4.3.1 Compilation Phase

Files during compilation CRESM

4.3.2 Running Phase

For CRESM and its coupler, there are several input files which are listed in Table 4.4. Please provide all of these files even if some of the component models (like ice) are not used. All files which do not belong exclusively to either ROMS or WRF is included in this category. The acronym "IO" implies input-output. (Some words are used interchangably: ocean/ROMS, atmosphere/WRF, data ocean/xroms in Table 4.4 and this will be corrected in the future.)

4.3.2.1 rpointer Files

The rpointer here means "restart pointer" which informs CRESM about restart date and time. There are 5 rpointer files, one for each component as shown below:

atmosphere/WRF:	rpointer.atm
data ocean:	rpointer.docn
driver/coupler:	rpointer.drv
ROMS:	rpointer.roms
ocean:	rpointer.ocn (symbolic link to rpointer.roms)

Please note that the time format for these files are different (like 2010-01-04_00_00_00 for atm and 2010-01-04-00000 for drv). The rpointer.drv file use acronym cpl in it (like TXGLO.cpl.r.2010-01-04-00000.nc) instead of drv. Also there are two entries for the rpointer.docn and other have just one entry. The usage details of rpointer files are listed

Sl. No.	Filename	Component Model	File Type	File Description
1	domain.txglo.nc	CRESM	netCDF	Domain file
2	map_a2o_aave.nc	CRESM	netCDF	Mapping weight, atm2ocn, area-average
3	map_a2o_blin.nc	CRESM	netCDF	Mapping weight, atm2ocn, bi-linear
4	map_o2a_aave.nc	CRESM	netCDF	Mapping weight, ocn2atm, area-average
5	seq_maps.rc	CRESM	txt	path to mapping weight files
6	#*_xroms_sstice.nc	CRESM	netCDF	SST, ice & mask for xroms
7	drv_in	CRESM	txt	driver file with date, time, processor information
8	ocn_in	CRESM	txt	ocean model time manager & IO
9	lnd_in	CRESM	txt	path to land model files
10	ice_in	CRESM	txt	path to ice model files
11	docn_in	CRESM	txt	data ocean namelist file
12	docn_ocn_in	CRESM	txt	data ocean namelist file & map scheme
13	docn.streams.txt.prescribed	CRESM	txt	path to CRESM domain and xroms files
14	ocn_modelio.nml	CRESM	txt	ocean model IO settings
15	atm_modelio.nml	CRESM	txt	atmospheric model IO settings
16	cpl_modelio.nml	CRESM	txt	coupler IO settings
17	lnd_modelio.nml	CRESM	txt	land model IO settings
18	ice_modelio.nml	CRESM	txt	ice model IO settings
19	glc_modelio.nml	CRESM	txt	ice-sheet model IO settings
20	rpointer.ocn	CRESM	txt	ocean restart pointer
21	rpointer.roms	CRESM	txt	ROMS restart pointer (same as rpointer.ocn)
22	rpointer.atm	CRESM	txt	atmosphere/WRF restart pointer
23	rpointer.drv	CRESM	txt	coupler/driver restart pointer
24	rpointer.docn	CRESM	txt	data ocean restart pointer
25	#cpl.r.2010-*.nc	CRESM	netCDF	coupler restart file
26	cresm	CRESM	exe	cresm executable
27	run_cresm.job	CRESM	txt	job submission file

Table 4.4: List of CRESM specific input netCDF and namelist files. Item 24 is only required for restart runs.

below.

- For the very first run from initial condition files, set the entries in all rpointer files to expected restart date.
- For all successfull runs, the rpointer files are automatically updated with the most recet restart date and time.
- If you intend to continue a run from most recent restart files, just these updated rpointer files to the restart run directory.
- If you are making restart run from a different restart file, please update the rpointer files accordingly.
- For restart runs, the first entry in rpointer files should correspond to the restart date and time.

```
TXGLO.atm.r.2010-04-20_00_00_00.nc
TXGLO.atm.r.2010-04-30_00_00_00.nc
```

Please note that the restart file writing frequency is not determined by rpointer files but by the value of "restart_n" in drv_in (see Section 3.3.1 for details).

4.3.2.2 docn_ocn_in

The syntax for years in the streams entry of "docn_ocn_in" is as follows:

```
streams = "docn.streams.txt.prescribed YrAlign yrFirst yrLast"
```

It appears that the YrAlign should be same as YrFirst always!!!!!

4.3.2.3 drv_in

The number of processors/cores (PEs) for running CRESM and its component models should be clearly mentioned in drv_in (&ccsm_pes namelist). If drv_in is edited to update PE count or layout, pleae edit the ocean.in (Section 4.1.2.5) and run_cresm.job file (Section 4.4.1) accordingly. Please note that the total number of PEs are devided between atm_ntasks and ocn_ntasks. Also, atm_rootpe is 0 and ocn_rootpe is same as atm_ntasks. All other component model mirrors the settings for the atm. Two examples for total PE counts of 552 and 120 are provided in Table 4.6.

ccsm_pe field	Total PE=552	Total PE=120
atm_ntasks	168	40
atm_nthreads	1	1
atm_rootpe	0	0
atm_pestride	1	1
atm_layout	‘concurrent’	‘concurrent’
lnd_ntasks	168	40
lnd_nthreads	1	1
lnd_rootpe	0	0
lnd_pestride	1	1
lnd_layout	‘concurrent’	‘concurrent’
ocn_ntasks	384	80
ocn_nthreads	1	1
ocn_rootpe	168	40
ocn_pestride	1	1
ocn_layout	‘concurrent’	‘concurrent’
ice_ntasks	168	40
ice_nthreads	1	1
ice_rootpe	0	0
ice_pestride	1	1
ice_layout	‘concurrent’	‘concurrent’
glc_ntasks	168	40
glc_nthreads	1	1
glc_rootpe	0	0
glc_pestride	1	1
glc_layout	‘concurrent’	‘concurrent’
cpl_ntasks	168	40
cpl_nthreads	1	1
cpl_rootpe	0	0
cpl_pestride	1	1

Table 4.6: PE layout in drv_in. Total number of PE is determined by the sum of number of atm model PE (atm_ntasks) and ocn model PE (ocn_ntasks). Please note that the root PE for atm is 0 and that for ocn in atm_ntasks. Other component models mirror atm model settings.

Sl. No.	Filename	Component Model	File Type	File Description
1	cresm	CRESM	exe	cresm executable
2	run_cresm.job	CRESM	txt	job submission file

Table 4.7: List ofother machine dependent input files, including CRESM executable.

4.3.2.4 docn.streams.txt.prescribed

This file provides path and filenames for the domain info file (eg. domain.txglo.nc) and the xROMS sea surface temperature (SST) and ice fields (eg. ∗_xroms_sstice.nc). Please update the value for "<filePath>" and "<fileNames>" for both "<domainInfo>" and "<fieldInfo>" entries as appropriate.

4.3.2.5 Mapping Weight Files

Coupled model components can have different resolutions. CRESM requires precomputed interpolation weights to map surface quantities between different coupled model components. Interpolation options like bilinear and averaging options like area-average are available with the ESMF tool.

For a detailed discussion on mapping weight files and how to make them, please see Section 5.2 in *Montuoro* (2017).

4.3.2.6 xroms_sstice.nc

CRESM need data for SST and ice over entire domain. With xROMS set up (Section 3.1.2), user has to provide an xROMS file with SST and ice for the entire xROMS domain. SST for the bigger domain is typycally available in WRF lower boundary input files. Current test cases use ice as 0 everywhere.

A simple approach is to use matlab to interpolate WRF SST onto xROMS grid and then write the interpolated SST to a proper xROMS SST netCDF file (use the file from Gulf of Mexico test case as a reference). A matlab

4.4 Other Input Files

Other files required by CRESM during running phase are listed in Table 4.7.

4.4.1 run_cresm.job

This is the file used to submit a CRESM job to the job scheduler on the supercomputer. The total PEs requested should be in agreement with the total PEs in drv_in (Section 4.3.2.3) computed as the sum of atm_ntasks and ocn_ntasks.

4.5 All Output Files

Complete list of output files from a CRESM run is provided in Table 4.8.

Sl. No.	Filename	Component Model	File Type	File Description
1	atm.log	CRESM	txt	atm model log file
2	cpl.log	CRESM	txt	coupler log file, run statistics
3	ocn.log	ROMS	txt	ocean log file
4	roms.ocn.log	ROMS	txt	ocean log file (same as ocn.log)
5	cresm.log	CRESM	txt	CRESM log file
6	data.ocn.log	CRESM	txt	data ocean log file
7	main.ocn.log	CRESM	txt	mct log
8	namelist.output	WRF	txt	WRF options summary
9	rsl.error.????	WRF	txt	WRF std. error
10	rsl.out.????	WRF	txt	WRF std. out
11	∗.ocn.hi.∗.nc	ROMS	netCDF	ROMS/ocn history files
12	∗.ocn.r.∗.nc	ROMS	netCDF	ROMS/ocn restart files
13	∗.atm.hi.∗.nc	WRF	netCDF	WRF/atm history files
14	∗.atm.r.∗.nc	WRF	netCDF	WRF/atm restart files
15	∗.cpl.r.∗.nc	CRESM	netCDF	coupler restart files
16	∗.docn.rs1.∗.bin	CRESM	binary	data ocean restart files

Table 4.8: List of all output files from a CRESM run.

Chapter 5

Tools

Tools available with the cresm_1.0.0 version source code is described here.

Within the CRESM source code, cresm-1.0.0/tools/misc have tools made by Jaison and cresm-1.0.0/tools and cresm-1.0.0/scripts have tools made by *Montuoro* (2017). Please take a look at the README∗ files in those directoris for a list and description of available tools.

5.1 Requirements

Some of the tools made by (*Montuoro*, 2017) are in python and require netCDF library. Do the following if you do not have a working python on TAMU ada. Assuminig the existence of $SCRATCH/SOFTWARE/Python/MYpython directory

```
[user@comp]$ module purge
[user@comp]$ module load Python/2.7.12-intel-2017A
[user@comp]$ virtualenv \
             $SCRATCH/SOFTWARE/Python/MYpython/2.7.12-intel-2017A
[user@comp]$ source  $SCRATCH/SOFTWARE\
        /Python/MYpython/2.7.12-intel-2017A/bin/activate
[user@comp]$ pip install netCDF4
```

Please see `https://hprc.tamu.edu/wiki/SW:Python` for more information about python on TAMU Ada.

Within the CRESM source code, cresm-1.0.0/tools/misc have tools made by Jaison and cresm-1.0.0/tools and cresm-1.0.0/scripts have tools made by *Montuoro* (2017). Please take a look at the README∗ files in those directoris for a list and description of available tools.

5.2 Tools for src

5.2.1 Compare src directories

tools/misc/compare_src.csh: Compare two versions of CRESM source code in two directories and printout a list of differing files/subdirectories.

5.3 Tools for Input Files

5.3.1 CRESM Files

tools/misc/func_create_cresm_file.m : Matlab function to create CRESM files. Current capabilities are:

- xROMS SST/ice file (fltype 'xroms')

5.3.2 Data Ocean Domain File

According to Raffaele's documentation, the data ocean domain file is created using wrf2dom.py. However, there is the following problem with respect to the files obtained from Raffaele for GOM:

- WRF is bigger domain, but coarse resolution
- ROMS is smaller domain, but fine resolution
- xROMS data ocean domain is same as WRF with ROMS resolution
- which script to use to make domain.nc file?
  - Raffaele's domain.nc file claims to have used wrf2dom.py
  - using the same script with expected files yield a coarse resolution domain instead of fine resolution one!!!!

A solution to this problem will be creating another ROMS grid identical to the data ocean grid and using fields from this grid file along with wrf2dom.py to make the data ocean domain file.

5.4 Tools for model run

5.4.1 Copy/Create Run Directory

tools/misc/mkcopy.csh: Tool to make copy of input files in a run directory to a new (eg. restart) directory. Symbolic links will be copied as symbolic links. This tool needs an input file 'wrf_inputs.txt' (available in tools/misc/ directory).

tools/create_newcase: (Section.6, *Montuoro* (2017)) can also be used for creating a template run directory base on how the model has been configured.

5.4.2 Check for completion of CRESM run

You can confirm a CRESM run/job is completed if you see lines as shown below at the end of respective log/out files:

```
ocn.log or roms.ocn.log
   ROMS/TOMS: DONE... Wednesday - August 9, 2017 - 11:58:00 AM

cpl.log
   (seq_mct_drv): ==== pes max memory last usage (MB) -0.001 =====

rsl.out.0000
   d01 2010-01-04_00:00:00 wrf: SUCCESS COMPLETE WRF
```

CRESM may not exit gracefully from the queue (see Section 2.2.5) and in those cases the job can be cancelled after above check.

5.4.3 Archive CRESM output files

mvoutput.csh: Move CRESM outputs from run directory to an archiving directory.

5.4.4 Clean Up Run Directory

tools/misc/mkclean.csh: To cleanp run directory (eg. before resubmitting a run after a blow up). Be careful with removing output NetCDF files.

tools/misc/wrftable_clean.csh : To delete WRF table files (typically symbolic links from WRF src dir). This tool needs an input file 'wrf_inputs.txt' (available in tools/misc/ directory).

Bibliography

Bailey, D., M. Holland, E. Hunke, B. Lipscomb, B. Briegleb, C. Bitz, and J. Schramm (2010), *Community Ice CodE (CICE) User's Guide Version 4.0*, Available from http://www.cesm.ucar.edu/models/ccsm4.0/cice/ice_usrdoc.pdf.

CESM1.1 (), Cesm 1.1 series public release, http://www.cesm.ucar.edu, Accessed 2016/12.

Craig, A. P., M. vertenstein, and R. Jacob (2012), A new flexible coupler for earth system modelling developed for CCSM4 and CESM1, *Int. J. High Perform. C.*, *26*, 31–42.

Lawrence, D. M., K. W. Oleson, M. G. Flanner, P. E. Thornton, S. C. Swenson, P. J. Lawrence, X. Zeng, Z. L. Yang, S. Levis, K. Sakaguchi, G. B. Bonam, and A. G. Slater (2011), Parameterization improvements and functional and structural advances in version 4 of the Community Land Model, *J. Adv. Model. Earth Sys.*, *3*, doi:10.1029/2011MS000045.

Montuoro, R. (2017), The Community Regional Earth System Model (CRESM), Version 1.0.0, *Tech. rep.*, Texas A&M University.

Shchepetkin, A., and J. C. McWilliams (2005), The Regional Oceanic Modeling System: A split-explicit, free-surface, topography-following-coordinate ocean model, *Ocean Modell.*, *9*(4), 347–404, doi:10.1016/j.ocemod.2004.08.002.

Skamarock, W. C., and J. B. Klemp (2008), A time-split nonhydrostatic atmospheric model for weather research and forecasting applications, *J. Comput. Phys.*, *227*, 3465–3485.