Community Climate System Model
National Center for Atmospheric Research, Boulder, CO
http://www.ccsm.ucar.edu/models

Community Ice CodE (CICE) User's Guide
Version 4.0

Released with CCSM4.0

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

Contents

1 Introduction

This User's Guide accompanies the CCSM4 User's Guide, and is intended for those who would like to run CICE coupled, on a supported platform, and "out of the box". Users running CICE fully coupled should first look at the CCSM4 User's Guide. It includes a quick start guide for downloading the CCSM4 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 CCSM4 User's Guide. The CICE User's Guide is intended for users interested in making modifications to the ice model scripts or namelists or running the uncoupled ice model. Users interested in modifying the source code should see the CICE Code Reference/ Developer's Guide.

CICE4 is the latest version of the Los Alamos Sea Ice Model, sometimes referred to as the Community Ice CodE. 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 stand-alone ice model. It has been released as the sea ice component of the Community Climate System Model (CCSM), a fully-coupled global climate model that provides simulations of the earths past, present and future climate states. CICE4 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 CICE is available separately from Los Alamos National Laboratory:

`http://oceans11.lanl.gov/trac/CICE`.

It provides a means of running the sea ice model independent of the other CCSM components. It reads 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.

The physics in the uncoupled ice model are identical to those in the ice model used in the fully coupled system. CICE is a dynamic-thermodynamic model that includes a subgrid-scale ice thickness distribution (Bitz et al. (2001); Lipscomb (2001)). It uses the energy conserving thermodynamics of Bitz and Lipscomb (1999), 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 Hunke and Dukowicz (1997). Sea ice ridging follows Rothrock (1975) and Thorndike et al. (1975). A slab ocean mixed layer model is included. A Scientific Reference is available that contains more detailed information on the model physics.

An attempt has been made throughout this document to provide the following text convention. Variable names used in the code are `typewritten`. Subroutine names are given in *italic*, and file names are in **boldface**.

1.1 What's new in CICE4?

CICE4 is an upgraded version of the Community Sea Ice Model, CSIM5, which was based on CICE3, and was released in June 2004. The model physics are similar to that of CSIM5, but it was decided to move to CICE, the LANL sea ice model for practical reasons. The major changes are:

- The incremental remapping transport scheme is now the default and is available in the modules called **ice_transport_driver.F90** and **ice_transport_remap.F90**. The MPDATA transport scheme, is no longer supported in CICE4. The upwind advection scheme is the only additional option and is contained in **ice_transport_driver.F90**.
- The standalone ice model is now only available through Los Alamos National Laboratory.
- Several physics options have been shifted around into other or new modules. For example, most of **ice_albedo.F90** is now in **ice_shortwave.F90**. The new module contains all of the shortwave radiative transfer plus the basic albedo calculations.
- The mechanical redistribution scheme has been changed significantly and is available in **ice_mechred.F90**.

- A new drivers area has been created for modules that are specific to the CCSM as opposed to the standalone CICE model. The new CCSM drivers are contained in the cpl_mct and cpl_share subdirectories. The ESMF driver (cpl_esmf) is still under development. The source subdirectory now contains driver independent source code for the most part.
- A new bld subdirectory has been introduced which contains CCSM specific build and configure scripts. These scripts handle the namelist generation, defaults, and configuration details.

The CICE source code is based on the Los Alamos sea ice model CICE model version 4. The main source code is very similar in both versions, but the drivers are significantly different. If there are some topics that are not covered in the CICE documentation, users are encouraged to look at the CICE documentation Hunke and Lipscomb (2008). It is available at Los Alamos National Laboratory at:

`http://oceans11.lanl.gov/trac/CICE`.

2 The CICE Scripts

The setup scripts for the coupled model are located in **ccsm4/scripts**. The directory structure of CICE4 within CCSM is shown below.

```
                            ccsm4         (main directory)
                              |
                              |
                  models--------+--------- scripts
                     |                        |
                     |              * * * * *|* * * * *
            bld------+------ice      *build scripts for*
             |               |       *  coupled model  *
         (Makefile           |       * * * * * * * * * *
          macros)            |
                           cice         (active ice component)
                             |
 bld ---------- docs -------+------- src
                  |                   |
                (CICE                 |
            documentation)            |
                                      |
                                      |
                   drivers --- mpi ---+--- serial --- source
                      |
                      |
                      |
cice4 ---- cpl_esmf --+-- cpl_mct ---- cpl_share
```

2.1 Coupled Model Scripts

The CCSM4 scripts have been significantly upgraded from CCSM3 and are based on a completely different design philosophy. The new scripts will 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 previously in the **/scripts** directory for CCSM3 are now generated automatically. See the CCSM4 User's Guide for information on how to use the new scripts. The file that contains the ice model namelist is now located in **$CASE/Buildconf**. The script containing the environment variables used for building the executable file for the ice model is also in **$CASE/Buildconf**. The contents of the ice model namelist are described in section 3.

2.2 The Build Environment

The build and configure environment has changed significantly from previous versions of CCSM. See the CCSM documentation.

2.2.1 CICE Preprocessor Flags

Preprocessor flags are activated in the form `-Doption` in the **cice.buildexe.csh** script. The flags specific to the ice model are

```
CPPDEFS :=  $(CPPDEFS) -DCCSMCOUPLED -Dcoupled -Dncdf -DNCAT=5 -DNXGLOB=$()
-DNYGLOB=$() -DNTR_AERO=3 -DBLCKX=$() -DBLCKY=$() -DMXBLCKS=$()
```

The options `-DCCSMCOUPLED` 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 CCSM coupler multiplies the fluxes by the ice area, so they are divided by the ice area in CICE to get the correct fluxes.

The options `-DBLCKX=$(NX)` and `-DBLCKY=$(NY)` set the number of processors used in each grid direction. These values are set automatically in the scripts for the coupled model. Note that `NX` and `NY` must divide evenly into the grid, and are used only for MPI grid decomposition. If `NX` or `NY` do not divide evenly into the grid, the model setup will exit from the setup script and print an error message to the **ice.log*** (standard out) file.

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.

The flat `-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 flag `-D_MPI` sets up the message passing interface. This must be set for runs using a parallel environment. To get a better idea of what code is included or excluded at compile time, grep for `ifdef` and `ifndef` in the source code or look at the ***.f90** files in the **/obj** directory.

3 Namelist Variables

CICE uses the same namelists for both the coupled and uncoupled models. This section describes the namelist variables in the namelist `ice_nml`, which determine time management, output frequency, model physics, and filenames The ice namelists for the coupled model are now located in **$CASE/Buildconf**.

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 and aerosol options along with the history namelist variables are set in **ice_prescribed.F90**, **ice_prescaero.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 Tables 1-8, 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 3.9.1.

The main run management namelist options are shown in Table 1. While additional namelist variables are available in the uncoupled version, they are set by the driver in CCSM. Variables set by the driver include: `dt`, `runid`, `runtype`, `istep0`, `days_per_year`, `restart` and `dumpfreq`. These should be changed in the CCSM configuration files:

`CCSM scripts (http://www.ccsm.ucar.edu/models/ccsm4.0/ccsm).`

Table 1: Namelist Variables for Run Management

Varible	Type	Default Value	Description
`ice_ic`	character	default	Filename for initial and branch runs 'default' uses default initialization 'none' initializes with no ice
`xndt_dyn`	Integer	1	Times to loop through (sub-cycle) ice dynamics
`diagfreq`	Integer	24	Frequency of diagnostics written (min, max, hemispheric sums) to standard output 24 => writes once every 24 timesteps 1 => diagnostics written each timestep 0 => no diagnostics written
`histfreq`	Character Array	'm','x','x','x','x'	Frequency of output written to history streams 'D' or 'd' writes daily data 'W' or 'w' writes weekly data '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 history data is written to each stream
`hist_avg`	Logical	.true.	If true, averaged history information is written out at a frequency determined by histfreq. If false, instantaneous values rather than time-averages are written.
`pointer_file`	Character	'rpointer.ice'	Pointer file that contains the name of the restart file.
`lcdf64`	Logical	.false.	Use 64-bit offset in netcdf files

Table 2: Maximum values for ice model timestep `dt`

Grid	$min(\Delta x, \Delta y)$	$max\Delta t$
gx3v5	28845.9 m	4.0 hr
gx1v3	8558.2 m	1.2 hr

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:

$$\Delta t < \frac{min(\Delta x, \Delta y)}{4max(u, v)}. \quad (1)$$

Maximum values for `dt` for the two standard CCSM POP grids, assuming $max(u, v) = 0.5m/s$, are shown in Table 2. The default timestep for CICE is 30 minutes, which must be equal to the coupling interval set in the CCSM configuration files.

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 (`xndt_dyn`) 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. By default, diagnostics are written out once every 48 timesteps to the ascii file **ice.log.$LID** (see section 7.1). `$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 7.3. 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 7.3.

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 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 7.2.

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 CCSM driver. The default format for restart files is now netCDF, but this can be changed to binary through the namelist variable, `restart_format`.

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.

`incond_dir`, `restart_dir` and `history_dir` are the directories where the initial condition file, the restart files and the history files will be written, respectively. These values are set at the top of the setup script and have been modified from the default values to meet the requirements of the CCSM filenaming convention. This allows each type of output file to be written to a separate directory. If the default values are used, all of the output files will be written to the executable directory.

`incond_file`, `dump_file` and `history_file` are the root filenames for the initial condition file, the restart files and the history files, respectively. These strings have been determined by the requirements of the CCSM filenaming convention, so the default values are set by the CCSM driver. See 7.2 and 7.3 for an explanation of how the rest of the filename is created.

3.3 Model Physics

The namelist variables for the ice model physics are listed in Table 3. `restart` is almost always true since most run types begin by reading in a binary restart file. See section 5 for a description of the run types and about using restart files and internally generated model data as initial conditions. `kcolumn` is a flag that will run the model as a single column if is set to 1. This option has not been thoroughly tested and is not supported.

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`.

$$dte < T < dt \tag{2}$$

This relationship is discussed in Hunke (2001); also see Hunke and Lipscomb (2008), section 4.4. The best ratio for [dte : T : dt] is [1 : 40 : 120]. Typical combinations of `dt` and `ndte` are (3600., 120), (7200., 240) (10800., 120). The default ndte is 120 as set in **ice_init.F90**.

Table 3: Namelist Variables for Model Physics

Varible Name	Type	Default Value	Description
`ndte`	Integer	1	Number of sub-cycles in EVP dynamics.
`kcolumn`	Integer	0	Column model flag. 0 = off 1 = column model (not tested or supported)
`kitd`	Integer	1	Determines ITD conversion 0 = delta scheme 1 = linear remapping
`kdyn`	Integer	1	Determines ice dynamics 0 = No ice dynamics 1 = Elastic viscous plastic dynamics
`kstrength`	Integer	1	Determines pressure formulation 0 = Hibler (1979) parameterization 1 = Rothrock (1975) parameterization
`evp_damping`	Logical	.false.	If true, use damping procedure in evp dynamics (not supported).
`advection`	Character	'remap'	Determines horizontal advection scheme. 'remap' = incremental remapping 'upwind' = first order advection
`shortwave`	Character	'dEdd'	Shortwave Radiative Transfer Scheme 'default' = CCSM3 Shortwave 'dEdd' = delta-Eddington Shortwave
`albicev`	Double	0.73	Visible ice albedo (CCSM3)
`albicei`	Double	0.33	Near-infrared ice albedo (CCSM3)
`albsnowv`	Double	0.96	Visible snow albedo (CCSM3)
`albsnowi`	Double	0.68	Near-infrared snow albedo (CCSM3)
`R_ice`	Double	0.0	Base ice grain radius tuning parameter (dEdd)
`R_pnd`	Double	1.5	Base snow grain radius tuning parameter (dEdd)
`R_snw`	Double	0.0	Base pond grain radius tuning parameter (dEdd)
`dT_mlt_in`	Double	1.5	Snow melt onset temperature parameter (dEdd)
`rsnw_mlt_in`	Double	1500.0	Snow melt maximum radius (dEdd)

Table 4: Namelist Variables for Tracers

Varible	Type	Default Value	Description
`tr_iage`	Logical	.true.	Ice age passive tracer
`tr_FY`	Logical	.true.	First-year ice area passive tracer
`tr_lvl`	Logical	.false.	Level ice area passive tracer
`tr_pond`	Logical	.true.	Melt pond physics and tracer
`tr_aero`	Logical	.true.	Aerosol physics and tracer

`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 (Lipscomb (2001)). The delta function method represents *g(h)* in each category as a delta function (Bitz et al. (2001)). 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 Hunke and Dukowicz (1997). If `kdyn` is set to o 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 Hibler (1979) calculation depends on mean ice thickness and open water fraction. The calculation of Rothrock (1975) is based on energetics and should not be used if the ice that participates in ridging is not well resolved.

`evp_damping` is used to control the damping of elastic waves in the ice dynamics. It is typically set to `.true`. for high-resolution simulations where the elastic waves are not sufficiently damped out in a small timestep without a significant amount of subcycling. This procedure works by reducing the effective ice strength that's used by the dynamics and is not a supported option.

`advection` determines the horizontal transport scheme used. The default scheme is the incremental remapping method (Lipscomb and Hunke (2004)). This method is less diffusive and is computationally efficient for large numbers of categories or tracers. The upwind scheme is also available. The upwind scheme is only first order accurate.

The base values of the snow and ice albedos for the CCSM3 shortwave option are set in the namelist. The ice albedos are those for ice thicker than `ahmax`, which is currently set at 0.5 m. This thickness is a parameter that can be changed in **ice_shortwave.F90**. The snow albedos are for cold snow.

For the new delta-Eddington shortwave radiative transfer scheme Briegleb and Light (2007), the base albedos are computed based on the inherent optical properties of snow, sea ice, and melt ponds. These albedos are tunable through adjustments to the snow grain radius, `R_snw`, temperature to transition to melting snow, and maximum snow grain radius.

3.4 Tracer Namelist

The namelist parameters listed in Table 4 are for adding tracers. See section on tracers.

3.5 Prescribed Ice Namelist

The namelist parameters listed in Table 5 are for the prescribed ice option as used in AMIP and F compset (standalone CAM) runs.

3.6 Prescribed Aerosol Namelist

The namelist parameters listed in Table 6 are for the prescribed aerosol option. See section on the aerosol tracer.

Table 5: Namelist Variables for Prescribed Ice Option

Varible	Type	Default Value	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		Prescribed ice stream data file
`stream_fldfilename`	Character		Prescribed ice stream data file
`stream_fldvarname`	Character	ice_cov	Ice fraction field name

Table 6: Namelist Variables for Prescribed Aerosol Option

Varible	Type	Default Value	Description
`prescribed_aero`	Logical	.false.	Flag to turn on prescribed ice
`prescribed_aero_fill`	Logical	.false.	Flag to turn fill option
`stream_year_first_aero`	Integer	1	First year of aerosol deposition data
`stream_year_last_aero`	Integer	1	Last year of aerosol deposition data
`model_year_align_aero`	Integer	1	Year in model run that aligns with stream_year_first
`stream_domfilename_aero`	Character		Prescribed aerosol stream data file
`stream_fldfilename_aero`	Character		Prescribed aerosol stream data file
`stream_fldvarname_aero`	Character		Aerosol field names

3.7 Grid Namelist

The namelist parameters listed in Table 7 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 7.

For coupled runs, supported grids include the `'displaced_pole'` grids (gx3v7 and gx1v6) and the `'tripole'` grids.

3.8 Domain Namelist

The namelist parameters listed in Table 8 are for computational domain decomposition information. These are generally set in the build configure scripts based on the number of processors. See the CCSM scripts documentation.

3.9 Example Namelists

This section shows several examples of namelists from the coupled ice model. These examples are taken directly from **cice.buildnml.csh** 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

Table 7: Namelist Variables for Grid and Mask Information

Varible	Type	Default Value	Description
`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.
`kmt_file`	Character	'data.domain.kmt'	Input filename containing land mask information.
`kcatbound`	Integer	0	How category boundaries are set (0 or 1)

Table 8: Namelist Variables for Domain Decomposition Information

Varible	Type	Default Value	Description
`processor_shape`	Character	'square-pop'	Approximate block shapes
`ew_boundary_type`	Character	'cyclic'	Boundary conditions in E-W direction
`ns_boundary_type`	Character	'open'	Boundary conditions in N-S direction
`distribution_type`	Character	'cartesian'	How blocks are split onto processors 'cartesian' 'spacecurve' 'rake'
`distribution_wght`	Character	'erfc'	How blocks are weighted when using space-filling curves (erfc or file)
`distribution_wght_file`	Character	''	File containing space-filling curve weights when not using erfc weighting

"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`.

3.9.1 Example 1: CCSM Fully Coupled

The following example is the namelist used for CCSM fully coupled, or the B configuration. The variables that are still set to shell script variables have been set at the top of **cice.buildnml.csh** or in other scripts. A completely resolved version of the namelist will be written to **ice_in** in the executable directory.

```
&setup_nml
 diagfreq              =  24
 hist_avg              = .true.
 histfreq              = 'm','x','x','x','x'
 histfreq_n            = 1,1,1,1,1
 ice_ic        = 'b40.1850.track1.1deg.006.cice.r.0301-01-01-00000.nc'
 lcdf64        = .false.
 pointer_file          = 'rpointer.ice'
 xndt_dyn              =  1.0
/
&grid_nml
```

```
 grid_file              = '/fis/cgd/cseg/csm/inputdata/ice/cice/global_gx1v6_200
10402.grid'
 grid_format            = 'bin'
 grid_type              = 'displaced_pole'
 kcatbound              =  0
 kmt_file               = '/fis/cgd/cseg/csm/inputdata/ice/cice/global_gx1v6_200
90204.kmt'
/
&ice_nml
 advection              = 'remap'
 albedo_type            = 'default'
 albicei                = 0.45
 albicev                = 0.75
 albsnowi               = 0.73
 albsnowv               = 0.98
 evp_damping            = .false.
 kdyn           =  1
 kitd           =  1
 krdg_partic            = 1
 krdg_redist            = 1
 kstrength              =  1
 ndte           =  120
 r_snw          =  1.5
 shortwave              = 'dEdd'
/
&tracer_nml
/
&domain_nml
 distribution_type              = 'cartesian'
 ew_boundary_type               = 'cyclic'
 ns_boundary_type               = 'open'
 processor_shape                = 'square-pop'
/
&ice_prescribed_nml
 prescribed_ice         = .false.
/
&ice_prescaero_nml
 model_year_align_aero          =  1
 prescribed_aero_fill           = .true.
 stream_domareaname_aero                = "area"
 stream_domfilename_aero                = '/fis/cgd/cseg/csm/inputdata/ice/cice/
aerosoldep_monthly_1850_mean_1.9x2.5_c090421.nc'
 stream_dommaskname_aero                = "mask"
 stream_domtvarname_aero                = "time"
 stream_domxvarname_aero                = "lon"
 stream_domyvarname_aero                = "lat"
 stream_fldfilename_aero                = '/fis/cgd/cseg/csm/inputdata/ice/cice/
aerosoldep_monthly_1850_mean_1.9x2.5_c090421.nc'
 stream_fldvarname_aero         = 'BCPHODRY:BCDEPWET:BCPHIDRY:DSTX01WD:DSTX01DD:
DSTX02WD:DSTX02DD:DSTX03WD:DSTX03DD:DSTX04WD:DSTX04DD'
 stream_year_first_aero         =  1
 stream_year_last_aero          =  1
/
```

3.9.2 Example 2: History File Namelist

The second namelist controls what variables are written to the history file. By default, all files are written to the history file. Variables that are not output are set in the namelist `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. The ice areas and volumes for categories six through ten are not used, since the default thickness distribution consists of five ice categories.

```
f_aero          = 'mxxxx'
f_aicen                 = 'mxxxx'
f_aisnap                = 'mdxxx'
f_apondn                = 'mxxxx'
f_congel                = 'mxxxx'
f_daidtd                = 'mxxxx'
f_daidtt                = 'mxxxx'
f_divu          = 'mxxxx'
f_dvidtd                = 'mxxxx'
f_dvidtt                = 'mxxxx'
f_faero_atm             = 'mxxxx'
f_faero_ocn             = 'mxxxx'
f_fhocn                 = 'mxxxx'
f_fhocn_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_fy            = 'mdxxx'
f_hisnap                = 'mdxxx'
f_icepresent            = 'mxxxx'
f_meltb                 = 'mxxxx'
f_meltl                 = 'mxxxx'
f_meltt                 = 'mxxxx'
f_mlt_onset             = 'xxxxx'
f_opening               = 'mxxxx'
f_shear                 = 'mxxxx'
f_sig1          = 'mxxxx'
f_sig2          = 'mxxxx'
f_snoice                = 'mxxxx'
f_sss           = 'xxxxx'
f_sst           = 'xxxxx'
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               = 'xxxxx'
f_strtlty               = 'xxxxx'
```

```
 f_uocn         = 'xxxxx'
 f_uvel         = 'mxxxx'
 f_vicen                 = 'mxxxx'
 f_vocn         = 'xxxxx'
 f_vvel         = 'mxxxx'
/
```

4 Model Input Datasets

The coupled CICE model requires a minimum of three files to run:

- **global_${ICE_GRID}.grid** is a binary file containing grid information
- **global_${ICE_GRID}.kmt** is a binary file containing land mask information
- **iced.0001-01-01.${ICE_GRID}.20lay** are binary files containing initial condition information for the gx1v6 and gx3v7 grids, respectively. The thickness distribution in this restart file contains 5 categories, each with 4 layers.

Depending on the grid selected in the scripts, the appropriate **global*** and **iced*** files will be used in the executable directory. These files are read directory from the system input data directory and not copied to the executable directory. Currently, only gx3v7, gx1v6, tx1v1, and tx0.1v2 grids are supported for the ice and ocean models. Note that these files can now be used in netCDF format.

5 Run Types

The run types available for the coupled model are described in the CCSM User's Guide.

6 Changing the Number of Ice Thickness Categories

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

- `$NCAT` in the run script
- The source code module **ice_model_size.F90**
- The initial condition (restart) file in the input file directory

The number of ice thickness categories is set in **ccsm3/scripts/$CASE/Buildexe/cice.buildexe.csh** using the variable called `$NCAT`. The default value is 5 categories. `$NCAT` is used to determine the CPP variable setting (NCAT) in **ice_model_size.F90**. `$RES` is the resolution of the grid, 100x116 (gx3v7) and 320x384 (gx1v6) for low and medium resolution grids, respectively.

NOTE: To use one ice thickness category, the following changes will need to be made in the namelist:

```
, 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.

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 `$NCAT` to the desired number of categories in **ccsm3/scripts/$CASE/Buildexe/cice.buildexe.csh**.
- Set the namelist variable `dumpfreq = 'm'` in **ccsm3/scripts/$CASE/Buildnml_Prestage/cice.buildnml.csh** to print out restart files monthly.
- Set the namelist variable `restart = .false.` in **ccsm3/scripts/$CASE/Buildnml_Prestage/cice.buildnml.csh** 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*.
- Copy the file into the input data directory or directly into the the executable directory.

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.

7 Output Data

The ice model produces three types of output data. 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. A series of binary restart files necessary to continue the run are created. A series of netCDF history files containing gridded instantaneous or time-averaged output are also generated during a run. These are described below.

7.1 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 **ccsm.log.$LID** or **cpl.log.$LID** files in the executable directory. See the CCSM scripts documentation.

7.2 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 CCSM 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.

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.

7.3 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 `history_dir` defined in the namelist. The netCDF file names are prepended by the character string given by `history_file` in the `ice_nml` namelist. This character string has been set according to CCSM Output Filename Requirements. If `history_file` is not set in the namelist, the default character string `'iceh'` is used. 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: **$CASE.cice.h?.yyyy-mm-dd.nc**
Instantaneous (`histfreq` = 'y', 'm', or 'd'): **$CASE.cice.h?.yyyy-mm-dd-sssss.nc**
Instantaneous (written every dt, `histfreq` = 1): **$CASE.cice.h?.yyyy-mm-dd-sssss.nc**

`$CASE` is set in the main setup script. Note that the `?` denotes the multiple stream option where the first stream is just `.h.` and subsequent streams are `h1, h2, etc`. All history files are written in the executable directory. Changes to the frequency and averaging will affect all output fields. The best description of the history data comes from the file itself using the netCDF command `ncdump -h filename.nc`. Variables containing grid information are written to every file and are listed in Table 9. In addition to the history files, a netCDF file containing a snapshot of the initial ice state can be created at the start of each run. The file name is **$CASE.cice.i.yyyy-mm-dd-sssss.nc** and is written in the executable directory.

7.3.1 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
- Fhnet - 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 Flwout 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
- Flwout - 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.

7.3.2 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. 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.

7.3.3 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 `&icefields_nml` namelist in **cice.buildnml.csh** 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.

Table 10: Standard Fields Available for Output to History File

Logical Variable	Description	Units
`f_hi`	grid box mean ice thickness	m
`f_hs`	grid box mean snow thickness	m

continued on next page

Table 9: Time and Grid Information Written to History File

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
`tmask`	ocean grid mask (0=land, 1=ocean)	
`tarea`	T grid cell area	m^2
`uarea`	U grid cell area	m^2
`dxt`	T cell width through middle	m
`dyt`	T cell height through middle	m
`dxu`	U cell width through middle	m
`dyu`	U cell height through middle	m
`HTN`	T cell width North side	m
`HTE`	T cell width East side	m
`ANGLET`	angle grid makes with latitude line on T grid	radians
`ANGLE`	angle grid makes with latitude line on U grid	radians
`ice_present`	fraction of time-averaging interval that any ice is present	

continued from previous page

`f_fs`	grid box mean snow fraction	%
`f_Tsfc`	snow/ice surface temperature	C
`f_aice`	ice concentration (aggregate)	%
`f_aice1`	ice concentration (category 1)	%
`f_aice2`	ice concentration (category 2)	%
`f_aice3`	ice concentration (category 3)	%
`f_aice4`	ice concentration (category 4)	%
`f_aice5`	ice concentration (category 5)	%
`f_aice6`	ice concentration (category 6)	%
`f_aice7`	ice concentration (category 7)	%
`f_aice8`	ice concentration (category 8)	%
`f_aice9`	ice concentration (category 9)	%
`f_aice10`	ice concentration (category 10)	%
`f_vice1`	ice volume (category 1)	m
`f_vice2`	ice volume (category 2)	m
`f_vice3`	ice volume (category 3)	m
`f_vice4`	ice volume (category 4)	m
`f_vice5`	ice volume (category 5)	m
`f_vice6`	ice volume (category 6)	m
`f_vice7`	ice volume (category 7)	m
`f_vice8`	ice volume (category 8)	m
`f_vice9`	ice volume (category 9)	m
`f_vice10`	ice volume (category 10)	m
`f_uvel`	zonal ice velocity	cm s^{-1}
`f_vvel`	meridional ice velocity	cm s^{-1}
`f_fswdn`	downwelling solar flux	W m^{-2}

continued on next page

continued from previous page		
`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
`f_sss`	sea surface salinity	g kg^{-1}
`f_uocn`	zonal ocean current	cm s^{-1}
`f_vocn`	meridional ocean current	cm s^{-1}
`f_frzmlt`	freeze/melt potential	W m^{-2}
`f_fswabs`	absorbed solar flux sent to coupler	W m^{-2}
`f_fswabs_ai`	absorbed solar flux in snow/ocn/ice	W m^{-2}
`f_aldvr`	visible direct albedo	%
`f_aldvi`	near-infrared direct 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_flwout`	outgoing longwave flux sent to coupler	W m^{-2}
`f_flwout_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_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_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_fhnet`	ice/ocn net heat flux sent to coupler	W m^{-2}
`f_fhnet_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}
`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}
continued on next page		

continued from previous page		
`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	

8 Troubleshooting

8.1 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_model_size.F90** file in the source code directory, they will be overwritten by the file in **input_templates**.

8.2 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. Set

```
, xndt_dyn = 2
```

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

8.3 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, but sometimes can be indicative of a timestep problem in the ice.

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

8.4 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.

8.5 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.

8.6 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_run.xml** script. Before running the model, be sure to delete the object files so that the source code will be recompiled. If a core file is created, it will be in the executable directory. Use dbx to look at the core file. Useful information may also appear in the standard error and output files.

References

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

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

Briegleb, B. P. and B. Light, 2007: *A Delta-Eddington Multiple Scattering Parameterization for Solar Radiation in the Sea Ice Component of the Community Climate System Model.* NCAR Technical Note NCAR/TN-472+STR, National Center for Atmospheric Research, Boulder, Colorado.

Hibler, W. D., 1979: A dynamic thermodynamic sea ice model. *J. Phys. Oceanogr.*, **9**, 815–846.

Hunke, E. C., 2001: Viscous-plastic sea ice dynamics with the evp model: Linearization issues. *J. Comp. Phys.*, **170**, 18–38.

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

Hunke, E. C. and W. H. Lipscomb, 2008: *CICE: The Los Alamos Sea Ice Model. Documentation and Software User's Manual. Version 4.0.* T-3 Fluid Dynamics Group, Los Alamos National Laboratory, Tech. Rep. LA-CC-06-012.

Lipscomb, W. H., 2001: Remapping the thickness distribution in sea ice models. *J. Geophys. Res.*, **106**, 13,989–14,000.

Lipscomb, W. H. and E. C. Hunke, 2004: Modeling sea ice transport using incremental remapping. *Mon. Wea. Rev.*, **132**, 1341–1354.

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

Thorndike, A. S., D. S. Rothrock, G. A. Maykut and R. Colony, 1975: The thickness distribution of sea ice. *J. Geophys. Res.*, **80**, 4501–4513.