LIM

The Louvain-la-Neuve sea Ice Model

Martin Vancoppenolle,
Sylvain Bouillon, Thierry Fichefet, Hugues Goosse, Olivier Lecomte
Georges Lemaître Centre for Earth and Climate Research
contact:`mvlod@locean-ipsl.upmc.fr`
Earth and Life Institute Université catholique de Louvain, Louvain-la-Neuve, Belgium.

Miguel Angel Morales Maqueda
Proudman Oceanographic Laboratory, Liverpool, UK.

Gurvan Madec
Laboratoire d'Océanographie et du Climat, Paris, France.

Note du Pôle de modélisation de l'Institut Pierre-Simon Laplace No 31
ISSN No 1288-1619

January 17, 2012

Contents

1 Introduction

Sea ice, which refers to any form of ice found at sea which has originated from the freezing of seawater [WMO 1970], covers about 7% of the global ocean and 4.5% of the Earth surface, on annual average. Sea ice is recognized as a fundamental component of the Earth system. Some of the reference textbooks on the topic are Untersteiner [1986] and Thomas and Dieckmann [2010].

The present contribution focuses on the description of LIM (the Louvain-la-Neuve sea Ice Model), developed at the Université catholique de Louvain (UCL), Louvain-la-Neuve, Belgium[1]. LIM is a numerical model of sea ice designed for climate studies and operational oceanography. It is coupled to the ocean general circulation model OPA (Ocean Parallélisé) and is part of NEMO (Nucleus for European Modeling of the Ocean) system. The code of the coupled model is freely available under a software licensing agreement.

Two versions of the model are presently available (LIM2, LIM3) as well as a one-dimensional version (LIM1D) for process studies. LIM was originally developed by Fichefet and Morales Maqueda [1997]. The code was then rewritten formally at the Laboratoire d'Océanographie et du Climat (LOCEAN) by Christian Ethé and Gurvan Madec [see Timmermann et al. 2005], which resulted in LIM2. More recently, a new version of the model, LIM3, has been developed [Vancoppenolle et al. 2009b]. Here the focus is on LIM3 only, which will hereafter simply be referred to as "LIM".

LIM is a C-grid dynamic-thermodynamic sea ice model with thickness, enthalpy, salinity and age distributions. It includes the following components: thermodynamics, dynamics, advection, ridging and rafting. The ice pack is represented as a series of ice categories with specific area, thickness and state variables

[1] For more information, visit www.climate.be/lim/.

in order to represent unresolved subgrid-scale variations in ice thickness. Snow and ice thermodynamics compute the local growth and melt rates, the vertical temperature and salinity profiles based on the radiation and turbulent heat fluxes. Ice dynamics compute the ice drift as a function of wind, ocean and internal stresses. An advection scheme computes the rates of transport of the extensive state variables (area, volume, energy, salt). A ridging / rafting module computes the mechanical redistribution of state variables among categories as a function of the ice velocity field.

In addition to the authors of this note, there are a great number of people, or more exactly a number of great people, that are gratefully acknowledged for their various inspiring and or technical contributions to LIM.

Pierre-Yves Barriat - UCL, Belgium
Rachid Benshila - LOCEAN, Paris, France
Cecilia Bitz - University of Washington, Seattle, WA
Eric Deleersnijder - UCL, Belgium
Valérie Dulière - MUMM, Brussels, Belgium
Christian Ethé - LOCEAN / LSCE, Paris, France
Xavier Fettweis - ULg, Liège, Belgium
Jari Haapala - FIMR, Helsinki, Finland
Robinson Hordoir - SMHI, Sweden, Finland
Elizabeth Hunke - LANL, Los Alamos, NM
William H. Lipscomb - LANL, Los Alamos, NM
Olivier Lietaer - UCL, Belgium
Olivier Marti - LSCE, France
Pierre Mathiot - UCL, Belgium
Sébastien Masson - LOCEAN, Paris, France
François Massonnet - UCL, Belgium
NEMO team - LOCEAN, France
Claude Talandier - IFREMER, Brest, France
Benoît Tartinville - UCL, Belgium
Ralph Timmermann - AWI, Bremerhaven, Germany
Jean-Louis Tison - ULB, Brussels, Belgium

Of course, by essence, both the code and documentation are imperfect. You are welcome to contact us if you have any comment or question. This documentation is valid for model release NEMOv3.3, September 2011.

2 General features of the model

In this chapter, we give information on the representation of the ice in the model and on the basic model equations, for the reader who wants a general idea of the model but does not care about the details.

The mass balance of sea ice is the key diagnostic for climate simulations. The sea ice mass balance in a given region is determined, on the one hand, by ice growth and melt and by ice import in or export out of the region, on the other hand. The ice volume is the domain integral of ice concentration multiplied by ice thickness, while information on sea ice export requires the ice volume and velocity fields. Coverage, thickness and velocity are the main fields that LIM aims at simulating.

The representation of sea ice in LIM is done using a series of thermodynamic state variables (thickness, concentration, temperature, ...) and a horizontal velocity vector. Model variables change with time due to atmospheric and oceanic influence via dynamic and thermodynamic processes.

Ice in LIM has a snow cover which largely influences surface albedo and heat conduction. In addition, in order to resolve subgrid-scale variations in ice thickness, the ice pack in LIM is divided in sea ice categories with specific thickness, coverage and state variables, that move at the same velocity.

2.1 Ice thickness distribution theory

Ice thickness and coverage are tightly linked. Ice thickness presents large variations at spatial scales that are much smaller than the typical size of model grid cells. In order to account for subgrid-scale variations in ice thickness (h), a distribution function g for ice thickness is introduced (Thorndike et al., 1975). $g(t, \boldsymbol{x}, h)dh$ represents the relative area covered by sea ice with thickness between h and $h + dh$, at time t in a given region of area R centered at spatial coordinates $(\boldsymbol{x})$ (Thorndike et al., 1975). In other words,

$$\int_{h_1}^{h_2} g(t, \boldsymbol{x}, h)dh = \frac{S(h_1, h_2)}{R}, \tag{2.1}$$

where $S(h_1, h_2)$ is the area in R covered by ice with thickness between h_1 and h_2. In this framework, ice thickness h is an independent variable, that can be thought as random, which is on the same level as space and time. The mean ice thickness H and concentration A over R derive from g:

$$A(t, \boldsymbol{x}) = \int_{0^+}^{\infty} g(t, \boldsymbol{x}, h)dh \tag{2.2}$$

$$H(t, \boldsymbol{x}) = \int_{0^+}^{\infty} g(t, \boldsymbol{x}, h)hdh \tag{2.3}$$

where the 0^+ boundary implies that the means exclude open water. LIM tries to diagnose the temporal evolution of g over the entire ocean domain. For numerical purposes, the thickness distribution is discretized into several **ice thickness categories**. Invoking continuity (i.e., assuming that $g(h)$ is valid for any subregion of R), the conservation of ice area can be written:

$$\frac{\partial g}{\partial t} = -\nabla.(g\mathbf{u}) - \frac{\partial}{\partial h}(fg) + \psi^g, \tag{2.4}$$

The terms on the right-hand side refer to **horizontal transport** by the velocity field, advection in thickness space by the **thermodynamic** processes and **mechanical redistribution** due to ridging and rafting, respectively. In equation (2.4), **u** is the 2-D horizontal ice velocity vector, f is the thermodynamic vertical growth rate (m/s) and ψ^g is the so-called mechanical redistribution function. Functional of h and $g(h)$, it determines the changes in $g(h)$ during a deformation event, which redistributes ice area between the different thickness categories. Equation (2.4) is forced with **u**, f and the strain rate tensor. We redirect the reader interested in the basics of the ice thickness distribution theory to Thorndike et al. [1975].

2.2 State variables

In addition to ice thickness and concentration, changes in several other extensive thermodynamic state variables $X(t, \boldsymbol{x}, h)$ are diagnosed. The formulation of the changes in X follows the changes in the ice thickness distribution [Bitz et al. 2001], therefore:

$$\frac{\partial X}{\partial t} = -\nabla.(\mathbf{u}X) + D\nabla^2 X + \Psi^X + \Theta^X - \frac{\partial}{\partial h}(fX), \tag{2.5}$$

where Θ^X and Ψ^X represent the effects of thermodynamical and mechanical redistribution processes on X, respectively, and D is a horizontal diffusivity. Diffusion is added to smoothen the fields and avoid instabilities. Besides, there are diffusion effects associated with random-motion and the passage of synoptic scale weather systems [Thorndike 1986]. There is no explicit account for lateral melting. Lateral melting is controlled by floe size [Steele 1992], which is not a model variable. Nevertheless, as in other models including an ice thickness distribution, there is some implicit lateral melting accounted for by the melt of thin ice [Bitz et al. 2001].

The **global** (extensive) thermodynamic state variables of LIM are: **ice concentration**, **ice volume**, **snow volume**, **ice enthalpy**, **snow enthalpy**, **ice salt content**, **ice age content**. Ice velocity is also computed. The global thermodynamic state variables are used by all but thermodynamic model routines, in which they are, when necessary, converted into **equivalent** state (intensive) variables (i.e., thickness, temperatures). In Fig. 2.1, a synthesis plot showing how some of the global variables are discretized in thickness space is shown.

The modules associated to the computation of all of these terms are:

- **limdyn.F90**: Ice dynamics [Bouillon et al. 2009].
- **limtrp.F90**: Horizontal transport [Prather 1986].
- **limitd_me.F90**: Mechanical redistribution due to ridging and rafting [Vancoppenolle et al. 2009b].
- **limthd.F90**: Vertical ice thermodynamics.
- **limitd_th.F90**: Transports in thickness space due to ice growth and melt [Lipscomb 2001].
- **limthd_lac.F90**: Formation of new ice in open water.
- **limvar.F90**: Conversion between global (extensive) and equivalent (intensive) variables.

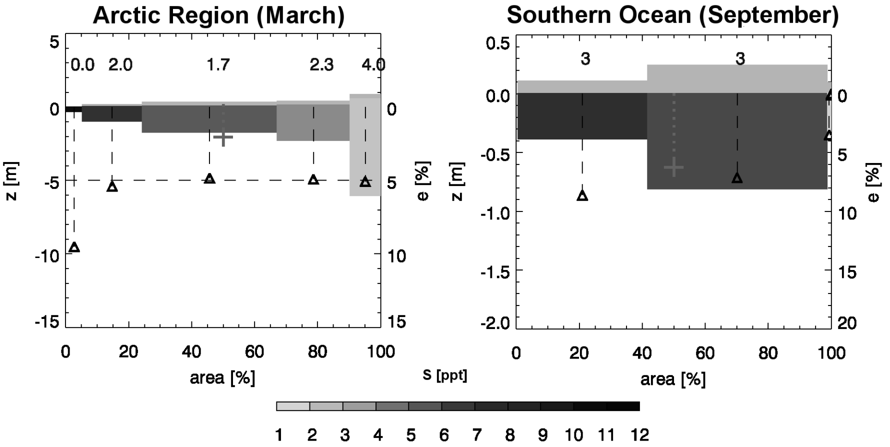

Figure 2.1: Synthesis plot of the representation of the ice pack for a hindcast run with LIM3 at 2° resolution [from Vancoppenolle et al. 2009b]. The average winter sea ice physical state (1979-2006) is depicted for the Arctic regions (including the Arctic Ocean and the Beaufort, Chukchi, East Siberian, Laptev and Kara Seas) and the whole Southern Hemisphere, detailed for each ice thickness category. The ice concentrations are represented horizontally. The ice thicknesses (negative values) and snow depths (positive values) are shown vertically (note the difference in scale between the two plots). The grey colors refer to the ice salinity. The black triangles indicate the relative brine volume e (computed from temperature and salinity), averaged over the vertical ice layers. The numbers on top are ice ages (years for the Arctic and months for the Antarctic). The cross indicates the mean ice thickness.

2.3 Ice thermodynamics

Ice thermodynamics correspond to all the processes that involve energy transfer through and storage into the ice, and hence may lead to net growth and melt of ice. The thermodynamic state of sea ice is defined in each grid cell, at each depth and for each thickness category, by the two sea ice state variables, temperature (T) and salinity (S). LIM thermodynamics include: the creation of new ice in open water, the congelation of ice at the base and the melt at both upper and lower interfaces, and the conversion of snow into ice at the ice surface (*snow-ice formation*). The computation of these terms involve: the interfacial energy budgets and the diffusion of heat, the radiative transfer, and the desalination of the ice. All these processes are assumed purely vertical in LIM.

Internal energy. The definition of internal energy is central to the conversion of energy into mass within the sea ice component. Following Bitz and Lipscomb [1999], we define the internal energy based on a specific energy of melting q, which is the total energy required for a unit volume of sea ice to warm, melt and reach the conventional energy reference level of 0°C:

$$q(S,T) = \rho_i c_0(-\mu S - T) + \rho_i L_0 \left(1 + \frac{\mu S}{T}\right) + c_w \mu S, \qquad (2.6)$$

for T in °C and S in ppt. ρ_i (kg.m^{-3}), c_0 (J.kg^{-1}.K^{-1}) and L_0 (J.kg^{-1}) are the specific heat, latent heat of fusion and density of fresh ice, respectively, μ is an empirical constant relating the seawater freezing point to salinity $T_m = -\mu S$ and c_w is the specific heat of seawater. For snow, the specific energy of melting is simply $-\rho_s(c_0 T + L_0)$, where ρ_s is the snow density (kg.m^{-3}). In this context, the total sea ice enthalpy in a given grid cell is:

$$Q(t, \boldsymbol{x}) = \int_{0^+}^{\infty} dhg(t, \boldsymbol{x}, h) \int_0^h dzq(t, \boldsymbol{x}, h) \qquad (2.7)$$

Creation of new ice. A surface ocean cell becomes ice-covered once temperature is equal to freezing and if the surface keeps losing heat. The volume of new sea ice formed per unit area is computed from the ocean surface heat loss. The initial thickness at which sea ice is formed is prescribed and the fractional area occupied by new ice is calculated accordingly. Creation of new ice is computed in **limthd_lac.F90**.

Heat diffusion, vertical growth and decay. The sea ice growth and melt rates (f) are determined by the balance between external heat fluxes and internal conduction fluxes. Vertical heat conduction and storage in the sea ice-snow system are governed by the heat diffusion equation:

$$\rho_i c \frac{\partial T}{\partial t} = -\frac{\partial}{\partial z}(F_c + F_r) \qquad (2.8)$$

where F_c and F_r are the conductive and radiation heat fluxes and $c = c(S,T)$ is the sea ice specific heat. Conduction fluxes follow Fourier's law and radiation fluxes are computed using Beer's law. The heat diffusion equation is resolved in LIM using one layer in snow and N vertical layers in sea ice (**limthd_dif.F90**).

The boundary conditions applied to (2.8) at the interfaces with atmosphere and ocean express the balance internal conduction and external heat fluxes. Any imbalance in the heat fluxes is converted in ice growth or melt. At the air-snow/ice interface, the energy balance is given by:

$$F^{net}(T_{su}) = F^{sw}(1-\alpha)(1-i_0) + F^{\downarrow lw} - \epsilon\sigma T_{su}^4 - F^{sh} - F^{lh} + F_c, \qquad (2.9)$$

where F_{sw} is the shortwave downelling radiation reaching the ice surface (positive downwards), α the surface albedo, i_0 the fraction of the solar radiation penetrating within the ice (i.e., which does not contribute to the surface energy balance), $F_{lw}^{\downarrow}$ is the downwelling longwave radiation flux (positive downwards), ϵ is the emissivity of the surface, σ is the Stefan-Boltzmann constant, F_{sh} and F_{lh} (positive upwards) are the turbulent sensible and latent heat fluxes, respectively, and F_c is the conductive flux from the sea ice/snow interior towards the top surface. If $F^{net} \geq 0$, then the temperature of the ice surface is fixed at the melting point (equal to 0°C) and ablation occurs according to:

$$F^{net}(T) = -q\frac{dh_x}{dt}, \tag{2.10}$$

where q is the sea ice energy of melting (defined as the energy required to melt a unit volume of sea ice) and the subscript x refers either to snow or ice. The temperature at the ice base equals the freezing point of seawater. At the ice base, the energy balance is given by:

$$F_w - F_c = -q\frac{dh_i}{dt}, \tag{2.11}$$

where F_w is the heat flux from the ocean and F_c is the conductive flux from the ice base towards the interior. Growth and melt rates are computed in **limthd_dh.F90**. To compute radiative transfer, exponential attenuation of radiation within ice and snow is assumed. In addition, the broadband surface albedo depends on surface temperature, snow depth, ice thickness and cloud fraction [Shine and Henderson-Sellers 1985].

Snow-ice formation. Snow-ice forms at the ice surface if the snow load is large enough to depress the snow-ice interface below sea level. It is assumed that snow-ice forms by the refreezing of seawater from the upper ocean level in the submerged snow layer. Heat and salt conservation are ensured during the process.

Ice salinity. There are two aspects of ice salinity handled by LIM. First, to account for the effect of internal phase changes associated to ice salinity, the thermal properties (c, k and q) depend on ice salinity and temperature. Second, LIM includes a parameterization of temporal changes in ice salinity, based on an equation for vertically-averaged ice salinity (**limthd_sal.F90**). The terms of tendency correspond to the entrapment of salt during basal growth and snow-ice formation, as well as the loss of salt due to gravity drainage and flushing. A vertical salinity profile is diagnosed from the bulk salinity. The parameterization well reproduces the large-scale ice salinity pattern [Vancoppenolle et al. 2009a].

More general information on ice thermodynamics can be found in: Maykut and Untersteiner [1971], Bitz and Lipscomb [1999]. More information on ice salinity can be found in: Notz and Worster [2009], Vancoppenolle et al. [2005; 2006; 2007; 2009a; 2010].

2.4 Ice motion

Ice moves and deforms under the influence of winds and ocean currents. The ice velocity **u** is determined from the conservation of linear momentum:

$$m\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + A\left(\boldsymbol{\tau}_a + \boldsymbol{\tau}_w\right) - mf\boldsymbol{k} \times \boldsymbol{u} - mg\boldsymbol{\nabla}\eta, \tag{2.12}$$

where m is the ice mass per unit area, $\boldsymbol{u}$ is the ice velocity, $\boldsymbol{\sigma}$ is the internal stress tensor, A is the ice concentration, $\boldsymbol{\tau}_a$ and $\boldsymbol{\tau}_w$ are the air and ocean stresses, respectively, f is the Coriolis parameter, $\boldsymbol{k}$ is a unit vector pointing upwards, g is the gravity acceleration and η is the ocean surface elevation with respect to zero sea level. LIM solves this equation assuming that sea ice is a viscous-plastic continuum. Numerically, this is done using the elastic-viscous-plastic (EVP) on a C-grid [Bouillon et al. 2009] The ice momentum equation is solved in **limdyn.F90** and **limrhg.F90**. More general information on ice dynamics can be found in Hibler [1979], Hunke and Dukowicz [1997], Leppäranta [2005], Bouillon et al. [2009].

2.5 Coupling with the atmosphere

LIM makes use of the same surface module as NEMO. Hence, the interested reader should refer to NEMO documentation to find the relevant information. However, we review the key principles of coupling in forced atmosphere and coupled modes.

2.5.1 Forced atmosphere

Dynamical forcings of LIM are the components of the wind stress vector (τ_x, τ_y, N/m^2). They can be specified directly (option 1) or computed via the components of the wind speed vector (option 2). Option 2 has been found to give much better results [see, e.g. Timmermann et al. 2005, Vancoppenolle et al. 2009b], as it allows the stress to decrease for relatively high ice velocities.

Thermodynamic forcings of LIM are the downwelling solar flux F^{sol}, the non-solar heat flux F^{nsol} (W/m^2), solid precipitation (kg/m^2/s) and the derivatives of the non-solar and latent heat fluxes with respect to surface temperature dF^{nsol}/dT_{su} and dF^{lh}/dT_{su}. The non-solar derivative is needed because the surface energy budget is linearized in ice thermodynamics. The latent heat derivative term is needed because the sublimation term needs the value of the latent heat flux at the end of the time step.

2.5.2 Interactive Atmosphere

By the time of writing this documentation, LIM3 was not yet coupled to an atmospheric model. We should include something here once the coupling is ready. The only ice-atmosphere interface that is presently available is valid for one-category sea ice model (e.g. LIM2). This interface would work with LIM3, though not ideally, since each ice category would see the same heat fluxes.

The fields received from the atmosphere are:

- 10-m wind
- Surface stress
- Solar flux
- Non-solar flux
- Sensitivity of the non-solar heat flux with respect to surface temperature
- Freshwater budget (E-P), runoff and iceberg calving

The fields sent to the atmosphere are:

- Surface temperature
- Albedo
- Ice thickness
- Ice concentration
- Surface current

2.6 Coupling with the ocean

Ice and ocean exchange momentum, heat, water and salt. The ice-ocean coupling is conceptually similar to the one of Goosse and Fichefet [1999]. Ice and ocean exchange momentum through an ice-ocean stress deriving from a quadratic bulk formula.The oceanic heat flux from the ocean to the ice is a function of the temperature of the ocean and turbulent mixing and derives from McPhee [1992] and Goosse and Fichefet [1999]. Freshwater fluxes for ocean and sea ice use the approach of Tartinville et al. [2001]. Adaptations were made to account for time-varying sea ice salinity. The following convention is adopted. First, if liquid or solid water is added to or extracted from the ice-ocean system (e.g., through precipitation or evaporation), then it is introduced as a freshwater flux. Second, if a surface process affects ocean salinity without a net gain or loss of water for the ice-ocean system (e.g., through ice melting or freezing) this internal exchange is transformed into an equivalent salt flux that is applied to the ocean. In this approach, sea ice acts as a negative reservoir of salt inside the ice-ocean system. As ice is formed, salt is released into the ocean. As ice melts, salt is taken out of the ocean.

2.7 Results of the model

Analysis of the results of the model typically include ice concentration, thickness and drift. For more info on model performance, the reader is redirected towards Vancoppenolle et al. [2009b] and Massonnet et al. [2011]. Both studies indicate that LIM3 is better than LIM2 with respect to several aspects both using ORCA2 and ORCA1 grids.

Analysis of LIM3 results show the following items. (i) The annual cycle of sea ice growth and decay is realistically captured with ice area, thickness, drift and snow depth in good agreement with observations. (ii) A detailed representation of the ice thickness distribution in LIM3 increases the seasonal to inter-annual variability of ice extent compared to LIM2, with spectacular improvement for the simulation of the recent observed summer Arctic sea ice retreats. (iii) More detailed physics in LIM3 than in LIM2 result in an overall improvement of the simulated geographical distribution of ice thickness and concentration. However, LIM3 still features an overestimation (underestimation) of ice thickness in the Beaufort gyre (around the North Pole) as well as an underestimation of ice thickness in the Southern Ocean. (iv) The elastic-viscous-plastic rheology in LIM3 enhances the response of ice to wind stress, compared to the classical viscous-plastic approach. (v) The grid formulation and the air-sea ice drag coefficient affect the simulated ice export through Fram Strait and the ice accumulation along the Canadian Archipelago. (vi) LIM2 and LIM3 models show less skill in the Southern Ocean, probably due to the low quality in this region of the atmospheric reanalyses used to drive the model and to the absence of important small-scale oceanic processes at current resolutions.

This closes this section on the general description of the model. We hope that you have now a general idea of how the model works. For more details on the representation of sea ice and snow physics in LIM3, see the corresponding sections in Chapter 3.

3 Model Components

This section is designed to the more interested reader who needs to put his hands in the code or understand in more detail a particular part of it. We tried to follow the structure of the code, following the routine codes in the main model module ***sbcice_lim.F90***. We provide both the technical information and some element on specific concepts when we thought that was needed.

3.1 Discretization and initialization of the ice thickness distribution

3.1.1 Discretization of the ITD

Module: ***iceini.F90***
Subroutine: **lim_itd_ini**

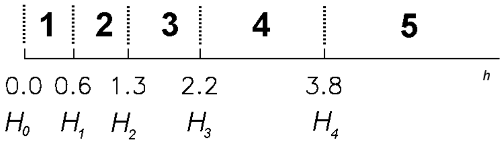

Figure 3.1: Boundaries of the model ice thickness categories (m) for 5 thickness categories.

The thickness distribution function $g(h)$ is numerically discretized into several ice thickness categories. The numerical formulation of the thickness categories follows Bitz et al. (2001) and Lipscomb (2001). A fixed number L of thickness categories with a typical value of $L = 5$ is imposed. For some variables, sea ice in each category is further divided into N vertical layers of ice and one layer of snow. In the remainder of the text, the $l = 1, ..., L$ index runs for ice thickness categories and $k = 1, ..., N$ for the vertical ice layers. Each thickness category has a mean thickness h_l^i ranging over $[H_{l-1}, H_l]$. $H_0 = 0$, while the other boundaries are chosen with greater resolution for thin ice (see Fig. 3.1), using:

$$H_l = H_{l-1} + \frac{3}{L} + \frac{30}{L}\left[1 + tanh\left(\frac{3l - 3 - 3L}{L}\right)\right]. \tag{3.1}$$

Each ice category has its own set of global state variables (see Table 3.1 for a list). The global ice state variables are extensive variables (e.g. concentration, volume per unit area, ...). The global state variables are used by all but thermodynamic model routines, in which they are, when necessary, converted into equivalent state (intensive) variables (i.e., thickness, temperatures, ...; see Table 3.2 for a list).

Table 3.1: Sea ice global (extensive) variables in LIM3.

Symbol	Description	Units
u	Sea ice velocity	$[m.s^{-1}]$
g_l^i	Concentration of sea ice in category l	[-]
v_l^i	Volume of sea ice per unit area in category l	$[m]$
v_l^s	Volume of snow per unit area in category l	$[m]$
$e_{l,k}^i$	Sea ice enthalpy per unit area in category l and layer k	$[J.m^{-2}]$
e_l^s	Snow enthalpy per unit area in category l	$[J.m^{-2}]$
M_l^s	Sea ice salt content in category l	$[‰.m]$
O_l	Sea ice age content in category l	$[days.m]$

3.1.2 Initialization of the ITD

Module: ***limistate.F90***

Subroutine: **lim_istate**

At the first model time step, if no *a priori* information is available, the state variables have to be initialized for each category. First, ice is assumed to be present initially where the ocean is sufficiently cold, .e.g., if $(SST - T^{fr,1})$ <*ttest*, where *ttest* is a namelist parameter (2 K by default) and $T^{fr,1}$ is the seawater freezing point. Then, numbers have to be assigned to all state variables.

Two quantities are prescribed for each hemisphere in *namelist_ice*: the mean ice thickness H^i (*hginn, hgins*) and the total concentration A^i, summed over the

Table 3.2: Equivalent variables in LIM3.

Symbol	Description	Units
$h_l^i = v_l^i / g_l^i$	Ice thickness	$[m]$
$h_l^s = v_l^s / g_l^i$	Snow depth	$[m]$
$q_{l,k}^i = e_{l,k}^i / (h_l^i / N)$	Ice specific energy of melting	$[J.m^{-3}]$
$q_l^s = e_l^s / h_l^s$	Snow specific energy of melting	$[J.m^{-3}]$
$T_{l,k}^i = T(q_{l,k}^i)$	Ice temperature	$[K]$
$T_l^s = T(q_l^s)$	Snow temperature	$[K]$
$\overline{S}_l^i = M_l^s / v_l^i$	Ice bulk salinity	$[‰]$
$o_l^i = O_l / g_l^i$	Ice age	$[days]$

categories (*aginn*, *agins*). The product $V^i = A^i H^i$ is the volume of ice per unit area in the grid cell.

First, we initialize the ice thickness (h_l^i) and concentration (g_l^i) into each category. We want g_l^i and h_l^i to respect area and volume conservation:

$$A^i = \sum_{l=1}^{L} g_l^i, \tag{3.2}$$

$$V^i = \sum_{l=1}^{L} g_l^i h_l^i. \tag{3.3}$$

Another constraint is that ice thickness has to be within the category bounds for the first $L-1$ categories, therefore we impose:

$$h_l^i = \frac{H_l + H_{l-1}}{2} \quad l = 1, ..., L-1. \tag{3.4}$$

In this context, there remains $L+1$ constants to be evaluated: h_L^i and g_l^i, $l = 1, ..., L$. Area and volume conservation provides two constraints, therefore we need $L-1$ additional equations to close the system.

In order to to this, we attempt to construct a gaussian distribution for concentrations. First, we decide that the maximum concentration is $A^i/\sqrt{L}$ and falls into the category l_0, which is such that $H_{l_0-1} < H^i < H_{l_0}$. This ensures that H^i represents the most probable thickness into the grid cell. For the other categories, we use a gaussian formulation. Therefore, initial ice concentrations read:

$$g^i_{l_0} = A^i/\sqrt{L} \tag{3.5}$$

$$g^i_l = g^i_{l_0} exp\left[-\frac{h^i_l - H^i}{H^i/2}\right]^2 \quad \forall l \neq l_0 \text{ and } l < L \tag{3.6}$$

$$\tag{3.7}$$

Those provide the $L - 1$ required constraints. Area and volume conservations are then used to compute the last two parameters:

$$g^i_L = A^i - \sum_{l=1}^{L-1} g^i_l, \tag{3.8}$$

$$h^i_L = \left(V^i - \sum_{l=1}^{L-1} g^i_l h^i_l\right) \Big/ g^i_L. \tag{3.9}$$

Finding such a gaussian distribution is not always possible for small values of H^i and can lead to potential problems, such as:

- Violation of area or volume conservation
- Thickness of the last category out of bounds: $h^i_L < H_{l-1}$
- Negativity of one of the concentrations

If that is the case, we prescribe the concentration in the last category to be zero and restart the operation for the $L-1$ categories, and so on, until a suitable distribution is found.

3.1.3 Initialization of the other state variables

Ice salinity (*sinn*, *sins*) and snow depth (*hninn*, *hnins*) are also prescribed in *namelist_ice* but have no category dependence. Initial ice age is set to 1 day. Surface, snow and ice temperatures are set to 270K. The remainder of the variables (ice volume, heat contents, ...) can then all be derived from the previously defined variables.

3.2 Ice dynamics and rheology

Modules: ***limdyn.F90***, ***limrhg.F90***
Subroutine: **lim_dyn**, **lim_rhg**

- I think here, we may add something more explicit on how forcings are computed

- Besides, I think we may like to have more info on the code (in particular the namelist parameters)

Besides ice thickness and concentration, the other important variable to diagnose the sea ice mass balance is the velocity vector. It is assumed in LIM that all ice categories behave as a single constituent moving at the same velocity. Therefore, the ice velocity vector is only a function of space and time. The ice velocity affects the ice thickness distribution through ice transport and mechanical redistribution.

The basis of the formulation of sea ice dynamics in LIM3 lies on the viscous-plastic (VP) approach of Hibler [1979]. In this framework, the ice cover is treated as a two-dimensional compressible fluid driven by winds and oceanic currents. Sea ice resists deformation with a strength which increases monotonically with ice thickness and concentration. Ice dynamics can be turned off using *ln_lim_dyn* in *namelist_ice*.

The ice velocity $\mathbf{u} = (u, v)$ is determined from the conservation of linear momentum:

$$m\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + A\left(\boldsymbol{\tau}_a + \boldsymbol{\tau}_w\right) - mf\boldsymbol{k} \times \boldsymbol{u} - mg\boldsymbol{\nabla}\eta, \tag{3.10}$$

where m is the ice mass per unit area, $\boldsymbol{u}$ is the ice velocity, $\boldsymbol{\sigma}$ is the internal stress tensor, A is the ice concentration, $\boldsymbol{\tau}_a$ and $\boldsymbol{\tau}_w$ are the air and ocean stresses, respectively, f is the Coriolis parameter, $\boldsymbol{k}$ is a unit vector pointing upwards, g is the gravity acceleration and η is the ocean surface elevation with respect to zero sea level. Scale analysis shows that momentum advection is at least one order of magnitude smaller than the acceleration term [Thorndike 1986] and therefore ignored in (3.10). Wind and ocean stresses are multiplied by the ice concentration as suggested by [Connolley et al. 2004]. The interested reader is redirected to Leppäranta [2005] for a complete discussion of ice dynamics.

Calculation of sea ice internal forces in LIM uses the elastic-viscous (EVP) approach [Hunke and Dukowicz 1997], which is a regularization of the VP model. A description of the general framework for the VP and EVP formulations of the ice internal stresses is given in Hunke and Dukowicz [2002] and in Hunke and Lipscomb [2010]. For completion, we reproduce here the key elements of such a framework. Let us denote σ_{11}, σ_{22} and σ_{12} the components of the ice internal

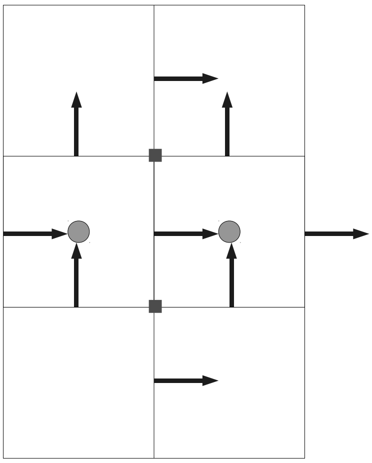

Figure 3.2: Localization of the dynamical fields on the C-grid. Circles indicate scalar points where horizontal divergence and tension are defined. Arrows indicate velocity points, and squares indicate where shear is defined.

stress tensor, and let

$$\sigma_1 = \sigma_{11} + \sigma_{22}, \tag{3.11}$$

$$\sigma_2 = \sigma_{11} - \sigma_{22}, \tag{3.12}$$

$$D_D = \frac{1}{h_1 h_2}\left(\frac{\partial}{\partial \xi_1}(h_2 u) + \frac{\partial}{\partial \xi_2}(h_1 v)\right), \tag{3.13}$$

$$D_T = \frac{1}{h_1 h_2}\left(h_2^2 \frac{\partial}{\partial \xi_1}(u/h_2) - h_1^2 \frac{\partial}{\partial \xi_2}(v/h_1)\right), \tag{3.14}$$

$$D_S = \frac{1}{h_1 h_2}\left(h_1^2 \frac{\partial}{\partial \xi_2}(u/h_1) + h_2^2 \frac{\partial}{\partial \xi_1}(v/h_2)\right), \tag{3.15}$$

where D_D is the divergence, D_T and D_S are the horizontal tension and shearing strain rates, ξ_1 and ξ_2 are generalized orthogonal coordinates, and h_1 and h_2 are the associated scale factors. With these definitions, an equivalent form of the VP rheology of Hibler [1979] is given by:

$$\sigma_1 = \left(\frac{D_D}{\Delta} - 1\right) P, \tag{3.16}$$

$$\sigma_2 = \frac{D_T}{e^2 \Delta} P, \tag{3.17}$$

$$\sigma_{12} = \frac{D_S}{2e^2 \Delta} P, \tag{3.18}$$

where P is the ice compressive strength, e is the eccentricity of the ice elliptical yield curve (see below) and Δ, a measure of the deformation rate, is given by

$$\Delta = \sqrt{D_D^2 + \frac{1}{e^2}\left(D_T^2 + D_S^2\right)}. \tag{3.19}$$

This rheology links the compressive stress, σ_1, to the shearing stress, $\sigma_s = \sqrt{\sigma_2^2 + 4\sigma_{12}^2}$, by a quadratic relation: $(\sigma_1 + P)^2 + e^2\sigma_s^2 = P^2$, which defines an elliptical yield curve. The ice compressive strength P is empirically related to the ice thickness per unit area, h, and ice concentration, A, by $P = P^* h\ e^{-C(1-A)}$, where P^* and C are empirical constants.

In (3.16)-(3.18), a regularization is needed when Δ goes to zero. A simple regularization is to set a lower bound, Δ_{min}, for Δ (VP regularization). In LIM, we use rather a regularization which consists in introducing time dependence and an artificial elastic term in (3.16)-(3.18), leading to the EVP formulation [Hunke and Dukowicz 1997]:

$$2T\sigma_{1,t} + \sigma_1 = \left(\frac{D_D}{\Delta} - 1\right)P, \tag{3.20}$$

$$\frac{2T}{e^2}\sigma_{2,t} + \sigma_2 = \frac{D_T}{e^2\Delta}P, \tag{3.21}$$

$$\frac{2T}{e^2}\sigma_{12,t} + \sigma_{12} = \frac{D_S}{2e^2\Delta}P, \tag{3.22}$$

where T is a time scale that controls the rate of damping of elastic waves. Note that, while (3.20)-(3.22) become (3.16)-(3.18) in the steady state, static flow in the EVP rheology is represented by an elastic deformation, and so imposing a minimum value of Δ is only necessary to prevent divisions by zero. In practice, a larger threshold value can be used in case of numerical instabilities, for example at high resolution, because this rheology has not yet been validated for such applications. Hunke and Dukowicz [1997] showed that the numerical solution of (3.10) in combination with (3.20)-(3.22) does indeed converge to the VP stationary solution as long as the elastic time scale T is several times smaller than the time scale of variation of the external forcing.

The components of the internal stress force are [Hunke and Dukowicz 2002]:

$$2F_1 = \frac{1}{h_1}\frac{\partial\sigma_1}{\partial\xi_1} + \frac{1}{h_1 h_2^2}\frac{\partial(h_2^2\sigma_2)}{\partial\xi_1} + \frac{2}{h_1^2 h_2}\frac{\partial(h_1^2\sigma_{12})}{\partial\xi_2}, \tag{3.23}$$

$$2F_2 = \frac{1}{h_2}\frac{\partial\sigma_1}{\partial\xi_2} - \frac{1}{h_1^2 h_2}\frac{\partial(h_1^2\sigma_2)}{\partial\xi_2} + \frac{2}{h_1 h_2^2}\frac{\partial(h_2^2\sigma_{12})}{\partial\xi_1}. \tag{3.24}$$

3.2.1 Ice strength

The ice compressive strength is by default formulated following [Hibler 1979]:

$$P = P^\star V^i exp[-C^{rhg}(1 - A^i)] \tag{3.25}$$

P^*, the ice strength parameter (*pstar* in *namelist_ice*) is set to $40,000$ N m^{-1}, which lies in the $30,000$–$45,000$ N m^{-1} satellite range [Tremblay and Hakakian 2007]. $C^{rhg} = 20$ (*c_rhg*) is an empirical constant. V^i is the total ice volume per unit ice area and A^i is the total concentration (see Section 3.1.2). The scheme based on Rothrock [1975] with a direct connexion between ice strength and ridge building is available in the code, but leads to instabilities. The associated parameter $C^p = 17$ (*Cp*) is the ratio of ridging work to increase in potential energy. Ice strength is smoothed to avoid steep gradients. Finally, an option to include a dependence on brine volume is available in the code (*brinstren_swi*) but is not activated by default.

3.2.2 Boundary conditions

On the C grid used in LIM, the normal velocity is defined and set to zero at the coast, but the tangential velocity is not defined. To impose a zero tangential velocity at the coast, a mirror velocity point is defined inland of the boundary, and its value is set to the opposite of the tangential velocity component seaward of the coast, thus delivering a no slip condition on the coast.

3.2.3 Practical details

The EVP rheology is used in LIM with an elliptical yield curve aspect ratio $e = 2$ (*ecc* in *namelist_ice*). The momentum equation is solved in ORCA2 using 300 (*nevp*) sub-iterations for ice dynamics, so that the time step used in the dynamic component is $\Delta t_{dyn} = 96$ s. The damping time scale is set to 9600 s (*telast*), which is high enough to ensure model stability Hunke [2001] and small enough so that elastic waves are damped when ice dynamics have converged. The stress tensor at the end of each time step is used at the beginning of the next iteration, which improves convergence and global model stability. The invariants of the stress tensor used to force the mechanical redistribution module are computed at the end of the dynamic time step.

3.3 Horizontal transport

Module: ***limtrp.F90***

This section focuses on how LIM solves the red part of the general equation:

$$\frac{\partial X}{\partial t} = -\nabla.(\mathbf{u}X) + D\nabla^2 X + \Psi^X + \Theta^X - \frac{\partial}{\partial h}(fX), \qquad (3.26)$$

where $X = X(t, \boldsymbol{x}, h)$ refers to any global sea ice state variable (see Table 3.1 for a list). Because it is useful to compute the net convergence of ice that is required when computing mechanical redistribution, the open water fraction ($A^{ow} = 1 - \sum_{l=1}^{L} g_l^i$) is also transported horizontally.

3.3.1 Diffusion

Modules: ***limtrp.F90***, ***limhdf.F90***

Subroutines: **lim_trp**, **lim_hdf**

Diffusion of state variables is a numerical artefact designed to dampen instabilities due to non-linearities between advection and ice dynamics. Horizontal diffusion is somewhat physically justified since there are some actual diffusive effects on the sea ice properties seen in nature [Thorndike 1986, Rampal et al. 2009]. In LIM, the diffusive term is solved explicitly. The horizontal diffusivity D is assumed constant inside the pack (namelist parameter *ahi0*) and zero at the ice edge. At a resolution of 2°, a standard value of 350 m^2/s is used. In order to keep diffusion relatively small compared to model dynamics, D should decrease with increasing resolution following an inverse square-root law. Tab. 3.3 gives typical values of D for different grid resolutions. Advection is assumed to conserve the vertical structure of the temperature profile, because we have no better hypothesis to constrain the model.

Table 3.3: Reference values of horizontal diffusivity D.

Resolution	D (m^2/s)
2°	350
1°	90
1/2°	20
1/4°	5
1/12°	2.5

3.3.2 Advection

Modules: ***limtrp.F90***, ***limadv.F90***

Subroutines: **lim_trp**, **lim_adv_x**, **lim_adv_y**

Advection of sea ice properties by the horizontal velocity field is computed based on the scheme of Prather [1986]. This scheme explicitly computes the conservation of second-order moments of the spatial distribution of global sea ice state variables. This scheme preserves positivity of the transported variables and is practically non-diffusive. It is also computationally expensive, however it allows to localize the ice edge quite accurately. As the scheme is conditionally stable, the time step is readjusted if the ice drift is too fast, based on the CFL criterion.

For each state variable, the 0^{th} (mean), 1^{st} (x, y) and 2^{nd} (xx, xy, yy) order moments of the spatial distribution are transported. At 1st time step, all moments are zero (if prescribed initial state); or read from a restart file, and then evolve through the course of the run. Therefore, for each global variable, 5 additional tracers have to be kept into memory and written in the restart file, which significantly increases the required memory. Advection following x and y are computed independently. The succession order of x- and y- advection is reversed every day.

3.4 Mechanical redistribution

Module: ***limitd_me.F90***

This section focuses on how LIM solves the red part of the general equation:

$$\frac{\partial X}{\partial t} = -\nabla.(\mathbf{u}X) + D\nabla^2 X + \Psi^X + \Theta^X - \frac{\partial}{\partial h}(fX), \tag{3.27}$$

where X refers to any global sea ice state variable (see Table 3.1 for a list).

3.4.1 Introduction to mechanical redistribution

Divergence and shear open the ice pack and create ice of zero thickness. Convergence and shear consumes thin ice and create thicker ice by mechanical deformation. The redistribution functions Ψ^X describe how opening and mechanical deformation redistribute the global ice state variables into the various ice thickness categories.

The fundamental redistribution function is Ψ^g, which accounts for area redistribution. The other redistribution functions Ψ^X associated with other state variables will derive naturally. The redistribution function Ψ^g should first ensure area conservation. By integrating the evolution equation for $g(h)$ over all thicknesses, recalling that $\int_0^\infty g(h) = 1$, and that the total areal change due to thermodynamics must be zero, e.g. $\int_0^\infty \partial(fg)/\partial h = 0$, then the area conservation reads:

$$\int_0^\infty h\Psi^g dh = \nabla \cdot \mathbf{u}. \tag{3.28}$$

Second, we must say something about volume conservation, and this will be done more specifically later. Following Thorndike et al. [1975], we separate the Ψ^X's into **(i)** *dynamical inputs*, **(ii)** *participation functions*, i.e., how much area of ice with a given thickness participates to mechanical deformation **(iii)** *transfer functions*, i.e., where in thickness space the ice is transferred after deformation.

3.4.2 Dynamical inputs

Subroutines: **lim_itd_me**

A general expression of Ψ^g, the mechanical redistribution function associated to the ice concentration, was proposed by Thorndike et al. [1975]:

$$\Psi^g = |\dot{\epsilon}|[\alpha_o(\theta)\delta(h) + \alpha_d(\theta)w_d(h,g)], \tag{3.29}$$

which is convenient to separate the dependence in $\mathbf{u}$ from those in g and h. The first and second terms on the right-hand side correspond to opening and deformation, respectively. $|\dot{\epsilon}| = (\dot{\epsilon}_I^2 + \dot{\epsilon}_{II}^2)^{1/2}$, where $\dot{\epsilon}_I = \nabla \cdot \mathbf{u}$ and $\dot{\epsilon}_{II}$ are the strain rate tensor invariants; $\theta =$ atan$(\dot{\epsilon_{II}}/\dot{\epsilon_I})$. $w_d(h,g)$, the deforming mode will be discussed in the next section. $|\dot{\epsilon}|\alpha_o$ and $|\dot{\epsilon}|\alpha_d$ are called the lead opening and closing rates, respectively.

The **dynamical** inputs of the mechanical redistribution in LIM are:

- $|\dot{\epsilon}|\alpha_o$, the opening rate,
- $|\dot{\epsilon}|\alpha_d$, the net closing rate.

Following Thorndike et al. [1975], we choose $\int_0^\infty w_d(h,g) = -1$. In order to satisfy area conservation, the relation $|\dot{\epsilon}|\alpha_o - |\dot{\epsilon}|\alpha_d = \nabla \cdot \mathbf{u}$ must be verified. In the model, there are two ways to compute the divergence of the velocity field. A first way is to use the velocity components ($\dot{\epsilon}_I = \nabla \cdot \mathbf{u}|^{rhg}$) as computed after the rheology (superscript rhg). Another way is to derive it from the horizontal transport of ice concentration and open water fraction. In principle, the equality $A^o + \sum_{l=1}^{L} g_L^i = 1$ should always be verified. However, after ice transport (superscript trp), this is not the case, and one can diagnose a velocity divergence using the departure from this equality: $\nabla \cdot \mathbf{u}|^{trp} = (1 - A^o - \sum_{l=1}^{L} g_L^i)/\Delta t$. In general, we will use $\dot{\epsilon}_I$ unless otherwise stated.

The **net closing rate** is written as a sum of two terms representing the energy dissipation by shear and convergence [Flato and Hibler 1995]:

$$|\dot{\epsilon}|\alpha_d(\theta) = C_s\frac{1}{2}(\Delta - |\dot{\epsilon}_I|) - \text{min}(\dot{\epsilon}_I, 0), \tag{3.30}$$

where Δ is a measure of deformation (defined in the rheology section). The factor $C_s = 0.5$ (*Cs* in *namelist_ice*) is added to allow for energy sinks other than ridge building (e.g., sliding friction) during shear. In case of convergence, the closing rate must be large enough to satisfy area conservation after ridging, so we take:

$$|\dot{\epsilon}|\alpha_d(\theta) = \text{max}(|\dot{\epsilon}|\alpha_d(\theta), -\nabla \cdot \mathbf{u}|^{trp}) \quad \text{if } \nabla \cdot \mathbf{u} < 0. \tag{3.31}$$

The **opening rate** is obtained by taking the difference:

$$|\dot{\epsilon}|\alpha_o = |\dot{\epsilon}|\alpha_d = \nabla \cdot \mathbf{u}|^{trp} \tag{3.32}$$

3.4.3 The two deforming modes

The deforming mode is separated into ridging w^{ri} and rafting w^{ra} modes:

$$w^d(h,g) = w^{ri}(g,h) + w^{ra}(g,h). \tag{3.33}$$

Rafting is the piling of two ice sheets on top of each other. Rafting doubles the participating ice thickness and is a volume-conserving process. Babko et al. [2002] concluded that rafting plays a significant role during initial ice growth in fall, therefore we included it into the model.

Ridging is the piling of a series of broken ice blocks into pressure ridges. Ridging redistributes participating ice on a various range of thicknesses. Ridging does not conserve ice volume, as pressure ridges are porous. Therefore, the volume of ridged ice is larger than the volume of new ice being ridged. In the model, newly ridged is has a prescribed porosity $p = 30\%$ (*ridge_por* in *namelist_ice*), following observations [Leppäranta et al. 1995, Høyland 2002]. The importance of ridging is now since the early works of [Thorndike et al. 1975].

The deforming modes are formulated using **participation** and **transfer** functions with specific contributions from ridging and rafting:

$$w_d(h,g) = -[b^{ra}(h) + b^{ri}(h)]g(h) + n^{ra}(h) + n^{ri}(h). \tag{3.34}$$

$b^{ra}(h)$ and $b^{ri}(h)$ are the rafting and ridging participation functions. They determine which regions of the ice thickness space participate in the redistribution. $n^{ra}(h)$ and $n^{ri}(h)$, called transfer functions, specify how thin, deforming ice is redistributed onto thick, deformed ice. Participation and transfer functions are normalized in order to conserve area.

3.4.4 Participation functions

Subroutine: **lim_itd_ridgeprep**

We assume that the participation of ice in redistribution does not depend upon whether the deformation process is rafting or ridging. Therefore, the participation functions can be written as follows:

$$b^{ra}(h) = \beta(h)b(h), \tag{3.35}$$

$$b^{ri}(h) = [1-\beta(h)]b(h), \tag{3.36}$$

where $b(h)$ is an exponential weighting function with an e-folding scale $a*$ [Lipscomb et al. 2007] (*astar* in *namelist_ice*) which preferentially apportions the thinnest available ice to ice deformation:

$$b(h) = \frac{\text{exp}[-G(h)/a^\star]}{a^\star[1-\text{exp}(-1/a^\star)]}, \tag{3.37}$$

It is numerically more stable than the original version of Thorndike et al. [1975]. This scheme is still present in the code and can be activated using *partfun_swi* from *namelist_ice*, with the associated parameter *Gstar*.

$\beta(h)$ partitions deforming ice between rafted and ridged ice. $\beta(h)$ is formulated following Haapala [2000], using the Parmerter [1975] law, which states that, under a critical participating ice thickness h_P, ice rafts, otherwise it ridges:

$$\beta(h) = \frac{\tanh[-C_{ra}(h - h_P)] + 1}{2}, \tag{3.38}$$

where $C_{ra} = 5$ m^{-1} (*Craft* in *namelist_ice*) and $h_P = 0.75$ m (*hparmeter* in *namelist_ice*) [Haapala 2000, Babko et al. 2002]. The $tanh$ function is used to smooth the transition between ridging and rafting. If namelist parameter *raftswi* is set to 0, ice only ridges and does not raft.

3.4.5 Transfer functions

Subroutine: **lim_itd_ridgeprep**

The rafting transfer function assumes a doubling of ice thickness :

$$n^{ra}(h) = \frac{1}{2}\int_0^\infty \delta(h - 2h')b(h')g(h')dh, \tag{3.39}$$

where δ is the Dirac delta function.

The ridging transfer function is :

$$n^{ri}(h) = \int_0^\infty \gamma(h', h)(1+p)b(h')g(h')dh. \tag{3.40}$$

The redistributor $\gamma(h', h)$ specifies how area of thickness h' is redistributed on area of thickness h. We follow [Hibler 1980] who constructed a rule, based on observations, that forces all ice participating in ridging with thickness h' to be linearly distributed between ice that is between $2h'$ and $2\sqrt{H^*h'}$ thick, where $H^\star = 100$ m (*Hstar* in *namelist_ice*). This in turn determines how to construct the ice volume redistribution function Ψ^v. Volumes equal to participating area times thickness are removed from thin ice. They are redistributed following Hibler's rule. The factor $(1+p)$ accounts for initial ridge porosity p (*ridge_por* in *namelist_ice*, defined as the fractional volume of seawater initially included into ridges. In many previous models, the initial ridge porosity has been assumed to be 0, which is not the case in reality since newly formed ridges are porous, as indicated by in-situ observations [Leppäranta et al. 1995, Høyland 2002]. In other words, LIM3 creates a higher volume of ridged ice with the same participating ice.

For the numerical computation of the integrals, we have to compute several temporary values:

- The thickness of rafted ice $h_l^{ra} = 2h_l^i$
- The mean thickness of ridged ice $h_l^{ri,mean} = \max(\sqrt{H^\star h_l^i}, h_l^i \cdot 1.1)$
- The minimum thickness of ridged ice $h_l^{ri,min} = \min[2 * h_l^i, 0.5 \cdot (h_l^{ri,mean} + h_l^i)]$
- The maximum thickness of ridged ice $h_l^{ri,min} = 2h_l^{ri,mean} - h_l^{ri,min}$
- The mean rate of thickening of ridged ice $k_l^{ri} = h_l^{ri,mean}/h_l^i$

3.4.6 Ridging shift

Subroutine: **lim_itd_ridgeshift**

The numerical computation of the impact of mechanical redistribution on ice concentration involves:

- A normalization factor that ensures volume conservation (*aksum*)
- The removal of ice participating in deformation (including the closing of open water)
- The addition of deformed ice

For ice concentrations, the numerical procedure reads:

$$\Delta g_l^i = C^{net} \Delta t \Bigg[-(b_l^{ri} + b_l^{ra}) + \sum_{l_2=1}^{L} \left(f_{l,l_2}^{ra} \frac{b_{l_2}^{ra}}{k^{ra}} + f_{l,l_2}^{ri} \frac{b_{l_2}^{ri}}{k_{l_2}^{ri}} \right) \Bigg] \tag{3.41}$$

- C^{net} is the normalized closing rate ($|\dot{\epsilon}|\alpha^d/$*aksum*)
- b_l^{ri} and b_l^{ra} are the area participating into redistribution for category l
- f_{l,l_2}^{ra} and f_{l,l_2}^{ri} are the fractions of are of category l being redistributed into category l_2
- k^{ra} is the rate of thickening of rafted ice (=2)

3.4.7 Iteration of mechanical deformation

Because of the nonlinearities involved in the integrals, the ridging procedure has to be iterated until $A^\star = A^{ow} + \sum_{l=1}^{L} g_l^i = 1$.

3.4.8 Mechanical redistribution for other global ice variables

The other global ice state variables redistribution functions Ψ^X are computed based on Ψ^g for the ice age content and on Ψ^{v^i} for the remainder (ice enthalpy and salt content, snow volume and enthalpy). The general principles behind this derivation are described in Appendix A of Bitz et al. [2001]. A fraction $f_s = 0.5$ (*fsnowrdg* and *fsnowrft* in *namelist_ice*) of the snow volume and enthalpy is assumed to be lost during ridging and rafting and transferred to the ocean. The contribution of the seawater trapped into the porous ridges is included in the computation of the redistribution of ice enthalpy and salt content (i.e., Ψ^{e^i} and Ψ^{M^s}). During this computation, seawater is supposed to be in thermal equilibrium with the surrounding ice blocks. Ridged ice desalination induces an implicit decrease in internal brine volume, and heat supply to the ocean, which accounts for ridge consolidation as described by Høyland [2002]. The inclusion of seawater in ridges does not imply any net change in ocean salinity. The energy used to cool down the seawater trapped in porous ridges until the seawater freezing point is rejected into the ocean.

3.5 Ice thermodynamics

This section focuses on how LIM computes the impact of thermodynamics on model variables

$$\frac{\partial X}{\partial t} = -\nabla.(\mathbf{u}X) + D\nabla^2 X + \Psi^X + \Theta^X - \frac{\partial}{\partial h}(fX), \tag{3.42}$$

3.5.1 Sea ice growth and melt

Modules: ***limthd.F90***

Subroutines: **lim_thd**

The main thermodynamic call is done in ***limthd.F90***. The computation of changes in thermodynamic variables associated with ice growth and melt has to be done for each category. For convenience, since there is no communication between adjacent cells for those routines, the ice-covered grid cells are selected and their spatial coordinates are stored into 1D arrays. Note that the category subscript l will be omitted for convenience in this subsection. Besides, we convert global variables into equivalent variables (volumes into thicknesses and enthalpy into temperatures) before thermodynamics. We return to global variables after thermodynamic computations.

For each category, a horizontally uniform sea ice layer is assumed. A vertical grid with one snow layer and N sea ice evenly-spaced layers is used (see Fig 3.3). By default, $N = 5$. The grid cell thickness changes in time due to ice growth and melt and is recomputed at each time step. In order to ensure energy conservation , internal energy has to be remapped on the new grid at each time step. The computational sequence is the following: (1) surface energy budget, diffusion of heat and radiative transfer; (2) ice growth and melt; (3) vertical remapping for energy, enthalpy and (4) ice desalination. Most of the the physical constants that should generally not be changed are defined in *phycst.F90*.

Surface energy balance

The energy flux from the atmosphere to the ice (positive to the ice) is formulated as:

$$F(T^{su}) = F^{sol} + F^{nsol}(T^{su}) \tag{3.43}$$

where F^{sol} is the incoming net solar radiation, F^{nsol} is the non solar (longwave, sensible and latent heat) contribution which depends on surface temperature. Each component of the atmospheric heat flux depends on sea ice characteristics, hence a specific value has to be computed for each ice thickness category.There are several formulations available in the code:

- The **CLIO** formulation, from Goosse [1997] (forced mode)
- The **CORE** formulation, from Griffies et al. [2009] (forced mode)
- The **COUPLED** formulation, from XXX (coupled mode)

The CLIO formulation for atmospheric energy fluxes

The **CLIO** atmospheric forcing combines daily values of NCEP/NCAR daily reanalysis data of 10-m wind speed V^a and 2-m temperature T^a [Kalnay et al. 1996], and monthly climatologies of relative humidity [Trenberth et al. 1989], cloud fractions c [Berliand and Strokina 1980] and precipitation P [Large and Yeager 2004]. Surface radiative and turbulent heat fluxes, briefly described hereafter, follow from [Goosse 1997].

The **net solar radiation** flux F^{sol} in the **CLIO** formulation is a linear combination of clear ($F^{sw,cs}$) and overcast ($F^{sw,os}$) skies contributions:

$$F^{sol} = \frac{1}{2\pi} \int_{t_1}^{t_2} dt[(1-c)(1-\alpha^{cs})F^{sw,cs} + c(1-\alpha^{os})F^{sw,os}], \tag{3.44}$$

where t_1 and t_2 are the time of sunrise and sunset, respectively; c is the fractional cloud cover; and α^{cs} (α^{os}) is the surface albedo for clear (overcast) skies. Besides, for highy reflective surfaces such as sea ice, it is necessary to take into account multiple reflections between the clouds and the surface. Therefore, for ice and snow, the parameterization of Shine [1984], which is designed for computing the shortwave flux on high albedo surfaces is used:

$$
\begin{aligned}
F^{sw,os} &= \frac{(53.5 + 1274.5\ \text{cos}Z)\sqrt{\text{cos}Z}(1-0.996\alpha^{os})}{1+0.139(1-0.9345\alpha^{os})\tau}dt \qquad (3.45)\\
F^{sw,cs} &= \frac{S_0\ \text{cos}^2 Z}{1.2 cosZ + 10^{-3}e(1+\ \text{cos}Z)+0.0455}dt \qquad (3.46)\\
& \qquad (3.47)
\end{aligned}
$$

in which Z is the solar zenith angle, computed as a function of latitude, day and hour, using astronomical equations [see, e.g. Peixoto and Oort 1992]; e is the near-surface water vapor pressure (in millibars); $S^0 = 1368$ W/m^2 is the solar constant: and τ the cloud optical depth. The inclusion of albedo in this formulation reflects its importance for multiple reflections. The optical depth τ depends on latitude as suggested by [Chou and Curran 1981]. A further correction has been introduced to take into account the variation of the distance between the sun and the Earth during the year [Oberhuber 1988].

The **non-solar flux** is expressed as:

$$
F^{nsol} = F^{lw,net} - F^{sh} - F^{lh}. \tag{3.48}
$$

$F^{lw,net}$ is the **net longwave radiation flux** (positive downwards). The parameterization used is from Goosse [1997], based on Berliand and Berliand [1952]:

$$
F^{lw} = \epsilon\sigma T^{a^4}(0.39 - 0.05\sqrt{e/100})[1-f(c)] + 4\epsilon\sigma T^{a^3}(T^{su} - T^{a}), \tag{3.49}
$$

where $\epsilon = 0.97$ is the surface emissivity for snow or ice; $f(c) = 1 - \chi c^2$, is a correction factor to take the effects of clouds into account. χ ranges over [0,1] and increases with latitude [Budyko 1974]. Note that the classical longwave emission is included in this equation.

F^{sh} and F^{lh} are the **turbulent fluxes of sensible and latent heat**, respectively (positive upwards). Without going into all the details [Goosse 1997], the formulas are:

$$
F^{sh} = \rho^a c^a C^{sh} V^a (T^a - T^{su}), \tag{3.50}
$$

$$
F^{lh} = \rho^a L^v C^{lh} V^a (q^a - q^{su}). \tag{3.51}
$$

ρ^a is the air density, c^a (J/kg) is the air specific heat, L^v is the latent heat of ice sublimation (J/kg), the C's are drag coefficients which depend on atmospheric stability functions. q^a is the air specific humidity and q^{su} is the specific humidity of the surface, assumed to be at saturation: $q^{su} = q_{sat}(T^{su})$, which is a non-linear function of surface temperature. Because the surface temperature is solved iteratively, the derivative with respect to temperature of the non-solar flux has is also computed in a consistent manner.

The CORE formulation for atmospheric heat fluxes Whoever knows what is in there adds something

The COUPLED formulation for atmospheric heat fluxes Add something here

Diffusion of heat

Modules: ***limthd_dif.F90***

Subroutines: **lim_thd_dif**

In order to compute the system of new temperatures $(T^{su}, T^{s}, T_1^i, ..., T_k^i, ..., T_n^i)$, we have to numerically resolve the system:

$$\begin{aligned} F(T^{su}) &= F^c|^{su}, \\ \rho c \frac{\partial T}{\partial t} &= -\frac{\partial}{\partial z}(F^c + F^r). \end{aligned} \tag{3.52}$$

ρ is the medium (snow or ice) density, c is the medium specific heat. $F^c = -k\partial T/\partial z$ is the heat conduction flux, with k the medium thermal conductivity. $F^r = I_0 e^{-\kappa z}$ is the radiation flux penetrating through the ice at depth z, with I_0 is the radiation flux penetrating through below the surface and κ the attenuation coefficient.

This system of equations must be constrained by boundary conditions. The surface flux equation is used as boundary condition for heat diffusion unless T goes beyond T^m, the surface melting point. In this case, $T^{su} = T^m$ is imposed. This implies that $F^{net} = F + F^c \geq 0$. The differential of energy is converted into surface melt:

$$F^{net}(T^m) = -q^{x,su}\frac{dh^x}{dt}, \tag{3.53}$$

where q^{x,su} refers to the enthalpy of the first snow or ice layer. The surface melting point T^m is chosen to be 0°C, because surface is always almost fresh.

At the ice base, we impose $T^b = T^{fr,1}$, with

$$T^{fr,1} = -0.0575S^w + 1.710523 \times 10^{-3}S^{w^{3/2}} - 2.154996 \times 10^{-4}S^{w^2}, \tag{3.54}$$

the seawater freezing point with a non-linear dependence on the sea surface salinity S^w, from the UNESCO formulation used in NEMO.

In the present version of LIM, snow has constant properties: $\rho^s = 330$ kg·m^{-3}, specific heat $c^s = c^0 = 2067$ J·kg^{-1}, thermal conductivity $k^s = 0.31$ W·K^{-1}·m^{-1} and latent heat $L^0 = 334$ kJ·kg^{-1}. c^0 and L_0 are the specific heat and latent heat of fusion of fresh ice, respectively. Snow has only one temperature point, no salinity and does not transmit radiation, therefore I_0=0 below when snow is present. Snow depth increases when snow falls and decreases due to snow ice formation, sublimation and melt. Clearly, the representation of snow is rather simple and work is under way to tackle this problem in future model versions [Lecomte et al. 2011].

The ice thermal properties depend on salinity S and temperature T, in order to take brine pockets into account, following the lines drawn by Maykut and Untersteiner [1971] and Bitz and Lipscomb [1999]. The equations read more easily for temperatures in °C, however in the code, we use Kelvin degrees. The sea ice specific heat is given by [Malmgren 1927]:

$$c^i(S,T) = c^0 + L^0 \mu \frac{S}{T}. \tag{3.55}$$

μ is an empirical constant relating the seawater freezing point to its salinity $T^{fr,2} = -\mu S$. Note that this formulation for seawater freezing differs from the non-linear formulation used to prescribe the boundary condition on ice temperature at the ice base (3.54). This inconsistency should be resolved in future versions of the code as part of a more consistent treatment of ice salinity. The sea ice thermal conductivity follows Pringle et al. [2007]:

$$k^i(S,T) = k^0 + \beta_1 \frac{S}{T} - \beta_2 T, \tag{3.56}$$

where $k^0 = 2.031$ W·K^{-1}·m^{-1} is the thermal conductivity of pure ice and $\beta_1 = 0.09$ and $\beta_2 = 0.011$ are empirical constants. There is an option in the namelist (*thcon_i_swi*) to use the older conductivity formula of Untersteiner [1964]. For melting and forming sea ice, instead of using the pure ice latent heat, we use the saline ice internal energy defined in Eq. 2.6. The ice density $\rho_i = 917$ kg·m^{-3} is considered as a constant.

Finite difference scheme for heat diffusion equation

Depending on whether snow is present and surface is melting ($T^{su} = T^m$), the finite-difference form of the system of temperature equations (3.52) changes. There are four main cases to discretize the system:

1. $h^s > 0, T^{su} < T^m$,
2. $h^s > 0, T^{su} = T^m$,

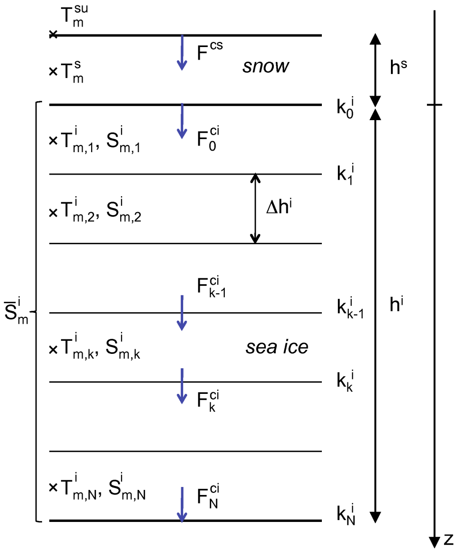

Figure 3.3: Vertical layering used to resolve the heat diffusion equation for each ice category.

3. $h^s = 0, T^{su} < T^m$,

4. $h^s = 0, T^{su} = T^m$.

There are N equations for the sea ice temperatures T^i_k, $k = 1, 2, ..., N$. If snow is present, there is an additional equation for T^s and if the surface is frozen, there is another one for T^{su}. The numerical scheme used is an implicit backwards-Euler, space-centered, with a Newton-Raphson method for coupled equations [Bitz and Lipscomb 1999], The scheme is first-order in time and second-order in space. The sea ice salinity S^i_k, $k = 1, 2, ..., N$ is assumed to be constant during heat diffusion.

Because the surface temperature equation and the sea ice thermal properties are non-linear in T^{su} and T^i, the numerical procedure has to be iterated. At each time step, temperatures are estimated by T^i_{n+1}. The iteration stops once the temperatures do not change by more than 10^{-4}°C (*nconv_i_thd* in *namelist_ice*)or when the number of iterations is greater than $N^{max} = 50$ (*maxer_i_thd* in *namelist_ice*).

The time derivative of snow and ice temperatures taken between the instants $n + 1$ and n is discretized as follows:

$$\frac{\partial T}{\partial t} = \frac{T_{n+1} - T_n}{\Delta t}, \tag{3.57}$$

with Δt the time step. The sea ice specific heat for layers $k = 1, 2, ..., N$ is:

$$c^i_k = c^0 + \frac{L_0 \mu S^i_k}{T^i_{n+1,k} T^i_{n,k}}. \tag{3.58}$$

This forms ensures that heat is conserved between steps n and $n+1$. The thermal conductivity (index k) at the interface between layers k and $k+1$ at time n reads:

$$\begin{aligned}
k_0^i &= k^0 + \beta_1 \frac{S_{n,1}^i}{T_{n,1}^i} - \beta_2 T_{n,1}^i, \\
k_k^i &= k^0 + \beta_1 \frac{S_{n,k}^i + S_{n,k+1}^i}{T_{n,k}^i + T_{n,k+1}^i} - \beta_2 \frac{T_{n,k}^i + T_{n,k+1}^i}{2} \qquad k = 1, 2, ..., N-1, \\
k_N^i &= k^0 + \beta_1 \frac{S_{n,N}^i}{T^b} - \beta_2 T^b,
\end{aligned} \tag{3.59}$$

The temporal index n in k and c was dropped for readability.

Now, let us develop the different terms of the system of equations 3.52, only for case 1, the other ones being quite similar. The linearization of the surface equation and the discretization of conductive and radiation fluxes are made in the following way:

$$\begin{aligned}
F_{n+1} &= F_* + \left.\frac{dF}{dT^{su}}\right|_* (T_{n+1}^{su} - T_*^{su}), \\
\frac{\rho^s c^s}{\Delta t}(T_{n+1}^s - T_n^s) &= -\frac{F_{n+1,0}^{ci} - F_{n+1}^{cs}}{h^s} + A^{sr}, \\
\frac{\rho^i c_k^i}{\Delta t}(T_{n+1,k}^i - T_{n,1}^i) &= -\frac{F_{n+1,k}^{ci} - F_{n+1,k-1}^{ci}}{\Delta h^i} + A_k^{ir}, \quad k = 1, ..., N.
\end{aligned} \tag{3.60}$$

The subscript $*$ refers to the onset of the latest iteration and the conductive fluxes correspond to vertical levels shown in Fig. 3.3. A_k^{ir} is the radiation flux absorbed in layer k. The conductive heat fluxes are discretized as follows:

$$\begin{aligned}
F^{cs} &= -\frac{2k^s}{h^s}(T^s - T^{su}), \\
F_0^{ci} &= -K^{int}(T_1^i - T^s), \\
F_k^{ci} &= -\frac{k_k^i}{\Delta h^i}(T_{k+1}^i - T_{k+1}^s), \quad k = 1, ..., N-1; \\
F_N^{ci} &= -\frac{2k_N^i}{\Delta h^i}(T^b - T_N^i),
\end{aligned} \tag{3.61}$$

where $K^{int} = 2k_0^i k^s / (k_0^i h^s + k^s \Delta h^i)$ is an effective conductivity for the interface that can be derived by equalling the lowermost heat conductive flux in the snow and the uppermost one in the sea ice. The radiation flux absorbed in each ice layer is formulated simply as:

$$\begin{aligned}
A_k^{ir} &= \frac{T_{k-1}^{ir} - T_k^{ir}}{\Delta h^i}, \\
T_k^{ir} &= I_0 e^{-\kappa^i z_k^i},
\end{aligned} \tag{3.62}$$

where z_k^i is the vertical cote of the top of layer k as referred to the snow-ice interface. Defining

$$\begin{aligned}
&\eta^s = \Delta t/(h^s \rho^s c^0), \\
&\eta^i = \Delta t/(\Delta h^i \rho^i c^0), \\
&K^s = 2k^s/h^s, \\
&K_k^i = k_k^i/\Delta h^i \quad k = 1, ..., N-1, \\
&K_N^i = 2k_N^i/\Delta h^i,
\end{aligned} \tag{3.63}$$

we get the equations in the final form:

$$\begin{aligned}
\left[\frac{dF}{dT^{su}}\bigg|_* - K^s \right] T_{n+1}^{su} + K^s T_{n+1}^s = \frac{dF}{dT^{su}}\bigg|_* T_*^{su} - F_*, \\
T_{n+1}^{su}[-\eta^s K^s] + T_{n+1}^s[1 + \eta^s(K^s + K^{int})] + T_{n+1}^{i1}[-\eta^s K_0^i] &= T_n^s + \eta^s A^{rs}, \\
T_{n+1}^s[-\eta^i K^{int}] + T_{1,n+1}^i[1 + \eta^i(K^{int} + K_1^i)] + T_{2,n+1}^i[-\eta^i K_1^i] &= T_{n,1}^i + \eta^i A_1^{ri}, \\
T_{n+1}^{k-1}[-\eta^i K_{k-1}^i] + T_{k,n+1}^i[1 + \eta^i(K_{k-1}^i + K_k^i)] + T_{k,n+1}^i[-\eta^i K_k^i] &= T_{n,k}^i + \eta^i A_k^{ri}, \\
k &= 2, ..., N-1 \\
T_{n+1}^{N-1}[-\eta^i K_{N-1}^i] + T_{N,n+1}^i[1 + \eta^i(K_N^i - 1 + K_N^i)] &= T_{n,N}^i + \eta^i A_N^{ri} \\
&+ T^b[\eta^i K_N^i],
\end{aligned} \tag{3.64}$$

which is a tridiagonal system that is resolved using the Thomas algorithm. At the end of the procedure, the temperatures and the energy flux are computed.

Radiative transfer

Downwelling radiative flux is separated into longwave ($F_{lw}^{\downarrow}$) and shortwave (F$_{sw}$) contributions. $F_{lw}^{\downarrow}$ is integrally absorbed at the surface, while F_{sw} is partly reflected upwards, absorbed at the surface, attenuated through the ice, and transmitted below the ice to the ocean. Several coefficients characterize shortwave radiation transfer through the sea ice (see Fig. 3.4).

Surface shortwave radiation budget

Modules: ***albedo.F90***, ***sbcblk_core.F90***, ***sbcblk_clio.F90***

The **surface albedo** α determines the fraction of incoming solar radiation reflected by the surface. The clear-sky sea ice albedo α_{cs}^i is parameterized following Shine and Henderson-Sellers [1985], as a function of snow depth, ice thickness and surface temperature (see Tab 3.4 and 3.5). As these variables are specific for each ice category, each category ends up with its own surface albedo. The parameterization interpolates albedo between several pivotal values, which are hard-coded in the routine (see Tab. 3.4). Depending on the forcing used, $\overline{\alpha}_b$ (*albice*) can be tuned within 10%, which has a moderate effect on summer ice characteristics.

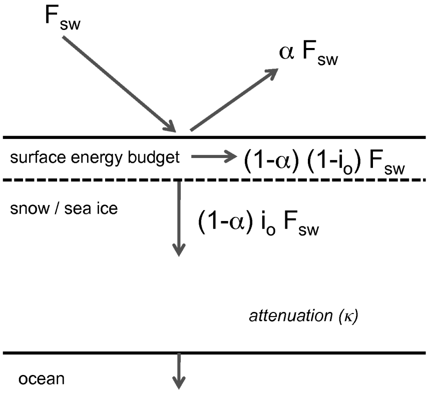

Figure 3.4: Shortwave radiation transfer through the sea ice for each thickness category, in the simple case of **CORE** forcing formulation

In the case of **CORE** forcing, only the clear-sky albedo value is used. In the case of **CLIO** forcing, an overcast-sky value is also computed $\alpha^i_{os} = \alpha^i_{cs} + 0.06$. The broadband SW sea ice albedo increases in nature because clouds change the spectral distribution of incoming light [Grenfell and Perovich 1984]. Following this, the absorbed shortwave flux is formulated as a linear combination of clear sky and overcast sky contributions, weighted by cloud fraction. Therefore, the radiation budget in case of the **CLIO** forcing is more complex than what is shown on Fig. 3.4).

F_{sw} is divided at the ice surface into several components:

- Reflected SW radiation: $F_{sw}^{\uparrow}$,
- Net incoming SW radiation: $F_{solar} = (F_{sw} - F_{sw}^{\uparrow})$
- SW radiation absorbed near the surface: $(1 - i_o)F_{solar}$,
- Transmitted radiation: $I_0 = i_o F_{solar}$.

The simple **CORE** forcing case, for which both F_{sw} and α do not depend on cloud fraction is shown in Fig. 3.4. i_o is a surface transmission parameter that has been introduced by Grenfell and Maykut [1977] in order to represent the surface scattering layer at the top of the surface, which implies that radiation does not follow a regular Beer-Lambert law near the surface. We set $i_o = 0$ in presence of snow and use the parameterization of Ebert and Curry [1993] for bare ice:

$$i_o = 0.18(1 - c)0.35c, \tag{3.65}$$

Table 3.4: Key albedo parameters in the parameterization of Shine and Henderson-Sellers [1985]

Snow / sea ice class	Symbol	Value	Variable name in the code
Thick dry snow	$\overline{\alpha}^s_f$	0.8	*alphd*
Thick melting snow	$\overline{\alpha}^s_m$	0.65	*alphc*
Thick frozen bare ice	$\overline{\alpha}^i_f$	0.72	*alphdi*
Thick melting bare ice	$\overline{\alpha}^i_m$	0.53	*albice*

Table 3.5: Surface albedo parameterization of Shine and Henderson-Sellers [1985]

Frozen bare ice	α^i_f	= $\overline{\alpha}^i_f$ if $h^i > 1.5$
		= $0.472 + 2(\overline{\alpha}^i_f - 0.472)(h^i - 1.0)$ if $1.0 \leq h^i \leq 1.5$
		= $0.2467 + 0.7049h^i - 0.8608{h^i}^2 + 0.3812h^{i3}$ if $0.05 \leq h^i \leq 1.0$
		= $0.1 + 3.6h^i$ if $0 \leq h^i \leq 0.05$
Melting bare ice	α^i_m	= $\overline{\alpha}^i_m$ if $h_i > 1.5$
		= $0.472 + 2(\overline{\alpha}^i_m - 0.472)(h^i - 1.0)$ if $1.0 \leq h^i \leq 1.5$
		= $0.2467 + 0.7049h^i - 0.8608{h^i}^2 + 0.3812h^{i3}$ if $0.05 \leq h^i \leq 1.0$
		= $0.1 + 3.6h^i$ if $0 \leq h^i \leq 0.05$
Dy snow	α^s_f	= $\overline{\alpha}^s_f$ if $h^s \geq 0.05$
		= $\alpha^i_f + h^s(0.8 - \alpha^i_f)/0.05$ if $h^s \leq 0.05$
Melting snow	α^s_m	= $\overline{\alpha}^s_m$ if $h^s \geq 0.1$
		= $\overline{\alpha}^i_m + (\overline{\alpha}^s_m - \overline{\alpha}^i_m)h^s/0.1$ if $h^s \leq 0.1$

where c is cloud fraction.

Attenuation of shortwave radiation through the ice

Modules: ***limthd_dif.F90***

Subroutines: **lim_dif**

We assume exponential attenuation of penetrating radiation into the ice, using a constant extinction coefficient κ (*kappa_i* in *namelist_ice*). We assume that the transmitted radiation stands in the visible spectrum only. Therefore, the value we use for the extinction coefficient (*kappa_i* is $\kappa = 1$ m^{-1}), is the mean of observed values for dry and melting ice in the visible band[Light et al. 2008]. The code is written in a way that adding penetration of radiation through snow is easy.

Vertical sources and sinks of snow and ice mass

Modules: ***limthd_dh.F90***

Subroutines: **lim_thd_dh**

Here, the vertical processes of ice growth and melt, which are treated identically for each thickness category, are described. The snow depth and ice thickness increments associated with vertical thermodynamics are:

$$\Delta h^s|^{\theta_v} = \Delta h^{s,p} + \Delta h^{s,su-} + \Delta h^{s,sub} + \Delta h^{s,si}, \tag{3.66}$$

$$\Delta h^i|^{\theta_v} = \Delta h^{i,su-} + \Delta h^{i,b+} + \Delta h^{i,b-} + \Delta h^{i,si}. \tag{3.67}$$

The processes that affect snow depth are: snow fall ($\Delta h^{s,p}$), surface melt ($\Delta h^{s,su-}$), and snow sublimation ($\Delta h^{s,sub}$). Changes in ice thickness are due to: surface melt ($\Delta h^{i,su-}$) basal growth ($\Delta h^{i,b+}$), basal melt ($\Delta h^{i,b-}$) and snow ice formation ($\Delta h^{i,si}$).

Snow fall

In the case of **CLIO** forcing, the monthly mean climatological value of total precipitation P [kg/m^2/s] is read from the monthly climatology of Large and Yeager [2004]. Only a fraction f^{solid} of total precipitation is assumed to fall under the form of snow:

$$f^{solid} = \begin{cases} 1 & \text{if } T^a \leq 253, \\ 0.5 + (272 - T^a)/38 & \text{if } 253 < T^a \leq 272, \\ (281 - T^a)/18 & \text{if } 272 < T^a \leq 281, \\ 0 & \text{if } 281 < T^a, \end{cases} \tag{3.68}$$

the remainder being liquid. T^a is the atmospheric temperature in K. This parameterization, from Ledley [1985] ascribes all precipitation to be solid (liquid) if the air is sufficiently cold (warm) with hybrid regimes in between. In the case of **CORE** forcing, solid precipitation field is directly read from the forcing files. In all cases, liquid precipitation, if any, falls directly into the ocean.

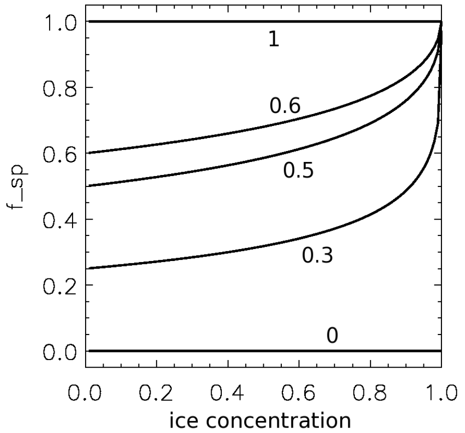

Figure 3.5: Evolution of blowing snow partition factor $f^{s,p}$ as a function of total ice concentration A for different values of the blowing snow parameter β^s.

Snowfall rate is converted into snow accumulation $\Delta h^{s,p}$ [m] using the relation:

$$\Delta h^{s,p} = f^{s,p} \rho^s f^{solid} P \Delta t \tag{3.69}$$

where $f^{s,p}$ is the fraction of solid precipitation falling on the ice-covered fraction of the cell:

$$f^{s,p} = \frac{1-(1-A)^{\beta^s}}{A}. \tag{3.70}$$

$0 < \beta^s \leq 1$ is the blowing snow parameter (*betas* in *namelist_ice*). If wind is assumed to have no impact on falling snow ($\beta^s = 1$), the same amount of snow falls onto ice and into open water. However, because of the winds, more snow falls in open water, which is represented by $\beta^s < 1$. The unrealistic extreme case for which all the snow falls into open water would be represented by the value $\beta^s = 0$. By default, in the namelist, $\beta^s = 0.6$. The blowing snow partition factor is depicted in Fig. 3.5.

Snow and ice surface melt.

Snow, if any, or conversely ice, melts at the surface if the temperature is equal to T^m, following:

$$\Delta h^{x,su-} = -\frac{F^{net}\Delta t}{q} \tag{3.71}$$

Freshly fallen snow – $q = q^s(T^a)$, snow – $q = q^s(T^s)$, and then the ice layers – $q = q^i(S^i_k, T^i_k)$, successively melt until the energy available for surface ablation $F^{net}\Delta t$ is exhausted.

Snow sublimation.

Sublimation is a transition from the solid to gas phase with no intermediate liquid stage. Sublimation is a phase transition that occurs at temperatures and pressures below the triple point. Snow and ice sublime at below-freezing temperatures. Condensation or deposition is the opposite process. In LIM, there is only a representation of snow sublimation. There is no deposition, which does not matter in both **CLIO** and **CORE** forcing configurations because the latent heat flux is prevented to be directed downwards anyway. Ice does not sublimate or condensate. Snow sublimation occurs if the latent heat flux is positive (e.g. directed upwards):

$$\Delta h^{s,sub} = -\frac{\text{max}(F^{lh}, 0)\Delta t}{\rho^s L^{sub}} \tag{3.72}$$

where $L^{sub} = 2834$ kJ/kg. The fact that a latent heat loss is required for sublimation may sound counter-intuitive. Sublimation requires heat (e.g. from solar radiation), but results in a latent heat loss for the snow/ice system. This latent heat loss corresponds to a transfer of heat from the snow to the atmosphere in sensible form, associated with the transfer of water vapor. The counter-intuitive nature of our parameterization is because sublimation is derived from the latent heat flux, whereas this latent heat flux is a consequence and not the cause of sublimation. Ideally, sublimation should be parameterized as a function of the air saturation state, which is possible only by the full coupling of water vapour with an atmospheric model. Model results using LIM1 [Fichefet and Morales Maqueda 1999] indicated that snow sublimation had considerable effect on simulated snow cover conditions around Antarctica, where winds are strong and air can be very dry in winter. Snow sublimation can be activated or desactivated using the namelist parameter *parsub*.

Besides, there are two energy conservation issues associated with snow sublimation that should be fixed.

1. The latent heat used to sublimate snow should include more terms:

$$L^{sub} = q^s(T) + \rho^s L^{vap} + \rho^a c^a T^a. \tag{3.73}$$

   The first term includes both the sensible heat required to warm snow and the latent heat of fusion; the second term is associated with the latent heat of vaporization; and the last term represents the fact that water vapor has to be given a certain atmospheric temperature, which would be assumed to be the air temperature.

2. In the present version of the code, the latent heat losses are counted twice. Once to cool the snow-ice system ice in the diffusion of heat. And a second time to sublimate snow.

Basal growth.
If heat is lost at the ice base, ice grows. The ice growth rate is given by:

$$\Delta h^{i,b+} = -\frac{\max[(F^w - F^c|^b), 0]\Delta t}{q(S^b, T^b)}, \tag{3.74}$$

where F^w is the turbulent oceanic heat flux (computed by the ocean model) and $F^c|^b$ is the internal heat conduction flux at the ice base. T^w is the ice-ocean interface temperature, which is equal to the seawater freezing point. S^b is the salinity of new ice. If salinity variations are activated, then, as shown later, S^b depends on the growth rate. Therefore, the equations for basal ice salinity, for new ice enthalpy and basal growth accumulation have to be solved together iteratively. The associated enthalpy per unit area is added at the ice base.

Basal melt.

Basal melt is very similar to basal growth. Basal melt occurs if heat converges at the ice base, i.e. if the ocean brings more heat than is conducted through the ice:

$$\Delta h^{i,b-} = -\frac{\min[(F^w - F^c|^b), 0]\Delta t}{q(S^i_N, T^i_N)}, \tag{3.75}$$

where this time, the ice enthalpy is the one of the lowermost ice layer. The ice layers successively melt until the energy available for surface ablation $F^{net}\Delta t$ is exhausted.

Snow ice.

The formation of snow ice results from the transformation of snow into ice at the snow-ice interface. As in Fichefet and Morales Maqueda [1997], we assume that when snow is thick enough to depress the snow-ice interface below the sea level, seawater infiltrates and refreezes into the snow, so that snow ice formation occurs:

$$\Delta h^{i,si} = \frac{\rho_s h^s - (\rho_w - \rho_i)h^i}{\rho_s + \rho_w - \rho_i}, \tag{3.76}$$

where ρ_w and ρ_s are the reference densities of seawater and snow, respectively. The newly created ice is assumed to have $T = T^w$ and a salinity S^{si}. A corresponding enthalpy per unit area $q(S^{si}, T^w)\Delta h^{si}$ is added on top of the sea ice surface and merged with the remainder of the sea ice column. The residual energy (used to warm the snow until T^w) is removed from the top ocean layer. The increment in snow depth is just the opposite of the increment in ice thickness: $\Delta h^{s,si} = -\Delta h^{i,si}$.

Vertical remapping of snow and ice enthalpy

Modules: ***limthd_ent.F90***

Subroutines: **lim_thd_ent**

LIM requires one layer of snow and N sea ice layers of equal thickness. Hence, after ice growth and melt, temperatures at the mid-point have to be recomputed in an energy-conserving manner. This is done by redistributing the enthalpy per unit area $Q^x = q^x \Delta h^x$ (J/m^2) on the new grid. This procedure, called **vertical remapping** is illustrated for one particular case in Fig. 3.6.

The steps involved in this operation are the following. First, the growth and melt processes that are active are identified. Then, the vertical distribution of enthalpy *before* vertical remapping (”*old*” profile, index ** in Fig. 3.6) is stored in an array. The old enthalpies are computed using the temperatures evaluated after diffusion. They also include the enthalpy sources associated with snowfall, basal growth and snow ice formation. The vertical coordinates of all these *old* layers are also stored in an array. Once this is done, the vertical coordinates of

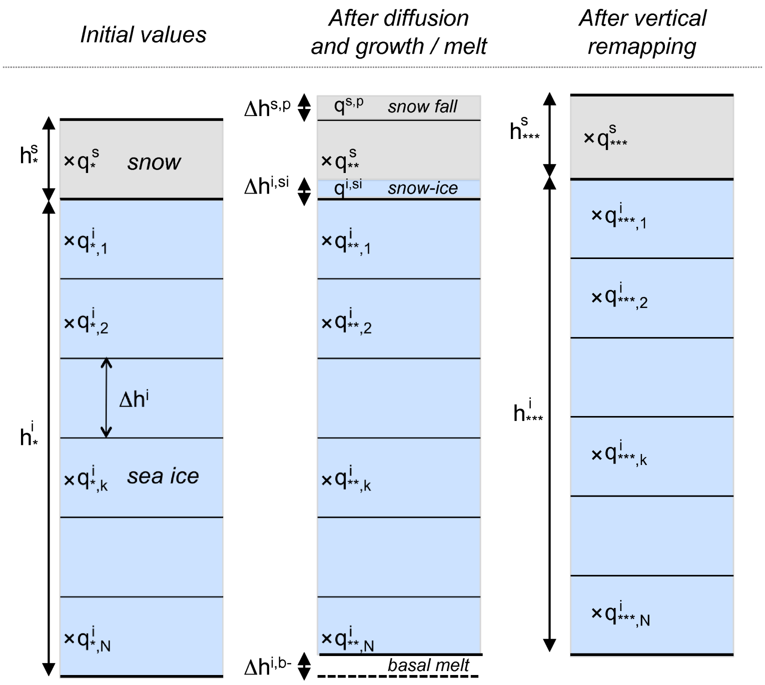

Figure 3.6: An illustration of the vertical enthalpy remapping procedure in a case for which snow falls, snow ice forms and ice melts at the base.

layers of the new, uniform grid are computed (referred to as the *new* profile, index *** in Fig. 3.6).

The Q^x are linearly redistributed from the old to the new grid:

$$Q^{x,new}_{k^{new}} = \sum_{k^{old}} w_{k^{old},k^{new}} Q^{x,old}_{k^{old}} . \tag{3.77}$$

The weight factors $w_{k^{old},k^{new}}$,

$$w_{k^{old},k^{new}} = \frac{\max[0, \min(z^{old}_{k^{old}}, z^{new}_{k^{new}})] - \max[0, \min(z^{old}_{k^{old}-1}, z^{new}_{k^{new}-1})]}{\Delta h^{old}_{k^{old}}}, \tag{3.78}$$

give the contribution of each *old* layer to the *new* layers. Finally, the new enthalpies $Q^{x,new}$ are converted into specific enthalpies q^x and then, into temperatures T^x.

3.5.2 Sea ice halodynamics

The global sea ice state variable associated to the ice salinity in the l^{th} category is the sea ice salt content $M_l^s = \overline{S}^i_l v^i_l$, $l = 1, ..., L$, where $\overline{S}^i_l$ is the vertically averaged (hereafter referred to as "bulk") sea ice salinity in the l^{th} category.

The *num_sal* parameter in *namelist_ice* controls the five available options:

1. Constant ice salinity (value given by *bulk_sal* in *namelist_ice*);
2. Salinity varying both vertically and in time;
3. Prescribed multi-year ice salinity profile following observations by [Schwarzacher 1959];
4. Salinity varying vertically only
5. Ice salinity function of ice thickness, following the parameterization of [Cox and Weeks 1974]).

Hereafter, we describe the default option (2). In that case, M_l^s follows an evolution equation of the type of eq. [3.87]. Θ^{M^s} includes (1) brine entrapment and drainage and (2) transport in thickness space due to ice growth/melt. As transport is thickness space is discussed later in this text, we focus on brine entrapment and drainage. For thermodynamic computations only, we assume a vertical salinity profile $S^i_{l,k}$, with a shape that depends on $\overline{S}^i_l$.

Vertically-averaged ("bulk") salinity variations

The representation of brine entrapment and drainage is based on a simplification of the more complex Vancoppenolle et al. [2007] model. Field data [e.g. Kovacs 1996] and 1-D model simulations [Vancoppenolle et al. 2006] indicate that the sea ice desalination can be divided into initial, winter and summer stages. The processes of gravity drainage and flushing dominate winter and summer desalinations, respectively [see Notz and Worster 2009, for a review]. In addition, salinity data from ice cores drilled in some of the Southern Ocean deep snow-covered regions suggest that salt concentration occurs at the ice surface during snow ice formation [Jeffries et al. 1997].

Therefore, for each ice thickness category (we omit the category index l), the change in bulk salinity $\overline{S}^i$ due to brine drainage is formulated as:

$$\Delta\overline{S}^i = (S^b - \overline{S}^i)\frac{\Delta h^{i,b+}}{h^i} + (S^{si} - \overline{S}^i)\frac{\Delta h^{i,si}}{h^i} - \left(\frac{\overline{S}^i - S^G}{T^G}\right) I^G \Delta t - \left(\frac{\overline{S}^i - S^F}{T^F}\right) I^F \Delta t. \tag{3.79}$$

The terms on the right-hand side refer to basal ice formation, snow ice formation, gravity drainage and flushing, respectively. The salinity of new ice is simply formulated as proportional to sea surface salinity: $S^b = \nu S^w$. ν is a fractionation coefficient which depends on the basal ice growth rate [Cox and Weeks 1988]:

$$\nu = \begin{cases} 0.12 & \text{if } \frac{\Delta h^{i,b+}}{\Delta t} < 2 \cdot 10^{-8} \;, \\ 0.8925 + 0.0568 \text{log } (100 \cdot \frac{\Delta h^{i,b+}}{\Delta t}) & \text{if } 2 \cdot 10^{-8} \leq \frac{\Delta h^{i,b+}}{\Delta t} < 3.6 \cdot 10^{-7} \;, \\ 0.26/[0.26 + 0.74 \text{exp}(-7.243 \cdot 10^{6} \frac{\Delta h^{i,b+}}{\Delta t})] & \text{if } \frac{\Delta h^{i,b+}}{\Delta t} > 3.6 \cdot 10^{-7} \;. \end{cases} \tag{3.80}$$

The salinity of snow ice is a weighted mean of snow and seawater contributions:

$$S^{si} = (\rho_i - \rho_s)/\rho_i S^w. \tag{3.81}$$

T^G and S^G (T^F and S^F) are the winter (summer) desalination time scales and equilibrium salinities. In *namelist_ice*, they are referred to as *time_G*, *sal_G*, *time_F* and *sal_F*. $I^G = 1$ if the difference between surface and bottom ice temperatures is negative and 0 otherwise. This corresponds to a temperature gradient directed downwards, which, since brine salinity increases toward low temperatures, leads to an unstable brine density gradient in the brine network and a subsequent desalination [Cox and Weeks 1988]. $I^F = 1$ if surface melts and 0 otherwise.

Vertical salinity profile

We assume that salinity has a vertical distribution S_k^i only for vertical thermodynamic computations. Observations from ice cores suggest that the shape of the salinity profile in sea ice depends on bulk salinity. This because the strongest changes in profile shape are due to flushing, which affects both the mean salinity and profile shape [Vancoppenolle et al. 2007]. Numerical experiments have shown that a linear profile well approximates the sea ice salinity profile [Vancoppenolle et al. 2005]. Therefore, we parameterize the shape of the salinity profile as follows. At high mean salinity, i.e., if $\overline{S}^i > S_2 = 4.5$ ‰, the profile is isosaline: $S_k^i = S_\infty(z) = \overline{S}^i$. At low mean salinity, i.e., if $\overline{S}^i < S_1 = 3.5$ ‰, the profile is linear: $S_k^i = S_0(z)$. This profile is determined by applying two constraints. First, its mean salinity should be equal to $\overline{S}^i$ (given by 3.79), and second the salinity should be zero at the surface. Therefore,

$$S_0(z) = 2\frac{\overline{S}^i}{h^i}z. \tag{3.82}$$

At intermediate salinities, i.e., between S_1 and S_2, S_k^i is a combination of S_0 and S_∞. In summary,

$$S_k^i(z) = \alpha^S(\overline{S}^i)S_0(z_k) + [1 - \alpha^S(\overline{S}^i)]S_\infty(z_k), \tag{3.83}$$

where α^S is:

$$\alpha^S(\overline{S}^i) = \begin{cases} 1 & \text{if } \overline{S}^i < S_1 \\ (\overline{S}^i - S_2)/(S_1 - S_2) & \text{if } S_1 \le \overline{S}^i \le S_2 \\ 0 & \text{if } \overline{S}^i > S_2 \;. \end{cases} \tag{3.84}$$

In *namelist_ice*, S_1 and S_2 are referred to as *s_i_0* and *s_i_1*, respectively. Local thermodynamic equilibrium is assumed during brine drainage. Therefore, an increase in temperature compensates any desalination.

3.6 Ice age.

For diagnostic purposes, the ice age o_m^i of each thickness category is computed. Utilisation of age in marine modelling has shown that the values and interpretation of model age are strongly dependent on the age definition [see, e.g., Deleersnijder et al. 2001]. We assume that the age is associated with the areal sea ice age content $O_l = o_l^i g_l^i$, which verifies:

$$\frac{\partial O_l}{\partial t} = -\nabla.(O_l \mathbf{u}) + \Theta_l^O + \Psi_l^O. \tag{3.85}$$

For coherence, first, the mechanical redistribution function associated to ice age Ψ^O is constructed based on Ψ^g (see earlier). Second, in contrast to Harder and Lemke [1994], vertical growth and melt do not affect the ice age (i.e., vertically accreted new ice is assumed to have the age of existing ice). Nevertheless, new ice formed in open water has an age equal to zero. Therefore, our value reflects an areal residence time, larger than the actual ice age, and the thermodynamic ageing term reads:

$$\Theta_l^O = g_l^i - \frac{\partial (f_l O_l)}{\partial h}, \tag{3.86}$$

where f_l is the sea ice growth/melt rate in the l^{th} category. The first term on the right-hand side represents ice ageing. The second term accounts for the formation of new ice in open water and for the transport in thickness space of the age content due to ice growth and melt.

3.7 Transport in thickness space

Modules: ***limitd_th.F90*** *Subroutines*: **lim_itd_th**, **lim_itd_th_rem**, **lim_itd_fitline**, **lim_itd_shiftice**, **lim_itd_th_reb**.

This section focuses on how LIM computes transport in thickness space

$$\frac{\partial X}{\partial t} = -\nabla.(\mathbf{u}X) + D\nabla^2 X + \Psi^X + \Theta^X - \frac{\partial}{\partial h}(fX), \tag{3.87}$$

3.7.1 Introduction to transport in thickness space

Let us first explain what transport in thickness space is. To do this, we introduce the cumulative thickness distribution $G = \int_0^h g(h)dh$, which helps to formulate the problem. When ice thickness h changes due to growth and melt at a rate f, all of the ice which is thinner than h at a time t will be thinner than $h + fdt$ at time $t + dt$. Thus

$$G(h,t) = G(h + fdt, t + dt) = G(h, t + dt) + fdt\frac{\partial G}{\partial h} + \text{terms in } dt^2, \tag{3.88}$$

which can be rearranged to show that $\partial G/\partial t = -f\partial G/\partial h$. Differentiating with respect to h gives $\partial g/\partial t = -\partial/\partial h(fg)$. Hence, in the absence of ice motion and deformation, the conservation of area can be written as follows [Thorndike et al. 1975]:

$$\frac{\partial g}{\partial t} = -\frac{\partial(fg)}{\partial h}. \tag{3.89}$$

This equation is analogous to a one-dimensional continuity equation where g is equivalent to density. $f = dh/dt$ [m/s], the rate of change of ice thickness is an equivalent for velocity in thickness space. In this context, the first part of the problem is to determine f, i.e., how any given ice thickness evolves with time, which is approximated using the thermodynamic component. The second part of the problem, the *transport in thickness space*, is to determine how the thickness distribution responds to changes in ice thickness, i.e., to advect g in thickness space given a velocity field $f(h)$. Because h is an independent variable, the other state variables $X(t, \boldsymbol{x}, h)$ have also to be transported in thickness space:

$$\frac{\partial X}{\partial t} = -\frac{\partial(fX)}{\partial h} \tag{3.90}$$

In order to numerically resolve this, we use the *linear remapping* scheme of Lipscomb [2001]. This is a second-order, semi-lagrangian scheme, which is less diffusive and converges faster than other ones. The scheme is valid as long as the velocities in thickness space are not too large. In this case, following [Bitz et al. 2001], we use a representation in terms of delta functions.

3.7.2 Resolution in standard case: *Linear remapping*

Let us consider the thickness distribution discretization presented in Section 3.1. The ice categories are considered as lagrangian cells moving in thickness space. The scheme is based on three steps: (i) displacement of the category boundaries; (ii) approximation of the thickness distribution on the displaced categories; (iii) restoration of the original boundaries.

At time n, each category l is covered by an ice area $g^i_{l,n}$ of thickness $h^i_{l,n}$. Using the thermodynamic component described in the previous section, the new thicknesses $h^i_{l,n+1}$ are computed. The ice growth rate in category l is $f_l = (h^i_{l,n+1} - h^i_{l,n})/\Delta t$. The ice growth rate at the category boundaries F_l is linearly interpolated:

$$F_l = f_l + \frac{f_{l+1} - f_l}{h^i_{l+1} - h^i_l}(H_l - h^i_l). \tag{3.91}$$

If both adjacent categories have $g^i_m = 0$, then F^l is chosen to be zero. If one of the adjacent categories has no ice, then we assign to F_l the value of the non zero category. At time $n+1$, the category boundaries are: $H^\star_l = H_l + F_l \Delta t$. The area in the displaced categories is conserved, such that $g^{\star i}_{l,n+1} = g^i_{l,n}$, while the volume is $v^{\star i}_{l,n+1} = g^{\star i}_{l,n} h^i_{l,n+1}$. To work properly, $H_{l-1} < H^\star_l < H_{l+1}$ must be verified. Let us replace $H^\star_{l-1} = H_L$ and $H_l = H_R$.

To compute how much ice is transferred between categories, we must build an approximation for $g(h)$ in each displaced category, verifying:

$$g^{\star i}_{l,n+1} = \int_{H_L}^{H^R} g(h)dh, \tag{3.92}$$

$$v^{\star i}_{l,n+1} = \int_{H_L}^{H^R} g(h)hdh. \tag{3.93}$$

A linear polynomial approximation is suitable: $g(\eta) = g_0 + g_1\eta$, where $\eta = h - H_L$. This function, computed by **lim_itd_fitline**, takes 3 forms, whether $h^i_{l,m+1}$ is in the first, the second or the third part of the interval $[H_L, H_R]$.

Once we constructed the displaced thickness distribution, we can remap the thickness distribution to the original boundaries. To do this, in the case of a transfer from l to $l+1$ categories, we have to integrate $g(h)$ between H_l and $H^\star_l$:

$$\Delta g^i_l = \int_{H_l}^{H^\star_l} g(h)dh, \tag{3.94}$$

$$\Delta v^i_l = \int_{H_l}^{H^\star_l} hg(h)dh. \tag{3.95}$$

The change in other state variables due to transport in thickness space is $X\Delta v_l^i/v_l^i$ if the variable follows ice volume and $X\Delta g_l^i/g_l^i$ if the variable follows ice concentration. This is done by routine **lim_itd_shiftice**.

3.7.3 New ice formation

Modules: ***limthd_lac.F90***

Subroutines: **lim_thd_lac**

By *new ice formation*, we refer to ice which is formed in ice-free ocean cells, or in the open water fraction of ice-covered grid cells. If seawater in the top oceanic layer is at its freezing point and if the surface is losing heat, new ice forms. Ice forming in open water is assumed to form with a given cutoff thickness h^0 and a volume given by:

$$\Delta v^{i,0} = -\frac{F^{ow}\Delta t}{q(S^0, T^w)}, \tag{3.96}$$

where F^{ow} is the heat loss in open water areas and S^0 is the salinity of new ice. h^0 represents the thickness of new ice formed during a time step. In the standard version, h^0 is prescribed using a namelist parameter (*hiccrit*). Finally, the fractional area of new ice is diagnosed using the relation $\Delta A^0 = \Delta v^{i,0}/h^0$. This area of new ice is put into the category where h^0 falls into. If the resulting total ice concentration is greater than one, the volume in excess is kept into memory and added at the base of each category. All other state variables have to be updated following new ice formation. The salinity of new ice S^0 is formulated as a function of ice thickness, following the observation-based regression of Kovacs [1996]

$$S^0 = 4.606 + \frac{0.91603}{h^0}. \tag{3.97}$$

More on h^0 should come here once we are done with pancakes.

3.8 Coupling with the ocean

Review this section to be closer to the code

The ice-ocean coupling is conceptually similar to Goosse and Fichefet [1999]. Ice and ocean exchange momentum through an ice-ocean stress deriving from a quadratic bulk formula assuming an oceanic drag coefficient of $C_w = 5.0 \times 10^{-3}$.

The oceanic heat flux, function of the temperature of the ocean and turbulent mixing, derives from McPhee [1992] and Goosse and Fichefet [1999]:

$$F_w = \rho_w c_w C_h u^*_{io}(T^w - T^{ml}), \tag{3.98}$$

where c_w is the seawater specific heat, $C_h = 0.006$ is a heat transfer coefficient [McPhee 1992], T^{ml} is the mixed layer temperature and u^*_{io} is the friction velocity, which is a function of the ice-ocean velocity difference.

Freshwater fluxes for ocean and sea ice use the approach of [Tartinville et al. 2001]. Adaptations were made to account for time-varying sea ice salinity. From Tartinville et al. [2001], we adopt the following convention. First, if liquid or solid water is added to or extracted from the ice-ocean system (e.g., through precipitation or evaporation), then it is introduced as a freshwater flux. Second, if a surface process affects ocean salinity without a net gain or loss of water for the ice-ocean system (e.g., through ice melting or freezing) this internal exchange is transformed into an equivalent salt flux that is applied to the ocean. In this approach, sea ice acts as a negative reservoir of salt inside the ice-ocean system. As ice is formed, salt is released into the ocean. As ice melts, salt is taken out of the ocean.

The freshwater flux at the surface of the ocean F^f (in kg/m^2/s) is given by :

$$F^f = P - E - \frac{\rho_s}{\sum_{m=1}}^{M} \frac{\partial v_m^{s,-}}{\partial t}, \tag{3.99}$$

where P is the precipitation, E the evaporation, $\partial v_m^{s,-}/\partial t$ is the snow volume loss per unit area in the m_{th} category due to snow melt and snow ice formation. The salt flux F^s reads :

$$F^s = F^s_b + F^s_{eq}. \tag{3.100}$$

F^s_b is the salt flux due to brine drainage :

$$F^s_b = \rho_i \sum_{m=1}^{M} v^i_m \left(\frac{\partial S^i_m}{\partial t} \bigg|_G + \frac{\partial S^i_m}{\partial t} \bigg|_F \right). \tag{3.101}$$

The partial derivatives on the right-hand side refer to the loss of salt by gravity drainage and flushing (next-to-last and last terms of (3.79)), respectively. F^s_{eq} is the equivalent salt flux due to ice growth and melt:

$$\begin{aligned} F^s_{eq} = \rho_i (S^w - S^0) \tfrac{\partial v^{i,n}}{\partial t} + \rho_i \textstyle\sum_{m=1}^M g^i_m [(S^w - S^{i,b}_m) \tfrac{\partial h_m^{i,b+}}{\partial t} \\ +(S^w - S^{si}_m) \tfrac{\partial h_m^{i,si}}{\partial t} + (S^w - S^i_m) \tfrac{\partial h_m^{i,-}}{\partial t}] \end{aligned} \tag{3.102}$$

The terms on the right-hand side respectively correspond to the salt rejected during new ice formation in open water, to bottom congelation growth, to snow ice formation and to the melt of saline ice. For the definitions of ice salinities, see Sec. 3.5.2 and 3.7.3.

4 Technical aspects

Here, some of the technical features of the code are described. First, we describe the *namelist_ice* file. Then we review the code modules and routines. As environment and compilation are dealt with within the NEMO code itself, we report the reader interested in the code compilation to the NEMO documentation and web page.

4.1 Physical constants

The sea ice related physical constants are handled in the NEMO routine *phycst.F90* (Table 4.1).

Table 4.1: Physical constants of the model.

Physical constant	Symbol	Variable name	Value	Units	Routine
Melting point of snow	T^m	*rt0_snow*	273.16	K	*phycst.F90*
Melting point of ice	T^m	*rt0_ice*	273.16	K	*phycst.F90*
Snow thermal conductivity	k^s	*rcdsn*	0.3	W/m/K	*phycst.F90*
Pure ice thermal conductivity	k^i	*rcdic*	2.034396	W/m/K	*phycst.F90*
Pure ice specific heat	c^0	*cpic*	2067	J/kg/K	*phycst.F90*
Latent heat of sublimation of ice	L^{sub}	*lsub*	2.834×10^6	J/kg	*phycst.F90*
Latent heat of fusion of ice	L^0	*lfus*	0.334×10^6	J/kg	*phycst.F90*
Sea ice density	ρ^i	*rhoic*	917	kg/m^3	*phycst.F90*
Snow density	ρ^s	*rhosn*	330	kg/m^3	*phycst.F90*
Sal. dependence of melting point	μ	*tmut*	0.054	°C/‰	*phycst.F90*
Surface emissivity	ϵ	*emic*	0.97	-	*phycst.F90*

4.2 Namelist parameters

Namelists parameters are reviewed in Tables 4.2, 4.3, 4.4 and 4.5. There are two additional namelist sections that are not described formally: (i) *nameicedia* corresponds to text-formatted ice global diagnostics (*evolu*) file; (ii) *nameiceout* parameterizes which variables are saved. The frequency of writing in the output file is handled in the NEMO namelist. Some parameters are obsolete and will be removed in future updates of the code.

Table 4.2: Namelist parameters (1): Shared parameters and initialization. NH (SH) stands for Northern (Southern) hemisphere.

Shared parameters	*nameicerun*		
ln_limdyn	*.TRUE.*	Ice dynamics switch	*ln_limdyn.F90*
acrit	$1 \times 10^{-2}, 1 \times 10^{-2}$	Min. open water frac. (NH, SH)	obsolete
hsndif	0. (=yes)	Compute snow temp. or not	obsolete
hicdif	0. (=yes)	Compute ice temp. or not	obsolete
cai	$1.4{\times}10^{-3}$	Atm. drag coefficient over ice	*sbcblk_clio.F90*
cao	$1.0{\times}10^{-3}$	Atm. drag coefficient over ocean	*sbcblk_clio.F90*

Ice initialization	*nameiceini*		
ttest	2	SST parameter for ice presence	*limistate.F90*
hninn	0.3	Initial snow depth, NH	*limistate.F90*
hnins	0.1	Initial snow depth, SH	*limistate.F90*
hginn	3.5	Initial ice thickness, NH	*limistate.F90*
hgins	1.0	Initial ice thickness, SH	*limistate.F90*
aginn	1.0	Initial ice concentration, NH	*limistate.F90*
agins	1.0	Initial ice concentration, SH	*limistate.F90*
sinn	6.3	Initial ice salinity, NH	*limistate.F90*
sins	6.3	Initial ice salinity, SH	*limistate.F90*

Table 4.3: Namelist parameters (2): Ice dynamics and transport.

Ice dynamics	*nameicedyn*		
epsd	$1\ \times10^{-20}$	Tolerance parameter	*lim_rhg.F90*
alpha	0.5	Coefficient for semi-implicit Coriolis	obsolete
dm	600	Diffusion constant for dynamics	obsolete
nbiter	1	Max. num. of iterations for relaxation	obsolete
nbitdr	500	Max. num. of iterations for relaxation	obsolete
om	0.5	Relaxation constant	*lim_hdf.F90*
resl	0.6×10^{3}	Max. val. for residual of relaxation	obsolete
cw	0.6×10^{-3}	Ocean-ice drag coefficient	*lim_dyn.F90*
angvg	0.	Turning angle for oceanic stress	*lim_dyn.F90*
pstar	4×10^{4}	Ice strength	*lim_itd_me.F90*
c_rhg	20	Exponential decay of ice strength	*lim_itd_me.F90*
etamn	0	Minimum viscosity	obsolete
creepl	0	Creep limit	*lim_rhg.F90*
ecc	2	Eccentricity of the elliptical yield curve	*lim_rhg.F90*
ahi0	350	Horizontal diffusivity for sea ice	*lim_trp.F90*
nevp	300	Number of EVP subcyclings	*lim_rhg.F90*
telast	9600	Timescale for elastic waves	*lim_rhg.F90*
alphaevp	1	Coeff. for semi-implicit stress	*lim_rhg.F90*

Ice transport	*nameicetrp*		
bound	0	Boundary condition on ice velocity no-slip [0], free-slip [1]	obsolete

Table 4.4: Namelist parameters (3): Ice thermodynamics.

Ice thermodynamics	*nameicethd*		
hmelt	-0.15	Max. melt at ice base (L=1 only)	*limthd_dh.F90*
hiccrit	0.1, 0.1	Thickness of newly formed ice (NH, SH).	*lim_thd_lac.F90*
fraz_swi	0	Use frazil ice collection thickness [1]	*lim_thd_lac.F90*
maxfrazb	0	Max. vol. fraction of basal frazil accretion	*lim_thd_lac.F90*
Cfrazb	5	Squeezing coeff. for basal frazil accretion	*lim_thd_lac.F90*
hicmin	0.2	Ice thick. corr. to max. en. stored in brines	obsolete
hiclim	0.1	Minimum ice thickness	*limupdate.F90*
amax	0.999	Maximum lead fraction	*limupdate.F90*
sbeta	1.	Crank-Nicholson parameter (heat diffusion)	obsolete
parlat	1.	Energy frac. used for lateral ablation	obsolete
hakspl	0.25	Slope of distr. Hakkinen and Mellor	obsolete
hibspl	0.25	Slope of distr. Hibler	obsolete
exld	2.0	Exponent for lead-closure rate	obsolete
hakdif	1.0	coefficient for diffusions of ice and snow	obsolete
thth	0.2	threshold computation for comp of th. cond.	obsolete
hnzst	0.1	thickness of the surface layer	obsolete
parsub	1.0	Sublimation or not	*lim_thd_lac.F90*
alphs	1.0	Snow density coeff. in snow ice	obsolete
betas	0.6	Open water partition coeff. for snow fall	*lim_thd_dh.F90*
kappa_i	1.0	Extinction coeff. for radiation in sea ice	*lim_thd_dif.F90*
nconv_i_thd	50	Max. num. of iter. for heat diffusion	*lim_thd_dif.F90*
maxer_i_thd	0.0001	Max. temp. err. after heat diffusion	*lim_thd_dif.F90*
thcon_i_swi	1	Ice thermal cond^{ty}: Untersteiner [1964] [0] or Pringle et al. [2007] [1]	*lim_thd_dif.F90*

Ice salinity	*nameicesal*		
num_sal	2	Ice salinity: $S = cst$ [1], $S = S(z,t)$ [2],	*lim_thd_sal.F90*
		$S = S(z)$ [3]; $S = S(t)$ [4]; $S = S(h^i)$ [5]	*lim_thd_sal.F90*
bulk_sal	4	Vert.-av. bulk ice salinity (*num_sal*=1)	*lim_thd_sal.F90*
sal_G	5	Restoring salinity (gravity drainage)	*lim_thd_sal.F90*
time_G	1.728×10^6	Salinity e-fold time scale (gravity drainage)	*lim_thd_sal.F90*
sal_F	2	Restoring salinity (flushing)	*lim_thd_sal.F90*
time_F	8.64×10^5	Salinity e-fold time scale (flushing)	*lim_thd_sal.F90*
s_i_max	20	Maximum salinity	*lim_thd_sal.F90*
s_i_min	0.1	Minimum salinity	*lim_thd_sal.F90*
s_i_0	3.5	1st salinity profile parameter	*lim_thd_sal.F90*
s_i_1	4.5	2nd salinity profile parameter	*lim_thd_sal.F90*

Table 4.5: Namelist parameters (4): Mechanical redistribution.

Mechanical redistribution	*nameiceitdme*		
ridge_scheme_swi	0	Ice strength: Rothrock [1975] [1] or Hibler [1979] [≠1]	*lim_itd_me.F90*
Cs	0.5	Frac. of shearing energy being dissipated	*lim_itd_me.F90*
Cf	17	Ratio (ridging work) / (increase in pot. energy)	*lim_itd_me.F90*
fsnowrdg	0.5	Snow volume fraction surviving ridging	*lim_itd_me.F90*
fsnowrft	0.5	Snow volume fraction surviving rafting	*lim_itd_me.F90*
Gstar	0.15	Relative area of thin ice being deformed	*lim_itd_me.F90*
astar	0.05	Participation parameter	*lim_itd_me.F90*
Hstar	100	Determines maximum ridged ice thick.	*lim_itd_me.F90*
raftswi	1	Rafting or not	*lim_itd_me.F90*
hparmeter	0.75	Threshold thickness for rafting	*lim_itd_me.F90*
Craft	5	Squeezing coefficient for rafting	*lim_itd_me.F90*
ridge_por	0.3	Initial ridge porosity	*lim_itd_me.F90*
sal_max_ridge	15	Maximum ridged ice salinity	*lim_itd_me.F90*
partfun_swi	1	Linear (0) or exp. (1) participation func.	*lim_itd_me.F90*
transfun_swi	1	Uniform (0) or exp. (1) transfer func.	obsolete
brinstren_swi	0	Use brine volume fraction in ice strength	*lim_itd_me.F90*

4.3 Index of modules and routines

Table 4.6: List of modules in LIM3.

Main routine	
sbcice_lim.F90	main sea ice routine
Tool routines	
ice.F90	defines shared ice variables
par_ice.F90	sea ice parameters
iceini.F90	initializes sea ice code
limistate.F90	initializes sea ice variables
dom_ice.F90	defines domain variables
limmsh.F90	definition of the mesh parameters
limvar.F90	performs variables conversions
limcons.F90	some conservation checks
Ice dynamics	
limdyn.F90	resolves ice motion equation
limrhg.F90	sea ice rheology
limtrp.F90	transport extensive (global) ice variables
limadv.F90	advection routines
limhdf.F90	horizontal numerical diffusion
Ice thermodynamics	
thd_ice.F90	thermodynamic ice variables
limthd.F90	main thermodynamic routine
limthd_dif.F90	heat diffusion in sea ice
limthd_dh.F90	vertical accretion / ablation
limthd_sal.F90	sea ice salinity
limthd_ent.F90	redistributes heat content vertically
limthd_lac.F90	growth of new ice (=lateral accretion)
limtab.F90	reforms arrays (1D-2D) for ice thermodynamics
Ice thickness distribution	
limitd_th.F90	thermodynamic transport in thickness space
*limitd_me.F90**	mechanical redistribution
Update and ice-ocean fluxes	
*limupdate.F90**	updates global ice variables
*limsbc.F90**	ice-ocean heat and salt fluxes
Inputs and outputs	
limrst_dimg.h90	binary restart (not working)
limwri_dimg.h90	binary write (not working)
limdia.F90	on-line global diagnostics (volume, extent, ...)
limrst.F90	netcdf restart
limwri.F90	netcdf output

*=needs cleaning

sbcice_lim.F90.

Goal: Main sea ice module.
Subroutines: **sbc_ice_lim**, **lim_ctl**, **lim_prt_state**.

⋆ **sbc_ice_lim**
Goal: Main sea ice routine.
Called by: **sbc** (*sbcmod.F90*); itself called in **stp** (*step.F90*); itself called in **opa_model** (*opa.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Initialize the ice (**call ice_init**).
- Get ocean velocities and freezing temperature from the coupler.
- Compute sea ice albedo (**call albedo_ice**).
- Mask surface temperature.
- Call bulk atmospheric formulae (**call blk_ice_clio**).
- Put old variables, trend terms, fluxes, diagnostics to zero.
- **call lim_rst_opn** (reads restart).
- **call lim_dyn** (ice dynamics).
- Aggregate and convert variables (**call lim_var_agg** & **lim_var_glo2eqv**).
- **call lim_itd_me** (mechanical redistribution).
- Aggregate and convert variables (**call lim_var_agg** & **lim_var_glo2eqv**).
- **call lim_thd** (ice thermodynamics)
- Age the ice.
- Convert variables (**call lim_var_glo2eqv**).
- **call lim_itd_th** (transport in thickness space by thermodynamics)
- Aggregate and convert variables (**call lim_var_agg** & **lim_var_glo2eqv**).
- **call lim_update** (update all ice variables).

- Aggregate and convert variables (**call lim_var_agg** & **lim_var_glo2eqv**).
- **call lim_sbc_flx** (fluxes of mass and heat to the ocean).
- **call lim_sbc_tau** (ice-ocean stress to the ocean).
- Diagnostics and outputs.

⋆ **lim_ctl**

Goal: Write alerts in *ocean.output* if ln_nicep is .TRUE.
Called by: in every routine.
Major inputs:
Major outputs:
Remarks:
Structure:
Alerts are (2) incompatible volume and concentration, (3) very thick ice, (4) very fast ice, (6) ice on continents, (7) very fresh ice, (9) very old ice, (5) high salt flux, (8) very high non solar flux, (10) very warm ice. Tests all points and outputs where the routine thinks it is wrong. At the end, counds the number of times the alerte has been triggered in the time step. With a grep ”ALERT” on ocean.output, all alerts go out at the same time.

⋆ **lim_prt_state**

Goal: Performs control prints on a given point with coordinates jiindx, jjindx. Typically activated if ln_nicep is .TRUE.
Called by: in every routine.
Major inputs:
Major outputs:
Remarks:
Structure: There are three options: (1) simple ice state, (2) exhaustive state, (3) ice/ocean salt fluxes.

dom_ice.F90

Goal: Defines coriolis factors, surface area of the grid cell and masks.
Subroutines: none.

ice.F90

Goal: Defines global sea ice variables, parameters and diagnostics.
Subroutines: **none**.

iceini.F90

Goal: Model initialization.
Subroutines: **ice_init**, **ice_run**, **lim_itd_ini**.

⋆ **ice_init**
Goal: Model initialization.
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Open and read some namelists (**call ice_run**).
- Mesh initialization (**call ice_msh**).
- Ice thickness distribution initalization (**call lim_itd_ini**)
- Initialize variables (**call lim_istate** or **call lim_rst_read**), convert variables and aggregate.
- initialize sea ice fraction, deal with time step and indices of iterations.

⋆ **ice_run**
Goal: Reads namelist (nameicerun).
Called by: **ice_init** (*iceini.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

⋆ **lim_itd_ini**
Goal: Initializes numerical bounds of the thickness distribution.
Called by: **ice_init** (*iceini.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

limadv.F90

Goal: Ice advection [Prather 1986].

Subroutines: **lim_adv_x**, **lim_adv_y**.

⋆ **lim_adv_x** (**lim_adv_y**)
Goal: Advection in the x (y) direction.
Called by: **lim_trp** (*limtrp.F90*).
Major inputs: zonal (meridional) velocity, area of the grid cell, field to be advected, first-order moments (x,y), second-order moments (xx,yy,xy).
Major outputs: New field.
Remarks:
Structure:

- Limitation of moments
- Initialize areas
- Calculate fluxes and moments between boxes i and i+1 (cases u >0 and $u < 0$)
- Lateral boundary conditions

limcons.F90

Goal: Check energy conservation.
Subroutines: **lim_column_sum**, **lim_column_sum_energy**, **lim_cons_check**.

limdia.F90

Goal: Write global hemispheric diagnostics in ice.evolu.
Subroutines: **lim_dia**, **lim_dia_init**.

⋆ **lim_dia**
Goal: Write diagnostics in evolu file.
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:
Remarks:
Structure: Computes ice area, extent, volume, snow volume, mean age, mean salinity, salt flux, brine drainage flux, equivalent salt flux, mean sst, mean sss, mean snow temperature, ice heat content, ice volumes in each categories, mass flux, volume and area export through Fram Strait.

limdyn.F90

Goal: Ice dynamics.

Subroutines: **lim_dyn**, **lim_dyn_init**.

⋆ **lim_dyn**

Goal: Compute ice velocity vectors (u_ice, v_ice).
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs: u_ice, v_ice
Remarks:
Structure:

- **call lim_dyn_init** (initialization)
- Define the limits where the rheology is computed
- **call lim_rhg** (ice rheology)
- Computation of friction velocity (ust2s) required in lim_thd for computation of oceanic heat flux

⋆ **lim_dyn_init**

Goal: Reads ice dynamics namelist.
Called by: **lim_dyn** (*limdyn.F90*).

limhdf.F90

Goal: Horizontal diffusion of global ice variables.
Subroutines: **lim_hdf**.

⋆ **lim_hdf**

Goal: Horizontal diffusion of global ice variables. This stabilizes the Prather [1986] scheme.
Called by: **lim_trp** (*limtrp.F90*).
Major inputs: field to be diffused (called 'ptab')
Major outputs: new value of the diffused field ('ptab')
Remarks:
Structure:

- Initialization.
- Metric coefficient.
- Sub-time step loop to compute diffusive fluxes in zonal and meridional directions. Includes a convergence test.

limistate.F90

Goal: Initialize global sea ice variables.
Subroutines: **lim_istate**, **lim_istate_init**.

⋆ **lim_istate**

Goal: Fill global ice variables array with prescribed initial values read in the namelists.
Called by: **ice_init** (*ice_init.F90*).
Major inputs:
Major outputs: Global variables filled with inital values and masked. Structure:

- Initialize t_bo.
- Compute dummy factors (zgfactorn, zgfactors).
- Initialize area and thickness (cases $jpl = 1$, $jpl > 1$).
- Initialize snow depth, ice salinity, ice age, sea ice surface temperature, snow temperature and heat content, ice salinity, temperature and heat content. All this is done in 2 steps, one for each hemisphere.
- Set some global variables to zero ($fsbbq, u\_ice, v\_ice$, stresses, moments for advection).
- Apply lateral boundary conditions.

⋆ **lim_istate_init**

Goal: Reads ice initialization namelist.
Called by: **lim_istate** (*limistate.F90*).

limitd_me.F90

Goal: Redistributes sea ice mechanically due to ridging and rafting.
Subroutines: **lim_itd_me**, **lim_itd_me_icestrength**, **lim_itd_me_ridgeprep**, **lim_itd_me_ridgeshift**, **lim_itd_me_asumr**, **lim_itd_me_init**, **lim_itd_me_zapsmall**.

⋆ **lim_itd_me**

Goal: Computes the mechanical redistribution of ice thickness.
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:

Remarks: Needs testing (conservation) and fixing some parts probably, like the trend terms and the pathological case in the end.
Structure:

- **call lim_itd_me_init** (initialization)
- Specify an upper limit to ridging, compute the proportionality constant for potential energy
- **call lim_itd_ridgeprep** (prepare ridging)
- Set some arrays to zero
- Dynamical inputs ($closing\_net$, $divu\_adv$, $opning$)
- ***Ridging iteration***
  - closing gross rate
  - **call lim_itd_me_ridgeshift** (redistribute area, volume, energy, salt).
  - **call lim_itd_me_asumr** (compute ice + open water area after ridging).
  - Test to whether iteration should be continued (once $asum$ converges to 1).
- Ridging diagnostics.
- Heat and freshwater salt fluxes.
- Update state variables and trend terms (includes a call to lim_itd_zapsmall).
- Back to initial values.
- Pathological case (advection in an ice-free cell, newly ridged ice).

⋆ **lim_itd_me_icestrength**
Goal: Computes ice strength under the method of Rothrock [1975] or Hibler [1979].
Called by: **lim_itd_me** (*limitd_me.F90*).
Major inputs:
Major outputs: Ice strength arrray ($strength(ji, jj)$)
Remarks: Needs cleaning and debugging. (i) Presently, only Hibler's method is stable. Hence the question arises whether we should keep Rothrock's method in the code. (ii) In addition, the call to ridgeprep should be included in the IF that follows. (iii) There is also an option of the impact of brine volume on ice strength

that is not really active. We should choose what to do with it if we keep it or not. (iv) Finally, there is a not very clean smoothing option for ice strength in there that should be assessed.
Structure:

- **call lim_itd_me_ridgeprep** (compute ITD of ridging and ridged ice).
- Compute strength using Rothrock [1975].
- Compute strength using Hibler [1979].
- Smooth ice strength spatially and or temporally.
- Apply lateral boundary condition to ice strength.

⋆ **lim_itd_me_ridgeprep**
Goal: Compute the thickness distribution of the ice and open water participating in ridging and of the resulting ridges.
Called by: **lim_itd_me** (2x), **lim_itd_me_icestrength** (*limitd_me.F90*).
Major inputs: Participation functions (), transfer functions ()
Major outputs:
Remarks:
Structure:

- Computation of participation functions
  - Cumulative ice thickness distribution ($Gsum$)
  - Participation function ($athorn(ji,jj) = b(h).g(h)$). If $krdg\_index$=0, $athorn$ is computed as a linear function of $Gsum$. In the case $krdg\_index = 1$, $athorn$ has an exponential dependence on $Gsum$.
  - Participation function is then divided into ridging and rafting contributions ($aridge$ and $araft$)
- Computation of transfer functions characteristics: minimum and maximum thickness of ridges, rafted ice thickness ($hrmin$, $hrmax$, $hraft$, $krdg$); $aksum$ (normalization factor).

⋆ **lim_itd_me_ridgeshift**
Goal: Shift global state variables from the giving ice categories to the receiving ones, due to ridging and rafting.
Called by: **lim_itd_me**, (*limitd_me.F90*).
Major inputs: Participation functions (), transfer functions ()
Major outputs:
Remarks:
Structure:

- Compte change in open water area due to closing and opening
- Save initial state variables
- Remove ice from ridging and rafting categories (participation)
  - Identify grid celss with nonzero ridging
  - Compute area of ridging ice (*ardg1*) and of new ridges (*ardg2*)
  - Compute ridging / rafting fractions, make sure it is smaller than 1.
  - Substract area, volume and energy from ridging / rifting categories
  - Compute properties of new ridges
  - Increment ridging diagnostics
  - Put the removed snow into the ocean
  - Compute ridging ice enthalpy, remove it from ridging ice and compute ridged ice enthalpy
- Add area, volume and energy of new ridges to each ridged category

⋆ **lim_itd_me_ridgeshift**
Goal: Compute the total area of ice plus open water in each grid cell.
Called by: **lim_itd_me**, (*limitd_me.F90*).
Major inputs: open water area, ice concentrations
Major outputs: total ice + open water area
Remarks: total ice + open water area can temporarily be > 1.
Structure:

- Add open water
- Add ice concentrations in each category

⋆ **lim_itd_me_init**
Goal: Initialize physical constant and paramters related to the mechanical ice redistribution.
Called by: **lim_itd_me**, (*limitd_me.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Reads namelist section *nameiceitdme*

⋆ **lim_itd_me_zapsmall**
Goal: Remove the excessively small ice concentrations and correct ice ocean salt fluxes
Called by: **lim_itd_me**, (*limitd_me.F90*).
Major inputs:
Major outputs:
Remarks: Needs cleaning and check .
Structure:

- Count categories that have to be zapped
- Remove ice energy and use ocean heat to melt ice
- Zap ice and snow volume, add water and salt to the ocean

limitd_th.F90

Goal: Compute transport in thickness space.
Subroutines: **lim_itd_th**, **lim_itd_th_rem**, **lim_itd_th_reb**, **lim_itd_fitline**, **lim_itd_shiftice**.

⋆ **lim_itd_th**
Goal: Compute thermodynamics of the ice thickness distribution
Called by: **sbc_ice_lim**.
Major inputs:
Major outputs:
Remarks:
Structure:

- Transport ice between thickness categories
- Add new ice forming in open water
- Compute trend terms and get back to old values

⋆ **lim_itd_th_rem**
Goal: Computes the redistribution of ice thickness after thermodynamic growth of ice thickness.
Called by: **lim_itd_th_rem** (*limitd_th.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Conservation check

- Compute total ice concentration
- Identify grid cells with ice
- Compute new category boundaries
- Identify cells where ITD has to be remapped
- Fill arrays with lowermost / uppermost boundaries of new categories
- Compute new ITD
- Compute area and volume to be shifted across each boundary
- Shift ice between categories
- Make sure ice thickness is within bounds
- Conservation check

⋆ **lim_itd_fitline**

Goal: Fit g(h) with a line using area and volume constraints.
Called by: **lim_itd_th_rem** (*lim_itd_th.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

⋆ **lim_itd_shiftice**

Goal: Shift ice across category boundaries, conserving everything (area, volume, energy, age*vol, mass of salt).
Called by: **lim_itd_th_rem** (*limitd_th.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Define a variable equal to $a\_i * T\_su$
- Check for daice or dvice out of range
- Transfer volume, energy, etc... between categories
- Update ice thickness and temperatures

⋆ **lim_itd_th_reb**
Goal: Rebin thicknesses into defined categories (Dirac case).
Called by: **lim_update** (*limupdate.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Compute ice thickness
- Make sure ice thicknesses are within bounds
- If a category thickness is not in bounds, shift the entire area, volume and energy to the neighboring category
- Conservation check

lim_msh.F90

Goal: Definition of the ice mesh.
Subroutines: **lim_msh**.

⋆ **lim_msh**
Goal: Definition of the characteristics of the numerical grid.
Called by: **ice_ini** (*iceini.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Initialization of some variables
- Geometric tables
- Metric coefficients for ice dynamics
- Metric coefficients for divergence of the stress tensor
- Initialization of ice masks

lim_rhg.F90

Goal: Sea ice rhelogy.
Subroutines: **lim_rhg**.

⋆ **lim_rhg**
Goal: Determines sea ice drift from wind stress, ice-ocean stress and sea-surface slope, using the EVP rheology.
Called by: **lim_dyn** (*lim_dyn.F90*).
Major inputs:
Major outputs: Ice velocity vector components
Remarks:
Structure:

- Ice-snow mass, ice strength
- Wind / ocean stress, mass terms, coriolis terms
- Solution of momentum equation, iterative procedure
  - Divergence, tension and shear
  - Delta at the centre of the grid cells
  - Stress tensor components zs1 and zs2 at the centre of grid cells
  - Delta on corners
  - Stress tensor components zs12 at corners
  - Ice internal stresses
  - Computation of ice velocity
- Prevent large velocities when the ice is too thin
- Recompute delta, shear and divergence, which are inputs for mechanical redistribution
- Store stress tensor and invariants
- Control prints and charge ellipse

limrst.F90

Goal: Manage sea ice model restarts.
Subroutines: **lim_rst_opn**, **lim_rst_write**, **lim_rst_read**.

⋆ **lim_rst_opn**
Goal: Open an existing the restart file.
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:

Remarks:
Structure:

⋆ **lim_rst_write**
Goal: Output sea ice model state in a restart file (NCDF).
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:
Remarks:
Structure: Puts all required variables in the restart file. This includes the state variables, the ice velocity, but also the ice stress, as well as the moments for advection.

⋆ **lim_rst_read**
Goal: Read sea ice model state from a restart file (NCDF).
Called by: **ice_ini** (*iceini.F90*).
Major inputs:
Major outputs:
Remarks:
Structure: Gets all required variables from the restart file. This includes the state variables, the ice velocity, but also the ice stress, as well as the moments for advection.

limsbc.F90

Goal: Momentum, heat, salt and freshwater fluxes at the ice / ocean interface.
Subroutines: **lim_sbc_flux**, **lim_sbc_tau**.

⋆ **lim_sbc_tau**
Goal: Update the ocean surface stresses due to the ice.
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- CASE *kcpl*=0: computation at present time step
- CASE *kcpl*=1: use the velocity at each ocean time step
- CASE *kcpl*=2: mixed case 0 and 1

⋆ **lim_sbc_flx**
Goal: Update the surface ocean boundary condition for heat, salt and mass over

areas where sea ice is present.

Called by: **sbc_ice_lim** (*sbcice_lim.F90*).

Major inputs:

Major outputs:

Remarks: Needs cleaning and conservation checks

Structure:

- Heat flux
- Mass flux

limtab.F90

Goal: Convert 2D arrays into 1D arrays for faster vertical ice thermodynamics.

Subroutines: **tab_2d_1d**, **tab_1d_2d**.

⋆ **tab_2d_1d**

Goal: Convert 2d arrays into 1d arrays.

Called by: **lim_thd** (*limthd.F90*), **lim_thd_lac** (*limthd_lac.F90*).

Major inputs:

Major outputs:

Remarks:

Structure:

⋆ **tab_1d_2d**

Goal: Convert 1d arrays into 2d arrays.

Called by: **lim_thd** (*limthd.F90*), **lim_thd_lac** (*limthd_lac.F90*).

Major inputs:

Major outputs:

Remarks:

Structure:

limthd_dh.F90

Goal: Vertical accretion / ablation of ice.

Subroutines: **lim_thd_dh**.

⋆ **lim_thd_dh**

Goal: Ice growth and melt.

Called by: **lim_thd** (*limthd.F90*).

Major inputs:

Major outputs:

Remarks: Works with 1d arrays

Structure:

- Compute heat available for surface ablation
- Compute layer thicknesses and snow / ice enthalpy
- Surface ablation and sublimation
- Basal growth and melt
- Pathological cases
- Snow ice formation

limthd_dif.F90

Goal: Compute diffusion of heat in the snow-ice system.
Subroutines: **lim_thd_dif**.

⋆ **lim_thd_dif**
Goal: Compute the time evolution of snow and sea-ice temperature profiles.
Called by: **lim_thd** (*limthd.F90*).
Major inputs:
Major outputs: Temperature in the snow-ice system
Remarks: Works with 1d arrays
Structure:

- Initialization
- Radiative transfer
- Iterative procedure begins
- Thermal conductivity of sea ice
- Kappa factors (thermal conductivity)
- Specific heat and eta factors
- Surface flux
- Tridiagonal system terms (4 cases)
- Solution of the tridiagonal system
- Test convergence
- Conductive heat fluxes at the interfaces

limthd_ent.F90

Goal: Remap ice and snow enthalpies on the new grid.
Subroutines: **lim_thd_ent**.

⋆ **lim_thd_ent**

Goal: Computes the new vertical grid and remap ice and snow enthalpies on the new grid.
Called by: **lim_thd** (*limthd.F90*).
Major inputs:
Major outputs: Ice and snow temperatures
Remarks:
Structure:

- Compute new grid
- Compute some switches
- Redistribution of snow enthalpy
  - Old profile features
  - Energy given by snow in snow ice formation
  - New snow profile (vector indices, layer coordinates, layer thicknesses, weight factors, recover heat content, recover temperature
- Redistribution of ice enthalpy
  - Old profile features (vector indices, cotes of old layers, thickness of old ice layers, inner layers heat content, bottom layers heat content, snow ice layer heat content)
  - New ice profile (vector indices, layer thicknesses, layer cotes, weight factors, heat conservation check)
- Update salinity and recover temperature

limthd.F90

Goal: Manges ice thermodynamics.
Subroutines: **lim_thd**, **lim_thd_glohec**, **lim_thd_con_dif**, **lim_thd_con_dh**, **lim_thd_enmelt**, **lim_thd_init**.

⋆ **lim_thd**

Goal: Main ice thermodynamics routine.

Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Initialization of diagnostic variables, heat content, set some dummies to zero, compute global heat content
- Partial computation of forcing for the thermodynamic sea ice model
- Select icy points and fulfill arrays for the vectorial grid
- Thermodynamic computation
  - Move to 1d arrays
  - Call thermodynamic subroutines (**lim_thd_dif**, **lim_thd_enmelt**, **lim_thd_dh**, **lim_thd_ent**, **lim_thd_sal**)
  - Move back to 2d arrays
- Global variables and diagnostics (ice heat content, snow heat content, change thickness to volume, diagnostic thermodynamic growth rates

⋆ **lim_thd_glohec**
Goal: Compute total heat content for each category.
Called by: **lim_thd** (*limthd.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

⋆ **lim_thd_condif**
Goal: Test energy conservation after heat diffusion.
Called by: **lim_thd** (*limthd.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

⋆ **lim_thd_condh**
Goal: Test energy conservation after enthalpy distribution.
Called by: **lim_thd** (*limthd.F90*).

Major inputs:
Major outputs:
Remarks:
Structure:

⋆ **lim_thd_enmelt**
Goal: Computes sea ice energy of melting.
Called by: **lim_thd** (*limthd.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

⋆ **lim_thd_init**
Goal: Initialize sea ice thermodynamics.
Called by: **lim_thd** (*limthd.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

limthd_lac.F90

Goal: Creation of new ice in open water.
Subroutines: **XXX**.

⋆ **lim_thd_lac**
Goal: New ice formation in open water
Called by: **lim_itd_th** (*limitd_th.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Conservation checks
- Convert units for ice internal energy
- Thickness of newly forming ice
- Identify points where ice forms (select points, move from 2d to 1d vectors)
- Compute characteristics of new ice (thickness, salinity, age, energy budget, volume, salt flux, area)

- Redistribute new ice into ice categories
  - Keep old ice areas and volume in memory
  - Compute excessive new ice area and volume
  - Add ice variables in categories
  - Add excessive new ice at the ice base
- Move back from 1d to 2d vectors
- Change units for ice enthalpy
- Conservation checks

limthd_sal.F90

Goal: Compute salinity variations in sea ice.
Subroutines: **lim_thd_sal**, **lim_thd_sal_init**.

⋆ **lim_thd_sal**

Goal: Salinity variations in sea ice.
Called by: **lim_thd** (*limthd.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Module 1: Vertically-averaged, stationary salinity
- Module 2/4: Salinity varying in time
  - Ice thickness at previous time step
  - Global heat content
  - Switches, salinity tendencies, update salinity, vertical profile if *num_sal*=2 or not if *num_sal*=4
  - Heat flux, brine drainage
  - Salt flux, brine drainage
  - Temperature update
- Module 3: Profile of salinity, constant in time [Schwarzacher 1959]
- Module 5: Constant salinity varying in time [Cox and Weeks 1974]

⋆ **lim_thd_sal_init**
Goal: Initialize salinity variations in sea ice.
Called by: **lim_thd_sal** (*limthdsal.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

lim_trp.F90

Goal: Ice transport.
Subroutines: **lim_trp**., **lim_trp_init**

⋆ **lim_trp**
Goal: Advection / diffusion of sea ice properties .
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- CLF test, reduce time step if needed
- Computation of transported fields
- Advection of ice fields
- Diffusion of ice fields
- Update and limite ice properties after transport

⋆ **lim_trp_init**
Goal: Initialize advection / diffusion.
Called by: **lim_trp** (*limtrp.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

limupdate.F90

Goal: Update sea ice variables.
Subroutines: **lim_update**.

⋆ **lim_update**
Goal: Update sea ice global variables, treat pathological cases.
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:
Remarks: Significant cleaning is required
Structure:

limvar.F90

Goal: Manage the conversion of variables.
Subroutines: **lim_var_agg**, **lim_var_glo2eqv**, **lim_var_eqv2glo**, **lim_var_salprof**, **lim_var_bv**, **lim_var_salprof1d**.

⋆ **lim_var_agg**
Goal: Aggregate sea ice variables over the different categories.
Called by: Used in many instances.
Major inputs: option *n*
Major outputs:
Remarks:
Structure:

- Zero everyting
- Compute total ice concentration and volume per unit area, mean ice thickness and open water fraction
- If n>1, compute, mean snow content, mean salinity, and mean ice age (for output only)

⋆ **lim_var_glo2eqv**
Goal: Compute equivalent variables as a function of global variables.
Called by: Used in many instances.
Major inputs:
Major outputs:
Remarks:
Structure:

- Compute ice thickness, snow thickness, ice salinity, ice age
- Compute ice temperatures
- Compute snow temperatures

- Compute mean temperature

⋆ **lim_var_eqv2glo**
Goal: Compute global variables as a function of equivalent variables.
Called by: Used in many instances.
Major inputs:
Major outputs:
Remarks:
Structure:

- Compute ice volume, snow volume, salt content, age content

⋆ **lim_var_salprof**
Goal: Compute the vertical distribution of ice salinity as a function of bulk salinity (cases *num_sal*=2 or 4) or using the MY ice salinity profile of Schwarzacher [1959] (*num_sal*=3).
Called by: Used in many instances.
Major inputs:
Major outputs:
Remarks: Works with the 2d arrays
Structure:

- Compute vertical salinity profile as a function of bulk salinity
- Compute the vertical profile following [Schwarzacher 1959]

⋆ **lim_var_salprof1d**
Goal: Compute the vertical distribution of ice salinity as a function of bulk salinity (cases *num_sal*=2 or 4) or using the MY ice salinity profile of Schwarzacher [1959] (*num_sal*=3).
Called by: Used in many instances.
Major inputs:
Major outputs:
Remarks: Works with the 1d arrays
Structure:

- Compute vertical salinity profile as a function of bulk salinity
- Compute the vertical profile following [Schwarzacher 1959]

⋆ **lim_var_bv**
Goal: Compute the mean brine volume.
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:

Major outputs:
Remarks:
Structure:

limwri.F90

Goal: Writes sea ice model state in the NCDF output file.
Subroutines: **lim_wri**, **lim_wri_init**.

⋆ **lim_wri**
Goal: Writes sea ice model state in the NCDF output file.
Called by: **sbc_ice_lim** (*sbcice_lim.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

- Initialization
- Write the *icemoa* file (with output fields depending on the ice categories).
- Write the *icemod* file (with output fields independent of the ice category).

⋆ **lim_wri_init**
Goal: Initialize output NCDF file.
Called by: **lim_wri** (*limwri.F90*).
Major inputs:
Major outputs:
Remarks:
Structure:

par_ice.F90

Goal: Definition of some ice parameters (number of layers, categories).

thd_ice.F90

Goal: Definition of some ice thermodynamic parameters as well as 1d arrays.

Bibliography

O. Babko, D. A. Rothrock, and G. A. Maykut. Role of rafting in the mechanical redistribution of sea ice thickness. *Journal of Geophysical Research*, 107:3113, doi:10.1029/1999JC000190, 2002. doi: doi:10.1029/1999JC000190.

M. E. Berliand and T. G. Berliand. Determining the net long-wave radiation of the earth with consideration of the effect of cloudiness (in Russian). *Isv. Akad. Nauk. SSSR Ser. Geophys.*, 1, 1952.

M.E. Berliand and T. G. Strokina. Global distribution of the total amount of clouds (in Russian). *Hydrometeorological, Leningrad*, page 71, 1980.

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

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

S. Bouillon, M. A. Morales Maqueda, V. Legat, and T. Fichefet. An Elastic-Viscous-Plastic Sea Ice Model formulated on Arakawa B and C Grids. *Ocean Modelling*, 27:174–184, 2009. doi: 10.1016/j.ocemod.2009.01.004.

M. I. Budyko. *Climate and Life*. Academic Press, 1974.

S.-H. Chou and R.J. Curran. The effects of surface evaporation parameterizations on climate sensitivity to solar constant variations. *Journal of Atmospheric Sciences*, 38:931–938, 1981.

W. M. Connolley, J.M. Gregory, E.C. Hunke, and A.J. McLaren. On the consistent scaling of terms in the sea-ice dynamics equation. *Journal of Physical Oceanography*, 34(7):1776–1780, 2004.

G. F. N. Cox and W. F. Weeks. Numerical simulations of the profile properties of undeformed first-year sea ice during growth season. *Journal of Geophysical Research*, 93:12,449–12,460, 1988.

G. F. N. Cox and W. S. Weeks. Salinity variations in sea ice. *Journal of Glaciology*, 13:109–120, 1974.

E. Deleersnijder, J.-M. Campin, and E.J.M. Delhez. The concept of age in marine modelling: I. theory and preliminary model results. *Journal of Marine Systems*, 28:229–267, 2001.

E. E. Ebert and J. A. Curry. An intermediate one-dimensional thermodynamic sea ice model for investigating ice-atmosphere interactions. *Journal of Geophysical Research*, 98:10,085–10,109, 1993.

T. Fichefet and M. A. Morales Maqueda. Sensitivity of a global sea ice model to the treatment of ice thermodynamics and dynamics. *Journal of Geophysical Research*, 102:12,609–12,646, 1997.

T. Fichefet and M. A. Morales Maqueda. Modelling the influence of snow accumulation and snow-ice formation on the seasonal cycle of the antarctic sea-ice cover. *Climate Dynamics*, 15:251–268, 1999.

G. M. Flato and W. D. Hibler. Ridging and strength in modeling the thickness distribution of arctic sea ice. *Journal of Geophysical Research*, 100:18611–18626, 1995.

H. Goosse. *Modeling the large-scale behavior of the coupled ocean-sea ice system*. PhD thesis, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1997.

H. Goosse and T. Fichefet. Importance of ice-ocean interactions for the global ocean circulation: a model study. *Journal of Geophysical Research*, 104: 23337–23355, 1999.

T. C. Grenfell and G. A. Maykut. The optical properties of ice and snow in the arctic basin. *Journal of Glaciology*, 18:445–463, 1977.

T.C. Grenfell and D. K. Perovich. Spectral albedos of sea ice and incident solar irradiance in the southern beaufort sea. *Journal of Geophysical Research*, 89: 3573–3580, 1984.

Stephen M. Griffies, Arne Biastoch, Claus Böning, Frank Bryan, Gokhan Danabasoglu, Eric P. Chassignet, Matthew H. England, Rüdiger Gerdes, Helmuth Haak, Robert W. Hallberg, Wilco Hazeleger, Johann Jungclaus, William G. Large, Gurvan Madec, Anna Pirani, Bonita L. Samuels, Markus Scheinert, Alex Sen Gupta, Camiel A. Severijns, Harper L. Simmons, Anne Marie Treguier, Mike Winton, Stephen Yeager, and Jianjun Yin. Coordinated ocean-ice reference experiments (cores). *Ocean Modelling*, 26(1-2):1 – 46, 2009. ISSN 1463-5003. doi: DOI: 10.1016/j.ocemod.2008.08.007. URL `http://www.sciencedirect.com/science/article/pii/S1463500308001182`.

J. Haapala. On the modelling of ice-thickness redistribution. *Journal of Glaciology*, 46(154):427–437, 2000.

M. Harder and P. Lemke. *The Polar Oceans and their role in shaping the global environment*, volume The Nansen Centenial Volume 85 of *Geophysical Monograph*, chapter Modelling the extent of sea ice ridging in the Weddell Sea, pages 187–197. American Geophysical union, 1994.

W. D. Hibler. A dynamic thermodynamic sea ice model. *Journal of Physical Oceanography*, 9:815–846, 1979.

W. D. Hibler. Modeling a variable thickness sea ice cover. *Monthly Weather Review*, 108:1943–1973, 1980.

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

E. C. Hunke and W. H. Lipscomb. Cice: The los alamos sea ice model documentation and software user's manual. Technical report, Los Alamos National Laboratory, 2010.

E.C. Hunke. Viscous plastic sea ice dynamics with the evp model: Linearization issues. *Journal of Computational Physics*, 170:18–38, 2001. doi: doi:10.1006/jcph.2001.6710.

E.C. Hunke and J. K. Dukowicz. The elastic-viscous-plastic sea ice dynamics model in general orthogonal curvilinear coordinates on a sphere-incorporation of metric terms. *Monthly Weather Review*, 130:1848–1865, 2002.

K.V. Høyland. Consolidation of first-year sea ice ridges. *Journal of Geophysical Research*, 107(C6):3062, 2002. doi: doi:10.1029/2000JC000526.

M.O. Jeffries, A.P. Worby, K. Morris, and W. F. Weeks. Seasonal variations in the properties and structural composition of sea ice and snow cover in the bellingshausen and amundsen sea, antarctica. *Journal of Glaciology*, 43(143):138–151, 1997.

E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, A. Leetmaa, B. Reynolds, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. Mo, C. Ropelewski, J. Wang, R. Jenne, and D. Joseph. The NCEP/NCAR 40-Year Reanalysis Project. *Bulletin of the American Meteorological Society*, 77:437–471, 1996.

A. Kovacs. Sea ice, part i. bulk salinity versus ice floe thickness. Technical Report 96-7, Cold Regions Research and Engineering Laboratory, Hanover, New Hampshire, 1996.

W.G. Large and S.G. Yeager. Diurnal to decadal global forcing for ocean and sea ice models: The datasets and climatologies. Technical Report TN-460+STR, NCAR Technical Report, 2004.

O. Lecomte, T. Fichefet, M. Vancoppenolle, and M. Nicolaus. A new snow thermodynamic scheme for large-scale sea-ice models. *Annals of Glaciology*, 52: 337–346, 2011.

T. S. Ledley. Sensitivity of a thermodynamic sea ice model with leads to time step size. *Journal of Geophysical Research*, 90 (D1):2251–2260, 1985.

M. Leppäranta. *The Drift of Sea ice*. Springer, 2005.

M. Leppäranta, M. Lensu, and P. Koslof B. Witch. The life story of a first-year sea ice ridge. *Cold Regions Science and Technology*, 23(3):279–290, 1995.

B. Light, T. C. Grenfell, and D. K. Perovich. Transmission and absorption of solar radiation by arctic sea ice during the melt season. *Journal of Geophysical Research*, 113:C03023, 2008. doi: 10.1029/2006JC003977.

W. H. Lipscomb. Remapping the thickness distribution in sea ice models. *Journal of Geophysical Research*, 106:13989–14000, 2001.

W. H. Lipscomb, E. Hunke, W. Maslowski, and J. Jackaci. Ridging, strength, and stability in high-resolution sea ice models. *Journal of Geophysical Research*, 112:C03S9, 2007. doi: doi:10.1029/2005JC003355.

F. Malmgren. *The Norwegian North Polar Expedition with the 'Maud' 1918-1925 vol. 1a no.5*, chapter On the properties of sea ice, pages 1–67. John Griegds Boktr., Bergen, Norway, 1927.

F. Massonnet, T. Fichefet, H. Goosse, M. Vancoppenolle, P. Mathiot, and C. König-Beatty. On the influence of model physics on simulations of arctic and antarctic sea ice. *The Cryosphere*, 5:687–699, 2011. doi: 10.5194/tc-5-687-2011.

G. A. Maykut and N. Untersteiner. Some results from a time-dependent thermodynamic model of sea ice. *Journal of Geophysical Research*, 76:1550–1575, 1971.

M.G. McPhee. Turbulent heat flux in the upper ocean under sea ice. *Journal of Geophysical Research*, 97(C4):5365–5379, 1992.

D. Notz and M. G. Worster. Desalination processes in sea ice. *Journal of Geophysical Research*, 114:C05006, 2009. doi: 10.1029/2008JC004885.

J.M. Oberhuber. An atlas based on the coads data set: the budgets of heat, buoyancy and turbulent kinetic energy at the surface of the global ocean. Technical Report 15, Max-Planck-Institut für Meterologie, Hamburg, Germany, 1988.

R. R. Parmerter. A model of simple rafting in sea ice. *Journal of Geophysical Research*, 80:1948–1952, 1975.

J. P. Peixoto and A. H. Oort. *Physics of climate*. Springer-Verlag, 1992.

M. J. Prather. Numerical advection by conservation of second-order moments. *Journal of Geophysical Research*, 91:6671–6681, 1986.

D. J. Pringle, H. Eicken, H. J. Trodahl, and L.G.E. Backstrom. Thermal conductivity of landfast antarctic and arctic sea ice. *Journal of Geophysical Research*, 112:C04017, 2007. doi: 10.1029/2006JC003641.

P. Rampal, J. Weiss, and D. Marsan. Positive trend in the mean speed and deformation rate of Arctic sea ice, 1979–2007. *Journal of Geophysical Research*, 114:C05013, 2009. doi: 10.1029/2008JC005066.

D. Rothrock. The energetics of the plastic deformation pack ice by ridging. *Journal of Geophysical Research*, 80:4514–4519, 1975.

W. Schwarzacher. Pack-ice studies in the arctic ocean. *Journal of Geophysical Research*, 64:2357–2367, 1959.

K. Shine. Parameterization of the shortwave flux over high albedo surfaces as a function of cloud thickness and surface albedo. *Quarterly Journal of the Royal Meteorological Society*, 110:747–764, 1984.

K. P. Shine and A. Henderson-Sellers. The sensitivity of a thermodynamic sea ice model to changes in surface albedo parameterization. *Journal of Geophysical Research*, 90:2243–2250, 1985.

M. Steele. Sea ice melting and floe geometry in a simple ice-ocean model. *Journal of Geophysical Research*, 97(C11):17,729–17,738, 1992.

B. Tartinville, J.-M. Campin, T. Fichefet, and H. Goosse. Realistic representation of the surface freshwater flux in an ice-ocean general circulation model. *Ocean Modelling*, 3:95–108, 2001.

D. N. Thomas and G. Dieckmann, editors. *Sea Ice (2nd Edition)*. Wiley-Blackwell, 2010.

A. S. Thorndike. Diffusion of sea ice. *Journal of Geophysical Research*, 91: 7691–7696, 1986.

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

R. Timmermann, H. Goosse, G. Madec, T. Fichefet, C. Ethé, and V. Dulière. On the representation of high latitude processes in the orcalim global coupled sea ice-ocean model. *Ocean Modelling*, 8:175–201, 2005.

L.-B. Tremblay and M. Hakakian. Estimating the sea-ice compressive strength from satellite-derived sea-ice drift and ncep reanalysis data. *Journal of Physical Oceanography*, 36:2165–2172, 2007.

K.E. Trenberth, J.G. Olson, and W.G. Large. A global ocean wind stress climatology based on the ECMWF analyses. Technical Report NCAR/TN-338+STR, 93pp., National Center for Atmospheric Research, Boulder, Colorado, 1989.

N. Untersteiner. Calculations of temperature regime and heat budget of sea ice in the central arctic. *Journal of Geophysical Research*, 69:4755–4766, 1964.

N. Untersteiner, editor. *The Geophysics of Sea Ice*, volume 146 of *NATO ASI Series. Series B, Physics*. Plenum Press, New York, 1986.

M. Vancoppenolle, T. Fichefet, and C. M. Bitz. On the sensitivity of undeformed Arctic sea ice to its vertical salinity profile. *Geophysical Research Letters*, 32: L16502, 2005. doi: 10.1029/2005GL023427.

M. Vancoppenolle, C. M. Bitz, and T. Fichefet. Summer landfast sea ice desalination at Point Barrow : Model and Observations. *Journal of Geophysical Research*, 112:C04022, 2007. doi: 10.1029/2006JC003493.

M. Vancoppenolle, T. Fichefet, and H. Goosse. Simulating the mass balance and salinity of Arctic and Antarctic sea ice. 2. Sensitivity to salinity processes. *Ocean Modelling*, 27 (1–2):54–69, 2009a. doi: 10.1016/j.ocemod.2008.11.003.

M. Vancoppenolle, T. Fichefet, H. Goosse, S. Bouillon, G. Madec, and M. A. Morales Maqueda. Simulating the mass balance and salinity of Arctic and Antarctic sea ice. 1. Model description and validation. *Ocean Modelling*, 27 (1–2):33–53, 2009b. doi: 10.1016/j.ocemod.2008.10.005.

M. Vancoppenolle, H. Goosse, A. de Montety, T. Fichefet, B. Tremblay, and J.-L. Tison. Modeling brine and nutrient dynamics in Antarctic sea ice: the case of dissolved silica. *Journal of Geophysical Research*, 115 (C2):C02005, 2010. doi: 10.1029/2009JC005369.

Martin Vancoppenolle, T. Fichefet, and C. M. Bitz. Modeling the salinity profile of undeformed Arctic sea ice. *Geophysical Research Letters*, 33:L21501, 2006. doi: 10.1029/2006GL028342.

WMO. WMO sea-ice nomenclature, terminology, codes and illustrated glossary. *World Meteorological Organisation, Geneva*, WMO/OMM/BMO 259, TP 145, 1970.