CESM CICE5 Users Guide

Release CICE 5.1.2

David Bailey **Alice DuVivier** **Marika Holland**
Elizabeth Hunke **Bill Lipscomb** **Bruce Briegleb**
Cecilia Bitz **Julie Schramm**

Jun 06, 2018

CONTENTS

CHAPTER ONE

WHAT IS CICE5

This CICE User's Guide accompanies the CESM2.0 User's Guide, and is intended for those who would like to run CICE coupled, on a supported platform, and "out of the box". It includes a quick start guide for downloading the CESM2 source code and input datasets, and information on how to configure, build and run the model. The supported configurations and scripts for building the fully coupled model are also described in the CESM2 User's Guide:

http://www.cesm.ucar.edu/models/cesm2.0

The CICE User's Guide is intended for users interested in making modifications to the ice model scripts or namelists within the CESM. Users interested in modifying the source code or using the standalone version should see the CICE Code Reference/Developer's Guide *[6]*.

CICE5.1.2 is the latest version of the Los Alamos Sea Ice Model, sometimes referred to as the Community Ice CodE *[6]*. It is the result of a community effort to develop a portable, efficient sea ice model that can be run coupled in a global climate model or uncoupled as a standalone ice model. CICE5 has been released as the sea ice component of the Community Earth System Model (CESM), a fully-coupled global climate model that provides simulations of Earth's past, present, and future climate states. CICE5 in the CESM is supported on high- and low-resolution Greenland Pole and tripole grids, which are identical to those used by the Parallel Ocean Program (POP) ocean model. The high resolution version is best suited for simulating present-day and future climate scenarios while the low resolution option is used for paleoclimate simulations and debugging.

An uncoupled version of CICE5.1.2 is available separately:

https://github.com/CICE-Consortium/CICE-svn-trunk

This standalone CICE configuration provides a means of running the sea ice model independent of the other CESM components. It can read in atmospheric and ocean forcing, which eliminates the need for the flux coupler, and the atmosphere, land and ocean data models. It can be run on a reduced number of processors, or without MPI (Message Passing Interface) for researchers without access to these computer resources.

CICE is a dynamic-thermodynamic model that includes a subgrid-scale ice thickness distribution *[6]*. It uses the energy conserving thermodynamics of *[10]* or *[2]*, has multiple layers in each thickness category, and accounts for the influences of brine pockets within the ice cover. The ice dynamics utilizes the elastic-viscous-plastic (EVP) rheology of *[4]*. Sea ice ridging has the options of *[8]* and *[9]* or the newer ridging scheme of *[7]*. A slab ocean mixed layer model is included. A Scientific Reference Guide *[6]* is available that contains more detailed information on the model physics. The physics available in the uncoupled ice model are identical to those in the ice model used in the fully coupled system.

This document uses the following text conventions: Variable names used in the code are `typewritten`. Subroutine are given in *italic*. File and directory names are in **boldface**.

1.1 What's new in CICE5?

CICE5 is very similar in code structure to the previous version CICE4 and was released in March of 2015. CICE4 was an upgraded version of the Community Sea Ice Model, CSIM5, which was based on CICE3. The major changes are:

- The new mushy-layer thermodynamics (`ktherm = 2`) is the default *[10]*.
- The new level melt pond scheme (`tr_pond_lvl = .true.`) is the default *[5]*.
- The default number of ice layers is now 8 (previously 4).
- The default number of snow layers is now 3 (previously 1).
- The freezing point at the sea ice-ocean interface is now salinity dependent following *[1]*.

The CICE source code used in the CESM is based on the Los Alamos Sea Ice Model CICE model version 5. The main source code is very similar in both versions, but the drivers are significantly different. If there are topics that are not covered in this CICE documentation, users are encouraged to look at the CICE documentation available at:

https://github.com/CICE-Consortium/CICE-svn-trunk

CONFIGURING AND BUILDING CICE

2.1 Overview

The setup scripts for the coupled model are located in **cesm2/scripts**.

The directory structure of CICE5 within CESM is shown below.

```
                      cesm2         (main directory)
                        |
                        |
     components --------+--------- scripts
        |                        |
        |               * * * * *|* * * * *
       cice             *build scripts for*
        |               *  coupled model  *
        |               * * * * * * * * * *
        |
bld ----+---- cime_config ---- docs ------ src
        |                       |           |
      (Build scripts          (CICE         |
       and CIME config)    documentation)   |
                                            |
                                            |
                         drivers --- mpi ---+--- io_pio --- serial --- source
                            |
                            |
                            |
           cesm --- cice ---+--- hadley
```

The CIME scripts generate a set of “resolved scripts” for a specific configuration determined by the user. The configuration includes components, resolution, run type, and machine. The run and setup scripts that were in the **/scripts** directory for previous versions are now generated automatically. See the CESM2 User’s Guide for information on how to use the new scripts.

http://www.cesm.ucar.edu/models/cesm2

The file that contains the ice model namelist is now located in **$CASE/CaseDocs**. The file containing the environment variables used for building the executable file for the ice model is in **$CASE/env_build.xml**. The contents of the ice model namelist are described in section *CICE Namelists*.

2.2 Building the CICE library

2.2.1 The Build Environment

The **cime_config/build_cpp** script sets all compile time parameters, such as the horizontal grid, the sea ice mode (prognostic or prescribed), tracers, etc. However, to change the CPP variables, one needs to add these to the `CICE_CONFIG_OPTS` variable in the **env_build.xml** file. Additional options can be set here, such as the decomposition and the number of tasks.

2.2.2 CICE Preprocessor Flags

Preprocessor flags are activated in the form -Doption in the **buildcpp** script. Only advanced users should change these options. See the CESM User's Guide or the CICE reference guide for more information on these. The flags specific to the ice model are:

```
CPPDEFS :=  $(CPPDEFS) -DCESMCOUPLED -Dcoupled -Dncdf -DNICECAT=5 -DNXGLOB=$()
-DNYGLOB=$() -DNTRAERO=3 -DNTRISO=0 -DNBGCLYR=0 -DNICELYR=8 -DNSNWLYR=3
-DTRAGE=1 -DTRFY=1 -DTRLVL=1 -DTRPND=1 -DTRBRI=0 -DTRBGCS=0
-DBLCKX=$() -DBLCKY=$() -DMXBLCKS=$()
```

The options `-DCESMCOUPLED` and `-Dcoupled` are set to activate the coupling interface. This will include the source code in **ice_comp_mct.F90**, for example. In coupled runs, the CESM coupler multiplies the fluxes by the ice area, so they are divided by the ice area in CICE to get the correct fluxes. Note that the **ice_forcing.F90** module is not used in coupled runs.

The options `-DBLCKX=$(CICE_BLCKX)` and `-DBLCKY=$(CICE_BLCKY)` set the block sizes used in each grid direction. These values are set automatically in the scripts for the coupled model. Note that `CICE_BLCKX` and `CICE_BLCKY` must divide evenly into the grid, and are used only for MPI grid decomposition. If `CICE_BLCKX` or `CICE_BLCKY` do not divide evenly into the grid, which determines the number of blocks in each direction, the model setup will exit from the setup script and print an error message to the **ice.bldlog** (build log) file. To override these values, one must set the variable `CICE_AUTO_DECOMP` to `false` in **env_build.xml** and then the variables `CICE_BLCKX`, `CICE_BLCKY`, and `CICE_MBLCKS` can be set manually.

The flag `-DMXBLCKS` is essentially the threading option. This controls the number of "blocks" per processor. This can describe the number of OpenMP threads on an MPI task, or can simply be that a single MPI task handles a number of blocks. This is set automatically, but can be changed as described above.

The number of categories `-DNICECAT` can be changed at build time. There is a separate discussion of this in *CICE Thickness Categories*.

The number of ice and snow layers are set at compile time via the CPP flags. They can technically be changed via the `CICE_CONFIG_OPTS` variable in **env_build.xml**, but it this is not recommended. We have provided an option to use the older CICE4 physics, inluding 4 ice levels and 1 snow level. This option also turns on `ktherm=1` and `tr_pond_cesm=.true.` To use the older CICE4 physics options, one should add/change `-phys cice4` in the XML variable `CICE_CONFIG_OPTS`.

The flag `-DNTR\_AERO=n` flag turns on the aerosol deposition physics in the sea ice where n is the number of tracer species and 0 turns off the tracers. More details on this are in the section on tracers. The default here is 3 and should only be changed when adding additional aerosol tracers. This can be turned off by setting `CICE_CONFIG_OPTS` to `-ntr_aero=0` in the **env_build.xml** file.

The flag `-DNTR\_ISO=n` flag turns on the isotopes and is not yet supported.

The flags `-DBGCLYR`, `-DTRBRI`, and `-DTRBGCS` are for the skeletal biogeochemistry. These have not been tested within CESM and more information can be found in the CICE reference guide *[6]*.

The other tracer flags, `-DTRAGE`, `-DTRFY`, `-DTRLVL`, `-DTRPND` are for the age, first-year ice, level ice, and melt pond tracers. These are either on or off using 1 or 0. By default, all are turned on. More information on these can be found in the CICE reference guide *[6]*.

More information on the compile settings for CICE can be found here:

http://www.cesm.ucar.edu/models/cesm2/component_settings/cice_input.html

CHAPTER

THREE

CICE NAMELISTS

CICE uses the same namelists for both the coupled and uncoupled models. This section describes the namelist variables, which determine time management, output frequency, model physics, and filenames. The ice namelists for the coupled model are now located in **$CASE/CaseDocs**. Some additional documentation on the CICE namelist is available here:

http://www.cesm.ucar.edu/models/cesm2/component_settings/cice_nml.html

A script reads the input namelist at runtime, and writes the namelist information to the file **ice_in** in the directory where the model executable is located. Therefore, the namelist will be updated even if the ice model is not recompiled. The default values of the ice setup, grid, tracer, and physics namelists are set in **ice_init.F90**. The prescribed ice option along with the history namelist variables are set in **ice_prescribed.F90** and **ice_history.F90** respectively. If they are not set in the namelist in the script, they will assume the default values listed in the following tables, which list all available namelist parameters. The default values shown here are for the coupled model, which is set up for a production run. Only a few of these variables are required to be set in the namelist; these values are noted in the paragraphs below. An example of the default namelist is shown in Section *CICE Namelist Examples*.

The main run management namelist options are shown in *Table 1: Setup Namelist Options*. While additional namelist variables are available in the uncoupled version, they are set by the driver in CESM. For a full list of namelist variables, you should consult the CICE Reference Guide *[6]*.

Variables set by the driver include: `dt`, `runid`, `runtype`, `istep0`, `days_per_year`, `restart` and `dumpfreq`. These should be changed in the CESM configuration files.

Table 3.1: Table 1: Setup Namelist Options

Variable Name	Type	Default	Description
&setup_nml			
`ice_ic`	character	default	Filename for initial and branch runs. Set by driver scripts
`pointer_file`	character	’rpointer.ice’	Pointer file that contains the name of the restart file
`restart_file`	character	none	Restart file prefix. Set by driver.
`restart_format`	character	none	Restart file format. bin = binary, nc = netcdf, pio = use pio library (default).
`restart_ext`	logical	.false.	Write ghost cells as a part of restarts
`history_file`	character	‘unknown’	History file prefix. Set by driver. ’default’ uses default initialization. ’none’ initializes with no ice.
`days_per_year`	integer	365	Standard number of days per year for calendar. Does interact with Gregorian calendar setting. Set by driver.
`year_init`	integer	1	Used in leap year calculation. Do not change
`ndtd`	integer	1	Number of dynamic timesteps per thermodynamic timestep
`histfreq`	char array	’m’,’x’,’x’,’x’,’x’	Unit for frequency of output written to history streams
			’H’ or ’h’ writes hourly data
			’D’ or ’d’ writes daily data

Continued on next page

Table 3.1 – continued from previous page

Variable Name	Type	Default	Description
			’M’ or ’m’ writes monthly data
			’Y’ or ’y’ writes yearly data
			’1’ writes every timestep
			’x’ no history data is written
`histfreq_n`	integer	1,1,1,1,1	Frequency of histfreq history data is written to each stream
`dumpfreq`	character	‘x’	Unit for frequency of dump files. Set by driver.
`dumpfreq_n`	integer	1	Frequency of dumpfreq dump files. Set by driver.
`hist_avg`	logical	.true.	If true, averaged history information is written out at a frequency determined by histfreq. If false, instantaneous values are written in all streams.
`write_ic`	logical	.true.	If true, write initial conditions
`diagfreq`	integer	24	Frequency of diagnostics written (min, max, hemispheric sums) to standard output.
			‘24’ = diagnostics written once every 24 timesteps
			‘1’ = diagnostics written each timestep
			‘0’ = no diagnostics written
`print_global`	logical	.true.	Print global diagnostics
`print_points`	logical	.true.	Print diagnostics at latpnt and lonpnt
`latpnt`	float arr	90.0, -65.0	Latitudes for diagnostic points (`print_points`)
`lonpnt`	float arr	0.0, -45.0	Longitudes for diagnostic points (`print_points`)
`lcdf64`	logical	.false.	Use 64-bit offset in netcdf files
`bfbflag`	logical	.false.	Require bit-for-bit global sums

3.1 Changing the timestep

`dt` is the timestep in seconds for the ice model thermodynamics. The thermodynamics component is stable but not necessarily accurate for any value of the timestep. The value chosen for `dt` depends on the stability of the transport and the grid resolution. A conservative estimate of `dt` for the transport using the upwind advection scheme is:

$$\Delta t < \frac{min(\Delta x, \Delta y)}{4 * max(u, v)}.$$

Maximum values for `dt` for the two standard CESM POP grids, assuming $max(u, v) = 0.5\ m/s$, are shown in *Table 2: Recommended timesteps*. The default timestep for CICE is 30 minutes for gx1, which must be equvielent to the coupling interval (`NCPL_ICE` and `NCPL_ATM`) set in the CESM configuration files **env_run.xml**. One should only change the CICE timestep using the `NCPL_ATM` variable in **env_run.xml**. For more on this see:

http://www.cesm.ucar.edu/models/cesm2/component_settings/drv_input_cesm.html

Table 3.2: Table 2: Recommended timesteps

Grid	$min(\Delta x, \Delta y)$	$max\Delta t$
gx3	28845.9 m	4.0 hr
gx1	8558.2 m	1.2 hr

Occasionally, ice velocities are calculated that are larger than what is assumed when the model timestep is chosen. This causes a CFL violation in the transport scheme. A namelist option was added (`ndtd`) to subcycle the dynamics to get through these instabilities that arise during long integrations. The default value for this variable is one, and is typically increased to two when the ice model reaches an instability. The value in the namelist should be returned to one by the user when the model integrates past that point.

3.2 Writing Output

The namelist variables that control the frequency of the model diagnostics, netCDF history, and restart files are shown in *Table 1: Setup Namelist Options*. By default, diagnostics are written out once every 48 timesteps to the ascii file **ice.log.$LID** (see section *Stdout Output*). $LID is a time stamp that is set in the main script.

The namelist variable `histfreq` controls the output frequency of the netCDF history files; writing monthly averages is the default. The content of the history files is described in section *CICE History Files*. The value of `hist_avg` determines if instantaneous or averaged variables are written at the frequency set by `histfreq`. If `histfreq` is set to `1` for instantaneous output, `hist_avg` is set to `.false.` within the source code to avoid conflicts. The latest version of CICE allows for multiple history streams, currently set to a maximum of 5. The namelist variables, `histfreq` and `histfreq_n` are now arrays which allow for different frequency history file sets. More detail on this is available in *CICE History Files*.

The namelist variable `pointer_file` is set to the name of the pointer file containing the restart file name that will be read when model execution begins. The pointer file resides in the scripts directory and is created initially by the ice setup script but is overwritten every time a new restart file is created. It will contain the name of the latest restart file. The default filename **ice.restart_file** shown in *Table 1: Setup Namelist Options* will not work unless some modifications are made to the ice setup script and a file is created with this name and contains the name of a valid restart file; this variable must be set in the namelist. More information on restart pointer files can be found in Section *CICE Restart Files*.

The variables `dumpfreq` and `dumpfreq_n` control the output frequency of the netCDF restart files; writing one restart file per year is the default and is set by the CESM driver. The default format for all reads and writes of files in CESM is now pio, but this can be changed to binary or netCDF through the namelist variable, `restart_format`.

The Parallel Input/Output libraries or “PIO” are used within the CESM for more efficient reading and writing. PIO includes options for binary, netCDF version3, parallel netCDF, or netCDF version 4 parallel. More on this can be found here: http://ncar.github.io/ParallelIO/

If `print_points` is `.true.`, diagnostic data is printed out for two grid points, one near the north pole and one near the Weddell Sea. The points are set via namelist variables `latpnt` and `lonpnt`. This option can be helpful for debugging.

3.3 Model Physics

Some of the most commonly used namelist variables for the ice model physics are listed in the following tables. More information can be found in the CICE reference guide at *[6]*.

The calculation of the ice velocities is subcycled `ndte` times per timestep so that the elastic waves are damped before the next timestep. The subcycling timestep is calculated as $dte = dt/ndte$ and must be sufficiently smaller than the damping timescale T, which needs to be sufficiently shorter than dt.

$$dte < T < dt$$

This relationship is discussed in *[6]*. The best ratio for $[dte : T : dt]$ is [1:40:120]. Typical combinations of (`dt`, `ndte`) are (3600., 120), (7200., 240) (10800., 120). The default `ndte` is `120` as set in **ice_init.F90**.

`kitd` determines the scheme used to redistribute sea ice within the ice thickness distribution (ITD) as the ice grows and melts. The linear remapping scheme is the default and approximates the thickness distribution in each category as a linear function. The delta function method represents *g(h)* in each category as a delta function. This method can leave some categories mostly empty at any given time and cause jumps in the properties of *g(h)*.

kdyn determines the ice dynamics used in the model. The default is the elastic-viscous-plastic (EVP) dynamics (kdyn = 1). If kdyn is set to 0, the ice dynamics is inactive. In this case, ice velocities are not computed and ice is not transported. Since the initial ice velocities are read in from the restart file, the maximum and minimum velocities written to the log file will be non-zero in this case, but they are not used in any calculations.

The value of kstrength determines which formulation is used to calculate the strength of the pack ice. The calculation depends on mean ice thickness and open water fraction. The calculation is based on energetics and should not be used if the ice that participates in ridging is not well resolved.

The variable advection determines the horizontal transport scheme used. The default scheme is the incremental remapping method (advection = “remap”). This method is less diffusive and is computationally efficient for large numbers of categories or tracers than other options. The upwind scheme is also available, but this scheme is only first order accurate.

Table 3.3: Table 3: Dynamics Namelist Options

Variable Name	Type	Default	Description
&dynamics_nml			
kdyn	Integer	1	Determines ice dynamics, 0 = No ice dynamics, 1 = Elastic viscous plastic dynamics
revised_evp	Logical	.false.	Revised EVP formulation
ndte	Integer	1	Number of sub-cycles in EVP dynamics.
advection	Character	remap	Determines horizontal advection scheme. ’remap’ = incremental remapping, ’upwind’ = first order advection
kstrength	Integer	1	Determines pressure formulation, 0 = parameterization, 1 = parameterization
krdg_partic	Integer	1	Ridging participation function, 0 = Thorndike, 1 = Expontential
krdg_redist	Integer	1	Ridging distribution function, 0 = Hibler , 1 = Expontential
mu_rdg	Real	4.0	e-folding scale of ridged ice
cf	Real	17.0	Ratio of ridging work to PE change

A new thermodynamics option (ktherm = 2) is now the default. This is the so-called mushy-layer thermodyanmics of *[10]*. The basic idea of this is that prognostic salinity is now used in the vertical thermodynamic calculation where this used to be a constant profile. The CESM1 and older option of *[3]*, (ktherm = 1) is still available. There are several additional thermodynamic options not listed that go with ktherm = 2, that are described more thoroughly in *[6]*.

Table 3.4: Table 4: Thermodynamics Namelist Options

Variable Name	Type	Default	Description
&thermo_nml			
kitd	Integer	1	Determines ITD conversion, 0 = delta scheme, 1=linear remapping
ktherm	Integer	1	Determines ice thermodynamics, 1 = BL99, 2 = mushy layer
conduct	Character	MU71	Determines conductivity formulation used with ktherm = 1, MU71, bubbly

For the newer delta-Eddington shortwave radiative transfer scheme shortwave = dEdd, the base albedos are computed based on the inherent optical properties of snow, sea ice, and melt ponds. These albedos are most commonly changed through adjustments to the snow grain radius, R_snw, temperature to transition to melting snow, dT_mlt_in, and maximum snow grain radius, rsnw_mlt_in. Note, the older CCSM3 radiation scheme is still available through shortwave = default.

Table 3.5: Table 5: Radiation Namelist Options

Variable Name	Type	Default: CESM-CAM4 gx3	Default: CESM-CAM4 gx1	Default: CESM-CAM5 gx1	Description
&shortwave_nml					
`shortwave`	Character	dEdd	dEdd	dEdd	Shortwave Radiative Transfer Scheme, 'dEdd' = delta-Eddington Shortwave, 'default' = CCSM3 Shortwave
`albicev`	Real	0.68	0.75	0.75	Visible ice albedo (CCSM3)
`albicei`	Real	0.30	0.45	0.45	Near-infrared ice albedo (CCSM3)
`albsnowv`	Real	0.91	0.98	0.98	Visible snow albedo (CCSM3)
`albsnowi`	Real	0.63	0.73	0.73	Near-infrared snow albedo (CCSM3)
`r_ice`	Real	0.0	0.0	0.0	Base ice tuning parameter (dEdd)
`r_pnd`	Real	0.0	0.0	0.0	Base pond tuning parameter (dEdd)
`r_snw`	Real	-2.0	1.5	1.75	Base snow grain radius tuning parameter (dEdd)
`dt_mlt`	Real	2.0	1.5	1.0	Snow melt onset temperature parameter (dEdd)
`rsnw_mlt`	Real	2000.	1500.	1000.	Snow melt maximum radius (dEdd)

3.4 Tracer Namelist

The namelist parameters listed in *Table 6: Tracer Namelist Options* are for adding tracers. The tracers should be added through the CESM driver scripts via the `CICE_CONFIG_OPTS` variable.

Table 3.6: Table 6: Tracer Namelist Options

Variable Name	Type	Default	Description
&tracer_nml			
`tr_aero`	Logical	.true.	Aerosol physics and tracer
`restart_aero`	Logical	.false.	Initialize aerosols to zero or from file.
`tr_iage`	Logical	.true.	Ice age passive tracer
`restart_age`	Logical	.false.	Initialize iage to zero or from file.
`tr_FY`	Logical	.true.	First-year ice area passive tracer
`restart_FY`	Logical	.false.	Initialize first-year ice to zero or from file.
`tr_lvl`	Logical	.false.	Level ice area passive tracer
`restart_lvl`	Logical	.false.	Initialize level ice to zero or from file.
`tr_pond_cesm`	Logical	.false.	The older CESM melt pond option.
`restart_pond_cesm`	Logical	.false.	Initialize CESM ponds to zero or from file.
`tr_pond_lvl`	Logical	.true.	The Hunke et al. level ice pond formulation
`restart_pond_lvl`	Logical	.false.	Initialize level ponds to zero or from file.
`tr_pond_topo`	Logical	.true.	The Felthem et al. topographic pond formulation
`restart_pond_topo`	Logical	.false.	Initialize topgraphic ponds to zero or from file.

3.5 Prescribed Ice Namelist

The namelist parameters listed in *Table 7: Prescribed Ice Namelist Options* are for the prescribed ice option as used in AMIP and F compset (standalone CAM) runs [prescribed].

Table 3.7: Table 7: Prescribed Ice Namelist Options

Variable Name	Type	Default	Description
`prescribed_ice`	Logical	.false.	Flag to turn on prescribed ice
`prescribed_ice_fill`	Logical	.false.	Flag to turn fill option
`stream_year_first`	Integer	1	First year of prescribed ice data
`stream_year_last`	Integer	1	Last year of prescribed ice data
`model_year_align`	Integer	1	Year in model run that aligns with stream_year_first
`stream_domfilename`	Character	none'	Prescribed ice stream data file
`stream_fldfilename`	Character	none	Prescribed ice stream data file
`stream_fldvarname`	Character	ice_cov	Ice fraction field name

3.6 Grid Namelist

The namelist parameters listed in *Table 8: Grid Namelist Options* are for grid and mask information. During execution, the ice model reads grid and land mask information from the files `grid_file` and `kmt_file` that should be located in the executable directory. There are commands in the scripts that copy these files from the input data directory, rename them from **global_$ICE_GRID.grid** and **global_$ICE_GRID.kmt** to the default filenames shown in *Table 8: Grid Namelist Options*.

Table 3.8: Table 8: Grid Namelist Options

Variable Name	Type	Default	Description
&grid_nml			
`grid_type`	Character	displaced_pole	Determines grid type.
			displaced_pole
			tripole
			rectangular
`grid_format`	Character	binary	Grid file format (binary or netCDF)
`grid_file`	Character	data.domain.grid	Input filename containing grid information.
`gridcpl_file`	Character	data.domain.grid	Input filename containing grid information if coupling grid is different than computational grid.
`kmt_file`	Character	data.domain.kmt	Input filename containing land mask information.
`kcatbound`	Integer	0	How category boundaries are set (0 or 1)

For coupled runs, supported grids include the 'displaced_pole' grids (gx3 and gx1) and the 'tripole' grids.

3.7 Domain Namelist

The namelist parameters listed in *Table 9: Domain Namelist Options* are for computational domain decomposition information. These are generally set in the build configure scripts through the variables `CICE_DECOMPTYPE` and `CICE_DECOMPSETTING` based on the number of processors. See the CESM scripts documentation.

Table 3.9: Table 9: Domain Namelist Options

Variable Name	Type	Default	Description
&domain_nml			
processor_shape	Character	square-ice	Approximate block shapes
			slenderX1
			slenderX2
			square-ice
			square-pop
			blocks
distribution_type	Character	spacecurve	How domain is split into blocks and distributed onto processors
			cartesian
			rake
			roundrobin
			sectcart
			sectrobin
			spacecurve
distribution_wght	Character	latitude	How blocks are weighted when using space-filling curves
			block
			latitude
			erfc
			file
distribution_wght_file	Character	none	File containing space-filling curve weights when using file weighting
ew_boundary_type	Character	cyclic	Boundary conditions in E-W direction
ns_boundary_type	Character	open	Boundary conditions in N-S direction
maskhalo_dyn	Logical	.true.	Use masked halos in dynamics.
maskhalo_remap	Logical	.true.	Use masked halos in remapping.
maskhalo_bound	Logical	.true.	Use masked halos in state bound.

3.8 PIO Namelist

PIO settings are now handled via the CESM driver.

CHAPTER

FOUR

CICE NAMELIST EXAMPLES

This section shows several examples of namelists from the coupled ice model. These examples are taken directly from **$CASE/CaseDocs/ice_in** for the coupled model. Most of the variables in the namelist are determined from environment variables set elsewhere in the scripts. Since the namelists from the coupled model are "resolved" by the scripts, meaning that the values of most of the shell script variables are put directly into the namelist, examples are shown for the most commonly used configurations. Variables that are commonly changed directly in the namelist are the timestep dt and the number of subcycles per timestep in the ice dynamics ndte.

4.1 Example 1: CESM Fully Coupled

The following example is the namelist used for CESM fully coupled, or the B configuration. A completely resolved version of the namelist will be written to **$CASE/CaseDocs/ice_in** and **ice_in** in the executable directory. While this list includes additional physics and biogeochemistry options, we have not tested these. More information can be found in the CICE Reference Guide *[6]*. Note that modifications to the CICE namelist go in **$CASE/user_nl_cice**.

```
&setup_nml
  bfbflag = .false.
  days_per_year = 365
  diagfreq = 24
  dumpfreq = "x"
  hist_avg = .true.
  histfreq = "m", "x", "x", "x", "x"
  histfreq_n = 1, 0, 0, 0, 0
  history_file = "unknown"
  history_precision = 4
  ice_ic = "b.e20.B1850.f09_g17.pi_control.all.297.cice.r.0130-01-01-00000.nc"
  latpnt = 90.0, -65.0
  lcdf64 = .true.
  lonpnt = 0.0, -45.0
  ndtd = 1
  pointer_file = "./rpointer.ice"
  print_global = .true.
  print_points = .false.
  restart_ext = .false.
  restart_file = ""
  restart_format = "pio"
  write_ic = .false.
  year_init = 1
/
&grid_nml
  grid_file = "/glade/p/cesmdata/cseg/inputdata/ocn/pop/gx1v7/grid/horiz_grid_
↪20010402.ieeer8"
  grid_format = "bin"
  grid_type = "displaced_pole"
  gridcpl_file = "unknown_gridcpl_file"
  kcatbound = 0
  kmt_file = "/glade/p/cesmdata/cseg/inputdata/ocn/pop/gx1v7/grid/topography_20161215.
↪ieeei4"
/
&tracer_nml
  restart_aero = .false.
  restart_age = .false.
  restart_fy = .false.
  restart_iso = .false.
  restart_lvl = .false.
  restart_pond_cesm = .false.
  restart_pond_lvl = .false.
  restart_pond_topo = .false.
  tr_aero = .true.
  tr_fy = .true.
  tr_iage = .true.
  tr_iso = .false.
  tr_lvl = .true.
  tr_pond_cesm = .false.
  tr_pond_lvl = .true.
  tr_pond_topo = .false.
/
&thermo_nml
  a_rapid_mode = 0.5e-03
  aspect_rapid_mode = 1.0
  conduct = "MU71"
  dsdt_slow_mode = -1.5e-07
  kitd = 1
  ktherm = 2
  phi_c_slow_mode = 0.05
  phi_i_mushy = 0.85
  rac_rapid_mode = 10
/
&dynamics_nml
  advection = "remap"
  cf = 17.0
  kdyn = 1
  krdg_partic = 1
  krdg_redist = 1
  kstrength = 1
  mu_rdg = 4.0
  ndte = 120
  revised_evp = .false.
/
&shortwave_nml
  ahmax = 0.3
  albedo_type = "default"
  albicei = 0.45
  albicev = 0.75
  albsnowi = 0.73
  albsnowv = 0.98
  dt_mlt = 1.50
  kalg = 0.0
  r_ice = 0.0
  r_pnd = 0.0
  r_snw = 1.25
  rsnw_mlt = 1500.
  shortwave = "dEdd"
/
&ponds_nml
  dpscale = 1.0e-3
  frzpnd = "cesm"
  hp1 = 0.01
  hs0 = 0.03
  hs1 = 0.03
  pndaspect = 0.8
  rfracmax = 0.85
  rfracmin = 0.15
/
&forcing_nml
  fbot_xfer_type = "constant"
  formdrag = .false.
  highfreq = .true.
  l_mpond_fresh = .false.
  natmiter = 5
/
&domain_nml
  distribution_type = "spacecurve"
  distribution_wght = "latitude"
  ew_boundary_type = "cyclic"
  maskhalo_bound = .true.
  maskhalo_dyn = .true.
  maskhalo_remap = .true.
  ns_boundary_type = "open"
  processor_shape = "square-ice"
/
&zbgc_nml
  bgc_data_dir = "unknown_bgc_data_dir"
  bgc_flux_type = "Jin2006"
  nit_data_type = "unknown"
  phi_snow = 0.5
  restart_bgc = .false.
  restart_hbrine = .false.
  restore_bgc = .false.
  sil_data_type = "unknown"
  skl_bgc = .false.
  tr_bgc_am_sk = .false.
  tr_bgc_c_sk = .false.
  tr_bgc_chl_sk = .false.
  tr_bgc_dms_sk = .false.
  tr_bgc_dmspd_sk = .false.
  tr_bgc_dmspp_sk = .false.
  tr_bgc_sil_sk = .false.
  tr_brine = .false.
/
```

4.2 Example 2: History File Namelist

The next sets of namelists control what variables are written to the history file. Variables that are not output are set in the namelists `icefields*_nml`. Some of the following fields are not written to the history file since they can be retrieved from the ocean history files. The melt and freeze onset fields are not used, since the information they contain may not be correct if the model is restarted mid-year. It is better to use daily data to compute these quantities. The ice areas and volumes for categories six through ten are not used, since the default thickness distribution consists of five ice categories.

```
&icefields_bgc_nml
  f_aero = "mxxxx"
  f_aeron = "xxxxx"
  f_bgc_am_ml = "xxxxx"
  f_bgc_am_sk = "xxxxx"
  f_bgc_c = "xxxxx"
  f_bgc_c_sk = "xxxxx"
  f_bgc_chl = "xxxxx"
  f_bgc_chl_sk = "xxxxx"
  f_bgc_dms = "xxxxx"
  f_bgc_dms_ml = "xxxxx"
  f_bgc_dms_sk = "xxxxx"
  f_bgc_dmsp_ml = "xxxxx"
  f_bgc_dmspd = "xxxxx"
  f_bgc_dmspd_sk = "xxxxx"
  f_bgc_dmspp = "xxxxx"
  f_bgc_dmspp_sk = "xxxxx"
  f_bgc_n = "xxxxx"
  f_bgc_n_sk = "xxxxx"
  f_bgc_nh = "xxxxx"
  f_bgc_nit_ml = "xxxxx"
  f_bgc_nit_sk = "xxxxx"
  f_bgc_no = "xxxxx"
  f_bgc_s = "xxxxx"
  f_bgc_sil = "xxxxx"
  f_bgc_sil_ml = "xxxxx"
  f_bgc_sil_sk = "xxxxx"
  f_bphi = "xxxxx"
  f_btin = "xxxxx"
  f_faero_atm = "mxxxx"
  f_faero_ocn = "mxxxx"
  f_fbri = "xxxxx"
  f_fn = "xxxxx"
  f_fn_ai = "xxxxx"
  f_fnh = "xxxxx"
  f_fnh_ai = "xxxxx"
  f_fno = "xxxxx"
  f_fno_ai = "xxxxx"
  f_fsil = "xxxxx"
  f_fsil_ai = "xxxxx"
  f_grownet = "xxxxx"
  f_hbri = "xxxxx"
  f_ppnet = "xxxxx"
/
&icefields_drag_nml
  f_cdn_atm = "xxxxx"
  f_cdn_ocn = "xxxxx"
  f_drag = "xxxxx"
/
&icefields_mechred_nml
  f_alvl = "xxxxx"
  f_aparticn = "xxxxx"
  f_araftn = "xxxxx"
  f_ardg = "xxxxx"
  f_ardgn = "xxxxx"
  f_aredistn = "xxxxx"
  f_dardg1dt = "xxxxx"
  f_dardg1ndt = "xxxxx"
  f_dardg2dt = "xxxxx"
  f_dardg2ndt = "xxxxx"
  f_dvirdgdt = "xxxxx"
  f_dvirdgndt = "xxxxx"
  f_krdgn = "xxxxx"
  f_opening = "xxxxx"
  f_vlvl = "xxxxx"
  f_vraftn = "xxxxx"
  f_vrdg = "xxxxx"
  f_vrdgn = "xxxxx"
  f_vredistn = "xxxxx"
/
&icefields_pond_nml
  f_apeff = "xxxxx"
  f_apeff_ai = "xxxxx"
  f_apeffn = "xxxxx"
  f_apond = "mxxxx"
  f_apond_ai = "mxxxx"
  f_apondn = "mxxxx"
  f_hpond = "mxxxx"
  f_hpond_ai = "mxxxx"
  f_hpondn = "mxxxx"
  f_ipond = "mxxxx"
  f_ipond_ai = "mxxxx"
/
&icefields_nml
  f_a11 = "xxxxx"
  f_a12 = "xxxxx"
  f_aice = "mxxxx"
  f_aicen = "mxxxx"
  f_aisnap = "xxxxx"
  f_albice = "mxxxx"
  f_albpnd = "mxxxx"
  f_albsni = "mxxxx"
  f_albsno = "mxxxx"
  f_alidf = "xxxxx"
  f_alidf_ai = "mxxxx"
  f_alidr = "xxxxx"
  f_alidr_ai = "mxxxx"
  f_alvdf = "xxxxx"
  f_alvdf_ai = "mxxxx"
  f_alvdr = "xxxxx"
  f_alvdr_ai = "mxxxx"
  f_angle = .true.
  f_anglet = .true.
  f_blkmask = .true.
  f_bounds = .false.
  f_cmip = "xxxxx"
  f_congel = "mxxxx"
  f_coszen = "xxxxx"
  f_daidtd = "mxxxx"
  f_daidtt = "mxxxx"
  f_divu = "mxxxx"
  f_dsnow = "xxxxx"
```

```
f_dvidtd = "mxxxx"
f_dvidtt = "mxxxx"
f_dxt = .false.
f_dxu = .false.
f_dyt = .false.
f_dyu = .false.
f_e11 = "xxxxx"
f_e12 = "xxxxx"
f_e22 = "xxxxx"
f_evap = "mxxxx"
f_evap_ai = "xxxxx"
f_fcondtop_ai = "mxxxx"
f_fcondtopn_ai = "mxxxx"
f_fhocn = "mxxxx"
f_fhocn_ai = "mxxxx"
f_flat = "mxxxx"
f_flat_ai = "mxxxx"
f_flatn_ai = "mxxxx"
f_flwdn = "mxxxx"
f_flwup = "mxxxx"
f_flwup_ai = "xxxxx"
f_fmeltt_ai = "mxxxx"
f_fmelttn_ai = "mxxxx"
f_frazil = "mxxxx"
f_fresh = "mxxxx"
f_fresh_ai = "mxxxx"
f_frz_onset = "xxxxx"
f_frzmlt = "xxxxx"
f_fsalt = "mxxxx"
f_fsalt_ai = "mxxxx"
f_fsens = "mxxxx"
f_fsens_ai = "xxxxx"
f_fsensn_ai = "xxxxx"
f_fsurf_ai = "mxxxx"
f_fsurfn_ai = "mxxxx"
f_fswabs = "mxxxx"
f_fswabs_ai = "mxxxx"
f_fswdn = "mxxxx"
f_fswfac = "mxxxx"
f_fswint_ai = "mxxxx"
f_fswthru = "mxxxx"
f_fswthru_ai = "mxxxx"
f_fswup = "mxxxx"
f_fy = "xxxxx"
f_hi = "mxxxx"
f_hisnap = "xxxxx"
f_hs = "mxxxx"
f_hte = .false.
f_htn = .false.
f_iage = "xxxxx"
f_icepresent = "mxxxx"
f_keffn_top = "xxxxx"
f_meltb = "mxxxx"
f_meltl = "mxxxx"
f_melts = "mxxxx"
f_meltt = "mxxxx"
f_mlt_onset = "xxxxx"
f_ncat = .true.
```

```
  f_qref = "mxxxx"
  f_rain = "mxxxx"
  f_rain_ai = "xxxxx"
  f_s11 = "xxxxx"
  f_s12 = "xxxxx"
  f_s22 = "xxxxx"
  f_shear = "mxxxx"
  f_sice = "xxxxx"
  f_sig1 = "mxxxx"
  f_sig2 = "mxxxx"
  f_sinz = "mxxxx"
  f_snoice = "mxxxx"
  f_snow = "mxxxx"
  f_snow_ai = "xxxxx"
  f_snowfrac = "mxxxx"
  f_snowfracn = "mxxxx"
  f_sss = "mxxxx"
  f_sst = "mxxxx"
  f_strairx = "mxxxx"
  f_strairy = "mxxxx"
  f_strcorx = "mxxxx"
  f_strcory = "mxxxx"
  f_strength = "mxxxx"
  f_strintx = "mxxxx"
  f_strinty = "mxxxx"
  f_strocnx = "mxxxx"
  f_strocny = "mxxxx"
  f_strtltx = "mxxxx"
  f_strtlty = "mxxxx"
  f_tair = "mxxxx"
  f_tarea = .true.
  f_tinz = "mxxxx"
  f_tmask = .true.
  f_tref = "mxxxx"
  f_trsig = "xxxxx"
  f_tsfc = "mxxxx"
  f_tsnz = "xxxxx"
  f_uarea = .true.
  f_uatm = "mxxxx"
  f_uocn = "xxxxx"
  f_uvel = "mxxxx"
  f_vatm = "mxxxx"
  f_vgrdb = .true.
  f_vgrdi = .true.
  f_vgrds = .true.
  f_vicen = "mxxxx"
  f_vocn = "xxxxx"
  f_vsnon = "mxxxx"
  f_vvel = "mxxxx"
  f_yieldstress11 = "xxxxx"
  f_yieldstress12 = "xxxxx"
  f_yieldstress22 = "xxxxx"
/
```

CHAPTER

FIVE

CICE INPUT DATA

All runs

The coupled CICE model requires a minimum of two files to run. Both are set in the `&grid_nml` section of the namelist (see *Table 8: Grid Namelist Options*) for more information

- `grid_file` is a binary or netcdf file containing grid information such as the latitude, longitude, grid cell area, etc.
- `kmt_file` is a binary or netcdf file containing land mask information. This points to the ocean model KMT file or the depths of the ocean columns.

Depending on the grid selected in the scripts, the appropriate `grid_file` and `kmt_file` files will be used in the executable directory. These files are read directly from the system input data directory and not copied to the executable directory. Currently, only the POP resolutions of gx3, gx1, tx1, and tx0.1 grids are supported for the ice and ocean models. Note that these files can now be used in netCDF format.

Initial and Hybrid runs

For initial or hybrid runs, a third variable is required and is set in the `&setup_nml` section of the namelist (see *Table 1: Setup Namelist Options*).

- `ice_ic = 'none'` initializes the sea ice to zero everywhere
- `ice_ic = 'default'` initializes the sea ice to 100% concentration where the SST is below the freezing point to a thickness of 2 m in the Northern Hemisphere or 1 m in the Southern Hemisphere.
- `ice_ic = 'filename.nc'` will read the state information from an initial file named “filename.nc”. The resolution of this file must match that in `grid_file`, as set above.

Restart and Branch runs

Restart or branch runs are discussed later.

The input datasets are generally handled by the CESM driver.

http://www.cesm.ucar.edu/models/cesm2

CHAPTER

SIX

CICE THICKNESS CATEGORIES

The number of ice thickness categories affects ice model input files in three places:

- `$DNICECAT` in the scripts
- The source code module **ice_domain_size.F90**
- The initial condition (restart) file in the input file directory

One must be very careful with changing the number of thickness categories as it impacts a number of places in the code. The number of ice thickness categories can be changed in **$CASE/env_build.xml** using the xml variable `CICE_CONFIG_OPTS`. One changes this by adding `-ncat 5` to the variable `CICE_CONFIG_OPTS`. The default value is 5 categories. `$DNICECAT` is used to determine the CPP variable setting `NICECAT` in **ice_domain_size.F90**. More information on the CPP variables can be found here:

http://www.cesm.ucar.edu/models/cesm2/component_settings/cice_input.html

The information in the initial restart file is dependent on the number of ice thickness categories and the total number of layers in the ice distribution. An initial condition file exists only for the default case of 5 ice thickness categories, with four layers in each category. To create an initial condition file for a different number of categories or layers, these steps should be followed:

- Set ncat to the desired number of categories in **$CASE/env_build.xml**.
- Set the namelist variable `dumpfreq = ’m’` in **$CASE/user_nl_cice** to print out restart files monthly.
- Set the namelist variable `ice_ic='default'` in **$CASE/user_nl_cice** to use the initial conditions within the ice model.
- Run the model to equilibrium.
- The last restart file can be used as an initial condition file.
- Change the name of the last restart file to **iced.0001-01-01.$GRID.nc**.
- Copy the file into the input data directory or directly into the the executable directory.
- There are a few restart files available in **$DIN_LOC_ROOT/ice/cice**.

`$GRID` is the name of the POP grid with resolution, `$RES` of 100x116 (gx3) and 320x384 (gx1) for low and medium resolution grids, respectively. Note that the date printed inside the binary restart file will not be the same as 0001-01-01. For coupled runs, `$BASEDATE` will be the starting o date and the date inside the file will not be used.

Note: To use one ice thickness category, the following changes will need to be made in the namelist and also adding `-ncat 1` to `CICE_CONFIG_OPTS`.

```
, kitd          = 0
, kstrength     = 0
```

With these settings, the model will use the delta scheme instead of linear remapping and a strength parameterization based on open water area and mean ice thickness.

CHAPTER

SEVEN

CICE OUTPUT

The ice model produces three types of output data.

1. A file containing ASCII text, also known as a log file, is created for each run that contains information about how the run was set up and how it progressed. See the *standard output* section.

2. A series of netCDF history files containing gridded instantaneous or time-averaged output are also generated during a run. See the *history files* section.

3. A series of binary restart files necessary to continue the run are created. See the *restart files* section.

These are described in the following sections.

CHAPTER

EIGHT

CICE HISTORY FILES

History files contain gridded data values written at specified times during a model run. By default, the history files will be written to the directory run directory defined in the CESM driver. The netCDF file names are prepended by the character string set by the CESM driver. This character string has been set according to CESM Output Filename Requirements. The user can specify the frequency at which the data are written. Options are also available to record averaged or instantaneous data. The form of the history file names are as follows:

- Yearly averaged: **$CASE.cice.h?.yyyy.nc**
- Monthly averaged: **$CASE.cice.h?.yyyy-mm.nc**
- Daily averaged: **$CASE.cice.h?.yyyy-mm-dd.nc**
- Hourly averaged: **$CASE.cice.h?.yyyy-mm-dd-sssss.nc**
- Instantaneous (`hist_avg` = .true.): **$CASE.cice.h?.yyyy-mm-dd-sssss.nc**
- Instantaneous (written every dt, `histfreq` = 1): **$CASE.cice.h?.yyyy-mm-dd-sssss.nc**

$CASE is set in the main setup script. Note that the **?** denotes the multiple stream option where the first stream is just .h. and subsequent streams are h1, h2, etc. All history files are written in the executable directory. Changes to the frequency and averaging will affect all output fields. The best description of the history data comes from the file itself using the netCDF command `ncdump -h filename.nc`. Variables containing grid information are written to every file and are listed in *Table 10: Required Grid History Variables*. There are additional optional grid variables available in *Table 11: Optional Grid History Variables*. In addition to the history files, a netCDF file containing a snapshot of the initial ice state can be created at the start of each run by setting the namelist variable `write_ic=.true.`. The file name is **$CASE.cice.i.yyyy-mm-dd-sssss.nc** and is written in the executable directory. Note that variables without the `f_` string in front are always written with every run *Table 10: Required Grid History Variables*, while the optional ones are namelist options *Table 11: Optional Grid History Variables*.

Table 8.1: Table 10: Required Grid History Variables

Field	Description	Units
`time`	model time	days
`time_bounds`	boundaries for time-averaging interval	days
`TLON`	T grid center longitude	degrees
`TLAT`	T grid center latitude	degrees
`ULON`	U grid center longitude	degrees
`ULAT`	U grid center latitude	degrees
`NCAT`	category maximum thickness	m
`VGRIDi`	vertical ice levels	
`VGRIDs`	vertical snow levels	

Table 8.2: Table 11: Optional Grid History Variables

Field	Description	Units
`f_tmask`	ocean grid mask (0=land, 1=ocean)	
`f_blkmask`	ice block mask	
`f_tarea`	T grid cell area	m^2
`f_uarea`	U grid cell area	m^2
`f_dxt`	T cell width through middle	m
`f_dyt`	T cell height through middle	m
`f_dxu`	U cell width through middle	m
`f_dyu`	U cell height through middle	m
`f_HTN`	T cell width North side	m
`f_HTE`	T cell width East side	m
`f_ANGLET`	angle grid makes with latitude line on T grid	radians
`f_ANGLE`	angle grid makes with latitude line on U grid	radians
`f_bounds`	corner points of grid cells	degrees

8.1 Caveats Regarding Averaged Fields

In computing the monthly averages for output to the history files, most arrays are zeroed out before being filled with data. These zeros are included in the monthly averages where there is no ice. For some fileds, this is not a problem, for example, ice thickness and ice area. For other fields, this will result in values that are not representative of the field when ice is present. Some of the fields affected are:

- Flat, Fsens - latent and sensible heat fluxes
- evap - evaporative water flux
- Fhocn - ice/ocn net heat flux
- Fswabs - snow/ice/ocn absorbed solar flux
- strairx, strairy - zonal and meridional atm/ice stress
- strcorx, strcory - zonal and meridional coriolis stress

For some fields, a non-zero value is set where there is no ice. For example, Tsfc has the freezing point averaged in, and Flwup has σT_f^4 averaged in. At lower latitudes, these values can be erroneous.

To aid in the interpretation of the fields, a field called `ice_present` is written to the history file. It contains information on the fraction of the time-averaging interval when any ice was present in the grid cell during the time-averaging interval in the history file. This will give an idea of how many zeros were included in the average.

The second caveat results from the coupler multiplying fluxes it receives from the ice model by the ice area. Before sending fluxes to the coupler, they are divided by the ice area in the ice model. These are the fluxes that are written to the history files, they are not what affects the ice, ocean or atmosphere, nor are they useful for calculating budgets. The division by the ice area also creates large values of the fluxes at the ice edge. The affected fields are:

- Flat, Fsens - latent and sensible heat fluxes
- Flwup - outgoing longwave
- evap - evaporative water flux
- Fresh - ice/ocn fresh water flux
- Fhnet - ice/ocn net heat flux
- Fswabs - snow/ice/ocn absorbed solar flux

When applicable, two of the above fields will be written to the history file: the value of the field that is sent to the coupler (divided by ice area) and a value of the flux that has been multiplied by ice area (what affects the ice). Fluxes multiplied by ice area will have the suffix `_aice` appended to the variable names in the history files. Fluxes sent to the coupler will have “sent to coupler” appended to the `long_name`. Fields of rainfall and snowfall multiplied by ice area are written to the history file, since the values are valid everywhere and represent the precipitation rate on the ice cover.

8.2 Changing Frequency and Averaging

The frequency at which data are written to a history file as well as the interval over which the time average is to be performed is controlled by the namelist variable `histfreq`. Data averaging is invoked by the namelist variable `hist_avg`. The averages are constructed by accumulating the running sums of all variables in memory at each timestep. The options for both of these variables are described in *Table 1: Setup Namelist Options*. If `hist_avg` is true, and `histfreq` is set to monthly, for example, monthly averaged data is written out on the last day of the month.

8.3 Changing Content

The second namelist in the setup script controls what variables are written to the history file. To remove a field from this list, add the name of the character variable associated with that field to the **$CASE/user_nl_cice** file and assign it a value of ’xxxxx’. For example, to remove ice thickness and snow cover from the history file, add

```
&icefields_nml
    f_hi    =  'xxxxx'
  , f_hs    =  'xxxxx'
/
```

to the namelist. An incomplete list of history variables is available in *Table 12: History Variables*. Note that there is a new flag `f_CMIP` that will turn on all of the SIMIP variables.

Table 8.3: Table 12: History Variables

Logical Variable	Description	Units
`f_hi`	ice volume per unit area	m
`f_hs`	snow volume per unit area	m
`f_snowfrac`	snow fraction	1
`f_Tsfc`	snow/ice surface temperature	C
`f_aice`	ice concentration (aggregate)	1
`f_uvel`	x component ice velocity	m s^{-1}
`f_vvel`	y component ice velocity	m s^{-1}
`f_uatm`	x component wind velocity	m s^{-1}
`f_vatm`	y component wind velocity	m s^{-1}
`f_sice`	bulk ice salinity	ppt
`f_fswdn`	downwelling solar flux	W m^{-2}
`f_fswup`	upward reflected solar flux	W m^{-2}
`f_flwdn`	downwelling longwave flux	W m^{-2}
`f_snow`	snow fall rate received from coupler	cm day^{-1}
`f_snow_ai`	snow fall rate on ice cover	cm day^{-1}
`f_rain`	rain fall rate received from coupler	cm day^{-1}
`f_rain_ai`	rain fall rate on ice cover	cm day^{-1}
`f_sst`	sea surface temperature	C

Continued on next page

Table 8.3 – continued from previous page

Logical Variable	Description	Units
f_sss	sea surface salinity	g kg^{-1}
f_uocn	x component ocean current	m s^{-1}
f_vocn	y component ocean current	m s^{-1}
f_frzmlt	freeze/melt potential	W m^{-2}
f_fswabs	total absorbed solar flux sent to coupler	W m^{-2}
f_fswabs_ai	total absorbed solar flux in snow/ocn/ice	W m^{-2}
f_fswint_ai	internal absorbed solar flux in snow/ice	W m^{-2}
f_fswfac	shortwave scaling factor	1
f_coszen	cosine of the zenith angle	radians
f_albsni	snow ice broadband albedo	%
f_alvdr	visible direct albedo sent to coupler	%
f_alidr	near-infrared direct albedo sent to coupler	%
f_alvdf	visible diffuse albedo sent to coupler	%
f_alidf	near-infrared diffuse albedo sent to coupler	%
f_alvdr_ai	visible direct albedo	%
f_alidr_ai	near-infrared direct albedo	%
f_alvdf_ai	visible diffuse albedo	%
f_alidf_ai	near-infrared diffuse albedo	%
f_albsni	snow ice broadband albedo	%
f_albsno	snow broadband albedo	%
f_albpnd	pond broadband albedo	%
f_albice	bare ice broadband albedo	%
f_flat	latent heat flux sent to coupler	W m^{-2}
f_flat_ai	ice/atm latent heat flux	W m^{-2}
f_fsens	sensible heat flux sent to coupler	W m^{-2}
f_fsens_ai	ice/atm sensible heat flux	W m^{-2}
f_flwup	outgoing longwave flux sent to coupler	W m^{-2}
f_flwup_ai	ice/atm outgoing longwave flux	W m^{-2}
f_evap	evaporative water flux sent to coupler	cm day^{-1}
f_evap_ai	ice/atm evaporative water flux	cm day^{-1}
f_Tair	air temperature	C
f_Tref	2 m reference temperature	C
f_Qref	2 m reference specific humidity	g/kg
f_congel	basal ice growth	cm day^{-1}
f_frazil	frazil ice growth	cm day^{-1}
f_snoice	snow-ice formation	cm day^{-1}
f_meltb	basal ice melt	cm day^{-1}
f_melts	surface snow melt	cm day^{-1}
f_meltt	surface ice melt	cm day^{-1}
f_meltl	lateral ice melt	cm day^{-1}
f_fresh	ice/ocn fresh water flux sent to coupler	cm day^{-1}
f_fresh_ai	ice/ocn fresh water flux	cm day^{-1}
f_fsalt	ice to ocn salt flux sent to coupler	kg m^{-2} day^{-1}
f_fsalt_ai	ice to ocn salt flux	kg m^{-2} day^{-1}
f_fhocn	ice/ocn net heat flux sent to coupler	W m^{-2}
f_fhocn_ai	ice/ocn net heat flux	W m^{-2}
f_fswthru	SW transmitted through ice to ocean sent to coupler	W m^{-2}
f_fswthru_ai	SW transmitted through ice to ocean	W m^{-2}
f_strairx	zonal atm/ice stress	N m^{-2}

Continued on next page

Table 8.3 – continued from previous page

Logical Variable	Description	Units
f_strairy	meridional atm/ice stress	N m^{-2}
f_strtltx	zonal sea surface tilt	m m^{-1}
f_strtlty	meridional sea surface tilt	m m^{-1}
f_strcorx	zonal coriolis stress	N m^{-2}
f_strcory	meridional coriolis stress	N m^{-2}
f_strocnx	zonal ocean/ice stress	N m^{-2}
f_strocny	meridional ocean/ice stress	N m^{-2}
f_strintx	zonal internal ice stress	N m^{-2}
f_strinty	meridional internal ice stress	N m^{-2}
f_strength	compressive ice strength	N m^{-1}
f_divu	velocity divergence	% day^{-1}
f_shear	strain rate	% day^{-1}
f_opening	lead opening rate	% day^{-1}
f_sig1	normalized principal stress component	
f_sig2	normalized principal stress component	
f_daidtt	area tendency due to thermodynamics	% day^{-1}
f_daidtd	area tendency due to dynamics	% day^{-1}
f_dvidtt	ice volume tendency due to thermo.	cm day^{-1}
f_dvidtd	ice volume tendency due to dynamics	cm day^{-1}
f_mlt_onset	melt onset date	
f_frz_onset	freeze onset date	
f_icepresent	fraction of time with ice present in grid cell	
f_aicen	ice concentration (category)	1
f_vicen	ice volume (category)	m
f_vsnon	snow volume (category)	m

CHAPTER

NINE

CICE RESTART FILES

Restart files contain all of the initial condition information necessary to restart from a previous simulation. These files are in a standard netCDF 64-bit binary format. A restart file is not necessary for an initial run, but is highly recommended. The initial conditions that are internal to the ice model produce an unrealistic ice cover that an uncoupled ice model will correct in several years. The initial conditions from a restart file are created from an equilibrium solution, and provide more realistic information that is necessary if coupling to an active ocean model. The frequency at which restart files are created is controlled by the namelist parameter `dumpfreq`. The names of these files are proceeded by the namelist parameter `dump_file` and, by default are written out yearly to the executable directory. To change the directory where these files are located, modify the variable $RSTDIR at the top of the setup script. The names of the restart files follow the CESM Output Filename Requirements. The form of the restart file names are as follows:

```
**$CASE.cice.r.yyyy-mm-dd-sssss.nc**
```

For example, the file **$CASE.cice.r.0002-01-01-00000.nc** would be written out at the end of year 1, month 12. A file containing the name of a restart file is called a restart pointer file. This filename information allows the model simulation to continue from the correct point in time, and hence the correct restart file.

Changing the restart frequency is handled by the CESM driver in **env_run.xml**. The variables are `REST_DATE`, `REST_N` and `REST_OPTION`. See the CESM documentation here:

http://www.cesm.ucar.edu/models/cesm2/component_settings/drv_input.html

9.1 Restart Pointer Files

A pointer file is an ascii file named **rpointer.ice** that contains the path and filename of the latest restart file. The model uses this information to find a restart file from which initialization data is read. The pointer files are written to and then read from the executable directory. For startup runs, a pointer is created by the ice setup script Whenever a restart file is written, the existing restart pointer file is overwritten. The namelist variable `pointer_file` contains the name of the pointer file. Pointer files seldom need editing. The contents are usually maintained by the setup script and the component model.

CHAPTER

TEN

STDOUT OUTPUT

Diagnostics from the ice model are written to an ASCII file that contains information from the compilation, a record of the input parameters, and how hemispherically averaged, maximum and minimum values are evolving with the integration. Certain error conditions detected within the ice setup script or the ice model will also appear in this file. Upon the completion of the simulation, some timing information will appear at the bottom of the file. The file name is of the form **ice.log.$LID**, where `$LID` is a timestamp for the file ID. It resides in the executable directory. The frequency of the diagnostics is determined by the namelist parameter `diagfreq`. Other diagnostic messages appear in the **cesm.log.$LID** or **cpl.log.$LID** files in the executable directory. See the CIME scripts documentation.

CHAPTER

ELEVEN

TROUBLESHOOTING

11.1 Code does not Compile or Run

Check the **ice.log.*** or **ice.bldlog.*** files in the executable directory, or the standard output and error files for information. Also, try the following:

- Delete the executable directory and rebuild the model.
- Make sure that there is a **Macros.< OS >** file for your platform. Modify the directory paths for the libraries.
- Make sure all paths and file names are set correctly in the scripts.
- If changes were made to the **ice_domain_size.F90** file in the source code directory, they will be overwritten by the file in **input_templates**.

11.2 Departure points out of bounds.

This error is written from **ice_transport_remap.F90** when the ice speed is causing parcels of ice to go beyond a grid cell. This is akin to a CFL violation. Generally changing the timestep in **env_run.xml** with `ATM_NCPL` will allow the model to proceed. Note this can only be done for hybrid or startup runs. One can try just adjusting the dynamic timestep as described in the next section.

11.3 Negative Ice Area in Horizontal Remapping

This error is written from **ice_transport_remap.F90** when the ice model is checking for negative ice areas. If it happens well into a model integration, it can be indicative of a CFL violation. The output looks like:

```
60: New area < 0, istep = 119588
60: (my_task,i,j,n) = 4 21 380 1
60: Old area = 0.960675000975677174E-05
60: New area = -0.161808948357841311E-06
60: Net flux = -0.976855895811461324E-05
60:(shr_sys_abort) ERROR: remap transport: negative area
60:(shr_sys_abort) WARNING: calling shr_mpi_abort() and stopping
60:(shr_mpi_abort):remap transport: negative area 0
```

The dynamics timestep should be reduced to integrate past this problem. In **user_nl_cice** set

```
ndtd = 2
```

and restart the model. When the job completes set the value back to 1.

11.4 Picard convergence error

This is an error from the mushy layer thermodynamics `ktherm = 2`. One can try changing `nit_max` in the **ice_therm_mushy.F90** code, but this does not often help. Most likely this is an indication of a problem in the forcing. Sometimes reducing the overall timestep may help.

11.5 Tsn init problems

Sometimes the surface temperature or snow temperature at the beginning of the thermodynamic iteration may become unrealistic. The lower bound on this error is -100C. This either indicates a problem with the CICE initial file or the forcing. Changing the timestep will not help.

11.6 Thermodynamic Iteration Error

This error is written from **ice_therm_vertical.F90** when the ice model temperature iteration is not converging in the thermodynamics. This is usually a problem with the forcing from the atmosphere or ocean, but sometimes can be indicative of a timestep problem in the ice. Check the forcing files at point i,j first.

```
Thermo iteration does not converge
istep1, my_task, i, j:
```

11.7 Conservation Error

This error is written from **ice_itd.F** when the ice model is checking that initial and final values of a conserved field are equal to within a small value. The output looks like:

```
Conservation error: vice, add_new_ice
11  :  14  185
Initial value =  1362442.600400560
Final value =  1362442.600400561
Difference =  2.328306436538696D-10
(shr_sys_abort) ERROR: ice: Conservation error
(shr_sys_abort) WARNING: calling shr_mpi_abort() and stopping
(shr_mpi_abort):ice: Conservation error  0
```

Non-conservation can occur if the ice model is receiving very bad forcing, and is not able to deal with it. This has occurred after a CFL violation in the ocean. The timestep in the ocean may be decreased to get around the problem.

11.8 NX does not divide evenly into grid

If you modify the number of tasks used by the ice model, the model may stop with this error written to the log file:

```
'ERROR: NX must divide evenly into grid,100,8'
```

The number of MPI processors used by the ice model must divide evenly into the grid dimensions. For example, running the ice model with 8 tasks on the gx3v7 grid will result in an error, since 8 does not divide evenly into the 100 longitude points. To fix this error, change the value of $NTASKS for the uncoupled ice model in the main script. In this case, a value of 4 would work, and the task geometry would also have to be changed.

11.9 Enabling the Debugger

This section explains how to set some compiler options for debugging. For the coupled model, set DEBUG to TRUE in the **env_build.xml** script. Before running the model, be sure to delete the object files or a clean build so that the source code will be recompiled. If a core file is created, it will be in the executable directory. Use some debugging tools for your platform to look at the core file. Useful information may also appear in the standard error and output files.

CHAPTER

TWELVE

REFERENCES

References

CHAPTER THIRTEEN

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

[1] A. Assur. 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.

[2] C. M. Bitz, M.M. Holland, M. Eby, and A. J. Weaver. Simulating the ice-thickness distribution in a coupled climate model. *J. Geophys. Res.*, 106:2441–2463, 2001.

[3] C. M. Bitz and W. H. Lipscomb. An energy-conserving thermodynamic model of sea ice. *J. Geophys. Res.*, 104:15,669–15,677, 1999.

[4] E. C. Hunke and J. K. Dukowicz. An elastic-viscous-plastic model for sea ice dynamics. *J. Phys. Oceanogr.*, 27:1849–1867, 1997.

[5] E. C. Hunke, 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.

[6] E. C. Hunke, W. H. Lipscomb, A. K. Turner, N. Jeffery, and Scott Elliott. *CICE: The Los Alamos Sea Ice Model. Documentation and Software User's Manual. Version 5.1*. T-3 Fluid Dynamics Group, Los Alamos National Laboratory, Tech. Rep. LA-CC-06-012, 2015.

[7] W. H. Lipscomb, E. C. Hunke, W. Maslowski, and J. Jakacki. Ridging, strength, and stability in high-resolution sea ice models. *J. Geophys. Res. Oceans*, 112(C03S91):18pp, 2007. URL: http://dx.doi.org/10.1029/2005JC003355.

[8] Drew A. Rothrock. The energetics of the plastic deformation of pack ice by ridging. *J. Geophys. Res.*, 80:4514–4519, 1975.

[9] Alan S. Thorndike, Drew. S. Rothrock, Gary A. Maykut, and Roger Colony. The thickness distribution of sea ice. *J. Geophys. Res.*, 80:4501–4513, 1975.

[10] A. K. Turner and E. C. Hunke. Impacts of a mushylayer thermodynamic approach in global seaice simulations using the CICE seaice model. *J. Geophys. Res. Oceans*, 120:1253–1275, 2015. URL: http://dx.doi.org/10.1002/2014JC010358.