RegESM (Regional Earth System Model) 1.0 User Guide

Ufuk Utku Turuncoglu
Istanbul Technical University
Informatics Institute
Computational Science and Engineering Dept.
Turkey

E-mail / Web: ufuk.turuncoglu@itu.edu.tr / http://faculty.itu.edu.tr/turuncogl1/

1. Model Design

Regional Earth System Model (RegESM) is designed to be a state-of-art coupled modeling system that allows using variety of different earth system models. It also supports easy to plug new sub-components by using its simplified interface. In this case, developers from different disciplines might easily adapt their sub components (i.e. wave, ice, land surface) to the modeling system by following the common conventions, which is used in RegESM.

The designed modeling system currently includes three different model components, which are connected via ESMF[1] (Earth System Modeling Framework):

- **Atmosphere:** ICTP's RegCM[2] (4.5) regional climate model
- **Ocean:** In this case modeling system supports two different ocean models. Rutgers University's ROMS[3] (r809) ocean model (s-coordinate model) or MITgcm[4] (c63s; z-coordinate model)
- **River Routing:** Max-Planck Institute's HD[5] (1.0.2 with modifications) river routing model.
- **Wave:** ECMWF's WAM (Cycle_4.5.3_MPI with modifications)

The main aim is to use such kind of coupling library (like ESMF) is to standardize the coupling interfaces and having efficient interaction among the model components. The key component of the coupled model is the "driver" (or "coupler"). It is basically responsible to synchronize the model components and the interaction (via exchange fields) among them. In general, the transferred exchange fields depend on the interaction between the components and the designed application itself. In this design, the configuration file to allow wide variety of applications easily modifies the exchange fields. The information about the definition of the exchange fields between the modeling components can be seen in the Section 3.1.

The design of the RegESM follows the common conventions about the multi-component earth system model. It is designed as an orchestrator to control the plugged sub-components. The RegESM itself is also called as "driver" or "coupler" and it basically has no code related with the physical sub-models. It just holds the definition of the "Initialize", "Run" and "Finalize" routines, component grid structure information (masking, grid coordinates and decomposition properties) and time information to achieve synchronization among the components.

The modeling system uses Earth System Modeling Framework (ESMF) as a coupler library to connect different variety of standalone earth system models. In this case, each component is assigned as a gridded component and the interaction between them is defined using connectors - The National Unified Operational Prediction Capability (NUOPC[6]) - interface. In the future, we are planning to use mediator instead of connectors to have much more flexible and efficient design.

The interaction of the main components can be seen in Fig. 1.

---

1 http://www.earthsystemmodeling.org/
2 http://gforge.ictp.it/gf/project/regcm/
3 http://www.myroms.org/
4 http://mitgcm.org/download/
5 http://http//www.mpimet.mpg.de/en/science/the-land-in-the-earth-system/terrestrial-hydrology/hd-model.html
6 http://www.earthsystemmodeling.org/conventions/nuopc.shtml

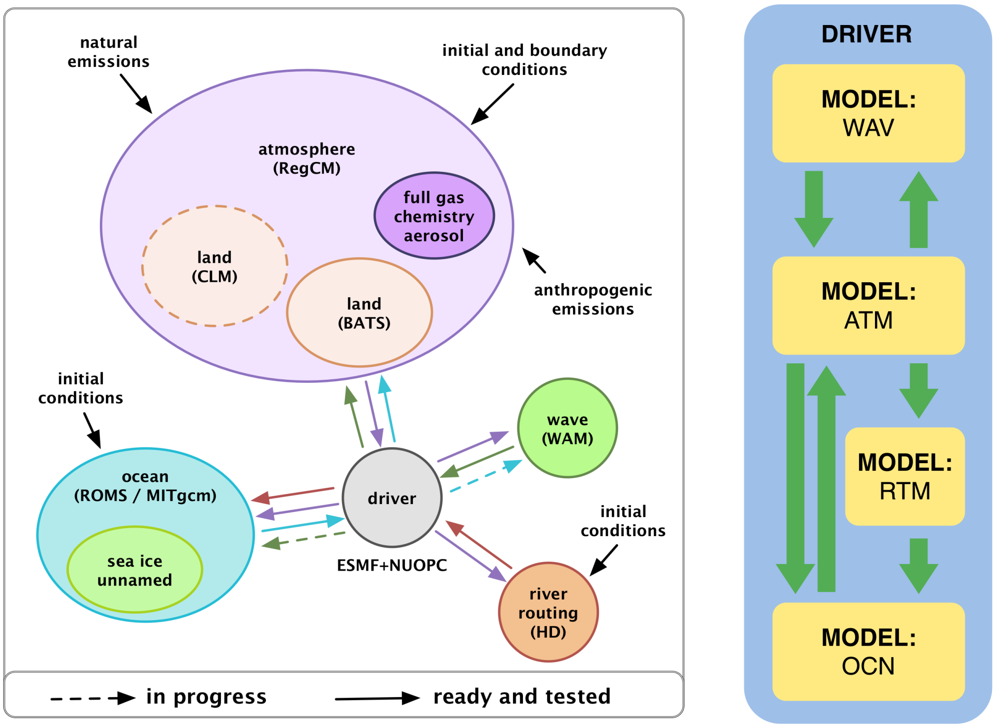


Figure 1 Components of RegESM modeling system. The arrows indicate the interaction direction between the sub-components.

For instance, the following figure (Fig. 2) shows the interaction between four-component earth system models (atmosphere-ocean-river-wave).

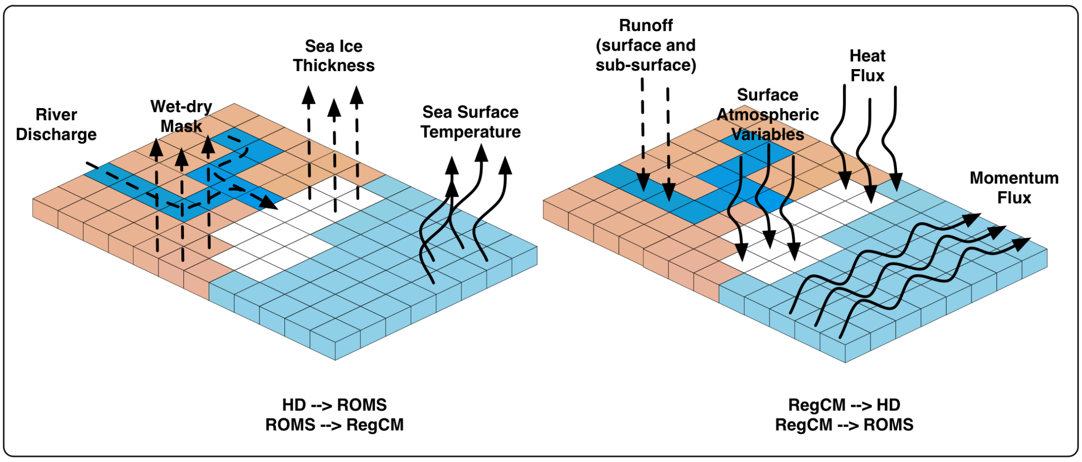


Figure 2 Example of interaction between components. Arrows shows the exchange fields.

The current design allows us to define different coupling intervals among the sub-components (i.e. fast and slow time steps). This is crucial because the response time of the components and the physical processes might be in different time scales. So, it is better to have a flexible modeling system to define different coupling interval among the model components. In Fig. 3, the river routing component (RTM) is interacting with atmosphere (ATM) and ocean (OCN) components with 1-day interval but OCN, ATM and wave (WAV) components exchange data (i.e. sea surface temperature, heat and momentum fluxes) in 3-hour interval. In this case, the coupler component (RegESM) is responsible from the synchronization. It is also note that the current implementation of the coupled model does not allow to setup different coupling time step between atmosphere-ocean and ocean-atmosphere coupling directions. The wave model must be in the same time step with the atmosphere and ocean.

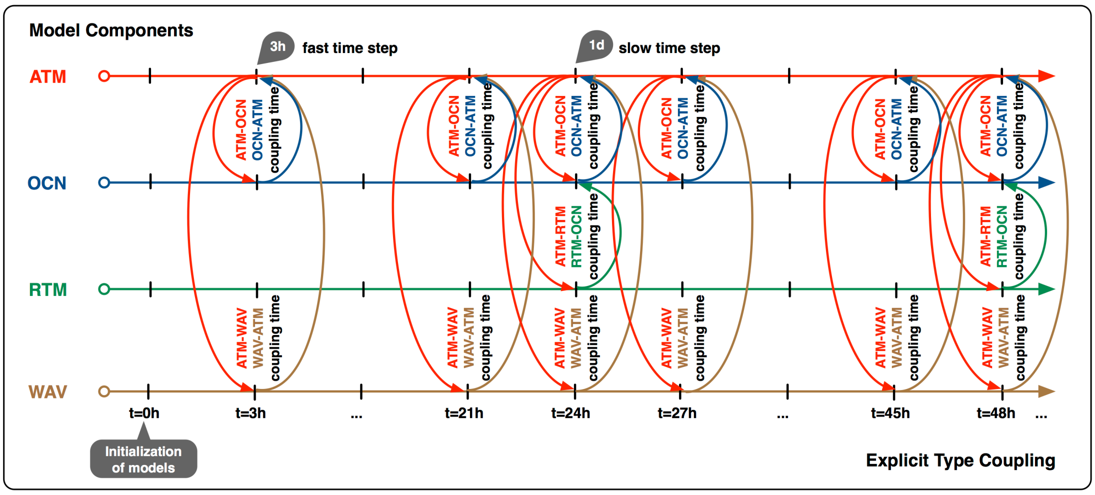


Figure 3 The run sequences for four components (atmosphere-ocean-river-wave).

The components of the modeling system can be activated or deactivated easily via driver's configuration file (namelist.rc, more information can be found in Appendix A). The driver configuration file is also responsible for distributing CPU (or computing resource) to the model components. In this case, model components are able to run in different number of processors (or cores). Additionally, the current design of the modeling system supports both **sequential** and **concurrent** execution of the model components (Fig. 4).

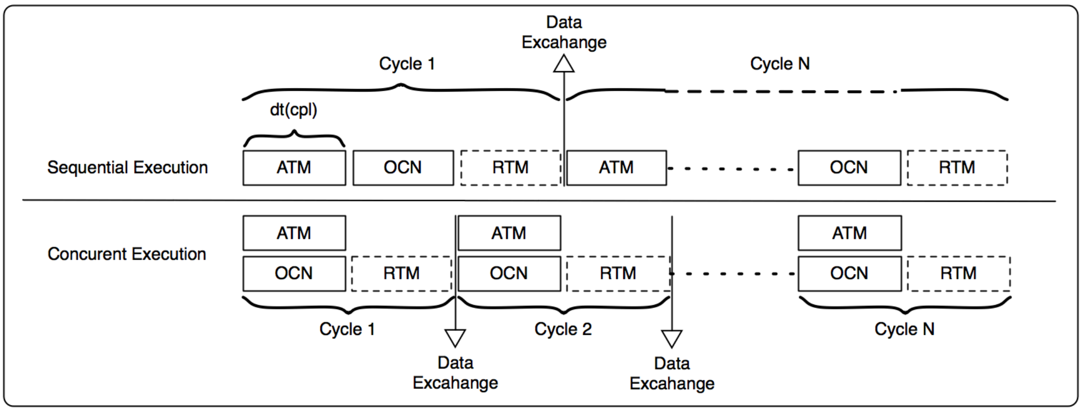


Figure 4 Sequential vs. concurrent execution mechanism for three components case (atmosphere-ocean-river).

In sequential execution mode, all model components and also driver use all the available cores except RTM component. RTM component shares last core with the OCN component (unless it is defined to have additional core in driver namelist file) because it is not parallelized using MPI and it uses only one core. In this mode, models are executed in an order and each one of the model components waits others to run again. After running all model components (just only for single coupling time step), the models exchange the data and start to run again in an ordered fashion until next data exchange time. This can be seen in the upper part of the Fig. 4. In this case, using RTM component is optional and it is shown as dashed lines.

The coupled model also supports concurrent execution mode. In this case, each model component uses its own set of processor (RTM component can be configured to have its own resource or not, see the details of namelist.rc in Appendix A) and driver. The driver uses all available cores between the components to exchange data and perform interpolation. The coupler can be configured to assign different number of core for each model component using its configuration file.

2. Performance

To analyze the performance of the coupled model (RegESM), it is necessary to consider the performance of the individual model components (i.e. RegCM, ROMS) at the beginning. Then the results of the standalone model components can be used to investigate the overall performance of the coupled modeling system and overhead of the driver (or coupler).

As it can be seen from the Fig. 5, we consider two different atmospheric model domains (ATM and ATM-Extended) to see the effect of the domain size in the results. The benchmark results of the atmospheric component can be seen in Fig. 6 (for ATM domain) and Fig. 7 (for ATM-Extended domain). The results show that the model scales pretty well if the chunks of two-dimensional decomposition is big enough to feed the processors. It is clear that model scales better for ATM-Extended domain than original configuration of the atmospheric component (ATM) especially for higher number of cores. The results also show that the tile parameters for each direction (x and y) do not affect the performance results of the atmospheric model. It is also observed that the fat nodes scale better than the thin nodes when the number of processor is increased (tested only for ATM domain).

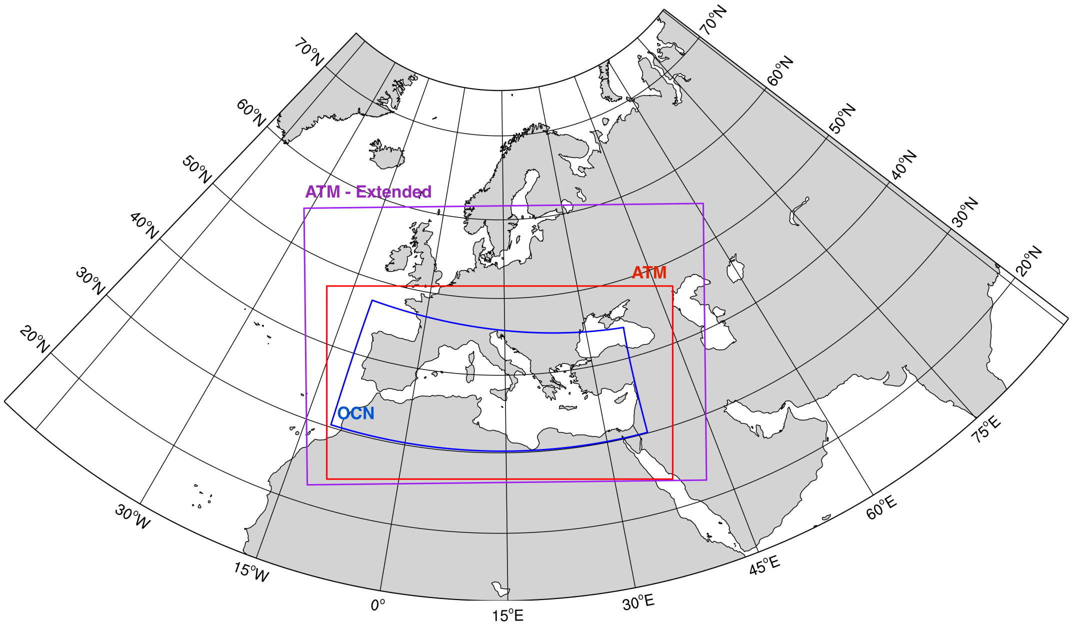


Figure 5 Model domains (ATM, ATM Extended and OCN) used in this study. The resolution of the model components: ATM is 12 km (232x416, L23), ATM Extended is 12 km (332x480, L23) and OCN is 1/12° (~8 km; 264x570, L32).

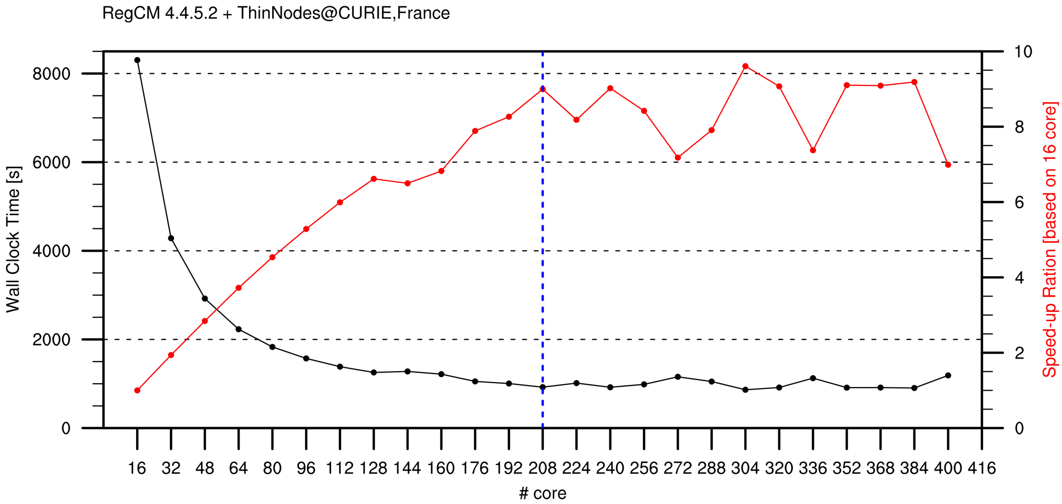


Figure 6 Benchmark results (wall-clock time in right and speed-up based on 16 cores in left) of standalone atmosphere model (RegCM). The 12 km **ATM** domain is used in this case. The blue dashed line shows the higher number of core that gives the best performance. The simulation length is 5 days and model time step is 30 s.

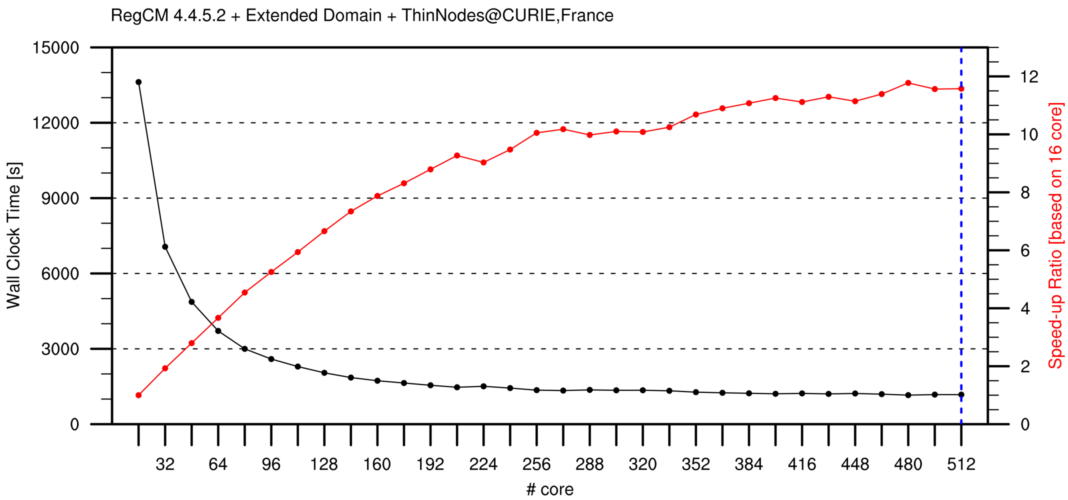


Figure 7 Benchmark results (wall-clock time is in the right and speed-up based on 16 cores is in the left) of standalone atmosphere model (RegCM). The 12 km ATM Extended domain is used in this case. The blue dashed line shows the higher number of core that gives the best performance. The simulation length is 5 days and model time step is 30 s.

Similar to atmospheric component, the ocean model is also tested to find the best tile (x and y direction) configuration (for two-dimensional decomposition) and to check the overall performance and the scalability of the model component. As it can be seen from the Fig. 8, the selection of the tile configuration affects the overall performance of the ocean model. In general, model scales better if tile in x direction is bigger than tile in y direction but this is more evident in the small number of processor. This might due to the memory management of Fortran (column-major order). On the other hand, the tile options must be selected carefully while considering the dimension of the model domain in each direction. In some tile configuration, it is not possible to run the model due to the used algorithm and the required minimum ghost points for underlying finite difference scheme. To summarize, the ocean model scales well until 400 cores but after this level the model does not scale anymore but with higher resolution (both in horizontal and vertical) ocean component probably gives better scaling results.

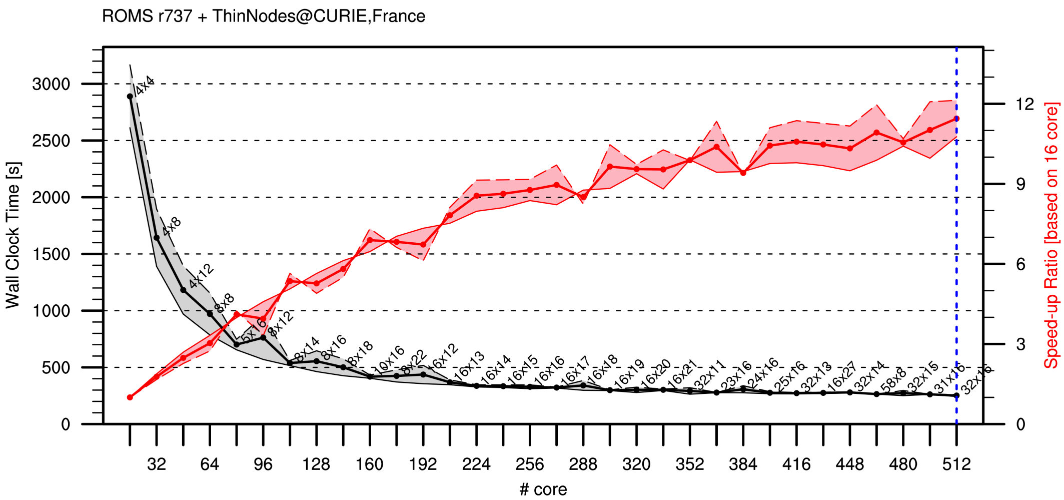


Figure 8 Benchmark results (wall-clock time is in the right and speed-up based on 16 cores is in the left) of standalone ocean model (ROMS). The 1/12° OCN domain is used in this case. The grey and red areas show the performance of the ocean model with different 2d decomposition configuration (tile in x and y directions). The simulation length is 5 days and model time step is 120 s.

Using the performance analysis of standalone atmosphere and ocean models, the performance of the fully coupled earth system model is tested. In this case, the best tile options for the ocean component is used in the coupled model simulations for the particular number of processor assigned to the ocean component. The tests are also performed with different coupling intervals (30-min, 1-hour and 3-hour), different types of coupling scheme (sequential vs. concurrent) and different number of components (two-component - ATM-OCN and three-component - ATM-OCN-RTM).

The coupling interval might affect the overall performance of the coupled modeling system because the driver performs interpolation (two level – interpolation and extrapolation to fill the unaligned land-sea masks) between the model components in each coupling interval due to the differences in the horizontal grid resolutions. The ESMF library basically calculates and stores the weight matrix for each different combination of the exchange at the beginning of the simulation and use it when the model needs data exchange. In this case, ESMF basically performs sparse matrix multiplication (in parallel) to transfer the data from one grid to another. The Fig. 9 shows the measurements of three different cases (30-min, 1-hour and 3-hour) of two components (ATM and OCN) model configuration run in concurrent mode. In concurrent mode, each model component has its own MPI communicator and resources. In this case, all the components run concurrently in a given time until they reach to the data exchange point. It is clear that every component must have same speed to get the best performance from the modeling system. Otherwise, components might need to wait until other components ready to exchange the information. The standalone test shows that the atmospheric model needs more resource than the ocean component in this specific configuration. The initial estimation shows that atmospheric model is 4.5-5.0 times slower than the ocean model for ATM-Expanded case. For ATM domain case, this difference is smaller and reduces to 2.6-3.7. To get the best performance form the coupled model these ratios must be considered when the resources are assigned to the individual model components. As it can be seen from the figure, the models with different coupling interval show very similar scaling behavior.

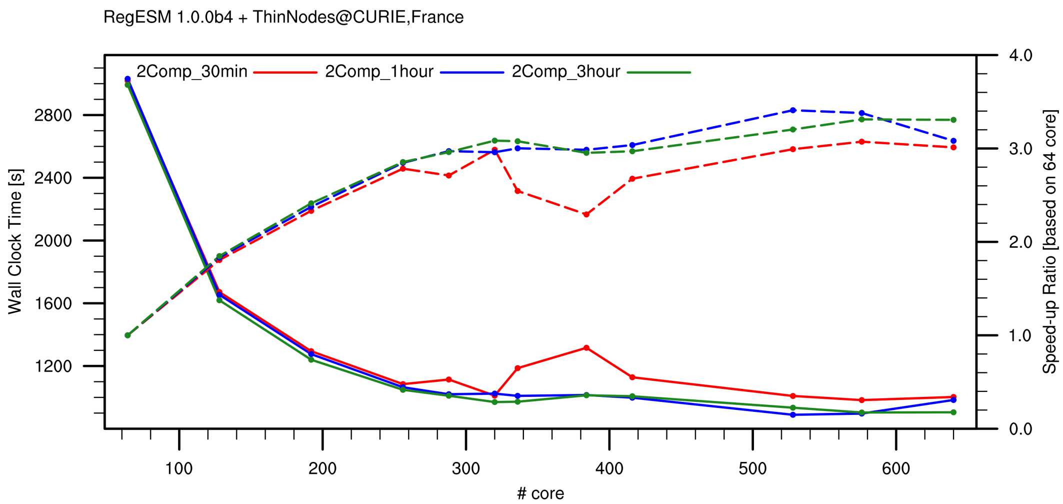


Figure 9 Effect of coupling interval (30 minutes, 1 hours and 3 hours) in the performance of the coupled model (RegESM). The speed-up is calculated based on 64 cores. In this case, only ATM and OCN components are used and the RTM component is not activated. The grey and red areas show the performance of the ocean model with different 2d decomposition configuration. The simulation length is 5 days.

The case that has 30 minutes coupling interval shows small fluctuations around 300-400 cores but this might be related with the overload in the cluster in that particular time and also related with the unbalanced model component performances. In overall, the effect of the coupling time step is limited and fluctuates around %7.5-10 when the results are compared with the fastest run (see Table 1). The results also show that the overhead of the coupling time step is smaller in the lower number of cores (%1.2-5).

The performance analysis of the two component (ocean and atmosphere) coupled model can be seen in Fig. 10. It is shown that the current configuration (OCN+ATM Extended domains) of the coupled model scales up to 512 cores in thin nodes without any particular performance problem. The comparison of the results with the results of the standalone model components and the results collected with Allinea's MAP tool show that the coupling overhead is %3-5 of the total execution time. It is also reveal that coupling overhead reduces with increased number of processor. The better scaling results might be achieved by increasing horizontal (from 1/12° to 1/16°) and vertical resolution (from 32 sigma layers to 48 or 64) of the ocean component of the coupled modeling system but it needs more work (creating horizontal and vertical grids, finding tuned configuration parameters for vertical grid structure etc.) than atmospheric component and it will be done in the future.

Table 1 The estimation for overhead of coupling time step. The overhead is calculated as ((Tmax-Tmin)/Tmin)*100. The calculated average overhead is %10.58. It is %7.5 if two highest data (336 and 384) is not considered in the calculation.

Total #core	ATM #core	OCN #core	Overhead %
64	48	16 (4x4)	1.29
128	96	32 (4x8)	3.23
192	144	48 (4x12)	4.38
256	192	64 (8x8)	3.47
288	208	80 (5x16)	10.23
320	240	80 (5x16)	5.55
336	240	96 (8x12)	22.01
384	288	96 (8x12)	29.89
416	304	112 (8x14)	13.04
528	384	144 (8x18)	13.50
576	384	192 (8x24)	9.57
640	384	256 (16x16)	10.77

In concurrent coupling case (not shown in here), all the model components active in the same time and the models exchange data in every coupling time step. In this case, the model performance is strongly affected by the distribution of the total compute resource into individual model components. The analysis of the measurement of standalone model components shows that atmosphere model (ATM Extended domain) is 4.5-5 times slower than the ocean component. This ratio is calculated 2.6-3.7 for ATM domain case. To gain the maximum performance from the coupled modeling system, it is necessary to assign more processor or compute resource to the atmospheric component. By this way, the better synchronization of the model components is achieved and the idle time of the used compute resource is decreased. The tests show that it is hard to find the best combination of the individual core (or processor) count to assign individual model components. In this case, sequential type of execution might be considered as a good alternative for the simulations.

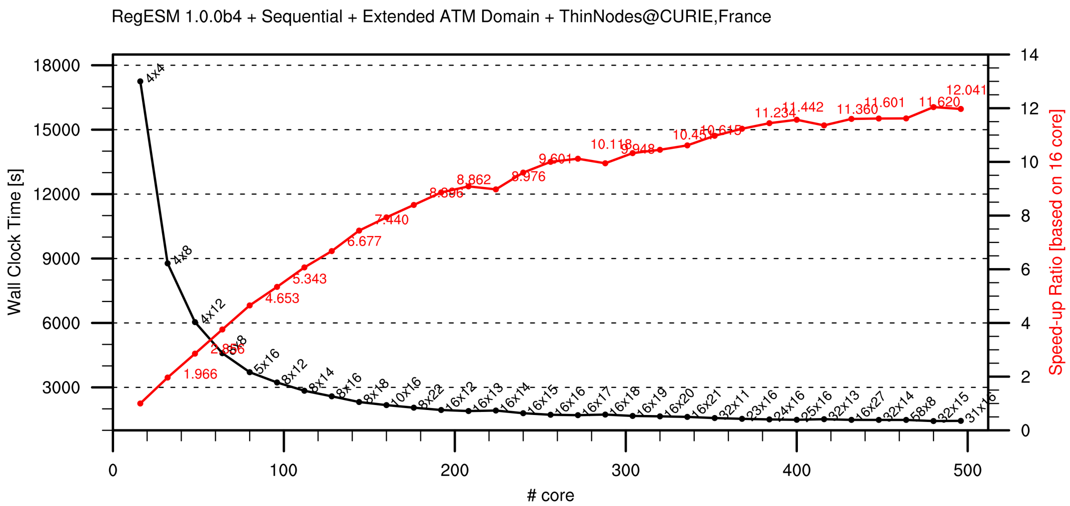


Figure 10 Benchmark results (wall-clock time is in the right and speed-up based on 16 cores is in the left) of fully coupled model (RegESM). The 12 km **ATM Extended** domain and 1/12° OCN domain (Fig. 5) are used in this case. The black numbers show the 2d decomposition configuration (tile in x and y directions) of the ocean model. The coupling time step 1 hour and only two components are active. The simulation length is 5 days.

In addition to measure the performance of the two component coupled model (atmosphere and ocean), the three component modeling system is also tested. In this case, the coupled model also includes river routing component (RTM), which is a sequential code. As it can be seen from the Fig. 11, the third component affects the overall performance of the coupled modeling system. In this case, the scaling results are worse than the two-component case. The main reason of the scaling problem is to use sequential model component and sequential type execution. In this case, last available processor (i.e. PET) is also assigned for the river routing component, which is shared with the ocean model. The performance could be improved when the optimal configuration of the processor distribution of the model components.

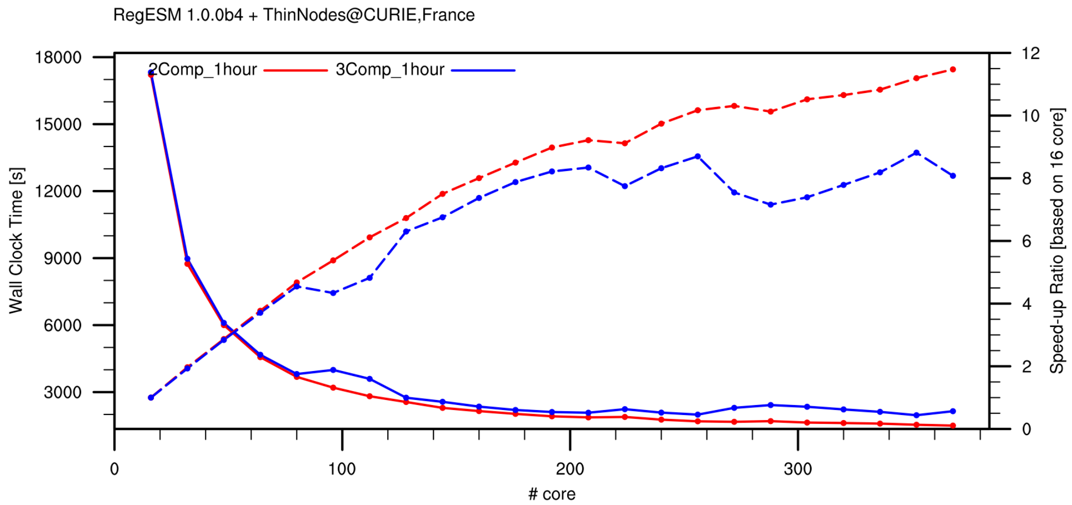


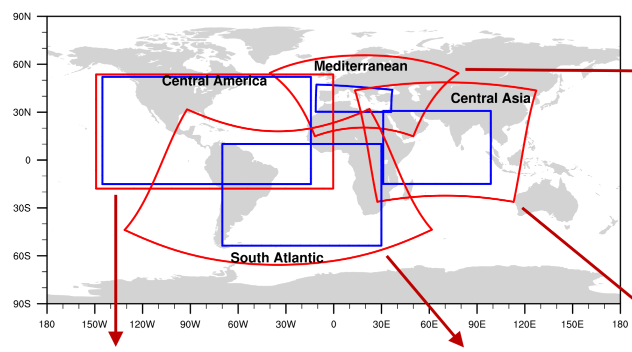
Figure 11 Benchmark results (wall-clock time is in the right and speed-up based on 16 cores is in the left) of fully coupled model (RegESM). The 12 km **ATM Extended** domain and 1/12° OCN domain (see Fig. 5) are used in this case. The coupling time step is defined as 1 hour for atmosphere-ocean and 1 day for atmosphere-river and river-ocean. The simulation length is 5 days.

3. Example Configurations

The current version of the coupled model is using for the different applications and domains that are distributed all over the world (Table 2).

Table 2 Matrix for tested model configurations.

Domain	Contact Person	Components	Detail
Mediterranean Sea	Ufuk Utku Turuncoglu (ITU), ROMS Gianmaria Sanino (ENEA), MITgcm Laura Mariotti (ICTP), MITgcm	RegCM+ROMS RegCM+MITgcm+HD RegCM+WAM	Closed Atlantic Ocean, no active river component Turuncoglu and Saninno, 2016 (CD, under review)
Caspian Sea	Ufuk Utku Turuncoglu (ITU)	RegCM+ROMS+HD	Turuncoglu et al., 2013 (GMD)
Black Sea	Ufuk Utku Turuncoglu (ITU)	RegCM+ROMS	-
Indian Ocean	Fabio di Sante (ICTP)	RegCM+MITgcm+HD	
Central America (Caribbean)	Ramon Fuentes-Franco and Lina Sitz (ICTP)	RegCM+MITgcm+HD	
South Atlantic Ocean	Lina Sitz and Marco Reale (ICTP)	RegCM+MITgcm+HD	



Mediterranean

	ATM	OCN	RTM
Models	RegCM 4.4.5.7	MITgcm c64s	HD
Res.	20 km, 23L 353x253	1/12º, 75L 276x408	0.5º global
ICBC	ERA-Int ERSST	12 km ALADIN	online ATM
Details	BATS Grell+MIT	GGL90	17 major rivers

- Coupled with RegESM driver
- 3 hours coupling interval between ATM and OCN and 1 day in interaction with RTM
- 1979-2013 (35 years)

Central America

	ATM	OCN	RTM
Models	RegCM 4.4.5.9	MITgcm c63s	HD
Res.	50 km, 23L 308x170	1/8º, 40L 1050x540	0.5º global
ICBC	ERA-Int ERSST	0.25º MOM	online ATM
Details	CLM, Grell, KF	KPP	10 major rivers

- Coupled with RegESM driver
- 3 hours coupling interval between ATM and OCN and 1 day in interaction with RTM
- ?

South Atlantic

	ATM	OCN	RTM
Models	RegCM 4.4.5.9	MITgcm c63s	HD
Res.	50 km, 23L 330x206	1/8º, 40L 800x510	0.5º global
ICBC	ERA-Int ERSST	0.25º MOM	online ATM
Details	CLM, UW-PBL, MIT	KPP	7 major rivers

- Coupled with RegESM driver
- 3 hours coupling interval between ATM and OCN and 1 day in interaction with RTM
- ?

Central Asia

	ATM	OCN	RTM
Models	RegCM 4.4.5.9	MITgcm c63s	HD
Res.	50 km, 18L 170x216	1/6º, 45L 276x408	0.5º global
ICBC	ERA-Int ERSST	0.25º MOM	online ATM
Details	CLM, UW-PBL, MIT	KPP	30 major rivers

- Coupled with RegESM driver
- 3 hours coupling interval between ATM and OCN and 1 day in interaction with RTM
- 1979-2007 (25 years)

Figure 12 Different configurations of RegESM modeling system and their configurations.

The list is updated with regular basis and if you want to add your name along with the details of the configuration, please send a mail to here.

4. Installation

The basic requirements of the installation processes of the RegESM can be divided into three main sections:

- In the first step, user needs to prepare required working environment for the RegESM installation. In this case, a set of third-party software libraries (i.e. netCDF, ESMF etc.) should be installed or available in the host system that will be used to run the coupled model
- Then, each model components (i.e. RegCM, ROMS or MITgcm, HD and WAM) must be installed with coupling support. Due to the various design of the standalone model components, the coupling support are achieved by different ways.
    - The latest version of RegCM (version 4.5) can be used as a model component in RegESM without doing any extra modification or applying patch (just compile with --enable-cpl option) but it does not have the coupling with the wave model. If there is a plan to use wave coupling along with the RegCM, please contact with the developers.
    - For ROMS (revision 809) ocean model, user needs to apply a lightweight patch to use the model as a RegESM component. The patch basically used to overwrite the forcing data read from the input files.
    - Unlike, RegCM and ROMS, MITgcm has an additional module that allows coupling. The module is still testing and if there is a plan to use MITgcm as ocean component, please contact with the developers.
    - The HD and WAM models have more modification and they are distributed separately in case of any user request.
- The last step is the installation of RegESM by help of external libraries and the installed model components.

In this section, it is assumed that the user has basic information about usage and installation of the individual model components. So, this document does not include detailed information about the installation of the libraries and the configuration of the individual model components, which is mainly depend on the application and used model configuration. The following sections mainly aim to give detailed information about the installation procedure about the given steps.

4.1. Installation of Third-Party Libraries

This section includes a set of example commands to install required libraries and tools on a Linux (ICTP's Argo Cluster, Centos 6.4 Final) based system by using Intel Compiler (13.1.0) and OpenMPI (installed with same version of Intel Compiler). The users also note that the installation procedure might change in other systems and compiler combinations.

<u>The **PROG** environment variable, which is used in this section, is mainly indicates the directory for tools and library installations.</u>

4.1.1. Hierarchical Data Format (HDF5)

Before installation of HDF5 library (1.8.11), it is necessary to install a compression library. The HDF5 supports both zlib[7] and szip[8] libraries but in this document we choose to install zlib (1.2.8) instead of szip library.

```
cd $PROGS
wget http://zlib.net/zlib-1.2.8.tar.gz
tar -zxvf zlib-1.2.8.tar.gz
cd zlib-1.2.8
export CC=icc
export FC=ifort
./configure --prefix=`pwd`
make
```

[7] http://www.zlib.net
[8] http://www.hdfgroup.org/doc_resource/SZIP/

```
make install
```

Installation of HDF5 library:

```
cd $PROGS
wget http://www.hdfgroup.org/ftp/HDF5/releases/hdf5-1.8.11/src/hdf5-1.8.11.tar.gz
tar -zxvf hdf5-1.8.11.tar.gz
./configure --prefix=`pwd` --with-zlib=$PROGS/zlib-1.2.8 --enable-fortran --enable-cxx CC=icc FC=ifort
CXX=icpc
make
make install
```

4.1.2. Network Common Data Form (netCDF)

The netCDF library is distributed separately for each programming language. Because of this restriction, it is necessary to follow specific order to install netCDF C (4.3.0), C++ (4.2) and Fortran (4.2) libraries.

```
cd $PROGS
wget ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf-4.3.0.tar.gz
tar -zxvf netcdf-4.3.0.tar.gz
cd netcdf-4.3.0
mkdir src
mv * src/.
cd src
./configure --prefix=$PROGS/netcdf-4.3.0 CC=icc FC=ifort LDFLAGS="-L$PROGS/zlib-1.2.8/lib -
L$PROGS//hdf5-1.8.11/lib" CPPFLAGS="-I$PROGS/zlib-1.2.8/include -I/$PROGS/hdf5-
1.8.11/include"
make
make install
export LD_LIBRARY_PATH=$PROGS/netcdf-4.3.0/lib:$LD_LIBRARY_PATH
```

```
cd $PROGS
wget ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf-cxx-4.2.tar.gz
cd netcdf-cxx-4.2
mkdir src
mv * src/.
cd src
./configure --prefix=$PROGS/netcdf-cxx-4.2 CC=icc CXX=icpc LDFLAGS="-L$PROGS/zlib-1.2.8/lib -
L$PROGS/hdf5-1.8.11/lib -L$PROGS/netcdf-4.3.0/lib" CPPFLAGS="-I$PROGS/zlib-1.2.8/include -
I/$PROGS/hdf5-1.8.11/include -I$PROGS/netcdf-4.3.0/include"
make
make install
```

```
cd $PROGS
wget ftp://ftp.unidata.ucar.edu/pub/netcdf/netcdf-fortran-4.2.tar.gz
cd netcdf-fortran-4.2
mkdir src
mv * src/.
cd src
./configure --prefix=$PROGS/netcdf-fortran-4.2 CC=icc FC=ifort LDFLAGS="-L$PROGS/zlib-1.2.8/lib -
L$PROGS/hdf5-1.8.11/lib -L$PROGS/netcdf-4.3.0/lib" CPPFLAGS="-I$PROGS/zlib-1.2.8/include -
I/$PROGS/hdf5-1.8.11/include -I$PROGS/netcdf-4.3.0/include"
make
make install
```

```
cd $PROGS/netcdf-4.3.0/lib
ln -s ../../netcdf-cxx-4.2/lib/* .
ln -s ../../netcdf-fortran-4.2/lib/* .
ln -s ../../netcdf-cxx-4.2/include/* .
ln -s ../../netcdf-fortran-4.2/include/* .
export NETCDF=$PROGS/netcdf-4.3.0
export PATH=$NETCDF/bin:$PATH
```

The last section of commands is required for reaching to all netCDF interfaces or APIs (C, C++ and Fortran) from the single directory.

4.1.3. Parallel netCDF (optional)

Actually, this step optional but if there is a plan to use ESMF netCDF I/O capabilities to write exchange fields to disk, then parallel netCDF library (1.3.1) is required.

```
cd $PROGS
wget http://ftp.mcs.anl.gov/pub/parallel-netcdf/parallel-netcdf-1.3.1.tar.gz
tar -zxvf parallel-netcdf-1.3.1.tar.gz
cd parallel-netcdf-1.3.1
./configure --prefix=`pwd` --with-mpi=/opt/openmpi/1.6.5/intel/2013psm FC=mpif90 F77=mpif90
CXX=mpiccpc
make
make install
export PNETCDF=$PROGS/parallel-netcdf-1.3.1
```

<u>Note that the path for MPI installation might be change. In this case, user must supply correct path to "--with-mpi" configuration option.</u>

4.1.4. Apache Xerces C++

This library is required for ESMF installation. It is basically responsible from reading/writing grid definitions and attributes (field, component and state level) as XML format.

```
cd $PROGS
wget http://apache.bilkent.edu.tr//xerces/c/3/sources/xerces-c-3.1.1.tar.gz
tar -zxvf xerces-c-3.1.1.tar.gz
cd xerces-c-3.1.1
./configure --prefix=$PROGS/xerces-c-3.1.1 CC=icc CXX=icpc
make
make install
```

4.1.5. Earth System Modeling Framework (ESMF)

The one of the main component of the coupled model is the coupling library, which is used to create driver to control the standalone model components. The detailed and up-to-date information of the installation procedure can be found here[9]. Before starting to the installation of ESMF library, the users need to pay attention to the following issues;

- The coupled model needs special features of **ESMF (7.0.0)**. If there is a plan to use the debug level (>2) to check the exchange fields, then ESMF library might be installed with parallel I/O and netCDF support (for more information, look at the example environment variables defined in ESMF installation related with NETCDF and PNETCDF).
- After netCDF version 4.3.0 the C++ interface is changed and ESMF (tested with coupled model) is not compatible with it. So, it is better to use the <= 4.3.0 version of netCDF in this case.

Example environment variable definitions for ESMF installations:

<u>ARGO[10] (ICTP's Cluster System, Italy):</u>

Note that the list of environment variables (given as tcsh or csh shell syntax, for bash type shell the *setenv* command must be replaced by *export* command along with the equal sign) is specific to **ARGO** cluster, its architecture **(x64_64)**, operating system **(Centos 6.4 final)**, installed compiler **(Intel Compiler, 13.1.0)** and MPI **(OpenMPI, 1.6.5)** versions. They must be modified to install ESMF library to the other computing systems or clusters. The detailed information about definition of environment variables for ESMF library (<u>see known bugs section before</u>) itself can be found in ESMF Web site[11].

[9] http://www.earthsystemmodeling.org/esmf_releases/last_built/ESMF_usrdoc/
[10] http://argo.ictp.it
[11] http://www.earthsystemmodeling.org/download/platforms/

```
setenv ESMF_OS Linux
setenv ESMF_TESTMPMD OFF
setenv ESMF_TESTHARNESS_ARRAY RUN_ESMF_TestHarnessArray_default
setenv ESMF_TESTHARNESS_FIELD RUN_ESMF_TestHarnessField_default
setenv ESMF_DIR $PROGS/esmf_7.0.0
setenv ESMF_TESTWITHTHREADS OFF
setenv ESMF_INSTALL_PREFIX ${ESMF_DIR}/install_dir
setenv ESMF_COMM openmpi
setenv ESMF_TESTEXHAUSTIVE ON
setenv ESMF_BOPT O
setenv ESMF_SITE default
setenv ESMF_ABI 64
setenv ESMF_COMPILER intel
setenv ESMF_PIO internal
setenv ESMF_PNETCDF "standard"
setenv ESMF_PNETCDF_INCLUDE $PROGS/parallel-netcdf-1.3.1}/include
setenv ESMF_PNETCDF_LIBPATH $PROGS/parallel-netcdf-1.3.1}/lib
setenv ESMF_NETCDF "split"
setenv ESMF_NETCDF_INCLUDE $PROGS/netcdf-4.3.0/include
setenv ESMF_NETCDF_LIBPATH $PROGS/netcdf-4.3.0/lib
setenv ESMF_XERCES "standard"
setenv ESMF_XERCES_INCLUDE $PROGS/xerces-c-3.1.1/include
setenv ESMF_XERCES_LIBPATH $PROGS/xerces-c-3.1.1/lib
```

Then following command must be issued to install ESMF,

```
cd $ESMF_DIR
make >&make.log
make install
```

After installation of the ESMF library, user can create a new environment variable (ESMF_LIB and ESMFMKFILE) to help RegESM configure script to find the location of the required files of the ESMF library. The example environment variables for C shell,

```
setenv ESMF_LIB
"${ESMF_INSTALL_PREFIX}/lib/lib${ESMF_BOPT}/${ESMF_OS}.${ESMF_COMPILER}.${ESMF_ABI
}.${ESMF_COMM}.${ESMF_SITE}"
setenv ESMFMKFILE "${ESMF_LIB}/esmf.mk"
setenv PATH
"${ESMF_DIR}/apps/apps${ESMF_BOPT}/${ESMF_OS}.${ESMF_COMPILER}.${ESMF_ABI}.${ESM
F_COMM}.${ESMF_SITE}:${PATH}"
```

At this level it is always better to install ESMF library also with debug support (change *ESMF_BOPT* environment variable as *g* and install the library again after issuing *make clean* command). To have both optimized and debug version of ESMF library might help to find the source of the possible errors caused by the coupled model. The production runs can be done with the optimized version but for debugging use could install the model with debug version of the ESMF library.

To install bit-to-bit reproducible version of the ESMF library, please read the Section 7.

4.2. Installation of RegESM

After installing the required libraries, it is necessary to install individual model components first. In this case, user might install the individual model components to system and defines the installation directories in the configuration phase of the coupled model to build RegESM executable. At this stage, it is better to follow a common convention about the installation and create hierarchical directory structure for installation of the model components, their input and output files. This is crucial because it helps to find the source of the possible errors easily and it also helps to keep clean the working directory. In this case, the suggested directory structure to install RegESM can be seen in Figure 13. As it can be seen from the figure, each model components (including "driver" itself – called as "drv") use its own directory for the source files, input and output. Due to some limitations of the MITgcm model and its design, it is impossible to write/read the model files to/from specific directory and those files must be stored in the **BASE_DIR** (main RegESM working directory – it is defined by user).

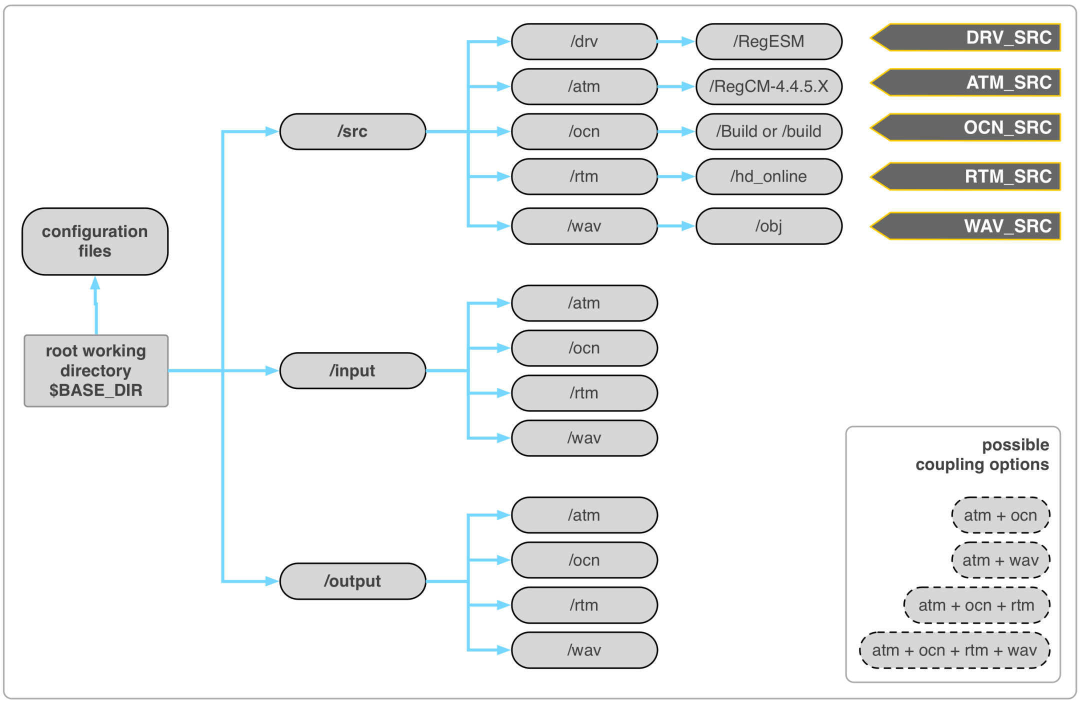


Figure 13 Suggested directory structure for RegESM installation.

In addition to the directory structure, the configuration files, run script (OpenPBS, LSF etc.), input and output files (just for MITgcm model) must be placed in the main working directory (**BASE_DIR**). Also, the RegESM executable is placed in the working directory (the soft link must be created (i.e. following command can be used for this purpose - *cd $BASE_DIR; ln -s $DRV_SRC/regesm.x .*).

4.2.1. Regional Climate Model (RegCM, 4.5)

To install RegCM with coupling support user must issue following commands:

```
mkdir $ATM_SRC
cd $ATM_SRC
wget https://gforge.ictp.it/gf/download/frsrelease/252/1580/RegCM-4.5.0.tar.gz
tar –zxvf RegCM-4.5.0.tar.gz
cd RegCM-4.5.0
autoreconf –f –i (or bootstrap.sh) ← issuing this command could be unnecessary in some cases
./configure –prefix=`pwd` --enable-cpl CC=icc FC=ifort
make
make install
```

The ATM_SRC environment variable is used to point the installation path of the atmospheric model and user might replace it by any valid directory.

The command that is given here is for Intel Compiler and Linux operating system and the commands might change in usage of different compiler and computing environment. The main important issue is that the model components does not need ESMF library anymore (all the ESMF related code is moved to the RegESM (namely ‘driver’). So, the installation of the model is almost same as the standalone version except given extra configure option (--enable-cpl) to enable coupling support.

The additional configuration parameters (see Doc/README.namelist in the RegCM source directory) for the RegCM must be used when the model is running in the coupled mode. In this case, the user must activate following options:

```
&physicsparam
…
iocncpl =       1, ← activates the coupling in RegCM side
iwavcpl =        1, ← activates the coupling in RegCM side
…
```

```
/
…
&cplparam
cpldt      = 10800., ← coupling interval in seconds. It must be same with driver namelist file!
zomax      = 0.005, ← threshold for surface roughness, only valid for wave coupling
ustarmax   = 1.2, ← threshold for frictional velocity, only valid for wave coupling
/
```

The threshold values for surface roughness and frictional velocity are used to keep atmospheric model stable and user might need to play with those number to find their reasonable (as large as possible) values. The detailed information about RegCM, usage and its configuration parameters can be found in the user guide[12] and the reference manual[13].

4.2.2. Regional Ocean Model (ROMS, r809)

To install ROMS with coupling support, the user need to patch the original version of the model. The patch includes set of minor modifications to prepare ROMS ocean model for the coupling. The reader also notes that there is no any generic patch for the all versions ROMS due to the existence of the different versions and branches (i.e. Rutgers University's ROMS, ARGIF ROMS, UCLA ROMS, ROMS with sea-ice) of the model.

The current version of the RegESM comes with a patch that is created by using a snapshot of the ROMS branch[14] with sea-ice support and it can be found in the "tools/ocn" (latest one is roms-r809.patch). Doing modifications and applying the patch to the model are the responsibility of the user but the given version of the patch can be used as a reference to modify the any ROMS version.

```
> svn checkout -r 809 --username [ROMS USER NAME] https://www.myroms.org/svn/src/trunk roms-r809
> patch -p 3 < roms_r809.patch
```

To activate coupling in ROMS, user need to do following definition in the application header file (holds the all CPP options to configure the model, *.h),

Definition for coupling:

```
#define REGCM_COUPLING ← activates the coupling in ROMS side
#define MODIFIED_CALDATE ← fix some bugs in ROMS date conversion routine
#define HD_COUPLING ←optional, if rivers are defined as source points (rivOpt = 2)
#define PERFECT_RESTART ← it is required to restart the coupled model (do not forgot to set the LcycleRST == F and NRST to daily restart output)
```

Suggested definitions for atmospheric forcing:

```
#define ATM_PRESS
#define SHORTWAVE
```

Then ROMS official documentation[15] can be used to install and create a realistic case for the ROMS ocean model.

4.2.3. MITgcm Ocean Model

To install MITgcm (MITgcm_c63s) with coupling support, the user must need to add the ESMF capability to the MITgcm firstly. Please contact with the developer of RegESM for the MITgcm package to activate coupling. Then MITgcm official documentation[16] can be used to install and create a realistic case.

4.2.4. HD River Routing Model

The HD river routing model is the property of Max Planck Institute, Germany and it is distributed via a user agreement. So, it is not distributed freely along with the driver. In our

[12] https://gforge.ictp.it/gf/download/docmanfileversion/63/1152/UserGuide.pdf
[13] https://gforge.ictp.it/gf/download/docmanfileversion/64/1153/ReferenceMan.pdf
[14] https://github.com/kshedstrom/roms
[15] https://www.myroms.org/wiki/index.php/Documentation_Portal
[16] http://mitgcm.org/public/docs.html

design, we modified the HD model to allow easy to use and plug into RegESM driver. The following list summarizes the modification done by ITU.

- The file format of the model output and restart files are changed from binary (SRV format) to netCDF.
- The model main code is split into three parts: initialization, run and finalize by following common convention of the ESMF library.
- The necessary modifications are done for the coupling. In this case new preprocessor flag is added to activate coupling.
- A set of NCL functions is created to drive the standalone model using RegCM output and to create SRV formatted input files.

Because of the license restriction that is mentioned previously, the HD model and the modifications for the coupling are not distributed publicly via GitHub repository. If there is plan (and need) to use the three component modeling system, then user should get the license from the Max Planck Institute for the standalone version of the HD model and contact with ITU/ICTP to get the modified version of the HD model to use in the coupled model configuration. The user also port his/her own RTM component to the coupled modeling system by following same methodology given the list and the help of the RegESM source code (mod_esmf_rtm.F90).

To install modified version of HD model with coupling support, user should edit the Makefile and issue following commands:

```
mkdir $RTM_SRC
tar –zxvf hd_online.tar.gz
cd hd_online
make
make install
```

The Makefile has a set of macro to define the compiler, netCDF library and coupling support. –DCPL option must be added to FC macro along with the compiler command to activate the coupling support.

4.2.5. WAM Wave Model

The ECMWF's WAM version used in the coupled model is Cycle_4.5.3_MPI. The original version of WAM model is modified to add it as a component in the coupled modeling system. To install modified version of WAM model with coupling support, user should issue following commands:

```
mkdir $WAM_SRC
tar -zxvf wav.tar.gz
cd mk
Edit hidden .dirset file. Change installation path (PRODADMDIR) and FCFLAGS if it is needed.
The FCFLAGS must include -DCPL to activate coupling support. You might also need to modify
the make_netcdf and pnetcdf.mk to point NetCDF library location.
./create_binaries
```

4.3. Installation of RegESM Driver

After installation of the individual model components (i.e. RegCM, ROMS or MITgcm), the RegESM driver can be installed. The RegESM basically uses the object (*.o), module (*.mod) and header (*.h) files of the model components to create static library files for each model component (libatm.a for atmosphere, libocn.a for ocean, librtm.a for river routing and libwav.a for wave model) and uses these static libraries to create the final RegESM single executable.

The ESMF library is also required to compile ESMF related component codes in "driver" side. The configuration script basically tries to find the installation directory of ESMF by looking into a specific environment variable (**ESMF_LIB**). If **ESMF_LIB** environment variable points the directory of ESMF shared library (libesmf.so) and configuration file (esmf.mk) exist and valid then it could be used to compile the RegESM. In case of non-defined **ESMF_LIB** environment variable, user might specify the ESMF library directory by using **--with-esmf** configure option.

Currently, RegESM project is maintained and distributed by using a GitHub repository[17] under GNU GPL license. The user guide can be found under the doc/ directory. To open new issue ticket, bug report or feature request that are related with the driver, the GitHub page is also used. Due to the limitation of the human resource to develop and maintain the coupled modeling system, the solution of the possible bugs and issues could be delayed.

The model can be installed using following commands ($ESM_SRC is the main directory for RegESM installation):

```
cd $ESM_SRC
git clone https://github.com/uturuncoglu/RegESM.git
cd RegESM
*** if four components (ATM-OCN-RTM-WAV) are activated ***
./configure --prefix=`pwd --with-atm=$ATM_SRC --with-ocn=$OCN_SRC/Build --with-rtm=$RTM_SRC
–with-wav=$WAV_SRC/obj CC=icc FC=ifort
*** if three components (ATM-OCN-RTM) are activated ***
./configure --prefix=`pwd --with-atm=$ATM_SRC --with-ocn=$OCN_SRC/Build --with-rtm=$RTM_SRC
CC=icc FC=ifort
*** if only two components (ATM-OCN) are activated ****
./configure --prefix=`pwd --with-atm=$ATM_SRC --with-ocn=$OCN_SRC/Build CC=icc FC=ifort
*** if only two components (ATM-WAV) are activated ****
./configure --prefix=`pwd --with-atm=$ATM_SRC --with-wav=$WAV_SRC/obj CC=icc FC=ifort
make
make install
```

The configure options **--with-atm**, **--with-ocn**, **--with-rtm** and **--with-wav** are used to point the installation directories of the model components. For ROMS case, **--with-ocn** option must point "**Build**" directory that holds the compiled source files of the ROMS installation but **"build"** directory for MITgcm. This is also similar for the wave component (**"obj"** directory for WAM).

The configure script is smart enough to check some key files to find the types of the ocean model component (ROMS, ROMS with sea-ice support or MITgcm) and compiles the required files suitable for selected model components and the configuration. In addition, the configure script also checks the given ROMS model installation directory (if this is the case) to enable the sea-ice related part of the data exchange routines in the "driver" side. So, user does not need to set any other option when using sea-ice enabled ROMS version.

[17] https://github.com/uturuncoglu/RegESM

5. Usage

To run the RegESM model, user needs to create or modify two configuration files of driver in addition of the configuration files of individual model components:

- Exchange field table (exfield.tbl)
- Driver configuration file (namelist.rc)

Then, user is able to run the coupled modeling system. Also note that both exchange field definition and the model configuration file must be placed with the same directory as RegESM executable file.

5.1. Exchange Field Table (exfield.tbl)

The exchange field table basically keeps the definition of the exchange fields and their attributes (i.e. description, units, grid location etc.). The main structure of the exchange field table can be seen in Table 3.

Table 3 Main structure of the exchange filed table.

[Number of Fields - N] [Direction of Coupling] [Activate Extrapolation – T or F] [Exchange Field 1] [Exchange Field 2] … **[**Exchange Field N] **[Number of Fields - N] [Direction of Coupling] [Activate Extrapolation – T or F]** [Exchange Field 1] [Exchange Field 2] … [Exchange Field N] **[Number of Fields - N] [Direction of Coupling] [Activate Extrapolation – T or F]** … …

As it can be seen from the table, the file has header section for each coupling direction. In this case the number of field defined in the header must be same with the number of field added just after the header section. Currently, the direction of the coupling can be defined as:

- atm2ocn
- ocn2atm
- atm2rtm
- rtm2ocn
- atm2wav
- wav2atm

but this section will be expanded when new modeling components (i.e. ice etc.) will be added to the coupled modeling system. The next parameter enables or disables the support of extrapolation (via two step extrapolation). It can be simply set as T (for True – enables extrapolation support) or F (for False – disables extrapolation support). In this case, setting true for the extrapolation support is suggested.

After definition of the header section, the user needs to define the exchange fields and their attributes based on the table (see Appendix, Table 4). In this case, attributes of the exchange fields are separated ":" (double-comma) symbol. The example exchange field tables for different model setups and detailed definition abut them can be found under **external/** directory of the RegESM source.

- **exfield_000.tbl** – Two-component configuration (ATM-OCN; RegCM+ROMS with ice): The ROMS uses BULK_FLUX parameterization in this case.
- **exfield_001.tbl** – Three component configuration (ATM-OCN-RTM; RegCM+ROMS with ice+HD): Again ROMS model uses BULK_FLUX parameterization and river discharge is activated by using specific CPP options in ROMS model.
- **exfield_002.tbl** – Two component configuration (ATM-OCN; RegCM+MITgcm)
- **exfield_003.tbl** – Two component configuration w/o using bulk flux algorithm to define surface heat and momentum fluxes (ATM-OCN; RegCM+ROMS)

The definition of the exchange field table mainly depends on the application and activated individual model components itself and it is impossible to create a generic definition for all the application and cases.

5.2. RegESM Configuration (namelist.rc)

It is simply followed ESMF convention to support generic configuration file for RegESM, The configuration file is manly responsible from PET assignment to model components (both number of CPU or core and the type of execution – sequential vs. concurrent), definition of high-level simulation period (start, stop, restart time and calendar), coupling time step (slowest one), matrix of coupling time step multiplier to calculate fast and slow time steps for data exchange among the components, debug level and list of rivers (along with coordinates, number of source point and monthly correction factors) that is used for the RTM coupling. The detailed explanation of each configuration option can be found in Appendix section (Table 5).

5.3. Running RegESM

Running of the coupled model is very similar to running any other standalone model component. In this case, all the configuration files of individual model components must be in the same director with the executable.

The following sample scripts can be used as a base.

RegCM+MITgcm using OpenPBS job scheduler (on ICTP's ARGO cluster): regesm.job

```
#!/bin/bash

#PBS -N test
#PBS -l walltime=24:00:00
#PBS -l nodes=72
#PBS -q esp

# load required modules
. /etc/profile.d/modules.sh
module purge
module load intel/2013
module load openmpi/1.6.5/intel/2013

setenv PROGS /home/netapp/clima-users/users/uturunco/progs
setenv XERCES $PROGS/xerces-c-3.1.1
setenv PNETCDF $PROGS/parallel-netcdf-1.3.1
setenv NETCDF $PROGS/netcdf-4.3.0
setenv HDF5 $PROGS/hdf5-1.8.11
setenv PATH $NETCDF/bin:$PNETCDF/bin:$PATH
setenv LD_LIBRARY_PATH $NETCDF/lib:$PNETCDF/lib:$XERCES/lib:$HDF5/lib:$PROGS/zlib-
1.2.8/lib:$LD_LIBRARY_PATH

setenv ESMF_OS Linux
setenv ESMF_TESTMPMD OFF
setenv ESMF_TESTHARNESS_ARRAY RUN_ESMF_TestHarnessArray_default
setenv ESMF_TESTHARNESS_FIELD RUN_ESMF_TestHarnessField_default
setenv ESMF_DIR $PROGS/esmf-6.2.0
setenv ESMF_TESTWITHTHREADS OFF
setenv ESMF_INSTALL_PREFIX ${ESMF_DIR}/install_dir
setenv ESMF_COMM openmpi
setenv ESMF_TESTEXHAUSTIVE ON
setenv ESMF_BOPT O
setenv ESMF_OPENMP OFF
setenv ESMF_SITE default
setenv ESMF_ABI 64
setenv ESMF_COMPILER intel
setenv ESMF_PIO internal
setenv ESMF_PNETCDF "standard"
setenv ESMF_PNETCDF_INCLUDE ${PNETCDF}/include
setenv ESMF_PNETCDF_LIBPATH ${PNETCDF}/lib
setenv ESMF_NETCDF "split"
setenv ESMF_NETCDF_INCLUDE ${NETCDF}/include
setenv ESMF_NETCDF_LIBPATH ${NETCDF}/lib
```

```
setenv ESMF_XERCES "standard"
setenv ESMF_XERCES_INCLUDE ${XERCES}/include
setenv ESMF_XERCES_LIBPATH ${XERCES}/lib

setenv ESMF_LIB
"${ESMF_INSTALL_PREFIX}/lib/lib${ESMF_BOPT}/${ESMF_OS}.${ESMF_COMPILER}.${ESMF_ABI
}.${ESMF_COMM}.${ESMF_SITE}"
setenv ESMFMKFILE "${ESMF_LIB}/esmf.mk"
setenv PATH
"${ESMF_DIR}/apps/apps${ESMF_BOPT}/${ESMF_OS}.${ESMF_COMPILER}.${ESMF_ABI}.${ESM
F_COMM}.${ESMF_SITE}:${PATH}"

setenv LD_LIBRARY_PATH ${ESMF_LIB}:${LD_LIBRARY_PATH}
setenv PATH
${ESMF_INSTALL_PREFIX}/bin/bin${ESMF_BOPT}/${ESMF_OS}.${ESMF_COMPILER}.${ESMF_AB
I}.${ESMF_COMM}.${ESMF_SITE}:${PATH}

# run coupled model
cd /home/netapp/clima-users/users/uturunco/MED/RegESM/run2
ulimit -s unlimited
mpirun -v ./regesm.x regcm.in_MED50km >& regesmout.txt
```

RegCM+ROMS using LSF job scheduler (on UHEM, Turkey): regesm.lsf

```
#!/bin/bash
#BSUB -P avktis
#BSUB -J cpl
##BSUB -q mid
#BSUB -q deci9
#BSUB -m karadeniz_prace
#BSUB -o %J.out
#BSUB -e %J.err
#BSUB -a intelmpi
#BSUB -n 32

mpirun.lsf ./regesm.x regcm.in_MED50km med12.in > regesmout.txt
```

Also not that in this case RegESM executable gets two-configuration file (one for RegCM and one for ROMS).

6. Known Limitations

6.1. Conservation of the exchange fields

In the current version of the coupled modeling system, a prototype version of the conservation algorithm is implemented. In this case, the conservation algorithm ensures that the global integral of the source and destination fields (over the matched regions) will remain same but it does not guarantee local conservation of the exchange fields. It just applies the difference of the global integral of source and destination fields to the destination field across the domain.

The implemented conservation algorithm works as following;

1. First, the algorithm finds the matched grid points between the model components (atmosphere and ocean models).
2. It calculates the global integral of the source and destination fields
3. It calculates the difference of the global integrals (destination – source)
4. Then, it calculates the unit difference (by dividing the difference to the total surface area in the destination grid)
5. Then, it applies the unit difference to each grid cell

The local conservation feature might be implemented in the future version of the modeling system with availability of the high-order conservative type interpolation in the ESMF side. Also, user might also aware that the first-order conservative interpolation technique, which is currently supported by ESMF, might create artifacts (square like shapes in the destination field) in the interpolation when the ration between grid resolution of components is high (i.e. 50 km ATM and 5-7km in OCN).

The user also notes that the current conservation algorithm is only works with bilinear type interpolation (It can be defined in the exchange field table for each variable separately) and also only supports data exchange between atmosphere and ocean components (ATM-OCN and OCN-ATM) when extrapolation is activated.

7. Known Issues / Problems / Bugs

7.1. Two Level Interpolation

There is a problem in the nearest neighbor search if the internal piece of the source grid is completely masked out on a processor. The problem appears along with the large and unrealistic values in the result field (at source grid) when specific number of processor is used. The number of processor that causes to the error depends on the grid size, land-sea masking and also number of tiles in each direction (x and y) that is used for the domain decomposition purposes. The problem exists in the ESMF library (6.3.0r) but it is fixed in the newer versions (> 6.3.0r).

7.2. Problem in Three Component Case

The ESMF library version (< 7.0.0b38) has a bug and the modeling system does not run as expected when three component case (ATM+OCN+RTM) is activated. The problem is in the NUOPC layer of the ESMF library but it is fixed in version 7.0.0b38. Due to this reason, the latest version of RegESM is only compatible with the versions >ESMF 7.0.0b38.

7.3. Unaligned and/or Mismatched Component Grids

<u>In the current version of the coupled modeling system, numerical grid of the components (OCN, RTM and WAV) must be completely covered by the atmospheric model (RegCM) grid and all the components should use the same convention for the definition of **latitude (-90 to +90) and longitude (-180 to +180) limits**</u>. Due to these restrictions, the problem in the definition of the component grids might produce error in the ESMF library when ESMF tries to create weight matrices (called routehandle in ESMF terminology) for the interpolation among the grid components.

The following error message is an example for the violation of latitude/longitude convention. In this case, atmosphere model uses -90 to +90 for latitude and -180 to +180 for the longitude but ocean model uses -90 to +90 for latitude and 0 to +360 for the longitude to define the numerical grids.

<u>Error message from standard output of the coupled model:</u>

```
20150708 170134.569 ERROR           PET47 ESMCI_Regrid_F.C:474 c_esmc_regrid_create()
Internal error: Bad condition - Condition {sf.size() == rf.size()} failed at /home/netapp/clima-users/users/uturunco/progs/esmf-7.0.0b38/src/Infrastructure/Mesh/src/ESMCI_CommReg.C, line:108
20150708 170134.569 ERROR           PET47 ESMF_Regrid.F90:321 ESMF_RegridStore Internal error: Bad condition - Internal subroutine call returned Error
20150708 170134.569 ERROR           PET47 ESMF_FieldRegrid.F90:1035
ESMF_FieldRegridStoreNX Internal error: Bad condition - Internal subroutine call returned Error
20150708 170134.569 ERROR           PET47 mod_esmf_cpl.F90:344 Internal error: Bad condition - Passing error in return code
20150708 170134.569 ERROR           PET47 ATM-TO-OCN:src/addon/NUOPC/src/NUOPC_Connector.F90:1613 Internal error: Bad condition - Passing error in return code
20150708 170134.569 ERROR           PET47
regesm:src/addon/NUOPC/src/NUOPC_Driver.F90:1190 Internal error: Bad condition - Phase 6 Initialize for connectorComp 1 -> 2: ATM-TO-OCN did not return ESMF_SUCCESS
20150708 170134.569 ERROR           PET47 regesm:src/addon/NUOPC/src/NUOPC_Driver.F90:783 Internal error: Bad condition - Passing error in return code
20150708 170134.569 ERROR           PET47 regesm.F90:107 Internal error: Bad condition - Passing error in return code
```

To be sure that the model components are totally covered by the atmospheric model, the user could set the debug level in the coupler configuration file (namelist.rc) as 2 (or higher) and run the model again. In this case, coupled modeling system creates set of files with *.vtk (Visualization Toolkit) extension. These files basically store the grid information of all the model components and grid stencils (cross, dot, u and v). By using visualization tools like

ParaView[18] and VisIt[19], the generated VTK files can be read and checked. The Figure 13 is belongs to the case that produces error.

Figure 14 Example grids of model components (red is atmosphere and blue is ocean) that use different longitude convention. The boxes with same color show domain decomposition elements of the model component.

As it can be seen from the Figure, the both grids use different convention in the definition of longitude coordinates. In this case, the ocean model grid (blue) does not match with the atmosphere grid (red) even they are seems overlapped with other visualization tools such as NCL.

7.4. Bit-to-bit Reproducibility

The bit-to-bit reproducibility becomes important in the scientific studies when user needs to compare the results of the different configurations of the same model simulation to check the effects of the different components. The bit-to-bit reproducibility is still active research area in the earth system science and needs special attention to the used model components and the working environment (i.e. operating system, compiler). The issue is mainly related with the floating-point arithmetic and the representation of the numbers in the computer (it has a finite resolution to store the numbers). Due to this reason, numerical results change run to run of the same executable. Numerical results also change between different systems. For example, using same operating system along with different CPU version (Intel® Xeon® Processor E5540 vs. Intel® Xeon® Processor E3-1275) probably affect the results. The problem could be caused by different reasons: the algorithm, alignment of the data in heap and stack (kind of memory), task or thread scheduler in OpenMP and MPI parallelism. The order of the numerical operations has also impact on the rounded results (i.e. reduction operator in MPI code).

The RegESM model is able to generate bit-to-bit reproducible results when specific compiler flags are used to compile/install the model components and also ESMF library.

ESMF:

To install ESMF library along with the support of bit-to-bit reproducibility, user needs to modify build_rules.mk configuration file (under build_config/ directory in the ESMF source code). The memory model of the compiler defined in ESMF (x86_64_32, x86_64_small, x86_64_medium, x86_64_mic, ia64_64) could change based on the used system and its architecture. In the following example, it is x86_64_small. In this case, following modifications must be done before installing ESMF library.

Original: build_config/Linux.intel.default/build_rules.mk:

```
…
…
ifeq ($(ESMF_ABISTRING),x86_64_small)
ESMF_CXXCOMPILEOPTS       += -m64 -mcmodel=small
ESMF_CXXLINKOPTS          += -m64 -mcmodel=small
ESMF_F90COMPILEOPTS       += -m64 -mcmodel=small
ESMF_F90LINKOPTS          += -m64 -mcmodel=small
endif
…
…
```

[18] http://www.paraview.org

[19] https://wci.llnl.gov/simulation/computer-codes/visit/

Modified: build_config/Linux.intel.default/build_rules.mk:

```
…
…
ifeq ($(ESMF_ABISTRING),x86_64_small)
ESMF_CXXCOMPILEOPTS       += -m64 -mcmodel=small -fp-model precise
ESMF_CXXLINKOPTS          += -m64 -mcmodel=small -fp-model precise
ESMF_F90COMPILEOPTS       += -m64 -mcmodel=small -fp-model precise
ESMF_F90LINKOPTS          += -m64 -mcmodel=small -fp-model precise
endif…
…
```

Then, ESMF library can be installed as usual. Also note that ESMF_DIR and ESMF_LIB and ESMF_INC environment variables might be changed based on the new installation (see Section 4.1.5 for the ESMF installation). The ESMF installed with "-fp-model precise" option will support bit-to-bit reproducibility.

Model Components:

After installation of ESMF library, all the model components must be installed with "-fp-model precise" compiler flag (Intel). The RegCM version < 4.4.5.7 could have "-fp-model fast=2" by default. If this is the case, user must edit *configure.ac* to change "-fp-model fast=2" to "-fp-model precise". Then, *bootstrap.sh* script must be executed to create modified configure script.

General Comments and References:

Also note that the compiler flag (-fp-model precise) controls the numerical operations to achieve the bit-to-bit reproducibility but it might reduce the overall performance of the model components and the driver by %5-10. More information about the issue can be found in the following references:

[1] Song-You Hong, Myung-Seo Koo, Jihyeon Jang, Jung-Eun Esther Kim, Hoon Park, Min-Su Joh, Ji-Hoon Kang, and Tae-Jin Oh, 2013: An Evaluation of the Software System Dependency of a Global Atmospheric Model. Mon. Wea. Rev., 141, 4165–4172.

[2] Liu, L., Li, R., Zhang, C., Yang, G., Wang, B., and Dong, L.: Enhancement for bitwise identical reproducibility of Earth system modeling on the C-Coupler platform, Geosci. Model Dev. Discuss., 8, 2403-2435, doi:10.5194/gmdd-8-2403-2015, 2015.

[3] http://scienceblogs.com/stoat/2013/07/29/repeatability-of-large-computations/

[4] http://www.easterbrook.ca/steve/2010/09/verification-and-validation-of-earth-system-models/

[5] https://software.intel.com/en-us/articles/consistency-of-floating-point-results-using-the-intel-compiler

[6] http://www.eecs.berkeley.edu/~hdnguyen/public/talks/IWASEP9.pdf

7.5. Problem in RegCM Related with SST Update

The diurnal cycle SST scheme (idcsst) must be deactivated (set to 0) along with the ocean model coupling. It basically overwrites the SST data come from ocean component and might create strange results in the model. This problem exists in all RegCM 4.4.X.X versions. The new control will be added to RegCM 4.5 to automatically disable diurnal cycle SST scheme when the coupling is activated (iocncpl = 1).

8. Reference

<u>Please cite following paper/s if you use RegESM in your study.</u>

- Turuncoglu, U.U., Sannino, G., 2016. Validation of newly designed regional earth system model (RegESM) for Mediterranean Basin, Climate Dynamics, doi:10.1007/s00382-016-3241-1
- Turuncoglu, U.U., Giuliani, G., Elguindi, N., Giorgi, F., 2013. Modelling the Caspian Sea and its catchment area using a coupled regional atmosphere-ocean model (RegCM4-ROMS): model design and preliminary results, Geosci. Model Dev., 6, pp. 283-299, DOI: 10.5194/gmd-6-283-2013

APPENDIX - A

Table 4 Definition of the fields (or columns) of the exchange field table

#	Column Name	Description
1	**Short Name**	**The sort name of the exchange field. The list of the available fields:** taux - zonal surface wind stress (N/m^2 or Pa) tauy - meridional surface wind stress (N/m^2 or Pa) wndu - zonal wind component (m/s) wndv - meridional wind component (m/s) wspd - wind speed (m/s) psfc - surface pressure (hPa or mb) tsfc - 2 meter surface temperature (K) qsfc - 2 meter specific humidity (kg/kg) lwrd - net longwave radiation (W/m^2) swrd - net shortwave radiation (W/m^2) dlwr - downward longwave radiation (W/m^2) dswr - downward shortwave radiation (W/m^2) lhfx - latent heat flux (W/m^2) shfx - sensible heat flux (W/m^2) nflx - net heat flux, latent+sensible+longwave-shortwave (W/m^2) prec - total precipitation, P (m/s) evap - evaporation, E (m/s) sflx - net freshwater flux, E-P (m/s) rnof - surface runoff (m/s, just over land) snof - sub-surface runoff (m/s, just over land)
2	**Standard Name**	The standard name of the exchange filed. The user can use anything in here but using CF conventions for the field is suggested.
3	**Interpolation Type**	The type of interpolation that is used to transfer data form source grid to destination. It can be defined as “bilinear”, “conserv”, “nearstod”, “neardtos” and “none”. __Restrictions:__ • When transferring data from RTM component to the ocean model the interpolation type must be selected as “nearstod” The “bilinear” type interpolation type must be use to apply conservation correction to the field
4	**Source Grid Point**	This field basically used to define the location of the source filed in the grid definition (Arakawa type grids). It can be defined as “cross”, “dot”, “u” and “v”.
5	**Destination Grid Point**	It is same with the previous filed but in this case it defined the destination point in the destination grid.
6	**Input Unit**	It is just for information. The model does not use this information. The space character is not allowed.
7	**Output Unit**	Same as input unit
8	**Scale**	Used for the unit conversion of the exchange field. The source field is multiplied by this scale factor after interpolation. The user can use scale factor as a number or can use following shortcuts. cf1 – rho0*cp (rho0 = 1025.0 kg/m^3, water density and cp = 3985.0 J/kg/K, heat capacity) cf2 – 1/cf1 cf3 – 1/rho0 It is also possible to use – symbol in the definition of the shortcuts (i.e. –cf1).
9	**Offset**	Used for unit conversion of the exchange filed. The value is added to the source field after applying scale factor. So, the combination of the Scale and Offset parameters might help to convert the units.
10	**Flag for Conservation**	It can be “T” or “F”. If it is defined as true (T), then coupler component applies the difference of source and destination field integral to the destination field. It can be applied in field basis. In general, heat flux components can be defined as true to conserve the fields globally. __Restrictions:__ • It just works between ATM and OCN components The interpolation type of the field must be defined as “bilinear”

Table 5 Configuration options and their descriptions

#	Option	Description
1	**PETLayoutOption**	It is used to define the execution mode of the coupled model (see Fig.4). The value can be **“sequential”** or **“concurrent”**. • If **“sequential”** is selected then model components are triggered by sequential fashion (one after another). In this case, set same number of processor to all components using “PETs” parameter of the configuration file except RTM (it must be 1) component and physically same PETs will be used by the components with order. • If **“concurrent”** option is selected, then model component run in parallel (each component uses distinct PETs). In this case, PETs are not overlapped and it allows parallel execution of the components. The user also note that the informative output of the model components (basically standard output or stdout) can be mixed due to the race conditions among the independent PETs.
2	**CouplingType**	It is used to set coupling type. Currently modeling system supports two different coupling types: **explicit** and **semi-implicit**
3	**PETs**	It defined the PET distribution of the model components. The number of PETs must be given in a specific order with a space between them. Currently, the order is ATM, OCN, RTM and WAV respectively. • If **“sequential”** PET layout selected and same number of PET must be given to all components then driver uses same number of PETs in the job execution. • In contrast to **“sequential”** execution mode, user might select the **“concurrent”** type execution. In this case, the number of PETS is assigned to the components based on the given number of processor. The total usage of the PETs will be the sum of PETs defined for components. The RTM component is sequential and uses only single processor. If the number of processor sets as -1, then coupler assigns the last PET to the RTM component. If the number of processor sets as 1 (the total number of processor used by the coupler is ATM+OCN+WAV), then coupler uses extra resource for the RTM component and assigns one extra processor for it. In this case, total number of processor used by the coupler is the sum of ATM+OCN+WAV+1.
4	**UnmappedFill**	Activate extrapolation for unmapped grid cells. It basically aims to handle unaligned land-sea mask among the model components automatically by performing nearest-neighbor type interpolation. It is suggested to keep it active. Otherwise, user must manually edit components land-sea masks.
5	**DebugLevel**	The debug level is used to find the source of the possible errors (i.e. wrong grid representation, artifacts in the exchange fields, upper and lower limits of the decomposition elements etc.). Actually, there are four level of debug option: **1.** It enables no debug output. This is not suggested generally. **2.** It enables minimal debug output from driver component. In this case, model only prints informative messages related with the upper and lower bounds of the decomposition elements, name of running components etc. **3.** It enables writing grid information of the components in VTK format. The VTK files can be used to create visual output of the grid structures. In this case, Visit or another similar visualization tool that supports VTK format. **4.** It enables writing exchange fields to a netCDF file. In this case, files are written to disk in each coupling time step. This option must be use for short runs to check the correctness of the exchange fields. It must be used with caution because it creates lots of file (depends on coupling time step, number of component and number of exchange field). For this option, ESMF must be compiled with parallel-netcdf support. The user also notes that the exchange fields might be in wrong dimension order (transposed). This is not a bug or error. It is normal and just way of storing arrays in ESMF side (ESMF uses right-hand coordinate system). **5.** This level is same with the previous level but in this case the exchange fields are written to disk in ASCII format. Due to the limitation of the keeping track of the unit numbers in Fortran, this level might produce corrupted files and it must be used with caution. The user also notes that all the levels (< 4) also include the previous level of debug output.
6	**Calendar**	This configuration option defines the global calendar type used in the time synchronization among the components. The coupled model currently supports three different calendar option:

		• gregorian • 360_day • julian The “driver” basically uses this information to check the components calendars to have a consistent view of the time dimension among the model components.
7	**StartTime**	Sets simulation start time. It must be consistent among the model components otherwise “driver” triggers an error message and kills the all processes.
8	**RestartTime**	Sets simulation restart time. If start time and restart time differs then “driver” try to run the model components in restart mode. It must be consistent among the model components otherwise “driver” triggers an error message and kills the all processes. There is also **“restart.sh”** script under tools/ directory to help to organize the configuration files of the model components in case or restarting. Also note that this script does not modify the **namelist.rc** and user must need to modify it manually.
9	**StopTime**	Sets simulation start time. It must be consistent among the model components otherwise “driver” triggers an error message and kills the all processes.
10	**TimeStep**	This configuration of option is used to set the coupling interval (the slowest one) among the components. The model components might interact with each other with different coupling interval (fast vs. slow time step – asynchronous coupling). In this case, **DividerForTStep** configuration option (basically it holds the divider matrix) is used to calculate coupling interval among the components.
11	**DividerForTStep**	It is used to calculate coupling interval among the model components. By using this option, user might define different coupling interval for each coupling direction. This is basically required for asynchronous coupling due to the different response time of the model components and their time resolution limitations. For example, HD river routing model runs in daily time scale and there is no any mean to couple RTM component with ATM and OCN with time scale less than day but ATM and OCN component might be coupled with 3-hour interval. So, this is the possible case to use this configuration as a divider matrix. It basically divides the time step (defined in **TimeStep** option) to get the coupling direction time step. For example, if the Time Step is defined as 1-day and divider matrix is set to 8 in ATM-OCN and OCN-ATM direction then the coupling step for ATM-OCN and OCN-ATM direction will be 3-hours. The **TimeStep** must be defied as a slowest time interval.
12	**RiverOpt**	It is used to define the type of river discharge handling. The latest version of the code supports two different options. **1.** The rivers are defined as point sources. In this case, the required configuration must be done in the ocean model side. **2.** The rivers can be defined as surface boundary condition (SBC). In this case, the river discharge comes from RTM component is distributed to the ocean surface using effective radius defined in the RiverList parameter.
13	**RiverList**	This configuration is basically required for coupling with RTM component. It holds: • **Definition of river position:** index (I,J) or coordinate (LON,LAT) pair. • **River type:** (0) not active, (1) active via RTM and (2) active but constant monthly values • **Effective Radius:** It is used only for **RiverOpt = 2**. The unit is kilometer and it controls the area of extent of the river discharge. • **River Coordinate:** If **River Type = 0**, then I, J pair must be given. Otherwise LON, LAT is expected. The algorithm finds the closest ocean model grid and distributes the river discharge by using effective radius. • **River direction:** It is used in ROMS coupling when **RiverOpt = 1** • **Number of source point for specific mouth:** In some cases it is better to distribute the river discharge along the source point for same river. The “driver” basically divides the river discharge to this number to find the discharge of the each individual source point. It is used in ROMS coupling when **RiverOpt = 1** • **Monthly values:** If **River type = 1**, then these values are assumed as correction factors or weights (one for each month – total count is 12) for simplified bias correction. The weights must be between 0 and 1 to correct seasonal distribution of river discharge. The correction factors can be calculated by comparing RTM model generated river discharges with the observations and set in here. If **River type = 2**, then these values are used as monthly river discharge for specified river. RTM component does not provide discharge for these rivers. This option currently only works with the RegCM+ROMS+HD configuration and still testing for RegCM+MITgcm+HD.