

---

# **norESM documentation Documentation**

**The NorESM developers group**

**Dec 01, 2020**



## CONTENT:

<b>1</b>	<b>How to contribute</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Downloading NorESM2 . . . . .	5
1.3	Running NorESM2 . . . . .	8
1.4	NorESM2 output . . . . .	60
1.5	NorESM2 diagnostics . . . . .	63
1.6	NorESM2 Model Description . . . . .	75
1.7	NorESM2 key simulations . . . . .	83
1.8	Access NorESM data: CMIP and other . . . . .	83
1.9	Contribute to NorESM2 . . . . .	90
1.10	Frequently asked questions . . . . .	95
<b>2</b>	<b>Search</b>	<b>101</b>



The purpose of the NorESM2 User's Guide is to provide a common place for NorESM2 users and developers to share information. E.g. How is the code organised? What tools are used? Which version should be run for what purpose?

The NorESM1 version ReadTheDocs can be found here

<https://noresm-docs.readthedocs.io/en/noresm1/>



## HOW TO CONTRIBUTE

If you have any questions or interested in actively participate in the code or documentation, create a new issue on github at <https://github.com/NorESMhub/NorESM/issues>

Further information on how to contribute to this documentation can be found at <https://github.com/NorESMhub/NorESM/blob/master/doc/README.md>.

For obtaining news per email register to [users@noesm.org](mailto:users@noesm.org) at <https://www.noesm.org> or via this [link](#).

### 1.1 Introduction

#### 1.1.1 NorESM2 User's Guide

This guide instructs both novice and experienced users on building and running NorESM2 for various experiment set ups. The chapters attempt to provide relatively detailed information on how to make, set up, build, run and modify experiments using NorESM2.

#### 1.1.2 NorESM2

The Norwegian Earth System Model version 2 (NorESM2) is an coupled Earth System Model developed by the Norwegian Climate Consortium (NCC). NorESM2 is based on the second version of the Community Earth System Model, CESM2 (<http://www.cesm.ucar.edu/models/cesm2/>), developed and operated at the National Center for Atmospheric Research (NCAR), Boulder, US.

The NorESM specific development is led by the Norwegian Meteorological Institute and NORCE Norwegian Research Centre AS. Other partners involved are the University of Oslo (UiO), CICERO, Nansen Environmental and Remote Sensing Center (NERSC) and the University of Bergen (UiB).

#### **NorESM2 specific additions to CESM2 includes (but is not limited to):**

- Atmosphere model : CAM6-Nor replaces standard CAM
  - Atmospheric chemistry/aerosol/cloud module: OsloAero6 (Kirkevåg et al. GMD, 2018)
  - Atmospheric dynamics/physics: Improved conservation of energy and angular momentum (Tonizzo et al. GMD, 2020)
  - Parameterization of turbulent air-sea fluxes (see *Atmosphere* for more details)
- Sea-ice model:
  - Wind drift of snow

- Ocean model : Isopycnic coordinate model BLOM
- Ocean biogeochemical model : iHAMOCC
  - HAMBurg Ocean Carbon Cycle model (HAMOCC) adopted for use with the isopycnic ocean model BLOM and further developed (Tjiputra et al. GMD, 2020)

For a short description of the model components, please see *NorESM2 Model Description*

### Ocean model component in NorESM2:

BLOM/iHAMOCC replaces MICOM/HAMOCC as the combined physical and biogeochemical ocean model component in NorESM2. BLOM/iHAMOCC is publically available source code licensed under a LGPLv3 license, but is otherwise a direct descendant of the MICOM/HAMOCC model component. New applications of NorESM2 will only use BLOM/iHAMOCC, but older data sets may still refer to MICOM/HAMOCC.

### NorESM2 exists in three versions:

- **NorESM2-MM**
  - 1 degree resolution for all model components
- **NorESM2-LM**
  - 2 degree resolution for the atmosphere and land components
  - 1 degree resolution for the ocean and sea-ice components
  - CO2 concentration driven (default)
- **NorESM2-MH**
  - 1 degree resolution for the atmosphere and land components
  - 0.25 degree resolution for the ocean and sea-ice components

NorESM2 can be run in emission driven mode for interactive carbon-cycle studies. Currently, this configuration is only supported for the LM-resolution

NorESM2 contributes to the 6th phase of the Coupled Model Intercomparison Project (CMIP6):

<https://www.wcrp-climate.org/wgcm-cmip/wgcm-cmip6>

Scientific documentation in the GMD – Special issue “The Norwegian Earth System Model: NorESM”:

[http://www.geosci-model-dev.net/special\\_issue20.html](http://www.geosci-model-dev.net/special_issue20.html)

NorESM1 Documentation is found here: <https://noresm-docs.readthedocs.io/en/noresm1/>



### 1.1.3 References

Seland, Ø., Bentsen, M., Seland Graff, L., Olivié, D., Toniazzi, T., Gjermundsen, A., Debernard, J. B., Gupta, A. K., He, Y., Kirkevåg, A., Schwinger, J., Tjiputra, J., Schancke Aas, K., Bethke, I., Fan, Y., Griesfeller, J., Grini, A., Guo, C., Ilicak, M., Hafsaahl Karset, I. H., Landgren, O., Liakka, J., Onsum Moseid, K., Nummelin, A., Spensberger, C., Tang, H., Zhang, Z., Heinze, C., Iversen, T., and Schulz, M.: The Norwegian Earth System Model, NorESM2 – Evaluation of the CMIP6 DECK and historical simulations, *Geosci. Model Dev. Discuss.*, <https://doi.org/10.5194/gmd-2019-378>, in review, 2020.

Tjiputra, J. F., Schwinger, J., Bentsen, M., Morée, A. L., Gao, S., Bethke, I., Heinze, C., Goris, N., Gupta, A., He, Y.-C., Olivié, D., Seland, Ø., and Schulz, M.: Ocean biogeochemistry in the Norwegian Earth System Model version 2 (NorESM2), *Geosci. Model Dev.*, 13, 2393–2431, <https://doi.org/10.5194/gmd-13-2393-2020>, 2020.

Toniazzi, T., Bentsen, M., Craig, C., Eaton, B. E., Edwards, J., Goldhaber, S., Jablonowski, C., and Lauritzen, P. H.: Enforcing conservation of axial angular momentum in the atmospheric general circulation model CAM6, *Geosci. Model Dev.*, 13, 685–705, <https://doi.org/10.5194/gmd-13-685-2020>, 2020.

Kirkevåg, A., Grini, A., Olivié, D., Seland, Ø., Alterskjær, K., Hummel, M., Karset, I. H. H., Lewinschal, A., Liu, X., Makkonen, R., Bethke, I., Griesfeller, J., Schulz, M., and Iversen, T.: A production-tagged aerosol module for Earth system models, OsloAero5.3 – extensions and updates for CAM5.3-Oslo, *Geosci. Model Dev.*, 11, 3945–3982, <https://doi.org/10.5194/gmd-11-3945-2018>, 2018.

## 1.2 Downloading NorESM2

### 1.2.1 Downloading the model code

The NorESM2 model code is available through a public GitHub repository: <https://github.com/NorESMhub/NorESM>

#### Git and GitHub

To download the model, you need access to a git command-line client on the machine where you want NorESM2 to build and run. You also need a git user and permissions to obtain the code:

- **Create a github user:** You can create the github user yourself. Go to <https://github.com/join> and create a user. (Choose a user name which is easy to understand, for example FirstnameLastname. You can attach several email-addresses to the same user.)
- Visit this page to learn how to **configure git** (for instance setting your name and email adress, this will be used in git commits): <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>
- Send email to [oyvind.seland@met.no](mailto:oyvind.seland@met.no) to **get the right permissions** for the new github user (The email must contain who you are and the github username).
- When you have the right permissions, you can obtain the code.

### Make a clone of the NorESM repository

You can obtain the code using the command-line git client on the appropriate machine as follows:

```
git clone https://github.com/NorESMhub/NorESM.git <noresm-base>
```

where <noresm-base> is the name of the directory where the latest version of the released code will be stored. You can replace <noresm-base> with the directory name you like.

Enter the <noresm-base> folder:

```
cd <noresm-base>
```

Now you can check which remote servers you have configured:

```
> git remote -v
origin      https://github.com/NorESMhub/NorESM.git (fetch)
origin      https://github.com/NorESMhub/NorESM.git (push)
```

And check which branch you are using:

```
> git branch
```

To use another version of the code, you can check out a specific tag or a branch.

### Check out a specific NorESM branch, eg NorESM2.0.1

List all available tags:

```
> git tag --list
```

To check out a specific tag, use **git checkout <tag-name>** where tag-name is a tag for the list, for instance release-noresm2.0.1:

```
> git checkout release-noresm2.0.1
```

List all available branches:

```
> git branch --all
```

To check out a specific branch, for instance noresm2:

```
> git checkout -b noresm2 origin/noresm2
```

You can now inspect which tag or branch you are using by invoking the **git branch** command again. You can also inspect the commits log by invoking the **git log** command (to for instance only see the 3 commits, apply the **-n 3** option).

## Manage externals

Then you need to launch the download:

```
./manage_externals/checkout_externals [this will take one to a few minutes ...]
```

this will use the repositories, tags, branches as specified in Externals.cfg (see below for its manipulation)

**If you run into several SVN-related errors when launching the model, you may want to try to change `required=True` to `required=False` for `pop2` and `ww3` in `Externals.cfg`. `POP2` and `WW3` are not needed in `NorESM2`. Then try again.**

**To confirm a successful download of all components**, you can run `checkout_externals` with the status flag `-S` to show the status of the externals or `--logging` to get a log of reported errors (if any):

```
./manage_externals/checkout_externals -S [-S shows status of externals]
./manage_externals/checkout_externals --logging [write log of errors in manage_
↪externals.log]
```

The `checkout_externals` script will read the configuration file called `Externals.cfg` and will download all the external component models and CIME into `/path/to/<noresm-base>`.

Now you have a complete copy of the NorESM code in the directory `<noresm-base>`. Now you can to the subdirectory `cime/scripts` and start creating a case! (see *Experiments*)

## Configure Externals.cfg

The file can be modified to use another repo, fork, branch or tag or release for any of the model components.

See more info here: <https://github.com/ESCOMP/CESM/blob/master/README.rst>

## 1.2.2 Downloading inputdata

Input datasets are needed to run the model. We don't recommend downloading the entire dataset because of the size (~1TB). Input datasets needed for a specific case and configuration will be automatically downloaded when a user submits a case. We recommend to only have one input directory on a machine which is shared for all users.

The input data needs to be stored in a local directory on the machine the model is build and run. The path to the local directory is set in `<noresm-base>/cime/config/cesm/machines/config_machines.xml`

```
<DIN_LOC_ROOT>/cluster/shared/noresm/inputdata</DIN_LOC_ROOT>
```

The main download location for input data is <https://noresm.org/inputdata/> from where the downloading request might be redirected to another location.

### NorESM specific inputdata

The recipe to download the complete NorESM2 code is based on how it is done for CESM. For more details please see [https://escomp.github.io/CESM/release-cesm2/downloading\\_cesm.html](https://escomp.github.io/CESM/release-cesm2/downloading_cesm.html)

The input data sets are downloaded from multiple servers. The servers and download protocols used are listed in

```
<noresm-base>/cime/config/cesm/config_inputdata.xml
```

### SVN problems during downloading

If the user encounter problems using svn, e.g. that the CESM2 files from ucar.edu are not downloaded automatically when submitting a case, there are several workarounds:

- 1. On some machines you need to first download one file to permanently add the certificate. If the certificate is not issued by a trusted authority, you need to use the fingerprint to validate the certificate manually. This is done by explicit download one file with svn such that the user can choose “(p) permanently” to add the certificate. One example file:

```
svn export https://svn-ccsm-inputdata.cgd.ucar.edu/trunk/inputdata/atm/waccm/lb/LBC_
↪1750-2014_CMIP6_0p5degLat_c170126.nc
```

or

- 2. Put the CESM2 inputdata (with svn protocol) before the noresm.org in cime/config/cesm/config\_inputdata.xml, then the downloading happens in one go.

or

- 3. In the case folder, run `./check_inputdata -download` twice

or

- 4. Submit the case twice i.e run `./case.submit`

## 1.3 Running NorESM2

### 1.3.1 Newbies guide

This guide provides basic instructions on how set up and run a standard NorESM case by executing 4 steps:

- create a new case (the `create_newcase` script)
- configure case (the `case_setup` script)
- build case (the `case_build` script)
- submit case (the `case_submit` script).

It is assumed that you have successfully downloaded the model (see [Downloading the model code](#)), which means you have a copy of the model on your computer in a folder with a name of your choice. For simplicity we call this folder `<noresm-base>` in this guide.

## Create a new case

The `create_newcase` script is an executable python script located in:

```
<noresm-base>/cime/scripts/
```

The script for creating a new case takes several command line arguments as input to know how to configure your case. Some of the most important arguments are as follows:

- `-case` defines a casename of your choice and created a folder by that name in `<noresm-base>/cases/`.
- `-mach` defines the machine you will run the model on. The model NorESM2 has been configured to be run on a set of different machines (see list at [Running on different platforms](#)). If you are running the model on a machine not listed you will need to configure the model beyond this newbie guide.
- `-res` defines the resolution of your run. See [Experiments](#) for more details.
- `-compset` defines what compset you will be using. A list of compsets for fully-coupled configurations can be found in the file `<noresm_base>/cime_config/config_compsets.xml` (see [Atmosphere](#) for compsets for AMIP-type simulations)

To investigate the full list of arguments, enter the `<noresm_base>/cime/scripts` folder and run `create_newcase` with the `-help` argument:

```
cd <noresm_base>/cime/scripts/
./create_newcase --help
```

To create a new case, enter the scripts directory and run the `create_newcase` scripts:

```
cd <noresm_base>/cime/scripts/
./create_newcase --case ../../cases/$CASENAME --mach $MACHINE --res f19_g16 --compset
↪ $COMPSET
```

You have now created the case folder `<noresm_base>/cases/$CASENAME!` Go to the case folder to start configuring your experiment.

## More advanced examples

The following example creates a case (test1910\_1) on the machine Fram:

```
./create_newcase --case ../../cases/test1910_1 --compset N1850 --res f19_tn14 --
↪ machine fram --project snic2019-1-2 --user-mods-dir cmip6_noresm_DECK
```

Here we use the N1850 compset, which configures the case as a 1850 pre-industrial control simulation. Note that we use the argument `-run-unsupported`, which required if the grid resolution is not supported in the compset (see [Experiments](#)), `-project` to set the id of the project used in the batch system accounting on Fram, `-user-mods-dir` to set the path to a folder containing files that will further configure your case (like user namelists, shell scripts with `xmlchange` commands or `SourceMods`).

The following example creates a case (also called test1910\_1), but on the machine Tetralith::

```
./create_newcase --case ../../cases/test1910_1 --walltime 24:00:00 --compset N1850 --
↪ res f19_tn14 --machine tetralith --project snic2019-1-2 --output-root /proj/bolinc/
↪ users/${USER}/NorESM2/noresm2_out
```

Note that here we use the argument `-output-root`, which is only required if the `noresm_run_dir` (the running directory of the mode) differs from default running directory `<path_to_run_dir>/noresm/`

### Configure the case

The case folder `<noresm_base>/cases/$CASENAME/` is where you configure your case by changing environment files (such as the `<noresm_base>/cases/$CASENAME/env_run.xml` file; see `experiment_environments`), changing the user namelists for the different model components (files named `user_nl_$COMP` where `$COMP` is a model component such as `cam`), or even add your own code changes to `SourceMods/src.$COMP/`. But for now we stick to the standard out-of-the-box set up and configure the case as follows:

```
cd <noresm_base>/cases/$CASENAME
./case.setup
```

### Build the case

After your configuration is finished you can start building your case by invoking the `case.build` script from your case folder:

```
./case.build
```

Which may take a while.

### Submit your case

When your case has finished building you are ready to submit and run your case. This is done by invoking the `case.submit` script from your case folder:

```
./case.submit
```

If your model simulation was successful, you should find the following line in `slurm.out` (or similar) in your case folder

```
Tue Feb 9 21:41:33 CET 2016 -- CSM EXECUTION BEGINS HERE Wed Feb 10 13:37:56 CET 2016
↔-- CSM EXECUTION HAS FINISHED
(seq_mct_drv) : ===== SUCCESSFUL TERMINATION OF CPL7-CCSM =====
```

And you are finished!

## 1.3.2 Experiments

NorESM is part of the CESM family of earth system models and shares a lot of the configuration options with CESM. Many of the simulation configuration settings are defined by the so called compsets.

For a quick-start guide on how to create, configure, build, and submit a NorESM experiment, see the *Newbies guide*. More details are provided below, for the more advanced users.

## Create and configure a new case

To start a new experiment you need to create and configure a case. After running the

```
./create_newcase
```

script, a case folder `<path_to_case_dir>/<casename>` is created that contains set-up files for your experiment. Then, after running the

```
./case.setup
```

script, several other files and directories needed to build the case are created, including the user user namelists files.

The `create_newcase` script includes a `--compset` option. A compset, or component set, is a collection of predefined setting that defines your experiment set-up, including which model components that should be activated. Some of the available compsets are described below.

The case folder contains predefined namelist (with namelist settings partly depending on compset option). The default namelist options for the case can be overwritten by changing/adding the new namelist options in the `user_nl_<component>` file.

Several configuration options are available in the usermods directories in `<noresm_base>/cime_config/usermods_dirs/`. These folders contain information about output variables and frequencies from `clm` (land) and `cam` (atmosphere). In addition one SourceMod is included in `SourceMods/src.cam/preprocessorDefinitions.h` to define if `AEROFFL` and `AEROCOM` are included for extra aerosol diagnostics (for more details about the aerosol diagnostics see ??)

Remember that the amount of diagnostics and the output frequency have a huge impact on both the run time and storage.

```
--user-mods-dir cmip6_noresm_*
```

```
cmip6_noresm_DECK (AEROFFL)
cmip6_noresm_hifreq (high frequency output, AEROFFL)
cmip6_noresm_hifreq_xaer (high frequency output, AEROFFL and AEROCOM)
cmip6_noresm_xaer (AEROFFL and AEROCOM)
```

For more details about the user-mod-dir options, check this folder

```
<noresm_base>/cime_config/usermods_dirs
```

## The xmlchange and xmlquery scripts

The `xmlchange` and `xmlquery` scripts are located in your case folder and lets you change or query the contents of variables in the `evn_*.xml` files without entering the files. There are two advantages of using `xmlchange` to edit the xml files rather than doing by hand: (1) the `xmlchange` script checks that the new setting is valid and (2) the change is echoed to the `CaseStatus` file, thus automatically documented. To change from the default `ndays` to `nmonths`

```
./xmlchange STOP_OPTION=nmonths
```

It's also possible to change several variables at once, for instance

```
./xmlchange STOP_OPTION=nmonths,STOP_N=14
```

See the header of `xmlchange` and `xmlquery` for more details and examples.

## Create a clone case

To create clone cases from a control case can be very useful for e.g. sensitivity studies. If you want to make a copy of a case (i.e. identical `./create_newcase` command and identical `env_*.xml`, `user_nml_<component>` and `SourceMods` files) that can be done by the use of `./create_clone`. You only need to give the casename of the new case and the casename of the case which should be cloned (copied). The case will have identical set up (`env_*.xml`, `user_nml_<component>` files and `SourceMods`) as the clone, but these files can of course be modified before building the case.

## Compsets

Compsets, or component sets, specify which component models will be used in your simulation along with which forcing files, and even which physics options to use. Each compset has a long name (lname) and an alias. For instance N1850 is the alias for the NorESM compset for pre-industrial (1850) conditions. The long name for N1850 is

```
1850_CAM60%NORESM_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC
↪%BDRDDMS
```

The long name generally follows the notation

```
TIME_ATM[%phys]_LND[%phys]_ICE[%phys]_OCN[%phys]_ROF[%phys]_GLC[%phys]_WAV[%phys] [_ESP
↪%phys] [_BGC%phys]
```

(see the help section of the file `<noresm_base>/cime_config/config_compsets.xml` for details). The compsets can also include information on which grids are scientifically supported (see below for details).

All predefined compsets for **coupled simulations** can be found in

```
<noresm_base>/cime_config/config_compsets.xml
```

Predefined compsets for **AMIP-type (atmosphere/land-only) simulations** can be found in

```
<noresm_base>/components/cam/cime_config/config_compsets.xml
```

Predefined compsets for running the sea-ice model as a stand-alone model can be found in

```
<noresm_base>/components/cice/cime_config/config_compsets.xml
```

Predefined compsets for running the land model as a stand-alone model can be found in

```
<noresm_base>/components/clm/cime_config/config_compsets.xml
```

Predefined compsets for running the ocean model as a stand-alone model can be found in

```
<noresm_base>/components/blom/cime_config/config_compsets.xml
```

The compsets starting with N are NorESM coupled configurations. Compsets starting with NF are NorESM AMIP (atmosphere only) configurations. Some examples are given below.

**N1850 and N1850frc2** Coupled configuration for NorESM for pre-industrial (1850) conditions.

**NHIST and NHISTfrc2** Historical configuration from 1850 up to year 2015 (see detailed description below; ‘Create your own compsets for AMIP simulations’)

**NSSP126frc2, NSSP245frc2, NSSP370frc2, NSSP585frc2** Future scenario compsets from 2015 to 2100

**NFHISTnorppdmsbc** AMIP simulation with time-evolving prescribed observed values for SSTs and sea ice and upper-ocean DMS values derived from a fully coupled NorESM2 simulation for present-day conditions



**frc2 emission files** The frc2 option uses differently organized emission files. The frc2 files are located in

```
<PATH_TO_INPUTDATA>/noresm/inputdata/atm/cam/chem/emis/cmip6_emissions_
↪version20190808
```

A new set of emission files have been made to avoid the occurrence of random mid-month model crashes. These crashes are related to the reading of emission files, but are still under investigation. To use the newest emission files choose compsets including *frc2* or if you want to create a new compset add

```
%FRC2
```

to NORESM2. For a detailed description, see **Creating your own compset** below.

For an overview of the compsets provided for CESM2, please see: <http://www.cesm.ucar.edu/models/cesm2/config/compsets.html>.

### Supported grids

Most compsets contain an entries listing which which grid(s) are scientifically supported for that compset

```
<science_support grid="xxx"/> fields
```

When a compset has a scientifically-supported grid, you can create a new case (with the `create_newcase` script) without having to use the option `--run-unsupported`. If the compset does not list any scientifically-supported grids, or if you want to use a grid configuration is not included in the definition of the compset, the

```
--run-unsupported
```

option is required when a case is created or the `create_newcase` script will fail.

### Creating your own compset

The essential file to edit for a new coupled NorESM compset is

```
<noresm_base>/cime_config/config_compsets.xml
```

and for a new AMIP NorESM compset is

```
<noresm_base>/components/cam/cime_config/config_compsets.xml
```

### Coupled simulation

This examples shows how to simply add the “N1850frc2” compset to `config_compsets.xml`. In `<noresm_base>/cime_config/config_compsets.xml` the N1850frc2 is set as

```
<compset>
  <alias>N1850frc2</alias>
  <lname>1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_
↪SWAV_BGC%BDRDDMS</lname>
</compset>
```

where

`<alias>COMPSETNAME</alias>` sets the compsets name used when building a new case. Make sure to use a new and unique compset name. The details of the compset i.e. which models components and component-specific configurations to use are set in

```
<lname>1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_  
↔SWAV_BGC%BDRDDMS</lname>
```

It is also possible to just add that line (without the <lname>) when creating a new case.

'\_' separates between model components

```
_  
<MODEL>
```

and '%' sets the component-specific configuration

```
%MODEL_CONFIGURATION
```

E.g.

- **1850\_CAM60%NORESM%FRC2**
  - Forcing and input files read from pre-industrial conditions (1850). If you need a historical run replace 1850 with HIST
  - Build CAM6.0 (the atmosphere model) with NorESM configuration and FRC2 organized emission files
- **CLM50%BGC-CROP**
  - Build CLM5 (land model) with Biogeochemistry and prognotic crop package
- **CICE%NORESM-CMIP6**
  - Build CICE (sea-ice model) with NorESM2-CMIP6 setup
- **BLOM%ECO**
  - Build BLOM (ocean model) including iHAMOCC biogeochemistry model
- **MOSART**
  - Build MOSART (river runoff model) with default configurations
- **SGLC\_SWAV**
  - The SGLC (land-ice) and SWAV (ocean-wave) models are not interactive, but used only to satisfy the interface requirements
- **BGC%BDRDDMS**
  - ocean biogeochemistry model iHAMOCC run with interactive DMS

### AMIP simulation

For details about AMIP simulation compsets, please see [Atmosphere](#)

### Building the case

The case is built by

```
./case.build
```

All user modifications to `env_run.xml`, `env_mach_pes.xml`, `env_batch.xml` must be done before `case.build` is invoked. This is also the case for the aforementioned user-made namelists: i.e. `user_nl_cam`, `user_nl_cice`, `user_nl_clm`, `user_nl_blom`, `user_nl_cpl`).

If you want to ensure your case is ready for submission, you can run

```
./check_case
```

which will:

- Ensure that all of the env xml files are in sync with the locked files
- Create namelists (thus verifying that there will be no problems with namelist generation)
- Ensure that the build is complete

Running this is completely optional: these checks will be done automatically when running case.submit. However, you can run this if you want to perform these checks without actually submitting the case.

As a last step, remember to copy restart files to run directory if you are running a branch run or a hybrid run.

### Submitting the case

The case is submitted by

```
./case.submit
```

### Resolution

Model resolution is set when the case is created. Below some common resolutions are listed. A complete list of model grids can be found here:

```
<noresm_base>/cime/config/cesm/config_grids.xml
```

### Atmospheric grids

```
f19_f19 - atm lnd 1.9x2.5  
f09_f09 - atm lnd 0.9x1.25  
f09_f09_mg17
```

### Ocean grids

Currently, BLOM supports three resolutions, nominal 2,1, and 1/4 degrees in a tripolar grid configuration:

```
tnxlv4 - tripolar ocn ice 1-degree grid  
tnx2v1 - tripolar ocn ice 2-degree grid  
tx0.25v4 - tripolar ocn ice 1/4-degree grid
```

## Coupled

```
f19_tn14 - atm lnd 1.9x2.5, ocnice tn1v4 [CMIP6 grid, NorESM2-LM]
f09_tn14 - atm lnd 0.9x1.25, ocnice tn1v4 [CMIP6 grid, NorESM2-MM]
f09_tn0254 - atm lnd 0.9x1.25, ocnice tn0.25v4
```

## Forcing

Please see *Input data sets*

## Choosing output

please see *NorESM2 output*

## Setting up a nudged simulation

please see *Nudged experiments*

### 1.3.3 Experiment environments

After creating a case (see *Experiments*) the environment settings can be modified in the `env_*.xml` files and the `user_nl_<component>` files contained in the case folder

#### The case folder contains:

- **README.case:** File detailing your create\_newcase usage. This is a good place for you to keep track of runtime problems and changes. The file contains information about e.g. how the case was created, compset details, grid information and which branch, git commit and model code were used for case creation.
- **CaseStatus:** File containing a list of operations done in the current case.
- **Buildconf:** Directory containing scripts to generate component namelists and component and utility libraries (e.g., PIO, MCT). *You should never have to edit the contents of this directory*
- **SourceMods:** Directory where you can place modified source code. In SourceMods there are subfolders for the different models; cam, clm, cice, blom, mosart and so on . If you want to change the code or add subroutines, you place copies of the fortran files here.
- **user made namelists:** you can place your own namelists for the different models where you can change parameters and model settings and so on (i.e. `user_nl_cam`, `user_nl_cice`, `user_nl_clm`, `user_nl_blom`, `user_nl_cpl`). See details below.
- **CaseDocs:** here you find the namelists containing all the subroutines and parameters used. These files will be modified after rebuild. The details of parameter values and input files are listed in the `<component>_in` files. *You should never have to edit the contents of this directory*. If you wish to make changes to the `<component>_in` files, you change the `user_nl_<component>` and rebuild.
- **LockedFiles:** Directory that holds copies of files that should not be changed. *You should never edit the contents of this directory*
- **Tools:** Directory containing support utility scripts.

## Machine specific environment

The file

```
env_mach_pes.xml
```

sets component machine-specific processor layout. The settings are critical to a well-load-balanced simulation. Here you set the number of cpus used for each model component and the coupler (usually land + ice + rof (river run off) = atm = coupler). An example of such an env\_mach\_pes.xml file:

```
<entry id="NTASKS">
  <type>integer</type>
  <values>
    <value compclass="ATM">768</value>
    <value compclass="CPL">768</value>
    <value compclass="OCN">186</value>
    <value compclass="WAV">300</value>
    <value compclass="GLC">768</value>
    <value compclass="ICE">504</value>
    <value compclass="ROF">8</value>
    <value compclass="LND">256</value>
    <value compclass="ESP">1</value>
  </values>
  <desc>number of tasks for each component</desc>
</entry>
```

## Run environment

The file

```
env_run.xml
```

sets the configuration details for the experiment. E.g. time settings such as length of run, frequency of restart files produced, output of coupler diagnostics, and short-term and long-term restart file archiving.

## NorESM2 specific configuration settings

- FLUXSCHEME=1
  - In the coupled NorESM2 simulations, the flux parameterization used for the transfer of heat, moisture and momentum between the ocean and atmosphere is the so-called COARE flux parameterization. This choice is activated by FLUXSCHEME=1 in envrun.xml, and ends up in the driver\_in namelist as fluxscheme=1. This parameterisation is different from the standard flux-parameterization used in CESM, which is activated by FLUXSCHEME=0.
- ALBCOSZAVG=.true.
  - a feature of the coupled NorESM2 simulations, i.e., taking into account the fact that the solar zenith angle used for the calculation of the surface albedo changes over the atmospheric model time step of 30 minutes

## Some common configuration settings

- **RUN\_TYPE:**
  - **startup:** In a startup run (the default), all components are initialized using baseline states. These baseline states are set independently by each component and can include the use of restart files, initial files, external observed data files, or internal initialization (i.e., a cold start). Settings of initial files to be read are set in the `user_nl_<component>`
  - **branch:** In a branch run, all components are initialized using a consistent set of restart files from a previous run (determined by the `RUN_REFCASE` and `RUN_REFDATE` variables in `env_run.xml`). The case name is generally changed for a branch run, although it does not have to be. In a branch run, setting `RUN_STARTDATE` is ignored because the model components obtain the start date from their restart datasets. Therefore, the start date cannot be changed for a branch run. `RUN_REFCASE` and `RUN_REFDATE` are required for branch runs. To set up a branch run, locate the restart tar file or restart directory for `RUN_REFCASE` and `RUN_REFDATE` from a previous run, then place those files in the `RUNDIR` directory.
  - **hybrid:** Not as strict as branch. In a hybrid run the model is initialized as a startup, BUT uses initialization datasets from a previous case. This is somewhat analogous to a branch run with relaxed restart constraints. A hybrid run allows users to bring together combinations of initial/restart files from a previous case (specified by `RUN_REFCASE`) at a given model output date (specified by `RUN_REFDATE`). Unlike a branch run, the starting date of a hybrid run (specified by `RUN_STARTDATE`) can be modified relative to the reference case.
- **RUN\_REFCASE:**
  - Reference data used for hybrid or branch runs. The name of the reference simulation the model components are initialized from. The restart and rpointer files should be copied to the run directory before the job is submitted
- **RUN\_REFDATE:**
  - The reference date of the restart files from the simulation set in `RUN_REFCASE`
- **RUN\_STARTDATE:**
  - Set the date (of your own wish) for the beginning of the simulation
- **STOP\_OPTION:**
  - Sets the run length along with `STOP_N`. Can choose between e.g.: none, never, nstep, nhours, ndays, nday, nmonths, nyears, date.
- **STOP\_N:**
  - Provides a numerical count for `$STOP_OPTION`. E.g. if `STOP_OPTION` is set to years and `STOP_N` set to 20, the model will run for 20 years.
- **REST\_OPTION:**
  - Sets the frequency of model restart files output (same options as `STOP_OPTION`)
- **REST\_N:**
  - Provides a numerical count for `$REST_OPTION`. E.g. if `REST_OPTION` is set to years and `STOP_N` set to 5, the model will produce restart files every 5 years.
- **CONTINUE\_RUN:**
  - Needs to be `FALSE` when you first begin the run. When you successfully run and get a restart file (if the model crashes after the restart file is produced you can set `CONTINUE_RUN` to `TRUE` as well), you will need to change `CONTINUE_RUN` to `TRUE` for the remainder of your simulation.

- RESUBMIT:
  - If RESUBMIT is greater than 0, then case will automatically resubmit. Enables the model to automatically resubmit a new run. This is very useful for long simulations. E.g. RESUBMIT is set to 2 and the simulation length is set to 20 years (STOP\_OPTION is years and STOP\_N is 20), the total length of the simulation will be 60 years.
- RESUBMIT\_SETS\_CONTINUE\_RUN:
  - Needs to be TRUE (default) is the RESUBMIT flag causes a resubmission of the case
- DOUT\_S\_SAVE\_INTERIM\_RESTART\_FILES:
  - Logical to archive all the produced restart files and not just those at the end of the simulation. Default is FALSE.

## Setting up a hybrid simulation

Step by step guide for hybrid simulation/restart.

When the case is created and compiled, edit `env_run.xml`. Below is an example for restart with CMIP6 historical initial conditions:

```
<entry id="RUN_TYPE" value="hybrid">
<entry id="RUN_REFDIR" value="path/to/restarts"                # path to restarts
<entry id="RUN_REFCASE" value="NHISTfrc2_f09_tn14_20191025"> # experiment name
↳for restart files
<entry id="RUN_REFDATE" value="2015-01-01">                 # date of restart
↳files
<entry id="RUN_STARTDATE" value="2015-01-01">              # date in simulation
<entry id="GET_REFCASE" value="TRUE">                       # get refcase from
↳outside rundir
```

If it is not possible to link directly to restarts, copy the restart files and rpointer files to the run directory. Below is example changes to `env_run.xml`:

```
<entry id="RUN_TYPE" value="hybrid">
<entry id="RUN_REFCASE" value="NHISTfrc2_f09_tn14_20191025"> # Experiment name
↳for restart files
<entry id="RUN_REFDATE" value="2015-01-01">                 # date of restart
↳files
<entry id="RUN_STARTDATE" value="2015-01-01">              # date in simulation
<entry id="GET_REFCASE" value="FALSE">                      # get refcase from
↳outside rundir
```

## User namelists

### Frequency of output

There are plenty of default output variables which are automatically written,

The variables specified in the namelists will be written as output automatically, but if you need to customize the output fields you can edit the `user_nl_<component>` lists

E.g. if you edit `user_nl_cam` and add the following lines at the end of the file:

```
avgflag_pertape = 'A','I'
nhtfrq = 0 ,-6
mfilt = 1 , 30
ndens = 2 , 2
fincl1 = 'FSN200','FSN200C','FLN200',
'FLN200C','QFLX','PRECTMX:X','TREFMXAV:X','TREFMNAV:M',
'TSMN:M','TSMX:X'
fincl2 = 'T','Z3','U','V','PSL','PS','TS','PHIS'
```

- avgflag\_pertape

Sets the averaging flag for all variables on a particular history file series. Default is to use default averaging flags for each variable. Average (A), Instantaneous (I), Maximum (X), and Minimum (M).

- nhtfrq

Array of write frequencies for each history files series.

- nhtfrq = 0, the file will be a monthly average. Only the first file series may be a monthly average.
- nhtfrq > 0, frequency is input as number of timesteps.
- nhtfrq < 0, frequency is input as number of hours.

- mfilt

Array of number of time samples to write to each history file series (a time sample is the history output from a given timestep)

- nden

Array specifying output format for each history file series. Valid values are 1 or 2. '1' implies output real values are 8-byte and '2' implies output real values are 4-byte. Default: 2

- fincl1

List of fields to add to the primary history file.

- fincl2

List of fields to add to the auxiliary history file.

For a detailed description of NorESM2 output, please see [NorESM2 output](#)

## Parameter settings

If you need to change some variable values or activate/deactive flags, that can also be done in user\_nl\_<component>. The syntax is:

```
&namelist_group
  namelist_var = new_namelist_value
```

E.g for a quadrupling of the atmospheric CO2 concentration

```
&chem_surfvals_nl
  co2vmr      = 1137.28e-6
```

Note that BLOM uses a different syntax than the rest. In user\_nl\_bloom:

```
set BDMC2      = .15
set NIWGF      = .5
```



you need to include `set` before the name of the variable and it does not matter what namelist group the variable belong.

## Input data

All active and data components use input data sets. A local disk needs `DIN_LOC_ROOT` to be populated with input data in order to run NorESM. You can make links to the input data sets in the `user_nl_<components>`. Input data is handled by the build process as follows:

- The `buildnml` scripts in `Buildconf/` create listings of required component input datasets in the `Buildconf/<component>.input_data_list` files
- `./case.build` checks for the presence of the required input data files in the root directory `DIN_LOC_ROOT`. If all required data sets are found on local disk, then the build can proceed.
- If any of the required input data sets are not found, the build script will abort and the files that are missing will be listed. At this point, you must obtain the required data from the input data server using `check_input_data` with the `-export` option.

## Aerosol diagnostics

The model can be set up to take out AeroCom-specific output, effective forcing estimates, and other additional aerosol output. See *Aerosol diagnostics and output* for details.

## COSP

NorESM2 can be run with the CFMIP Observation Simulator Package (COSP) to calculate model cloud diagnostics that can be directly compared with satellite observations from ISCCP, CloudSat, CALIOP, MISR, and MODIS. Please see *COSP output* for details.

## Code modifications

If you want to make more substantial changes to the codes than what is possible by the use of `user_nl_<component>`, you need to copy the source code (the fortran file you want to modify) to the `SourceMods/src.<component>` folder in the case directory, then make the modifications needed before building the model. Make sure that you use the source code from the same commit as you used to create the case (for commit details see `README.case` in the case folder). **Do not change the source code in the `<noresm-base>` folder!**

## Run and archiving time environment

The file

```
env_batch.xml
```

sets the batch file or job script configurations. You need to specify two jobs; one for running the model (`case.run`) and one for moving the files from the `RUNDIR` to the archive directory (`case.st_archive`). The archiving is usually very fast (less than one hour), but for very large jobs (high resolution or large output) it can take several hours.

You can edit the xml files directly to change the variable values. Although you can edit this at any time, build environment variables should not be edited after a build is invoked.

There are also other `env_` files which you usually don't need to or can't change:

- `env_mach_specific.xml`: File used to set a number of machine-specific environment variables for building and/or running set in `<noresm-base>/cime/config/cesm/machines/config_machines.xml`.

- `env_case.xml`: Sets case specific variables (e.g. model components, model and case root directories) and *cannot be modified after a case has been created*. To make changes, you need to re-run `./create_newcase.sh` in `<noresm-base>/cime/scripts/` with different options.

### 1.3.4 Running on different platforms

#### HPC platforms

Below is a list of platforms where NorESM2 has been installed, including platform specific instructions.

`<noresm-base>`: the name of the folder where the model code has been downloaded (cloned from git)

#### Vilje @ Sigma2

Configuration files for running NorESM2 on Vilje are distributed in all branches of the noresm code.

Input data is stored in `/work/shared/noresm/inputdata/`

Apply for membership in NorESM shared data storage (manager: [mben@norceresearch.no](mailto:mben@norceresearch.no)) for access to the folder.

The run and archive directories are stored in `/work/<user_name>/`

Before configuring and compiling the model, add this code

```
module ()
{
  eval `/usr/bin/modulecmd bash $*`
}
module load intelcomp/15.0.1 mpt/2.10 python/2.7.9 netcdf/4.3.2
export PATH=$PATH:/opt/pbs/default/bin/
```

to your `.bashrc`

```
cd
vi .bashrc
```

Create a new case:

```
./create_newcase -case ../../../../cases/<casename> --mach vilje --res <resolution> --
↪compset <compset_name> --project <project_name> --user-mods-dir <user_mods_dir> --
↪run-unsupported
```

#### Fram @ Sigma2

Configuration files for running NorESM2 on Fram are distributed in all branches of the noresm code.

Input data is stored in `/cluster/shared/noresm/inputdata/`

Apply for membership in NorESM shared data storage (manager: [mben@norceresearch.no](mailto:mben@norceresearch.no)) for access to the folder.

The run and archive directories are stored `/cluster/work/users/<user_name>/`

Create a new case:

```
./create_newcase -case ../../../../cases/<casename> --mach fram --res <resolution> --
↪compset <compset_name> --project <project_name> --user-mods-dir <user_mods_dir> --
↪run-unsupported
```

## Queue options on Fram

On fram there are different queues for testing and development experiments (usually short runs on few nodes) and longer experiments. If you want to run simulations using different queue options than *normal*, you can add new machine options to `<noresm-base>/cime/config/cesm/machines/config_batch.xml`. Method (we are currently working on an improvement of this):

- 1. Copy the settings for Fram :

```
<batch_system MACH="fram" type="slurm">
  <batch_submit>sbatch</batch_submit>
  <submit_args>
    <arg flag="--time" name="$JOB_WALLCLOCK_TIME"/>
    <arg flag="-p" name="$JOB_QUEUE"/>
    <arg flag="--account" name="$PROJECT"/>
  </submit_args>
  <directives>
    <directive> --ntasks={{ total_tasks }}</directive>
    <directive> --export=ALL</directive>
    <directive> --switches=1</directive>
  </directives>
  <queues>
    <queue walltimemax="00:59:00" nodemin="1" nodemax="288" default="true">normal</
→queue>
  </queues>
</batch_system>
```

:: ‘

- 2. Change “fram” to “fram\_devel” or “fram\_short”
- 3. Change the line

```
<queue walltimemax="00:59:00" nodemin="1" nodemax="288" default="true">normal</queue>
```

to:

```
<queue walltimemax="00:30:00" nodemin="1" nodemax="4" default="true">devel</queue>
```

for the development queue and

```
<queue walltimemax="02:00:00" nodemin="1" nodemax="10" default="true">short</queue>
```

for the short queue.

- 4. Add to the directives:

```
<directive> -qos=devel</directive>
```

for the development queue and

```
<directive> --qos=short</directive>
```

for the short queue option.

**You need to make one config\_batch setting for each queue option. You also need to add fram\_devel and fram\_short to config\_machines.xml and config\_compilers.xml** To do so, you just copy-paste the settings for fram and change the name *fram* to *fram\_devel* and/or *fram\_short* in the files `config_machines.xml` and `config_compilers.xml`. Hopefullt this will improve very soon, but the good thing is that you only need to do it once.

The resulting `<noresm-base>/cime/config/cesm/machines/config_batch.xml`. file:

```

<batch_system MACH="fram" type="slurm">
  <batch_submit>SBATCH</batch_submit>
  <submit_args>
    <arg flag="--time" name="$JOB_WALLCLOCK_TIME"/>
    <arg flag="-p" name="$JOB_QUEUE"/>
    <arg flag="--account" name="$PROJECT"/>
  </submit_args>
  <directives>
    <directive> --ntasks={{ total_tasks }}</directive>
    <directive> --export=ALL</directive>
    <directive> --switches=1</directive>
  </directives>
  <queues>
    <queue walltimemax="00:59:00" nodemin="1" nodemax="288" default="true">normal</
→queue>
  </queues>
</batch_system>

<batch_system MACH="fram_devel" type="slurm">
  <batch_submit>SBATCH</batch_submit>
  <submit_args>
    <arg flag="--time" name="$JOB_WALLCLOCK_TIME"/>
    <arg flag="-p" name="$JOB_QUEUE"/>
    <arg flag="--account" name="$PROJECT"/>
  </submit_args>
  <directives>
    <directive> --ntasks={{ total_tasks }}</directive>
    <directive> --export=ALL</directive>
    <directive> --switches=1</directive>
    <directive> --qos=devel</directive>
  </directives>
  <queues>
    <queue walltimemax="00:30:00" nodemin="1" nodemax="4" default="true">devel</queue>
  </queues>
</batch_system>

<batch_system MACH="fram_short" type="slurm">
  <batch_submit>SBATCH</batch_submit>
  <submit_args>
    <arg flag="--time" name="$JOB_WALLCLOCK_TIME"/>
    <arg flag="-p" name="$JOB_QUEUE"/>
    <arg flag="--account" name="$PROJECT"/>
  </submit_args>
  <directives>
    <directive> --ntasks={{ total_tasks }}</directive>
    <directive> --export=ALL</directive>
    <directive> --switches=1</directive>
    <directive> --qos=short</directive>
  </directives>
  <queues>
    <queue walltimemax="02:00:00" nodemin="1" nodemax="10" default="true">short</
→queue>
  </queues>
</batch_system>

```

After, you can use the new machine settings when creating a new case: For the development queue::

```
./create_newcase -case ../../../../cases/<casename> --mach fram_devel --res <resolution> --
↳--compset <compset_name> --project <project_name> --user-mods-dir <user_mods_dir> --
↳run-unsupported
```

and for the short queue:

```
./create_newcase -case ../../../../cases/<casename> --mach fram_short --res <resolution> --
↳--compset <compset_name> --project <project_name> --user-mods-dir <user_mods_dir> --
↳run-unsupported
```

For a detailed guide on how to set up, submit and choosing the right job see:

<https://documentation.sigma2.no/jobs/submitting.html>

[https://documentation.sigma2.no/jobs/choosing\\_job\\_types.html](https://documentation.sigma2.no/jobs/choosing_job_types.html)

## Nebula @ NSC

Configuration files for running NorESM2 on Nebula are distributed in the release tags `release-noresm2*` and in the `noresm2 origin/noresm2` branch. If Nebula configurations are missing in your copy of the model, the files can be found in the following folder on Nebula:

```
/nobackup/forsk/noresm/nebula_config_noresm2/
```

Apply for membership in NorESM shared data storage (manager: [adag@met.no](mailto:adag@met.no)) for access to the folder.

Copy the files in the above folder to:

```
cd <noresm-base>/cime/config/cesm/machines/
cp /nobackup/forsk/noresm/nebula_config_noresm2/* .
```

Input data is stored in `/nobackup/forsk/noresm/inputdata/`

The run and archive directories are stored `/nobackup/forsk/<user_name>/`

Before configuring and compiling the model, add `export LMOD_QUIET=1` to your `.bashrc`

```
cd
vi .bashrc
```

Create a new case:

```
./create_newcase -case ../../../../cases/<casename> --mach nebula --res <resolution> --
↳compset <compset_name> --project <project_name> --user-mods-dir <user_mods_dir> --
↳run-unsupported
```

### Tetralith @ NSC

Configuration files for running NorESM2 on Tetralith are distributed in the featureCESM2.1.0-OsloDevelopment branch. If Tetralith configurations are missing in your copy of the model the files can be found in the following folder on Tetralith:

```
/proj/cesm_input-data/tetralith_config_noresm2
```

Apply for membership in CESM climate model shared data storage (SNIC 2019/32-10) for access to the folder.

Copy the files in the above folder to:

```
cd <noresm-base>/cime/config/cesm/machines/  
cp /proj/cesm_input-data/tetralith_config_noresm2/* .
```

Input data is stored /proj/cesm\_input-data/

Before configuring and compiling the model, clear your environment and load the following modules:

```
module purge  
module load buildenv-intel/2018.u1-bare  
module load netCDF/4.4.1.1-HDF5-1.8.19-nscl-intel-2018a-eb  
module load HDF5/1.8.19-nscl-intel-2018a-eb  
module load PnetCDF/1.8.1-nscl-intel-2018a-eb
```

Create a new case:

```
./create_newcase -case ../../../../cases/<casename> -mach triolith -res <resolution> -  
→compset <compset_name> -pecount M -ccsm_out <NorESM_output_folder>
```

### Virtual Machine with Conda (@ <https://www.nrec.no/> for example)

This section describes how to install all the software environment (including compilers and libraries) needed to run CESM/NorESM on a Virtual Machine (like those available on the Norwegian Research and Education Cloud, the Google Cloud Platform, etc.), but a similar process allows to run the model(s) on a personal computer, laptop or desktop running **Centos7** (this distribution is convenient to use since it already contains most of the essential software packages).

The objective here is not to compete against HPCs in terms of sheer computing power, but to satisfy the everyday needs of the vast majority of CESM/NorESM developers in terms of model development, debugging or testing, as well as for training/teaching purposes.

For this example we start with a completely empty machine with the Centos7 Linux Distribution, 16x Intel Core Processors (Haswell, no TSX), 128GB RAM, and a 100GiB volume (disk) attached on **/dev/sdb**.

The name of the user is **centos** (if your user name is different you will have to use your *username* instead).

The first step is to format the volume (if your disk is already formatted and/or contains data, skip this step, but still create the **/opt/uio** folder since this is where the model(s) are configured to read/write).

```
sudo mkfs.ext4 /dev/sdb
```

then mount it at /opt/uio

```
sudo mkdir /opt/uio  
sudo chown -R centos /opt/uio
```

(continues on next page)

(continued from previous page)

```
sudo chgrp -R centos /opt/uio  
sudo mount /dev/sdb /opt/uio  
cd /opt/uio
```

and create the following folders:

```
mkdir /opt/uio/inputdata  
mkdir /opt/uio/work  
mkdir /opt/uio/archive  
mkdir /opt/uio/archive/cases
```

Now we can install a few packages which will be needed later (to get the model(s), etc.) and miniconda (accept the terms of the license and accept the default location **/home/centos/miniconda3**, then answer yes to the question “*Do you wish the installer to initialize Miniconda3 by running conda init*”, exit the virtual machine and re-login).

```
sudo yum install wget git subversion csh -y  
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh  
bash Miniconda3-latest-Linux-x86_64.sh  
exit
```

You will notice the next time you login the Virtual Machine that the prompt starts with *(base)* which indicates that you are in the base conda environment (since you accepted it during the miniconda install).

We recommend to create a new **esm** conda environment before adding the **bioconda** and **conda-forge** channels (in this order) and installing **cesm**

```
conda create -n esm  
conda activate esm  
conda config --add channels bioconda  
conda config --add channels conda-forge  
conda install cesm=2.1.3
```

The prompt should start with *(esm)* indicating that the esm conda environment has been activated, and every time you login you will have to type **conda activate esm** to be able to run the model(s).

This will have installed CESM2.1.3 as well as all the necessary compilers and libraries (HDF5, NetCDF, MKL, etc.) and their dependencies, and the very same environment can be used with NorESM.

In order to run the model(s) you still need configuration files (namely *config*, *config\_machines.xml* and *config\_compilers.xml*). These will eventually come with NorESM, but for the sake of convenience we provide hereafter an example of such files which have to be located in a **.cime** folder in your home directory (simply copy & past the content of the following cell to generate the files automatically and be carefull not to add any odd characters or lines since CESM/NorESM are extremely picky about it).

Notice that you only need to do this once, since both CESM and NorESM will use these configurations, and that the name of the machine created is **espresso**.

```

cd /home/centos

mkdir .cime

cd .cime

cat >> config << EOF
[main]
CIME_MODEL=cesm
EOF

cat >> config_machines.xml << EOF
<?xml version="1.0"?>
<config_machines>
  <machine MACH="espresso">
    <DESC> Virtual Machine with 16 VCPUs and 128GiB memory
      OS is Centos7, Conda CESM environment
    </DESC>
    <NODENAME_REGEX>UNSET</NODENAME_REGEX>
    <OS>LINUX</OS>
    <PROXY>UNSET</PROXY>
    <COMPILERS>gnu</COMPILERS>
    <MPILIBS>mpich</MPILIBS>
    <SAVE_TIMING_DIR>UNSET</SAVE_TIMING_DIR>
    <CIME_OUTPUT_ROOT>/opt/uo/work</CIME_OUTPUT_ROOT>
    <DIN_LOC_ROOT>/opt/uo/inputdata</DIN_LOC_ROOT>
    <DIN_LOC_ROOT_CLMFORC>UNSET</DIN_LOC_ROOT_CLMFORC>
    <DOUT_S_ROOT>/opt/uo/archive/$CASE</DOUT_S_ROOT>
    <BASELINE_ROOT>UNSET</BASELINE_ROOT>
    <CCSM_CPRNC>UNSET</CCSM_CPRNC>
    <GMAKE_J>16</GMAKE_J>
    <BATCH_SYSTEM>none</BATCH_SYSTEM>
    <SUPPORTED_BY>noresmCommunity</SUPPORTED_BY>
    <MAX_TASKS_PER_NODE>16</MAX_TASKS_PER_NODE>
    <MAX_MPITASKS_PER_NODE>16</MAX_MPITASKS_PER_NODE>
    <PROJECT_REQUIRED>FALSE</PROJECT_REQUIRED>
    <mpirun mpilib="default">
      <executable>mpiexec</executable>
      <arguments>
        <arg name="anum_tasks"> -np $TOTALPES</arg>
      </arguments>
    </mpirun>
    <module_system type="none"/>
    <environment_variables>
      <env name="KMP_STACKSIZE">64M</env>
    </environment_variables>
    <resource_limits>
      <resource name="RLIMIT_STACK">-1</resource>
    </resource_limits>
  </machine>
</config_machines>
EOF

cat >> config_compilers.xml << EOF
<?xml version="1.0"?>

```

(continues on next page)



(continued from previous page)

```

<config_compilers version="2.0">
  <compiler MACH="espresso">
    <LD>mpifort</LD>
    <AR>x86_64-conda_cos6-linux-gnu-ar</AR>
    <SFC>x86_64-conda_cos6-linux-gnu-gfortran</SFC>
    <SCC>x86_64-conda_cos6-linux-gnu-cc</SCC>
    <SCXX>x86_64-conda_cos6-linux-gnu-c++</SCXX>
    <MPIFC>mpifort</MPIFC>
    <MPICC>mpicc</MPICC>
    <MPICXX>mpicxx</MPICXX>
    <NETCDF_PATH>/home/centos/miniconda3/envs/esm</NETCDF_PATH>
    <FFLAGS>
      <append DEBUG="FALSE"> -O2 </append>
      <append MODEL="blom"> -fdefault-real-8 </append>
      <append MODEL="cam"> -finit-local-zero </append>
    </FFLAGS>
    <SLIBS>
      <append> -L$(NETCDF_PATH)/lib -lnetcdff -lnetcdf -ldl </append>
      <append> -lmkl_gf_lp64 -lmkl_gnu_thread -lmkl_core -lomp -lpthread -lm </append>
    </SLIBS>
  </compiler>
</config_compilers>
EOF

```

To create a new CESM case F2000climo at resolution f19\_g17 and run it for **1 day**, and because (for CESM only) *create\_newcase* has been added to the *PATH*, simply type (from anywhere on the machine):

```

create_newcase --case /opt/uio/archive/cases/conda_CESM213_F2000climo_f19_g17 --
→compset F2000climo --res f19_g17 --machine espresso --run-unsupported

cd /opt/uio/archive/cases/conda_CESM213_F2000climo_f19_g17

NUMNODES=-1

./xmlchange --file env_mach_pes.xml --id NTASKS --val ${NUMNODES}

./xmlchange --file env_mach_pes.xml --id NTASKS_ESP --val 1

./xmlchange --file env_mach_pes.xml --id ROOTPE --val 0

./xmlchange STOP_N=1

./xmlchange STOP_OPTION=ndays

./case.setup

./case.build

./case.submit

```

Hopefully this should create the case, configure it, compile it (for this particular machine the compilation time is less than 3 minutes) and run it (starting with the download of the necessary input files the first time you run it).

For NorESM, first clone the github repository, here in `/opt/uio/noresm2`, as follows (be careful: you have to be in the **(base)** conda environment for that):

```
cd /opt/uio

git clone -b noresm2 https://github.com/NorESMhub/NorESM.git noresm2

cd noresm2/

sed -i.bak "s/'checkout'/'checkout', '--trust-server-cert'/" ./manage_externals/manic/
↪ repository_svn.py

./manage_externals/checkout_externals -v
```

To create a “similar” NorESM case NF2000climo at resolution f19\_tn14 and also run it for **1 day**, and after having activated the (**esm**) environment (if you are not already in it), do:

```
cd /opt/uio/noresm2/cime/scripts

./create_newcase --case /opt/uio/archive/cases/conda_NorESM_NF2000climo_f19_tn14 --
↪ compset NF2000climo --res f19_tn14 --machine espresso --run-unsupported

cd /opt/uio/archive/cases/conda_NorESM_NF2000climo_f19_tn14

NUMNODES=-1

./xmlchange --file env_mach_pes.xml --id NTASKS --val ${NUMNODES}

./xmlchange --file env_mach_pes.xml --id NTASKS_ESP --val 1

./xmlchange --file env_mach_pes.xml --id ROOTPE --val 0

./xmlchange STOP_N=1

./xmlchange STOP_OPTION=ndays

./case.setup

./case.build

./case.submit
```

On our machine the compilation takes less than 3 minutes, and if everything went well the input files should download automatically before the run starts.

Note: AeroTab is supposed to be a folder, if a file has been created instead simply add a “trailing slash” (/) at line 1946 in components/cam/bld/namelist\_files/namelist\_defaults\_cam.xml (which has to be written as: <aerotab\_table\_dir>noresm-only/atm/cam/camoslo/AeroTab\_8jun17/</aerotab\_table\_dir>) and resubmit.

### Adding a new platform

Edit the following files:

```
config_batch.xml
config_compilers.xml
config_machines.xml
```

located in

```
<noresm-base>/cime/config/cesm/machines/
```

## config\_batch.xml

Add a batch\_system entry in this file for your platform with appropriate settings. See examples below.

### Machine example with SLURM batch system

on Fram:

```
<batch_system MACH="fram" type="slurm">
  <batch_submit>sbatch</batch_submit>
  <submit_args>
    <arg flag="--time" name="$JOB_WALLCLOCK_TIME"/>
    <arg flag="-p" name="$JOB_QUEUE"/>
    <arg flag="--account" name="$PROJECT"/>
  </submit_args>
  <directives>
    <directive> --ntasks={{ total_tasks }}</directive>
    <directive> --export=ALL</directive>
    <directive> --switches=1</directive>
  </directives>
  <queues>
    <queue walltimemax="00:59:00" nodemin="1" nodemax="288" default="true">normal</
    ↪queue>
  </queues>
</batch_system>
```

On Tetralith:

```
<batch_system type="slurm" MACH="tetralith">
  <batch_submit>sbatch</batch_submit>
  <submit_args>
    <arg flag="--time" name="$JOB_WALLCLOCK_TIME"/>
    <arg flag="--account" name="$PROJECT"/>
  </submit_args>
  <queues>
    <queue walltimemax="168:00:00" nodemin="1" default="true">default</queue>
    <queue walltimemax="01:00:00" nodemin="1" nodemax="4" >development</queue>
  </queues>
</batch_system>
```

### Machine example with PBS batch system

```
<batch_system MACH="vilje" type="pbs">
  <submit_args>
    <arg flag="-N cesmRun"/>
  </submit_args>
  <directives>
    <directive>-A nn2345k</directive>
    <directive>-l select={{ num_nodes }}:ncpus={{ MAX_TASKS_PER_NODE }}:mpiprocs={{
    ↪tasks_per_node }}:omphreads={{ thread_count }}</directive>
  </directives>
  <queues>
    <queue walltimemax="00:59:00" nodemin="1" nodemax="9999" default="true">workq</
    ↪queue>
```

(continues on next page)

(continued from previous page)

```

</queues>
<!--walltimes>
                                <walltime default="true">00:59:00</walltime>
</walltimes-->
</batch_system>

```

## config\_compilers.xml

Add a compiler entry in this file for your platform with appropriate settings. See examples below.

On Fram:

```

<compiler MACH="fram">
  <CPPDEFS>
    <append> -D$(OS) </append>
  </CPPDEFS>
  <FFLAGS>
    <append> -xCORE-AVX2 -no-fma </append>
  </FFLAGS>
  <NETCDF_PATH>$(EBROOTNETCDFMINFORTRAN) </NETCDF_PATH>
  <PNETCDF_PATH>$(EBROOTPNETCDF) </PNETCDF_PATH>
  <MPI_PATH>$(MPI_ROOT) </MPI_PATH>
  <MPI_LIB_NAME>mpi </MPI_LIB_NAME>
  <FFLAGS>
    <append DEBUG="FALSE"> -O2 </append>
    <append MODEL="blom"> -r8 </append>
    <append MODEL="cam"> -init=zero,arrays </append>
  </FFLAGS>
  <MPICC> mpiicc </MPICC>
  <MPICXX> mpiicpc </MPICXX>
  <MPIFC> mpiifort </MPIFC>
  <PIO_FILESYSTEM_HINTS>lustre </PIO_FILESYSTEM_HINTS>
  <SLIBS>
    <append>-mkl=sequential -lnetcdf -lnetcdf </append>
  </SLIBS>
</compiler>

```

On Tetralith:

```

<compiler MACH="tetralith" COMPILER="intel">
  <MPICC> mpiicc </MPICC>
  <MPICXX> mpiicpc </MPICXX>
  <MPIFC> mpiifort </MPIFC>
  <PNETCDF_PATH>${ENV{PNETCDF_DIR}} </PNETCDF_PATH>
  <NETCDF_PATH>${ENV{NETCDF_DIR}} </NETCDF_PATH>
  <SLIBS>
    <append>-L$(NETCDF_PATH)/lib -lnetcdf -lnetcdf </append>
  </SLIBS>
  <FFLAGS>
    <append> -xHost -fPIC -mcmmodel=large </append>
  </FFLAGS>
  <FFLAGS>
    <append DEBUG="FALSE"> -O0 -xAVX </append>
    <append MODEL="blom"> -r8 </append>
  </FFLAGS>

```

(continues on next page)

(continued from previous page)

```

<CFLAGS>
  <append> -xHost -fPIC -mmodel=large </append>
</CFLAGS>
<LDFLAGS>
  <append> -mkl </append>
</LDFLAGS>
</compiler>

```

## config\_machines.xml

Add a machine entry in this file for your platform with appropriate settings. See examples below.

On Fram:

```

<machine MACH="fram">
  <DESC>Lenovo NeXtScale M5, 32-way nodes, dual 16-core Xeon E5-2683@2.10GHz, 64 GiB,
  ↪per node, os is Linux, batch system      is SLURM</DESC>
  <OS>LINUX</OS>
  <COMPILERS>intel</COMPILERS>
  <MPILIBS>impi</MPILIBS>
  <CIME_OUTPUT_ROOT>/cluster/work/users/$USER/noesm</CIME_OUTPUT_ROOT>
  <DIN_LOC_ROOT>/cluster/shared/noesm/inputdata</DIN_LOC_ROOT>
  <DIN_LOC_ROOT_CLMFORC>UNSET</DIN_LOC_ROOT_CLMFORC>
  <DOUT_S_ROOT>/cluster/work/users/$USER/archive/$CASE</DOUT_S_ROOT>
  <DOUT_L_ROOT>/projects/NS2345K/noesm/cases</DOUT_L_ROOT>
  <DOUT_L_HOSTNAME>login.nird.sigma2.no</DOUT_L_HOSTNAME>
  <!--DOUT_L_MSROOT>UNSET</DOUT_L_MSROOT-->
  <BASELINE_ROOT>UNSET</BASELINE_ROOT>
  <CCSM_CPRNC>UNSET</CCSM_CPRNC>
  <GMAKE_J>8</GMAKE_J>
  <BATCH_SYSTEM>slurm</BATCH_SYSTEM>
  <SUPPORTED_BY>noesmCommunity</SUPPORTED_BY>
  <MAX_TASKS_PER_NODE>32</MAX_TASKS_PER_NODE>
  <MAX_MPITASKS_PER_NODE>32</MAX_MPITASKS_PER_NODE>
  <PROJECT_REQUIRED>TRUE</PROJECT_REQUIRED>
  <mpirun mpilib="mpi-serial">
    <executable></executable>
  </mpirun>
  <mpirun mpilib="default">
    <executable>mpirun</executable>
  </mpirun>
  <module_system type="module">
    <init_path lang="perl">/cluster/software/lmod/lmod/init/perl</init_path>
    <init_path lang="python">/cluster/software/lmod/lmod/init/env_modules_python.py</
  ↪init_path>
    <init_path lang="csh">/cluster/software/lmod/lmod/init/csh</init_path>
    <init_path lang="sh">/cluster/software/lmod/lmod/init/sh</init_path>
    <cmd_path lang="perl">/cluster/software/lmod/lmod/libexec/lmod perl</cmd_path>
    <cmd_path lang="python">/cluster/software/lmod/lmod/libexec/lmod python</cmd_path>
    <cmd_path lang="sh">module</cmd_path>
    <cmd_path lang="csh">module</cmd_path>
    <modules>
      <command name="purge">--force</command>
      <command name="load">StdEnv</command>
      <!-- djlo Deactivated THT settings -->

```

(continues on next page)

(continued from previous page)

```

<!--command name="load">intel/2016a</command-->
<!--command name="load">netCDF-Fortran/4.4.3-intel-2016a</command-->
<!--command name="load">PnetCDF/1.8.1-intel-2016a</command-->
<!--command name="load">CMake/3.5.2-intel-2016a</command-->
<command name="load">intel/2018a</command>
<command name="load">netCDF-Fortran/4.4.4-intel-2018a-HDF5-1.8.19</command>
<command name="load">PnetCDF/1.8.1-intel-2018a</command>
<command name="load">CMake/3.9.1</command>
</modules>
</module_system>
<environment_variables>
  <env name="KMP_STACKSIZE">64M</env>
  <env name="I_MPI_EXTRA_FILESYSTEM_LIST">lustre</env>
  <env name="I_MPI_EXTRA_FILESYSTEM">on</env>
</environment_variables>
<resource_limits>
  <resource name="RLIMIT_STACK">-1</resource>
</resource_limits>
</machine>

```

**On Tetralith:**

```

<machine MACH="tetralith">
  <DESC>Tetralith Linux Cluster (NSC, Sweden), 32 pes/node, batch system SLURM</DESC>
  <OS>LINUX</OS>
  <COMPILERS>intel</COMPILERS>
  <MPILIBS>impi</MPILIBS>
  <PROJECT>snic2019-1-2</PROJECT>
  <CHARGE_ACCOUNT>bolinc</CHARGE_ACCOUNT>
  <CIME_OUTPUT_ROOT>/proj/$CHARGE_ACCOUNT/users/$ENV{USER}/noesm2</CIME_OUTPUT_ROOT>
  <DIN_LOC_ROOT>/proj/cesm_input-data/inputdata/</DIN_LOC_ROOT>
  <DIN_LOC_ROOT_CLMFORC>/proj/cesm_input-data/inputdata/atm/dtm7</DIN_LOC_ROOT_
  ↪CLMFORC>
  <DOUT_S_ROOT>$CIME_OUTPUT_ROOT/cesm_archive/$CASE</DOUT_S_ROOT>
  <BASELINE_ROOT>$CIME_OUTPUT_ROOT/cesm_baselines</BASELINE_ROOT>
  <CCSM_CPRNC>/$CIME_OUTPUT_ROOT/cesm_tools/cprnc/cprnc</CCSM_CPRNC>
  <GMAKE_J>4</GMAKE_J>
  <BATCH_SYSTEM>slurm</BATCH_SYSTEM>
  <SUPPORTED_BY>snic</SUPPORTED_BY>
  <MAX_TASKS_PER_NODE>32</MAX_TASKS_PER_NODE>
  <MAX_MPITASKS_PER_NODE>32</MAX_MPITASKS_PER_NODE>
  <PROJECT_REQUIRED>TRUE</PROJECT_REQUIRED>
  <mpirun mpilib="default">
    <executable>mpprun</executable>
  </mpirun>
  <module_system type="none">
  </module_system>
</machine>

```

### 1.3.5 Input data sets

The complete input data set is stored on Fram @ Sigma2. For access contact [mben@norceresearch.no](mailto:mben@norceresearch.no).

#### Atmospheric specific input data

Various external input data sets are used by the atmosphere model. These data sets often describe “boundary conditions” needed during the integration of the model (solar forcing, GHG concentrations, ozone concentrations, oxidant concentrations, emissions of short-lived species, production rate of H<sub>2</sub>O from CH<sub>4</sub> oxidation, stratospheric aerosol, upper ocean chlorophyll-a concentration, sea-surface temperature and sea-ice concentration). The boundary conditions can be constant, cyclic (often one annual cycle), or evolve according to a historical or future scenario.

- **Solar forcing** The solar forcing is prescribed following Matthes et al. (2017).
- **GHG concentrations : CO<sub>2</sub>, CH<sub>4</sub>, N<sub>2</sub>O, CFC1-eq, CFC12** Green-house concentrations are used in the radiative transfer calculations, and the values are taken from Meinshausen et al. (2017). The prescribed concentrations are assumed to represent the surface concentrations. Latitudinal and species dependent profiles are used to calculate the concentrations at different altitudes.
- **Ozone concentrations** Ozone concentration climatologies are used in the radiative transfer calculations. The ozone climatologies used in NorESM2 have a 5-day frequency, vary in height and latitude, but have no longitudinal dependence. They are based on simulations with CESM2-WACCM (Danabasoglu et al., 2019).
- **Oxidant concentrations : OH, ozone, NO<sub>3</sub>-radical, HO<sub>2</sub>** The oxidant concentrations are used in the description of secondary aerosol formation (sulfate and SOA). OH, ozone, NO<sub>3</sub> and H<sub>2</sub>O<sub>2</sub> (formed by HO<sub>2</sub>) are relevant for the oxidation of DMS, SO<sub>2</sub>, isoprene, and monoterpenes. These climatologies have a monthly frequency, and a 3-dimensional spatial distribution. They are based on simulations with CESM2-WACCM (Danabasoglu et al., 2019).
- **Emissions of short-lived species : BC, OM and SO<sub>2</sub>** Emissions of BC, OM and SO<sub>2</sub> consists in contributions from anthropogenic sources, biomass burning, and continuous outgassing of SO<sub>2</sub> by volcanoes. The anthropogenic sources (Hoesly et al., 2018) are given as 9 different sectors : emissions of agricultural activity, transport, domestic heating, solvents, waste and shipping are emitted at the surface; emissions from the energy and industrial sectors are emitted between 150 and 350 m height; aircraft emissions can go up to around 15 km. The biomass burning sources (van Marle et al., 2017) are give as 6 different categories, and emission heights differ among them : 0-100 m for agricultural waste burning and peat burning, 0-1 km for savannah burning and deforestation, 0-2 km for temperate forests, and 0-3 km for Boreal forests (Dentener et al., 2006). A fixed climatology of continuous tropospheric outgassing of SO<sub>2</sub> by volcanoes is also included (Dentener et al., 2006). Emission inventories provide emission strengths for OC. For fossil fuel combustions we assumed an OM:OC ratio of 1.4, and for biomass burning of 2.6. Part of the SO<sub>2</sub> (2.5% equivalent S) is emitted as SO<sub>4</sub>. Emission fields are provided both on the 1.9x2.5 and 0.9x1.25 horizontal grid.
- **Water vapour emissions (from methane oxidation)** Production of H<sub>2</sub>O from methane oxidation is prescribed using a climatologies based on simulations with CESM2-WACCM. This source is especially relevant in the stratosphere. These climatologies have a monthly frequency, and a 3-dimensional spatial distribution. They are based on simulations with CESM2-WACCM (Danabasoglu et al., 2019).
- **Stratospheric aerosol** To describe the impact of volcanic SO<sub>2</sub> emissions reaching the stratosphere and forming SO<sub>4</sub> aerosol, monthly varying climatologies of stratospheric aerosol properties are used.
- **Upper-ocean POM concentrations** To describe the emission strength of marine primary organic matter from the ocean, a monthly varying climatology of upper-ocean chlorophyll-a concentration is used.
- **Sea-surface temperature and sea-ice cover** In atmosphere-only simulations, the sea-surface temperature and sea-ice concentrations are prescribed as monthly varying climatologies. One can use climatologies that are based on observations (Hurrell et al., 2008), or climatologies derived from fully-coupled NorESM2 simulations.

## Aerosol specific input data

Some of the input data, the look-up tables (LUT) for NorESM specific aerosol optics and size information for calculation of cloud droplet activation, can be modified either for testing purposes or in order to take into account new developments in the aerosol microphysics scheme. Some typical examples of input that may need to be updated are: refractive indices; assumed (log-normal) size parameters at the point of emission or production; assumed hygroscopicities for sub-saturated conditions. Such changes can be made in the offline “sectional” aerosol module AeroTab (as in the example of new refractive indices), or both in AeroTab and in the online aerosol module OsloAero in the CAM6-Nor code (as in the example of assumed size parameters). Many aerosol related model changes may be done without having to touch the AeroTab code and the LUT at all, such as e.g. the emissions (whether they are prescribed or interactive).

A user’s guide for the AeroTab code, with some additional information about OsloAero code (in CAM6-Nor) which makes use of the AeroTab LUT, can be found at [https://github.com/NorESMhub/NorESM/blob/noresm2/doc/configurations/AeroTab-user-guide\\_v16april2020.pdf](https://github.com/NorESMhub/NorESM/blob/noresm2/doc/configurations/AeroTab-user-guide_v16april2020.pdf). This AeroTab presentation <https://github.com/NorESMhub/NorESM/blob/noresm2/doc/configurations/AeroTab-slides-updateJan2020.pdf> may be useful as a first introduction. For questions about AeroTab, contact Alf Kirkevåg (alfk at met.no) or Øyvind Seland (oyvind at met.no).

## Ocean specific input data

In case of a startup run (i.e. if the model is not re-started from a previous simulation) the ocean is initialized from rest, and the initial ocean temperature and salinity are from the Polar Science Center Hydrographic Climatology (PHC) 3.0, updated from Steele et al. (2001). The initial condition files containing ocean temperature and salinity are located in the directory

```
/DIN_LOC_ROOT/ocn/blom/inicon/inicon_<gridspec>_<date>.nc,
```

where DIN\_LOC\_ROOT is the base input data directory (depends on the machine; on fram it is /cluster/shared/noresm/inputdata), *gridspec* specifies the ocean grid used, and *date* specifies a date tag for the file. The files contains values for layered potential density (sigma), potential temperature (temp), salinity (saln) and layer thickness (dz):

```
dimensions:
  x = 360 ;
  y = 385 ;
  z = 53 ;
variables:
  double sigma(z, y, x) ;
  double temp(z, y, x) ;
  double saln(z, y, x) ;
  double dz(z, y, x) ;
```

Boundary conditions for the ocean component (e.g. tidal dissipation, SSS climatologies for OMIP configuration) are located in

```
/DIN_LOC_ROOT/ocn/blom/bndcon/,
```

and grid specific information (grid input file, files defining ocean basins and sections) are located in

```
/DIN_LOC_ROOT/ocn/blom/grid/.
```



## Ocean carbon cycle specific input data

The ocean carbon cycle in NorESM2 (iHAMOCC) is initialized from gridded observation based data sets for DIC, alkalinity, phosphate, nitrate, oxygen, and silica. These data sets have been provided by CMIP6-OMIP (Orr et al. 2017), and are located in the same directory as the BLOM initial conditions.

Further, iHAMOCC requires three input data sets specifying boundary conditions: 1) monthly climatological dust deposition based on Mahowald et al. (2006), 2) riverine inputs, which contain an annual climatology (normalized to year 2000) of fluxes of organic and inorganic carbon and nutrient constituents based on the Global-NEWS2 model and other datasets (Mayorga et al., 2010; Hartmann, 2009; Chester, 1990), and 3) atmospheric nitrogen deposition, provided through the CMIP6 protocol in monthly deposition fields of wet or dry and oxidized or reduced nitrogen deposition rates, all of which are added to the nitrate pool in the top-most ocean layer.

By default, these external inputs are activated, but the user can choose not to include riverine and nitrogen deposition by setting BLOM\_RIVER\_NUTRIENTS and BLOM\_N\_DEPOSITION to FALSE in `env_run.xml`.

While the initial conditions are interpolated by the model (using nearest neighbor interpolation), the boundary condition datasets need to be pre-interpolated to the ocean grid used. These data sets are available for 2, 1, and 1/4 degree resolution (the `tnx2v1`, `tnx1v4`, and `tnx0.25v4` grids). Note however, that for running CMIP scenario simulations, specific N-deposition data sets are necessary. These might not be available for a given grid, so they may need to be created and tested.

## Adding new inputfiles

All BLOM/iHAMOCC input file names are specified via `namelist` (including the full path name). If a user would like to use a different input file, it is recommended to place this file in the user's work directory, and specify the corresponding file name (including the full path) as a `namelist` option in `user_nl_biom` (see *Ocean and Sea-Ice*).

## References

- Chester, R.: Marine Geochemistry, 1st ed., 702p, Springer, Netherlands, 1990.
- Danabasoglu, G., Lamarque, J.-F., Bacmeister, J., Bailey, D. A., DuVivier, A. K., Edwards, J., Emmons, L. K., Fasullo, J., Garcia, R., Gettelman, A., Hannay, C., Holland, M. M., Large, W. G., Lawrence, D. M., Lenaerts, J. T. M., Lindsay, K., Lipscomb, W. H., Mills, M. J., Neale, R., Oleson, K. W., Otto-Bliesner, B., Phillips, A. S., Sacks, W., Tilmes, S., van Kampenhout, L., Vertenstein, M., Bertini, A., Dennis, J., Deser, C., Fischer, C., Fox-Kemper, B., Kay, J. E., Kinnison, D., Kushner, P. J., Long, M. C., Mickelson, S., Moore, J. K., Nienhouse, E., Polvani, L., Rasch, P. J., and Strand, W. G.: The Community Earth System Model version 2 (CESM2), Submitted to *J. Adv. Model. Earth Syst.*, 2019.
- Dentener, F., Kinne, S., Bond, T., Boucher, O., Cofala, J., Generoso, S., Ginoux, P., Gong, S., Hoelzemann, J. J., Ito, A., Marelli, L., Penner, J. E., Putaud, J.-P., Textor, C., Schulz, M., van der Werf, G. R., and Wilson, J.: Emissions of primary aerosol and precursor gases in the years 2000 and 1750 prescribed data-sets for AeroCom, *Atmospheric Chemistry and Physics*, 6, 4321–4344, <https://doi.org/10.5194/acp-6-4321-2006>, 2006.
- Hartmann, J.: Bicarbonate-fluxes and CO<sub>2</sub>-consumption by chemical weathering on the Japanese Archipelago – Application of a multi- lithological model framework, *Chemical Geology*, 265, 237–271, 2009.
- Hoesly, R. M., Smith, S. J., Feng, L., Klimont, Z., Janssens-Maenhout, G., Pitkanen, T., Seibert, J. J., Vu, L., Andres, R. J., Bolt, R. M., Bond, T. C., Dawidowski, L., Kholod, N., Kurokawa, J.-I., Li, M., Liu, L., Lu, Z., Moura, M. C. P., O'Rourke, P. R., and Zhang, Q.: Historical (1750–2014) anthropogenic emissions of reactive gases and aerosols from the Community Emissions Data System (CEDS), *Geoscientific Model Development*, 11, 369–408, <https://doi.org/10.5194/gmd-11-369-2018>, 2018.
- Hurrell, J.W., J.J. Hack, D. Shea, J.M. Caron, and J. Rosinski: A New Sea Surface Temperature and Sea Ice Boundary Dataset for the Community Atmosphere Model. *J. Climate*, 21, 5145–5153, <https://doi.org/10.1175/2008JCLI2292.1>, 2008.

Mahowald, N., Baker, A., Bergametti, G., Brooks, N., Duce, R., Jickells, T., Kubilay, N., Prospero, J., and Tegen, I.: Atmospheric global dust cycle and iron inputs to the ocean, *Global Biogeochem. Cycles*, 19, 4025, <https://doi.org/10.1029/2004GB002402>, 2005.

Matthes, K., Funke, B., Andersson, M. E., Barnard, L., Beer, J., Charbonneau, P., Clilverd, M. A., Dudok de Wit, T., Haberreiter, M., Hendry, A., Jackman, C. H., Kretzschmar, M., Kruschke, T., Kunze, M., Langematz, U., Marsh, D. R., Maycock, A. C., Misios, S., Rodger, C. J., Scaife, A. A., Seppälä, A., Shangguan, M., Sinnhuber, M., Tourpali, K., Usoskin, I., van de Kamp, M., Verronen, P. T., and Versick, S.: Solar forcing for CMIP6 (v3.2), *Geoscientific Model Development*, 10, 2247–2302, <https://doi.org/10.5194/gmd-10-2247-2017>, 2017.

Mayorga, E., Seitzinger, S. P., Harrison, J. A., Dumont, E., Beusen, A. H. W., Bouwman, A. F., Fekete, B. M., Kroeze, C., and Van Drecht, G.: Global Nutrient Export from WaterSheds 2 (NEWS 2): Model development and implementation, *Environmental Modelling and Software*, 25, 837–853, 2010.

Meinshausen, M., Vogel, E., Nauels, A., Lorbacher, K., Meinshausen, N., Etheridge, D. M., Fraser, P. J., Montzka, S. A., Rayner, P. J., Trudinger, C. M., Krummel, P. B., Beyerle, U., Canadell, J. G., Daniel, J. S., Enting, I. G., Law, R. M., Lunder, C. R., O’Doherty, S., Prinn, R. G., Reimann, S., Rubino, M., Velders, G. J. M., Vollmer, M. K., Wang, R. H. J., and Weiss, R.: Historical greenhouse gas concentrations for climate modelling (CMIP6), *Geoscientific Model Development*, 10, 2057–2116, <https://doi.org/10.5194/gmd-10-2057-2017>, 2017.

Orr, J. C., Najjar, R. G., Aumont, O., Bopp, L., Bullister, J. L., Danabasoglu, G., Doney, S. C., Dunne, J. P., Dutay, J.-C., Graven, H., Griffies, S. M., John, J. G., Joos, F., Levin, I., Lindsay, K., Matear, R. J., McKinley, G. A., Mouchet, A., Oschlies, A., Romanou, A., Schlitzer, R., Tagliabue, A., Tanhua, T., and Yool, A.: Biogeochemical protocols and diagnostics for the CMIP6 Ocean Model Intercomparison Project (OMIP), *Geosci. Model Dev.*, 10, 2169–2199, <https://doi.org/10.5194/gmd-10-2169-2017>, 2017.

van Marle, M. J. E., Kloster, S., Magi, B. I., Marlon, J. R., Daniau, A.-L., Field, R. D., Arneth, A., Forrest, M., Hantson, S., Kehrwald, N. M., Knorr, W., Lasslop, G., Li, F., Mangeon, S., Yue, C., Kaiser, J. W., and van der Werf, G. R.: Historic global biomass burning emissions for CMIP6 (BB4CMIP) based on merging satellite observations with proxies and fire models (1750–2015), *Geoscientific Model Development*, 10, 3329–3357, <https://doi.org/10.5194/gmd-10-3329-2017>, 2017.

## 1.3.6 Atmosphere

### CAM6-Nor

In NorESM2, the atmospheric model CAM6-Nor replaces standard CAM. CAM6-Nor differs from CAM6 in several ways. Some of the changes are modifications to existing files (scripts, xml-files, F90-files), some code changes are made by copying original files in a different directory and modifying them, and some changes are really new code. In the F90 files, some of the changes are activated by #CAMOSLO . Below the main differences in the code are described:

#### Atmospheric chemistry/aerosol/cloud module: OsloAero6

A different aerosol scheme is included in CAM6-Nor. The code for this scheme can be mainly be found in the following directories under <noresm-base>:

- components/cam/src/chemistry/pp\_trop\_mam\_oslo [contains code generated by a preprocessor]
- components/cam/src/chemistry/oslo\_aero [contains a lot of code which originally resides in other cam-directories]
- components/cam/src/physics/cam\_oslo
- components/cam/NorESM/
- components/cam/NorESM/fv

### CAM

Some modifications in CAM (which are not related to the CAM-Oslo aerosol scheme) are implemented in the original code, but with namelist choices such that the modifications can be activated/deactivated:

- Parameterization of turbulent air-sea fluxes (see AMIP-type experiments for more details)
- Averaging over (changing) zenith angle during one model time step
- Improved conservation of energy and angular momentum

In addition some parameter settings, emission and input files will differ from standard CAM6 set-up.

### CAM6-Nor source directories

The different source directories are included/activated in the following way:

1. In components/cam/cime\_config/config\_component.xml:

If the long compset name contains:

```
_CAM60%NORESM
```

or:

```
_CAM60%PTAERO
```

then CAM\_CONFIG\_OPTS will contain “-chem trop\_mam\_oslo”.

2. In components/cam/bld/configure:

If chem contains the string “\_oslo”, the following two directories are added as source:

```
cam/src/chemistry/oslo_aero
cam/src/physics/cam_oslo
```

The directory cam/src/chemistry/pp\_\$chem\_pkg (in our case thus pp\_trop\_mam\_oslo) will be added as source.

The following two directories are always added [this should in principle not be the case for a pure CESM compset, but that switch hasn't been built in yet]:

```
cam/src/NorESM
cam/src/NorESM/$dyn (in our case is thus often cam/src/NorESM/fv)
```

## Initial conditions

### Startup runs

If your experiment is a **startup** run, the atmosphere is initialized using basestate files. The full pathname of the basestate file is given in the cam namelist variable **ncdata** which is set by build-namelist. For NorESM2-LM, the default basestate file is located in the inputdata folder (on Fram this is /cluster/shared/noresm/inputdata/):

```
atm/cam/inic/fv/cami-mam3_0000-01-01_1.9x2.5_L32_c150407.nc
```

For NorESM2-MM, the default basestate file is:

```
atm/cam/inic/fv/cami-mam3_0000-01-01_0.9x1.25_L32_c141031.nc
```

See the following file in your NorESM2 repository for a full list of basestate files for different configurations:

```
<noresm>/components/cam/bld/namelist_files/namelist_defaults_cam.
```

Alternatively, as NorESM2 and CESM2 has the same default files, the different default basestate files can be inspected by looking at the entry for **ncdata** in the CESM2 documentation for CAM6.0 Fortran Namelist Definitions: [http://www.cesm.ucar.edu/models/cesm2/settings/current/cam\\_nml.html](http://www.cesm.ucar.edu/models/cesm2/settings/current/cam_nml.html)

### Branch runs

If your experiment is a **branch** run, the atmosphere is initialized using the restart files from a previous run. The restart file is determined by the variables `RUN_REFCASE` and `RUN_REFDATE` in `env_run.xml`. The full pathname of the restart file is given by the cam namelist variable **cam\_branch\_file**. This variable does not have a default value.

### Hybrid runs

If your experiment is a **hybrid** run, the different model components are initialized as if it was a startup, but using initialization files from a previous case at a given date (again set by the `RUN_REFCASE` and `RUN_REFDATE` variables). CAM is initialized using a initial-condition file from the previous case. The full pathname of the initial-condition file will be set in the cam namelist variable **ncdata**.

## Setting up an AMIP-type experiment

Atmospheric Model Intercomparison Project (AMIP) style runs are runs in which the **atmosphere and land components are active while values for sea surface temperatures and sea ice are prescribed** (that is, read from a file). The sea-ice model CICE then runs in a simplified mode and computes surface fluxes, snow depth, albedo, and surface temperatures using 1D thermodynamics without conserving energy. The sea-ice thickness is assumed to be 2 m in the Northern Hemisphere and 1 m in the Southern Hemisphere.

The AMIP simulation is created in the same manner as a coupled simulation, but using compsets starting with NF.

### AMIP compsets

Compsets starting with NF are NorESM AMIP (atmosphere/land-only) configurations. Predefined compsets for AMIP simulations can be found in:

```
<noresm_base>/components/cam/cime_config/config_compsets.xml
```

To create an AMIP-type experiment:

```
./create_newcase --case <PAT_TO_CASEFOLDER>/CASENAME --compset NFHISTnorbc --res f19_
↪f19 --mach fram --project nn2345k --user-mods-dir cmip6_noresm_fsst_xaer
```

Defined user-mod-dirs are available in:

```
<noresm-base>/components/cam/cime_config/usermods_dirs/
```

Available user-mod-dir options for NorESM2 used in CMIP6:

- `cmip6_noresm_fsst_xaer` (history\_aerosol=.true. , AEROFFL and AEROCOM defined)
- `cmip6_noresm_fsst_hifreq_xaer` (high frequency output, history\_aerosol=.true. , AEROFFL and AEROCOM defined)

## Creating your own compset for AMIP simulations

The essential file to edit for a new AMIP NorESM compset is:

```
<noresm_base>/components/cam/cime_config/config_compsets.xml
```

This examples shows how to simply add the “NFHIST” compset to config\_components.xml. In <noresm\_base>/components/cam/cime\_config/config\_compsets.xml the NFHIST is set as

```
<!-- fSST : evolving NorESM derived ; DMS: evolving NorESM derived -->
<compset>
  <alias>NFHISTnorbc</alias>
  <lname>HIST_CAM60%NORESM%NORBC_CLM50%BGC-CROP_CICE%PRES_DOCN%DOM_MOSART_SGLC_SWAV</
  ↪lname>
  <science_support grid="f09_f09_mg17"/>
</compset>
```

E.g.

- **HIST\_CAM60%NORESM%NORNC**
  - Forcing and input files read from historical conditions (1850 - 2015)
  - Build CAM6.0 (the atmosphere model) with NorESM specific additions and NorESM derived boundary conditions (for the boundary conditions, please see explanation below).
  - Note for some AMIP compsets CAM60%PTAERO may be used instead of CAM60%NORESM. Don't worry, those are identical.
- **CLM50%BGC-CROP**
  - Build CLM5 (land model) with a global crop model (interactive vegetation)
  - If you want pre-described vegetation, use CLM50%SP
- **CICE%PRES**
  - Build CICE (sea-ice model) with prescribed sea-ice
- **DOCN%DOM**
  - Build data ocean with fixed SSTs.
- **MOSART**
  - Build MOSART (river runoff model) with default configurations
- **SGLC\_SWAV**
  - The SGLC (land-ice) and SWAV (ocean-wave) models are not interactive, but used only to satisfy the interface requirements

To use different prescribed fields for SSTs and sea-ice cover than the default, change the value of the variable **SSTICE\_DATA\_FILENAME** in the **evn\_run.xml** file to the full path of a different file that complies to the requirements of the CICE and the data-ocean model.

## AMIP-style simulations with observed SSTs and frc2 emission files

The compsets using observed sea surface temperatures include compset names with *fsst*:

```
<compset>
  <alias>NFHISTfsst</alias>
  <lname>HIST_CAM60%NORESM%FSST_CLM50%BGC-CROP_CICE%PRES_DOCN%DOM_MOSART_SGLC_SWAV</lname>
  <science_support grid="f09_f09_mg17"/>
</compset>
```

The *frc2* option uses differently organized emission files. A new set of emission files have been made to avoid the occurrence of random mid-month model crashes. These crashes are related to the reading of emission files, but are still under investigation. To use the newest emission files choose compset names with *frc2* or if you want to create a new compset add *%FRC2*:

```
<compset>
  <alias>NFHISTfsstfrc2</alias>
  <lname>HIST_CAM60%NORESM%FSST%FRC2_CLM50%BGC-CROP_CICE%PRES_DOCN%DOM_MOSART_SGLC_SWAV</lname>
  <science_support grid="f09_f09_mg17"/>
</compset>
```

## NorESM2-derived boundary conditions for AMIP-style simulations

While the prescribed values used in atmosphere-only simulations are often based on observations, one might also want to use values that resemble those from a fully-coupled simulation with NorESM2. To achieve this, it is necessary to use prescribed boundary conditions for SST, sea-ice cover and upper-ocean DMS concentrations (all three fields taken from the fully-coupled simulation). Up to now, 4 sets of boundary conditions have been made:

- **a pre-industrial climatology with 2-degree resolution**
  - 2x2 degree resolution in the horizontal
  - contains 12 monthly values
  - based on a 30-year period (years 1751–1780) from the CMIP6 pre-industrial control (piControl) simulation with 2x2 degree resolution (NorESM2-LM).
  - was used for the CMIP6 simulation piClim-control, and all simulations that are perturbation runs based on piClim-control, with NorESM2-LM (mostly 30-year long simulations)
- **a pre-industrial climatology with 1-degree resolution**
  - as above but on 1x1 resolution in the horizontal, and based on years 1351-1380 from the CMIP6 piControl simulation with 1x1 degree resolution (NorESM2-MM)
- **the historical period**
  - 2x2 degree resolution in the horizontal
  - contains monthly values for years 1849-20155
  - based on the period 1850–2014 from the CMIP6 historical simulation with 2x2 degree resolution (NorESM2-LM).
  - was used for the CMIP6 simulation histSST, and all simulations that are perturbation runs based on sstHIST, with NorESM2-LM (165-year long simulations).
- **a future period based on SSP3-7.0**

- 2x2 degree resolution in the horizontal
- contains monthly values for years 2014-2101
- based on years 2015-2100 from the CMIP6 SSP3-7.0 simulation with 2-degree resolution (NorESM2-LM).
- was used for the CMIP6 simulation ssp370SST, and all simulations that are perturbation runs based on ssp370SST, with NorESM2-LM (86-year long simulations).
- for comparison of piClim-control and piControl, one should focus on the 30-year periods mentioned above (year 1751–1780 and 1351–1380) due to inter-decadal variability and/or drifts in piControl

Another thing that must be kept in mind when doing AMIP-style simulations that should resemble the coupled NorESM2 climate as closely as possible is the choice of flux parameterization used for the transfer of heat, moisture and momentum between the ocean and atmosphere, the so-called COARE flux parameterization. The flux parameterization is controlled by the variable **OCN\_FLUX\_SCHEME** in the `env_run.xml` file. The standard choice in CESM is:

```
OCN_FLUX_SCHEME=0
```

This parameterisation is different from the standard flux parameterization used in NorESM2, which is activated by:

```
OCN_FLUX_SCHEME=1.
```

and ends up in the `drv_in` namelist as:

```
flux_scheme=1.
```

## Code modifications

To make more substantial modification to the code than what is possible by the use of `user_nl_cam`, there are two methods:

1. Make a branch from the NorESM2 version (branch or release) you want to modify, checkout this branch in order to make code changes directly in the source code folder.
2. Copy the source code (the fortran file(s) you want to modify) to the `SourceMods/src.cam` folder in the case directory, and then make the modifications needed before building the model. By the use of this method, you will not change the source code in the `<noresm-base>` folder.

The CAM6/CAM6-Nor source code is located in:

```
<noresm-base>/components/cam/
```

### 1.3.7 Ocean and Sea-Ice

#### BLOM

In NorESM2, the ocean component BLOM/iHAMOCC can be either run fully coupled to the other model components, or in ocean carbon-cycle stand alone configuration. The latter setup follows the CMIP6-OMIP protocol (see below) and is driven by atmospheric input data derived from reanalysis products (called *data-atmosphere*, DATM).

## OMIP-type experiments

The Ocean Model Intercomparison Project (OMIP; Griffies et al., 2016) is an endorsed project in the CMIP6. OMIP provides a protocol for global ocean/sea-ice models forced by a common prescribed atmospheric forcing, and a protocol for ocean diagnostics to be saved as part of CMIP6. OMIP includes a physical component (Griffies et al., 2016) and a biogeochemistry component (Orr et al. 2017).

Prior to OMIP, the framework of the Coordinated Ocean-ice Reference Experiments (CORE) provides ocean and climate modellers with a common protocol for running coupled ocean/sea-ice models with boundary forcing derived from common atmospheric datasets. The CORE forcing dataset was described by Large and Yeager (2004, 2009), which is largely based on the surface-atmospheric fields derived from NCEP/NCAR atmospheric reanalysis. CORE has subsequently evolved into phase 1 of the physical part of OMIP (OMIP-1). The OMIP1/CORE-II experiment forces the ocean through use of the interannually varying atmospheric state of Large and Yeager (2009), along with river runoff data based on Dai et al. (2009).

The Large and Yeager (2009) forcing dataset covers the period from 1948 to 2009, and has not been updated since 2009. Thereafter, Tsujino et al. (2018) developed a surface atmospheric dataset based on the Japanese 55-year atmospheric reanalysis (JRA-55; Kobayashi et al., 2015), referred to as JRA55-do, which has been endorsed under the protocol for phase 2 of OMIP (OMIP-2). Currently, JRA55-do covers the period from 1958 to 2018 with planned continuous annual updates. Comparing to CORE-II, the JRA55-do forcing has an increased temporal frequency (from 6 hours to 3 hours), and a refined horizontal resolution (from  $1.875^\circ$  to  $0.5625^\circ$ ).

The readers are referred to Large and Yeager (2009) and Tsujino et al. (2018) for a detailed description of the CORE-II and JRA55-do forcing, respectively, including the bulk formulae used for computing turbulent fluxes for heat and momentum. An evaluation and comparison of the simulated ocean and sea ice mean states and variability from 11 state-of-the-art global ocean/sea-ice models (including NorESM-BLOM), based on OMIP-1/CORE-II and OMIP-2/JRA55-do, is presented by Tsujino et al. (2020).

## BLOM OMIP specifics

- In the BLOM OMIP simulations, sea surface salinity is restored to monthly climatology with a piston velocity of 50 m per 300 days applied globally for both OMIP-1 and OMIP-2 simulations. The restoring salt flux is normalized so that the global area weighted sum of the restoring flux is zero.
- A dimensionless parameter  $\alpha = 1$  is used in the estimation of the near-surface wind correction, e.g.  $\tau_{atm-ocn}$ , to account for the imprint of the ocean currents on the surface wind stress in an ocean/sea-ice model. The OMIP community has not reached a consensus on the way  $\alpha$  should be imposed; for a discussion of  $\alpha$  and its values among other OMIP models, refer to Tsujino et al. (2020).
- The OMIP-1 and OMIP-2 simulations of BLOM have completed 6 repeating forcing cycles of the forcing periods 1948-2009 and 1958-2018, respectively.
- The OMIP-1/CORE-II experiment is documented by Bentsen et al. (2020, *in prep.*), including a comparison to a previous CORE-II experiment simulated by the CMIP5 version of NorESM1. In addition, a number of common global ocean and sea ice metrics simulated by BLOM (including both OMIP-1 and OMIP-2 experiments) is included and evaluated in the multi-model synthesis paper of Tsujino et al. (2020).



## Creating an OMIP case

General syntax:

```
cd <noresm-base>/cime/scripts
./create_newcase --case <path_to_case_dir>/<casename> --walltime <time> --compset
↪<compset_name> --res <resolution> --machine <machine_name> --project <project_name>
↪--user-mods-dir <user_mods_dir> --output-root <path_to_run_dir>/<noresm_run_dir> --
↪run-unsupported
```

An example for creating the OMIP-1 case:

```
./create_newcase --case ../../cases/NOIIAOC20TR_T62_tn14_20190628 --compset
↪NOIIAOC20TR --res T62_tn14 --machine vilje --project nn2345k --run-unsupported
```

An example for creating the OMIP-2 case:

```
./create_newcase --case ../../cases/NOIIAJRAOC20TR_TL319_tn14_20190710 --compset
↪NOIIAJRAOC20TR --res TL319_tn14 --machine vilje --project nn2345k --run-unsupported
```

## OMIP compsets

- OMIP-1/CORE-II: NOIIAOC20TR
- OMIP-2/JRA55-do: NOIIAJRAOC20TR

## Forcing datasets

### • OMIP-1/CORE-II

Forcing dataset described in detail by Large and Yeager (2009). The forcing files are located in

```
/DIN_LOC_ROOT/ocn/iaf/,
```

where DIN\_LOC\_ROOT is the base input data directory, which depends on the machine (on fram it is /cluster/shared/noresm/inputdata/).

### • OMIP-2/JRA55-do

Forcing dataset described in detail by Tsujino et al. (2018). The forcing files are located at (on Fram)

```
/DIN_LOC_ROOT/ocn/jra55/v1.3_noleap/
```

## Modify user namelist for BLOM/iHAMOCC

Model parameters and adjusting model output can be done in `user_nl_blom`, which is present in the case directory after `./case.setup` has been executed. The resolved namelist for BLOM/iHAMOCC is saved in `CaseDocs/ocn_in`, and specifies a number of physical parameters (such as vertical and horizontal mixing), as well as model output settings and frequencies. Output settings include options for daily (`hd/hbgcd`), monthly (`hm/hbgcm`), and yearly (`hy/hbgcy`) output, where the files containing 'bgc' in their filenames are iHAMOCC output files. Note that the resolved namelist (`CaseDocs/ocn_in`) should **never** be used to place user defined changes, since this file is re-created (overwritten) every time the model is submitted. User defined namelist changes need to be placed in `user_nl_blom`, for example

```
BDMC2 = .15
NIWGF = .5
```

Note that it does not matter which namelist group the variables belong (namelist groups must not be specified in user\_nl\_biom). BLOM parameters that can be changed via namelist settings are documented in the resolved namelist file (see CaseDocs/ocn\_in after running ./preview\_namelist or building the case). In iHAMOCC, model parameters are currently hard-coded, i.e. they cannot be changed through namelist settings. To change iHAMOCC model parameters, please see below under ‘Code modifications’.

For changing the output in BLOM or iHAMOCC, the example below shows how to change the monthly mean (default) to yearly mean layered ocean temperature. The default setting (as can be seen in CaseDocs/ocn\_in after running ./preview\_namelist or building the case) is

```
&DIAPHY
  GLB_FNAMETAG = 'hd', 'hm', 'hy',
  GLB_AVEPERIO = 1, 30, 365,
  ...
  LYR_TEMP      = 0, 4, 0,
  ...
```

which means that the model layered temperature has a monthly mean output with single precision (4-byte;real4), e.g.

```
0 - variable is not written
2 - variable is written as int2 with scale factor and offset
4 - variable is written as real4
8 - variable is written as real8
```

If one would like output of yearly mean layered temperature, simply change LYR\_TEMP in user\_nl\_biom to:

```
LYR_TEMP      = 0, 0, 4,
```

Available output variables for BLOM and iHAMOCC are documented in the resolved namelist file (see CaseDocs/ocn\_in after running ./preview\_namelist or building the case).

### Code modification

To make more substantial modification to the BLOM/iHAMOCC code than what is possible by the use of user\_nl\_biom, there are two methods:

1. Make a branch from the NorESM2 version (branch or release) you want to modify, checkout this branch in order to make code changes directly in the source code folder.
2. Copy the source code (the fortran file(s) you want to modify) to the SourceMods/src.biom folder in the case directory, and then make the modifications needed before building the model. By the use of this method, you will not change the source code in the <noresm-base> folder.

As mentioned above, if you need to change a model parameter for iHAMOCC you need to modify the source code. All iHAMOCC parameters are defined in the routine beleg\_parm.F90.

The BLOM source code is located in:

```
<noresm-base>/components/blom/
```

The iHAMOCC source code is located in:

```
<noresm-base>/components/blom/hamocc
```

## CICE

The sea ice model component is based upon version 5.1.2 of the CICE sea ice model of Hunke et al. (2015).

### Initial conditions

By default, the CICE model is initialized with a ‘default’, simplified, sea ice field with sea ice in cold regions (air temperature below 0 degree C), north of 70 N and south of 60 S. The sea ice thickness in these regions is horizontal homogeneous, with a uniform snow cover. This behavior is given by the `ice_ic` variable in the namelist. This can be changed to start without sea ice by setting:

```
&setup_nml
  ice_ic = "none"
```

in the `user_nl_cice` in the case folder, or by specifying a restart file which would give the desired sea ice state:

```
&setup_nml
  ice_ic = "PATH_TO_FILE/NAME_OF_FILE.cice.r.YEAR-01-01-00000.nc"
```

The file used for NorESM2-MM CMIP6 piControl simulation is:

```
finidat = N1850_f09_tn14_20190913.cice.r.1200-01-01-00000.nc
```

The file used for NorESM2-LM CMIP6 piControl simulation is:

```
finidat = N1850_f19_tn14_11062019.cice.r.1600-01-01-00000.nc
```

Information about which file is used as an initial condition (in addition to parameter settings and other files used as input) file is in `ice_in`. This file can be found in:

```
<casefolder>/CaseDocs/ice_in
```

and in the Run folder:

```
<RUN_DIR>/case/run/ice_in
```

This information is also written to the `ice.log.*` file generated during the run.

### NorESM2 specific addition

A NorESM2-specific change is including the effect of wind drift of snow into ocean following Lecomte et al. (2013) This change can be tuned on/off in the `user_nl_cice` in the case folder. Default is:

```
&snowphys_nml
  blowingsnow = "lecomte2013"
  ksno = 0.3
  rhos = 330.0
```

and will use NorESM2 treatment of wind drift of snow. Setting

```
&snowphys_nml
  blowingsnow = "none"
```

will reset the NorESM2 specific addition and the effect of wind drift of snow into ocean will not be included. It is also possible to change the snow density `rhos` and the snow thermal conductivity `ksno`. Be aware that this will influence the overall tuning of the coupled model.

## Modify user name lists for CICE

Output from the model is changed by controlling the `user_nl_cice` file in your casefolder. By default, the file typically looks like this:

```
histfreq = 'm','d','x','x','x'
histfreq_n = 1,1,1,1,1
f_CMIP = 'mdxxx'
f_hi = "mxxxx"
f_hs="mxxxx"
f_fswdn="mxxxx"
f_fswabs="mxxxx"
f_congel="mxxxx"
f_frazil="mxxxx"
f_meltd="mxxxx"
f_melts="mxxxx"
f_meltd="mxxxx"
f_meltd="mxxxx"
f_fswthru="mxxxx"
f_dvidtd="mxxxx"
f_dvidtd="mxxxx"
f_daidtd="mxxxx"
f_daidtd="mxxxx"
f_apond_ai="mxxxx"
f_hpond_ai="mxxxx"
f_apeff_ai="mxxxx"
f_snowfrac="mxxxx"
f_aicen="mxxxx"
f_snowfracn="mxxxx"
```

where the `f_*` flags are used to change the writing of specific variables, and the `histfreq` and `histfreq_n` variables are used to specify type of history files written, and their frequency. The `f_CMIP` flag activates the specific SIMIP/CMIP variables used the CMIP6 runs. By default, the model writes extensive output with a monthly frequency, and more limited at daily basis.

The easiest way to turn of daily output from CICE is to put

```
histfreq = 'm','x','x','x','x'
```

in the `user_nl_cice` file.

High-frequency output can be achieved by manipulating the `histfreq` and `histfreq_n` variables, together with the specific variable should be at higher frequency. To use 3-hourly output of the sea ice velocity from the model set

```
histfreq = 'm','d','h','x','x'
histfreq_n = 1,1,3,1,1
f_siu = 'm,d,h,x,x'
f_siv = 'm,d,h,x,x'
```

Be aware that the model writes one file per time step. Therefore, this should be done for short runs, only, and the high-frequency output should be collected together in one (or a few) larger files after the model run, e.g. by using the `ncrcat` command.

## Code modification

To make more substantial modification to the code than what is possible by the use of `user_nl_cice`, there are two methods:

1. Make a branch from the NorESM2 version (branch or release) you want to modify, checkout this branch in order to make code changes directly in the source code folder.
2. Copy the source code (the fortran file(s) you want to modify) to the `SourceMods/src.cice` folder in the case directory, and then make the modifications needed before building the model. By the use of this method, you will not change the source code in the `<noresm-base>` folder.

The CICE source code is located in:

```
<noresm-base>/components/cice/src/
```

More information is found in the CESM-CICE User Guide: <https://cesm.cice.readthedocs.io/en/latest/>

## References

Dai, A., Qian, T., Trenberth, K. E., and Milliman, J. D.: Changes in continental freshwater discharge from 1948 to 2004, *J. Climate*, 22, 2773–2792, <https://doi.org/10.1175/2008JCLI2592.1>, 2009.

Griffies et al., Coordinated Ocean-ice Reference Experiments (COREs), *Ocean Model.*, 26, 1–46, doi:10.1016/j.ocemod.2008.08.007, 2009.

Griffies et al., OMIP contribution to CMIP6: experimental and diagnostic protocol for the physical component of the Ocean Model Intercomparison Project, *Geosci. Model Dev.*, 9, 3231–3296, <https://doi.org/10.5194/gmd-9-3231-2016>, 2016.

Hunke, E. C., et al. “CICE: The Los Alamos Sea ice Model Documentation and Software User’s Manual Version 5 (Tech. Rep. LA-CC-06–012).” Los Alamos, NM: Los Alamos National Laboratory (2015).

Hunke, Elizabeth, Lipscomb, William, Jones, Philip, Turner, Adrian, Jeffery, Nicole, and Elliott, Scott. CICE, The Los Alamos Sea Ice Model. Computer software. <https://www.osti.gov/servlets/purl/1364126>.

Large, W. and S. Yeager, 2004: Diurnal to decadal global forcing for ocean and sea-ice models: the datasets and flux climatologies. NCAR Technical Note: NCAR/TN-460+STR, CGD Division of the National Centre for Atmospheric Research.

Large, W.G. and S.G. Yeager. 2009: The global climatology of an interannually varying air-sea flux data set. *Climate Dynamics*, 33, 341-364, doi:10.1007/s00382-008-0441-3.

Lecomte, O., T. Fichefet, M. Vancoppenolle, F. Domine, F. Massonnet, P. Mathiot, S. Morin, and P.Y. Barriat (2013), On the formulation of snow thermal conductivity in large-scale sea ice models, *J. Adv. Model. Earth Syst.*, 5, 542–557, doi:10.1002/jame.20039

Orr et al., Biogeochemical protocols and diagnostics for the CMIP6 Ocean Model Intercomparison Project (OMIP), *Geosci. Model Dev.*, 10, 2169–2199, <https://doi.org/10.5194/gmd-10-2169-2017>, 2017.

Steele, M., Morley, R., and Ermold, W.: PHC: A Global Ocean Hydrography with a High-Quality Arctic Ocean, *J. Climate*, 14, 2079–2087, 2001.

Tsujino et al., Evaluation of global ocean–sea-ice model simulations based on the experimental protocols of the Ocean Model Intercomparison Project phase 2 (OMIP-2), *Geosci. Model Dev. Discuss.*, <https://doi.org/10.5194/gmd-2019-363>, in review, 2020.

### 1.3.8 Land and river run off

#### CLM5

The land model used in NorESM2 is the Community Land Model version 5 (CLM5): <http://www.cesm.ucar.edu/models/clm/>

Specific questions about CLM can be addressed Lei Cai, email: [leca@norceresearch.no](mailto:leca@norceresearch.no)

#### CLM5 model configurations available in NorESM2

CLM5 can be run with a prognostic crop model with prognostic vegetation state and active biogeochemistry. The global crop model is on in BGC default configuration with 8 temperate and tropical crop types and has the capability to dynamically simulate crop management and crop management change through time. The BGC-CROP option is used in all NorESM2 CMIP6 experiments and is activated in the compset by:

```
CLM5%BGC-CROP
```

CLM5 in NorESM2 can also be run with a prescribed satellite vegetation phenology model. This option can be activated in the compset by:

```
CLM5%SP
```

Note that the BGC-CROP option is more expensive than SP (+ca 10-15% CPU time)

#### CLM5 atmospheric coupling

The current state of the atmospheric component, CAM6-Nor, at a given time step is used to force the land model. The land model then initiates a full set of calculations for surface energy, constituent, momentum, and radiative fluxes. The land model calculations are implemented in two steps:

- 1. The land model proceeds with the calculation of surface energy, constituent, momentum, and radiative fluxes using the snow and soil hydrologic states from the previous time step.
- 2. The land model then updates the soil and snow hydrology calculations based on these fluxes. These fields are passed to the atmosphere. The albedos sent to the atmosphere are for the solar zenith angle at the next time step but with surface conditions from the current time step.

From CLM5 user guide: [https://escomp.github.io/ctsm-docs/versions/release-clm5.0/html/tech\\_note/Ecosystem/CLM50\\_Tech\\_Note\\_Ecosystem.html#atmospheric-coupling](https://escomp.github.io/ctsm-docs/versions/release-clm5.0/html/tech_note/Ecosystem/CLM50_Tech_Note_Ecosystem.html#atmospheric-coupling)

#### CLM5 surface data

Required surface data for each land grid cell include:

- the glacier, lake, and urban fractions of the grid cell (vegetated and crop occupy the remainder)
- the fractional cover of each plant functional type (PFT), monthly leaf and stem area index and canopy top and bottom heights for each PFT,
- soil color, soil texture, soil organic matter density,
- maximum fractional saturated area, slope, elevation,
- biogenic volatile organic compounds (BVOCs) emissions factors,

- population density
- gross domestic production
- peat area fraction
- peak month of agricultural burning. Optional surface data include crop irrigation and managed crops.

From CLM5 user guide: [https://escomp.github.io/ctsm-docs/versions/release-clm5.0/html/tech\\_note/Ecosystem/CLM50\\_Tech\\_Note\\_Ecosystem.html#surface-data](https://escomp.github.io/ctsm-docs/versions/release-clm5.0/html/tech_note/Ecosystem/CLM50_Tech_Note_Ecosystem.html#surface-data):

### Steps to create a complete set of surface data files for CLM

A technical description on how to create new surface data sets is found here: [https://github.com/NorESMhub/NorESM/blob/norasm2/doc/configurations/clm\\_surfdata.pdf](https://github.com/NorESMhub/NorESM/blob/norasm2/doc/configurations/clm_surfdata.pdf)

### The initial state of CLM5

The land model needs to read in the initial state from a restart file. This can be customized in `user_nl_clm` in the case folder

```
finidat = '<path_to_inputdata>/inputdata/<path_to_file>/CLMFILENAME.clm2.r.YR-01-01-
↳00000.nc'
```

e.g. Fram @ Sigma2:

```
finidat = '/work/shared/norasm/inputdata/cesm2_init/b.e20.B1850.f09_g17.pi_control.
↳all.297/0308-01-01/b.e20.B1850.f09_g17.pi_control.all.297.clm2.r.0308-01-01-00000.nc
↳'
```

The file used for NorESM2-MM CMIP6 piControl simulation is:

```
finidat = N1850_f09_tn14_20190913.clm2.r.1200-01-01-00000.nc
```

The file used for NorESM2-LM CMIP6 piControl simulation is:

```
finidat = N1850_f19_tn14_11062019.clm2.r.1600-01-01-00000.nc
```

Information about which file is used as an initial state (in addition to parameter settings and other files used as input) file is in `lnd_in`. This file can be found in:

```
<casefolder>/CaseDocs/lnd_in
```

and in the Run folder:

```
<RUN_DIR>/case/run/lnd_in
```

### User name list modifications

An example of how you can modify `user_nl_clm`. This adds four auxiliary history files in addition to the standard monthly files. The first two are daily, and the last two are six and three hourly:

```
hist_mfilt   = 1,365,30,120,240
hist_nhtfrq  = 0,-24,-24,-6,-3
hist_fincl2  = 'TSOI', 'TG', 'TV', 'FIRE', 'FSR', 'FSH', 'EFLX_LH_TOT', 'WT'
hist_fincl3  = 'FSA'
```

(continues on next page)

(continued from previous page)

```
hist_fincl4 = 'TSOI', 'TG', 'TV', 'FIRE', 'FSR', 'FSH', 'EFLX_LH_TOT', 'WT'
hist_fincl5 = 'TSOI', 'TG', 'TV', 'FIRE', 'FSR', 'FSH', 'EFLX_LH_TOT', 'WT'
```

If you are not interested in CLM output variables, you can remove any additional history list like `hist_fincl2` or `hist_fincl3` in `user_nl_clm` and set reduced output history frequency to every 50 or 100 years depending on your run length. For example if `STOP_N=50` years, you can set:

```
hist_mfilt = 50
hist_nhtfrq = -8760
```

-8760 means one average value per year, and 50 years in one file.

- The full namelist definitions and defaults in the CLM5: [http://www.cesm.ucar.edu/models/cesm2/settings/current/clm5\\_0\\_nml.html](http://www.cesm.ucar.edu/models/cesm2/settings/current/clm5_0_nml.html)

## Spin up of CLM5

A long spin up is required for running NorESM2 with CLM50%BGC-CROP to achieve e.g. land carbon balance. Therefore, an off-line spin up of CLM50%BGC-CROP has to be performed in order to save computation time.

### Generating atmospheric forcing data

Atmospheric forcing data from the coupled NorESM2 simulation are used to run CLM5 stand alone spin up. To output such atmospheric forcing data, the following commands have to be added to `user_nl_cpl` in the coupled simulation of interest::

```
&seq_infodata_inparm
  histaux_a2x      = .true.
  histaux_a2x1hr   = .true.
  histaux_a2x1hri  = .true.
  histaux_a2x3hr   = .true.
  histaux_a2x3hrp  = .false.
  histaux_a2x24hr  = .true.
  histaux_l2x      = .true.
  histaux_l2x1yrg  = .true.
  histaux_r2x      = .true.
```

### Running CLM stand alone with NorESM2 forcing data

To use NorESM2 history files as the forcing, `CPLHISTForcing` mode needs to be activated. In `CPLHISTForcing` mode, the model is assumed to have 3-hourly for a global grid from a previous simulation (see description above). The data atmosphere (`datm`) forcing is divided into three streams: precipitation, solar, and everything else.

To create a new case for stand alone CLM5 spin up with NorESM2 forcing data, one should choose the same resolution as the coupled simulation (`f19_tn14` for NorESM2-LM and `f09_tn14` for NorESM2-MM). The compset to use is `I1850BgcCropCmip6`. For example, to create a new NorESM2-LM case,

```
./create_newcase --case <PAT_TO_CASEFOLDER>/CASENAME --compset N1850BgcCropCmip6 --
↳ res f19_tn14 --mach fram --project nn9560k
```

Using the `CPLHIST` forcing, the offline spin up needs to be run in two steps:

- **1. Accelerated spinup (300 years):**

When entering “Accelerated Spinup” mode, soil carbon pools will be scaled down by a factor ~40, vegetation pools scaled down by ~5 In `env_run.xml`, include:



```
./xmlchange RUN_TYPE="startup",STOP_N=400,STOP_OPTION="nyears",REST_N=50
./xmlchange CLM_ACCELERATED_SPINUP="on"
./xmlchange CLM_FORCE_COLDSTART="on"
./xmlchange DATM_MODE=CPLHIST,DATM_PRESAERO=cplhist,DATM_TOPO=cplhist
./xmlchange DATM_CPLHIST_DIR=/cluster/shared/noresm/inputdata/cplhist/N1850_f09_tn14_
↪20190726_751-850
./xmlchange DATM_CPLHIST_CASE=N1850_f09_tn14_20190726
./xmlchange DATM_CPLHIST_YR_ALIGN=751,DATM_CPLHIST_YR_START=751,DATM_CPLHIST_YR_
↪END=850
```

Note. The casename for the CPLHIT (N1850\_f09\_tn14\_20190726, N1850\_f09\_tn14\_20190726\_751-850) and all numbers need to be changed for the simulation of interest.

**In user\_nl\_clm set output frequency to every 50 or 100 years <= REST\_N:: hist\_mfilt = 50 hist\_nhtfrq = -8760**

- **2. Normal spinup (1800 years):**

When exiting Accelerated Spinup and entering normal spinup, the carbon pools will be scaled up back to normal levels

### Recoupling CLM5 with NorESM2

NorESM2 can then be run with CLM5 using the restart file from the end of the spinup as the initial file. To do this, modify the set up in user\_nl\_clm:

```
finidat = '<path_to_inputdata>/inputdata/<path_to_file>/CLM_SPINUP_FILENAME.clm2.r.YR-
↪01-01-00000.nc'
```

A description of the NorESM2 CLM5 spin up, recoupling and diagnostics can be found here: <https://github.com/NorESMhub/NorESM/blob/noresm2/doc/configurations/NorESM-CLM-memo.pdf>

### Code modification

To make more substantial modification to the BLOM/iHAMOCC code than what is possible by the use of user\_nl\_clm, there are two methods:

1. Make a branch from the NorESM2 version (branch or release) you want to modify, checkout this branch in order to make code changes directly in the source code folder.
2. Copy the source code (the fortran file(s) you want to modify) to the SourceMods/src.clm folder in the case directory, and then make the modifications needed before building the model. By the use of this method, you will not change the source code in the <noresm-base> folder.

The CLM5 source code is located in:

```
<noresm-base>/components/clm/src/
```

### Land-only experiments

**For land-only simulations**, there is no difference in running the CLM5 in CESM2 and that in NorESM2. For a detailed description on how to set up, modify, build and run CLM5 stand alone experiments, please see the CLM5.0 users guide: [https://escomp.github.io/ctsm-docs/versions/release-clm5.0/html/users\\_guide/setting-up-and-running-a-case/choosing-a-compset.html](https://escomp.github.io/ctsm-docs/versions/release-clm5.0/html/users_guide/setting-up-and-running-a-case/choosing-a-compset.html) (last accessed 7th May 2020)

### NorESM2 specific additions

Remove infiltration excess water as runoff if the temperature of the surface water pool is below freezing. For details please see model-description/lnd\_model

The NorESM2 specific addition can be tuned on/off by a flag in the user\_nl\_clm in the case folder. Setting:

```
reset_snow = .true.
```

will use NorESM2 treatment of the surface water in CLM (see previous description).

Setting:

```
reset_snow = .false.
```

will use CESM2 treatment of the surface water in CLM (see previous description).

### CLM5 specifics

- There is no information exchange within the CLM model between sub-grid tiles (landunits, columns, plant functional types (PFTs)).
- Sub-grid tiles only exchange information with the atmosphere. In the current CLM, there is no advection of heat and water at depth.
- The horizontal resolution of the CLM keeps the same as for the atmosphere (f19, f09).
- Vertically, there are four soil structures to set in the CLM namelist file. CLM5 model configurations available in NorESM2:

```
10SL_3.5m    = standard CLM4 and CLM4.5 version  
23SL_3.5m    = more vertical layers for permafrost simulations  
49SL_10m     = 49 layer soil column, 10m of soil, 5 bedrock layers  
20SL_8.5m    = 20 layer soil column, 8m of soil, 5 bedrock layers
```

By default, 20SL\_8.5m is employed.

### MOSART

The Model for Scale Adaptive River Transport (MOSART) is the default river model for CESM2, CLM5 and NorESM2. For more information please see:

<http://www.cesm.ucar.edu/models/cesm2/river/>

For a technical user guide, please see:

[https://escomp.github.io/ctsm-docs/versions/release-clm5.0/html/tech\\_note/MOSART/CLM50\\_Tech\\_Note\\_MOSART.html](https://escomp.github.io/ctsm-docs/versions/release-clm5.0/html/tech_note/MOSART/CLM50_Tech_Note_MOSART.html)

The methods and syntax for modifying the user namelist and code in MOSART are similar to CLM5, so the previous description can be used. The user namelist for MOSART is user\_nl\_mosart and source code files should be copied to SourceMods/src.mosart/ in the case folder.

The MOSART source code is located in:

```
<noresm-base>/components/mosart/src/
```

### 1.3.9 Ensemble experiments

#### How to run several members at once

Running several ensemble members in a single model experiment (case) is possible using the built-in **multi-instance component functionality**. This allows you to run multiple component instances under one model executable.

#### Creating an ensemble experiment:

To set up an experiment with 5 members, invoke the `create_newcase` script with the **multi-driver** and the **ninst** arguments. For instance

```
./create_newcase --case $HOME/noresm_cases/MY_AWESOME_ENSEMBLE_EXP --multi-driver --
↪ninst 5 --res f19_f19_mg17 --mach vilje --compset NFHISTnorpdmsbc --run-
↪unsupported --project <YOUR-PROJECT-FOR-CPU-HOURS-ON-VILJE>
```

will create a new case in the folder `$HOME/noresm_cases/MY_AWESOME_ENSEMBLE_EXP` with 5 ensemble members, the `f19_f19_mg17` resolution, the `NFHISTnorpdmsbc` compset, with machine settings for Vilje, and using CPU hours from `YOUR-PROJECT-FOR-CPU-HOURS-ON-VILJE`. `NFHISTnorpdmsbc` is the compset for the CMIP6 AMIP experiment in which sea surface temperatures and sea ice are prescribed to observed values.

When using the multi-instance component, *you will get one user namelist for each member and for each component* after running the script `case_setup` from your case folder. For the above case, these namelists are:

```
user_nl_cam_0001  user_nl_cice_0001  user_nl_clm_0001  user_nl_cpl          user_nl_
↪docn_0005      user_nl_mosart_0005
user_nl_cam_0002  user_nl_cice_0002  user_nl_clm_0002  user_nl_docn_0001   user_nl_
↪mosart_0001
user_nl_cam_0003  user_nl_cice_0003  user_nl_clm_0003  user_nl_docn_0002   user_nl_
↪mosart_0002
user_nl_cam_0004  user_nl_cice_0004  user_nl_clm_0004  user_nl_docn_0003   user_nl_
↪mosart_0003
user_nl_cam_0005  user_nl_cice_0005  user_nl_clm_0005  user_nl_docn_0004   user_nl_
↪mosart_0004
```

The namelists can be used to control various settings and to add output. Note that changes are made individually for the separate members, so if you for instance add extra output fields to `user_nl_cam_0001` these fields will only be written out for member 1. To also get the extra output for members 2-5, modify the four other user namelists from `cam`.

Something to keep in mind: the number of tasks set in `env_mach_pes.xml` corresponds to the number of tasks that one member will use. The total number of tasks used when running the experiment will thus be the tasks used by one member multiplied by the number of members. If you want to run a very large number of members, it might be a good idea to divide them into separate cases to make sure that you don't use too many CPUs per case.

#### Perturbing the ensemble members: PERTLIM

For the ensemble members to actually be different it is necessary to somehow perturb the initial condition of each member. The perturbation must be unique for each member.

One way of doing this is to use the CAM namelist parameter **PERTLIM**. The default value of **PERTLIM** is 0.0. By choosing a non-zero value, a random parturbation with a size up to that given by the **PERTLIM** value will be added to the initial temperature field in the atmosphere. The value could be round-off error size, for instance  $1e-14$ . **The PERTLIM value must be unique for each ensemble member.**

**PERTLIM only works for startup or hybrid runs.** It is not possible to use **PERTLIM** when doing a branch run.

While it is perhaps do-able to manually create 5 different namelists with five different PERTLIM values, you will probably want to do this in a more automated way if you want to run, say, 25 different members. The approach described below provides an example of how to do this. It is assumed that you have already created a template namelist called `user_nl_cam_template` which contains at least the following line:

```
pertlim          = 1e-14
```

The nice thing about using a template namelist is that you can add various content to the `user_nl_cam_template` that you want to apply to all members, such as additional output, and then you can create a set of namelists afterwards that only differ by their PERTLIM value using the script below:

```
#!/bin/bash

# Create cam namelists with 25 unique values for PERTLIM
nmems=25

# Default PERTLIM value in template version of cam namelist
# (will be replaced by random value)
pertlimSuffix=e-14
pertlimTemplate=1$pertlimSuffix

# Start of cam namelist
camNamelistStr=user_nl_cam_

# Name of cam template namelist
camNamelistTemplate=${camNamelistStr}template

# Path to perturbed cam namelists
path2camNamelist='namelists_perturbed'

# Create nmems unique values for PERTLIM
random=$(shuf -i 500-1500 -n $nmems)

# Loop through values random and make a new namelist file for cam for each
# value where the default PERTLIM value is replaced
counter=1
for val in $random ; do
    echo $counter
    x=$(bc -l <<< $(echo $val/1000))
    camNamelist=${camNamelistStr}${printf "%04d" $counter}
    pertlimNew=$(printf "%.02f" $x)$pertlimSuffix
    cp $camNamelistTemplate $path2camNamelist/$camNamelist
    sed -i -e 's/'"$pertlimTemplate"'/'"$pertlimNew"'/g' $path2camNamelist/
    ↪$camNamelist
    counter=$((counter+1))
done
```

The above script puts the namelists in a folder called `namelists_perturbed`, located in your current working directory. Remember that the namelists must be moved to the case folder when you are happy with them.

### Starting an ensemble run from a deterministic run:

In some cases, you may want to start an ensemble run as a hybrid run from a deterministic run (that is, a case with only one member). To achieve this, some manual interference is required to make sure that there are restart files and rpointers for each member. The script below provides an example of how to do this automatically for the case MY\_AWESOME\_ENSEMBLE\_EXP:

```
#!/bin/sh

# Input data
path2restfiles=<PATH-TO-RESTART-FILES-YOU-WANT-TO-USE>

# You can list several cases here
cases='MY_AWESOME_ENSEMBLE_EXP'

for case in $cases ; do
  path2runDir=<PATH-TO-RUN-DIRECTORY-OF-CASE>
  compsNetcdf='cam cpl cice clm2 docn mosart'
  compsRpointers='atm drv ice lnd ocn rof'

  for comp in $compsNetcdf ; do
    files=$(ls $path2restfiles/*$comp*)
    for file in $files ; do
      ln -sf $file $path2runDir/.
      for mem in $(seq -w 0001 0005) ; do
        ln -sf $file $path2runDir/${basename ${file/$comp/${comp}}_${mem}}
      done
    done
  done

  for comp in $compsRpointers ; do
    files=$(ls $path2restfiles/rpointer*$comp*)
    for file in $files ; do
      echo $file
      for mem in $(seq -w 0001 0005) ; do
        cp $file $path2runDir/${basename ${file/$comp/${comp}}_${mem}}
        if [ $comp == "atm" ] ; then
          sed -i -e 's/cam/cam_'"$mem"'/g' \
            $path2runDir/${basename ${file/$comp/${comp}}_${mem}}
        fi
        if [ $comp == "drv" ] ; then
          sed -i -e 's/cpl/cpl_'"$mem"'/g' \
            $path2runDir/${basename ${file/$comp/${comp}}_${mem}}
        fi
        if [ $comp == "ice" ] ; then
          sed -i -e 's/cice/cice_'"$mem"'/g' \
            $path2runDir/${basename ${file/$comp/${comp}}_${mem}}
        fi
        if [ $comp == "lnd" ] ; then
          sed -i -e 's/clm2/clm2_'"$mem"'/g' \
            $path2runDir/${basename ${file/$comp/${comp}}_${mem}}
        fi
        if [ $comp == "ocn" ] ; then
          sed -i -e 's/docn/docn_'"$mem"'/g' \
            $path2runDir/${basename ${file/$comp/${comp}}_${mem}}
        fi
        if [ $comp == "rof" ] ; then
          sed -i -e 's/mosart/mosart_'"$mem"'/g' \
```

(continues on next page)

(continued from previous page)

```

                                $path2runDir/$(basename ${file}/${comp}/${comp}_${mem} )
        fi
    done
done
done
done
done

```

### 1.3.10 Nudged experiments

#### Step by step guide for setting up a nudged simulation

##### Create a new case

with a compset that supports nudging e.g. NFHISTnorpddmsbcsdyn, or use your own compset and add:

```
-offline_dyn
```

to CAM\_CONFIG\_OPTS in env\_build.xml after creating a case.

#### Meteorology

Modify user\_nl\_cam include information about the meteorology you want to nudge to

```

&metdata_nl
  met_filenames_list = '/cluster/shared/noresm/inputdata/noresm-only/inputForNudging/
↳ERA_f09f09_32L_days/ fileList2001-2015.txt'
  met_data_file = '/cluster/shared/noresm/inputdata/noresm-only/inputForNudging/ERA_
↳f09f09_32L_days/ 2014-01-01.nc'

```

met\_filenames\_list points to a txt-file that must include all the meteorological nudging data that will be used for the entire simulation. The met\_data\_file points to the file in this list that includes the starting date of your simulation. The example above shows how to point to ERA-Interim data, created by Inger Helene Hafsaahl Karset (<https://www.duo.uio.no/handle/10852/72779>). You can also create your own model produced data (explanation further down in this document).

#### Nudging strength

Modify user\_nl\_cam to include information about the strength of the nudging:

```
met_rlx_time = 6
```

met\_rlx\_time is the relaxation time scale. If the timestep of the model is 0.5 hrs, a relaxation time scale of 6 corresponds to a nudging strength of  $0.5/6 \sim 0.083 = 8.3\%$ , meaning that 8.3 % of the nudged component (for example the wind) comes from the value in the met\_data\_file, while 93.7 % will come from the model itself. It is recommended to set met\_rlx\_time to the same value as the time frequency of the nudging data.

## Vertical levels

Modify `user_nl_cam` to include information about which levels in the vertical the nudging should apply to

```
met_rlx_bot = 60
met_rlx_top = 70
met_rlx_bot_bot = 0
met_rlx_bot_top = 0
```

By using the values in the example above, nudging will be applied to all levels in the vertical. If `met_rlx_bot_bot` and `met_rlx_bot_top` is set to heights (given in km) above the bottom layer of the model (0 km), `met_rlx_time` will decrease exponentially from `met_rlx_top` (where it will have the value of `met_rlx_time`) to `met_rlx_bot_bot` (where it will be zero from this level and all the way down to the ground). If you want to dampen or turn off the nudging intensity higher up, the same can be done to `met_rlx_bot` and `met_rlx_top` by setting these values to be lower in the atmosphere than the model top.

## Nudging only winds and surface pressure

Modify `user_nl_cam` if you only want to nudge winds and surface pressure:

```
met_nudge_only_uvps = .true.
```

This is recommended when looking at aerosol-cloud interactions, especially when nudging to meteorology that is not produced by the model itself (Zhang et al., 2014).

## Appropriate topography

Modify `user_nl_cam` to point to an appropriate topography file if nudging to meteorology from ERA-Interim or other meteorology that is not produced by the model itself. It is the field named 'PHIS' in the topography file that need to correspond to the source of the nudging data. `&cam_inparm bnd_topo = '/cluster/shared/noresm/inputdata/noresm-only/inputForNudging/ERA_f09f09_32L_days/ ERA_bnd_topo_noresm2_20191023.nc'`

## Correct calender

If nudging to reanalysis data, `CALENDAR` in `env_build.xml` should be changed from `NO_LEAP` to `GREGORIAN`.

## Correct start date

Modify `env_run.xml` to have the same `RUN_STARTDATE` as given in the `met_data_file`.

## Ready-steady-go

You are now ready to setup, build and submit your case.

## How to generate your own nudging inputdata

### Create a case

Create a case you want to generate data from

### Modify user\_nl\_cam

Modify user\_nl\_cam and/or other user namelists to output the preferred nudging data

```
&camexp
  mfilt = 1, 4,
  nhtfrq = 0, -6,
  avgflag_pertape='A','I',
  fincl2 = 'PS','U','V','T'
```

The example above will output ordinary h0 monthly mean files, one pr month, but also h1- files with instantaneous values of PS, U, V and T every six hours, four pr file.

### Move the nudging data to a preferred folder

Move the nudging data (the h1-files) over to a preferred folder and create a txt-file including a list of all the nudging data files that later can be pointed to as met\_filenames\_list: `ls -d -l $PWD/h1..nc > fileList.txt`

For more information, look into the file where most of the nudging code is found:

```
/components/cam/src/NorESM/fv/metdata.F90.
```

There are also other options for namelist modifications regarding nudging: [http://www.cesm.ucar.edu/models/cesm2/settings/current/cam\\_nml.html](http://www.cesm.ucar.edu/models/cesm2/settings/current/cam_nml.html) and search for “**met\_**”

## 1.4 NorESM2 output

### 1.4.1 Standard output

During a model run, each component (i.e. atm, lnd, ocn, cice, rof) produces its own output datasets consisting of history, restart and output log files. By default, each component periodically writes

- history files (usually monthly) in netCDF format
- writes netCDF or binary restart files in the RUNDIR directory. The history and log files are controlled independently by each component. History output control (i.e. output fields and frequency) is set in the Buildconf/component.buildnml.csh files.
- Component history files and restart files are in netCDF format.
- Restart files are used to either restart the model or to serve as initial conditions for other model cases.

For details about how to modify the user namelists in order to change the output, please see **User namelists** in *Experiment environments*.

Archiving is a phase of the model run where the generated output data is moved from RUNDIR to the archive folder. This job needs its own cpu time which is set in env\_batch.xml.

### Standard output for the model components in NorESM2



Lists of output from standard set-up model configurations:

- **CAM6-Nor** (atmosphere): cam\_standard\_out
- **CLM5** (land): clm\_standard\_out
- **MOSART** (river run off): mosart\_standard\_out
- **CICE** (sea-ice): cice\_standard\_out
- **BLOM** (ocean): blom\_standard\_out
- **iHAMOCC** (ocean biogeochemistry): hamocc\_standard\_out

## 1.4.2 Aerosol diagnostics and output

### Configuring a run with more aerosol diagnostics in NorESM2

To run NorESM2 with more aerosol diagnostics add to user\_nl\_cam:

```
&phys_ctl_nl
  history_aerosol = .true.
```

Adding history\_aerosol = .true. to user\_nl\_cam gives additional 577 variables (+ ca. 13 % CPU-time). Please see an overview of additional output variables: aerosol\_output\_history\_aerosol\_variables

### Decomposition of aerosol direct, semidirect and indirect radiative forcing

For effective radiative forcing estimates, multiple calls to the radiation code are necessary (see Ghan et al. 2012 for a detailed explanation). To enable this, copy the file preprocessorDefinitions.h from to the SourceMods/src.cam/ folder in your case directory:

```
cp <noresm_base>/components/cam/src/physics/cam_oslo/preprocessorDefinitions.h <case_
↪folder>/SourceMods/src.cam/.
```

To change the preprocessor definition to true, i.e. replace:

```
#undef AEROFFL
```

with:

```
#define AEROFFL
```

in the preprocessorDefinitions.h file.

The **AEROFFL**-token tells the model to do additional radiation-diagnostics for aerosol indirect effect. Including #define AEROFFL to preprocessorDefinitions.h gives 8 additionally variables (+ ca. 5% CPU-time). Please see an overview of the additional output variables: aerosol\_output\_aeroffl\_variables

## Enable diagnostics for AEROCOM

NorESM2 can be set up to take out additional aerosol output for use in AeroCom (<https://aerocom.met.no/index.html>) or other studies where there is a need for extensive aerosol diagnostics. To enable this, copy the file `preprocessorDefinitions.h` from to the `SourceMods/src.cam/` folder in your case directory:

```
cp <noresm_base>/components/cam/src/physics/cam_oslo/preprocessorDefinitions.h <case_
↪folder>/SourceMods/src.cam/.
```

To change the preprocessor definition to true, i.e. replace:

```
#undef AEROCOM
```

with:

```
#define AEROCOM
```

in the `preprocessorDefinitions.h` file.

If `#define AEROCOM` is activated, we additionally get 149 variables (+ ca. 13% CPU-time). Please see an overview of the additional output variables: `aerosol_output_aerocom_variables`

**The model may be run with any combination of these options: with AEROFFL only, with AEROCOM only, or with AEROFFL and AEROCOM activated at the same time.**

## References

Ghan, S.J., X. Liu, R.C. Easter, R. Zaveri, P.J. Rasch, J. Yoon, and B. Eaton, 2012: Toward a Minimal Representation of Aerosols in Climate Models: Comparative Decomposition of Aerosol Direct, Semidirect, and Indirect Radiative Forcing. *J. Climate*, 25, 6461–6476, <https://doi.org/10.1175/JCLI-D-11-00650.1>

### 1.4.3 COSP output

The CFMIP Observation Simulator Package (COSP) is an integrated part of the Community Atmosphere Model (CAM) and hence NorESM2. COSP calculates model cloud diagnostics that can be directly compared with satellite observations from ISCCP, CloudSat, CALIOP, MISR, and MODIS. The use of COSP facilitates “apples-to-apples” comparison of observed cloud data and model-simulated clouds, but an increase the run time of an experiment is expected (+ ca. 10% CPU-time).

To activate `cosp`, run `xmlchange` in the case folder (before building the model):

```
./xmlchange --append CAM_CONFIG_OPTS='-cosp'
```

or you can add `-cosp` to `CAM_CONFIG_OPTS` in `env_build.xml` (before building the model):

```
<entry id="CAM_CONFIG_OPTS" value="-phys cam6 -co2_cycle -chem trop_mam_oslo -cosp">
```

To the `user_nl_cam`, add:

```
&cospimulator_nl
  docosp      = .true.
  cosp_amwg   = .true.
```

Taking out COSP data, the following 57 output variables (of which 7 are 4 D) are added to the output (+ ca. 10% CPU-time): `cosp_extra_output`

For a detailed description of COSP, please see <https://climatedataguide.ucar.edu/climate-data/cosp-cloud-feedback-model-intercomparison-project-cfmip-observation-simulator-package>

## 1.5 NorESM2 diagnostics

### 1.5.1 ESMValTool

The **Earth System Model eValuation Tool (ESMValTool)** is a community-development that aims at improving diagnosing and understanding of the causes and effects of model biases and inter-model spread. The ESMValTool can process multiple model outputs, including NorESM, but these model output need to be post-processed to conform with the CMIP data standard (i.e., **CMOR**). ESMValTool support all the published model data output on the **ESGF**

The `esmvaltool-on-nird` discussion group can be a useful resource if you plan to run ESMValTool on Nird.

#### Run ESMValTool on NIRD service node

The ESMValTool is currently installed under the dedicated IPCC service node (`<ipcc.nird.sigma2.no>`) for data post-processing. The following NIRD project areas are mounted under `/projects` of the ```IPCC``` node: NS2345K, NS9034K, NS9039K, NS9252K, NS9560K, NS9588K.

#### Activate and run ESMValTool

An example of steps to run the ESMValTool on `ipcc.nird.sigma2.no`

1. login in the ipcc node:

```
ssh -l username ipcc.nird.sigma2.no
```

2. load ESMValTool on nird:

```
conda activate /conda/esmvaltool/2.0.0/
```

(note, the `esmvaltool` may be upgraded in the future. Therefore, use `ls /conda/esmvaltool/` to check the currently installed vesion if 2.0.0 does not exist.)

3. run initialisation:

```
esmvaltool config get_config_user
```

4. copy the following config file to `~/.esmvaltool/`:

```
cp /projects/NS9252K/share/esmvaltool/config/config-ipcc.yml ~/.esmvaltool/
```

You may copy and modify the `config-ipcc.yml` file to store some intermediate data files and final plots to your own directory.

5. copy a recipe to `~/your-recipes/`:

```
mkdir ~/your-recipes
cp /projects/NS9252K/share/esmvaltool/recipes/recipe_validation_CMIP6.yml ~/your-
↳ recipes/
```

### 6. run esmvaltool:

```
esmvaltool run --config_file .esmvaltool/config-ipcc.yml ./your-recipes/recipe_
↪validation_CMIP6.yml
```

### 7. Check for results, sample results are under:

<http://ns2345k.web.sigma2.no/diagnostics/esmvaltool/>

## Download data automatically with Synda

**Synda** is a tool to download and manage model data form the **ESGF**, it can be called by the `esmvaltool` as a command line option `--synda` so that it can automatically download necessary model data as specified in the recipe of **ESMValTool**. For example,

```
esmvaltool -c config_heyc.yml ./recipe_seaice.yml --synda
```

Note, it is **OPTIONAL** to use `synda` to download data automatically during the runtime of **ESMValTool**. One can download the data by some other tools or scripts before running the **ESMValTool**.

If you do want to use `synda`, there are two major steps you need to do:

### Activate the tool

It is install with `conda`, so you can add `synda` executable to your search path, e.g.,

```
export PATH:$PATH:/projects/NS9252K/conda/synda/bin/synda
```

Then it should be available in your command window by the `synda` command.

### Set up your credentials

1. register (one of) the **ESGF node** and **Globus transfer** (not toally sure if account on globus is mandatory if this option will not be used, but seem yes according to my experience, you can firstly try without it)
2. paste your username and password to in `~/ .synda/conf/credentials.conf`
3. configure `synda` parameters in `~/ .synda/conf/sdt.conf`, use my setting as template.
4. maybe you need get *globus token*. Run the following command:

```
synda token -p globus renew
```

it will give out something like:

```
Native App Authorization URL:
https://auth.globus.org/v2/oauth2/authorize?code_challenge=BrmiBhFVVuHVNyGDj6hn5N8M1-
↪EKJNnNgptobIsbTqI&state=_default&redirect_uri=https%3A%2F%2Fauth.globus.org%2Fv2
↪%2Fweb%2Fauth-code&response_type=code&client_id=83ec00c1-e67a-4356-9f1f-
↪f7e31177e31a&scope=openid+email+profile+urn%3Aglobus%3Aauth%3Ascope%3Atransfer.api.
↪globus.org%3Aall&code_challenge_method=S256&access_type=offline
Enter the auth code:
```

Paste the above `https` address to browser, and you will find a authen code, and past back to the command line.

## Run ESMValTool on NIRD toolkit service

The ESMValTool is also available from [NIRD Toolkit](#) applications (Jupyter and JupyterHub), provided by Docker images.

[NIRD Toolkit](#) is a cloud infrastructure that gives access to compute nodes on Nird. The service is managed by [Kubernetes](#) which launches applications from [Docker containers](#).

Access to the NIRD Toolkit service is provided according to a 3 tier ranking

1. **Owner** : a project leader/executive officer of a NSxxxxK project
2. **Admin** : a person authorized by an **owner** to manage NIRD Toolkit applications
3. **Member** : a person authorized by the **owner** or **admin** to access a NIRD Toolkit application

In order to use NIRD Toolkit, an **owner** of a NSxxxxK project should create a group in [dataporten](#), and invite **members** to join, who may be given **admin** status by the **owner** or existing **admin** people. See [Sigma2](#) information page about [deploying service](#) for details about the procedure.

## Install and run an application

Applications in NIRD Toolkit are available from:

<https://apps.sigma2.no/nird>

An owner/admin can launch a new application by installing it, and request resources to be made available to the application via the Kybernetes management system. A member can launch an existing application that has previously been set up by an owner/admin. The application setup allows access to storage areas under NSxxxxK storage volumes (read-only by default) and a user storage area under a specific NSxxxxK/subfolder with write access, but only applications pre-defined in the docker image provided to the Kybernetes system can be used.

## Install ESMValTool docker image

ESMValTool is not included in the default docker images provided by Sigma2, but has been installed in modified docker images. These are created by building ESMValTool on top of an official Sigma2 docker image, and package in a new docker container. Such modified docker images are available for *jupyter* and *jupyterhub* applications.

1. **jupyterhub**: nordicesmhub/jupyterhub-nird-toolkit
  - *source* : <https://github.com/NorESMhub/jupyterhub-nird-toolkit>
  - *docker* : <https://hub.docker.com/r/nordicesmhub/jupyterhub-nird-toolkit>  
nordicesmhub/jupyterhub-nird-toolkit:latest
2. **jupyterhub**: tomastorsvik/nird\_jupyterhub-singleuser\_esmvaltool
  - *source* : <https://github.com/TomasTorsvik/jupyterhub-nird-toolkit>
  - *docker* : [https://hub.docker.com/repository/docker/tomastorsvik/nird\\_jupyterhub-singleuser\\_esmvaltool](https://hub.docker.com/repository/docker/tomastorsvik/nird_jupyterhub-singleuser_esmvaltool)  
tomastorsvik/nird\_jupyterhub-singleuser\_esmvaltool:latest
3. **jupyter**: tomastorsvik/nird\_jupyter-spark\_esmvaltool
  - *source* : <https://github.com/TomasTorsvik/jupyter-spark-nird-toolkit>
  - *docker* : [https://hub.docker.com/repository/docker/tomastorsvik/nird\\_jupyter-spark\\_esmvaltool](https://hub.docker.com/repository/docker/tomastorsvik/nird_jupyter-spark_esmvaltool)

```
tomastorsvik/nird_jupyter-spark_esmvaltool:latest
```

To install the docker image you need access to the Nird Toolkit service through a project, and install a new instance of e.g. “jupyter”. On the installation page you select whatever standard settings you like (application name, projectspace, persistent storage, machine type), and then select “Show advances configuration. . .” and replace the standard docker-image with:

```
tomastorsvik/nird_jupyter-spark_esmvaltool:latest
```

One can activate the “JupyterLab” option in the configuration (but it may not necessary to do so to run ESMValTool). To test if the image is installed correctly (provided the jupyter notebook environment builds without errors) you can open a terminal in the environment and type `esmvaltool -h`, which should give you the help page for the tool in the terminal window.

As the development of ESMValTool and the updating of the NIRD Toolkit base system are not synchronous, there may sometimes be conflicts in the package dependencies of what ESMValTool wants and what NIRD Toolkit provides. Hence, it is sometimes necessary to build `esmvaltool` in a separate `conda` environment (not “base”). If `esmvaltool -h` fails to give the expected output, check if there exist any alternative environments by doing

```
conda env --list
```

If you find an `esmvaltool` environment, this can be activated by

```
source activate esmvaltool
```

At the time of writing (10 June 2020), the preferred activation method `conda activate esmvaltool` is not recognized inside a NIRD Toolkit application.

## 1.5.2 NorESM2 Diagnostics Package

### Introduction

**The NorESM Diagnostic Package:** is a NorESM model evaluation tool written with a set of scripts (bash, NCL etc) to provide a general evaluation and quick preview of the model performance with only one command line. This toolpackage works on the original model output and has NorESM-specific diagnostics.

**The tool package consists of:**

- CAM\_DIAG: (NCAR’s AMWG Diagnostics Package)
- CLM\_DIAG: (CESM Land Model Diagnostics Package)
- CICE\_DIAG: snow/sea ice volume/area
- HAMOCC\_DIAG: time series, climatology, zonal mean, regional mean
- BLOM\_DIAG: time series, climatologies, zonal mean, fluxes, etc

### NorESM diagnostics on GitHub

The source codes of the NorESM diagnostics packages are developed and maintained in the Git version control repository: <https://github.com/NordicESMhub/noresmdiagnosics>.

And the observation dataset and grid files are hosted at: <https://www.noresm.org/diagnostics>, with a total size of ~100 GB.

### NorESM diagnostics on NIRD

The full diagnostic package (including source files and data files) are currently hosted on NIRD: `/projects/NS2345K/diagnostics/noresmdiagnosics`

## Installation

### Use the preinstalled package

You don't need to install this diagnostic package, but you can call it as a command line directly on NIRD. As a prerequisite, you should have access permission to the NS2345K project on NIRD. There is no need to install the diagnostic packages, but just add the `diag_run` to your search path, or add it as an alias in `$HOME/.bashrc` (assuming you are using Bash Shell),

```
alias diag_run="/projects/NS2345K/diagnostics/noresmdiagnosics/bin/diag_run"
```

`source ~/.bashrc` or close and reopen your terminal to make these changes take effect.

**DO NOT** make changes directly in this preinstalled package.

### Clone and run your own copy

If you want to make some changes of the diagnostic package for your own purpose or/and want to contribute to the development of it, you can install it on NIRD under your personal folder or your own project area (i.e., `/projects/NSxxxxK`). Briefly, there are several steps to install it:

1. Fork the NorESM Diagnostic Package [Github repository](https://github.com/NORESM/noresmdiagnosics) to your own Github repository. For example, [https://github.com/YOU\\_GITHUB\\_USERNAME/noresmdiagnosics](https://github.com/YOU_GITHUB_USERNAME/noresmdiagnosics)
2. Change to your preferred location, say `DIAGROOT`, where you want to install the tool, and `git clone https://github.com/YOU_GITHUB_USERNAME/noresmdiagnosics`
3. Change to `$DIAGROOT/bin`, and link or download all the observation and grid data files.
  - If you are installing the tool on NIRD, you just need to link all the data to your clone by running the script `linkdata.sh`, given you have access to the `/project/NS2345K` project
  - If you are not remember of NS2345K or you are installing it on platforms other than NIRD, you should download all the data to your clone by executing `dloaddata.sh`. If you are not running it on NIRD, you should have CDO, NCO and NCL installed.
4. Make changes to the code/scripts for your purpose. And call `diag_run` of your own clone.
5. If you would like to contribute your function enhancements or bug fixes to the original diagnostic package, you should firstly make these changes in a new git branch, and commit the changes to your fork repository, then create an Issue at the [Github repository](https://github.com/NORESM/noresmdiagnosics), and finally make a `pull` request to the original Github repository to incorporate your changes.

## Run the diagnostic tool

Each package can be run/configured from the command line using the wrapper script for NorESM diagnostics program `diag_run`:

```

-----
Program:
/projects/NS2345K/noresm_diagnostics/bin/diag_run
Version: 2.0
-----

Short description:
A wrapper script for NorESM diagnostic packages.

Basic usage:
diag_run -m [model] -c [test case name] -s [test case start yr] -e [test case end yr]
↳ # Run model-obs diagnostics
diag_run -m [model] -c [test case name] -s [test case start yr] -e [test case end yr]
↳ -c2 [cntl case name] -s2 [cntl case start yr] -e2 [cntl case end yr] # Run model1-
↳ model2 diagnostics
nohup /projects/NS2345K/noresm_diagnostics/bin/diag_run -m [model] -c [test case
↳ name] -s [test case start yr] -e [test case end yr] &> out & # Run model-obs
↳ diagnostics in the background with nohup

Command-line options:
-m, --model=MODEL                Specify the diagnostics package
↳ (REQUIRED).

Valid arguments:
    cam      : atmospheric package (AMWG)
    clm      : land package (LMWG)
    cice     : sea-ice package
    blom     : ocean package
    hamocc   : biogeochemistry package
    all      : configure all available
↳ packages.
-c, -c1, --case=CASE1, --case1=CASE1    Test case simulation (OPTIONAL).
-s, -s1, --start_yr=SYR1, --start_yr1=SYR1 Start year of test case climatology
↳ (OPTIONAL).
-e, -e1, --end_yr=EYR1, --end_yr1=EYR1   End year of test case climatology
↳ (OPTIONAL).
-c2, --case2=CASE2                    Control case simulation (OPTIONAL).
-s2, --start_yr2=SYR2                  Start year of control case climatology
↳ (OPTIONAL).
-e2, --end_yr2=EYR2                    End year of control case climatology
↳ (OPTIONAL).
-i, -i1, --input-dir=DIR, --input-dir1=DIR Specify the directory where the test
↳ case history files are located (OPTIONAL).
Default is --input-dir=/projects/
↳ NS2345K/noresm/cases
-i2, --input-dir2=DIR                    Specify the directory where the control
↳ case history files are located (OPTIONAL).
Default is --input-dir=/projects/
↳ NS2345K/noresm/cases
-o, --output-dir=DIR                    Specify the directory where the
↳ package(s) the climatology and time-series files should be stored (OPTIONAL).
Default is --output-dir=/projects/
↳ NS2345K/noresm_diagnostics/out/$USER
-p, --passive-mode                       Run the script in passive mode: the
↳ diagnostic script will be configured but not executed (OPTIONAL).

```

(continues on next page)



(continued from previous page)

```

-t, --type=TYPE                Specify climatology or time series.
↳diagnostics (OPTIONAL): valid options are --type=climo and --type=time_series.
                               Default is to run both. Note that the
↳time series are computed over the entire simulation.
-w, --web-dir=DIR              Specify the directory where the html
↳should be published (OPTIONAL).
                               Default is --web-dir=/projects/NS2345K/
↳www/noresm_diagnostics
--no-atm                       Run CLM diagnostics without CAM data.
↳Must be used for offline CLM simulations.

```

## Description

`diag_run` is a wrapper script, which is used to run the diagnostics for each NorESM component (cam, clm, cice, blom, and hamocc). The diagnostic packages can be used to plot model results with respect to either observations (so-called model-obs diagnostics), or to another simulation (model1-model2 diagnostics). The diagnostics for the atmosphere (cam), land (clm) and sea-ice (cice) are based on the NCAR packages, but has undergone some major improvements, particularly in the climatology and time-series computations. The ocean (blom) and its biogeochemistry (hamocc) have been developed in-house.

Please note, the ocean component of the NorESM2, BLOM, is an updated version of MICOM. It is named MICOM in NorESM1 for CMIP5 experiments and in NorESM2 for many (but not all) CMIP6 experiments. Therefore, for experiments with MICOM as the ocean component of NorESM, `-m micom` can be used in the command line option for `diag_run`. To compare a simulation with either MICOM or BLOM to the other (model1-model2 diagnostics), you can either use `-m micom` or `-m blom`. Both options should work.

`diag_run` has two modes:

- an “active-mode”, for which `diag_run` runs the diagnostic scripts
- a “passive-mode”, for which `diag_run` only configures the scripts.

In the passive-mode the diagnostic scripts have to be run manually by the user. By default, `diag_run` is always in the active-mode, but switches into passive-mode if at least one of these two criteria are fulfilled:

1. The user invokes the option `-p` (see below), or
2. The user does not give enough information needed to run the diagnostics (next subsection).

## Active-mode

If you want to use `diag_run` to run the full (climatology and time-series) diagnostics, the minimum requirement is to specify the options `model`, `case_name`, `start_yr` and `end_yr` (`-m`, `-c`, `-s` and `-e`), e.g.:

```
diag_run -m cam -c N1850_f19_tn14_191017 -s 21 -e 50
```

This command runs atmospheric model-obs diagnostics of the case `N1850_f19_tn14_191017` using a climatology between model years 21 and 50. It is assumed that the `N1850_f19_tn14_191017` history files are located in `/projects/NS2345K/noresm/cases`. By default, the resulting plots and html will be stored in

```

/projects/NS2345K/www/diagnostics/noresmdiagnosics/$USER/N1850_f19_tn14_191017/CAM_
↳DIAG,

```

or, if you specify to store them under a command folder, i.e. with `-w /projects/NS2345K/www/diagnostics/noresmdiagnosics/common`. It links to links to the following URL: [http://ns2345k.web.sigma2.no/diagnostics/noresmdiagnosics/common/N1850\\_f19\\_tn14\\_191017/CAM\\_DIAG/yrs21to50-obs.html](http://ns2345k.web.sigma2.no/diagnostics/noresmdiagnosics/common/N1850_f19_tn14_191017/CAM_DIAG/yrs21to50-obs.html).

The climatology and time-series files in `/projects/NS2345K/diagnostics/noresmdiagnosics/out/$USER/CAM_DIAG` (where `$USER` is your NIRD username).

If you want to run model1-model2 diagnostics, you also need to specify `case_name2`, `start_yr2` and `end_yr2` (`-c2`, `-s2`, `-e2`) in addition, i.e.:

```
diag_run -m cam -c N1850_f19_tn14_191017 -s 21 -e 50 -c2 B1850MICOM_f09_tn14_01 -s2_
↪21 -e2 50
```

would be the same as in Example 1 above, except for comparing `N1850_f19_tn14_191017` to `B1850MICOM_f09_tn14_01` instead of observations.

In Example 1 and Example 2 the options `-s` and `-e` (as well as `-s2`, `-e2`) refer to the start and end years of the climatology. The time-series are calculated from all the history files in the case directory (`input_dir`). This is always the case unless the user invokes the option `-t time_series`. If this option is invoked, `start_yr` and `end_yr` refer to the beginning and end of the time series instead of the climatology, hence:

Example 3:

```
diag_run -m blom -c N1850_f19_tn14_blom_20200608 -t time_series -s 1 -e 10
```

would produce `blom` time-series plots between years 1 and 20. Note that omitting `start_yr` and `end_yr` when the option `-t time_series` is invoked computes the time-series over the entire experiment (all history files in the case directory, `input_dir`):

```
diag_run -m cam -c N1850_f19_tn14_191017 -t time_series
```

`diag_run` uses some template scripts for each of the model components. When `diag_run` is executed, these scripts are changed according to the user-specified settings and renamed with a time stamp. For example, if you run the `blom` diagnostics, the run script template (`blom_diag_template.sh`) will be renamed with a time-stamp as `blom_diag_YYMMDD_HHMMSS`.

`diag_run` also creates a config and output file with the same time stamp (`config_YYMMDD_HHMMSS` and `out_YYMMDD_HHMMSS`, respectively). The config file stores information about changes in the diagnostics scripts invoked by the user, and the output file contains the standard output and error (i.e. what is shown in your terminal during runtime). When the diagnostics a component is finished the run scripts are copied to:

```
output_dir/$USER/model_diag/config/case_name/run_scripts
```

and the config and output files to:

```
output_dir/$USER/model_diag/config/case_name/logs
```

Hence, for Example 1 above, the run scripts are saved in:

```
/projects/NS2345K/diagnostics/noresmdiagnosics/out/ $USER/CAM_DIAG/config/N1850_f19_
↪tn14_191017/run_scripts
```

and the config and out files in:

```
/projects/NS2345K/diagnostics/noresmdiagnosics/out/$USER/CAM_DIAG/config/N1850_f19_
↪tn14_191017/logs
```

## Passive-mode

Another important property of `diag_run` is that it will only run the diagnostics if sufficient information has been provided by the user; otherwise it switches into passive-mode. `diag_run` will then configure the diagnostics scripts as much as possible (based on the information provided by the user), and also add information to the config file about which variables are still required to be modified by the user in order to run the diagnostic script. This option is particularly useful if you want to do some development work on the diagnostics scripts, or if you want to change any variables in the diagnostics scripts that are not included as an option in `diag_run`. Hence, if you run the following command:

```
diag_run -m clm
```

the following will appear on the screen:

```
[nird@login0 ~]$ /projects/NS2345K/diagnostics/noresmdiagnosics/diag_run -m clm
-----
Program:
/projects/NS2345K/noresm_diagnostics/bin/diag_run
Version: 2.0
-----
-CHANGING DIAGNOSTICS DIRECTORY to
/projects/NS2345K/diagnostics/noresmdiagnosics/out/xxx/CLM_DIAG in lnd_template.csh
-CHANGING ROOT DIRECTORY FOR CODE AND DATA to
/projects/NS2345K/diagnostics/noresmdiagnosics/packages/CLM_DIAG in lnd_template.csh
-CHANGING INPUT DIR 1 to /projects/NS2345K/noresm/cases in lnd_template.csh
-CHANGING publish_html_root to /projects/NS2345K/www/diagnostics/noresmdiagnosics in
lnd_template.csh
-SETTING UP TIME-SERIES DIAGNOSTICS FOR ENTIRE EXPERIMENT
CLM DIAGNOSTICS SUCCESSFULLY CONFIGURED in
/projects/NS2345K/diagnostics/noresmdiagnosics/out/xxx/CLM_DIAG
-----
lnd_template.csh IS NOT RUNNING: NOT ALL REQUIRED VARIABLES HAVE BEEN CONFIGURED
(see /projects/NS2345K/diagnostics/noresmdiagnosics/out/xxx/CLM_DIAG/config.log) .
-----
TOTAL diag_run RUNTIME: 0m2s
-CLM diagnostics: 0m2s
-----
DONE: fr. 20. april 15:37:42 +0200 2018
```

The (semi-configured) run script has then been copied to `/projects/NS2345K/diagnostics/noresmdiagnosics/out/<username>/CLM_DIAG` and all information about the configuration is contained in `/projects/NS2345K/diagnostics/noresmdiagnosics/out/<username>/CLM_DIAG`.

## Options

`diag_run` options (flags) typically come in both short (single-letter) and long forms. A complete description of all options is given below in alphabetical order of the short option letter. When invoked without options, `diag_run` prints a table containing all options along with some examples (see also below).

```
-c case_name (-c1, --case, --case1)
```

Name of the test case experiment that you want to run diagnostics for. This option is required if you want to use `diag_run` in active-mode.

```
-c2 case_name2 (--case2)
```

Name of the control case experiment. This option is required if you want to run model1-model2 diagnostics in active-mode.

```
-e end_year (-e1, --end_yr, --end_yr1)
```

If `-type=time_series`, this option refers to the end year of time-series for `case_name`. Otherwise, it refers to the end year of climatology. This option is optional if `-type=time_series`, but required for active-mode diagnostics if `-type=climo` or if `type` is not invoked.

```
-e2 end_year (--end_yr2)
```

If `-type=time_series`, this option refers to the end year of time-series for `case_name2`. Otherwise, it refers to the end year of climatology. This option is optional if `-type=time_series`, but required for active-mode model1-model2 diagnostics if `-type=climo` or if `type` is not invoked.

```
-i input_dir (-i1, --input-dir, --input-dir1)
```

Name of the root directory of the monthly history files for `case_name`. For example, if your blom history files are located in `/this/is/a/directory/case1/ocn/hist`, this option should be set to `input_dir=/this/is/a/directory`. Default is `input_dir=/projects/NS2345K/noesm/cases`.

```
-i2 input-dir2 (--input-dir2)
```

Name of the root directory of the monthly history files for `case_name2`. Also here, default is `input_dir2=/projects/NS2345K/noesm/cases`.

```
-m model (--model)
```

Name of the model you want to run the diagnostics for. Valid options are `cam`, `clm`, `cice`, `blom`, `hamocc` and `all`. This is the only option that is required for both the active and passive mode. If you invoke the “all” option, the `cam`, `clm`, `cice`, `blom` and `hamocc` diagnostics will be run subsequently. It is also possible to combine different models as you wish within this option: for example, if you only want to run `cam` and `clm` diagnostics, you can simply add the names of those models and separate them with a comma (`-m cam,clm`).

```
--no-atm
```

This option, which takes no argument, skips the usage of CAM history files in the CLM diagnostics. This option is necessary for offline CLM simulations.

```
-o output_dir (--output_dir)
```

Root directory where you want to store the output from the diagnostics (i.e. the climatology and time-series files). For example, if you set `output_dir=/just/another/directory`, the climatology and time-series files from the blom diagnostics will be stored in:

```
/just/another/directory/BLOM_DIAG/.
```

Default is:

```
output_dir=/projects/NS2345K/diagnostics/noesmdiagnosics/out/$USER
```

where `$USER` is your user name on NIRD.

```
-p, --passive-mode
```

This option, which takes no argument, forces `diag_run` into passive-mode. This means, even if you have given sufficient information to run in active-mode, the diagnostic scripts will not be executed.

```
-s start_year (-s1, --start_yr, --start_yr1)
```

If `-type=time_series`, this option refers to the start year of time-series for `case_name`. Otherwise, it refers to the start year of climatology. This option is optional if `-type=time_series`, but required for active-mode diagnostics if `-type=climo` or if `type` is not invoked.

```
-s2 start_year2 (--start_yr2)
```

If `-type=time_series`, this option refers to the start year of time-series for `case_name2`. Otherwise, it refers to the start year of climatology. This option is optional if `-type=time_series`, but required for active-mode model1-model2 diagnostics if `-type=climo` or if `type` is not invoked.

```
-t type (--type)
```

Specifies if you only run climatology or time-series diagnostics: valid options are `-type=climo` and `-type=time_series`. Default is to run both.

```
-w webdir (--web-dir)
```

Specifies the directory where the html should be stored. This directory should preferably be linked to a web server so that one can look at the results with a web browser. Default is:

```
--web-dir=/projects/NS2345K/www/diagnostics/noresmdiagnosics/
```

## Examples

Model-obs diagnostics of `case=N1850_f19_tn11_exp1` (climatology between yrs 21 and 50) for all model components:

```
diag_run -m all -c N1850_f19_tn11_exp1 -s 21 -e 50
```

Model-obs diagnostics in CAM, publish the html in `/path/to/my/html`:

```
diag_run -m cam -c N1850_f19_tn11_exp1 -s 21 -e 50 -w /path/to/my/html
```

Model-obs time-series diagnostics in BLOM for all years the model output directory (`/projects/NS2345K/noesm/cases/N1850_f19_tn14_blom_20200608/ocn/hist/`):

```
diag_run -m blom -c N1850_f19_tn14_blom_20200608 -t time_series
```

Configure (but do not run) model-obs diagnostics for CICE:

```
diag_run -m cice -c N1850_f19_tn11_exp1 -s 21 -e 50 -p
```

Model1-model2 diagnostics for CLM with user-specified history file directories:

```
diag_run -m clm -c N1850_f19_tn11_exp1 -s 21 -e 50 -i /input/directory1 -c2
```

`N1850_f19_tn11_exp2 -s2 21 -e2 50 -i2 /input/directory2` Model-obs climatology diagnostics (no time series) for BLOM:

```
diag_run -m blom -c N1850_f19_tn14_blom_20200608 -s 1 -e 10 -t climo
```

Install CAM diagnostics in `/my/dir` with minimal configuration:

```
diag_run -m cam -o /my/dir
```

Model-obs diagnostics for BLOM and HAMOCC:

```
diag_run -m blom,hamocc -c N1850_f19_tn14_blom_20200608 -s 1 -e 10
```

Model-obs time-series diagnostics for an offline (uncoupled) CLM simulation:

```
diag_run -m clm -c N1850_f19_tn11_clmexpl -s 71 -e 100 --no-atm
```

Model-obs time-series diagnostics in HAMOCC between yrs 31 and 100:

```
diag_run -m hamocc -c N18500C_f19_tn11_expl -s 31 -e 100 -t time_series
```

### 1.5.3 Aerosol diagnostics

For utilizing the comparison package described below, NorESM2 should be run with extra aerosol diagnostics and output, i.e. adding `history_aerosol = .true.` in `user_nl_cam` and enabling AROFFL and AEROCOM in preprocessor-Definitions.h. For a more detailed description, please see: [Aerosol diagnostics and output](#)

#### NCL Model Version Comparison package ModIvsModII

ModIvsModII produces plots and global life-cycling data (for use in a table) of often used aerosol, cloud and radiative flux fields, including ERFs. This is done for two model versions to be compared, either CAM4-Oslo vs. CAM6-Nor or CAM6-Nor vs. CAM6-Nor (but late versions of CAM5.3-Oslo can replace CAM6-Nor), and for PD and PI conditions, i.e. 4 simulations. If only e.g. the PD simulations are available, one can replace the PI data with those in the input to ModIvsModII, as long as one keeps that in mind when interpreting the shown diagnostics. ModI and ModII can also be the same, if you need to use these diagnostics but only have one model simulation for PD or PI or both.

- Make a local copy (on Linux) of the directory `models/atm/cam/tools/diagnostics/ncl/ModIvsModII`
- Prepare climatological average monthly files of the (up to 4) simulations you want to compare, e.g. by use of the script `create-clim.sh`
- Assuming that the simulations have been run with `history_aerosol = .true.` and `#define AEROCOM & AROFFL` (otherwise only some of the diagnostics will be produced):
  - In `ModIvsModII.csh` (note: read the header info):
    - \* edit model info for the first model (shown to the left in the plots): `modelI = CAM4-Oslo` or `modelI = CAM5-Oslo` (or later) ?
    - \* provide paths and partial file names of the model data (PD and PI) for Model I (CAM4-Oslo or CAM5-Oslo) and Model II (must be CAM5-Oslo or later)
    - \* choose desired plot format (`plotf=ps, eps, pdf` or `png`)
    - \* Run the script: `./ModIvsModII.csh`
- Furthermore, to display the plots in an organized form by use of a web browser (only possible if the chosen plot format is `png`):
  - edit general model info (only) in `ModIvsModII.htm`, and manually cut and paste the mass budget numbers from the script output into this file (bottom part)
  - copy all `png` (plots) and `htm` files to the desired output (common) directory
  - open `ModIvsModII.htm` in your browser: hyper-links to all other `htm` files, including plots, are found here

Example: [http://ns2345k.web.sigma2.no/aerosol\\_diagnostics/NFHISTnorppdmsbcodyn\\_f09\\_mg17\\_20191101\\_vs\\_NHIST\\_f19\\_tn14\\_20190710/ModIvsModII.htm](http://ns2345k.web.sigma2.no/aerosol_diagnostics/NFHISTnorppdmsbcodyn_f09_mg17_20191101_vs_NHIST_f19_tn14_20190710/ModIvsModII.htm)

## 1.6 NorESM2 Model Description

### 1.6.1 Overview

The Norwegian Earth System Model version 2 (NorESM2; Seland et al., in review for GMD) is the second generation of the coupled Earth System Model developed by the Norwegian Climate Center (NCC), and is the successor of NorESM1 (Bentsen et al., 2013; Iversen et al., 2013; Kirkevåg et al., 2013; Tjiputra et al., 2013) which has been used in the 5th phase of the Coupled Model Intercomparison Project (CMIP5; Taylor et al., 2012), and for evaluation of the difference between a 1.5 and 2 deg. C warmer world than pre-industrial (Graff et al., 2019). NorESM2 is based on the Community Earth System Model CESM2.1 (Danabasoglu et al., 2019). Although large parts of NorESM are similar to CESM, there are several important differences: NorESM uses the isopycnic coordinate Bergen Layered Ocean Model (BLOM; Bentsen et al., in prep.), uses a different aerosol module OsloAero6 (Kirkevåg et al., 2018; Olivíe et al., in prep.), contains specific modifications and tunings of the atmosphere component (Toniazzi et al., 2019; Toniazzi et al., in prep.), and contains the iHAMOCC model to describe ocean biogeochemistry (Tjiputra et al., 2019).

Many changes have contributed to the development of NorESM1 into NorESM2. The model has benefited from the evolution of the parent model CCSM4.0 into CESM2.1, comprising the change of the atmosphere component from CAM4 to CAM6, the land component from CLM4 to CLM5, and the sea ice component from CICE4 to CICE5. Also, specific developments have been implemented in the description of aerosols and their coupling to clouds and radiation (Kirkevåg et al., 2018), in addition to harmonizing the implementation of the aerosol scheme with the standard aerosol schemes in CESM. To extend the capabilities of NorESM as an Earth System Model, a strong focus has been put on the interactive description of natural emissions of aerosols and their precursors, and tightening the coupling between the different Earth System components. Finally, the ocean model (Bentsen et al., in prep.) and the ocean biogeochemistry module (Schwinger et al., 2016; Tjiputra et al., 2019) have been further developed.

### References

Seland, Ø., Bentsen, M., Seland Graff, L., Olivíe, D., Toniazzi, T., Gjermundsen, A., Debernard, J. B., Gupta, A. K., He, Y., Kirkevåg, A., Schwinger, J., Tjiputra, J., Schancke Aas, K., Bethke, I., Fan, Y., Griesfeller, J., Grini, A., Guo, C., Ilicak, M., Hafsaal Karset, I. H., Landgren, O., Liakka, J., Onsum Moseid, K., Nummelin, A., Spensberger, C., Tang, H., Zhang, Z., Heinze, C., Iversen, T., and Schulz, M.: The Norwegian Earth System Model, NorESM2 – Evaluation of the CMIP6 DECK and historical simulations, *Geosci. Model Dev. Discuss.*, <https://doi.org/10.5194/gmd-2019-378>, in review, 2020.

Tjiputra, J. F., Schwinger, J., Bentsen, M., Morée, A. L., Gao, S., Bethke, I., Heinze, C., Goris, N., Gupta, A., He, Y., Olivíe, D., Seland, Ø., and Schulz, M.: Ocean biogeochemistry in the Norwegian Earth System Model version 2 (NorESM2), *Geosci. Model Dev. Discuss.*, <https://doi.org/10.5194/gmd-2019-347>, in review, 2020.

### 1.6.2 The atmosphere model

The atmospheric model component of NorESM2, **CAM6-Nor**, (Seland et al., in review for GMD) is built on the CAM6 version from CESM2.1, but with particulate aerosols and the aerosol-radiation-cloud interaction parameterisation from NorESM1 and NorESM1.2 as described by Kirkevåg et al. (2013, 2018). NorESM2-specific changes to model physics and dynamics which are not aerosol related, are described by Toniazzi et al. (2019) and Toniazzi et al. (in prep.). The latest updates in the aerosol modules (that is, the changes between NorESM1.2 and NorESM2) are described by Olivíe et al. (in prep.).

One particular part of the CAM6-Nor which is not described in great detail in any of the most recent papers (2020) is the offline sectional aerosol model AeroTab6. This is based on AeroTab5.3 (Kirkevåg et al., 2018), but uses updated



complex refractive indexes for mineral dust for wavelengths below 15 m, changed according to recent research (for details, see Oliv   et al. (in prep.)). A user guide for this model code is found here: [https://github.com/NorESMhub/NorESM/blob/norasm2/doc/configurations/AeroTab-user-guide\\_v16april2020.pdf](https://github.com/NorESMhub/NorESM/blob/norasm2/doc/configurations/AeroTab-user-guide_v16april2020.pdf), see also the chapter about aerosol specific input data in Sect 5.3 of <https://noresm-docs.readthedocs.io/en/norasm2/configurations/input.html>.

## References

Seland,  ., Bentsen, M., Seland Graff, L., Oliv  , D., Toniazzo, T., Gjermundsen, A., Debernard, J. B., Gupta, A. K., He, Y., Kirkev  g, A., Schwinger, J., Tjiputra, J., Schancke Aas, K., Bethke, I., Fan, Y., Griesfeller, J., Grini, A., Guo, C., Ilicak, M., Hafsaal Karset, I. H., Landgren, O., Liakka, J., Onsum Moseid, K., Nummelin, A., Spensberger, C., Tang, H., Zhang, Z., Heinze, C., Iversen, T., and Schulz, M.: The Norwegian Earth System Model, NorESM2 – Evaluation of the CMIP6 DECK and historical simulations, *Geosci. Model Dev. Discuss.*, <https://doi.org/10.5194/gmd-2019-378>, in review, 2020.

Toniazzo, T., Bentsen, M., Craig, C., Eaton, B. E., Edwards, J., Goldhaber, S., Jablonowski, C., and Lauritzen, P. H.: Enforcing conservation of axial angular momentum in the atmospheric general circulation model CAM6, *Geosci. Model Dev.*, 13, 685–705, <https://doi.org/10.5194/gmd-13-685-2020>, 2020.

Kirkev  g, A., Grini, A., Oliv  , D., Seland,  ., Alterskj  r, K., Hummel, M., Karset, I. H. H., Lewinschal, A., Liu, X., Makkonen, R., Bethke, I., Griesfeller, J., Schulz, M., and Iversen, T.: A production-tagged aerosol module for Earth system models, OsloAero5.3 – extensions and updates for CAM5.3-Oslo, *Geosci. Model Dev.*, 11, 3945–3982, <https://doi.org/10.5194/gmd-11-3945-2018>, 2018.

### 1.6.3 The ocean model

The ocean component Bergen Layered Ocean Model (**BLOM**; Bentsen et al., 2020, *in prep.*) employs an isopycnic vertical coordinate, with near-isopycnic interior layers and variable density layers in the surface mixed boundary layer. BLOM originates from the Miami Isopycnic Coordinate Ocean Model (MICOM; Bleck and Smith, 1990; Bleck et al., 1992), with largely modified codes compared to the original MICOM (Bentsen, 2013; 2020, *in prep.*).

- **C-grid discretization:** BLOM uses a C-grid discretization with 51 isopycnic layers referenced at 2000 dbar, and a surface mixed layer divided into two non-isopycnic layers. A tripolar grid is used instead of the bipolar grid in CMIP5 version of NorESM1, allowing for approximately a doubling of the model time step. For the CMIP6 configuration of NorESM, the BLOM grid resolution is 1   zonally and 1/4   meridionally at the equator, gradually approaching more isotropic grid cells at higher latitudes. The model bathymetry is found by averaging the S2004 (Marks and Smith, 2006) data points contained in each model grid cell with additional editing of sills and passages to their actual depths. The metric scale factors are edited to the realistic width of the Strait of Gibraltar so that strong velocity shears can be formed, enabling realistic mixing of Mediterranean water entering the Atlantic Ocean.
- **Vertical shear-induced mixing:** A second-order turbulence closure (k- model) is now used for vertical shear-induced mixing, replacing a parameterisation using the local gradient Richardson number according to Large et al. (1994).
- **Mesoscale eddy-induced transport:** The parameterisation of mesoscale eddy-induced transport is modified to more faithfully comply with the Gent and McWilliams (1990) formulation. The estimation of diffusivity for eddy-induced transport and isopycnic eddy diffusion of tracers is based on the Eden et al. (2009) implementation of Eden and Greatbatch (2008) with their diagnostic equation for the eddy length scale, but modified to give a spatially smoother and generally reduced diffusivity. The isopycnal eddy diffusivity is set equal to the thickness diffusivity.
- **Mixed layer physics:** Mixed layer physics have been improved compared to the CMIP5 version of NorESM1, in part to enable sub-diurnal coupling of the ocean. The hourly coupling now used has made it possible to add additional energy sources for upper ocean vertical mixing such as wind work on near-inertial motions and surface turbulent kinetic energy source due to wind stirring to the k- model.



- **Mixing in gravity currents:** To achieve more realistic mixing in gravity currents, the layer thickness at velocity points has been redefined and realistic channel widths are used (e.g., Strait of Gibraltar).
- **Absorption of shortwave radiation:** The penetration profile of shortwave radiation is modified, leading to a shallower absorption in NorESM2 compared to NorESM1.
- **Salinity dependent seawater freezing temperature:** With respect to coupling to the sea ice model, BLOM and CICE now use a consistent salinity dependent seawater freezing temperature (Assur, 1958). Selective damping of external inertia–gravity waves in shallow regions is enabled to mitigate an issue with unphysical oceanic variability in high latitude shelf regions, causing excessive sea ice formation due to breakup and ridging in CMIP5 versions of NorESM1.

## References

- Assur, A.: “Composition of Sea Ice and its Tensile Strength,” pp. 106–138, in Arctic Sea Ice, U.S. National Academy of Sciences-National Research Council Pub. 598, 1958.
- Bentsen, M., Bethke, I., Debernard, J. B., Iversen, T., Kirkevåg, A., Seland, Ø., Drange, H., Roelandt, C., Seierstad, I. A., Hoose, C., and Kristjánsson, J. E.: The Norwegian Earth System Model, NorESM1-M – Part 1: Description and basic evaluation of the physical climate, *Geosci. Model Dev.*, 6, 687–720, <https://doi.org/10.5194/gmd-6-687-2013>, 2013.
- Bleck, R., Smith, L. T.: A wind-driven isopycnic coordinate model of the north and equatorial Atlantic Ocean: 1. Model development and supporting experiments. *J. Geophys. Res.*, 95C, 3273–3285, <https://doi.org/10.1029/JC095iC03p03273>, 1990.
- Bleck, R., Rooth, C., Hu, D., and Smith, L. T.: Salinity-driven Thermocline Transients in a Wind- and Thermohaline-forced Isopycnic Coordinate Model of the North Atlantic. *J. Phys. Oceanogr.*, 22, 1486–1505, [https://doi.org/10.1175/1520-0485\(1992\)0222.0.CO;2](https://doi.org/10.1175/1520-0485(1992)0222.0.CO;2), 1992.
- Eden, C. and Greatbatch, R. J.: Towards a mesoscale eddy closure, *Ocean Model.*, 20, 223–239, <https://doi.org/10.1016/j.ocemod.2007.09.002>, 2008.
- Eden, C., Jochum, M., and Danabasoglu, G.: Effects of different closures for thickness diffusivity, *Ocean Model.*, 26, 47–59, <https://doi.org/10.1016/j.ocemod.2008.08.004>, 2009.
- Gent, P. R., and McWilliams, J. C.: Isopycnal Mixing in Ocean Circulation Models. *J. Phys. Oceanogr.*, 20, 150–155, [https://doi.org/10.1175/1520-0485\(1990\)020<0150:IMIOCM>2.0.CO;2](https://doi.org/10.1175/1520-0485(1990)020<0150:IMIOCM>2.0.CO;2), 1990.
- Large, W. G., McWilliams, J. C., and Doney, S. C.: Oceanic vertical mixing: a review and a model with a nonlocal boundary layer parameterization, *Rev. Geophys.*, 32, 363–403, 1994.
- Marks, K. and Smith, W.: An evaluation of publicly available global bathymetry grids, *Mar. Geophys. Res.*, 27, 19–34, <https://doi.org/10.1007/s11001-005-2095-4>, 2006.

### 1.6.4 The ocean BGC model

The ocean biogeochemistry (BGC) component **iHAMOCC** (isopycnic coordinate HAMBurg Ocean Carbon Cycle model) is an updated version of the ocean biogeochemistry module used in NorESM1. The iHAMOCC is based on the the HAMOCC5.1 model developed in Max-Planck Institute of Meteorology (Maier-Reimer et al., 2005) and was later modified for an isopycnal-coordinate ocean general circulation model (Assmann et al., 2010).

The iHAMOCC prognostically simulates five key ocean biogeochemical cycle processes:

- **1. Inorganic seawater carbon chemistry**

The carbon chemistry formulation in iHAMOCC is based on the Ocean Carbon cycle Model Intercomparison Project (OCMIP) protocols. It computes the partial pressure of CO<sub>2</sub> gas in surface layer (pCO<sub>2</sub>) based on the temperature, salinity, dissolved inorganic carbon and alkalinity concentrations. The pCO<sub>2</sub> is used to estimate

the air-sea CO<sub>2</sub> fluxes. In addition to surface pCO<sub>2</sub>, water column pH, carbonate ion, and calcite saturation state are also computed.

- **2. NPZD-type ecosystem module**

In the euphotic layer of the model (top 100 m), the lower trophic ecosystem dynamic is simulated using an NPZD (Nutrient Phytoplankton Zooplankton Detritus) ecosystem module (Six and Maier-Reimer, 1996). One phytoplankton and one zooplankton bulk compartments are simulated together with multiple limiting nutrients (nitrate, phosphate, and dissolved iron), dissolved organic carbon and particulate matters. In addition nutrient, the primary production is also limited by light availability and temperature. A fixed stoichiometry Redfield Ratio is used to govern the fluxes of nutrients and carbon among the different ecosystem compartments.

- **3. Air-sea gas exchange**

The air-sea gas exchange formulation according to Wanninkhof (2014) is applied to compute fluxes of CO<sub>2</sub>, O<sub>2</sub>, DMS, N<sub>2</sub>, N<sub>2</sub>O, CFC-11, CFC-12, and SF<sub>6</sub> gases. The fluxes of these gases are computed as a function of surface wind speed, Schmidt number, gas solubility, and partial pressure difference of the respective gases. For DMS, the flux is always from the ocean to the atmosphere.

- **4. Vertical fluxes of inorganic and organic particles**

The particulate matters produced by the biological activity in the upper ocean are transported vertically and remineralized in the water column (Schwinger et al., 2018). For particulate organic carbon, the sinking speed is increased with depth while a constant remineralization rates is used throughout the water column. For biogenic silica (opal), both constant vertical sinking speed and dissolution rate are used. For particulate inorganic carbon (CaCO<sub>3</sub>), a constant sinking speed is used while the dissolution is formulated as a function of calcite saturation state.

- **5. Sediment biogeochemistry**

A 12-layer sediment module is included in iHAMOCC (Heinze et al., 1999). It collects the sinking particle matters that are not completely dissolved or remineralized in the water column. It includes four solid sediment components (CaCO<sub>3</sub>, opal, organic carbon, and clay), and five pore water substances (dissolved inorganic carbon, alkalinity, phosphate, oxygen, and silicate). In addition to particle deposition, it simulates fluxes of tracers with the bottom-most ocean layer through pore water chemistry and diffusion.

Details on the updates and improvements of the ocean biogeochemical component of NorESM2 are provided in Tjiputra et al. (2020).

## References

- Assmann, K. M., Bentsen, M., Segschneider, J., and Heinze, C.: An isopycnic ocean carbon cycle model, *Geosci. Model Dev.*, 3, 143–167, <https://doi.org/10.5194/gmd-3-143-2010>, 2010.
- Heinze, C., Maier-Reimer, E., Winguth, A. M. E., and Archer, D.: A global oceanic sediment model for long-term climate studies, *Glob. Biogeochem. Cy.*, 13, 221–250, 1999.
- Maier-Reimer, E., Kriest, I., Segschneider, J., and Wetzol, P.: The Hamburg Ocean Carbon Cycle Model HAMOCC5.1 – Technical Description Release 1.1, *Berichte zur Erdsystemforschung*, ISSN 1614–1199, Max Planck Institute for Meteorology, Hamburg, Germany, 50 pp., 2005.
- Schwinger, J., Goris, N., Tjiputra, J. F., Kriest, I., Bentsen, M., Bethke, I., Ilicak, M., Assmann, K. M., and Heinze, C.: Evaluation of NorESM-OC (versions 1 and 1.2), the ocean carbon-cycle stand-alone configuration of the Norwegian Earth System Model (NorESM1), *Geosci. Model Dev.*, 9, 2589–2622, <https://doi.org/10.5194/gmd-9-2589-2016>, 2016.
- Six, K. D. and Maier-Reimer, E.: Effects of plankton dynamics on seasonal carbon fluxes in an ocean general circulation model, *Global Biogeochem. Cy.*, 10, 559–583, 1996.
- Tjiputra, J. F., Schwinger, J., Bentsen, M., Morée, A. L., Gao, S., Bethke, I., Heinze, C., Goris, N., Gupta, A., He, Y., Olivie, D., Seland, Ø., and Schulz, M.: Ocean biogeochemistry in the Norwegian Earth System Model version 2 (NorESM2), *Geosci. Model Dev. Discuss.*, <https://doi.org/10.5194/gmd-2019-347>, in press, 2020.
- Wanninkhof, R.: Relationship between wind speed and gas exchange over the ocean revisited, *Limnol. Oceanogr.: Methods*, 12, 351–362, <https://doi.org/10.4319/lom.2014.12.351>, 2014.

### 1.6.5 The sea-ice model

The sea ice model component is based upon version 5.1.2 of the **CICE** sea ice model of Hunke et al. (2015). A NorESM2-specific change is including the effect of wind drift of snow into ocean following Lecomte et al. (2013), as described in Bentsen et al. (in prep). The NorESM version is also equipped with the possibility to modify snow density and snow thermal conductivity in the namelist.

For more detailed documentation about the CICE model, please see: <http://www.cesm.ucar.edu/models/cesm2/sea-ice/>

### Main features of the CICE configuration in NorESM2 and differences from NorESM1

1. Same horizontal tripolar grid as the ocean model (BLOM)
2. The discretization of the sea ice thickness distribution uses 5 thickness categories.
3. EVP dynamics
4. 8 sea ice layers and 3 snow layers in the vertical
5. Mushy layer thermodynamics with prognostic internal energy and salinity within the sea ice (Turner and Hunke, 2015).
6. Delta-Eddington shortwave radiation
7. Melt ponds are allowed to form on undeformed ice (Hunke et al 2013).
8. Freezing point of sea water is salinity dependent and based on Assur (1958). The treatment is consistent between the internal mushy-layer thermodynamics, the sea ice - ocean interface, and open water sea ice growth (calculated in the ocean component).

## References

- Assur, A.. Composition of sea ice and its tensile strength. In Arctic sea ice; conference held at Easton, Maryland, February 24–27, 1958, volume 598, pages 106–138. *Publs. Natl. Res. Coun. Wash., Washington, D.C.*, 1958.
- Hunke, E. C., D. A. Hebert, and O. Lecomte. Level-ice melt ponds in the Los Alamos sea ice model, CICE. *Ocean Modelling*, 71:26–42, 2013. URL: <http://dx.doi.org/10.1016/j.ocemod.2012.11.008>.
- Hunke, E. C., et al. “CICE: The Los Alamos Sea ice Model Documentation and Software User’s Manual Version 5 (Tech. Rep. LA-CC-06–012).” Los Alamos, NM: Los Alamos National Laboratory (2015).
- Hunke, Elizabeth, Lipscomb, William, Jones, Philip, Turner, Adrian, Jeffery, Nicole, and Elliott, Scott. CICE, The Los Alamos Sea Ice Model. Computer software. <https://www.osti.gov/servlets/purl/1364126>.
- Lecomte, O., T. Fichefet, M. Vancoppenolle, F. Domine, F. Massonnet, P. Mathiot, S. Morin, and P.Y. Barriat (2013), On the formulation of snow thermal conductivity in large-scale sea ice models, *J. Adv. Model. Earth Syst.*, 5, 542–557, doi:10.1002/jame.20039
- Turner, A. K. and E. C. Hunke. Impacts of a mushy-layer thermodynamic approach in global sea-ice simulations using the CICE sea-ice model. *J. Geophys. Res. Oceans*, 120:1253–1275, 2015. URL: <http://dx.doi.org/10.1002/2014JC010358>.

### 1.6.6 The land model

The NorESM2 employs the latest version of “Community Land Model” (CLM5; Lawrence et al., 2019) as the land component. In the CLM5, Biogeophysical and biogeochemical processes are simulated for each subgrid land unit, column, and plant functional type (PFT) independently and each subgrid unit maintains its own prognostic variables. The same atmospheric forcing is used to force all subgrid units within a grid cell. The surface variables and fluxes required by the atmosphere are obtained by averaging the subgrid quantities weighted by their fractional areas. The processes simulated include:

1. Surface characterization including land type heterogeneity and ecosystem structure
2. Absorption, reflection, and transmittance of solar radiation
3. Absorption and emission of longwave radiation
4. Momentum, sensible heat (ground and canopy), and latent heat (ground evaporation, canopy evaporation, transpiration) fluxes
5. Heat transfer in soil and snow including phase change
6. Canopy hydrology (interception, throughfall, and drip)
7. Soil hydrology (surface runoff, infiltration, redistribution of water within the column, sub-surface drainage, groundwater)
8. Snow hydrology (snow accumulation and melt, compaction, water transfer between snow layers)
9. Stomatal physiology, photosynthetic capacity, and photosynthesis
10. Plant hydraulics
11. Lake temperatures and fluxes
12. Glacier processes
13. River routing and river flow
14. Urban energy balance and climate
15. Vegetation carbon and nitrogen allocation

16. Vegetation phenology
17. Plant respiration
18. Soil and litter carbon decomposition
19. Fixation and uptake of nitrogen
20. External nitrogen cycling including deposition, denitrification, leaching, and losses due to fire
21. Plant mortality
22. Fire ignition, suppression, spread, and emissions, including natural, deforestation, and agricultural fire
23. Methane production, oxidation, and emissions
24. Crop dynamics, irrigation, and fertilization
25. Land cover and land use change including wood harvest
26. Biogenic volatile organic compound emissions
27. Dust mobilization and deposition
28. Carbon isotope fractionation

The above information is available in the CLM5 technical Note. For more information, check <http://www.cesm.ucar.edu/models/cesm2/land/>

The **CLM5** in the NorESM2 comes with one minor modification (Seland et al., 2020 in review for GMD). This specific modification was made to the surface water treatment in CLM. The surface water pool is a new feature replacing the wetland land unit in earlier versions of CLM (introduced in CLM4.5). This water pool does not have a frozen state, but is added to the snow-pack when frozen. To avoid water being looped between surface water and snow during alternating cold and warm periods, we remove infiltration excess water as runoff if the temperature of the surface water pool is below freezing. This was done to mitigate a positive snow bias and an artificial snow depth increase found in some Arctic locations during melting conditions.

In the NorESM/CTSM repository, the master branch is forked from ECOMP/CTSM, which is the CLM version used in the CESM2, while the branch release-clm5.0.14-Nor (based on the CLM version 5.0.14) stores the CLM version used in the NorESM with the modification above.

## References

Lawrence, D. M., Fisher, R. A., Koven, C. D., Oleson, K. W., Swenson, S. C., Bonan, G., et al. (2019). The Community Land Model version 5: Description of new features, benchmarking, and impact of forcing uncertainty. *Journal of Advances in Modeling Earth Systems*, <https://doi.org/10.1029/2018MS001583>

### 1.6.7 The coupler, CIME

The state and flux exchanges between model components and software infrastructure for configuring, building and execution of model experiments is handled by the CESM2 coupler Common Infrastructure for Modeling the Earth (CIME; Danabasoglu et al., 2019). Among the common utility functions CIME provides is the estimation of solar zenith angle. In NorESM2, this utility function is modified with associated changes in atmosphere, land and sea ice components, ensuring that all albedo calculations use zenith angle averaged over the components time-step instead of instantaneous angles.

## 1.6.8 References

- Bentsen, M., Bethke, I., Debernard, J. B., Iversen, T., Kirkevåg, A., Seland, Ø., Drange, H., Roelandt, C., Seierstad, I. A., Hoose, C., and Kristjánsson, J. E.: The Norwegian Earth System Model, NorESM1-M – Part 1: Description and basic evaluation of the physical climate, *Geoscientific Model Development*, 6, 687–720, <https://www.geosci-model-dev.net/6/687/2013/>, 2013.
- Danabasoglu, G., Lamarque, J.-F., Bacmeister, J., Bailey, D. A., DuVivier, A. K., Edwards, J., Emmons, L. K., Fasullo, J., Garcia, R., Gettelman, A., Hannay, C., Holland, M. M., Large, W. G., Lawrence, D. M., Lenaerts, J. T. M., Lindsay, K., Lipscomb, W. H., Mills, M. J., Neale, R., Oleson, K. W., Otto-Bliesner, B., Phillips, A. S., Sacks, W., Tilmes, S., van Kampenhout, L., Vertenstein, M., Bertini, A., Dennis, J., Deser, C., Fischer, C., Fox-Kemper, B., Kay, J. E., Kinnison, D., Kushner, P. J., Long, M. C., Mickelson, S., Moore, J. K., Nienhouse, E., Polvani, L., Rasch, P. J., and Strand, W. G.: The Community Earth System Model version 2 (CESM2), Submitted to *J. Adv. Model. Earth Syst.*, 2019.
- Graff, L. S., Iversen, T., Bethke, I., Debernard, J. B., Seland, Ø., Bentsen, M., Kirkevåg, A., Li, C., and Olivié, D. J. L.: Arctic amplification under global warming of 1.5 and 2 °C in NorESM1-Happi, *Earth System Dynamics*, 10, 569–598, <https://www.earth-syst-dynam.net/10/569/2019/>, 2019.
- Hunke, E. C., Lipscomb, W. H., Turner, A. K., Jeffery, N., and Elliott, S.: CICE: the Los Alamos Sea Ice Model Documentation and Software User’s Manual Version 5.1, Tech. Rep. LA-CC-06-012, Los Alamos National Laboratory, Los Alamos, New Mexico, USA, <https://github.com/CICE-Consortium/CICE-svn-trunk/releases/tag/cice-5.1.2>, 2015.
- Iversen, T., Bentsen, M., Bethke, I., Debernard, J. B., Kirkevåg, A., Seland, Ø., Drange, H., Kristjánsson, J. E., Medhaug, I., Sand, M., and Seierstad, I. A.: The Norwegian Earth System Model, NorESM1-M – Part 2: Climate response and scenario projections, *Geoscientific Model Development*, 6, 389–415, <https://doi.org/10.5194/gmd-6-389-2013>, 2013.
- Kirkevåg, A., Iversen, T., Seland, Ø., Hoose, C., Kristjánsson, J. E., Struthers, H., Ekman, A. M. L., Ghan, S., Griesfeller, J., Nilsson, E. D., and Schulz, M.: Aerosol–climate interactions in the Norwegian Earth System Model – NorESM1-M, *Geoscientific Model Development*, 6, 207–244, <https://doi.org/10.5194/gmd-6-207-2013>, 2013.
- Kirkevåg, A., Grini, A., Olivié, D., Seland, Ø., Alterskjær, K., Hummel, M., Karset, I. H. H., Lewinschal, A., Liu, X., Makkonen, R., Bethke, I., Griesfeller, J., Schulz, M., and Iversen, T.: A production-tagged aerosol module for Earth system models, OsloAero5.3 – extensions and updates for CAM5.3-Oslo, *Geoscientific Model Development*, 11, 3945–3982, <https://doi.org/10.5194/gmd-11-3945-2018>, 2018.
- Lawrence, D. M., Fisher, R. A., Koven, C. D., Oleson, K. W., Swenson, S. C., Bonan, G., Collier, N., Ghimire, B., van Kampenhout, L., Kennedy, D., Kluzek, E., Lawrence, P. J., Li, F., Li, H., Lombardozzi, D., Riley, W. J., Sacks, W. J., Shi, M., Vertenstein, M., Wieder, W. R., Xu, C., Ali, A. A., Badger, A. M., Bisht, G., van den Broeke, M., Brunke, M. A., Burns, S. P., Buzan, J., Clark, M., Craig, A., Dahlin, K., Drewniak, B., Fisher, J. B., Flanner, M., Fox, A. M., Gentine, P., Hoffman, F., Keppel-Aleks, G., Knox, R., Kumar, S., Lenaerts, J., Leung, L. R., Lipscomb, W. H., Lu, Y., Pandey, A., Pelletier, J. D., Perket, J., Randerson, J. T., Ricciuto, D. M., Sanderson, B. M., Slater, A., Subin, Z. M., Tang, J., Thomas, R. Q., Val Martin, M., and Zeng, X.: The Community Land Model Version 5: Description of New Features, Benchmarking, and Impact of Forcing Uncertainty, *Journal of Advances in Modeling Earth Systems*, n/a, <https://doi.org/10.1029/2018MS001583>, 2019.
- Lecomte, O., Fichefet, T., Vancoppenolle, M., Domine, F., Massonnet, F., Mathiot, P., Morin, S., and Barriat, P. Y.: On the formulation of snow thermal conductivity in large-scale sea ice models, *Journal of Advances in Modeling Earth Systems*, 5, 542–557, <https://doi.org/10.1002/jame.20039>, 2013.
- Schwinger, J., Goris, N., Tjiputra, J. F., Kriest, I., Bentsen, M., Bethke, I., Ilicak, M., Assmann, K. M., and Heinze, C.: Evaluation of NorESM-OC (versions 1 and 1.2), the ocean carbon-cycle stand-alone configuration of the Norwegian Earth System Model (NorESM1), *Geoscientific Model Development*, 9, 2589–2622, <https://doi.org/10.5194/gmd-9-2589-2016>, 2016.
- Seland, Ø., Bentsen, M., Seland Graff, L., Olivié, D., Toniazzo, T., Gjermundsen, A., Debernard, J. B., Gupta, A. K., He, Y., Kirkevåg, A., Schwinger, J., Tjiputra, J., Schancke Aas, K., Bethke, I., Fan, Y., Griesfeller, J.,

Grini, A., Guo, C., Ilicak, M., Hafsaht Karset, I. H., Landgren, O., Liakka, J., Onsum Moseid, K., Nummelin, A., Spensberger, C., Tang, H., Zhang, Z., Heinze, C., Iverson, T., and Schulz, M.: The Norwegian Earth System Model, NorESM2 – Evaluation of the CMIP6 DECK and historical simulations, *Geosci. Model Dev. Discuss.*, <https://www.geosci-model-dev-discuss.net/gmd-2019-378/>, 2020.

Tjiputra, J. F., Schwinger, J., Bentsen, M., Morée, A. L., Gao, S., Bethke, I., Heinze, C., Goris, N., Gupta, A., He, Y., Olivié, D., Seland, Ø., and Schulz, M.: Ocean biogeochemistry in the Norwegian Earth System Model version 2 (NorESM2), *Geosci. Model Dev. Discuss.*, <https://doi.org/10.5194/gmd-2019-347>, in review, 2020.

Toniazzo, T., Bentsen, M., Craig, C., Eaton, B., Edwards, J., Goldhaber, J., Jablonowski, C., and Lauritzen, P. J.: Enforcing conservation of axial angular momentum in the atmospheric general circulation model CAM6, *Geosci. Model Dev. Discussions*, <https://doi.org/10.5194/gmd-2019-254>, 2019.

## 1.7 NorESM2 key simulations

For a detailed overview of the NorESM2 key simulations e.g. the spin up and CMIP6 DECK simulations, also see: <https://noresmhub.github.io/noresm-exp/intro.html>

## 1.8 Access NorESM data: CMIP and other

### 1.8.1 CMIP6 archive of NorESM results

NorESM2 contributes to the 6th phase of the Coupled Model Intercomparison Project (CMIP6; Eyring et al., 2016): <https://www.wcrp-climate.org/wgcm-cmip/wgcm-cmip6>

#### CMIP6:

- Project under World Climate Research Programme (WCRP)
- Since 1995 CMIP has coordinated climate model experiments
- Defines common experiment protocols, forcings and output.
- 33 model groups participate

#### Advantages:

- Several models perform the same experiments (enabling comparisons across models and multi-model ensembles)
- Homogenized and standardized outputs (including variable names). More specifically, all data is in compliance with the standards of the Climate Model Output Rewriter (CMOR).



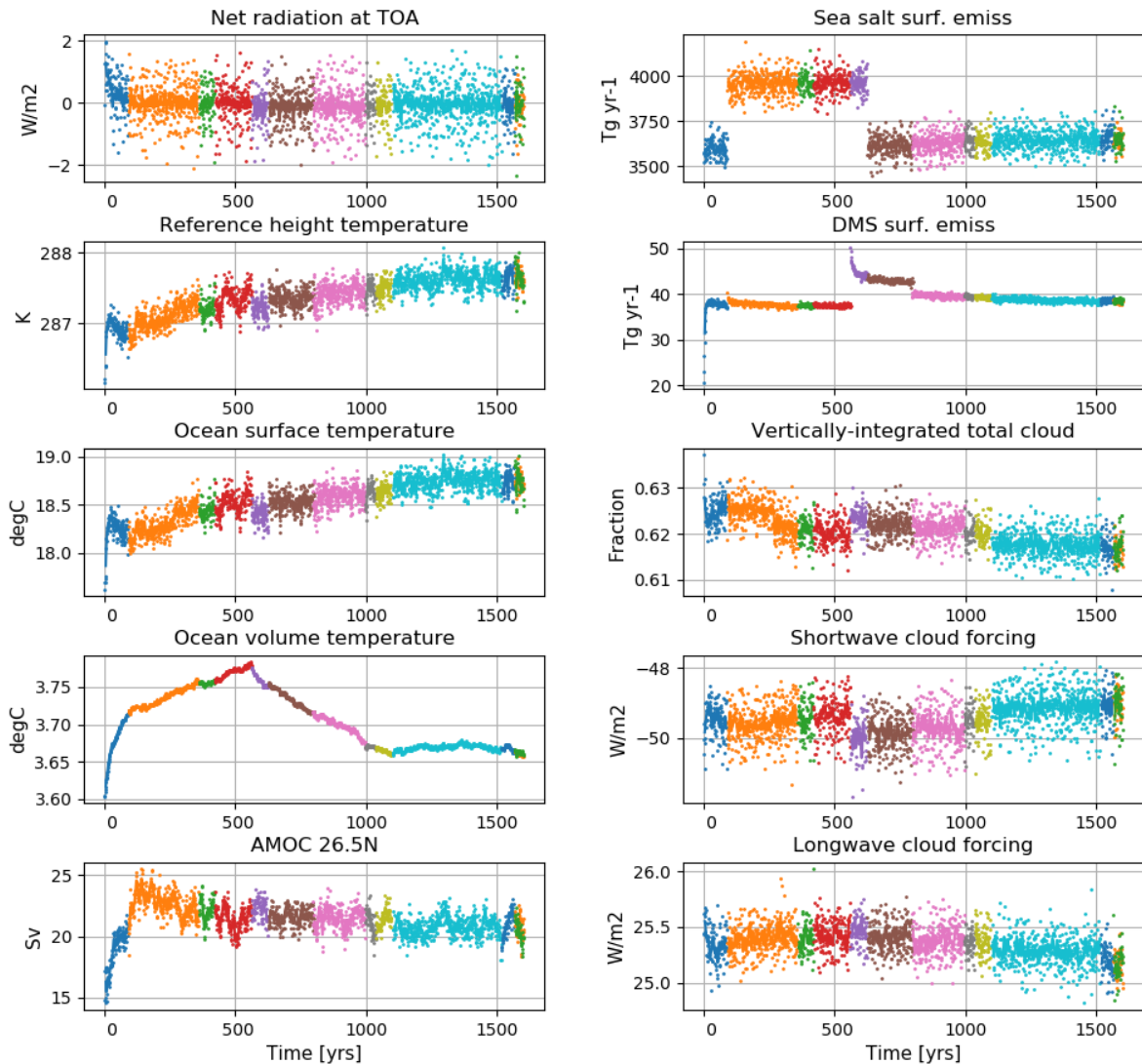


Fig. 1: **Left column (from top to bottom):** Globally and annually averaged Net radiation @ top of model, Surface (2m) air temperature, Sea surface temperature (SST), global and volume averaged ocean temperature, Atlantic meridional overturning circulation (AMOC) @ 26.5N. **Right column (from top to bottom):** Globally and annually sum of Sea salt surface emissions, DMS (dimethylsulfide) surface emissions, globally and annually averaged vertically-integrated total cloud cover, shortwave cloud forcing and longwave cloud forcing.



## Data access

### The Earth System Grid Federation (ESGF):

All data is CMOR-ized and public available here: <https://esg-dn1.nsc.liu.se/search/cmip6-liu/>

### The (Norwegian) National Infrastructure for Research Data (NIRD)

All NIRD users can access the archive of CMOR-ized CMIP6 NorESM data under project NS9034K:

```
/projects/NS9034K/CMIP6/
```

This is the same data that you can access through the ESGF. Note data is organized by the different contributions/MIPs. The DECK and historical simulations are located under “CMIP”.

**NIRD users that are members of NS9252K (NFR storage project for KeyCLIM) can access CMOR-ized CMIP6 data from multiple models under:**

```
/projects/NS9252K/CMIP6/
```

Note this data has a different structure than the data residing in /projects/NS9034K/CMIP6/. If you want to make a request for data or for more data to be downloaded to this folder, contact [jang@met.no](mailto:jang@met.no). If you are a NIRD user, but not a member of this project and would like to request access, contact [michaels@met.no](mailto:michaels@met.no).

**NIRD users that are members of NS9560K (NFR storage project for INES) can access most of the raw (non-CMOR-ized) model data under:**

```
/projects/NS9560K/noresm/cases/
```

If you are a NIRD user, but not a member of this project, and would like to request access, contact [mber@norceresearch.no](mailto:mber@norceresearch.no).

## DECK contributions

NorESM2 contributions to the CMIP6 Diagnostic, Evaluation and Characterization of Klima (DECK) and CMIP6 historical simulations

Exp. short name	Exp. long name	NorESM2 version (members)
amip	a historical Atmospheric MIP simulation	NorESM2-LM (1), NorESM2-MM (1)
piControl	pre-industrial control simulation	NorESM2-LM (1), NorESM2-MM (1)
historical	historical	NorESM2-LM (3), NorESM2-MM (1)
abrupt-4xCO2	forced by an abrupt quadrupling of atmospheric CO2	NorESM2-LM (1), NorESM2-MM (1)
1pctCO2	forced by a 1 % pr. year atmospheric CO2 increase	NorESM2-LM (1), NorESM2-MM (1)

## MIPs contributions

MIP short name	MIP long name	NorESM2 version	Contact person(s)*
CMIP	Coupled MIP	NorESM2-LM, NorESM2-MM	ØS, TT, DJLO
AerChemMIP	Aerosols and Chemistry MIP	NorESM2-LM	DJLO, MS
C4MIP	Coupled Climate Carbon Cycle MIP	NorESM2-LME	JS, JT
CDRMIP	The Carbon Dioxide Removal MIP	NorESM2-LM	JT, JS
CFMIP	Cloud Feedback MIP	No contribution yet	TS, ØS
DAMIP	Detection and Attribution MIP	NorESM2-LM	ØS
DCPP	Decadal Climate Prediction Project	NorCPM1	NK, FC
GeoMIP	Geoengineering MIP	No contribution yet	HM
LUMIP	Land-Use MIP	No contribution yet	HL, YH
OMIP	Ocean MIP	NorESM2-LM	MB, JS
PAMIP	Polar Amplification MIP	NorESM2-LM	LSG, MB
PMIP	Paleoclimate MIP	NorESM1-F, NorESM2-LM	CG, ZZ
RFMIP	Radiative Forcing MIP	NorESM2-LM, NorESM2-MM	AK, DJLO
ScenarioMIP	Scenario MIP	NorESM2-LM, NorESM2-MM	MB, MS
SIMIP	Sea Ice MIP	No contribution yet	JBD
ZECMIP	Zero Emissions Commitment MIP	NorESM2-LM	JS, JT

\* **MB:** Mats Bentsen <mats.bentsen@uni.no>, **FC:** Francois Counillon <francois.counillon@nersc.no>, **JBD:** Jens Bolding Debernard <jensd@met.no>, **LSG:** Lise Seland Graff <lisesg@met.no>, **YH:** Yanchun He <yanchun.he@nersc.no>, **AK:** Alf Kirkevåg <alfk@met.no>, **NK:** Noel Keenlyside <noel.keenlyside@gf.uib.no>, **HL:** Hanna Lee <hanna.lee@uni.no>, **HM:** Helene Muri <helene.muri@ntnu.no>, **DJLO:** Dirk Jan Leo Olivie <dirkjlo@met.no>, **TT:** Thomas Toniazzo <thomas.toniazzo@uni.no>, **JS:** Jörg Schwinger <jorg.schwinger@norceresearch.no>, **MS:** Michael Schultz <michaels@met.no> **TS:** Trude Storelvmo <trude.storelvmo@geo.uio.no>, **ØS:** Øyvind Seland <oyvind.seland@met.no>, **JT:** Jerry Tjiputra <Jerry.Tjiputra@norceresearch.no>, **CG:** Chuncheng Guo <chgu@norceresearch.no>, **ZZ:** Zhongshi Zhang <zhzh@norceresearch.no>.

Overview CMIP6-Endorsed MIPs:

<https://www.wcrp-climate.org/modelling-wgcm-mip-catalogue/modelling-wgcm-cmip6-endorsed-mips>

## References

Seland, Ø., Bentsen, M., Seland Graff, L., Olivie, D., Toniazzo, T., Gjermundsen, A., Debernard, J. B., Gupta, A. K., He, Y., Kirkevåg, A., Schwinger, J., Tjiputra, J., Schancke Aas, K., Bethke, I., Fan, Y., Griesfeller, J., Grini, A., Guo, C., Ilicak, M., Hafsahl Karset, I. H., Landgren, O., Liakka, J., Onsum Moseid, K., Nummelin, A., Spensberger, C., Tang, H., Zhang, Z., Heinze, C., Iverson, T., and Schulz, M.: The Norwegian Earth System Model, NorESM2 – Evaluation of the CMIP6 DECK and historical simulations, *Geosci. Model Dev. Discuss.*, <https://doi.org/10.5194/gmd-2019-378>, in review, 2020.

Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., and Taylor, K. E.: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization, *Geosci. Model Dev.*, 9, 1937–1958, <https://doi.org/10.5194/gmd-9-1937-2016>, 2016.

## 1.8.2 CMIP5 archive of NorESM results

NorESM1-M contributes to the 5th phase of the Coupled Model Intercomparison Project (CMIP5):

<https://www.wcrp-climate.org/wgcm-cmip/wgcm-cmip5>

### CMIP5:

- Project under World Climate Research Programme (WCRP)
- Since 1995 CMIP has coordinated climate model experiments
- Defines common experiment protocols, forcings and output.

### Advantages:

- Several models perform the same experiments (enabling comparisons across models and multi-model ensembles)
- Homogenized and standardized outputs (including variable names). More specifically, all data is in compliance with the standards of the Climate Model Output Rewriter (CMOR).

### Data access

#### The Earth System Grid Federation (ESGF):

**All data is CMOR-ized and available from:**

<https://esgf-node.llnl.gov/search/cmip5/>

#### The (Norwegian) National Infrastructure for Research Data (NIRD)

**All NIRD users** can access the archive of CMOR-ized CMIP5 NorESM data under project NS9034K:

```
/projects/NS9034K/CMIP5/
```

### References

Bentsen, M., Bethke, I., Debernard, J. B., Iversen, T., Kirkevåg, A., Seland, Ø., Drange, H., Roelandt, C., Seierstad, I. A., Hoose, C., and Kristjánsson, J. E.: The Norwegian Earth System Model, NorESM1-M – Part 1: Description and basic evaluation of the physical climate, *Geosci. Model Dev.*, 6, 687–720, <https://doi.org/10.5194/gmd-6-687-2013>, 2013.

Iversen, T., Bentsen, M., Bethke, I., Debernard, J. B., Kirkevåg, A., Seland, Ø., Drange, H., Kristjánsson, J. E., Medhaug, I., Sand, M., and Seierstad, I. A.: The Norwegian Earth System Model, NorESM1-M – Part 2: Climate response and scenario projections, *Geosci. Model Dev.*, 6, 389–415, <https://doi.org/10.5194/gmd-6-389-2013>, 2013.

Kirkevåg, A., Iversen, T., Seland, Ø., Hoose, C., Kristjánsson, J. E., Struthers, H., Ekman, A. M. L., Ghan, S., Griesfeller, J., Nilsson, E. D., and Schulz, M.: Aerosol–climate interactions in the Norwegian Earth System Model – NorESM1-M, *Geosci. Model Dev.*, 6, 207–244, <https://doi.org/10.5194/gmd-6-207-2013>, 2013.

Tjiputra, J. F., C. Roelandt, M. Bentsen, D. M. Lawrence, T. Lorentzen, J. Schwinger, Ø. Seland, and C. Heinze *Geosci. Model Dev.*: Evaluation of the carbon cycle components in the Norwegian Earth System Model (NorESM), 6, 301–325, <https://doi.org/10.5194/gmd-6-301-2013>, 2013.

### 1.8.3 HAPPI and HappiEVA data

#### Experiments for 1.5 and 2.0 degree warming

##### AMIP-style HAPPI experiments

A set of AMIP-style large ensemble experiments (125 members) were carried out with NorESM1-Happi (Graff et al., 2019) for three climate states:

- the present-day climate (the All-Hist experiment),
- a climate that is 1.5 degrees warmer than pre-industrial conditions (the Plus15-Future experiment), and
- a climate that is 2.0 degrees warmer than the pre-industrial (the Plus20-Future experiments)

as a part of the HAPPI initiative (Half A degree additional warming, Prognosis and Projected Impacts; <http://happimip.org>; Mitchell et al., 2017).

**CMOR-ized data for NorESM1-Happi and the other participating models can be retrieved from** <http://portal.nersc.gov/c20c/data.html>

**Raw model output for NorESM1-Happi can be retrived from the NIRD research data archive:**

- <https://doi.org/10.11582/2019.00006> (All-Hist)
- <https://doi.org/10.11582/2019.00002> (Plus15-Future)
- <https://doi.org/10.11582/2019.00003> (Plus20-Future)

##### Fully-coupled experiments

###### Fully-coupled experiments targeting

- 1.5 degree warming (relative to pre-industrial conditions) and
- 2.0 degree warming

were conducted with NorESM1-Happi. See Graff et al. (2019) for details.

**The data (raw model output) can be retrieved from the NIRD research data archive:** <https://doi.org/10.11582/2020.00021>

##### Slab-ocean experiments

###### Slab-ocean experiments were conducted for

- a present-day climate that is as similar to that in the NorESM1-Happi All-Hist experiment as possible
- a 1.5 degree warmer climate (relative to the pre-industrial) that is as similar to that in the NorESM1-Happi Plus15-Future experiment as possible
- a 2.0 degree warmer climate (relative to the pre-industrial) that is as similar to that in the NorESM1-Happi Plus20-Future experimet as possible

**The data (raw model output) can be retrieved from the NIRD research data archive:** <https://doi.org/10.11582/2020.00013>

## CMIP5 DECK and scenario experiments with NorESM1-Happi

A set of fully-coupled DECK experiments and scenario experiments were carried out with NorESM1-Happi (Graff et al., 2019) to enable an extensive validation of the model in line with the CMIP5 protocol (note that these experiments are not a part of the official CMIP5 data set):

- a pre-industrial control simulation (the **piControl** experiment)
- three historical members for the time period 1850-2005 (the **Hist1**, **Hist2**, and **Hist3** experiments)
- Hist1 with external forcings as in 1850, but with with varying (historical)
  - greenhouse gases
  - aerosols and aerosol precursor emissions
  - natural solar radiation and volcanos
- RCP scenario experiments in which the radiative forcing at the end of the 21st century corresponds to
  - 2.6 W/m<sup>2</sup>
  - 4.5 W/m<sup>2</sup>
  - 8.5 W/m<sup>2</sup>
- experiments in forcings are as in the pre-industrial climate except the CO<sub>2</sub> concentrations which are
- instantly quadrupled at the beginning of the experiment (the **abrupt-4xCO<sub>2</sub>** experiment)
  - increased by 1% per year until quadrupling and then held constant (the **gradual-4xCO<sub>2</sub>** experiment)

**The data (raw model output) can be retrieved from the NIRD research data archive:** <https://doi.org/10.11582/2020.00021>

**In addition, a set of slab-ocean experiments were conducted to assess the climate sensitivity of NorESM1-Happi under the slab-**

- a pre-industrial (1850) control simulation
- a 2xCO<sub>2</sub> experiment
- a 4xCO<sub>2</sub> experiment

**Data from the slab-ocean experiments can be retrieved from the NIRD research data archive:** <https://doi.org/10.11582/2020.00014>

## References

Graff, L. S., Iversen, T., Bethke, I., Debernard, J. B., Seland, Ø., Bentsen, M., Kirkevåg, A., Li, C., and Olivieri, D. J. L.: Arctic amplification under global warming of 1.5 and 2 °C in NorESM1-Happi, *Earth System Dynamics*, 10, 569–598, <https://doi.org/10.5194/esd-10-569-2019>, 2019.

Mitchell, D., AchutaRao, K., Allen, M., Bethke, I., Forster, P., Fuglestedt, J., Gillett, N., Haustein, K., Iverson, T., Massey, N., Schleussner, C.-F., Scinocca, J., Seland, Ø., Shioyama, H., Shuckburgh, E., Sparrow, S., Stone, D., Wallom, D., Wehner, M., and Zaaboul, R.: Half a degree Additional warming, Projections, Prognosis and Impacts (HAPPI): Background and Experimental Design, *Geosci. Model Dev.*, 10, 571–583, <https://doi.org/10.5194/gmd-10-571-2017>, 2017.

## 1.9 Contribute to NorESM2

Recipes for GIT usage

### 1.9.1 Git Recipes

Before doing things locally

```
git status
git pull
```

Merging a file from another branch into current branch

```
clone, git pull, go to directory where to put file
git checkout -b newbranch
git checkout master file
git commit -m "add file"
git pull
git push
```

Check the branch you are on

```
git branch -a
```

### 1.9.2 Version control - GitHub

**Obtain a copy of the model (using git)**

- **Create a github user:** You can create the github user yourself. Go to <https://github.com/join> and create a user (Make user-name which is easy to understand, for example FirstnameLastname. You can attach several email-addresses to the same user.)
- Visit this page: <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>
- Send email to [oyvind.seland@met.no](mailto:oyvind.seland@met.no) to get the right permissions for the new github user (The email must contain who you are and the github username).
- When you have the right permissions, you can obtain the code.

```
git clone https://githubUserName@github.com/metno/noresm.git
```

The last point will create a new directory called “noresm” in the place you checked out the model. Go to that directory before executing any git-commands.

If you get error messages, verify that you can open the page <https://github.com/metno/noresm> in a web-browser. If you can not, you are probably not a github-user or not member of the noresm group on github.

- Also do the following on all machines where you use git:
- **Make sure you have a version of git >= 2.0** (add the line “module load git” to your .bashrc files on hexagon, vilje)
- **git config - -global push.default simple** (Will edit your ~/.gitconfig file to a safer way to share your modifications, see <http://stackoverflow.com/questions/13148066/warning-push-default-is-unset-its-implicit-value-is-changing-in-git-2-0>)

Note that with git, the main branch is no longer called “trunk”, it is called “master”!

## Verify that you have the correct checkout

When you have cloned the model, check that you have gotten what you wanted!

Check that your favourite branch is available using the command

```
git branch --all
```

(You should see the branch “master” on top with a star next to it. This is the branch you get by default. The other branches are listed below with remotes/origin/branchName, but you can not work on them until you check them out, see below)

To check out (locally) your favourite branch and to start working on it, write

```
git checkout -b myBranchName origin/myBranchName
```

(Note that myBranchName must be one of the branches listed by the above command)

If you don’t use the “-b” option, you will get something which is not correct. Make sure you are tracking a remote branch. You can write

```
git branch -vv
```

to see which remote branch you are tracking. The output will be something like:

```
\* myCheckedOutBranchName 1a08184 [origin/myCheckedOutBranchname]_
↪LatestCommitMessageOnBranch
```

Note that once a branch has been checked out using the -b option, you can switch between any of your checked out branches using the command

```
git checkout aCheckedOutBranchName
```

Note that in git, switching to a new branch change the files in your working directory. Git will warn you if you have any modified files before switching to a new branch. This is different from how svn works.

## Modify files

Modify the code (for example a file named myChangedFile.F90) and send back to your local repository through

```
git add myChangedFile.F90
git commit -m "aMessage"
```

The message should link to the issue on github, so if you fix issue number 100 by this code change, you would probably write something like

```
git commit -am "Did part of the work to resolve metno/noresm#100"
```

Verify, using the tool “gitk” that the changes make sense.

### Get modifications from github

Get modification from remote source by

```
git pull
```

To be absolutely sure about branch names etc, you can do

```
git pull remoteName remoteBranchName:myLocalBranchName
```

which if your are picking up changes the master-branch would translate to

```
git pull origin master:master
```

### Send modifications to github

This command assumes that your changes go to the remote branch named like your branch (which is most of the times the case)

```
git push
```

You can also do (to be completely sure):

```
git push remoteName myLocalBranchName:remoteBranchName
```

which if your are changing the master-branch would translate to

```
git push origin master:master
```

(The above command means push my changes to the remote named “origin” from my local branch named master to the remote branch named master. If you are changing another branch than master, you must obviously not write “master”.)

### Git workflows - centralized vs. fork-and-branch workflow

Before you start to clone the repository to your local machine, decide which workflow which is best suited for your work. See <https://www.atlassian.com/git/tutorials/comparing-workflows>.

When working with documentation and text which is not critical i.e. breaking any software or build, a simple workflow like the **Centralized Workflow** should work well. However, when collaborating on a software development project, it is recommended to use the **Forking Workflow** <https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow>. Note that this includes the **Feature Branch Workflow** <https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow>.

There are many advantages with forking workflow, e.g. you cannot mess up the official repository, only your own, but the greatest benefit is that instead of pushing directly to the official repository, you instead create a **pull request (PR)** a.k.a. **merge request** to the upstream repository. This allows commits/branches to be reviewed by others and create a discussion thread before the PR is merged into the upstream repository.

Basically, the “fork and branch” workflow looks something like this:

- Fork a GitHub repository.
- Clone the forked repository to your local system.
- Add a Git remote for the original repository.



- Create a feature branch in which to place your changes.
- Make your changes to the new branch.
- Commit the changes to the branch.
- Push the branch to GitHub.
- Open a pull request from the new branch to the original repo.
- Clean up after your pull request is merged.

To start off:

- Press the Fork button in the project, and clone the forked project.
- Add the remote upstream repository that you pull from, in order to keep your forked updated with the main development, e.g.

```
git remote add upstream https://github.com/NorESMhub/NorESM.git
git remote -v                # check that you are tracking the right repositories
↔(origin and upstream)
git pull upstream master    # pull latest from the upstream master branch; do it
↔often if possible
git push origin master      # do this when upstream is ahead of you local (origin)
↔repos, to stay in sync.
git checkout -b my-feature  # create and switch to a new branch "my-feature".
...                          # edit some code
git commit -a -m "Add first draft of my feature"
git push
```

You are now ready to make a pull request of my-feature branch. This can be done from github after you pushed. Remember that after the PR is created and reviewed by others, you may need to go back and fix things before it is accepted and can be merged. After merging the PR, you should normally delete the feature branch and update your local repos. to keep things clean. It will still be seen as a (merged) branch in the upstream repository.

**Note:** If your feature branch has many commits, it may be smart to “squash” the history before creating the PR, so that it is easier to review the full changes by others (and yourself). This can be done using the **git rebase** command, but this will not be covered here.

### Development branch vs. continuous integration tool (CI)

When working using the forking workflow and committing code through reviewed pull requests, there will still be times when code changes will break the software build for various reasons. It is therefore common to merge PR’s into a **development branch** in the upstream repository, rather than directly to **master**. This adds additional management, because administrator must merge the development branch into master frequently and regularly, unless the build is broken. The gain is that **master** *always should work*.

An alternative to this scheme is to configure the workflow using a **CI/CD tool** that automates this process. I.e. when the pull request is created, the branch will automatically checked out on a dedicated build server and built. The pull request will not be published before the build is successful on the build server. On github, this is possible with **GitHub Actions** <https://help.github.com/en/actions>. It requires effort to get this in place for complex projects, but is normally worth it for large projects.

Another huge benefit of using a CI-tool is that it can automatically run test-suites in your project. E.g. a limited test-suite after successful build (part of evaluating that the build was OK), and a larger set test-suite after nightly builds.

### Hotfix branches

A **hotfix** branch is created to fix a specific problem or bug. It should normally branch off and merge back to **master**, but may also merge to **development** or **release** branches. The procedures for hotfix branches are the same as feature branches in terms of creation and merging through pull requests. The main difference is if a single bug fix should be introduced in multiple branches.

To introduce a fix in multiple branches, the **hotfix** branch should be initiated at a common ancestor for all the branches, usually the last commit common to all branch histories. This preserves the development history for the fix and avoids the potential problem of propagating code between branches unintentionally. Fortunately, git can help to identify this point using the command **git merge-base**. In the most general case, introducing a fix in multiple branches, one would check out a new hotfix branch

```
git checkout -b hotfix/x.x.x-yy $(git merge-base --octopus branch1 branch2 ...  
↪branchN)
```

The naming convention for the hotfix branch is “hotfix/<latest-NorESM-version>-<hotfix-number>”. The “-octopus” flag is used if the merge-base involves more than two branches. In practical terms one would normally just include the hotfix for **master** and the latest release, e.g. the first **hotfix** branch for noresm2.0.2 would be

```
git checkout -b hotfix/2.0.2-1 $(git merge-base master noresm2)
```

After introducing the fix in the code, the hotfix branch should be merged to all relevant branches through normal pull requests.

### Tips and Gotcha’s when working with Git

Git is a very complex system, and combining it with a complex workflow, it can be overwhelming. Here are some tips to make things easier:

- **Limit number of simultaneous work branches.** The system can technically handle huge number of branches, but mentally it is very difficult to remember what exactly the different branches contain, especially if they are not sync with the master branch. Try not to have more than two feature branches alive at any time.
- **Make branches short-lived.** Unless you are making huge refactoring changes in the code (which should have been accepted by the team beforehand), you should generally always create feature-branches that are small enough to be finished within a day or two. When you are not able to finish the feature this rapid, create a **work-in-progress (WIP) pull request** so that the team is informed about what you work on and its progress.
- **Don’t underestimate the value of publishing your commits.** Public commits to git is very often the most valuable communication asset to the rest of the team (in some periodes, the only way you communicate). To view what others are doing is key to make your own commits consistent and in sync with others and the whole project. This is another important reason why you should avoid working privately on your own branches for prolonged periods. As mentioned above, also unfinished features are worthy a WIP pull request.

### If you don’t understand and want to get back to svn

<http://www.git-tower.com/blog/git-for-subversion-users-cheat-sheet/>

## 1.10 Frequently asked questions

### 1.10.1 Technical FAQ

### 1.10.2 Aerosol FAQ

All NorESM users and developers are here invited to submit questions and/or answers about aerosol related topics not covered elsewhere in NorESM-doc.

#### How can I get mass and number concentration diagnostics via AeroTab and postprocessing?

Example a) retrieve the aerocom mass and number smaller than a given size

Example b) for FORCES/CRESCENDO, using the SALSA definition of size range for aitken, accumulation, and coarse mode: Nucleation: 3-7.7 nm; Aitken: 7.7-50 nm, Accumulation: 50-700 nm, Coarse: 700-10 000 nm.

Availability of such output mainly follows the AeroCom requirements, which for specific size intervals only concern mass concentrations. For the most accurate calculations of size-specific number concentrations and mass concentrations which are not currently included in AeroTab and AeroOslo (NorESM2), this can with some effort be added by extending the code to do so. There are, however, alternative ways to approximate both mass and number concentrations through post-processing. To see what output is generally available from the look-up tables made by AeroTab and directly outputted in NorESM2 when configured with #define AEROCOM, see the AeroTab User's Guide at NorESMhub: [https://github.com/NorESMhub/NorESM/blob/norasm2/doc/configurations/AeroTab-user-guide\\_v16april2020.pdf](https://github.com/NorESMhub/NorESM/blob/norasm2/doc/configurations/AeroTab-user-guide_v16april2020.pdf)

Specifically, this particular configuration of NorESM2 produces output which, by use of the post-processing script createaerocom-NorESM2.sh, can be reformatted to conform with the AeroCom standard:

PM2.5 concentrations, accurate method (included in the look-up tables)

```
"sconcpm25&PM25&ug m-3&S"
```

and PM10 concentrations, approximate method (assuming no growth of large sizes)

```
"sconcpm10[time,lat,lon]&PMTOT(:, :, :)-PS(:, :, :)/287.0f/TS(:, :, :)*1.e9f*($F10DSTA3)
↪*DST_A3(:, $LL, :, :)+$F10SSA3)*SS_A3(:, $LL, :, :))&ug m-3&S"
```

where the mass fraction of DST\_A3 for  $d > 10 \mu\text{m}$  (from AeroTab, assuming no growth)

F10DSTA3="0.23f"

anf the mass fraction of SS\_A3 for  $d > 10 \mu\text{m}$  (from AeroTab, assuming no growth)

F10SSA3="0.008f"

PM1 is also produced by NorESM2, and can be added to the create-aerocom script. Mass concentrations for other intervals are not standard output, although AOD is taken out for diameters  $< 1 \mu\text{m}$  and for  $> 1 \mu\text{m}$ .

Concerning number and mass concentrations for various size intervals: Modified, aged (but dry) size distributions in AeroTab are fitted to lognormal size-distributions for use in cloud droplet activation, giving modal median radii  $R$  and standards deviations  $SIGMA$ , in addition to the modal number concentrations. From this, approximate mass and number concentrations for any (dry) size range can be calculated by integration over the given size range. A good example/template for this is the fortran code made for use in the PNSD (particle number size distribution) experiment for AeroCom, with the main program file pnsd.f (see link below). This has been used only with NorESM1.2/CAM5.3-Oslo data (the experiment was never followed through and published), and may need to be modified.

### Where is the example script which produces with AeroTab the tables necessary to postprocess diagnostics for eg mass and number below a certain diameter?

For general AeroCom use: `/components/cam/tools/aerocom/createaerocom-NorESM2.sh`

For the PNSD experiment (number concentrations): `/components/cam/tools/aerocom3-PNSD_scripts-and-code/pnsd/`

### Where is the documentation for doing an EasyAerosol simulation? Compset, input, etc.

This will be described once the actual SpAer code has been tested and committed to the new model version (NorESM2.0.1), where each of the components exist on separate repositories.

### What is needed to add a tracer to the NorESM2 model?

Øyvind, Dirk, or someone else who has experience with this?...

### What is the hygroscopicity for different species in NorESM2, expressed as kappa?

The aerosol hygroscopicities in NorESM2/CAM6-Nor are the same as in NorESM1.2/CAM5.3-Oslo, and are as described by Kirkevåg et al. (2018) (<https://www.geosci-model-dev.net/11/3945/2018/gmd-11-3945-2018.pdf>), see Section 2.3. The Kohler equation applied (in AeroTab) is as expressed on pages 4 and 21 in <https://github.com/NorESMhub/NorESM/blob/noresm2/doc/configurations/AeroTab-slides-updateJan2020.pdf>, where the whole expression marked with an **x** on p. 21 is the sought **kappa** (here summed over all components for an internal mixture). The first NorESM reference is Kirkevåg and Iversen (2001) (<https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2001JD000886>), but see also Ghan et al. (2001) (J. Geophys. Res., 106, 5295–5316, 2001) or Pruppacher and Klett, Microphysics of Clouds and Precipitation, 954 pp., Kluwer Acad., Norwell, Mass., 1997.

### Is there enhancement of absorption due to organic coating of BC or Organic aerosols in NorESM2 (AeroTab6)?

Internal mixtures of OM or water with BC is taken into account in the optics calculations, but with the assumption that the mixture for a certain size is homogeneous (but size-dependent). There is therefore no lensing effect, since this requires a core-shell treatment (and a different, more advanced Mie code). More specifically, AeroTab calculates complex refractive indices for an internal mixture of aerosol components based on the component specific refractive indices, where the method used is simple volume mixing for all components (including water) except BC. For mixing of BC with the rest, the Maxwell Garnett mixing rule is applied (see Kirkevåg et al., 2005: ([https://www.researchgate.net/publication/265111344\\_Revised\\_schemes\\_for\\_aerosol\\_optical\\_parameters\\_and\\_cloud\\_condensation\\_nuclei\\_in\\_CCM-Oslo](https://www.researchgate.net/publication/265111344_Revised_schemes_for_aerosol_optical_parameters_and_cloud_condensation_nuclei_in_CCM-Oslo)).

In NorESM1.2/CAM5.3-Oslo simulations for AeroCom's in-situ optics study, it was surprisingly little (almost no) absorption enhancement from increased hygroscopic swelling (for the various prescribed RH values). There is probably quite little enhancement also for OM, although that is not so readily seen without doing special sensitivity tests (similar to optics the in-situ study, but with varying OM instead of water).

## Which types of AeroTab tables are read in when running NorESM2?

An overview of which AeroTab look-up tables (LUT) are used for the various model configurations is found in Sect. 5 of the AeroTab User's Guide at [https://github.com/NorESMhub/NorESM/blob/norasm2/doc/configurations/AeroTab-user-guide\\_v16april2020.pdf](https://github.com/NorESMhub/NorESM/blob/norasm2/doc/configurations/AeroTab-user-guide_v16april2020.pdf). Also the nature and content of the LUT are described there. Not specifically mentioned there is that a specific set of LUT is only read in by the model for the configurations where they are actually used.

## How to invoke and activate the different levels of aerosol diagnostics in a NorESM model run?

In particular: **Which namelist arguments do what? Do they overlap, complement each other?**

This is covered here: [https://noesm-docs.readthedocs.io/en/norasm2/output/aerosol\\_output.html#aerosol-output](https://noesm-docs.readthedocs.io/en/norasm2/output/aerosol_output.html#aerosol-output). An exception is the namelist option "history\_budget=true.", which produces a budget for cloud water mass and number analysis, but has not been technically or scientifically tested in NorESM2. If still working, this should work as in earlier versions of NorESM, and is described at <https://noesm-docs.readthedocs.io/en/norasm1/modeldiagnostics.html#cloud-water-mass-and-number-analysis-budgets>.

## 1.10.3 Post-processing and plotting FAQ

### How do I compute a weighted average?

#### Using NCL

- Examples on how to compute and plot weighted averages: <https://www.ncl.ucar.edu/Applications/ave.shtml>
- See also the examples at the bottom of the documentation for the ncl function wgt\_areaavg (which computes the weighted average): [https://www.ncl.ucar.edu/Document/Functions/Built-in/wgt\\_areaave.shtml](https://www.ncl.ucar.edu/Document/Functions/Built-in/wgt_areaave.shtml)

#### Using python

When calculating annual averages from NorESM2 data it is important use appropriate monthly weights, especially for individual radiative fluxes (can have errors of the order of 0.5-1 W/m<sup>2</sup> if not used).

The monthly files in NorESM2 (not BLOM/MICOM/iHAMOCC files) are written *after* the last time step of the month. Consequently, the date in the netcdf file is the first of the following month. E.g. The date in FILENAME.cam.h0.0001-01.nc will be 01-02-0001 (the first of *February* and not January). This needs to be taken into account when calculating annual averages using python packages like xarray and iris. One method is to use the time bounds (instead of time), another method is to correct the time stamps in the time array.

#### xarray

For BLOM/MICOM/iHAMOCC files there are no issues with the time variable, and annual averages can be calculated:

```
def annual_mean_to_file(var, fname, weights=np.array([31, 28, 31, 30, 31, 30, 31, 31, 30,
→30, 31, 30, 31])/365):
    """
    Calculate annual means from monthly means assuming no-leap calendar
    """
    month_weights = xr.DataArray(np.tile(weights, len(var.time)//12), coords=[var.
→time], name='month_weights')
    annual_mean = (month_weights*var).groupby('time.year').sum('time')
    annual_mean = annual_mean.rename({'year': 'time'})
    annual_mean = annual_mean.where(annual_mean!=0)
    annual_mean.rename(var.name).to_dataset().to_netcdf(fname)
```

One way to handle the time issue is to take annual averages by looping over 12 files at the time (slow method):

```

def area_avg(ds, var, monthw = np.array([31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
↪31])):
    '''
    Calculate global and annual means from monthly means
    '''
    field = ds[var].mean(dim = 'lon')
    return np.sum(monthw*[ np.nansum((field[i,:]*ds.gw[0]).values)/
                            np.nansum(ds.gw[0]) for i in range(0,len(ds[var].time))])/np.
↪sum(monthw)

```

Weights for ocean calculations:

```

gridpath = 'ocngrid/tnxlv4/' # path to grid files
grid = xr.open_mfdataset(gridpath + 'grid.nc')
parea = grid.parea
pmask = grid.pmask
pweight = parea*pmask

```

-

## iris

It is also possible to use iris for analysing and visualising NorESM2 data Documentation: <https://scitools.org.uk/iris/docs/latest/>

```

def get_cube_varname(cube_list, var_name):
    '''
    Subtract cube with name var_name from the cube_list
    '''
    if type(var_name) is list:
        var_cube = iris.cube.CubeList()
        for name in var_name:
            print(name)
            for cube in cube_list:
                if cube.var_name == name:
                    var_cube.append(cube)
        return sum(var_cube)
    else:
        for cube in cube_list:
            if cube.var_name == var_name:
                return cube

def subtract_second_timedim(cube):
    '''
    Fix time issue by subtracting one second from the time array
    '''
    time = cube.coord("time")
    new_points = time.points - 1/86400
    new_time = DimCoord(new_points, standard_name="time",
                        units=time.units)
    cube.remove_coord("time")
    cube.add_dim_coord(new_time, 0)
    return cube

def annual_weighted_avg(path,file, varname):
    '''
    Calculate global and annual means from monthly means

```

(continues on next page)

(continued from previous page)

```
'''
cube = iris.load(path + file)
ts = get_cube_varname(cube, varname)
cube = subtract_second_timedim(ts)
lons = cube.coord("longitude")
lats = cube.coord("latitude")
lons.guess_bounds()
lats.guess_bounds()
weights = iris.analysis.cartography.area_weights(cube)
cube_collapsed =cube.collapsed(coords=["latitude", "longitude"],
                                aggregator=iris.analysis.MEAN,
                                weights=weights)

monthw=[31,28,31,30,31,30,31,31,30,31,30,31]
monthw = np.tile(monthw, 30)
monthw=monthw/np.sum(monthw)
n=len(monthw)
tmp = [cube_collapsed[i:i+n].collapsed('time', aggregator= iris.analysis.MEAN,
↪weights=monthw) for i in range(0,n*yrs,n)]
cubes_aa = iris.cube.CubeList(tmp).merge()
return cubes_aa[0]
```





**SEARCH**

- search