Part VI

Coupling with other models

Chapter 17

Sea ice

This chapter describes the sea ice model in MRI.COM. It treats formation, accretion, melting, and transfer of sea ice and snow. Heat, fresh water, salt, and momentum fluxes are exchanged with the ocean. Sea ice is categorized by its thickness, but it has a single layer. Snow does not have heat capacity (so-called zero-layer). Thus, it might be regarded as an intermediate complexity ice model.

The ice model can be only used as a part of the ocean model. In the program, a few main subroutines of the sea ice model are called from surface flux routines of the ocean model. The sea ice model is only called in the predictor step, not in the corrector step. The sea ice model uses the forward scheme in time integration. The sea ice part of an ocean-sea ice model gives sea surface boundary conditions to the oceanic part. Heat, fresh water, salt, and momentum are exchanged at their interfaces.

This chapter is organized as follows. Section 17.1 outlines the model. The following sections describe details of the solution procedure. According to the order of solving the equations, we deal with thermodynamics in Section 17.2, remapping among thickness categories in Section 17.3, dynamics (rheology) in Section 17.4, advection in Section 17.5, and ridging in Section 17.6, change from snow to sea ice in Section 17.7. Adjustment on the sea ice distribution is explained in Section 17.8. Section 17.9 summarizes fluxes from sea ice to ocean. The option of variable sea ice salinity (v5.1 or later) is explained in Section 17.10. Discretization issues are described in Section 17.13, coupling with an AGCM in Section 17.11, and nesting of the sea ice model in Section 17.12. Finally, usage notes are presented in Section 17.14.

State variables and fluxes in the sea ice model are summarized in Table 17.1, and constants and parameters in Section 17.1.6. For the thermal energy conservation in the sea ice model, see memorandum ("On the thermodynamics processes and the heat balance of the MRI.COM sea ice model (in Japanese)"*).

17.1 Outline

The ice model of MRI.COM is based on the ice-ocean coupled model of Mellor and Kantha (1989). For processes that are not explicitly discussed or included there, such as categorizing by thickness, ridging, and rheology, we adopt those of the Los Alamos sea ice model (CICE) version 3.14 (Hunke and Lipscomb, 2006).

17.1.1 The fundamental equation

The fundamental property that defines the state of sea ice is the fractional area as a function of location (μ, ψ) and thickness (h). The equation for this distribution function $(g(\mu, \psi, h))$ is

$$\frac{\partial g}{\partial t} = -\frac{\partial}{\partial h}(fg) - \frac{1}{h_\mu h_\psi}\left(\frac{\partial(g h_\psi u_I)}{\partial \mu} + \frac{\partial(g h_\mu v_I)}{\partial \psi}\right) + \chi_I^{ridge} + \chi_I^{slush} \tag{17.1}$$

where f is the thermodynamic growth rate of thickness, (u_I, v_I) is the velocity vector of ice pack, χ_I^{ridge} is the rate of change of distribution function caused by ridging (Hunke and Lipscomb, 2006, Section 3), and χ_I^{slush} is the transformation of drafted snow into sea ice (snow ice formation). MRI.COM does not deal with this continuity equation directly, but integrates it for each thickness category and predicts the integrated amount.

* https://mri-ocean.github.io/mricom/mri.com-user_doc.html

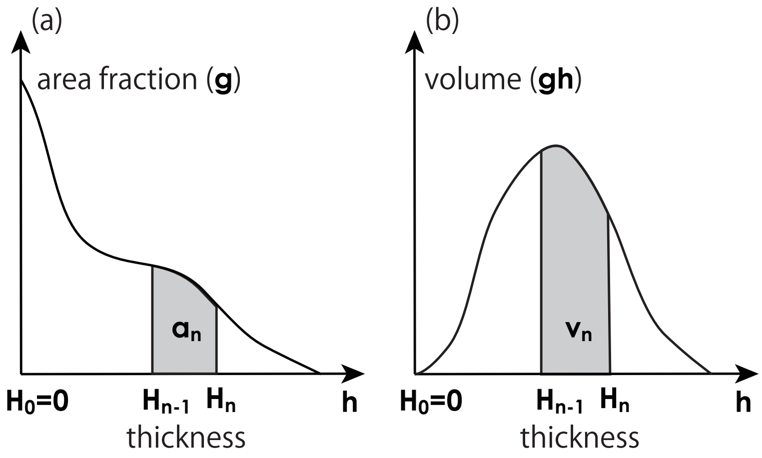

Figure17.1 Schematic views of categorization of sea ice thickness in terms of (a) fractional area a_n and (b) ice volume v_n.

17.1.2 Categorization of sea ice thickness

We discretize the thickness in several categories. If an ice pack is divided into N_c categories separated at $H_0 = 0\,\mathrm{m}$, H_1, H_2, ... and H_{N_c}, the fractional area of category n, a_n, is defined as follows:

$$a_n = \int_{H_{n-1}}^{H_n} g(h)dh. \tag{17.2}$$

As shown in Fig. 17.1, when g is considered as a continuous probability distribution function, a_n is a relative frequency of class n. If sea ice of category n does not exist on the grid, then $a_n = 0$.

Similar to the area fraction a_n, the sea ice volume of category n per unit area, v_n, is given below:

$$v_n = \int_{H_{n-1}}^{H_n} g(h)hdh. \tag{17.3}$$

Other prognostic variables, such as snow volume, v_s, internal energy of ice, e_n, surface temperature, T_3, are also defined for each category. Velocity is defined for an ice pack, the total ice mass in a grid cell. In the vertical direction, both sea ice and snow have one layer. Sea ice has heat capacity, but snow does not. The heat capacity for sea ice is due to brine and is represented by the temperature at the center of the sea ice. Figure 17.2 and Table 17.1 summarize symbols used in this chapter and their variable names in the source code of MRI.COM. Note that the symbol T is used to represent the temperature of any substance treated in this chapter. The unit is °C, though the use of an upper case letter (T) may recall absolute temperature.

Table17.1 State variables and fluxes used in the sea ice part (cf. Figure 17.2) and their variable names in the source code. Note that all water fluxes ($\mathrm{m\,s^{-1}}$) are expressed in terms of sea water with density ρ_o.

	Meaning	Units	Array name for each thickness category	Array name for average or sum over thickness categories

Continued on next page

Table 17.1 – continued from previous page

	Meaning	Units	Array name for each thickness category	Array name for average or sum over thickness categories
a_n, A	area fraction (compactness) (17.2)	1	`aicen`	`a0iceo apr`
v_n	volume per unit area (grid cell average of ice thickness) (17.3)	m	`hin`	`hi`
e_n	volume-integrated enthalpy ($\leq C_{po} m S_I v_n$)	$\mathrm{J\,kg^{-1}\,m}$	`eicen`	-
v_s	snow volume per unit area (grid cell average of snow thickness)	m	`hsn`	`hsnw`
h_I	mean ice thickness ($= v_n/a_n$) (17.15)	m	`hicen`	`hiceo`
h_s	mean snow thickness ($= v_s/a_n$)	m	`hsnwn`	`hsnwo`
q_n	enthalpy of ice per unit volume ($= e_n/v_n$)	$\mathrm{J\,kg^{-1}}$	`qicen`	-
T_3	skin temperature of upper surface	°C	`tsfcin`	`tsfci`
T_2	temperature at snow-ice interface	°C	`t2icen`	-
T_1	temperature of ice ($\leq m S_I$)	°C	`t1icen`	-
$T_{0_I} (= T_0)$	skin temperature of lower surface	°C	`t0icen(1 :)`	`t0iceo`
T_{0_L}	skin temperature of sea surface at open leads	°C	`t0icen(0)`	`t0icel`
$S_{0_I} (= S_0)$	skin salinity of lower surface	pss	`s0n(1 :)`	-
S_{0_L}	skin salinity of sea surface at open leads	pss	`s0n(0)`	-
Q_{I2}	heat flux in the upper half of ice ($= Q_S$)	$\mathrm{W\,m^{-2}}$	`fheatu`	-
Q_{IO}	heat flux on the ice side of the ice bottom	$\mathrm{W\,m^{-2}}$	`fheatn`	-
Q_{AO}	heat flux on the air side at open leads	$\mathrm{W\,m^{-2}}$	`fheat`	-
Q_{LO}	ice-ocean latent heat flux due to melting of snow	$\mathrm{W\,m^{-2}}$	-	`latent_forced_melt`
Q_{SO}	ice-ocean sensible heat flux due to melting of ice	$\mathrm{W\,m^{-2}}$	-	`sens_io_all_melt`
F_{T_I}	heat flux on the ocean side of the ice bottom	$\mathrm{W\,m^{-2}}$	`ftio`	-
F_{T_L}	heat flux on the ocean side at open leads	$\mathrm{W\,m^{-2}}$	`ftao`	-
F_{S_I}	virtual salinity flux below sea ice driving the top layer of the ocean model	$\mathrm{pss\,m\,s^{-1}}$	-	-
F_{S_L}	virtual salinity flux in open water driving the top layer of the ocean model	$\mathrm{pss\,m\,s^{-1}}$	-	-
W	fresh water flux driving the top layer of the ocean model ($= AW_{IO} + (1 - A)W_{AO} + W_{RO} + W_{FR}$)	$\mathrm{m\,s^{-1}}$	-	`wfluxi`
W_{AI}	fresh water flux due to snow fall at the upper surface of ice	$\mathrm{m\,s^{-1}}$	-	`snowfall`
W_{AI}	fresh water flux due to sublimation at the upper surface of ice	$\mathrm{m\,s^{-1}}$	`sublim`	-
W_{IO}	fresh water flux due to freezing and melting at the bottom of ice	$\mathrm{m\,s^{-1}}$	`wio`	-
W_{AO}	fresh water flux due to freezing at open leads	$\mathrm{m\,s^{-1}}$	`wao`	-
W_{ROice}	fresh water flux due to melting of sea ice at the upper surface of ice	$\mathrm{m\,s^{-1}}$	-	`wrsi`

Continued on next page

Table 17.1 – continued from previous page

	Meaning	Units	Array name for each thickness category	Array name for average or sum over thickness categories
W_{ROsnow}	fresh water flux due to melting of snow at the upper surface of ice	m s^{-1}	-	`wrss`
W_{FR}	fresh water flux due to formation of frazil ice	m s^{-1}	-	`wrso`
u_I	zonal component of ice pack velocity	m s^{-1}	-	`uice`
v_I	meridional component of ice pack velocity	m s^{-1}	-	`vice`

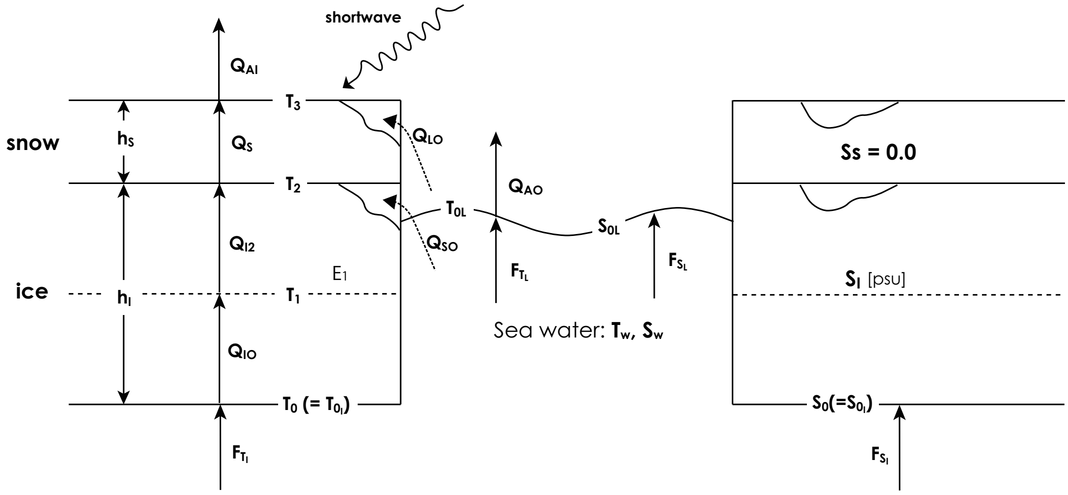

Figure17.2 Schematic diagram of the layer structure of the sea ice model, and related temperature and salinity. The left side is related to temperature and heat flux, and the right side is related to salinity. The central part without sea ice shows the modeling in open lead. See Sec. 17.2 and Table 17.1 for the layer structure and symbols.

17.1.3 The thickness-category governing equations

The sea ice model solves time evolution of the prognostic variables per unit area (sea ice volume, snow volume and sea ice enthalpy) integrated for each category, as defined in the previous section. Based on Eq. (17.1), the equation for the sea ice volume, v_n, is

$$\frac{\partial v_n}{\partial t} = f_I + \mathcal{A}(v_n) + \mathcal{D}(v_n) + \chi_I^{ridge} + \chi_I^{slush}, \tag{17.4}$$

where f_I indicates increase and decrease of sea ice due to thermodynamic processes, $\mathcal{A}(v_n)$ horizontal advection, and $\mathcal{D}(v_n)$ horizontal diffusion. Similarly, the equation for the snow volume, v_s, is

$$\frac{\partial v_s}{\partial t} = f_s + \mathcal{A}(v_s) + \mathcal{D}(v_s) + \chi_s^{ridge} + \chi_s^{slush}. \tag{17.5}$$

The equation for the sea ice enthalpy, e_n, is

$$\frac{\partial e_n}{\partial t} = f_e + \mathcal{A}(e_n) + \mathcal{D}(e_n) + \chi_e^{ridge} + \chi_e^{slush}. \tag{17.6}$$

It should be noted that the physical quantities divided by the density are treated as the conserved quantity, instead of mass conservation and energy conservation.

The growths of ice and snow, f_I and f_s, and the change of enthalpy, f_e, are computed by solving thermodynamic processes (Section 17.2). Using this result, thickness categories are remapped and the fractional area a_n is updated (Section 17.3). To compute the velocity of the ice pack (u_I, v_I), we have to solve the momentum equation (Section 17.4). Then, based on velocity (u_I, v_I), the advection terms, $\mathcal{A}$, and the diffusion terms, $\mathcal{D}$, are calculated (Section 17.5). Using the transported ice distribution function, the ridging process (χ^{ridge}) is solved (Section 17.6). Then, conversion of snow to sea ice due to slush formation, χ^{slush}, is obtained based on the predicted snow thickness and sea ice thickness (Section 17.7).

The outline of the thermodynamic processes is shown in Fig. 17.3. The amounts of change due to each process are calculated from the state variables and fluxes shown in Fig. 17.2 so as to satisfy the conservation laws. The formulation and solving procedure of each process are presented in later sections.

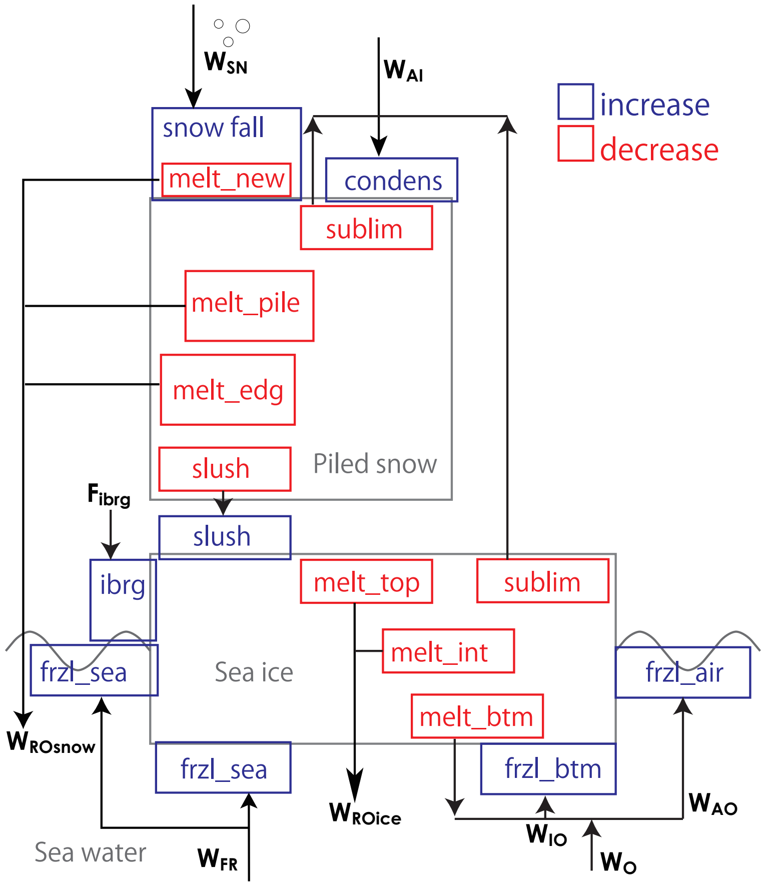

Figure17.3 Schematic diagram of changes of sea ice volume and snow volume due to thermodynamic processes and slash formation. Blue boxes indicate increase and red boxes indicate decrease. Arrows indicate water fluxes between sea ice and ocean or atmosphere.

17.1.4 Thermal energy of sea ice

In considering thermodynamics, thermal energy of sea ice should be defined. The basis of energy (i.e., zero energy) is defined here as that of sea water at 0 °C. The thermal energy (enthalpy; $E(T,r)$) of sea ice that has temperature $T(<0)$ [°C] and brine (salt water) fraction r is the negative of the energy needed to raise the temperature to 0 °C and melt all of it:

$$E(T,r) = r(C_{po}T) + (1-r)(-L_F + C_{pi}T), \tag{17.7}$$

where C_{po} and C_{pi} are the specific heats of sea water and sea ice, and L_F is the latent heat of melting/freezing. Thus defined, the energy of sea ice is negative definite.

Assuming that salinity in the brine is S_b and salinity in the other sea ice is 0, the average salinity S_I in the sea ice is given by

$$S_I = rS_b. \tag{17.8}$$

On the other hand, assuming that the change rate of the freezing temperature with respect to salinity is a constant negative value m, the freezing point in the brine is mS_b. In addition, the temperature of the entire sea ice, T_1, is assumed to be uniformly the freezing point, mS_b:

$$T_1 = mS_b. \tag{17.9}$$

Based on the above two equations, the brine fraction of sea ice, r, can be determined from S_I, T_1,

$$r = mS_I/T_1. \tag{17.10}$$

Since $r \leq 1$, a constraint of $T_1 \leq mS_I = T_{frz}$ is obtained, where T_{frz} is the freezing point of sea water with a salinty of S_I. As in Mellor and Kantha (1989), the specific heat of snow is not considered.

In the sea ice model, enthalpy E (more accurately, enthalpy integrated for each category, e_n), since it is a conserved quantity, is easier to be handled than temperature. Thus, T_1 is determined diagnostically based on E by solving the quadratic equation with (17.10),

$$E(T_1, r) = r(C_{po}T_1) + (1-r)(-L_F + C_{pi}T_1), \tag{17.11}$$

which gives

$$T_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \tag{17.12}$$

where

$$a = C_{pi}, \quad b = -L_F - E + mS_I(C_{po} - C_{pi}), \quad c = mS_I L_F. \tag{17.13}$$

Figure 17.4 shows that the other root is unphysical. †

In the sea ice model, enthalpy is often calculated from T_1 and S_I by (17.11) on the premise of Eq. (17.10). Hereafter, sea ice enthalpy calculated from S_I instead of r is referred to as E_S.

$$E_S(T_1, S_I) = mS_I\left(C_{po} - C_{pi} + \frac{L_F}{T_1}\right) + C_{pi}T_1 - L_F, \tag{17.14}$$

The domain of definition for S_I is $S_I \geq 0$, while that for T_1 is $T_1 \leq mS_I$.

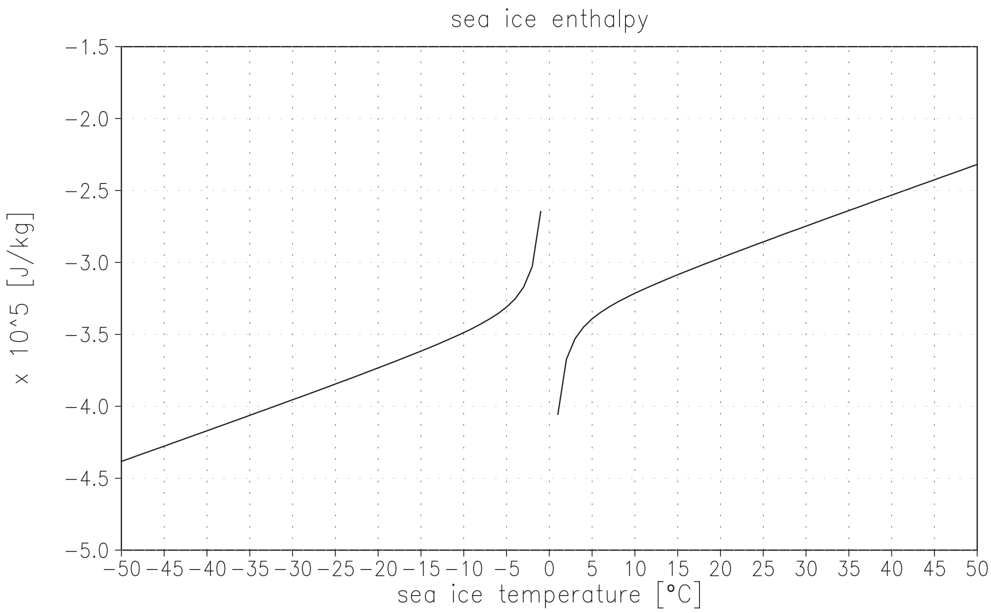

Figure17.4 Sea ice enthalpy E (17.11) as a function of sea ice temperature (T_1). Units in J kg^{-1}.

17.1.5 Diagnostic variables

Secondary diagnostic variables used in the sea ice model are summarized here.

The average sea ice thickness h_I for category n is determined as follows, if a_n is not zero,

$$h_I = v_n/a_n. \tag{17.15}$$

Similarly, the average snow thickness h_s is

$$h_s = v_s/a_n. \tag{17.16}$$

† It does not meet the assumption of Eq. (17.9.) When $r = 1$, that is, $T_1 = T_{frz} = mS_I$, enthalpy E takes the maximum value $C_{po}mS_I$ (sea water at freezing temperature).

The total sea ice area fraction for all categories is expressed as A.

$$A = \sum_{n=1}^{N_C} a_n. \tag{17.17}$$

The fraction of open lead is a_0,

$$a_0 = 1 - A. \tag{17.18}$$

If area fraction deviates from the domain, from zero to one, due to numerical error, it is corrected as follows:

- If $a_n < 0$, a_n is set as 0.
- If $A > 1$, a_n is replaced by a_n/A.

These two modifications guarantee $0 \le A \le 1$. Since area fraction itself is not a conserved quantity, such modification does not affect consistency of the model.

In the sea ice model, areas where the ocean is exposed to the atmosphere, that is, where there is no sea ice, may be treated differently depending on whether or not there is sea ice on the same grid. In this chapter, the former ($0 < a_0 < 1$) is called "open lead" and the latter ($a_0 = 1$) is called "open water".

17.1.6 Physical constant, parameters

Since the ice part is coded in SI units, constants and parameters are written in SI units.

a. Thermodynamics

Table17.2 Physical constants and parameters relevant to sea ice thermodynamics

parameter	value	variable name in MRI.COM
Thermal ice conductivity	$k_I = 2.04\,\mathrm{J\,m^{-1}\,s^{-1}\,K^{-1}}$	`cki`
Thermal snow conductivity	$k_s = 0.31\,\mathrm{J\,m^{-1}\,s^{-1}\,K^{-1}}$	`cks`
Specific heat of sea water	$C_{po} = 3990\,\mathrm{J\,kg^{-1}\,K^{-1}}$	`cp0`
Specific heat of air	Equation (14.130)	`cpair`
Specific heat of ice	$C_{pi} = 2093\,\mathrm{J\,kg^{-1}\,K^{-1}}$	`cpi`
Specific heat of snow	$C_{ps} = 0.0\,\mathrm{J\,kg^{-1}\,K^{-1}}$	—
Stefan Boltzmann constant	$\sigma = 5.67 \times 10^{-8}\,\mathrm{W\,m^{-2}\,K^{-4}}$	`stbl`
Albedo of open ocean surface	$\alpha_o = 0.1$ (default)	`albw`
Albedo of ice	$\alpha_I = 0.6$ (default)	`albi`
Albedo of snow	$\alpha_s = 0.75$ (default)	`albs`
Emissivity of ocean surface	$\epsilon_o = 1.0$	`eew`
Emissivity of ice surface	$\epsilon_I = 1.0$ (default)	`emissivity_sea_ice`
Emissivity of snow surface	$\epsilon_s = 1.0$ (default)	`emissivity_snow`
neutral bulk transfer coefficient for sensible heat	$C_{HAI} = 1.5 \times 10^{-3}$ (default)	`c_transfer_sensible_heat_ai`
neutral bulk transfer coefficient for latent heat	$C_{EAI} = 1.5 \times 10^{-3}$ (default)	`c_transfer_latent_heat_ai`
Latent heat of fusion	$L_F = 3.347 \times 10^5\,\mathrm{J\,kg^{-1}}$	`alf`
Latent heat of sublimation	Equation (14.132)	`rlth`
Constants for fusion phase equation: $T_f = mS + nz$	$m = -0.0543\,°\mathrm{C/pss}$ $n = -0.000\,759\,°\mathrm{C\,m^{-1}}$	`xmxm` `xnxn`
Ice roughness parameter	$z0_I = 0.05h_I/3$	`z0`

Continued on next page

Table 17.2 – continued from previous page

parameter	value	variable name in MRI.COM
Salinity of sea ice	$S_I = 4.0\,\mathrm{pss}$	`si`
von Karman's constant	$k = 0.4$	`xk`
Thickness/compactness diffusion of ice	$\kappa_H = 1.0 \times 10^3\,\mathrm{m^2\,s^{-1}}$	`akh`
Seawater kinematic viscosity	$\nu = 1.8 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$	`anu`
Seawater heat diffusivity	$\alpha_t = 1.39 \times 10^{-7}\,\mathrm{m^2\,s^{-1}}$	`at`
Seawater salinity diffusivity	$\alpha_b = 6.8 \times 10^{-10}\,\mathrm{m^2\,s^{-1}}$	`as`
Turbulent Prandtl number	$P_{rt} = 0.85$	`prt`
b in eqs (17.118),(17.120)	$b = 3.14$	`ab`
sea ice enthalpy (17.14)	$E_S(T_1, S_I)\,\mathrm{J\,kg^{-1}}$	`ts2qice_jpkg`

b. Dynamics

Table17.3 Physical constants and parameters relevant to sea ice dynamics

parameter	value	variable name in MRI.COM
Density of sea water	$\rho_o = 1036\,\mathrm{kg\,m^{-3}}$	`ro0`
Density of air	ρ_a, Equation (14.133)	`roair`
Density of sea ice	$\rho_I = 900\,\mathrm{kg\,m^{-3}}$	`rice`
Density of snow	$\rho_s = 330\,\mathrm{kg\,m^{-3}}$	
Reference snow density (ratio between snow and water)	ρ_s/ρ_o	`rdsw`
e-folding constant for ice pressure	$c^* = 20.0$ (default)	`e_fold_const_ice_prs`
pressure scaling factor	$P^* = 2.75 \times 10^4\,\mathrm{N\,m^{-2}}$ (default)	`prs_scale_factor`
drag coefficient (air-ice)	$C_{DAI} = 3.0 \times 10^{-3}$ (default)	`c_drag_ai`
drag coefficient (ice-ocean)	$C_{DIO} = 5.5 \times 10^{-3}$ (default)	`c_drag_io`
yield curve axis ratio	$e = 2.0$	`elips`
scaling factor for Young's modulus	$E_o = 0.25$	`eyoung`
water turning angle	$\theta_o = \pm cz$ (positive/negative in the northern/southern hemisphere, c: $1°\mathrm{m}^{-1}$, z: the first layer depth [m], default)	`turning_angle_drag_io`
air turning angle	$\theta_a = 0°$ (not considered)	–

17.2 Thermodynamic processes

Vertical one-dimensional thermodynamic processes are solved in each category based on Mellor and Kantha (1989). The sea ice model has a layered structure of sea ice (thickness: h_I) and piled snow (thickness: h_s), as shown in Fig. 17.2. The former is further divided into upper and lower halves. Thus a column of sea ice has three vertical layers. The temperatures at the layer boundaries are T_0, T_1, T_2, and T_3 from the bottom. Heat fluxes within each layer are Q_{IO}, Q_{I2}, and Q_S from the bottom. The heat flux at the air-ice interface is Q_{AI} and that at the ice-ocean interface is F_{T_I}. Sea ice is in fact categorized by thickness, and each symbol should have a suffix (n) of the category number. At open leads or open water, symbols have the suffix L. Sea surface temperature of the ocean is indicated by T_w, and sea surface salinity by S_w.

Although the surface fluxes are positive downward (positive toward the ocean) in the ocean model, the sea ice part is coded such that fluxes are positive upward. In this section, we assume that the fluxes are positive upward.

17.2.1 Fluxes at air-ice interface

To solve the thermodynamic processes, we first evaluate the air-ice heat flux, Q_{AI}, and the freshwater flux, W_{AI} . The heat flux, Q_{AI}, is expressed as follows:

$$Q_{AI} = -(1 - \alpha_I)SW - LW + Q_{SI} + Q_{LI} + \epsilon_I \sigma (T_3 + 273.15)^4, \tag{17.19}$$

where SW is downward shortwave radiation flux, LW is downward longwave radiation flux, Q_{SI} is sensible heat flux, Q_{LI} is latent heat flux, α_I is albedo, ϵ_I is emissivity, and σ is the Stefan-Boltzmann constant. Except for the last term of Eq.

(17.19) (thermal radiation), each of these terms is calculated as follows. The freshwater flux (does not include snow fall), W_{AI}, is caused only by sublimation and condensation on the top of the sea ice.

■ i. Shortwave: The downward shortwave radiation is represented by SW. The albedo of sea ice is α_I, which is 0.82 for cold ($T_3 < -1$ °C) snow, 0.73 for melting snow, and 0.64 for bare ice (while melting).

One might use a more sophisticated albedo scheme included in the Los Alamos sea ice model (CICE; Hunke and Lipscomb, 2006) by choosing `CALALBSI` option. We briefly describe how the incoming shortwave radiation is treated by this scheme. The downward shortwave radiation is treated at each interface as follows: First, among the net absorbed shortwave flux ($= (1-\alpha_I)SW$), some fraction (i_0) penetrates into ice and the rest is absorbed at the surface and used to warm the upper interface. See Table 17.4 for the specific value of i_0. Second, the part penetrating into the ice $SW_{surface}$,

$$SW_{surface} = (1-\alpha_I)i_0 SW, \tag{17.20}$$

is attenuated according to Beer's Law with the bulk extinction coefficient $\kappa_i = 1.4\,\mathrm{m}^{-1}$. The attenuated part ΔSW_{ice} is used to warm the ice interior (Sec. 17.2.3)

$$SW_{bottom} = (1-\alpha_I)i_0 SW \times \exp(-\kappa_i h_I), \tag{17.21}$$

$$\Delta SW_{ice} = SW_{surface} - SW_{bottom}. \tag{17.22}$$

Last, the rest SW_{bottom} enters the ocean (Sec. 14.3).

The albedos and penetration coefficients of CICE are listed on Table 17.4. The property f_{snow} is the snow fraction of the upper surface of the ice, which is expressed as follows:

$$f_{\mathrm{snow}} = \frac{h_s}{h_s + h_{snowpatch}}, \tag{17.23}$$

where h_s is the snow thickness and $h_{snowpatch} = 0.02$ m.

If the ice thickness (h_I) is less than $h_{\mathrm{ref}} = 0.5$ m, the albedo of thin ice is computed as

$$\alpha_{thinice} = \alpha_o + \beta(\alpha_{coldice} - \alpha_o), \tag{17.24}$$

where

$$\beta = \frac{\arctan(a_r h_I)}{\arctan(a_r h_{\mathrm{ref}})}, \quad a_r = 4.0, \tag{17.25}$$

and α_o is the albedo of the ocean.

If the top surface temperature T_3 becomes $-1 < T_3 < 0$ °C, the albedo of melting ice and snow is computed as

$$\alpha_{meltice} = \alpha_{thinice} - \gamma_i(T_3 + 1.0), \tag{17.26}$$

$$\alpha_{meltsnow} = \alpha_{coldsnow} - \gamma_s(T_3 + 1.0), \tag{17.27}$$

where the condition $\alpha_{meltice} > \alpha_o$ is imposed. Using the snow fraction on the surface of the ice f_{snow}, the total albedo is computed as

$$\alpha_i = \alpha_{meltice}(1 - f_{\mathrm{snow}}) + \alpha_{meltsnow} f_{\mathrm{snow}}. \tag{17.28}$$

The albedos for visible and near infra-red wave lengths are computed separately. If the shortwave flux is given as the sum of all four components (direct and diffuse for visible and near infra-red wave lengths), a constant ratio (visible) : (near infra-red) = 0.575 : 0.425 is assumed, and the total albedo is computed as the weighted average.

Table17.4 Albedo and surface transparency of the albedo scheme of CICE.

	near infra-red (> 700 nm)	visible (< 700 nm)
albedo for cold snow $\alpha_{coldsnow}$ ($T_3 < -1$ °C)	0.70	0.98
albedo for cold ice $\alpha_{coldice}$ ($T_3 < -1$ °C, $h_i > 0.5$ m)	0.36	0.78
reduction rate of albedo for melting ice γ_i (-1 °C $< T_3 < 0$, $h_i < 0.5$ m)	–0.075 /°C	–0.075 /°C
reduction rate of albedo for melting snow γ_s (-1 °C $< T_3 < 0$, $h_i < 0.5$ m)	–0.15 /°C	–0.10 /°C
fraction of transparent shortwave flux through the ice surface (i_0)	0.0	$0.7 \times (1.0 - f_{\mathrm{snow}})$

■ ii. Longwave: The downward longwave radiation from the atmosphere is represented by LW in (17.19). The black body radiation from the ice surface is $\epsilon_I \sigma (T_3 + 273.15)^4$, where ϵ_I is emissivity, and σ is the Stefan-Boltzmann constant. Hereinafter, we use

$$LW_I(T_3) = LW - \epsilon_I \sigma (T_3 + 273.15)^4 \tag{17.29}$$

as the net longwave radiation.

■ iii. Sensible heat flux: The sensible heat flux (Q_{SI} in (17.19)) is computed using a bulk formula:

$$Q_{SI}(T_3) = \rho_a C_{pa} C_{HAI} U_{10} (T_3 - T_A), \tag{17.30}$$

where ρ_a is the density of air, C_{pa} is the specific heat of air, C_{HAI} is the bulk transfer coefficient for heat, U_{10} is the scalar wind speed at 10 m, T_A is the surface air temperature. The derivative, $\partial Q_{SI}/\partial T_3$, is also used in later calculations.

$$\frac{\partial Q_{SI}}{\partial T_3} = \rho_a C_{pa} C_{HAI} U_{10}. \tag{17.31}$$

■ iv. Latent heat flux: The latent heat flux (Q_{LI} in (17.19)) is computed using a bulk formula:

$$Q_{LI}(T_3) = \rho_a L_s C_{EAI} U_{10} (q_i - q_A), \tag{17.32}$$

where L_s is the latent heat of sublimation, C_{EAI} is the bulk transfer coefficient for moisture, q_i is the saturation humidity at T_3, and q_A is the specific humidity of air. Section 14.12.2 details a computing method for q_i. The derivative, $\partial Q_{LI}/\partial T_3$ is also used later, as with sensible heat.

$$\frac{\partial Q_{LI}}{\partial T_3} = \rho_a L_s C_{EAI} U_{10} \frac{\partial q_i}{\partial T_3}. \tag{17.33}$$

■ v. Bulk coefficient over sea ice: Three schemes are available for bulk coefficients on sea ice, depending on the options `BULKNCAR`, `BULKECMWF`.

If neither `BULKNCAR`, `BULKECMWF` is specified, the following bulk coefficients are used according to Mellor and Kantha (1989).

$$C_{\mathrm{DAI}} = 3.0 \times 10^{-3} \tag{17.34}$$

$$C_{\mathrm{HAI}} = C_{\mathrm{EAI}} = 1.5 \times 10^{-3}. \tag{17.35}$$

These coefficients are given by namelist `nml_air_ice` at run time. See Table 17.11. The wind speed, air temperature, and specific humidity are used without altitude correction.

If `BULKNCAR` is specified, the constant value specified by `nml_air_ice` is usually used as neutral bulk coefficients at 10 m. To use the same scheme as Large and Yeager (2004), the following values must be specified at runtime.

$$C_{\mathrm{DAI10}} = C_{\mathrm{HAI10}} = C_{\mathrm{EAI10}} = 1.63 \times 10^{-3}. \tag{17.36}$$

The procedures except for neutral bulk coefficients is the same as Section 14.9.2. The bulk coefficients, air temperature and specific humidity are corrected for the altitude at which wind speed is defined, and then used for calculation of turbulent fluxes.

If `BULKECMWF` is specified, roughness length is calculated by the following equation (ECMWF, 2016b):

$$z_0 = \max\left(10^{-3}, 0.93 \times 10^{-3}(1 - a_n) + 0.05 \times 10^{-3} \exp\left[-17(a_n - 0.5)^2\right]\right), \tag{17.37}$$

$$z_{0\mathrm{T}} = 10^{-3}, \tag{17.38}$$

$$z_{0\mathrm{E}} = 10^{-3}. \tag{17.39}$$

Except for this, the procedure is the same as Section 14.9.3. Since the roughness length is uniquely determined by the sea ice fraction a_n, iteration for calculating the roughness length is unnecessary.

Regardless of which scheme you use, the wind stress over sea ice can be calculated with `TAUBULK` option.

■ vi. Fresh water flux: Fresh water loss due to sublimation is computed as

$$W_{AI} = \rho_a C_{EAI} U_{10}(q_i - q_A)/\rho_o, \tag{17.40}$$

where W_{AI} is converted to the volume of sea water by dividing by ρ_o. As is clear from comparison with Eq. (17.32), W_{AI} is determined by the latent heat flux at the air-ice boundary, Q_{LI}.

Finally, in preparation for the next section, we define the following heat flux at the ice-air boundary, Q_{AI}^{old} [W m $^{-2}$] (positive upward).

$$Q_{AI}^{old} = -(1-\alpha_I)(1-i_0)SW - LW_I(T_3^{old}) + Q_{SI}(T_3^{old}) + Q_{LI}(T_3^{old}) + \epsilon_I \sigma (T_3^{old} + 273.15)^4. \tag{17.41}$$

Permeation to the inner region is excluded, and T_3^{old} is the top surface temperature at the previous time step.

17.2.2 Interface between air and snow/sea ice

In response to the air-ice fluxes obtained in the previous section, the existing sea ice and snow are changed. Specifically, the following three steps are performed.

condensation and sublimation

Condensation and sublimation of the freshwater flux, W_{AI} [m / s], causes water mass change per unit area, Δm [kg],

$$\Delta m = -W_{AI} a_n \rho_o \Delta t, \tag{17.42}$$

during a time step interval of Δt for category n. Positive values indicate increase of sea ice or snow mass (condensation), and negative values indicate decrease (sublimation). (It should be noted that this effect works only on sea ice, not in the open water, open lead, and sea ice-free categories.) On the other hand, the mass of snow, m_s, and the mass of sea ice, m_I, existing at the current step for category n are

$$m_s = v_s \rho_{snow}, \qquad m_I = v_n \rho_I. \tag{17.43}$$

Using these masses, the mass change of snow, Δm_s, and that of sea ice, Δm_I, due to condensation and sublimation are calculated according to the following procedure. If condensation occurs, only snow is increased. If sublimation occurs, snow is reduced, and then, sea ice is reduced. When expressed with conditional branch,

$$\begin{cases} \Delta m_s = \Delta m, & \Delta m_I = 0 & \text{if} \quad -m_s < \Delta m \\ \Delta m_s = -m_s, & \Delta m_I = \Delta m + m_s & \text{if} \quad -m_s - m_I < \Delta m < -m_s \\ \Delta m_s = -m_s, & \Delta m_I = -m_I & \text{if} \quad \Delta m < -m_s - m_I \end{cases} \tag{17.44}$$

Note that the first condition includes two situations where condensation occurs ($0 < \Delta m$) and only snow sublimation occurs ($-m_s < \Delta m < 0$).

These conditions can be summarized as follows to express volume change of snow, Δv_s [m], and volume change of sea ice, Δv_n [m].

$$\begin{cases} \Delta v_s = -W_{AI} a_n \frac{\rho_o}{\rho_s} \Delta t, & \Delta v_n = 0 & \text{if} \quad -m_s < \Delta m \\ \Delta v_s = -v_s, & \Delta v_n = -W_{AI} a_n \frac{\rho_o}{\rho_I} \Delta t + v_s \frac{\rho_s}{\rho_I} & \text{if} \quad -m_s - m_I < \Delta m < -m_s \\ \Delta v_s = -v_s, & \Delta v_n = -v_n & \text{if} \quad \Delta m < -m_s - m_I \end{cases} \tag{17.45}$$

Since sea ice salinity is set constant as S_I, salinity of the first layer of the ocean is changed so as to compensate for the decrease of sea ice salinity.

$$S'_w = S_w - \Sigma_n \frac{S_I \Delta v_n \rho_I}{\Delta z \rho_o}. \tag{17.46}$$

The change of sea ice enthalpy, Δe_n, is proportional to the change of sea ice volume.

$$\Delta e_n = q_n \Delta v_n, \tag{17.47}$$

where q_n is sea ice enthalpy per unit volume at the previous step.

When the variable sea ice salinity option is used (v5.1), the change of integrated sea ice salinity, ΔS_v is always zero, because salinity included in sea ice does not increase or decrease during sublimation.

Determination of sea ice surface temperature, T_3

Sea ice surface temperature at the next time step, T_3^{new}, is determined so that the heat flux equilibrates on the sea ice surface. ‡ The basis of this calculation is the relationship between heat fluxes and temperature at the air-ice boundary, in the snow layer, and in the sea ice upper layer, as shown in Fig. 17.2. First, the heat flux at the air-ice boundary, Q_{AI}, is given from (17.19) except for permeation into sea ice:

$$Q_{AI} = -(1-\alpha_I)(1-i_0)SW - LW + Q_{SI}(T_3^{\mathrm{new}}) + Q_{LI}(T_3^{\mathrm{new}}) + \epsilon_I \sigma (T_3^{\mathrm{new}} + 273.15)^4, \tag{17.48}$$

where SW and LW have already been obtained, but Q_{LI}, Q_{SI} and thermal radiation depend on T_3^{new}. Next, the heat flux in the snow, Q_S, is constant within the snow layer (we neglect the heat capacity of snow,) and is computed as

$$Q_S = \frac{k_s}{h_s}(T_2^{\mathrm{new}} - T_3^{\mathrm{new}}), \tag{17.49}$$

where h_s is the thickness of the snow layer given by (17.16), and k_s is the thermal conductivity of snow. In the upper half of the ice layer, the heat flux Q_{I2} is computed as follows:

$$Q_{I2} = \frac{k_I}{h_I/2}(T_1^{\mathrm{old}} - T_2^{\mathrm{new}}), \tag{17.50}$$

where k_I is the thermal conductivity of sea ice. Since we neglect heat capacity of snow, the following equation is obtained:

$$Q_S = Q_{I2}. \tag{17.51}$$

We adopt the semi-implicit method described below to decide T_3^{new}. § The balanced surface temperature is computed by assuming that the fluxes on both sides are the same. That is, using the value at the previous time step, T_3^{old}, as the initial estimate,

$$T_3^{new} = T_3^{old} + \delta T_3. \tag{17.52}$$

the following equation is assumed:

$$Q_S = Q_{AI}. \tag{17.53}$$

Thus, based on T_3^{old} and T_1^{old}, the unknown variables, $\delta T_3, T_2^{new}, Q_{AI}, Q_S$ and Q_{I2} are found.

The equation for δT_3 can be derived using Eqs. (17.48), (17.49), (17.50), (17.51), (17.52) and (17.53).

$$\begin{aligned}\frac{k_I}{h_I/2}\{T_1^{old} - \frac{k_s(h_I/2)(T_3^{old}+\delta T_3) + k_I h_s T_1^{old}}{k_s(h_I/2)+k_I h_s}\} =& Q_{LI}(T_3^{old}+\delta T_3) + Q_{SI}(T_3^{old}+\delta T_3) \\ &- (1-\alpha_I)(1-i_0)SW - LW + \epsilon_I\sigma\{(T_3^{old}+\delta T_3)+273.15\}^4.\end{aligned} \tag{17.54}$$

By expanding the specific heat, latent heat, and thermal radiation in a Taylor series, we have,

$$\begin{aligned}\frac{k_I}{h_I/2}\{T_1^{old} - \frac{k_s(h_I/2)(T_3^{old}+\delta T_3) + k_I h_s T_1^{old}}{k_s(h_I/2)+k_I h_s}\} =& Q_{LI}(T_3^{old}) + Q_{SI}(T_3^{old}) + \frac{\partial Q_{LI}}{\partial T_3}\delta T_3 + \frac{\partial Q_{SI}}{\partial T_3}\delta T_3 - (1-\alpha_I)(1-i_0)SW \\ &- LW_I(T_3^{old}) + 4\epsilon_I\sigma(T_3^{old}+273.15)^3\delta T_3.\end{aligned} \tag{17.55}$$

Using this, we compute δT_3:

$$\delta T_3 = \frac{-Q_{SI}(T_3^{old}) - Q_{LI}(T_3^{old}) + (1-\alpha_I)(1-i_0)SW + LW_I(T_3^{old}) + c(T_1^{old}-T_3^{old})}{\frac{\partial Q_{LI}}{\partial T_3} + \frac{\partial Q_{SI}}{\partial T_3} + 4\epsilon_I\sigma(T_3^{old}+273.15)^3 + c}, \tag{17.56}$$

$$= \frac{-Q_{AI}^{old} + c(T_1^{old}-T_3^{old})}{\frac{\partial Q_{LI}}{\partial T_3} + \frac{\partial Q_{SI}}{\partial T_3} + 4\epsilon_I\sigma(T_3^{old}+273.15)^3 + c}, \tag{17.57}$$

$$c = \frac{k_s k_I}{k_s(h_I/2)+k_I h_s}, \tag{17.58}$$

‡ Thus, T_3 is a diagnostic amount. However, it is treated as a prognostic state variable in the program, since the value of the previous time step, T_3^{old}, is used as the initial estimate for T_3^{new}.

§ To be exact, the interface fluxes should be computed iteratively by adjusting surface temperature T_3^{new} until a balance is achieved, but this is not implemented.

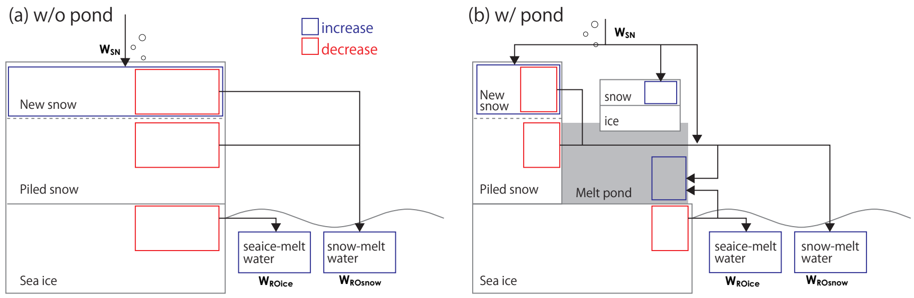

Figure17.5 Diagram of the modeled processes at the sea ice surface. Melt Pond is (a) not considered, and (b) considered.

where $Q_{AI}^{old}, \partial Q_{LI}/\partial T_3, \partial Q_{SI}/\partial T_3$ in the right hand side have been already obtained in Sec. 17.2.1, Eqs. (17.31), (17.33) and (17.41) . Since the solution procedures are basically the same with or without snow, a situation without snow ($h_s = 0, T_3 = T_2$) is included in the above equation. Note that the dependency of L_s on temperature (Section 14.12.2) is not considered in the partial differentiation with respect to temperature. The specific form for the partial derivative of specific humidity ($\partial q_i/\partial T_3$) is presented in Section 17.15.1.

Next, T_3^{new} is given by Eq. (17.52), though an upper limit is set depending on the presence or absence of snow, v_s, as

$$T_3^{new} = \begin{cases} 0 & \text{if} \quad v_s > 0 \quad \text{and} \quad T_3^{old} + \delta T_3 > 0 \\ T_{frz} & \text{if} \quad v_s = 0 \quad \text{and} \quad T_3^{old} + \delta T_3 > T_{frz} \\ T_3^{old} + \delta T_3 & \text{otherwise,} \end{cases} \tag{17.59}$$

where $T_{frz} = mS_I$ is the freezing point of sea ice. In the three cases, the top two conditions cause melting of sea ice or snow, which will be explained in the next section. Then, the interface temperature is computed as

$$T_2^{new} = \frac{k_s(h_I/2)T_3^{new} + k_I h_s T_1^{old}}{k_s(h_I/2) + k_I h_s}, \tag{17.60}$$

and $Q_{I2}(= Q_S)$ is given by Eq. (17.50).

Snowing and melting at the sea ice surface

Here, we solve the snowing and melting processes that occur on the sea ice surface, as shown in Fig. 17.5a (Melt pond is not considered). First, fresh snow is formed on the surface due to snowfall. Next, if the new surface temperature is above the freezing temperature (0.0 °C for snow and mS_I [°C] for sea ice), melting occurs as noted about Eq. (17.59). The whole procedure is as follows:

1. Fresh snow is formed by snowfall. (Δv_s^{new})
2. If surface melting occurs
   (a) All fresh snow melts. ($\Delta v_s^{new\_melt}$)
   (b) Piled snow and sea ice melt. ($\Delta v_s^{pile\_melt}, \Delta v_n$)
   (c) Melted water Inflows to the ocean. (W_{ROice}, W_{ROsnow})

The changes in sea ice and snow at each procedure are shown below.

The fresh snow volume, Δv_s^{new}, due to snowfall, W_{SN}, is given by the following equation:

$$\Delta v_s^{new} = \rho_o W_{SN} a_n \Delta t/\rho_s. \tag{17.61}$$

If T_3^{new} is below the freezing point, only this process occurs in this section.

Below this is the case where surface melting occurs. First, all the fresh snow melts, and the snow volume change due to this melting, $\Delta v_s^{new\_melt}$, is given by

$$\Delta v_s^{new\_melt} = -\Delta v_s^{new}, \tag{17.62}$$

where the negative sign means decrease.

Next, the amounts of melting of piled snow and sea ice are calculated as follows. The heat energy used for melting, ΔE, is given from the balance of heat flux by

$$\Delta E = (Q_{I2} - Q_{AI}^{old})a_n\Delta t, \tag{17.63}$$

where Q_{I2} is given by Eq.(17.50), and Q_{AI}^{old} is given by Eq. (17.41). On the other hand, the energy required to melt all the piled snow, E_s, and that to melt all the sea ice, E_I, are as follows:

$$E_s = v_s \rho_s L_F, \qquad E_I = v_n \rho_I L_3, \tag{17.64}$$

where L_3 is enthalpy change per unit mass due to melting of sea ice

$$L_3 = E_S(T_{frz}, S_I) - q_n = C_{p0} T_{frz} - q_n, \tag{17.65}$$

where q_n is sea ice enthalpy per unit mass. The piled snow is assumed to have zero heat capacity $E(0,0) = -L_F$. In addition, the lower limit, h_{min}, is set for the sea ice thickness, and the corresponding energy amount, E_{min}, is

$$E_{min} = a_n h_{min} \rho_I L_3. \tag{17.66}$$

From these energies, the snow volume change, $\Delta v_s^{pile\_melt}$, and the sea ice volume change, Δv_n, are calculated. When $\Delta E \leq E_s$, only snow is melted as

$$\Delta v_s^{pile\_melt} = -\Delta E / (\rho_s L_f), \qquad \Delta v_n = 0. \tag{17.67}$$

When $E_s < \Delta E \leq E_s + E_I - E_{min}$, snow and sea ice are melted

$$\Delta v_s^{pile\_melt} = -v_s, \qquad \Delta v_n = -(\Delta E - E_s)/(\rho_I L_3). \tag{17.68}$$

When $E_s + E_I - E_{min} < \Delta E$, all the piled snow and sea ice are melted away

$$\Delta v_s^{pile\_melt} = -v_s, \qquad \Delta v_n = -v_n, \tag{17.69}$$

and then excess heat flows into the ocean. That is, the sensible heat flux at the ice-ocean interface, Q_{SO}, is given by

$$Q_{SO} = Q_{AI}^{old} + (E_s + E_I)/(a_n \Delta t). \tag{17.70}$$

Regardless of the conditions, the sea ice enthalpy changes as the sea ice volume decreases, as in Eq. (17.47).

$$\Delta e_n = q_n \Delta v_n. \tag{17.71}$$

The freshwater flux between the ocean and sea ice models due to the melting of sea ice, W_{ROice}, is calculated as follows in terms of seawater volume:

$$W_{ROice} = \frac{\Delta v_n}{\Delta t} \frac{\rho_I}{\rho_o}. \tag{17.72}$$

The freshwater flux due to the melting of snow, W_{ROsnow}, is calculated as follows:

$$W_{ROsnow} = \frac{\Delta v_s^{new\_melt} + \Delta v_s^{pile\_melt}}{\Delta t} \frac{\rho_s}{\rho_o}. \tag{17.73}$$

Melted water is assumed to run off to the ocean immediately, that is, there is no explicit melt pond. However, effect of melt pond is implicitly parameterized by the albedo scheme. Assuming that seawater with a freezing temperature of T_{frz} pours into the ocean, the temperature transport between the ocean and sea ice models is given by

$$F_{top}^{\theta} = T_{frz} \sum_n \Delta v_n. \tag{17.74}$$

The temperature of the melted snow is assumed as 0° C.

Snowing and melting at the sea ice surface with melt pond (v5.1)

Modeling of melt pond is currently under development, some of which are explained here. When melt pond is considered, snowing and melting at the sea ice surface are processed as follows as a whole (Fig 17.5b):

1. Fresh snow is formed by snowfall (modified). (Δv_s^{new})
2. If surface melting occurs
   (a) All fresh snow melts. ($\Delta v_s^{new\_melt}$)
   (b) Piled snow and sea ice melt. ($\Delta v_s^{pile\_melt}, \Delta v_n$)
   (c) Melted water inflows to the ocean and melt pond (modified). (W_{ROice}, W_{ROsnow})

Snowfall causes increase of melt pond volume, as well as fresh snow formation on sea ice. Fresh snow volume, Δv_s^{new}, is

$$\Delta v_s^{new} = \frac{\rho_o W_{SN} \Delta t}{\rho_s}(a_n - a_p), \tag{17.75}$$

where a_p is the area fraction of ice pond. If there is ice on the pond, the volume of fresh snow on pond ice, Δv_{ps}^{new}, and the volume of snow that melts immediately after entering the pond, $\Delta v_s^{melt\_on\_pond}$, are

$$\Delta v_{ps}^{new} = \frac{\rho_o W_{SN} \Delta t}{\rho_s} a_p \tag{17.76}$$

$$\Delta v_s^{melt\_on\_pond} = 0. \tag{17.77}$$

If there is not ice, they are

$$\Delta v_{ps}^{new} = 0 \tag{17.78}$$

$$\Delta v_s^{melt\_on\_pond} = \frac{\rho_o W_{SN} \Delta t}{\rho_s} a_p. \tag{17.79}$$

17.2.3 Heat balance in the ice interior

Next, the time change of enthalpy inside the sea ice is solved from the heat fluxes obtained in the previous section. The thermal energy of the ice is affected by vertical heat fluxes and horizontal heat transport due to advection. The equation for the thermal energy (enthalpy) is written as follows:

$$\rho_I h_I \left[\frac{\partial}{\partial t} E(T_1, r_1) + u_{Ii} \frac{\partial}{\partial x_i} E(T_1, r_1)\right] = Q_{IO} - Q_{I2} + \Delta SW_{ice}, \tag{17.80}$$

where the heat flux Q_{IO} in the lower half of the ice layer is computed as

$$Q_{IO} = -\frac{k_I}{h_I/2}(T_1^{old} - T_0^{old}) = \frac{k_I}{h_I/2}(T_0^{old} - T_1^{old}). \tag{17.81}$$

The heat flux of Q_{I2} has been obtained by Eq. (17.50), and absorption of short waves inside sea ice, ΔSW_{ice}, by Eq. (17.22). When (17.80) is rewritten in term of volume-integrated values for each category,

$$\frac{\partial(\rho_I e_n)}{\partial t} + u_{Ii} \frac{\partial(\rho_I e_n)}{\partial x_i} = (Q_{IO} - Q_{I2} + \Delta SW_{ice}) a_n, \tag{17.82}$$

where integrated enthalpy, e_n, is one of the main prognostic variables of the model. The above equation can be solved explicitly without causing serious problems when the time step is not too long.

In the sea ice model, first, the vertical one-dimensional process of the above equation is solved by the forward finite difference. The enthalpy change due to the vertical heat flux, Δe^{est} [J m / kg], is estimated as follows.

$$\Delta e^{est} = (Q_{IO} - Q_{I2} + \Delta SW_{ice}) \frac{a_n \Delta t}{\rho_I}. \tag{17.83}$$

The upper limit of the sea ice enthalpy with a volume of v_n, e_{max} [J m / kg], depends on the freezing point $T_{frz} = mS_I$,

$$e_{max} = v_n C_{po} T_{frz}. \tag{17.84}$$

When $e_n + \Delta e^{est} \leq e_{max}$, only the enthalpy changes as

$$\Delta v_n = 0, \qquad \Delta v_s = 0, \qquad \Delta e_n = \Delta e^{est}. \tag{17.85}$$

When $e_n + \Delta e^{est} > e_{max}$, all sea ice and snow are melted,

$$\Delta v_n = -v_n, \qquad \Delta v_s = -v_s, \qquad \Delta e_n = -e_n. \tag{17.86}$$

(The integrated enthalpy at the next step is also set zero.) In this case, the heat flux across the sea ice bottom, Q_{IO}, is adjusted so that the enthalpy reaches the upper limit value at the end of the time step (Δt),

$$Q_{SO} = \frac{\rho_I (e_{max} - e^n)}{\Delta t} + (Q_{I2} - \Delta SW_{ice}) a_n, \tag{17.87}$$

Q_{SO} is used instead of $a_n Q_{IO}$ as the sensible heat flux at the ice-ocean interface (17.256).

According to the above decreases of v_n and v_s, fluxes between the sea ice and ocean are increased as follows.

$$\Delta W_{ROsnow} = \Delta v_s \rho_s / (\rho_o \Delta t) \tag{17.88}$$

$$\Delta W_{ROice} = \Delta v_n \rho_I / (\rho_o \Delta t) \tag{17.89}$$

$$\Delta Q_{LO} = \Delta v_s \rho_s L_F / \Delta t, \tag{17.90}$$

where ΔQ_{LO} means ocean heat used to melt snow. Assuming that seawater with a freezing temperature of T_{frz} has been formed, the temperature transport to the ocean is given by

$$F^{\theta}_{intr} = T_{frz} \sum_n \Delta v_n. \tag{17.91}$$

17.2.4 Formation of new sea ice

In addition to the vertical one-dimensional process of heat, sea ice is also formed by iceberg runoff and supercooling of seawater. This results in new ice formation in open water and open leads, which will be distributed to a category of sea ice (Sec. 17.3.2), as well as increase of sea ice in existing categories.

a. Input of iceberg

Input of iceberg (F_{ibrg}) is given as a sea-water volume flux per unit area (m^3 m^{-2} s^{-1}, negative values indicate input), and distributed in proportion to a_n. The increase of sea ice for existing category n is

$$\Delta v_n = -\frac{\rho_o}{\rho_I} F_{ibrg} a_n \Delta t, \qquad \Delta e_n = \Delta v_n q_n. \tag{17.92}$$

The new ice formation in open water and open leads is based on a_0 of (17.18),

$$\Delta v_0 = -\frac{\rho_o}{\rho_I} F_{ibrg} a_0 \Delta t, \qquad \Delta e_0 = \Delta v_0 E_S(T_{frz}, S_I^{ibrg}), \tag{17.93}$$

where $T_{frz} = mS_w$ is the freezing point of ocean (Sec. 14.7.2).

If the variable sea ice salinity option is used (v5.1), the change of salinity is

$$\Delta S_v = \Delta v_0 S_I^{ibrg} \quad \text{or} \quad \Delta v_n S_I^{ibrg}. \tag{17.94}$$

Considering that the salinity in the formed sea ice is supplied from the ocean, an upward salinity flux of F^S_{IO} [m psu s^{-1}] is given to the surface of the ocean model (that is, salinity reduction in ocean):

$$F^S_{IO} = \frac{\rho_I}{\Delta t \rho_o} \sum_{n=0}^{N_c} \Delta S_v(n). \tag{17.95}$$

If the salinity of the formed sea ice is assumed constant at S_I^{ibrg} (v5.0),

$$F^S_{IO} = -F_{ibrg} S_I^{ibrg}. \tag{17.96}$$

b. From sea water with temperature below the freezing point

Sea ice is formed when sea surface temperature is below the freezing point. In this model, this process is considered as follows. If the temperature of the first layer of the ocean model, T_w, is below the freezing point as a function of salinitym, T_{frz}, the temperature is set to the freezing point, and the heat needed to raise the temperature is regarded as the release of latent heat and is used to form new ice. Therefore, the thickness of the new ice (Δh_I) can be computed by assuming that the total thermal energy of the first layer of the ocean model (which has the layer thickness of $\Delta z_{\frac{1}{2}}$) is conserved before and after the sea ice formation:

$$\rho_o \Delta z_{\frac{1}{2}} E(T_w, 1) = \rho_o \Delta z_{\frac{1}{2}} [C_{po} T_w] = \rho_I \Delta h_I E_S(T_{\text{ice}}, S_I) + \rho_o (\Delta z_{\frac{1}{2}} - \rho_I \Delta h_I / \rho_o)(C_{po} T_{frz}) \tag{17.97}$$

where E is sea ice enthalpy per unit mass (17.7), and $T_{\rm ice}$ is the temperature of the newly formed sea ice. For $T_{\rm ice}$, $T_1(n)$ is used if sea ice exists for category n, while T_{0_L} in open leads and $T_{\rm frz}$ in open water. (In v5.1, $T_{\rm frz}$ is used in both of open leads and open water.) Using the above equation, we compute the thickness of the new ice:

$$\Delta h_I = \frac{\rho_o \Delta z_{\frac{1}{2}}}{\rho_I} \frac{C_{po}(T_{frz} - T_w)}{(T_{frz}C_{po} - E_S(T_{\rm ice}, S_I))}. \tag{17.98}$$

(The values at the m time step are used for all the state variables.)

Based on the sea ice growth, increase of sea ice volume during one time step is evaluated for each of the open water and leads and the sea ice bottom. First, in open water and leads,

$$\Delta v_0 = \Delta h_I a_0, \qquad \Delta e_0 = \Delta v_0 E_S(T_{frz}, S_I^{frzl}), \tag{17.99}$$

where S_I^{frzl} is salinity of the newly formed sea ice (a constant value). Sea ice increase for the existing category-n sea ice is

$$\Delta v_n = \Delta h_I a_n, \qquad \Delta e_n = \Delta v_n q_n. \tag{17.100}$$

If the variable sea ice salinity option is used (v5.1), the change of salinity is

$$\Delta S_v = \Delta v_0 S_I^{frzl} \quad \text{or} \quad \Delta v_n S_I^{frzl}. \tag{17.101}$$

The salinity of the formed sea ice, S_I^{frzl}, is determined by the growth rate of the sea ice (Cox and Week, 1978).

The freshwater flux between the ocean and sea ice due to the sea ice formation is given by

$$W_{FR} = \frac{\rho_I}{\rho_o}\left(\Delta v_0 + \sum_{n=1}^{N_c} \Delta v_n\right)/\Delta t. \tag{17.102}$$

The water transported from the ocean to the sea ice is assumed to have the sea surface (first level) temperature, T_w, and then

$$F_{frzl}^{\theta} = T_w \sum_{n=0}^{N_c} \Delta v_n. \tag{17.103}$$

Note that sea surface temperature, T_w, and sea surface salinity, S_w, are rewritten in the process. (In v5.1, instead of modifying T_w and S_w, temperature and salinity fluxes to the ocean are used.)

Note that this operation practically eliminates super-cooling in the ocean interior. Hence, the formation of frazil ice is not considered in this model.

17.2.5 Ice-ocean interface

Melting and freezing at the ice-ocean interface is computed using heat fluxes at the interface as depicted in Figure 17.2. The solution method slightly differs from that of Mellor and Kantha (1989). In this section, we first list main relational equations between state variables and fluxes. Then, the bulk coefficients, skin layer salinity and temperature, and heat fluxes are derived, and, based on them, sea ice change and fluxes to the ocean are evaluated. The ice-covered area and the open leads are treated separately, while this process does not work in open water.

Basic relations

In the ice-covered area, the heat flux on the ice side of the ice-ocean interface (Q_{IO}) has been already computed according to (17.81). In open leads, the heat flux on the air side of the air-ocean interface (Q_{AO}) has also been computed as in Chapter 14,

$$Q_{AO}^{all} = Q_{SO}(T_w) + Q_{LO}(T_w) - (1-\alpha_o)SW - LW + \epsilon_o\sigma(T_w + 273.15)^4. \tag{17.104}$$

All the heat and fresh water fluxes are evaluated using the temperature and salinity at the first level of the ocean model (T_w and S_w). By doing so, the equation to compute melting and freezing rates becomes linear. Here, shortwave radiation is assumed to pass through the skin layer without absorption and is excluded from the evaluation of the freezing rate in open water:

$$Q_{AO} = Q_{SO} + Q_{LO} - LW + \epsilon_o\sigma(T_w + 273.15)^4. \tag{17.105}$$

This operation causes the shortwave radiation to be absorbed in the ocean interior. In reality, the heat stored in the skin layer in open water is used to melt ice laterally (edge melting), and its details are described in a later part of this section.

Melting and freezing represented by W_{IO} and W_{AO} occur owing to the imbalance between fluxes above (Q_{IO}, Q_{AO}) and below (F_{T_I}, F_{T_L}) the interface:

$$F_{T_I} = Q_{IO} - W_{IO}\rho_o L_{o_I}(n), \quad (17.106)$$
$$F_{T_L} = Q_{AO} - W_{AO}\rho_o L_{o_L}, \quad (17.107)$$

where $L_{o_I}(n)$ and L_{o_L} are amounts of energy [J / kg] required to form unit mass of sea ice,

$$L_{o_I}(n) \equiv [E(T_w, 1) - q_n(n)], \quad (17.108)$$
$$L_{o_L} \equiv [E(T_w, 1) - E_S(\widetilde{T}_{0_L}, S_I^{frzl})], \quad (17.109)$$

where S_I^{frzl} is salinity of sea ice formed in open leads, and $\widetilde{T}_{0_L}$ is the ice skin temperature calculated for open leads at the previous time step. (In v5.1, the ocean freezing temperature, $T_{\mathrm{frz}} = mS_w$, is used instead of the somewhat virtual value $\widetilde{T}_{0_L}$.)

Following the formulation adopted by Mellor and Kantha (1989), we introduce skin layer temperature and salinity (T_{0_I}, T_{0_L}, S_{0_I}, and S_{0_L}) to solve for W_{IO} and W_{AO}, and thus F_{T_I} and F_{T_L}. To incorporate skin layer salinity in the system of equations, we consider the problem in terms of virtual salt flux, in which the effect of freshwater flux on salinity is considered in terms of salt flux by keeping the water volume. The flux balance for salt below the interface is written as follows:

$$F_{S_I} = W_{IO}(S_I - S_w), \quad (17.110)$$
$$F_{S_L} = W_{AO}(S_{\mathrm{frzl}} - S_w). \quad (17.111)$$

Here, unlike Mellor and Kantha (1989), S_w is used instead of the salinity at the skin layer (S_{0_I}, S_{0_L}). By doing so, the equations to solve for S_{0_I} and S_{0_L} become linear as shown below. It could also be said that it is natural to use the first level salinity itself in evaluating the salt flux that drives the first level of the ocean model. Note that only fresh water fluxes that are relevant to freezing and melting at the ice-ocean interface are included in the above equations. The restoration to climatological salinity and fresh water fluxes caused by surface melting, precipitation, and evaporation are excluded in the above balance.

Fluxes on the oceanic side of the interface (F_{T_I}, F_{T_L}, F_{S_I}, and F_{S_L}) can also be obtained as the boundary conditions ($z \longrightarrow 0$) for the molecular boundary layer:

$$F_{T_I}/(\rho_o C_{po}) = -C_{T_z}(T_{0_I} - T_w), \quad (17.112)$$
$$F_{T_L}/(\rho_o C_{po}) = -C_{T_z}(T_{0_L} - T_w), \quad (17.113)$$

and

$$F_{S_I} = -C_{S_z}(S_{0_I} - S_w), \quad (17.114)$$
$$F_{S_L} = -C_{S_z}(S_{0_L} - S_w). \quad (17.115)$$

The coefficients, C_{T_z} and C_{S_z}, will be explained in detail in the next section.

The above equations are solved simultaneously to obtain melting and freezing at the upper surface of the ocean under the following constraints:

$$T_{0_I} = mS_{0_I} \quad \text{and} \quad T_{0_L} = mS_{0_L}, \quad (17.116)$$

where m defines the freezing line as a function of salinity.

Bulk coefficients in the ice bottom boundary layer

Before solving the above system of equations, the bulk coefficients, C_{T_z} and C_{S_z}, must be determined. The temperature coefficient, C_{T_z}, is

$$C_{T_z} = \frac{u_\tau}{(P_{rt}k^{-1}\ln(-z/z_0) + B_T)}, \quad (17.117)$$

$u_\tau \equiv (\tau_{IO_x}^2 + \tau_{IO_y}^2)^{1/4}\rho_o^{-1/2}$ is the friction velocity, $k = 0.4$ is von Karman's constant, z_0 is the roughness parameter, $(\tau_{IO_x}, \tau_{IO_y})$ is the stress vector at the ocean-ice interface, and

$$B_T = b\left(\frac{z_0 u_\tau}{\nu}\right)^{1/2} Pr^{2/3}, \quad (17.118)$$

with $Pr \equiv \nu/\alpha_t = 12.9$. The specific values for other parameters are given in Section 17.1.6.

Parameters related to salinity are given by

$$C_{S_z} = \frac{u_\tau}{(P_{rt}k^{-1}\ln(-z/z_0) + B_S)}, \tag{17.119}$$

and

$$B_S = b\left(\frac{z_0 u_\tau}{\nu}\right)^{1/2} Sc^{2/3}, \tag{17.120}$$

where $Sc \equiv \nu/\alpha_b = 2432$.

Following Mellor and Kantha (1989), roughness parameter z_0 is computed as follows:

$$\ln z_0 = A \ln z_{0_I} + (1 - A)\ln z_{0_L}, \tag{17.121}$$

where

$$z_{0_I} = 0.05\frac{h_I}{h_{I\text{lim}}}, \quad h_{I\text{lim}} = 3.0\,\text{m}, \tag{17.122}$$

and

$$z_{0_L} = 0.016\,\frac{\rho_o}{\rho_a}\,\frac{u_\tau^2}{g}. \tag{17.123}$$

However, the roughness parameter below ice (17.122) is also used for open water (z_{0_L}) in MRI.COM.

Ice-bottom boundary flux based on skin salinity and temperature

Using the obtained bulk coefficients, the five unknown variables, $W_{IO}, F_{T_I}, F_{S_I}, T_{0_I}$ and S_{0_I} are solved under the five equations, (17.106), (17.110), (17.112), (17.114) and (17.116), for the ice-covered area. Also for open leads, the equation system is solved in the same way. Specifically, we first solve the skin layer salinity S_{0_I} (ice-covered area) and S_{0_L} (open leads) using (17.106), (17.110), (17.112), (17.114), (17.107), (17.111), (17.113), and (17.115):

$$S_{0_I} = \frac{C_{S_z} S_w + (\rho_o C_{po} C_{T_z} T_w - Q_{IO})(S_I - S_w)/\rho_o L_o}{C_{S_z} + \rho_0 C_{po} C_{T_z} m(S_I - S_w)/\rho_o L_o}, \tag{17.124}$$

$$S_{0_L} = \frac{C_{S_z} S_w + (\rho_o C_{po} C_{T_z} T_w - Q_{AO})(S_{\text{frzl}} - S)/\rho_o L_o}{C_{S_z} + \rho_o C_{po} C_{T_z} m(S_{\text{frzl}} - S_w)/\rho_o L_o}. \tag{17.125}$$

However, the upper and lower limits are set as $S_I \le S_{0_I} \le S_{\max}$ and $S_{\text{frzl}} \le S_{0_L} \le S_{\max}$, in order to avoid deviation from the domain due to rounding error etc. (In v5.1, since S_{0_L} is used only here, the upper and lower limits are not set according to the original paper.) Using S_{0_I} and S_{0_L}, T_{0_I} and T_{0_L} are computed from (17.116), and then, F_{T_I} and F_{T_L} are computed from (17.112) and (17.113).

Here, the meaning of the skin salinity, S_{0_I} or S_{0_L}, is explained in detail. In this scheme, the salinity flux between ocean and sea ice is treated as a "virtual salinity flux". In other words, decrease in ocean salinity due to sea ice melting is treated as an upward salinity outflow across the sea surface boundary for the ocean model, and increase in ocean salinity due to freezing (brine discharge) is treated as a downward inflow. To represent them in (17.114) and (17.115), the skin salinity, S_{0_I} or S_{0_L}, are virtually set. (Therefore, impossible values such as negative values are acceptable.) That is, the skin salinity is a "virtual boundary condition" for obtaining the salinity flux in the ocean model, and has no substance. Similarly, the skin temperature is also a virtual boundary value.

Sea ice change

In the ice-covered area, the difference between the heat flux inside the sea ice, Q_{IO}, and that in the ice-bottom boundary layer, F_{T_I}, is the amount of enthalpy used to melt sea ice or freeze sea water, Δe.

$$\Delta e = \frac{1}{\rho_I}(F_{T_I} - Q_{IO}) a_n \Delta t. \tag{17.126}$$

Meanwhile, the amount of enthalpy required to melt all sea ice and transform it into sea water with a temperature of T_w, Δe_{max}, is given by

$$\Delta e_{max} = v_n(E(T_w, 1) - q_n(n)) = C_{po} T_w v_n - e_n, \tag{17.127}$$

where subscripts n designate the thickness category. Using these two amounts, the sea ice change during one time step such as $\Delta v_n, \Delta v_s, \Delta e_n$ and Δa_n can be determined.

If $\Delta e < 0$, sea water freezes, and

$$\Delta v_n = -\frac{\Delta e}{L_{O_I}} = -\frac{\Delta e}{C_{po}T_w - q_n(n)}, \qquad \Delta v_s = 0, \qquad \Delta e_n = \Delta v_n q_n(n) = \Delta e \frac{q_n(n)}{q_n(n) - C_{po}T_w}, \qquad \Delta a_n = 0, \tag{17.128}$$

If $0 < \Delta e < \Delta e_{max}$, part of sea ice melts. Assuming that it melts into sea water with a temperature of T_w, Δv_n and Δe_n are

$$\Delta v_n = -\frac{\Delta e}{C_{po}T_w - q_n(n)}, \qquad \Delta e_n = \Delta v_n q_n(n) = \Delta e \frac{q_n(n)}{q_n(n) - C_{po}T_w}. \tag{17.129}$$

In this case, reduction of the ice fraction is allowed by "edge melting" as follows. When decrease of sea ice thickness is written as Δh, a part of it, $\Psi \Delta h$, is assumed to be a loss caused by decrease in ice fraction, Δa_n. When the original sea ice thickness is written as h, the following relationship holds

$$\Psi \Delta h a_n = (h - (1 - \Psi)\Delta h)\Delta a_n. \tag{17.130}$$

This becomes

$$c = \Psi \frac{\Delta v_n}{v_n - (1 - \Psi)\Delta v_n} \tag{17.131}$$

$$\Delta a_n = c a_n. \tag{17.132}$$

As snow on the reduced ice fraction melts,

$$\Delta v_s = \frac{\Delta a_n}{a_n} v_s = c v_s. \tag{17.133}$$

There seems to be no widely accepted parameterization scheme for edge melting. According to Steele (1992), bottom and top melting are dominant processes, and thus $\Psi \sim 0.0 - 0.1$ is usually used in MRI.COM.

Finally, if $\Delta e > \Delta e_{max}$, all sea ice and snow melt:

$$\Delta v_n = -v_n, \qquad \Delta v_s = -v_s, \qquad \Delta e_n = -e_n, \qquad \Delta a_n = -a_n. \tag{17.134}$$

In this case, the remaining heat of Δe is added to the heat flux to the ocean. Thus, $a_n F_{T_I}$ is re-evaluated and given to the ocean.

$$a_n F_{T_I} = a_n Q_{IO} + \frac{\Delta e_{max} \rho_I}{\Delta t}. \tag{17.135}$$

In open leads, the amount of enthalpy used for freezing on the sea surface, Δe, is

$$\Delta e = \frac{1}{\rho_I}(F_{T_L} - Q_{AO}) a_0 \Delta t. \tag{17.136}$$

If $\Delta e < 0$, frazil ice is formed in open leads. The volume of the newly-created ice, $\Delta v_{\text{frazil\_air}}$, and its enthalpy, $\Delta e_{\text{frazil\_air}}$, are

$$\Delta v_{\text{frazil\_air}} = -\frac{\Delta e}{L_{O_L}}, \qquad \Delta e_{\text{frazil\_air}} = \Delta v_{\text{frazil\_air}} E_S(T_{frz}, S_I^{frzl}) = \Delta e \frac{E_S(T_{frz}, S_I^{frzl})}{E_S(T_{frz}, S_I^{frzl}) - C_{po}T_w}, \tag{17.137}$$

where L_{O_L} is given by (17.109), assuming that sea ice with a temperature of T_{0_L} (T_{frz} in v5.1) and with a salinity of S_I^{frzl} is formed. Note that $\Delta v_{\text{frazil\_air}}$ is in the range of $[0., H_{N_c}]$, and that $\Delta e_{\text{frazil\_air}}$ is negative. If $\Delta e \geq 0$, frazil ice is not formed, and the heat flux at the top of the sea surface is used for the heat flux to the ocean instead of (17.113),

$$F_{T_L} = Q_{AO}. \tag{17.138}$$

Fluxes to ocean

At the end of this chapter, we summarize the water and heat fluxes to the ocean due to the sea ice changes. The water flux associated with the freezing and melting, W_O, is

$$W_O = a_0 W_{AO} + \sum_n a_n W_{IO} = \frac{\rho_I}{\rho_o \Delta t}\left(\Delta v_{\text{frazil\_air}} + \sum_n \Delta v_n\right), \tag{17.139}$$

using the volume of seawater as a unit. The freshwater flux due to melting of piled snow is

$$W_{ROsnow} = \frac{\rho_s}{\rho_o \Delta t}\left(\sum_n \Delta v_s\right). \tag{17.140}$$

where W_{ROsnow} is added to (17.73) calculated in Sec. 17.2.1. For the temperature transport (positive upward), $F^{\theta}_{btm\_frz}$ associated with freezing and $F^{\theta}_{btm\_melt}$ associated with thawing are calculated separately, though they will be added up later. Assuming seawater with a temperature of SST, T_w, for increase and decrease, (Even in the case of melting, sea ice is transformed into seawater with a temperature of T_w as in (17.129))

$$F^{\theta}_{btm\_frz} = T_w \sum_{n=0} \Delta v_n \qquad (\Delta v_n > 0) \tag{17.141}$$

$$F^{\theta}_{btm\_melt} = T_w \sum_{n=0} \Delta v_n \qquad (\Delta v_n < 0) \tag{17.142}$$

The heat flux that drives the first level of the ocean model, F_T, is given by combining those in open leads and ice-covered areas:

$$F_T = a_0 F_{T_L} + \sum_n (a_n F_{T_I}). \tag{17.143}$$

Since ocean's heat is used to melt snow, the (upward) latent heat flux from the ocean, Q_{LO}, is

$$Q_{LO} = -\frac{\rho_s L_F}{\Delta t}\sum_n (\Delta v_s). \tag{17.144}$$

17.3 Remapping in thickness space

17.3.1 Re-categorization

After the thermodynamic processes are solved, the resultant ice thickness in some thickness categories might not be within the specified bound. Following the method adopted by CICE (Sec.3.2), we assume that there is a thickness distribution function in each category and use it to redistribute the new thickness distribution into original categories.

This procedure corresponds to the first term on the r.h.s. of (17.1):

$$\frac{\partial g}{\partial t} = -\frac{\partial}{\partial h}(fg). \tag{17.145}$$

In practice, a thickness category is regarded as a Lagrange particle, and the category boundaries are displaced as a result of thermodynamics. A linear thickness distribution function is assumed within each displaced category, and ice is remapped into the original categories using these functions.

First, boundaries of thickness categories are displaced. If the ice thickness in category n ($h_{I,n}$) changes from $h^m_{I,n}$ to $h^{m+1}_{I,n}$ (m is the time step index), the growth rate (f_n) at thickness $h^m_{I,n}$ is represented as

$$f_n = (h^{m+1}_{I,n} - h^m_{I,n})/\Delta t \qquad \text{for} \quad n = 1, 2, \cdots, n_c \tag{17.146}$$

$$f_0 = (\Delta v_{\mathrm{frazil\_sea}} + \Delta v_{\mathrm{frazil\_air}})/(a_0 \Delta t). \tag{17.147}$$

($f_0 \geq 0$ by definition) Using this, the growth rate (F_n) at the upper category boundary H_n is obtained by linear interpolation:

$$F_n = f_n + \frac{H_n - h^m_{I,n}}{h^m_{I,n+1} - h^m_{I,n}}(f_{n+1} - f_n) \qquad \text{for } n = 1, 2, \cdots, n_c - 1. \tag{17.148}$$

Note that, if the fractional area is zero in either category n or $n+1$, F_n is set to the growth rate at the non-zero category, f_n or f_{n+1}. When the fractional area is zero on both categories, $F_n = 0$. The lowest boundary, F_0, is given by

$$F_0 = \begin{cases} f_0 & \text{if } f_0 > 0 \\ f_1 & \text{if } f_0 = 0 \text{ and } f_1 < 0 \\ 0 & \text{if } f_0 = 0 \text{ and } f_1 \geq 0. \end{cases} \tag{17.149}$$

The new category boundary after thermodynamics is obtained as

$$H_n^* = H_n + F_n \Delta t \qquad \text{for } n = 1, 2, \cdots, n_c - 1, \tag{17.150}$$

though H_n^* is assumed to be in the range from $h_{I,n}^{m+1}$ to $h_{I,n+1}^{m+1}$. The boundary for the maximum category n_c, $H_{n_c}^*$, is set by linear extrapolation as

$$H_{n_c}^* = h_{I,n_c}^{m+1} + 2(h_{I,n_c}^{m+1} - H_{n_c-1}) = 3h_{I,n_c}^{m+1} - 2H_{n_c-1}. \tag{17.151}$$

The lowest boundary is

$$H_0^* = \begin{cases} H_0 + F_0\Delta t & \text{if } F_0 > 0 \text{ i.e. } f_0 > 0 \\ 0 & \text{if } F_0 \le 0 \text{ i.e. } f_0 = 0. \end{cases} \tag{17.152}$$

Next, the thickness distribution function g within the displaced category $[H_{n-1}^*, H_n^*]$ is determined. First, tentatively, the lower boundary height at the $m+1$ step, H_L, is given by H_{n-1}^*, while the higher boundary height, H_R, is given by H_n^*. Function $g(h)$ should satisfy the following definitions for fractional area (a_n, 17.2) and volume (v_n, 17.3):

$$\int_{H_L}^{H_R} g(h)dh = a_n, \tag{17.153}$$

$$\int_{H_L}^{H_R} hg(h)dh = v_n, \tag{17.154}$$

where h is a continuous variable as explained in Sec. 17.1.2, and a_n and v_n in the right hand side are prognostic variables at the $m+1$ time step.

We adopt a linear function of thickness for g. The thickness space is transformed to $\eta = h - H_L$, and the thickness distribution function is written as $g = g_1\eta + g_0$. These are substituted into (17.153) and (17.154) to yield

$$g_1 \frac{\eta_R^2}{2} + g_0 \eta_R = a_n, \tag{17.155}$$

$$g_1 \frac{\eta_R^3}{3} + g_0 \frac{\eta_R^2}{2} = a_n \eta_n, \tag{17.156}$$

where $\eta_R = H_R - H_L$ and $\eta_n = h_{I,n}^{m+1} - H_L$, and $v_n = h_{I,n} a_n$ is used. These are algebraically solved for g_0 and g_1 as

$$g_0 = \frac{6a_n}{\eta_R^2}\left(\frac{2\eta_R}{3} - \eta_n\right) = \frac{a_n}{\eta_R^2}\left(4\eta_R - 6\eta_n\right), \tag{17.157}$$

$$g_1 = \frac{12a_n}{\eta_R^3}\left(\eta_n - \frac{\eta_R}{2}\right) = \frac{a_n}{\eta_R^3}\left(12\eta_n - 6\eta_R\right). \tag{17.158}$$

The distribution of g at the $m+1$ time step can be determined by the above equation, but the following modifications are required due to the constraint of $g \ge 0$. The values of the thickness distribution function at category boundaries are given as follows:

$$g(\eta = 0) = \frac{6a_n}{\eta_R^2}\left(\frac{2\eta_R}{3} - \eta_n\right), \tag{17.159}$$

$$g(\eta = \eta_R) = \frac{6a_n}{\eta_R^2}\left(\eta_n - \frac{\eta_R}{3}\right). \tag{17.160}$$

Equation (17.159) gives $g(0) < 0$ when the thickness is in the right third of the thickness range or $\eta_n > 2\eta_R/3$. Equation (17.160) gives $g(\eta_R) < 0$ when the thickness is in the left third of the thickness range or $\eta_n < \eta_R/3$. Since a negative g is physically impossible, we redefine the range of the thickness distribution function. Specifically, H_R and H_L are modified as

$$\begin{cases} H_L = H_{n-1}^*, \quad H_R = 3h_{I,n}^{m+1} - 2H_{n-1}^* & \text{if } h_{I,n}^{m+1} < H_{n-1}^* + \eta_R/3 \\ H_L = H_{n-1}^*, \quad H_R = H_n^* & \text{if } H_{n-1}^* + \eta_R/3 \le h_{I,n}^{m+1} \le H_{n-1}^* + 2\eta_R/3 \\ H_L = 3h_{I,n}^{m+1} - 2H_n^*, \quad H_R = H_n^* & \text{if } H_{n-1}^* + 2\eta_R/3 < h_{I,n}^{m+1}. \end{cases} \tag{17.161}$$

The values of η_R and η_n are calculated based on these H_L and H_R, then the distribution of $g(h)$ can be obtained through (17.157) and (17.158). The conditions of $H_{n-1}^* \le H_L$ and $H_R \le H_n^*$ are satisfied.

Finally, we remap ice into the original categories using the above thickness distribution function. If $H_R > H_n$, ice is transferred from category n to $n+1$. The transferred area $\Delta a_{n,+}$ and volume $\Delta v_{n,+}$ are

$$
\begin{aligned}
H'_n =& \max(H_L, H_n) \\
\Delta a_{n,+} =& \int_{H'_n}^{H_R} g(h)dh = \int_{H'_n}^{H_R} (g_1 h - g_1 H_L + g_0)dh \\
=& (H_R - H'_n)\left\{\frac{g_1}{2}(H_R + H'_n - 2H_L) + g_0\right\}
\end{aligned}
\tag{17.162}
$$

and

$$
\begin{aligned}
\Delta v_{n,+} =& \int_{H'_n}^{H_R} hg(h)dh = \int_{H'_n}^{H_R} h(g_1 h - g_1 H_L + g_0)dh \\
=& \left(H_R - H'_n\right)\left\{\frac{g_1}{6}\left(2H_R^2 + 2H_R H'_n + 2{H'_n}^2 - 3H_R H_L - 3H'_n H_L\right) + \frac{g_0}{2}\left(H_R + H'_n\right)\right\}.
\end{aligned}
\tag{17.163}
$$

If $H_L < H_{n-1}$, ice is transferred from category n to $n-1$. The transferred area $\Delta a_{n,-}$ and volume $\Delta v_{n,-}$ are

$$
\begin{aligned}
H'_{n-1} =& \min(H_R, H_{n-1}) \\
\Delta a_{n,-} =& \int_{H_L}^{H'_{n-1}} g(h)dh = \int_{H_L}^{H'_{n-1}} (g_1 h - g_1 H_L + g_0)dh \\
=& \left(H'_{n-1} - H_L\right)\left\{\frac{g_1}{2}\left(H'_{n-1} - H_L\right) + g_0\right\}
\end{aligned}
\tag{17.164}
$$

and

$$
\begin{aligned}
\Delta v_{n,-} =& \int_{H_L}^{H'_{n-1}} hg(h)dh = \int_{H_L}^{H'_{n-1}} h(g_1 h - g_1 H_L + g_0)dh \\
=& \left(H'_{n-1} - H_L\right)\left\{\frac{g_1}{6}\left(2{H'_{n-1}}^2 - H'_{n-1} H_L - H_L^2\right) + \frac{g_0}{2}\left(H'_{n-1} + H_L\right)\right\}.
\end{aligned}
\tag{17.165}
$$

However, no redistribution among categories occurs unless Δa_n and Δv_n exceed thresholds, whether frozen or thawed.

Snow and thermal energy are also transferred in proportion to the transferred volume. For example, $\Delta v_{sn} = v_{sn}(\Delta v_{in}/v_{in})$ for snow and $\Delta e_{in} = e_{in}(\Delta v_{in}/v_{in})$ for thermal energy.

As a special situation, consider the case where category 1 sea ice is melted ($f_1 < 0$) and ice is not created in open water ($f_0 = 0$). Then, the lower boundary of category 1, H_0^*, remains zero from (17.152), while its growth rate, F_0, is f_1 from (17.149). This means that the fractional area of category 1 thinner than $\Delta h_0 = -F_0 \Delta t = -f_1 \Delta t$ is added to open water area. In this operation, volume and energy are invariant. The area to be added to open water is ¶

$$
\begin{aligned}
\Delta a_0 =& \int_0^{\Delta h_0} g(h)dh = \int_0^{\Delta h_0} (g_1 h - g_1 H_L + g_0)dh \\
=& \Delta h_0 \left\{g_1\left(\frac{\Delta h_0}{2} - H_L\right) + g_0\right\}.
\end{aligned}
\tag{17.166}
$$

The right boundary of the thickest category n_c (H_{n_c}) is a function of its mean thickness h_{n_c}. When h_{n_c} is given, H_{n_c} is computed as $H_{n_c} = 3h_{n_c} - 2H_{n_c-1}$. It is guaranteed that $g(h) > 0$ for $H_{n_c-1} < h < H_{n_c}$ and $g(h) = 0$ for $H_{n_c} < h$.

17.3.2 Categorization of sea ice newly formed in open water/lead

If the sum of $\Delta v_{\mathrm{frazil\_sea}}$ (Δv_0 in (17.93) and (17.99)) and $\Delta v_{\mathrm{frazil\_air}}$ in (17.137), Δv, is bigger than zero, that is, if ice is created in open water and leads, the new sea ice is distributed to the category to which the average height h_{new} applies.

$$
\Delta v = \Delta v_{\mathrm{frazil\_sea}} + \Delta v_{\mathrm{frazil\_air}} \tag{17.167}
$$

$$
h_{\mathrm{new}} = \Delta v / a_0. \tag{17.168}
$$

The sea ice volume Δv is added to v_n, and the following Δa is moved from a_0 to a_n.

$$
\Delta a = \min(a_0, \Delta v / h_{\min}), \tag{17.169}
$$

¶ Strictly speaking, it may be better to integrate $g(h)$ in the range of $[H_L, \Delta h_0]$, instead of $[0, \Delta h_0]$.

where $h_{\min} = 0.1$[m] is the minimum thickness of the newly-formed sea ice, and the value at the previous time is used for a_0. The thermal energy of $\Delta e_{\text{frazil\_sea}}$ (Δe_0 in (17.93) and (17.99)) and $\Delta e_{\text{frazil\_air}}$ in (17.137) is also added to the corresponding category,

$$\Delta e = \Delta e_{\text{frazil\_sea}} + \Delta e_{\text{frazil\_air}}. \tag{17.170}$$

On the other hand, the snow volume in each category does not change. ‖

17.4 Dynamics

17.4.1 Momentum equation for ice pack

The momentum equation for an ice pack with mass $\rho_I A h_I$ is

$$\rho_I \frac{\partial}{\partial t}(A h_I u_I) - \rho_I A h_I f v_I = -\rho_I A h_I g \frac{1}{h_\mu}\frac{\partial h}{\partial \mu} + F_\mu(\sigma) + A(\tau_{AI_x} + \tau_{IO_x}), \tag{17.171}$$

$$\rho_I \frac{\partial}{\partial t}(A h_I v_I) + \rho_I A h_I f u_I = -\rho_I A h_I g \frac{1}{h_\psi}\frac{\partial h}{\partial \psi} + F_\psi(\sigma) + A(\tau_{AI_y} + \tau_{IO_y}), \tag{17.172}$$

where (u_I, v_I) is the velocity vector, h is the sea surface height, (F_μ, F_ψ) is the ice's internal stress (which is a function of internal stress tensor (σ)), and $\overrightarrow{\tau_{AI}}$ and $\overrightarrow{\tau_{IO}}$ are stresses exerted by the atmosphere and ocean.

17.4.2 Stresses at top and bottom

The stress at the top is wind stress:

$$\overrightarrow{\tau_{AI}} = C_{DAI}\rho_a |\mathbf{U}_a - \mathbf{u}_I| [(\mathbf{U}_a - \mathbf{u}_I)\cos\theta_a + \mathbf{k} \times (\mathbf{U}_a - \mathbf{u}_I)\sin\theta_a], \tag{17.173}$$

where $\mathbf{U}_a$ is the surface wind vector, C_{DAI} is the bulk transfer coefficient between air and ice, ρ_a is the density of air, and θ_a is the angle between the wind vector and the ice drift vector, which is set to zero in MRI.COM, since 10-m wind is generally similar in direction to the surface stress (Lepparanta, 2011).

Stress at the bottom is ocean stress:

$$\overrightarrow{\tau_{IO}} = C_{DIO}\rho_o |\mathbf{U}_w - \mathbf{u}_I| [(\mathbf{U}_w - \mathbf{u}_I)\cos\theta_o + \mathbf{k} \times (\mathbf{U}_w - \mathbf{u}_I)\sin\theta_o], \tag{17.174}$$

where $\mathbf{U}_w$ is the velocity of the first level of the ocean model, C_{DIO} is the bulk transfer coefficient between the ice and ocean, ρ_o is the density of sea water, and θ_o is the angle between the ice drift vector and the surface velocity of the ocean. The angle θ_o is set to cz in the northern and $-cz$ in the southern hemisphere, where z is the depth of the first layer (velocity point) and c is a constant of $1^\circ\ \mathrm{m}^{-1}$.

17.4.3 Internal stress

In a highly concentrated icepack, the effect of the internal stress is as large as the Coriolis effect and the surface stresses. The expression of the internal stress is derived by regarding the ice as continuous media. The elastic-plastic-viscous (EVP) model by Hunke and Dukowicz (1997, 2002) is adopted for the constitutive law (the relation between stress and strain rate). The EVP model is a computationally efficient modification of the viscous-plastic (VP) model (Hibler, 1979). In the VP model, the internal stress could be very large when the concentration is high and strain rate is near zero, which makes the explicit integration infeasible. An alternative, the implicit method, is usually adopted, but it is not suitable for parallel computing. The EVP model treats the ice as an elastic medium and a large local force is released by elastic waves, which would be damped within the time scale of the wind forcing.

The constitutive law of the EVP model is

$$\frac{1}{E}\frac{\partial \sigma_{ij}}{\partial t} + \frac{1}{2\eta}\sigma_{ij} + \frac{\eta - \zeta}{4\eta\zeta}\sigma_{kk}\delta_{ij} + \frac{P}{4\zeta}\delta_{ij} = \dot{\epsilon}_{ij}, \quad i, j = 1, 2, \tag{17.175}$$

where ζ and η are viscous parameters, P represents ice strength, and E is an elastic parameter (mimics Young's modulus). In the VP model, tendency terms are zero.

‖ The average snow depth can change due to change in area.

The r.h.s. ($\dot{\epsilon}_{ij}$) is the strain rate tensor, expressed in Cartesian coordinates as:

$$\dot{\epsilon}_{ij} = \frac{1}{2}\left(\frac{\partial u_{Ii}}{\partial x_j} + \frac{\partial u_{Ij}}{\partial x_i}\right). \tag{17.176}$$

The divergence, tension, and shear of the strain rate are defined as follows:

$$D_D = \dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \quad D_T = \dot{\epsilon}_{11} - \dot{\epsilon}_{22}, \quad D_S = 2\dot{\epsilon}_{12}. \tag{17.177}$$

The equation for the stress tensor for $\sigma_1 = \sigma_{11} + \sigma_{22}$ and $\sigma_2 = \sigma_{11} - \sigma_{22}$ is given by

$$\frac{1}{E}\frac{\partial \sigma_1}{\partial t} + \frac{\sigma_1}{2\zeta} + \frac{P}{2\zeta} = D_D, \tag{17.178}$$

$$\frac{1}{E}\frac{\partial \sigma_2}{\partial t} + \frac{\sigma_2}{2\eta} = D_T, \tag{17.179}$$

$$\frac{1}{E}\frac{\partial \sigma_{12}}{\partial t} + \frac{\sigma_{12}}{2\eta} = \frac{1}{2}D_S. \tag{17.180}$$

In generalized orthogonal coordinates, divergence, tension, and shear of the strain rate are expressed by

$$D_D = \frac{1}{h_\mu h_\psi}\left[\frac{\partial(h_\psi u_I)}{\partial \mu} + \frac{\partial(h_\mu v_I)}{\partial \psi}\right], \tag{17.181}$$

$$D_T = \frac{h_\psi}{h_\mu}\frac{\partial}{\partial \mu}\left(\frac{u_I}{h_\psi}\right) - \frac{h_\mu}{h_\psi}\frac{\partial}{\partial \psi}\left(\frac{v_I}{h_\mu}\right), \tag{17.182}$$

$$D_S = \frac{h_\mu}{h_\psi}\frac{\partial}{\partial \psi}\left(\frac{u_I}{h_\mu}\right) + \frac{h_\psi}{h_\mu}\frac{\partial}{\partial \mu}\left(\frac{v_I}{h_\psi}\right). \tag{17.183}$$

The internal stress is obtained as the divergence of the internal stress tensor,

$$F_\mu = \frac{1}{2}\left[\frac{1}{h_\mu}\frac{\partial \sigma_1}{\partial \mu} + \frac{1}{h_\mu h_\psi^2}\frac{\partial(h_\psi^2 \sigma_2)}{\partial \mu} + \frac{2}{h_\mu^2 h_\psi}\frac{\partial}{\partial \psi}\left(h_\mu^2 \sigma_{12}\right)\right], \tag{17.184}$$

$$F_\psi = \frac{1}{2}\left[\frac{1}{h_\psi}\frac{\partial \sigma_1}{\partial \psi} - \frac{1}{h_\mu^2 h_\psi}\frac{\partial(h_\mu^2 \sigma_2)}{\partial \psi} + \frac{2}{h_\mu h_\psi^2}\frac{\partial}{\partial \mu}\left(h_\psi^2 \sigma_{12}\right)\right]. \tag{17.185}$$

The viscous parameters are obtained from the ice strength and velocity as follows:

$$\zeta = \frac{P}{2\Delta}, \tag{17.186}$$

$$\eta = \frac{P}{2e^2\Delta}, \tag{17.187}$$

$$\Delta = \left[D_D^2 + \frac{1}{e^2}(D_T^2 + D_S^2)\right]^{1/2}. \tag{17.188}$$

The pressure (strength) of the ice is a function of ice concentration and thickness:

$$P = P^* A h_I e \exp[-c^*(1 - A)], \tag{17.189}$$

where P^* is the scaling factor for pressure, c^* is a parameter that defines the dependency on concentration, and e is the axis ratio of the elliptic yield curve ($e = 2$).

The elastic parameter E is given by

$$E = \frac{2E_o \rho_I A h_I}{\Delta t_e^2}\min(\Delta x^2, \Delta y^2), \tag{17.190}$$

where E_o is a tuning factor that satisfies $0 < E_o < 1$, Δt_e is the time step for ice dynamics, and Δx and Δy are the zonal and meridional grid widths.

17.4.4 Boundary conditions

Surface stresses on the ice are exerted for the fractional area of the ice within a grid cell. The ice concentration is multiplied by the wind and ocean stresses.

For the stress on the ocean, the ice-ocean stress is exerted for the ice-covered area, and the wind stress is exerted for the open water area:

$$\nu_V \left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z} \right) \bigg|_{k=0} = -\frac{A}{\rho_o} (\tau_{IO_x}, \tau_{IO_y}) + \frac{(1-A)}{\rho_o} (\tau_{AO_x}, \tau_{AO_y}). \tag{17.191}$$

Note that $(\tau_{IO_x}, \tau_{IO_y})$ is reversed in sign because it is defined by (17.174) as the stress on the ice.

17.4.5 Solution procedure

Given the surface wind vector and the surface velocity of the ocean needed to compute surface stresses, the momentum equations ((17.171) and (17.172)) and the equations for stress tensors ((17.178), (17.179), and (17.180)) are solved.

First, the stress tensor is computed using the equations for stress tensors, the momentum equation is then solved using the stress tensor. Basically, the implicit method is used for prognostic variables for each equation. For example, stress tensor σ_1 is solved for σ_1^{m+1} as follows:

$$\frac{1}{E} \frac{\sigma_1^{m+1} - \sigma_1^m}{\Delta t} + \frac{\sigma_1^{m+1}}{2\zeta^m} + \frac{P}{2\zeta^m} = D_D^m. \tag{17.192}$$

Note that strain rate tensors and viscous parameters are updated every time step using a new velocity.

The momentum equations are solved using σ^{m+1} above:

$$\begin{aligned} \rho_I A h_I \frac{u_I^{m+1} - u_I^m}{\Delta t} =& \rho_I A h_I f v_I^{m+1} - \rho_I A h_I g \frac{1}{h_\mu} \frac{\partial h}{\partial \mu} + F_\mu(\sigma^{m+1}) + A \tau_{AIx} \\ &+ A C_{DAI} \rho_o |\mathbf{U}_w - \mathbf{u}_I^m| [(U_w - u_I^{m+1}) \cos\theta_o - (V_w - v_I^{m+1}) \sin\theta_o], \end{aligned} \tag{17.193}$$

$$\begin{aligned} \rho_I A h_I \frac{v_I^{m+1} - v_I^m}{\Delta t} =& -\rho_I A h_I f u_I^{m+1} - \rho_I A h_I g \frac{1}{h_\psi} \frac{\partial h}{\partial \psi} + F_\psi(\sigma^{m+1}) + A \tau_{AIy} \\ &+ A C_{DAI} \rho_o |\mathbf{U}_w - \mathbf{u}_I^m| [(V_w - v_I^{m+1}) \cos\theta_o + (U_w - u_I^{m+1}) \sin\theta_o]. \end{aligned} \tag{17.194}$$

Note that the surface velocity of the ocean (U_w, V_w) is constant during the integration. The starting time level of the ocean model is used, $n-1$ for the leap-frog time step, and n for the Matsuno scheme. For the leap-frog time step of the ocean model,

$$\overrightarrow{\tau_{IO}} = C_{DAI} \rho_o |\mathbf{U}_w^{n-1} - \mathbf{u}_I^m| [(\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}) \cos\theta_o + \mathbf{k} \times (\mathbf{U}_w^{n-1} - \mathbf{u}_I^{m+1}) \sin\theta_o]. \tag{17.195}$$

The time step of the ice dynamics is limited by the phase speed of the elastic wave. To damp the elastic waves during the sub-cycle, the ice dynamics is sub-cycled several tens of steps during one ocean model time step.

17.5 Advection and Diffusion

Fractional area (for both sea ice and open ocean), snow volume, ice volume, ice energy, and ice surface temperature (optional; set `flg_advec_tskin = .true.` in `mod_seaice_cat.F90`) of each category are advected. A multidimensional positive definite advection transport algorithm (MPDATA; Smolarkiewicz, 1984) is used. For all quantities which are advected, the harmonic-type diffusion can be applied to remove noises. In MRI.COM, advection and diffusion are solved serially as follows.

The advection equation for a property (α) is given by

$$\frac{\partial \alpha}{\partial t} + \frac{1}{h_\mu h_\psi} \left\{ \frac{\partial (h_\psi u \alpha)}{\partial \mu} + \frac{\partial (h_\mu v \alpha)}{\partial \psi} \right\} = 0. \tag{17.196}$$

The specific representation for α is a_0 for open water fraction, a_n for fractional area, v_n for ice volume per grid area, v_s for snow volume per grid area, and e_n for ice energy.

In MPDATA, (17.196) is first solved to obtain a temporary value (α^*) using the upstream scheme with a mid-point velocity between time levels n and $n+1$. Using this temporary value, an anti-diffusive velocity is computed as

$$
\begin{aligned}
\tilde{u}_{i+\frac{1}{2},j} &= \frac{1}{\alpha^*}\left[\frac{1}{2}\left(|u^{n+\frac{1}{2}}|\Delta x - \Delta t (u^{n+\frac{1}{2}})^2\right)\frac{\partial \alpha^*}{\partial x} - \frac{1}{2}\Delta t u^{n+\frac{1}{2}} v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial y}\right] \\
&\quad - \frac{1}{2}\Delta t u^{n+\frac{1}{2}}\left(\frac{\partial u}{\partial x}^{n+\frac{1}{2}} + \frac{\partial v}{\partial y}^{n+\frac{1}{2}}\right), && (17.197) \\
\tilde{v}_{i,j+\frac{1}{2}} &= \frac{1}{\alpha^*}\left[\frac{1}{2}\left(|v^{n+\frac{1}{2}}|\Delta y - \Delta t (v^{n+\frac{1}{2}})^2\right)\frac{\partial \alpha^*}{\partial y} - \frac{1}{2}\Delta t u^{n+\frac{1}{2}} v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial x}\right] \\
&\quad - \frac{1}{2}\Delta t v^{n+\frac{1}{2}}\left(\frac{\partial u}{\partial x}^{n+\frac{1}{2}} + \frac{\partial v}{\partial y}^{n+\frac{1}{2}}\right). && (17.198)
\end{aligned}
$$

This velocity is used to compute a new value using the upstream scheme. See Section 17.13.1 for the discretized form.

Since MPDATA is positive definite, the new area and thickness should be positive. If the sum of the fractional area exceeds one, the ridging scheme will adjust the fractional area. Since energy is negative definite, the sign is reversed just before advection and returned to a negative value after the advection.

Next, the diffusion equation is solved for the distribution modified by advection. The specific form of the harmonic-type diffusion is represented as follows:

$$
\begin{aligned}
\frac{\partial \alpha}{\partial t} &= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial \mu}\left(\frac{h_\psi \kappa_H}{h_\mu}\frac{\partial \alpha}{\partial \mu}\right) + \frac{\partial}{\partial \psi}\left(\frac{h_\mu \kappa_H}{h_\psi}\frac{\partial \alpha}{\partial \psi}\right)\right\} \\
&= -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial F_\mu^\alpha}{\partial \mu} + \frac{\partial F_\psi^\alpha}{\partial \psi}\right\} && (17.199) \\
F_\mu^\alpha &= -\frac{h_\psi \kappa_H}{h_\mu}\frac{\partial \alpha}{\partial \mu} && (17.200) \\
F_\psi^\alpha &= -\frac{h_\mu \kappa_H}{h_\psi}\frac{\partial \alpha}{\partial \psi}, && (17.201)
\end{aligned}
$$

where κ_H is the horizontal diffusion coefficients (required parameter, `diff_seaice_m2ps`). The diffusion term is advanced in time using the flux form (17.199), which are calculated for α modified by advection. If the diffusion CFL condition is met, the diffusion acts to smooth the input distribution, so that the output value does not deviate from the domain of the input value. See Section 17.13.2 for the discretized form. (Though MPDATA is positive difinite, it may take a negative value due to a numerical error. If a_n becomes negative, state variables are modified considering that there is no sea ice in the category ($a_n = 0$).

17.6 Ridging

As a result of advection, the sum of the fractional area might exceed one, especially where the velocity field is convergent. In such a case, it is assumed that ridging occurs among ice to yield a sum equal to or less than one. Even if the sum is less than one, ridging or rafting might occur where the concentration is high.

The ridging scheme of MRI.COM follows that of CICE (originally based on Thorndike et al. (1975)), which is briefly summarized in this section. In this scheme, the change rate in ice area fraction due to ridging (total of the categories) is determined by the strain calculated in the dynamics scheme. This scheme decides how to distribute the amount of change to each category, and redistributes the main variables of each category (i.e. a_0, a_n, v_n, v_s and e_n) based on it.

First, in preparation for the scheme, we define the cumulative function $G(h)$ of ice area fraction:

$$G(h) = \int_0^h g(h')dh', \tag{17.202}$$

and $G(h)$ indicates the area of ice thinner than h including open water fraction (a_0). The boundary value for the discrete category, G_n, is given by

$$G_n = \begin{cases} G(H_n) & \text{for } n \geq 1 \\ a_0 & \text{for } n = 0 \end{cases} \tag{17.203}$$

The property G_n is the area fraction summed from category 0 to n. As a result of advection, $A + a_0 = \sum_{n=0}^{N_c} a_n = 1$ is not guaranteed, so MRI.COM adjusts it so that $G_{N_c} = 1$.

First, we determine a fractional area that undergoes ridging: $a_P(h) = b(h)g(h)$. The weighting function $b(h)$ is chosen such that the ridging occurs for thin ice:

$$b(h) = \begin{cases} \dfrac{2}{G^*}\left(1 - \dfrac{G(h)}{G^*}\right) & \text{if } G(h) < G^* \\ 0 & \text{otherwise} \end{cases} \tag{17.204}$$

where $G(h)$ is the area fraction of ice thinner than h and G^* is an empirical constant with $G^* = 0.15$. The participation function for category n (a_{P_n}) is obtained by integrating $a_P(h)$ for a range $[H_{n-1},\ H_n]$ as

$$a_{P_n} = \int_{H_{n-1}}^{H_n} b(h)g(h)dh = \frac{2}{G^*}(G_n - G_{n-1})\left(1 - \frac{G_{n-1} + G_n}{2G^*}\right), \tag{17.205}$$

where $dG = gdh$ is used. The property a_{P_n} is the fractional contribution from the category n among the total area of ice subject to ridging. This equation is used for the category that satisfies $G_n < G^*$. If $G_{n-1} < G^* < G_n$, then G^* is replaced by G_n. If $G_{n-1} > G^*$, then $a_{P_n} = 0$. If the fractional area of open leads exceeds G^* ($a_0 > G^*$), then ridging does not occur. In (17.205), a_{P_0} is open leads' allocation of the ridging effect, and means closing of open leads due to flow convergence,

$$a_{P_0} = \frac{2}{G^*}G_0\left(1 - \frac{G_0}{2G^*}\right). \tag{17.206}$$

Opening due to ridging will be discussed later.

Ridging occurs such that the total area is reduced while conserving ice volume and energy. It is assumed that ice of thickness h_n is homogeneously distributed between $H_{\min} = 2h_n$ and $H_{\max} = 2\sqrt{H^* h_n}$ after ridging, where $H^* = 30\,\mathrm{m}$ (as a default). The thickness ratio before and after the ridging is $k_n = (H_{\min} + H_{\max})/(2h_n)$. Therefore, when an area of category n is reduced by ridging at a rate r_n, the area of thicker categories is increased by r_n/k_n.

Among the new ridges, the fractional area that is distributed in category m is:

$$f_m^{\text{area}} = \frac{H_R - H_L}{H_{\max} - H_{\min}}, \tag{17.207}$$

where $H_L = \max(H_{m-1}, H_{\min})$ and $H_R = \min(H_m, H_{\max})$. The fractional volume that is distributed in category m is:

$$f_m^{\text{vol}} = \frac{(H_R)^2 - (H_L)^2}{(H_{\max})^2 - (H_{\min})^2}. \tag{17.208}$$

The snow volume and ice energy are distributed by the same ratio as the ice volume.

The net area lost by ridging and open water closing is assumed to be a function of the strain rates. The net rate of area loss of the ice pack (R_{net}) [unit: 1/s] is given by

$$R_{\text{net}} = \frac{C_s}{2}(\Delta - |D_D|) - \min(D_D, 0), \tag{17.209}$$

where C_s is the fraction of shear dissipation energy that contributes to ridge building (0.5 is used in MRI.COM), D_D is the divergence, and $\Delta = \left[D_D^2 + \frac{1}{e^2}(D_T^2 + D_S^2)\right]^{1/2}$. These strain rates are computed by the dynamics scheme.

The total rate of area loss due to ridging, $R_{\text{tot}} = \sum_{n=0}^{N_c} r_n$, is related to the net rate as follows:

$$R_{\text{net}} = r_0 + \sum_{n=1}^{N}\left(r_n - \frac{r_n}{k_n}\right) = \left[a_{P_0} + \sum_{n=1}^{N} a_{P_n}\left(1 - \frac{1}{k_n}\right)\right]R_{\text{tot}}, \tag{17.210}$$

where, r_0 indicates convergence or divergence of the open leads fraction, a_0. Since R_{net} is computed from (17.209), R_{tot} is computed from (17.210).

Based on the above discussion, the amounts of change due to ridging during the time step interval, Δt, is calculated. The decrease of area fraction for category n, a_r^n, is given by

$$a_r^n = r_n\Delta t = a_{P_n} R_{\text{tot}}\Delta t \qquad (n = 0 \sim N_c). \tag{17.211}$$

Taking into account the area reduction due to ridging ($1/k_n$), the area fraction transfer from category n to m, $\Delta a_{n,m}$, is

$$\Delta a_{n,m} = f_m^{\text{area}}\frac{a_r^n}{k_n} \qquad (m = 1 \sim N_c, n = 1 \sim m). \tag{17.212}$$

Note that transfers between the same category ($a_{m,m}$) can also exist. Similarly, the sea ice volume transfer, $\Delta v_{n,m}$, the snow volume transfer, $\Delta v_{s,n,m}$, and the enthalpy transfer, $\Delta e_{n,n}$, are

$$c_n = a_r^n / a_n \tag{17.213}$$
$$\Delta v_{n,m} = f_m^{\mathrm{vol}} a_r^n h_I = f_m^{\mathrm{vol}} c_n v_n \tag{17.214}$$
$$\Delta v_{s,n,m} = f_m^{\mathrm{vol}} a_r^n h_s = f_m^{\mathrm{vol}} c_n v_s \tag{17.215}$$
$$\Delta e_{n,m} = f_m^{\mathrm{vol}} a_r^n (e_n / a_n) = f_m^{\mathrm{vol}} c_n e_n. \tag{17.216}$$

Redistribution by ridging is calculated by the combination of the above transfer amounts. The area fraction decrease for category $n (= 1 \sim N_c)$, Δa_-^n, and the increase for category $m (= 1 \sim N_c)$, Δa_+^m, are

$$\Delta a_-^n = a_r^n \tag{17.217}$$
$$\Delta a_+^m = \sum_{n=1}^{m} \Delta a_{n,m}. \tag{17.218}$$

The area fraction decrease of open leads, Δa_-^0, and the increase, Δa_+^0, are

$$\Delta a_-^0 = a_r^0 \tag{17.219}$$
$$\Delta a_+^0 = R_{\mathrm{net}} \Delta t. \tag{17.220}$$

In response to area fraction change, the decreases of v_n, v_s and e_n for category n, Δv_-^n, Δv_{s-}^n and Δe_-^n are

$$\Delta v_-^n = \begin{cases} \displaystyle\sum_{m=n}^{N_c} \Delta v_{n,m} & \text{for } n \le N_c - 1 \\ 0 & \text{for } n = N_c \end{cases} \tag{17.221}$$

$$\Delta v_{s-}^n = \begin{cases} \displaystyle\sum_{m=n}^{N_c} \Delta v_{s,n,m} & \text{for } n \le N_c - 1 \\ 0 & \text{for } n = N_c \end{cases} \tag{17.222}$$

$$\Delta e_-^n = \begin{cases} \displaystyle\sum_{m=n}^{N_c} \Delta e_{n,m} & \text{for } n \le N_c - 1 \\ 0 & \text{for } n = N_c \end{cases} \tag{17.223}$$

Note that treatment at $n = N_c$ is different from a_n. Similarly, the increases of v_n, v_s and e_n for category m, Δv_+^m, Δv_{s+}^m and Δe_+^m are

$$\Delta v_+^m = \begin{cases} \displaystyle\sum_{n=1}^{m} \Delta v_{n,m} & \text{for } m \le N_c - 1 \\ \displaystyle\sum_{n=1}^{N_c-1} \Delta v_{n,m} & \text{for } m = N_c \end{cases} \tag{17.224}$$

$$\Delta v_{s+}^m = \begin{cases} \displaystyle\sum_{n=1}^{m} \Delta v_{s,n,m} & \text{for } m \le N_c - 1 \\ \displaystyle\sum_{n=1}^{N_c-1} \Delta v_{s,n,m} & \text{for } m = N_c \end{cases} \tag{17.225}$$

$$\Delta v_+^m = \begin{cases} \displaystyle\sum_{n=1}^{m} \Delta e_{n,m} & \text{for } m \le N_c - 1 \\ \displaystyle\sum_{n=1}^{N_c-1} \Delta e_{n,m} & \text{for } m = N_c \end{cases} \tag{17.226}$$

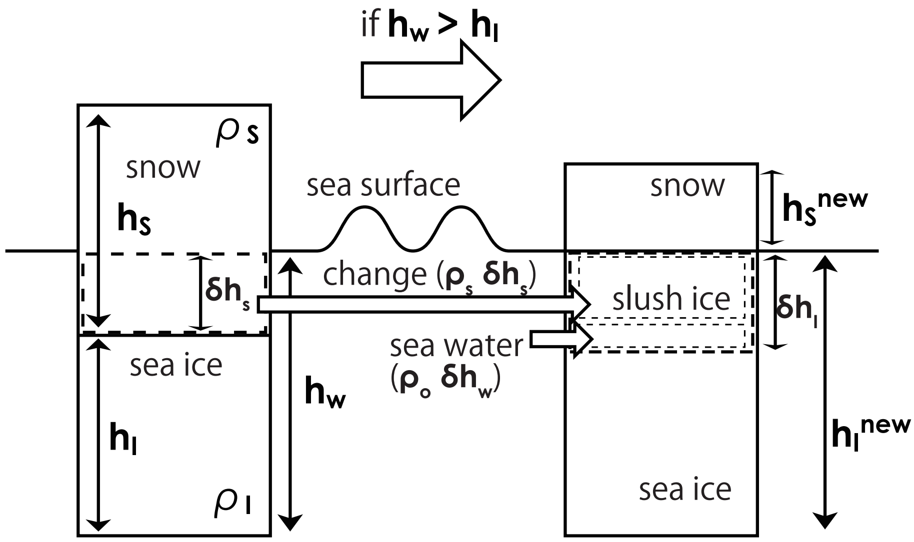

Figure17.6 A diagram of the modeled process in which the drafted part of the piled snow changes to sea ice (slush).

The notation of the loop is different for addition and subtraction, but it can be integrated in the same loop. Using these, the state variables after redistribution by ridging, a_n^{new}, v_n^{new}, v_s^{new} and e_n^{new} are

$$a_n^{\mathrm{new}} = a_n - \Delta a_-^n + \Delta a_+^n \qquad (n = 0 \sim N_c) \tag{17.227}$$

$$v_n^{\mathrm{new}} = v_n - \Delta v_-^n + \Delta v_+^n \qquad (n = 1 \sim N_c) \tag{17.228}$$

$$v_s^{\mathrm{new}} = v_s - \Delta v_{s-}^n + \Delta v_{s+}^n \qquad (n = 1 \sim N_c) \tag{17.229}$$

$$e_n^{\mathrm{new}} = e_n - \Delta e_-^n + \Delta e_+^n \qquad (n = 1 \sim N_c) \tag{17.230}$$

(Note that the category notation for incease is changed from m to n.)

In practice, we require that $a_r^n \leq a_n$ to avoid negative area fraction. If $A + a_0 > 1$ after ridging, R_{net} is adjusted to yield $A + a_0 = 1$, and the ridging procedure is repeated.

17.7 Transformation from snow to sea ice (slush ice formation)

When piled snow on sea ice is drafted, seawater permeates the drafted part and slush ice is formed. In the model, this process is expressed as shown in Fig. 17.6, and slush ice formation is treated as increase in sea ice (an extension of Hunke and Lipscomb (2006)). For the model stability, this process is calculated for each category after sea ice ridging. In this section, the case of variable sea ice salinity is described first, and then the case of fixed salinity is explained.

First, whether or not piled snow has drafted is determined as follows. The draft depth of sea ice, h_w, satisfies the following relation from the Archimedes' principle, assuming an equilibrium state,

$$\rho_I h_I + \rho_s h_s = \rho_o h_w, \tag{17.231}$$

thus

$$h_w = \frac{\rho_s}{\rho_o} h_s + \frac{\rho_I}{\rho_o} h_I. \tag{17.232}$$

Using the values at the next time step, which have been calculated, h_w and h_I are compared, and if $h_w > h_I$, that is,

$$h_s > \frac{\rho_o - \rho_I}{\rho_s} h_I, \tag{17.233}$$

it is judged that snow has drafted.

Next, as the thickness of the piled snow that changes to slush ice is written as δh_s and the thickness originated to sea water as δh_w, the masses of the snow and sea water in the slush ice are $\rho_s \delta h_s$ and $\rho_o \delta h_w$, respectively. Then, the mass ratio of sea water in the slush is

$$r_w = \rho_o \delta h_w / (\rho_o \delta h_w + \rho_s \delta h_s). \tag{17.234}$$

In the model, r_w is assumed as a constant value of 0.15 (Sturn and Masson, 2010). Since the sum of the decreased snow mass and sea water mass is equal to the mass of the newly formed sea ice,

$$\rho_I \delta h_I = \rho_s \delta h_s + \rho_o \delta h_w, \tag{17.235}$$

where the increased sea ice thickness is written by δh_I. Further, assuming that the buoyancy equilibrium is established even after the transformation, the following relationship is obtained.

$$\rho_o(h_I + \delta h_I) = \rho_I(h_I + \delta h_I) + \rho_s(h_s - \delta h_s). \tag{17.236}$$

Using the above three equations, $\delta h_I, \delta h_s$ and δh_w can be obtained as

$$\delta h_I = \frac{\rho_s h_s - (\rho_o - \rho_I)h_I}{\rho_o - r_w \rho_I}, \tag{17.237}$$

$$\delta h_s = (1 - r_w)\frac{\rho_I}{\rho_s}\delta h_I, \tag{17.238}$$

$$\delta h_w = r_w \frac{\rho_I}{\rho_o}\delta h_I. \tag{17.239}$$

In the model, the equations for the sea ice and snow volumes are derived by using the relations, $v_s = a_n h_s$ and $v_n = a_n h_I$, based on the above equations about one-dimensional heights. First, the draft judgment is

$$v_s > \frac{\rho_o - \rho_I}{\rho_s} v_n. \tag{17.240}$$

After some deformation, the increase of sea ice volume, Δv_n, the decrease of piled snow volume, Δv_s, and the increase of integrated sea ice salinity (referred by (17.259)), ΔF^S, for each category are obtained

$$\Delta v_n = a_n \delta h_I = \frac{\rho_s v_s - (\rho_o - \rho_I)v_n}{\rho_o - r_w \rho_I}, \tag{17.241}$$

$$\Delta v_s = a_n \delta h_s = (1 - r_w)\frac{\rho_I}{\rho_s}\Delta v_n, \tag{17.242}$$

$$\Delta F^S_{slush} = a_n \frac{\rho_o}{\rho_I} S_w \delta h_w = r_w \Delta v_n S_w, \tag{17.243}$$

where S_w is sea surface salinity. Thus, salinity in the newly formed slush ice, S_I, is

$$S_I = \Delta F^S_{slush}/\Delta v_n = r_w S_w. \tag{17.244}$$

Therefore, assuming that temperature of the slush ice is the freezing point, mS_I, the increase of sea ice enthalpy, Δe_n, can be given by

$$\Delta e_n = a_n E_S(mS_I, S_I)\delta h_I = E_S(mS_I, S_I)\Delta v_n, \tag{17.245}$$

where E_S is given by (17.14). In slush ice formation, latent heat cooling is required for sea water with a temperature of T_w to freeze, but the latent heat can be considered to be released to the atmosphere rather than exchange with the ocean. This is because the formation of slush ice under snow cover releases latent heat to the atmosphere through voids in the snow cover.

The upper and lower limits of the snow volume change, Δv_s, are shown here. From (17.240), (17.241), (17.242), $v_n > 0$ and $\rho_o > \rho_I$,

$$0 \le \Delta v_s < v_s. \tag{17.246}$$

Therefore, the modified snow volume, $v_s - \Delta v_s$, must not be negative. Since $\Delta v_n \ge 0$, sea ice must not decrease by this process.

This process causes water transport between the ocean and sea ice models. The water volume transport, W_{slush}, is

$$W_{slush} = \sum_n a_n \delta h_w/\Delta t = \frac{\rho_I}{\rho_o}\frac{r_w}{\Delta t}\sum_n \Delta v_n, \tag{17.247}$$

using the volume of seawater as a unit (positive upward). The temperature transport during a time step interval, F^θ_{slush} [$^\circ$ C m], is

$$F^\theta_{slush} = \frac{\rho_o}{\rho_I}\sum_n (T_w a_n \delta h_w) = T_w r_w \sum_n \Delta v_n, \tag{17.248}$$

where the sea ice volume is used as a unit to be consistent with other temperature transports. See (17.274) for the salinity transport flux.

At the end of this section, the case where sea ice salinity is fixed as $S_I = 4.0$ psu is explained (v5.0). First, $r_w = 0$ is used to consider only the change from piled snow to sea ice. The change amounts, (17.241), (17.242) and (17.245), can be used without modification. However, salinity is fixed so that

$$\Delta F^S_{slush} = S_I \Delta v_n. \tag{17.249}$$

Salinity in slush ice must be compensated from the first layer of the ocean model:

$$S^{\mathrm{new}} = S^{\mathrm{old}} - S_I \frac{\Delta v_n(\rho_I/\rho_o)}{\Delta z_1}. \tag{17.250}$$

17.8 Adjustment

As the final procedure in each time step, the following adjustment is applied to the sea ice distribution.

- If the thickness of sea ice of some category is out of its category bounds, the sea ice in that category is moved to the appropriate category.
- If the thickness of sea ice in category 1 is less than 0.1 meter, the thickness is set to 0.1 meter and the area fraction is adjusted accordingly.
- If the fractional area that sea ice occupies is too small, the sea ice of that category is forcibly melted (See Table 17.10).
- If sea ice temperature is above the freezing point, it melts.
- If ice area fraction is outside the definition range, the fraction is corrected to recalculate diagnostic variables (Section 17.1.5)

17.9 Fluxes from sea ice to ocean

The processes of exchanging physical quantities between the sea ice and ocean models can be roughly classified into two types, except for momentum. One is exchanges accompanied with water transport (mainly, melting and freezing of sea ice), by which water volume (I), temperature (F^θ_I) and salinity (F^S_I) are transported (Section 14.6.2). The other is heat flux (F^T) and salinity flux (F^S_{IO}) without water transport. Fluxes generated by various processes in the sea ice model are merged to the above five fluxes at the end of the sea ice model procedure, and they are handed over to the ocean model.

The fresh water transport, I, is

$$I = W_{FR} + W_O + W_{ROice} + W_{adjust} + W_{slush} + W_{ROsnow}, \tag{17.251}$$

where W_{FR} is given by (17.102), W_O by (17.139), W_{ROice} by (17.72) and (17.89), W_{ROsnow} by (17.73) and (17.140), and W_{slush} by (17.247) (see Fig. 17.3). The water flux W_{adjust} is due to the adjustment process of Sec. 17.8, which melts small sea ice below threshold.

The temperature transport, F^θ_I, is expressed by the sum of two components, the upward transport from ocean to sea ice, F^θ_{Iup}, and the downward transport from sea ice to ocean, F^θ_{Idown}, as follows

$$F^\theta_I = F^\theta_{Iup} + F^\theta_{Idown}, \tag{17.252}$$

$$F^\theta_{Iup} = F^\theta_{frzl} + F^\theta_{btm\_frz} + F^\theta_{slush} \tag{17.253}$$

$$F^\theta_{Idown} = F^\theta_{top} + F^\theta_{intr} + F^\theta_{btm\_melt} + F^\theta_{adjust}, \tag{17.254}$$

where F^θ_{top} is given by (17.74), F^θ_{intr} by (17.91), F^θ_{frzl} by (17.103), $F^\theta_{btm\_frz}$ by (17.141), $F^\theta_{btm\_melt}$ by (17.142), and F^θ_{slush} by (17.248). Note that the right hand side is all calculated by the sea ice volume. When passed to the ocean model, it is converted to sea water volume by multiplying by ρ_I/ρ_o.

The salinity transport, F^S_I, is

$$F^S_I = (W_{FR} + W_O + W_{ROice} + W_{adjust})S_I. \tag{17.255}$$

In v5.0, the sea water exchanged with ice is assumed to have a constant salinity ($S_I = 4.0$ pss). Note that the temperature and salinity transports by snow melting, W_{ROsnow}, are always zero because snow is assumed to be 0° C and 0 psu.

The heat flux that is not involved in water exchange is

$$F_T = Q_{SO} + Q_{LO} + (1 - A)F_{T_L} + \sum_n a_n F_{T_I} + F_{adjust}, \tag{17.256}$$

where Q_{SO} is the latent heat flux due to melting sea ice or snow (17.87). The flux F_T is handed over to the ocean model, and added to the sea surface heat flux, Q_{OTHER} in (14.37).

The salinity flux without water transport is given by (17.95). This is added to the sea surface salinity flux of the ocean model, (14.41).

Other impacts on the ocean model.

1. There is a salinity flux from the ocean to the sea ice in transformation from snow to sea ice, (17.250), but the model treats it specially.
2. The rise in sea water temperature due to freezing of supercooled sea water (Sec. 17.2.3) is also treated specially, and converted to the heat flux in the ocean model (Sec. 14.7.2).

In v5.1, they are treated in a unified manner as the above five fluxes between the sea ice and ocean models.

17.10 Variable salinity of sea ice (v5.1)

Sea ice salinity has been treated as a constant value of 4.0 psu, but an option to make it variable was introduced in v5.1. This section describes the governing equation of sea ice salinity for this option.

17.10.1 Prognostic variable

The prognostic variable to be solved is sea ice salinity integrated for each category n per unit area, $S_v(n)$ [psu m^3 m^{-2}]. Similar to the expression (17.3), it is defined by

$$S_v(n) = \int_{H_{n-1}}^{H_n} Sg(h)hdh. \tag{17.257}$$

In the following, we will not go into the details of salinity S changing spatially in sea ice, but directly consider the time evolution of $S_v(n)$ as a total amount. When there is no sea ice for category n, i.e. $a_n = 0$, $S_v(n)$ is set zero, and $S_v(n) \geq 0$ is alway satisfied. Salinity in snow is always zero.

17.10.2 The governing equation

The integrated salinity is governed by the following equation

$$\frac{dS_v(n)}{dt} = F_S(n) + F_{drain}(n) + \mathcal{R}_{\mathrm{remap}}(n) + \mathcal{A}(n) + \mathcal{R}_{\mathrm{ridge}}(n) + F^S_{slush}(n) + F_{\mathrm{adjust}}(n), \tag{17.258}$$

where F_S is change due to sea ice melting and freezing, F_{drain} is decrease due to desalination, $\mathcal{R}_{\mathrm{remap}}$ is transfer among categories, $\mathcal{A}$ is advection and diffusion, and $\mathcal{R}_{\mathrm{ridge}}$ is category redistribution by ridging. In addition, F^S_{slush} is increase due to slush ice formation (transformation of piled snow to sea ice), and F_{adjust} is increase or decrease due to other adjustments (Sec. 17.8).

The forward difference is used for the time difference, then

$$S_v^{m+1}(n) = S_v^m(n) + \Delta F_S(n) + \Delta F^S_{drain}(n) + \Delta\mathcal{R}_{\mathrm{remap}}(n) + \Delta\mathcal{A}(n) + \Delta\mathcal{R}_{\mathrm{ridge}}(n) + \Delta F^S_{slush}(n) + \Delta F_{\mathrm{adjust}}(n), \tag{17.259}$$

where the terms with Δ are amounts of change during the time step from m to $m+1$ (tendencies). In the following, how to evaluate each term on the right hand side and solve the time evolution will be explained according to the seven calculation steps.

First, $\Delta F_S(n)$ is evaluated by decomposing it into the contributions of each process as follows.

$$\Delta F_S(n) = \Delta F_{\mathrm{melt}}(n) + \Delta F_{\mathrm{frzl\_btm}}(n) + \Delta F_{\mathrm{frzl\_sea}} + \Delta F_{\mathrm{conduct}}(n) + \Delta F_{\mathrm{frzl\_air}}. \tag{17.260}$$

The term $\Delta F_{\mathrm{melt}}(n)$ is decrease due to melting at the sea ice surface (Sec. 17.2.1) and in the sea ice (Sec. 17.2.3),

$$\Delta F_{\mathrm{melt}}(n) = \frac{\rho_0}{\rho_I} W_{RO_{\mathrm{ice}}}(n) S^m(n) a_n \Delta t \qquad \leq 0, \tag{17.261}$$

where $S^m(n)$ indicates average salinity ($S_v^m(n)/v_n^m$) for category n at the time step m. The factor ρ_0/ρ_I is necessary to convert the volume of seawater into the volume of sea ice. The term $\Delta F_{frzl\_btm}(n)$ is increase due to frazil ice formation at the ice bottom (sea ice growth by ocean supercooling, i.e. sea water with a temperature below the freezing point, Sec. 17.2.3), and $\Delta F_{\mathrm{frzl\_sea}}$ is increase due to that in open water and open leads. They are given by Eqs. (17.94) and (17.101). The term $\Delta F_{\mathrm{conduct}}(n)$ indicates increase due to freezing or decrease due to melting at the ice bottom through heat conduction,

$$\Delta F_{\mathrm{conduct}}(n) = \begin{cases} \dfrac{\rho_0}{\rho_I} W_{IO}(n) S_{0_I} a_n \Delta t & \text{if } W_{IO}(n) > 0 \\ \dfrac{\rho_0}{\rho_I} W_{IO}(n) S^m(n) a_n \Delta t & \text{if } W_{IO}(n) < 0, \end{cases} \tag{17.262}$$

where S_{0_I} is the ice bottom skin salinity by the scheme of Mellor and Kantha (1989). The term $\Delta F_{\mathrm{frzl\_air}}$ is increase due to frazil ice formation in open water and open leads,

$$\Delta F_{\mathrm{frzl\_air}} = \Delta v_{\mathrm{frazil\_air}} S_{0_I}, \tag{17.263}$$

where $\Delta v_{\mathrm{frazil\_air}}$ is given by (17.137).

Second, the desalination, $\Delta F^S_{drain}(n)$, is decided (under development).

Thrid, increase or decrease due to the sea ice remapping, $\Delta\mathcal{R}_{\mathrm{remap}}$, is evaluated. Since the sea ice volume transfer among categories is given by (17.163) and (17.165),

$$\Delta\mathcal{R}_{\mathrm{remap}}(n) = \Delta v_{n-1,+} S'(n-1) - \Delta v_{n,+} S'(n) + \Delta v_{n+1,-} S'(n+1) - \Delta v_{n,-} S'(n), \tag{17.264}$$

where average salinity for category n, $S'(n)$, includes the effects of ΔF_S and $\Delta F^S_{drain}(n)$ at this time step, that is,

$$S'(n) = \left(S_v^m(n) + \Delta F_S + \Delta F^S_{drain}(n) \right) / v_n, \tag{17.265}$$

(The volume v_n has already been modified to reflect the thermodynamic process.) In addition, the sum of $\Delta F_{\mathrm{frzl\_air}}$ and $\Delta F_{\mathrm{frzl\_sea}}$ is added to $S_v(n)$ of the corresponding category after Sec. 17.3.2.

Fourth, advection and diffusion are solved. We use the MPDATA scheme like any other sea ice variable, and update S_v itself as written in Sec. 17.13.1 instead of finding the advection-diffusion term, $\mathcal{A}$. In the program, a provisional value is obtained by

$$S_v''(n) = S_v^m(n) + \Delta F_S(n) + \Delta\mathcal{R}_{\mathrm{remap}}(n), \tag{17.266}$$

and then

$$S_v'''(n) = S_v''(n) + \Delta\mathcal{A} \tag{17.267}$$

is derived directly from the scheme.

Fifth, increase or decrease due to sea ice ridging, $\Delta\mathcal{R}_{\mathrm{ridge}}$, is evaluated. As with any other variable, such as (17.215), the transfer from category n to m, $\Delta S_{v,n,m}$, is

$$\Delta S_{v,n,m} = f_m^{\mathrm{vol}} c_n S_v'''(n). \tag{17.268}$$

Using this,

$$\Delta S^n_{v,-} = \begin{cases} \displaystyle\sum_{m=n}^{N_c} \Delta S_{v,n,m} & \text{for } n \le N_c - 1 \\ 0 & \text{for } n = N_c \end{cases} \tag{17.269}$$

$$\Delta S^n_{v,+} = \begin{cases} \displaystyle\sum_{m=1}^{n} \Delta S_{v,m,n} & \text{for } n \le N_c - 1 \\ \displaystyle\sum_{m=1}^{N_c-1} \Delta S_{v,m,n} & \text{for } n = N_c \end{cases} \tag{17.270}$$

$$\Delta\mathcal{R}_{\mathrm{ridge}}(n) = -\Delta S^n_{v,-} + \Delta S^n_{v,+} \qquad (n = 1 \sim N_c). \tag{17.271}$$

Sixth, salinity increase due to slush ice formation, $\Delta F^S_{slush}(n)$, is given by (17.243).

Finally, regarding the values obtained from the above,

$$S_v''''(n) = S_v'''(n) + \Delta\mathcal{R}_{\mathrm{ridge}}(n) + \Delta F_{slush}^{S}(n) \tag{17.272}$$

the adjustment term, $\Delta F_{\mathrm{adjust}}(n)$, is calculated following sea ice incease or decrease in Sec. 17.8. Then, salinity at the next step, $m+1$, is determined by

$$S_v^{m+1}(n) = S_v''''(n) + \Delta F_{\mathrm{adjust}}(n). \tag{17.273}$$

17.10.3 Salinity transport between sea ice and ocean

In v5.0, where sea ice salinity is fixed, the temperature transport associated with water transport between sea ice and ocean, F_I^θ in (14.33), and the salinity transport, F_I^S in (14.34), are obtained directly from components of the water transport, I in (17.251), such as (17.255). In v5.1 or later, the transport calculation has been modified to allow variable sea ice salinity. Sea ice has different salinity and freezing point at each grid and each category, so that salinity and temperature fo the transported water due to ice forming or melting vary at each grid and each category. Therefore, the model has been changed to calculate F_I^S and F_I^θ for each process, as described below.

The salinity transport between ocean and sea ice during one time step, F^S [psu m^3 m^{-2}], is

$$F^S = \Sigma_{n=1}^{N_c} \left\{\Delta F_S(n) + \Delta F_{drain}^{S}(n) + \Delta F_{slush}^{S}(n) + \Delta F_{\mathrm{adjust}}(n)\right\}, \tag{17.274}$$

using the volume of sea ice as a unit. The positive values indicate transfer from ocean to sea ice (upward), while the negative values the reverse direction (downward). Based on this, the salinity flux for the ocean, F_I^S [psu cm^3 cm^{-2} s^{-1}], is given by

$$F_I^S = 10^2(\rho_I/\rho_o)F^S/\Delta t, \tag{17.275}$$

where the factor ρ_I/ρ_o is a unit conversion from sea ice volume to sea water volume, and 10^2 is a unit conversion from [m] to [cm].

The temperature transport, F_I^θ, is basically same as (17.252), though transport due to discharge of the brine water, $F_{\mathrm{drain}}^\theta$, is added

$$F_I^\theta = F_{Iup}^\theta + F_{Idown}^\theta + F_{drain}^\theta. \tag{17.276}$$

17.10.4 Usage

See `docs/README_namelist.md` for use of this scheme.

17.11 Coupling with an atmospheric model

17.11.1 General features

In coupled mode (`SCUPCGCM`), the boundary between the atmospheric component and the ocean-ice component is at the air-ice(snow) boundary. The fluxes above the air-ice boundary are computed by the atmospheric component and passed to the ocean-ice component via the coupler Scup (Yoshimura and Yukimoto, 2008). All the information needed by the oceanic component is received by calling `cgcm_scup__get_a2o` (`cgcm_scup.F90`) from `ogcm__run` (`ogcm.F90`) at the beginning of a coupling cycle. The information needed by the ice part is extracted by calling `get_fluxi_a2o` (`get_fluxs.F90`) from `iaflux` (`ice_flux.F90`). The main part of the ice is solved using surface fluxes and ice surface temperature from the atmospheric component.

In the atmospheric component, temperature in the atmospheric boundary layer and at the ice surface (T_3(`tsfcin`)) are computed along with heat flux in snow layer ($Q_S = Q_{I2}$ (`fheatu`)) using ice temperature (T_1(`t1icen`)), snow thickness (h_s(`hsnwn`)), and ice thickness (h_I(`hicen`)) given by the oceanic component (see Figure 17.2 and Table 17.1). The properties needed by the atmospheric component are sent via `cgcm_scup__put_o2a` (`cgcm_scup.F90`) from `ogcm__run` (`ogcm.F90`) at the end of a coupling cycle.

To conserve heat and water in the coupled atmosphere-ocean system, globally integrated heat and fresh water flux must match between the atmospheric and oceanic components. To achieve this, surface fluxes are adjusted in several steps, which are explained in the following.

17.11.2 Correct errors due to interpolation

First, to resolve errors that arise during the transformation process, a globally constant adjustment factor for heat flux $q_{\text{adj}}^{\text{trans}}$ is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\text{ini}} q_n + q_{\text{adj}}^{\text{trans}}\Big) dS = Q_{\text{ocean}}^{\text{atm}}, \tag{17.277}$$

where a_n^{ini} is the area fraction of sea ice for category n at the time of exchange, q_n is the net heat flux for category n, and $Q_{\text{ocean}}^{\text{atm}}$ is the net heat flux of the atmospheric model integrated over the ocean. The adjustment factor is given by

$$q_{\text{adj}}^{\text{trans}} = \Big[Q_{\text{ocean}}^{\text{atm}} - \int \sum_{n=0}^{ncat} a_n^{\text{ini}} q_n dS\Big] / S_{\text{ocean}}, \tag{17.278}$$

where S_{ocean} is the global ocean area of the ocean model. $q_{\text{adj}}^{\text{trans}}$ is added to the net longwave radiation flux.

For fresh water, $f_{\text{adj}}^{\text{trans}}$ is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\text{ini}} f_n^{\text{evap}} + f^{\text{prcp}} + f^{\text{rof}} + f_{\text{adj}}^{\text{trans}}\Big) dS = F_{\text{ocean}}^{\text{atm}}, \tag{17.279}$$

where a_n^{ini} is the area fraction of sea ice for category n at the time of exchange, f_n^{evap} is the evaporation or sublimation for category n, f^{prcp} is the precipitation, and f^{rof} is the continental run off, and $F_{\text{ocean}}^{\text{atm}}$ is the net fresh water of the atmospheric model integrated over the ocean. The adjustment factor is given by

$$f_{\text{adj}}^{\text{trans}} = \Big[F_{\text{ocean}}^{\text{atm}} - \int \Big(\sum_{n=0}^{ncat} a_n^{\text{ini}} f_n^{\text{evap}} + f^{\text{prcp}} + f^{\text{rof}}\Big) dS\Big] / S_{\text{ocean}}, \tag{17.280}$$

where S_{ocean} is the global ocean area of the ocean model. $f_{\text{adj}}^{\text{trans}}$ is added to either precipitation or evaporation according to its sign.

This operation is done at subroutine `adjust_fluxes_global` of `get_fluxs.F90`.

17.11.3 Taking into account of the evolution of sea ice state

Second, as the ocean-sea ice system evolves during a coupling cycle, the area fraction of sea ice changes from the one at the time of data exchange. Therefore, the global integral would be different from the initial state. To resolve this difference, another adjustment factor $q_{\text{adj}}^{\text{evo}}$ is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\text{evo}} q_n + q_{\text{adj}}^{\text{evo}}\Big) dS = \int \sum_{n=0}^{ncat} a_n^{\text{ini}} q_n dS, \tag{17.281}$$

where a_n^{evo} is the evolved area fraction of sea ice for category n. The adjustment factor is given as follows:

$$q_{\text{adj}}^{\text{evo}} = \Big[\int \sum_{n=0}^{ncat} a_n^{\text{ini}} q_n dS - \int \sum_{n=0}^{ncat} a_n^{\text{evo}} q_n dS\Big] / S_{\text{ocean}}. \tag{17.282}$$

In total, $q_{\text{adj}}^{\text{trans}} + q_{\text{adj}}^{\text{evo}}$ is added to the longwave radiation flux.

$$q_{\text{adj}\,n}^{LW} = q_{\text{atm}n}^{LW} + q_{\text{adj}}^{\text{trans}} + q_{\text{adj}}^{\text{evo}} \tag{17.283}$$

For fresh water, because evaporation or sublimation may be affected by the evolution of sea ice, $f_{\text{adj}}^{\text{evo}}$ is introduced so that

$$\int \Big(\sum_{n=0}^{ncat} a_n^{\text{evo}} f_n^{\text{evap}} + f_{\text{adj}}^{\text{evo}}\Big) dS = \int \sum_{n=0}^{ncat} a_n^{\text{ini}} f_n^{\text{evap}} dS, \tag{17.284}$$

where a_n^{evo} is the evolved area fraction of sea ice for category n. The adjustment factor is given as follows:

$$f_{\text{adj}}^{\text{evo}} = \Big[\int \sum_{n=0}^{ncat} a_n^{\text{ini}} f_n^{\text{evap}} dS - \int \sum_{n=0}^{ncat} a_n^{\text{evo}} f_n^{\text{evap}} dS\Big] / S_{\text{ocean}}. \tag{17.285}$$

$f_{\mathrm{adj}}^{\mathrm{evo}}$ is added to either evaporation or precipitation according to its sign.
This operation is done at subroutine `adjust_fluxes_local` of `get_fluxs.F90`.

17.11.4 Correct errors due to solution method

Third, discrepancies due to solution methods adopted at the air-ice interface (Section 17.2.1) are corrected. To ensure a stable model integration, the ice surface temperature is set as an average of those of the atmospheric component and the ocean-ice component. This new temperature T_3^o is used to replace the upward longwave flux of the adjusted heat flux above the air-ice interface (q_{adj}^{AI}) that contains the net longwave flux (q_{adj}^{LW}). Because the longwave flux q_{adj}^{LW} contains the upward flux due to T_3^a of the atmospheric model, q_{adj}^{LW} is replaced with $q_{\mathrm{adj}}^{LW'}$ in the following manner,

$$q_{\mathrm{adj}}^{LW'} = q_{\mathrm{adj}}^{LW} - \epsilon\sigma(T_3^a)^4 + \epsilon\sigma(T_3^o)^4. \tag{17.286}$$

For the surface where snow or ice is melting, the amount of melting is calculated using $q_{\mathrm{adj}}^{AI'}$ that contains $q_{\mathrm{adj}}^{LW'}$, the discrepancy of the longwave radiation over melting surface should be stored for flux correction.

For the surface where melting does not occur, the interior flux (Q_S or Q_{I2}, they are equal), is obtained by equating $q_{\mathrm{adj}}^{AI'}$ and Q_S. Q_S is different than q_{adj}^{AI} owing to the adjustment for the longwave radiation explained above. Furthermore, $q_{\mathrm{adj}}^{AI'}$ and Q_S are not completely the same owing to the approximations used in the solution method explained in Section 17.2.1. This is not a serious problem for an ocean-only simulation. But this should be taken into account for the coupled model where exact surface heat flux conservation is required. Thus the difference between the adjusted air-ice flux and the interior flux, $q_{\mathrm{adj}}^{AI} - Q_S$, should be stored.

To summarize, discrepancy of fluxes are stored in q_{diff} in the following manner,

$$q_{\mathrm{diff}} = q_{\mathrm{adj}}^{AI} - q_{\mathrm{adj}}^{AI'} = q_{\mathrm{adj}}^{LW}(T_3^a) - q_{\mathrm{adj}}^{LW}(T_3^o), \qquad \text{if surface is melting} \tag{17.287}$$

$$= q_{\mathrm{adj}}^{AI} - Q_S, \qquad \text{if surface is not melting.} \tag{17.288}$$

This is integrated over sea ice and divided by the oceanic area of the ocean model to give an offset factor $q_{\mathrm{adj}}^{\mathrm{stable}}$, which is added to the oceanic surface heat flux globally,

$$q_{\mathrm{adj}}^{\mathrm{stable}} = \int \sum_{n=1}^{ncat} a_n (q_{\mathrm{diff}})_n dS / S_{\mathrm{ocean}}. \tag{17.289}$$

Sampling is done at subroutine `iaflux` of ice_flux.F90 and flux adjustment is done at subroutine `si_exit` of `mod_seaice_cat.F90`.

17.12 Nesting

17.12.1 One-way nesting

In one-way nesting, it is recommended to set the boundary between coarse and fine resolution models (parent and child respectively) along the line connecting tracer points. Setting velocity points as the boundary is awkward for several reasons. For example, the velocity of the child model is zero at the boundary if sea ice of the parent model does not exist around boundary. However, sea ice may exist inside the boundary in the child model in one-way nesting. Lacking the transport removing sea ice from the child model through the boundary, sea ice may accumulate at tracer cells adjacent to the boundary. A serious discontinuity may result. If tracer points are set as the boundary, this kind of accumulation will not occur. However, conservation does not hold.

17.12.2 Two-way nesting

In two-way nesting, it is necessary to set velocity points as the boundary to impose global conservation including sea ice. By reflecting the child model state to the parent model, the problem seen in one-way nesting, an inconsistent distribution of sea ice around the boundary, may be avoided. However, owing to the application of the flux adjustment for conservation, the positive definiteness of area fraction and volume of sea ice and snow, or negative definiteness of sea ice enthalpy are not guaranteed after the advection equation is integrated.

This problem is avoided by the following treatment on the transport fluxes. First, the child model flux is adjusted to match the parent model flux (Chapter 22). Second, if the original flux calculated by the child model is going out of child model's main region, the flux received from the parent model is replaced by this outgoing flux. Third, the difference between the outgoing flux and the parent model flux is sent to the parent model. Fourth, if the value just inside the boundary of the child model is predicted to be negative, the boundary flux is adjusted so that the value does not get negative (a certain amount of quantity is taken from the parent). The adjustment factor is sent to the parent model. Fifth, in the parent model, the flux corrections received from the child model are summed over the corresponding parent grid and added to the original flux of the parent model.

Another problem arises for the interior temperature of sea ice. Owing to the discrepancy between the thermal energy and the volume of the ice pack after advection, the interior temperature of sea ice layer may get erroneous. Therefore, on both sides of the parent-child boundary, interior temperature of sea ice is calculated as the average of the top and bottom surfaces of the sea ice,

$$T_1^{\mathrm{adj}} = \frac{T_2 + T_0}{2}. \tag{17.290}$$

The difference between the predicted and adjusted ice energy is integrated along the parent-child boundary (Γ) as

$$Q_{\mathrm{diff}} = \int_\Gamma \sum_{n=1}^{ncat} \rho_I a_n h_n \left[E(T_1^{\mathrm{adj}}, r_1^{\mathrm{adj}}) - E(T_1, r_1)\right] dS. \tag{17.291}$$

This is reflected as a correction factor for the ocean surface heat flux of each model, which is computed as

$$q_{\mathrm{adj}}^{thermal} = Q_{\mathrm{diff}} / S_{\mathrm{ocean}}^{\mathrm{core}}, \tag{17.292}$$

where $S_{\mathrm{ocean}}^{\mathrm{core}}$ is the area of the main region of the parent or child model.

17.13 Discretization

17.13.1 Advection (MPDATA)

In MPDATA, tracer (α in Figure 17.7) is updated following a three-step procedure.

a. A temporary value is computed using an upstream scheme.

The tracer fluxes at the side boundaries of a T-cell are:

$$\begin{aligned} F_x\left(\alpha_{i,j}^n, \alpha_{i+1,j}^n, u_{i+\frac{1}{2},j}^n\right) &= \left[\alpha_{i,j}^n (u_{i+\frac{1}{2},j}^n + |u_{i+\frac{1}{2},j}^n|) + \alpha_{i+1,j}^n (u_{i+\frac{1}{2},j}^n - |u_{i+\frac{1}{2},j}^n|)\right] \frac{\Delta y}{2}, \\ F_y\left(\alpha_{i,j}^n, \alpha_{i,j+1}^n, v_{i,j+\frac{1}{2}}^n\right) &= \left[\alpha_{i,j}^n (v_{i,j+\frac{1}{2}}^n + |v_{i,j+\frac{1}{2}}^n|) + \alpha_{i,j+1}^n (v_{i,j+\frac{1}{2}}^n - |v_{i,j+\frac{1}{2}}^n|)\right] \frac{\Delta x}{2}, \end{aligned} \tag{17.293}$$

where

$$u_{i+\frac{1}{2},j} = \frac{1}{2}\left(u_{i+\frac{1}{2},j+\frac{1}{2}} + u_{i+\frac{1}{2},j-\frac{1}{2}}\right), \tag{17.294}$$

$$v_{i,j+\frac{1}{2}} = \frac{1}{2}\left(v_{i+\frac{1}{2},j+\frac{1}{2}} + v_{i-\frac{1}{2},j+\frac{1}{2}}\right). \tag{17.295}$$

The zonal flux is defined at the closed circle and the meridional flux is defined at the closed square in Figure 17.7. Using this, the temporary value (α^*) is computed using an upstream scheme:

$$\begin{aligned} \frac{(\alpha_{i,j}^* - \alpha_{i,j}^n)\Delta S_{i,j}}{\Delta t} &= F_x\left(\alpha_{i-1,j}^n, \alpha_{i,j}^n, u_{i-\frac{1}{2},j}^{n+\frac{1}{2}}\right) - F_x\left(\alpha_{i,j}^n, \alpha_{i+1,j}^n, u_{i+\frac{1}{2},j}^{n+\frac{1}{2}}\right) \\ &\quad + F_y\left(\alpha_{i,j-1}^n, \alpha_{i,j}^n, v_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}\right) - F_y\left(\alpha_{i,j}^n, \alpha_{i,j+1}^n, v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}\right). \end{aligned} \tag{17.296}$$

b. Compute the anti-diffusive transport velocity.

Using the temporary value computed in the first step, the anti-diffusive transport velocity is computed as follows:

$$\tilde{u}_{i+\frac{1}{2},j} = \frac{1}{2}\left[\frac{1}{\alpha^{*(2)}_{i+\frac{1}{2},j}}\left(|u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}|\Delta x - \Delta t(u^{n+\frac{1}{2}}_{i+\frac{1}{2},j})^2\right)\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} - \frac{1}{\alpha^{*(6)}_{i+\frac{1}{2},j}}\Delta t u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial y}\right)_{i+\frac{1}{2},j}\right] - \frac{1}{2}\Delta t u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+\frac{1}{2},j}, \tag{17.297}$$

$$\tilde{v}_{i,j+\frac{1}{2}} = \frac{1}{2}\left[\frac{1}{\alpha^{*(2)}_{i,j+\frac{1}{2}}}\left(|v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}|\Delta y - \Delta t(v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}})^2\right)\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} - \frac{1}{\alpha^{*(6)}_{i,j+\frac{1}{2}}}\Delta t v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(u^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial x}\right)_{i,j+\frac{1}{2}}\right] - \frac{1}{2}\Delta t v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}, \tag{17.298}$$

where

$$\alpha^{*(2)}_{i+\frac{1}{2},j} = \frac{1}{2}(\alpha^*_{i,j} + \alpha^*_{i+1,j}), \tag{17.299}$$

$$\alpha^{*(2)}_{i,j+\frac{1}{2}} = \frac{1}{2}(\alpha^*_{i,j} + \alpha^*_{i,j+1}), \tag{17.300}$$

$$\alpha^{*(6)}_{i+\frac{1}{2},j} = \frac{1}{8}(\alpha^*_{i,j-1} + \alpha^*_{i+1,j-1} + 2\alpha^*_{i,j} + 2\alpha^*_{i+1,j} + \alpha^*_{i,j+1} + \alpha^*_{i+1,j+1}), \tag{17.301}$$

$$\alpha^{*(6)}_{i,j+\frac{1}{2}} = \frac{1}{8}(\alpha^*_{i-1,j} + \alpha^*_{i-1,j+1} + 2\alpha^*_{i,j} + 2\alpha^*_{i,j+1} + \alpha^*_{i+1,j} + \alpha^*_{i+1,j+1}), \tag{17.302}$$

$$\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} = \frac{\alpha^*_{i+1,j} - \alpha^*_{i,j}}{\Delta x}, \tag{17.303}$$

$$\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} = \frac{\alpha^*_{i,j+1} - \alpha^*_{i,j}}{\Delta y}, \tag{17.304}$$

$$\left(v^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial y}\right)_{i+\frac{1}{2},j} = \frac{1}{4}\left[v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\left(\frac{\partial \alpha^*}{\partial y}\right)_{i,j-\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\left(\frac{\partial \alpha^*}{\partial y}\right)_{i+1,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\left(\frac{\partial \alpha^*}{\partial y}\right)_{i+1,j-\frac{1}{2}}\right], \tag{17.305}$$

$$\left(u^{n+\frac{1}{2}}\frac{\partial \alpha^*}{\partial x}\right)_{i,j+\frac{1}{2}} = \frac{1}{4}\left[u^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\left(\frac{\partial \alpha^*}{\partial x}\right)_{i-\frac{1}{2},j} + u^{n+\frac{1}{2}}_{i+\frac{1}{2},j+1}\left(\frac{\partial \alpha^*}{\partial x}\right)_{i+\frac{1}{2},j+1} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j+1}\left(\frac{\partial \alpha^*}{\partial x}\right)_{i-\frac{1}{2},j+1}\right], \tag{17.306}$$

$$\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \frac{1}{2}\left[\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j} + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i+1,j}\right], \tag{17.307}$$

$$\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = \frac{1}{2}\left[\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j} + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n+\frac{1}{2}}_{i,j+1}\right], \tag{17.308}$$

and

$$\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^{n}_{i,j} = \frac{(u^n_{i+\frac{1}{2},j}\Delta y_{i+\frac{1}{2},j} - u^n_{i-\frac{1}{2},j}\Delta y_{i-\frac{1}{2},j} + v^n_{i,j+\frac{1}{2}}\Delta x_{i,j+\frac{1}{2}} - v^n_{i,j-\frac{1}{2}}\Delta x_{i,j-\frac{1}{2}})}{\Delta S_{i,j}}. \tag{17.309}$$

c. Update tracer using the anti-diffusive velocity starting from the temporary value.

$$
\begin{aligned}
\frac{(\alpha_{i,j}^{n+1}-\alpha_{i,j}^{*})\Delta S_{i,j}}{\Delta t} =& F_x\left(\alpha_{i-1,j}^{*},\alpha_{i,j}^{*},\tilde{u}_{i-\frac{1}{2},j}\right)-F_x\left(\alpha_{i,j}^{*},\alpha_{i+1,j}^{*},\tilde{u}_{i+\frac{1}{2},j}\right)\\
&+F_y\left(\alpha_{i,j-1}^{*},\alpha_{i,j}^{*},\tilde{v}_{i,j-\frac{1}{2}}\right)-F_y\left(\alpha_{i,j}^{*},\alpha_{i,j+1}^{*},\tilde{v}_{i,j+\frac{1}{2}}\right).
\end{aligned}
\tag{17.310}
$$

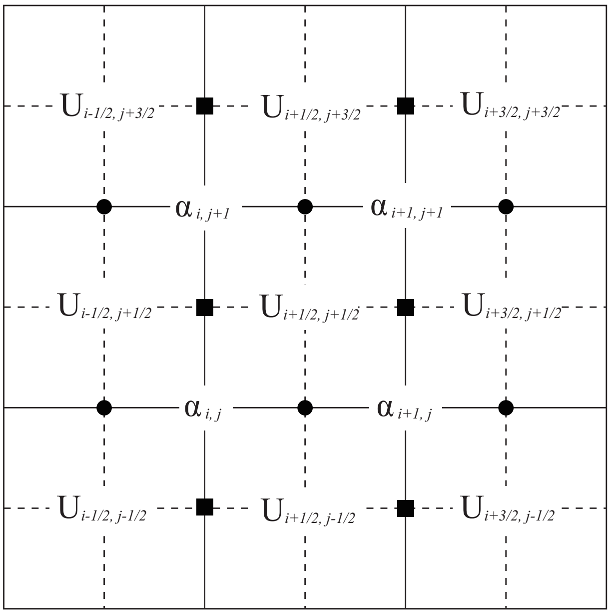

Figure17.7 Position for tracer (α) and velocity (U). Area and thickness are defined at tracer points. Zonal fluxes are computed at closed circles and meridional fluxes are computed at closed squares. The budget is computed for a unit cell for α.

17.13.2 Diffusion (Laplace operator)

The discrete equation of the diffusion equation (17.199) is

$$
\alpha_{i,j}^{\text{new}} =\alpha_{i,j}+\frac{\Delta t}{\Delta S_{i,j}}\left(F_{i-\frac{1}{2},j}^{x}-F_{i+\frac{1}{2},j}^{x}+F_{i,j-\frac{1}{2}}^{y}-F_{i,j+\frac{1}{2}}^{y}\right) \tag{17.311}
$$

$$
F_{i+\frac{1}{2},j}^{x} =-\kappa_H\Delta y_{i+\frac{1}{2},j}\frac{1}{\Delta x_{i+\frac{1}{2},j}}(\alpha_{i+1,j}-\alpha_{i,j}) \tag{17.312}
$$

$$
F_{i,j+\frac{1}{2}}^{y} =-\kappa_H\Delta x_{i,j+\frac{1}{2}}\frac{1}{\Delta y_{i,j+\frac{1}{2}}}(\alpha_{i,j+1}-\alpha_{i,j}), \tag{17.313}
$$

as in Section 11.2.1.

The grid widths (shown in Fig. 3.5) are given by

$$
\Delta x_{i+\frac{1}{2},j} = (\mathtt{dx\_br})_{i,j}+(\mathtt{dx\_bl})_{i,j} \tag{17.314}
$$

$$
\Delta y_{i+\frac{1}{2},j} = (\mathtt{dy\_br})_{i,j}+(\mathtt{dy\_tr})_{i,j-1} \tag{17.315}
$$

$$
\Delta x_{i,j+\frac{1}{2}} = (\mathtt{dx\_tr})_{i-1,j}+(\mathtt{dx\_tl})_{i,j} \tag{17.316}
$$

$$
\Delta y_{i,j+\frac{1}{2}} = (\mathtt{dy\_bl})_{i,j}+(\mathtt{dy\_tl})_{i,j}. \tag{17.317}
$$

17.13.3 Momentum equation

Specific forms of discretization for properties related to internal stress are given here.

The strain rate tensor ($\dot{\epsilon}$) and stress tensor (σ) are defined at tracer points (Figure 17.8).

Components (divergence, tension, and shear) of the strain rate tensor are expressed in a discretized form as follows:

$$
\begin{aligned}
(D_D)_{i,j} = \frac{1}{\Delta x_{i,j}\Delta y_{i,j}}\Big(&\frac{\Delta y_{i+\frac{1}{2},j}}{2}(u_{I\,i+\frac{1}{2},j+\frac{1}{2}} + u_{I\,i+\frac{1}{2},j-\frac{1}{2}}) - \frac{\Delta y_{i-\frac{1}{2},j}}{2}(u_{I\,i-\frac{1}{2},j+\frac{1}{2}} + u_{I\,i-\frac{1}{2},j-\frac{1}{2}}) \\
&+ \frac{\Delta x_{i,j+\frac{1}{2}}}{2}(v_{I\,i+\frac{1}{2},j+\frac{1}{2}} + v_{I\,i-\frac{1}{2},j+\frac{1}{2}}) - \frac{\Delta x_{i,j-\frac{1}{2}}}{2}(v_{I\,i+\frac{1}{2},j-\frac{1}{2}} + v_{I\,i-\frac{1}{2},j-\frac{1}{2}})\Big),
\end{aligned}
$$

$$
\begin{aligned}
(D_T)_{i,j} = &\frac{1}{2}\Big[\frac{\Delta y_{i,j+\frac{1}{2}}}{\Delta x_{i,j+\frac{1}{2}}}\Big(\frac{u_{I\,i+\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I\,i-\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}}\Big) + \frac{\Delta y_{i,j-\frac{1}{2}}}{\Delta x_{i,j-\frac{1}{2}}}\Big(\frac{u_{I\,i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{i+\frac{1}{2},j-\frac{1}{2}}} - \frac{u_{I\,i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big] \\
&- \frac{1}{2}\Big[\frac{\Delta x_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2},j}}\Big(\frac{v_{I\,i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I\,i+\frac{1}{2},j-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j-\frac{1}{2}}}\Big) + \frac{\Delta x_{i,j-\frac{1}{2}}}{\Delta y_{i,j-\frac{1}{2}}}\Big(\frac{v_{I\,i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i-\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I\,i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big],
\end{aligned}
$$

$$
\begin{aligned}
(D_S)_{i,j} = &\frac{1}{2}\Big[\frac{\Delta y_{i,j+\frac{1}{2}}}{\Delta x_{i,j+\frac{1}{2}}}\Big(\frac{v_{I\,i+\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{v_{I\,i-\frac{1}{2},j+\frac{1}{2}}}{\Delta y_{i-\frac{1}{2},j+\frac{1}{2}}}\Big) + \frac{\Delta y_{i,j-\frac{1}{2}}}{\Delta x_{i,j-\frac{1}{2}}}\Big(\frac{v_{I\,i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{i+\frac{1}{2},j-\frac{1}{2}}} - \frac{v_{I\,i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big] \\
&+ \frac{1}{2}\Big[\frac{\Delta x_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2},j}}\Big(\frac{u_{I\,i+\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I\,i+\frac{1}{2},j-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j-\frac{1}{2}}}\Big) + \frac{\Delta x_{i,j-\frac{1}{2}}}{\Delta y_{i,j-\frac{1}{2}}}\Big(\frac{u_{I\,i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x_{i-\frac{1}{2},j+\frac{1}{2}}} - \frac{u_{I\,i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2},j-\frac{1}{2}}}\Big)\Big].
\end{aligned}
$$

The internal stress is defined at velocity points and computed from stress tensor as follows:

$$
\begin{aligned}
(F_\mu)_{i+\frac{1}{2},j+\frac{1}{2}} = &\frac{1}{2}\Big[\frac{1}{2}\Big(\frac{(\sigma_1)_{i+1,j+1} + (\sigma_1)_{i+1,j} - (\sigma_1)_{i,j+1} - (\sigma_1)_{i,j}}{\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big) \\
&+ \frac{1}{2}\Big(\frac{\Delta y^2_{i+1,j+\frac{1}{2}}[(\sigma_2)_{i+1,j+1} + (\sigma_2)_{i+1,j}] - \Delta y^2_{i,j+\frac{1}{2}}[(\sigma_2)_{i,j+1} + (\sigma_2)_{i,j}]}{\Delta y^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big) \\
&+ \Big(\frac{\Delta x^2_{i+\frac{1}{2},j+1}[(\sigma_{12})_{i+1,j+1} + (\sigma_{12})_{i,j+1}] - \Delta x^2_{i+\frac{1}{2},j}[(\sigma_{12})_{i+1,j} + (\sigma_{12})_{i,j}]}{\Delta x^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big),
\end{aligned}
\tag{17.318}
$$

$$
\begin{aligned}
(F_\psi)_{i+\frac{1}{2},j+\frac{1}{2}} = &\frac{1}{2}\Big[\frac{1}{2}\Big(\frac{(\sigma_1)_{i+1,j+1} + (\sigma_1)_{i,j+1} - (\sigma_1)_{i+1,j} - (\sigma_1)_{i,j}}{\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big) \\
&- \frac{1}{2}\Big(\frac{\Delta x^2_{i+\frac{1}{2},j+1}[(\sigma_2)_{i+1,j+1} + (\sigma_2)_{i,j+1}] - \Delta x^2_{i+\frac{1}{2},j}[(\sigma_2)_{i+1,j} + (\sigma_2)_{i,j}]}{\Delta x^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta y_{i+\frac{1}{2},j+\frac{1}{2}}}\Big) \\
&+ \Big(\frac{\Delta y^2_{i+1,j+\frac{1}{2}}[(\sigma_{12})_{i+1,j+1} + (\sigma_{12})_{i+1,j}] - \Delta y^2_{i,j+\frac{1}{2}}[(\sigma_{12})_{i,j+1} + (\sigma_{12})_{i,j}]}{\Delta y^2_{i+\frac{1}{2},j+\frac{1}{2}}\Delta x_{i+\frac{1}{2},j+\frac{1}{2}}}\Big).
\end{aligned}
\tag{17.319}
$$

17.14 Usage

17.14.1 Compilation

The information needed to compile the sea ice model with the OGCM should be given to `configure.in`. At least, `ICE` option should be specified in the `OPTIONS` line and the number of categories of the sea ice thickness should be specified as `NUM_ICECAT`, for example, a line `NUM_ICECAT = 5` should be inserted somewhere in `configure.in`. The model options related to the sea ice model are listed on Table 17.5.

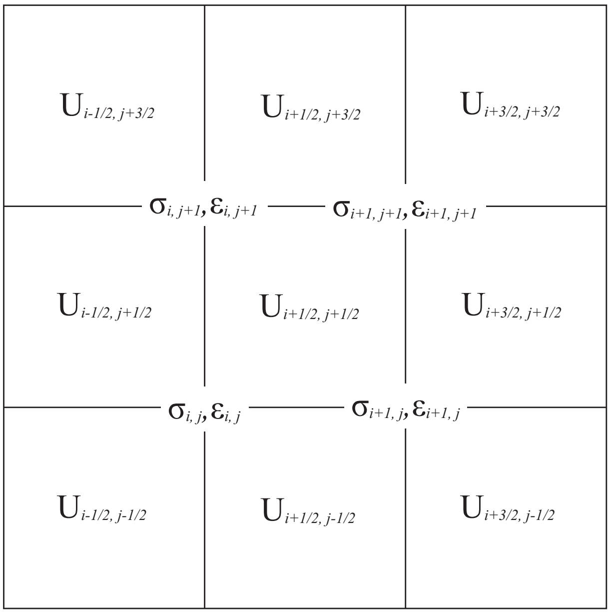

Figure17.8 Position of variables used by dynamics scheme.

Table17.5 Model options related to the sea ice model

option name	description	usage
`ICE`	Sea ice model is included to the OGCM	
`SIDYN`	Dynamics of sea ice is solved	otherwise, sea ice drifts with a third of the surface ocean velocity.
`CALALBSI`	An albedo scheme of CICE is used	otherwise constant
`HISTICECAT`	Monitor time evolution of basic state variables	Use `NAMELIST`.*name_model*.`MONITOR` to specify the output group and interval. *name_model* is the name of the model specified by `configure.in`
`ICEFULLMONIT`	Activate extensive monitor	Use `NAMELIST`.*name_model*.`MONITOR` to specify the output group and interval

17.14.2 Job parameters (namelist)

The runtime job parameters (namelist) that should be given by `NAMELIST`.*name_model* are listed on Tables 17.6 to 17.15. Job parameters related to monitoring are specified in the next subsection.

a. Thickness category

The bounds of thickness categories, whose number should be specified by `NUM_ICECAT` in `configure.in` before compilation, should be given at run time to namelist `nml_seaice_thickness_category`, which is explained in Table 17.6.

Table17.6 Namelist of thickness category of the sea ice model (`nml_seaice_thickness_category`)

variable name	dimension	description	usage (default value)
`hbound(0 : ncat)`	m	category boundary	required
			Continued on next page

Table 17.6 – continued from previous page

variable name	dimension	description	usage (default value)
lsicat_volchk	logical	flag for checking mass conservation	default = .false.

b. Time

The parameters related to time should be given by nml_seaice_time_conf and nml_seaice_time_calendar, which are explained in Table 17.7. Note that the specification of nml_seaice_time_conf is not mandatory, necessary information will be taken from the ocean model.

Table17.7 Namelist related to time of the sea ice model

variable name	group	description	usage (default value)
nstep_seaice	nml_seaice_time_conf	time steps to be proceed	same as OGCM
ibyri	nml_seaice_time_conf	start year of this run	same as OGCM
ibmni	nml_seaice_time_conf	start month of this run	same as OGCM
ibdyi	nml_seaice_time_conf	start day of this run	same as OGCM
ibhri	nml_seaice_time_conf	start hour of this run	same as OGCM
ibmii	nml_seaice_time_conf	start minute of this run	same as OGCM
ibsci	nml_seaice_time_conf	start second of this run	same as OGCM
l_force_leap	nml_seaice_calendar	forcibly specify leap year at run time with l_leap	.false.
l_leap	nml_seaice_calendar	specification of whether this year is leap or not (valid when l_force_leap = .true.)	.false.

c. Dynamics

The parameters related to dynamics of sea ice (SIDYN option) should be given to namelist nml_seaice_dyn, which is listed on Table 17.8

Table17.8 Namelist related to dynamics of the sea icel (nml_seaice_dyn)

variable name	dimension	description	usage (default value)
dt_seaice_dyn_sec	sec	time step interval for dynamics (in seconds)	about a tenth of the baroclinic time step of the ocean model
damp_vice	$\mathrm{m\,sec^{-1}}$	Rayleigh damping coefficient for rapid velocity	0.0
damp_vmax	$\mathrm{m\,sec^{-1}}$	minimum velocity for damping	1.0
prs_scale_factor	$\mathrm{N\,m^{-2}}$	pressure scaling factor	2.75×10^4
e_fold_const_ice_prs	1	e-folding constant for ice pressure	20

d. Diffusion

Coefficient for the harmonic-type horizontal diffusion should be given to namelist nml_seaice_diff, which is explained in Table 17.9.

Table17.9 Namelist of diffusion of the sea ice model (nml_seaice_diff)

variable name	dimension	description	usage (default value)
diff_seaice_m2ps	$\mathrm{m^2\,sec^{-1}}$	horizontal diffusion coefficient	required

e. Minimum fractional area

Minimum of fractional area applied to all thickness categories may be given at run time to namelist nml_seaice_model, which is explained in Table 17.10.

Table17.10 Namelist of minimum fractional area of the sea ice model (`nml_seaice_model`)

variable name	dimension	description	usage (default value)
`concentration_min`	1	minimum fractional area	1×10^{-4}

f. Air-ice transfer

The bulk transfer coefficient at the air - sea ice interface should be given to `nml_air_ice`, which is listed on Table 17.11.

Table17.11 Namelist of air-ice bulk transfer coefficient (`nml_air_ice`)

variable name	units	description	usage (default value)
`c_drag_ai`	1	air-ice exchange coefficient for wind	3.0×10^{-3}
`c_transfer_sensible_heat_ai`	1	air-ice exchange coefficient for sensible heat	1.5×10^{-3}
`c_transfer_latent_heat_ai`	1	air-ice exchange coefficient for evaporation	1.5×10^{-3}

g. Ice-ocean transfer

The bulk transfer coefficient at the ice - ocean interface should be given to `nml_ice_ocean`, which is listed on Table 17.12. If `BULKECMWF` is used, user must specify `nml_bulkecmwf_seaice_run` to control restart input and output for vertical velocity due to convection, $w_{\dagger}$, which is used to calculate the wind velocity in the lowest layer of the atmosphere model.

Table17.12 Namelist of ice-ocean bulk transfer coefficient (`nml_ice_ocean`)

variable name	dimension	description	usage (default value)
`c_drag_io`	1	ice-ocean exchange coefficient for wind	5.5×10^{-3}
`turning_angle_drag_io`	degree	turning angle for ice-ocean drag	-999 (set to $\pm cz$, where c: $1°\mathrm{m}^{-1}$, z: the first layer depth [m]; positive/negative in the northern/southern hemisphere)

Table17.13 Namelist `nml_bulkecmwf_seaice_run`)

variable name	units	description	usage
`l_rst_bulkecmwf_wstar_in`	logical	read initial restart for the free convection velocity scale over sea ice or not	default = `l_rst_seaice_in`

h. Restart

How to handle restart files is specified by namelists `nml_seaice_run` (Table 17.14), and `nml_restart` with `name` = "`Sea Ice`".

Table17.14 Namelist related to restart of the sea ice model

variable name	group	usage (default value)
`l_rst_seaice_in`	`nml_seaice_run`	`.true.` : the initial state is read from restart file `.false.`(default) : start from the state without sea ice

i. Albedo

When `CALALBSI` option is selected, an albedo scheme of CICE is used. The parameters of this scheme should be given by namelist `nml_albedo_seaice`, which is listed on Table 17.15.

Table17.15 Namelist for the albedo of sea ice used by `CALALBSI` option (`nml_albedo_seaice`)

variable name	dimension	description	usage (default value)
`alb_ice_visible_t0`		visible ice albedo for thicker ice	0.78
`alb_ice_nearIR_t0`		near infrared ice albedo for thicker ice	0.36
`alb_snw_visible_t0`		visible, cold snow albedo	0.98
`alb_snw_nearIR_t0`		near infrared, cold snow albedo	0.70
`alb_ice_visible_dec_ratio`	$(°C)^{-1}$	visible ice albedo declination rate	0.075
`alb_ice_nearIR_dec_ratio`	$(°C)^{-1}$	near infrared ice albedo declination rate	0.075
`alb_snw_visible_dec_ratio`	$(°C)^{-1}$	visible snow albedo declination rate	0.10
`alb_snw_nearIR_dec_ratio`	$(°C)^{-1}$	near infrared ice albedo declination rate	0.15
`hi_ref`	m	the maximum ice thickness to which connection function is used	0.50
`atan_ref`		the base value of the tangent hyperbolic connection function	4.0
`tsfci_t0`	°C	the temperature at which ice albedo is equated to that of ocean	0.0
`tsfci_t1`	°C	the temperature at which ice albedo is started to decline to that of the ocean	−1.0
`fsnow_patch`	meter	thickness of snow patch on melting bare ice	0.02

j. Emissivity

The emissivity of sea ice surface and snow surface should be given to `nml_seaice_emissivity`, which is listed on Table 17.16.

Table17.16 Namelist of emissivity of sea ice surface and snow surface (`nml_seaice_emissivity`)

variable name	dimension	description	usage (default value)
`emissivity_sea_ice`	1	Emissivity of sea ice surface	1
`emissivity_snow`	1	Emissivity of snow surface	1

17.14.3 Monitoring

To monitor or sample the state of sea ice, `HISTICECAT` option must be specified. Extensive monitoring is activated by further specifying `ICEFULLMONIT` option.

The general behavior should be specified by the `nml_seaice_budget` and `nml_seaice_hst` in `NAMELIST`.*name_model*, which is explained in Table 17.17.

Table17.17 Namelist of monitor of the sea ice model

variable name	group	description	usage
`nstep_si_budget_interval`	`nml_seaice_budget`	the interval of time step by which water budget is written	0 : none, −1 : monthly
`undef_value`	`nml_seaice_hst`	undefined value to fill ice free grid point	real(8)
`undef_for_land`	`nml_seaice_hst`	undefined value to fill land grid points	real(8)

Sampling variables are grouped according to the processes they are involved, the grid points they are defined, and the priorities given by Sea Ice Model Intercomparison Project (SIMIP) (Notz et al., 2016).

The groups are listed on Table 17.18. Users specify `nml_history` blocks in file `NAMELIST`.*name_model*.`MONITOR` in the following manner.

Specification of NAMELIST.*name_model*.MONITORforseaicemonitoring

```
&nml_history
  name          = 'sea ice basic state vars at t-point',
  file_base     = 'result/hs_ice_t_sta1',
  interval_step = 12,
  suffix        = 'minute',
&end
```

Table17.18 List of groups for sea ice monitoring

group name	usage note
sea ice basic state vars at t-point	"mean" is forced
sea ice state vars at t-point	"mean" is forced
tendencies of sea-ice mass and area fraction at t-point	"mean" is forced
heat and freshwater fluxes over sea ice fraction at t-point	"mean" is forced
sea ice state vars of each category at t-point	"mean" is forced
sea ice basic dynamics at u-point	"mean" is forced
sea ice dynamics at u-point	"mean" is forced
sea ice dynamics at t-point	"mean" is forced
sea ice vars at x-point	
sea ice vars at y-point	
sea ice integrated measures	
sea ice dynamics at u-point priority 3	valid if `ICEFULLMONIT`, "mean" is forced
sea ice dynamics at t-point priority 3	valid if `ICEFULLMONIT`, "mean" is forced
sea ice fluxes of each category at t-point	valid if `ICEFULLMONIT`
heat and freshwater fluxes over open leads at t-point	valid if `ICEFULLMONIT`

17.15 Appendix

17.15.1 Partial derivative of specific humidity with respect to temperature

In MRI.COM, surface temperature of ice (T_3) is computed using semi-implicit method, where an expression for the partial derivative of the saturation specific humidity with respect to temperature (T) is needed.

Properties of moist air used by MRI.COM is based on Gill (1982) and replicated in Section 14.12.2. Using (14.122), the saturation specific humidity (q_i) is given as

$$q_i = \epsilon e'_{si}/(p_s - (1-\epsilon)e'_{si}). \tag{17.320}$$

where saturation partial pressure of moist air over ice (e'_{si}) is given by (14.128) and (14.129). Using (17.320),

$$\frac{\partial q_i}{\partial T} = \frac{\epsilon p_s}{\{p_s - (1-\epsilon)e'_{si}\}^2}\frac{\partial e'_{si}}{\partial T}, \tag{17.321}$$

where $\partial e'_{si}/\partial T$ is expressed by setting $f_w = 1$ as

$$\frac{\partial e'_{si}}{\partial T} = \ln 10 \cdot 10^{g(T)} \cdot g'(T), \tag{17.322}$$

where $g(T) = (0.7859 + 0.03477T)/(1 + 0.00412T) + 0.00422T$ and $g'(T) = \partial g(T)/\partial T$, where T is temperature at the upper surface of sea ice in °C.

Chapter 18

Tide model

This chapter explains introduction of a tidal model. MRI.COM uses the scheme of Sakamoto et al. (2013), which runs a barotropic tide model with the same grid and topography as the barotropic model (Chapter 7) in parallel, and incorporates tidal effects into OGCM at each baroclinic time step. We describe derivation of the governing equations of the tide model (Sec. 18.1), incorporation of tidal effects into OGCM (Sec. 18.2), calculation of the tidal forcing potential (Sec. 18.3), execution of nesting experiments (Sec. 18.4), and usage (Sec. 18.5).

18.1 Governing equations

Adding the tide-related terms directly to the equation of motion of the barotropic model, Eqs. (7.1) and (7.2), gives: (e.g., Schiller, 2004)

$$\frac{\partial \mathbf{U}}{\partial t} + f\mathbf{k} \times \mathbf{U} = -g(\eta + H)\nabla(\eta - \beta\eta_{\mathrm{eq}} - \eta_{\mathrm{SAL}}) + \mathbf{F}^{\mathrm{horz}} + \frac{\tau^{\mathrm{btm}}}{\rho_0} + \mathbf{X}', \tag{18.1}$$

where η_{eq} is equilibrium tide, or astronomical tide-generating potential (18.3), β defines effect of tide-generating potential and correction due to earth tides, and η_{SAL} represents the self-attraction and loading of the ocean tides. (In MRI.COM, we usually set $\beta = 0.7$, though in general $\beta = 1 + k - h$ with k and h being Love numbers.) The term $\mathbf{F}^{\mathrm{horz}}$ is the vertically integrated horizontal viscosity parameterization terms, and τ^{btm} is the bottom friction terms that may be specialized for tidal currents. They are separated from $\mathbf{X}$ defined by (7.3) and (7.4) and $\mathbf{X}' \equiv \mathbf{X} - \mathbf{F}^{\mathrm{horz}} - \tau^{\mathrm{btm}}/\rho_0$. Note also that atmospheric pressure term is dropped for brevity. The continuity equation is same as Eq. (7.5):

$$\frac{\partial \eta}{\partial t} + \nabla \cdot \mathbf{U} = F_w, \tag{18.2}$$

where $F_w \equiv P - E + R + I$ is surface fresh water forcing that will be enter the system through the basic fields.

As Eq. (18.1) indicates, the terms specialized for tides affect the basic fields unintentionally. For example, the self-attraction and loading term will modify the relationships between the sea surface height gradient and the barotropic currents. Horizontal viscosity and bottom friction will also change the results, if they are modified to be suitable for tidal modeling. As a result, the model solution without tidal potential (i.e. $\eta_{\mathrm{eq}} = 0$) is no longer the same as that of the OGCM barotropic model. To resolve this problem, our tide scheme calculates the tidal fields separately from the basic fields within the barotropic submodel as explained by Sakamoto et al. (2013). The essence of the solution method is explained below.

Sakamoto et al. (2013) derived the governing equations of the barotropic tidal model, focusing on smooth coupling with an OGCM. First, the variables are decomposed into the basic and linear tidal components,

$$\mathbf{U} = \mathbf{U}_b + \mathbf{U}_{lt} \tag{18.3}$$

$$\eta = \eta_b + \eta_{lt} \tag{18.4}$$

$$\mathbf{F}^{\mathrm{horz}} = \mathbf{F}^{\mathrm{horz}}_b + \mathbf{F}^{\mathrm{horz}}_{lt} \tag{18.5}$$

$$\tau^{\mathrm{btm}} = \tau^{\mathrm{btm}}_b + \tau^{\mathrm{btm}}_{lt}, \tag{18.6}$$

where "b" and "lt" refer to the basic and linear tidal components, respectively. Now we decompose (18.1) and (18.2) to obtain governing equations for each of the two components.

For linear tidal component:

$$\frac{\partial \mathbf{U}_{lt}}{\partial t} + f\mathbf{k} \times \mathbf{U}_{lt} = -g(\eta + H)\nabla(\eta_{lt} - \beta\eta_{\mathrm{eq}} - \eta_{\mathrm{SAL}}) + \mathbf{F}^{\mathrm{horz}}_{lt} + \frac{\tau^{\mathrm{btm}}_{lt}}{\rho_0}, \tag{18.7}$$

$$\frac{\partial \eta_{lt}}{\partial t} + \nabla \cdot \mathbf{U}_{lt} = 0, \tag{18.8}$$

where we assume that self-attraction and loading acts only to linear tidal component in such a way as $\eta_{\mathrm{SAL}} = (1-\alpha)\eta_{lt}$.

For the basic component:

$$\frac{\partial \mathbf{U}_b}{\partial t} + f\mathbf{k}\times\mathbf{U}_b = -g(\eta+H)\nabla\eta_b + \mathbf{F}_b^{\mathrm{horz}} + \frac{\tau_b^{\mathrm{btm}}}{\rho_0} + \mathbf{X}', \tag{18.9}$$

$$\frac{\partial \eta_b}{\partial t} + \nabla\cdot\mathbf{U}_b = F_w. \tag{18.10}$$

This equation is basically same as the barotropic model written in Chapter 7.

In the above decomposition, the linear tidal component represents only the linear response to tidal forcing, η_{eq}, and the secondary effects of tide, such as tidal advection and internal tides, are represented by the basic component. The linear terms in the barotropic equations, such as the Coriolis force and the Laplacian horizontal viscosity with a constant viscosity coefficient, may be split into the basic and linear tidal components naturally.

The bottom friction term (τ^{btm}) is non-linear and should be treated carefully. The bottom friction term (see Section 8.3.6 for details) is expressed as

$$\tau^{\mathrm{btm}} = -\rho_0 C_D|\mathbf{u}|\mathbf{T}_\theta\mathbf{u}, \tag{18.11}$$

where C_D is a drag coefficient and $\mathbf{T}_\theta$ is a matrix representing horizontal veering with an angle of θ,

$$\mathbf{T}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{18.12}$$

Sum of the two components are used for the scalar part ($|\mathbf{u}|$) and only the vector part is decomposed:

$$\tau_b^{\mathrm{btm}} = -\rho_0 C_D|\mathbf{u}_b+\mathbf{u}_{lt}|\mathbf{T}_\theta\mathbf{u}_b, \tag{18.13}$$

$$\tau_{lt}^{\mathrm{btm}} = -\rho_0 C_D|\mathbf{u}_b+\mathbf{u}_{lt}|\mathbf{T}_\theta\mathbf{u}_{lt}. \tag{18.14}$$

Similarly, when the horizontal viscosity parameterization is non-linear as in the Smagorinsky scheme, the coefficient part (e.g., ν_H in the equations below) is calculated by using the full velocity, while the decomposed fields are applied to the viscosity operator. In MRI.COM, the horizontal tension D_T and shear D_S of the velocity field,

$$D_T = h_\psi\frac{\partial}{h_\mu\partial\mu}\left(\frac{u}{h_\psi}\right) - h_\mu\frac{\partial}{h_\psi\partial\psi}\left(\frac{v}{h_\mu}\right), \tag{18.15}$$

$$D_S = h_\mu\frac{\partial}{h_\psi\partial\psi}\left(\frac{u}{h_\mu}\right) + h_\psi\frac{\partial}{h_\mu\partial\mu}\left(\frac{v}{h_\psi}\right), \tag{18.16}$$

are used to calculate horizontal viscosity as

$$F_u^{\mathrm{horz}} = \frac{1}{h_\psi^2}\frac{\partial}{h_\mu\partial\mu}\left(h_\psi^2\nu_H D_T\right) + \frac{1}{h_\mu^2}\frac{\partial}{h_\psi\partial\psi}\left(h_\mu^2\nu_H D_S\right), \tag{18.17}$$

$$F_v^{\mathrm{horz}} = \frac{1}{h_\psi^2}\frac{\partial}{h_\mu\partial\mu}\left(h_\psi^2\nu_H D_S\right) - \frac{1}{h_\mu^2}\frac{\partial}{h_\psi\partial\psi}\left(h_\mu^2\nu_H D_T\right). \tag{18.18}$$

Both of horizontal viscosity of the basic component, $\mathbf{F}_b^{\mathrm{horz}}$, and that of the linear tidal component, $\mathbf{F}_{lt}^{\mathrm{horz}}$, are calculated by Eqs. (18.17) and (18.18) using common ν_H. However, D_T and D_S are calculated for $\mathbf{u}_b$ and $\mathbf{u}_{lt}$, respectively.

By the above decomposition, the effect of the tidal model can be taken in while using the original barotropic model. In fact, if $\eta_{\mathrm{eq}} \equiv 0$, then $\eta_{\mathrm{lt}} \equiv 0$, $\mathbf{U}_{lt} \equiv \mathbf{0}$, and the above equations return to the original ones.

By formulation, $\mathbf{F}^{\mathrm{horz}}$, τ^{btm}, and $\mathbf{X}'$ are computed at the baroclinic time level of n and kept fixed during the integration of barotropic submodel. However, when the barotropic field varies significantly during the integration, which is expected when tidal forcing is included, $\mathbf{F}^{\mathrm{horz}}$ and τ^{btm} should be allowed to vary with the evolution of barotropic velocity field. In this case, $\mathbf{F}_{lt}^{\mathrm{horz}}$ and τ_{lt}^{btm} in (18.7) may be replaced by the viscosity parameterization suitable to the tidal velocity fields (represented by $\mathbf{F}_{tide}^{\mathrm{horz}}$ and $\tau_{tide}^{\mathrm{btm}}$, respectively). Then, (18.7) becomes,

$$\frac{\partial \mathbf{U}_{lt}}{\partial t} + f\mathbf{k}\times\mathbf{U}_{lt} = -g(\eta+H)\nabla(\alpha\eta_{lt}-\beta\eta_{\mathrm{eq}}) + \mathbf{F}_{tide}^{\mathrm{horz}} + \frac{\tau_{tide}^{\mathrm{btm}}}{\rho_0}. \tag{18.19}$$

There is no strong restriction on the form of the viscosity parameterization applied to the tidal velocity fields. However, by using a common form for both baroclinic and barotropic fields, depth integrated viscosity terms would cancel when barotropic and baroclinic modes are synchronous (see (Eq. 18.23)). In MRI.COM, the bottom friction for the linear tidal component is calculated as follows:

$$\tau_{tide}^{\mathrm{btm}} = -\rho_0 C_D \left|\frac{\mathbf{U}_b + \mathbf{U}_{lt}}{\eta + H}\right| \mathbf{T}_\theta \frac{\mathbf{U}_{lt}}{\eta + H}. \tag{18.20}$$

Horizontal viscosity parameterization is the Laplacian operator with the viscosity coefficient fixed in time.

The space-time finite difference of the tidal model is omitted, since it is the same as that of the barotropic model. A sequence of operations explained in Section 7.4.1 is applied to both basic and tidal components. But now we use surface height at time level $t = t_n$ as the height of column applied to the pressure gradient term during the barotropic subcycle.

18.2 Incorporation of the tidal effects into MRI.COM

The solution method to predict the two components of the barotropic mode is schematically illustrated by Figure 18.1. On starting the barotropic submodel, the total barotropic mode is split into the basic and the linear tidal components (U_b, η_b, U_{lt}, and η_{lt}) and the time evolution is calculated separately. A simple sum of the two modes is returned to the main part of OGCM. For the next step, the total and the linear tidal components are retained. The basic component is calculated by subtracting the tidal part from the total.

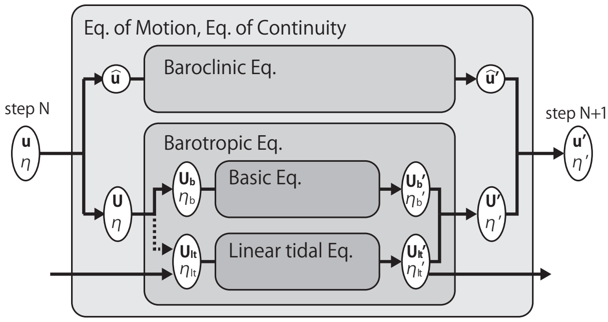

Figure18.1 Schematic figure of the separating basic and linear tidal component of the barotropic mode

The weighted average of the barotropic fields is written as follows:

$$\begin{aligned}\langle \mathbf{U}_{lt} \rangle^{n+1} =& \mathbf{U}_{lt}^0 - \Delta t_{\mathrm{cl}} f \mathbf{k} \times \langle\langle \mathbf{U}_{lt} \rangle\rangle^{n+\frac{1}{2}} - \Delta t_{\mathrm{cl}} g (\eta^n + H) \left\langle\!\left\langle \nabla(\alpha\eta_{lt} - \beta\eta_{\mathrm{eq}}) \right\rangle\!\right\rangle^{n+\frac{1}{2}} \\ &+ \Delta t_{\mathrm{cl}} \langle\langle \mathbf{F}_{tide}^{\mathrm{horz}} \rangle\rangle^{n+\frac{1}{2}} + \frac{1}{\rho_0} \Delta t_{\mathrm{cl}} \langle\langle \tau_{tide}^{\mathrm{btm}} \rangle\rangle^{n+\frac{1}{2}},\end{aligned} \tag{18.21}$$

$$\langle \mathbf{U}_b \rangle^{n+1} = \mathbf{U}_b^0 - \Delta t_{\mathrm{cl}} f \mathbf{k} \times \langle\langle \mathbf{U}_b \rangle\rangle^{n+\frac{1}{2}} - \Delta t_{\mathrm{cl}} g (\eta^n + H) \left\langle\!\left\langle \nabla\eta_b \right\rangle\!\right\rangle^{n+\frac{1}{2}} + \Delta t_{\mathrm{cl}} \left(\mathbf{X} - \mathbf{F}_{lt}^{\mathrm{horz}} - \frac{\tau_{lt}^{\mathrm{btm}}}{\rho_0} \right)^n. \tag{18.22}$$

These are summed and then combined with the baroclinic part (7.49) and (7.50) to give

$$\begin{aligned}\frac{(\langle \mathbf{u} \rangle^{n+1} + \mathbf{u}'_{k-\frac{1}{2}} - \overline{\mathbf{u}'}^z)\Delta z_{k-\frac{1}{2}}^{n+1} - \mathbf{u}_{k-\frac{1}{2}}^n \Delta z_{k-\frac{1}{2}}^n}{\Delta t_{\mathrm{cl}}} =& - f\mathbf{k} \times \langle\langle \mathbf{u} \rangle\rangle^{n+\frac{1}{2}} \Delta z_{k-\frac{1}{2}}^{n+1} - f\mathbf{k} \times [\mathbf{u}_{k-\frac{1}{2}} \Delta z_{k-\frac{1}{2}}]^{n+\frac{1}{2}} + f\mathbf{k} \times \overline{[\mathbf{u}]^{n+\frac{1}{2}}}^z \Delta z_{k-\frac{1}{2}}^{n+1} \\ &- g\Delta z_{k-\frac{1}{2}}^{n+1} \frac{\eta^n + H}{\eta^{n+1} + H} \left\langle\!\left\langle \nabla(\eta_b + \alpha\eta_{lt} - \beta\eta_{\mathrm{eq}}) \right\rangle\!\right\rangle^{n+\frac{1}{2}} \\ &+ \Delta z_{k-\frac{1}{2}}^{n+1} \frac{\langle\langle \mathbf{F}_{tide}^{\mathrm{horz}} \rangle\rangle^{n+\frac{1}{2}} - (\mathbf{F}_{lt}^{\mathrm{horz}})^n}{\eta^{n+1} + H} + \Delta z_{k-\frac{1}{2}}^{n+1} \frac{\langle\langle \tau_{tide}^{\mathrm{btm}} \rangle\rangle^{n+\frac{1}{2}} - (\tau_{lt}^{\mathrm{btm}})^n}{\rho_0(\eta^{n+1} + H)} + \Delta z_{k-\frac{1}{2}}^{n+1} \mathbf{F}^n.\end{aligned} \tag{18.23}$$

18.3 Tidal potential

The tide producing potential η_{eq} of MRI.COM includes the eight primary constituents that consist of the four largest equilibrium tides of the diurnal and semi-diurnal species (K_1, O_1. P_1, Q_1, M_2, S_2, N_2, K_2). Following Schwiderski

Table18.1 Constants of major tidal modes

n	Tidal Mode	K, m	σ, 10^{-4} /sec	χ, deg
1	K_1, declination luni-solar	0.141565	0.72921	$h_0 + 90$
2	Q_1, principal lunar	0.100514	0.67598	$h_0 - 2s_0 + 90$
3	P_1, principal solar	0.046843	0.72523	$-h_0 - 90$
4	Q_1, elliptical lunar	0.019256	0.64959	$h_0 - 3s_0 + p_0 - 90$
5	M_2, principal lunar	0.242334	1.40519	$2h_0 - 2s_0$
6	S_2, principal solar	0.112841	1.45444	0
7	N_2, elliptical lunar	0.046398	1.37880	$2h_0 - 3s_0 + p_0$
8	K_2, declination luni-solar	0.030704	1.45842	$2h_0$

Here h_0, s_0, and p_0 are the mean longitudes of the sun and moon and the lunar perigee at Greenwich midnight: $h_0 = 279.69668 + 36000.768930485T + 3.03 \times 10^4 T^2$, $s_0 = 270.434358 + 481267.88314137T - 0.001133T^2 + 1.9 \times 10^{-6}T^3$, $p_0 = 334.329653 + 4069.0340329575T - 0.010325T^2 - 1.2 \times 10^{-5}T^3$, where $T = (27392.500528 + 1.0000000356D)/36525$, $D = d + 365(y - 1975) + \mathrm{Int}[(y - 1973)/4]$, d is the day numbe of the year ($d = 1$ for January 1), $y \geq 1975$ is the year number, and $\mathrm{Int}[x]$ is the integral part of x.

(1980), we write the n-th ($n = [1, 4]$) diurnal equilibrium tide as

$$\begin{aligned}\eta_{\mathrm{eq},n} &= K_n \cos^2\phi \cos(\sigma_n t + \chi_n + 2\lambda) \\ &= K_n \cos^2\phi [\cos(\sigma_n t + \chi_n) \cos 2\lambda - \sin(\sigma_n t + \chi_n) \sin 2\lambda],\end{aligned} \tag{18.24}$$

and the n-th semi-diurnal component ($n = [5, 8]$) as

$$\begin{aligned}\eta_{\mathrm{eq},n} &= K_n \sin 2\phi \cos(\sigma_n t + \chi_n + \lambda) \\ &= K_n \sin 2\phi [\cos(\sigma_n t + \chi_n) \cos \lambda - \sin(\sigma_n t + \chi_n) \sin \lambda].\end{aligned} \tag{18.25}$$

The meaning of the mathematical symbols is as follows:

t	universal standard time in seconds
λ	east longitude
ϕ	latitude
K	amplitude of partial equilibrium tide in meters
σ	frequency of partial equilibrium tide in sec^{-1}
χ	astronomical argument of partial equilibrium tide relative to Greenwich midnight

And constants are listed on Table 18.1.

18.4 Nesting experiment with tide

The tide scheme can work together with the nesting schemes in MRI.COM. This subsection describes briefly how the tide scheme works with on-line nesting and off-line one-way nesting methods. See Chapter 22 for the general explanation of the nesting schemes.

For on-line nesting, the tide models communicate each other between the parent low-resolution model and the child high-resolution model in order to predict tides at the next step in the same manner as the barotropic model. The lateral boundaries of the child model receive $\mathbf{U}_{lt}$ and η_{lt} from the parent model every steps of the barotropic mode in the both cases of one-way and two-way nesting. On the other hand, in the case of two-way nesting with the "replace" configuration, the internal region of the parent model receives them from the child model.

For off-line one-way nesting, the communication process of the tide model is also the same as that of the barotropic model. The parent model saves $\mathbf{U}_{lt}$ and η_{lt} on bands corresponding to the lateral boundaries of the child model. The child model reads the files and interpolates them spatially and temporally to the lateral boundaries every barotropic steps. It should be noted that other lateral boundary data output by the parent model, such as three-dimensional velocity, sea surface height and x and y transports, also includes the linear tidal component. That is, the parent model saves $\mathbf{u}$, $\mathbf{U}_b + \mathbf{U}_{lt}$ and $\eta_b + \eta_{lt}$.

As a special function for off-line one-way nesting, the child model can be executed under tidal forcing even when the parent model does not include tides. That is, users can run the parent model without `TIDE`, while the child model with it. In this case, in advance, users have to create files for the child model boundaries of $\mathbf{U}_{lt}$ and η_{lt} in the same format as the parent model output. The MXE (see Section 25.6) package offers a tool for it based on the Matsumoto et al. (2000)'s dataset in the directory prep/nest/offline/. In addition, boundary files output by the parent model do not include the tidal component, which is different from the case that the parent model includes tides. Users have to specify namelist to handle this difference (l_parent_include_tide = .false. in nml_submodel_tide).

18.5 Usage

When TIDE is added to OPTIONS, the tide model is coupled. Specify parameters of the tide model by namelist nml_tide_model, the initial condition by nml_tide_run, and forced tidal constituents by nml_tide_forcing. If a specific state is used for the initial condition, three restart files must be prepared.

Chapter 19

Biogeochemical model

There are several options for Biogeochemical models in MRI.COM. These biogeochemical models have been developed for both ocean-only and coupled ocean-atmosphere-vegetation carbon cycle studies. They feature an explicit representation of a marine ecosystem, which is assumed to be limited by light, temperature, and nutrients availability. This chapter describes the details of the biogeochemical models.

19.1 Inorganic carbon cycle and biological model

Biogeochemical models are composed of inorganic carbon-cycle and ecosystem component models. In the inorganic carbon-cycle component, the partial pressure of CO_2 at the sea surface (pCO_2) is diagnosed from the values of dissolved inorganic carbon (DIC) and Alkalinity (Alk) at the sea surface, which should be affected by the ecosystem component. The difference in pCO_2 between the atmosphere and ocean determines ocean CO_2 uptake from or release to the atmosphere and is essential for simulating the CO_2 concentration in the atmosphere. Inorganic carbonate chemistry and partial pressure physics are well understood and can be reproduced with fair accuracy. The ecosystem component deals with various biological activities, and gives sources and sinks of the nutrients, DIC, Alk, and dissolved oxygen through these activities. Our knowledge of these activities is far from complete, and they are difficult to estimate even in state-of-the-art models.

There are many biological models and methods for calculating the ecosystem components. One of the simplest biological models has only one nutrient component (such as PO_4) as a prognostic variable and calculates neither phytoplankton nor zooplankton explicitly. In this case, the export of biologically generated soft tissue (organic matter) and hard tissue (carbonate) to the deep ocean, collectively known as the biological pump, is parameterized in terms of temperature, salinity, shortwave radiation, and nutrients.

A Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) model is more complex than the above model, but still a simple biological model. The NPZD model has four prognostic variables (nutrient, phytoplankton, zooplankton, and detritus). Though parameterized in a simple form, basic biological activities, such as photosynthesis, excretion, grazing, and mortality are explicitly calculated.

More complex models classify phytoplankton and zooplankton into several groups, and deal with many complex interactions between them. In general, it is expected that the more complex the biological model becomes the more realistic pattern the model can simulate. However, because of our incomplete knowledge about the biological activities, the complex models do not always yield better results, even though they require more computer resources.

To simulate the carbon cycle in the ocean, some biological processes should be calculated in the ecosystem component to obtain DIC at the sea surface. However, the carbon cycle component is not always necessary when our interests are to simulate the ecosystem itself. The Ocean Carbon-Cycle Model Intercomparison Project (OCMIP) protocols and studies of Yamanaka and Tajika (1996) and Obata and Kitamura (2003) focus on the former carbon cycle in the ocean, and the ecosystem components in these studies are quite simple. Biogeochemical models adopted in MRI.COM are classified in this category. The latter studies usually use complex biological models such as NEMURO (Kishi et al., 2001). Of course, this type of model could be adopted as an ecosystem component of the biogeochemical model in the former studies in hopes of better simulation of carbon cycle.

Originally, the carbon cycle component followed the OCMIP protocols (Orr et al., 1999) whose authority is recognized in the community. Recently, biogeochemical protocols for CMIP6 require that the former OCMIP code should be replaced by *mocsy* routines (Orr and Epitalon, 2015) to use the equilibrium constants recommended for best practices (Orr et al., 2017). MRI.COM can choose either option by setting a namelist. Here we describe the procedure using the *mocsy* routines.

MRI.COM has several options for the ecosystem component. At present, MRI.COM can incorporate the Obata and Kitamura model (Obata and Kitamura, 2003) or an NPZD model based on Oschlies (2001). The biogeochemical model of MRI.COM is largely based on Schmittner et al. (2008) when an NPZD model is adopted as an ecosystem component.

Basic units in MRI.COM are cgs, but in these biogeochemical subroutines, we use MKS units for the sake of future development. We use mol/m^3 for the units of nutrients. When the coefficients of their model are applied, they should be converted to the corresponding units.

19.2 Governing equations

Here we describe the biogeochemical models of MRI.COM. When an NPZD model is incorporated as the ecosystem component, the governing equations are as follows. When Obata and Kitamura model is used instead of the NPZD model, the first four biogeochemical compartments (DIC, Alk, PO_4, and O_2) are used.

$$\frac{\partial \mathrm{DIC}}{\partial t} = -\mathcal{A}(\mathrm{DIC}) + \mathcal{D}(\mathrm{DIC}) + S_b(\mathrm{DIC}) + J_v(\mathrm{DIC}) + J_g(\mathrm{DIC}), \tag{19.1}$$

$$\frac{\partial \mathrm{Alk}}{\partial t} = -\mathcal{A}(\mathrm{Alk}) + \mathcal{D}(\mathrm{Alk}) + S_b(\mathrm{Alk}) + J_v(\mathrm{Alk}), \tag{19.2}$$

$$\frac{\partial [\mathrm{PO_4}]}{\partial t} = -\mathcal{A}([\mathrm{PO_4}]) + \mathcal{D}([\mathrm{PO_4}]) + S_b([\mathrm{PO_4}]), \tag{19.3}$$

$$\frac{\partial [\mathrm{O_2}]}{\partial t} = -\mathcal{A}([\mathrm{O_2}]) + \mathcal{D}([\mathrm{O_2}]) + S_b([\mathrm{O_2}]) + J_g([\mathrm{O_2}]), \tag{19.4}$$

$$\frac{\partial [\mathrm{NO_3}]}{\partial t} = -\mathcal{A}([\mathrm{NO_3}]) + \mathcal{D}([\mathrm{NO_3}]) + S_b([\mathrm{NO_3}]), \tag{19.5}$$

$$\frac{\partial [\mathrm{PhyPl}]}{\partial t} = -\mathcal{A}([\mathrm{PhyPl}]) + \mathcal{D}([\mathrm{PhyPl}]) + S_b([\mathrm{PhyPl}]), \tag{19.6}$$

$$\frac{\partial [\mathrm{ZooPl}]}{\partial t} = -\mathcal{A}([\mathrm{ZooPl}]) + \mathcal{D}([\mathrm{ZooPl}]) + S_b([\mathrm{ZooPl}]), \tag{19.7}$$

$$\frac{\partial [\mathrm{Detri}]}{\partial t} = -\mathcal{A}([\mathrm{Detri}]) + \mathcal{D}([\mathrm{Detri}]) + S_b([\mathrm{Detri}]), \tag{19.8}$$

where $\mathcal{A}()$ and $\mathcal{D}()$ represent advection and diffusion operator, respectively, and $S_b()$ is source minus sink due to the biogeochemical activities. The square brackets mean dissolved concentration in $\mathrm{mol/m^3}$ of the substance within them. The terms represented by $J_g()$ and $J_v()$ are the air-sea gas fluxes (Section 19.3.1) and the dilution and concentration effects of evaporation and precipitation on DIC and Alk (Section 19.3.2), respectively, which appear only at the sea surface. The term $J_g()$ is calculated based on the OMIP protocol by using the air-sea gas transfer velocity and concentration in the seawater. The term $J_v()$ appears only when the salinity flux is given virtually instead of the increase or decrease of the volume at the surface layers due to evaporation and precipitation.

19.3 Carbon cycle component

To estimate J_g and J_v, we follow the protocol of OMIP, which is described in detail in Orr et al. (2017). The program to calculate them is based on the *mocsy* subroutines. We have modified this subroutine so that it can be used in the calculation of the MRI.COM code.

19.3.1 Air-sea gas exchange fluxes at the sea surface (J_g)

The air-sea gas transfer must be calculated for DIC and $[\mathrm{O_2}]$. The terms $J_g(\mathrm{DIC})$ and $J_g([\mathrm{O_2}])$ appear only in the uppermost layer. When these fluxes are expressed as $F_g(\mathrm{DIC})$ and $F_g([\mathrm{O_2}])$, $J_g(\mathrm{DIC})$ and $J_g([\mathrm{O_2}])$ are given as follows:

$$J_g(\mathrm{DIC}) = \frac{F_g(\mathrm{DIC})}{\Delta z_{\frac{1}{2}}}, \tag{19.9}$$

$$J_g([\mathrm{O_2}]) = \frac{F_g([\mathrm{O_2}])}{\Delta z_{\frac{1}{2}}}, \tag{19.10}$$

where

$$F_g(\mathrm{DIC}) = K_w^{\mathrm{CO_2}} * ([\mathrm{CO_2}]_{\mathrm{sat}} - [\mathrm{CO_2}]_{\mathrm{surf}}), \tag{19.11}$$

$$F_g([\mathrm{O_2}]) = K_w^{\mathrm{O_2}} * ([\mathrm{O_2}]_{\mathrm{sat}} - [\mathrm{O_2}]_{\mathrm{surf}}), \tag{19.12}$$

and $\Delta z_{\frac{1}{2}}$ is the thickness of the first layer of the model. Here a standard gas transfer formulation is adopted. $K_w^{\mathrm{CO_2}}$ and $K_w^{\mathrm{O_2}}$ are their gas transfer velocity, $[\mathrm{CO_2}]_{\mathrm{sat}}$ and $[\mathrm{O_2}]_{\mathrm{sat}}$ are their saturation concentrations with respect to the atmosphere, and $[\mathrm{CO_2}]_{\mathrm{surf}}$, $[\mathrm{O_2}]_{\mathrm{surf}}$, are their surface concentration.

a. Gas transfer velocity

The coefficients $K_w^{CO_2}$ and $K_w^{O_2}$ are the air-sea gas transfer (piston) velocity and are diagnosed as follows:

$$K_w^{CO_2} = a \left(\frac{Sc^{CO_2}}{660} \right)^{-1/2} U_{10}^2 (1 - f_i), \tag{19.13}$$

$$K_w^{O_2} = a \left(\frac{Sc^{O_2}}{660} \right)^{-1/2} U_{10}^2 (1 - f_i), \tag{19.14}$$

where

- f_i is the fraction of the sea surface covered with ice,
- U_{10} is 10 m scalar wind speed,
- a is the coefficient of 0.251 $\mathrm{cm\,hr^{-1}/(m\,s^{-1})^2}$, which will give K_w in units of $\mathrm{cm\,hr^{-1}}$ when winds speeds (U_{10}) are in $\mathrm{m\,s^{-1}}$. This is specified in the OMIP protocol,
- Sc^{CO_2} and Sc^{O_2} are the Schmidt numbers for CO_2 and O_2.

b. Saturation concentration with respect to the atmosphere

The surface water gas concentration in equilibrium with the atmosphere (saturation concentrations) for gas A has the following relationship.

$$\begin{aligned}
[A]_{\mathrm{sat}} &= K_0 f_A && (19.15)\\
&= K_0 C_f\, p_A && (19.16)\\
&= K_0 C_f\, (P_a - p\mathrm{H_2O}) x_A && (19.17)\\
&= \phi_A x_A && (19.18)\\
&\simeq \frac{P_a}{P_a^\circ} \phi_A^0 x_A \qquad \text{(with errors less than 0.1\%)} && (19.19)\\
&= \frac{P_a}{P_a^\circ} [A]_{\mathrm{sat}}^0, && (19.20)
\end{aligned}$$

where K_0 is its solubility, f_A is its atmospheric fugacity, C_f is its fugacity coefficient, p_A is its atmospheric partial pressure, x_A is its mole fraction in dry air, P_a is total atmospheric pressure, P_a° is the standard atmosphere (=1013.25 hPa), $p\mathrm{H_2O}$ is water vapor pressure at saturation, ϕ_A is its solubility function, ϕ_A^0 and $[A]_{\mathrm{sat}}^0$ are its solubility function and saturation concentration at the reference pressure.

Orr et al. (2017) explain two approaches to calculate the saturation concentration.

The first approach uses equations (19.19) or (19.20), computing solubility function ϕ_A^0 or saturation concentration $[A]_{\mathrm{sat}}^0$ at the reference atmospheric pressure first, and then converting them to those at the atmospheric pressure P_a. For gases that are often used as tracers in oceanography such as CFCs, ϕ_A^0 can be expressed as a function of in-situ temperature and salinity by using empirical fit. For oxygen, an empirical fit for $[O_2]_{\mathrm{sat}}^0$ is available. This is a rather conventional approach, and is used in MRI.COM except for carbon.

The other approach uses equations (19.15)-(19.17), computing its partial pressure $p_A \equiv (P_a - p\mathrm{H_2O}) x_A$, then multiplying by $K' \equiv K_0 C_f$, or multiplying by C_f and then multiplying by K_0 if atmospheric fugacity (f_A) is needed. For typical gases, K', K_0, and C_f are available as a function of temperature and salinity. The *mocsy* routine adopts this approach, and MRI.COM also follows it for calculating carbon flux at the surface.

c. Computing CO_2 concentrations at the surface

In the *mocsy* routine, $[CO_2]_{\mathrm{surf}}$ is diagnosed every step from DIC, Alk, temperature, salinity, $[PO_4]$, and silicate concentration at the surface. After $[CO_2]_{\mathrm{surf}}$ is computed, fugacity (f_{CO_2}) and partial pressure ($p\mathrm{CO_2}$) of the ocean surface are computed as

$$f_{CO_2} = [CO_2]_{\mathrm{surf}} / K_0, \tag{19.21}$$

$$p\mathrm{CO_2} = f_{CO_2} / C_f, \tag{19.22}$$

where K_0 and C_f are estimated by using temperature and salinity.

Diagnosis of $[CO_2]_{surf}$ is the most complex of the above calculations and has the largest computational cost. To be precise, diagnosis of $[CO_2]$ actually means diagnosing $[CO_2] + [H_2CO_3]$, which are difficult to distinguish analytically. These two species are usually combined and the sum is expressed as the concentration of a hypothetical species, $[CO_2^*]$ or $[H_2CO_3^*]$. Here, the former notation is used. The relationship between this $[CO_2^*]$ and DIC is as follows:

$$\mathrm{DIC} = [CO_2] + [H_2CO_3] + [HCO_3^-] + [CO_3^{2-}] \tag{19.23}$$

$$= [CO_2^*] \qquad\qquad + [HCO_3^-] + [CO_3^{2-}]. \tag{19.24}$$

In the OMIP protocol, the following equations are solved to obtain $[CO_2^*]$.
The equilibrium constants for dissociation reactions are:

$$K_1 = \frac{[H^+][HCO_3^-]}{[CO_2^*]} \qquad K_2 = \frac{[H^+][CO_3^{2-}]}{[HCO_3^-]}, \tag{19.25}$$

$$K_\mathrm{B} = \frac{[H^+][B(OH)_4^-]}{[B(OH)_3]}, \tag{19.26}$$

$$K_\mathrm{1P} = \frac{[H^+][H_2PO_4^-]}{[H_3PO_4]} \qquad K_\mathrm{2P} = \frac{[H^+][HPO_4^{2-}]}{[H_2PO_4^-]} \qquad K_\mathrm{3P} = \frac{[H^+][PO_4^{3-}]}{[HPO_4^{2-}]}, \tag{19.27}$$

$$K_\mathrm{Si} = \frac{[H^+][SiO(OH)_3^-]}{[Si(OH)_4]}, \tag{19.28}$$

$$K_W = [H^+][OH^-], \tag{19.29}$$

$$K_\mathrm{S} = \frac{[H^+]_F[SO_4^{2-}]}{[H\,SO_4^-]}, \tag{19.30}$$

and

$$K_\mathrm{F} = \frac{[H^+]_F[F^-]}{[HF]}, \tag{19.31}$$

where $[H^+]$ is the hydrogen ion concentration in sea water and $[H^+]_F$ is the free hydrogen ion concentration. There is another scale for the hydrogen ion concentration, the total hydrogen ion concentration $[H^+]_T$. The subscript T means "total" and F means "free." These three hydrogen ion concentrations are related as follows:

$$[H^+] = [H^+]_F\left(1 + \frac{S_T}{K_\mathrm{S}} + \frac{F_T}{K_\mathrm{F}}\right), \tag{19.32}$$

$$\text{and} \qquad [H^+]_T = [H^+]_F\left(1 + \frac{S_T}{K_\mathrm{S}}\right). \tag{19.33}$$

There are three pH scales corresponding to these three hydrogen ion concentrations.

The equilibrium constants K_x are given as a function of temperature, salinity, and pH. Note that the equilibrium constants are given in terms of concentrations, and that all constants are referenced to the seawater pH scale, except for K_S, which is referenced to the free pH scale.

The total dissolved inorganic carbon, boron, phosphate, silicate, sulfate, and fluoride are expressed as follows:

$$\mathrm{DIC} = [CO_2^*] + [HCO_3^-] + [CO_3^{2-}], \tag{19.34}$$

$$B_\mathrm{T} = [B(OH)_3] + [B(OH)_4^-], \tag{19.35}$$

$$P_\mathrm{T} = [H_3PO_4] + [H_2PO_4^-] + [HPO_4^{2-}] + [PO_4^{3-}], \tag{19.36}$$

$$Si_\mathrm{T} = [Si(OH)_4] + [Si(OH)_3^-], \tag{19.37}$$

$$S_T = [HSO_4^-] + [SO_4^{2-}], \tag{19.38}$$

and

$$F_T = [HF] + [F^-]. \tag{19.39}$$

Alkalinity used in this calculation is defined as follows:

$$\begin{aligned}\mathrm{Alk} = &[HCO_3^-] + 2[CO_3^{2-}] + [B(OH)_4^-] + [OH^-] + [HPO_4^{2-}] + 2[PO_4^{3-}] + [SiO(OH)_3^-] \\ &- [H^+]_F - [HSO_4^-] - [HF] - [H_3PO_4].\end{aligned} \tag{19.40}$$

These expressions exclude the contribution of NH_3, HS^-, and S^{2-}.

If we assume that DIC, Alk, P_T, and Si_T are known, this system contains 18 equations with 18 unknowns, so they can be solved. Detailed solver method has been described in Orr and Epitalon (2015). The concentration $[Si_T]$ is not predicted in the biogeochemical model adopted in MRI.COM but is rather specified by the annual mean value from WOA2013.

19.3.2 Dilution and concentration effects of evaporation and precipitation on DIC and Alk

The dilution and concentration effects of evaporation and precipitation significantly impact the concentrations of some chemical species in seawater. This is particularly true for DIC and Alk, which have large background concentrations compared with their spatial variability. MRI.COM uses a free surface, so the impact of evaporation and precipitation is straightforward to model as far as `WFLUX` is used. When `WFLUX` (or this option) is not used, salinity flux is diagnosed and applied instead of the freshwater flux. In this case, the dilution and concentration effect of evaporation (E) and precipitation (P) should be taken into account. Here, they are parameterized as virtual DIC and Alk fluxes, similar to the virtual salt flux used in physical ocean GCMs.

In MRI.COM, the tendency of salinity due to the virtual salt flux is given by

$$\texttt{sflux}(i,j) = -(P - E) * S(i,j,1)/\Delta z, \tag{19.41}$$

where $S(i,j,1)$ and Δz are the salinity and thickness of the uppermost layer. Note that the variable $\texttt{sflux}(i,j)$ is *not* the salinity flux but the time change rate of salinity due to the flux even though the spelling brings up the image of the flux.

In MRI.COM, DIC and Alk are modified by the virtual salt flux as follows:

$$J_v(\mathrm{DIC}(i,j,1)) = \texttt{sflux}(i,j)/S(i,j,1) * \mathrm{DIC}(i,j,1), \tag{19.42}$$
$$J_v(\mathrm{Alk}(i,j,1)) = \texttt{sflux}(i,j)/S(i,j,1) * \mathrm{Alk}(i,j,1). \tag{19.43}$$

Strictly speaking, air-sea fluxes of fresh water impact other species. However, these modifications are not usually applied because their spatial variabilities are significantly greater than those of DIC and Alk.

In the OMIP protocol, as well as the previous OCMIP protocol, the global averaged salinity S_g is used instead of $S(i,j,1)$ in equations(19.42,19.43). In addition, globally integrated $J_v(\mathrm{DIC})$ and $J_v(\mathrm{Alk})$ are set to 0. In MRI.COM, these modifications are not the default considering the use in regional ocean models.

19.4 Obata and Kitamura model

This section was contributed by A. Obata.

The Obata and Kitamura model used in MRI.COM simply represents the source and sink terms of DIC, Alk, $[PO_4]$, and $[O_2]$ due to the biogeochemical activities: new production driven by insolation and phosphate concentration in the surface ocean, its export to depth, and remineralization in the deep ocean. According to the Michaelis-Menten kinetics (Dugdale, 1967), phosphorus in the new production exported to depth (ExprodP) is parameterized as $rL[PO_4]^2/([PO_4]+k)$, where r is a proportional factor ($r = 0.9\,\mathrm{yr}^{-1}$), L is the insolation normalized by the annual mean insolation on the equator, and k is the half-saturation constant ($k = 0.377\,\mathrm{mmol/m^{-3}}$). The values of r and k are adjusted to reproduce the optimum atmospheric CO_2 concentration and ocean biogeochemical distribution for the preindustrial state of the model. The relationship between the changes in the chemical composition of seawater and the composition of particulate organic matter (POM) is assumed to follow the Redfield ratio $\mathrm{P:N:C:O_2} = 1 : 16 : 106 : -138 \equiv 1 : R_{np} : R_{cp} : -R_{op}$, where $R_{ab} = A/B$ and "o" represents O_2 (Redfield et al., 1963). The rain ratio of calcite to particulate organic carbon (POC) is 0.09, which is in the range proposed by Yamanaka and Tajika (1996). The surface thickness where the export production occurs is fixed at 60 m. The vertical distribution of POM and calcite vertical flux below a depth of 100 m is proportional to $(z/100\text{m})^{-0.9}$ and $\exp(-z/3500\text{m})$ (z is the depth in meters), respectively, following the work of Yamanaka and Tajika (1996). The remineralization of POM (RemiP for phosphorus) and the dissolution of calcite (SolnCa) at depth are parameterized by these fluxes. Oxygen saturation is prescribed at the sea surface. The solubility of oxygen is computed from the formula of Weiss (1970). Source and sink terms of $S_b()$ representing the above processes are as follows:

$$S_b(\text{DIC}) = R_{cp} * \text{RemiP} + \text{SolnCa} - R_{cp} * \text{ExprodP} \tag{19.44}$$

$$S_b(\text{Alk}) = 2 * \text{SolnCa} + R_{np} * \text{ExprodP} - R_{np} * \text{RemiP} \tag{19.45}$$

$$S_b([\text{PO}_4]) = \text{RemiP} - \text{ExprodP} \tag{19.46}$$

$$S_b([\text{O}_2]) = -R_{op} * S_b([\text{PO}_4]) \tag{19.47}$$

19.5 NPZD model

The NPZD model used in MRI.COM is constructed on the assumptions that the biological elemental composition ratio is nearly constant (Redfield ratio) and that the concentration of organisms can be estimated by nitrogen or phosphorus. The prognostic variables of nitrate (NO_3), phytoplankton (PhyPl), zooplankton (ZooPl), and detritus (Detri) are normalized in terms of nitrogen 1 mol/m^3. For example, [PhyPl] represents the concentration of phytoplankton estimated in terms of nitrogen in one cubic meter (N mol/m^3). The increase and decrease of carbon can be diagnosed by multiplying by R_{cn}.

Source and sink terms $S_b()$ calculated in the NPZD model are as follows. Those for DIC and Alk, $S_b(\text{DIC})$ and $S_b(\text{Alk})$, used for calculating the carbon cycle, are described later in this section.

$$S_b([\text{PhyPl}]) = \text{Priprod} - \text{MortP1} - \text{MortP2} - \text{GrP2Z} \tag{19.48}$$

$$S_b([\text{ZooPl}]) = assim * \text{GrP2Z} - \text{Excrtn} - \text{MortZ} \tag{19.49}$$

$$S_b([\text{Detri}]) = [(1 - assim) * \text{GrP2Z} + \text{MortP2} + \text{MortZ}] - \text{RemiD} - w_{detri}\frac{\partial \text{Detri}}{\partial z} \tag{19.50}$$

$$S_b([\text{NO}_3]) = \text{MortP1} + \text{Excrtn} + \text{RmeiD} - \text{PriProd} \tag{19.51}$$

$$S_b([\text{PO}_4]) = R_{pn} * S_b([\text{NO}_3]) \tag{19.52}$$

$$S_b([\text{O}_2]) = -R_{on} * R_{np} * S_b([\text{PO}_4]) \tag{19.53}$$

There is no input from the atmosphere such as nitrogen fixation in the above equations, so the sum of these five equations becomes zero at each grid point except for the term for detritus sinking ($-w_{detri}\frac{\partial \text{Detri}}{\partial z}$). The term for detritus sinking expresses the biological pump, whose role is to remove nutrients from the upper layers and transport them into the deep ocean where the plankton cannot use the nutrients. When vertically integrated, the sum of each grid is 0 even though this sinking term is included. The nutrients are transported horizontally through physical processes such as advection and diffusion.

In general, the nitrate limitation is more severe than the phosphate limitation so it is not always necessary to calculate phosphate. However, in the simpler model of Obata and Kitamura (2003), phosphate is used as a prognostic variable. So, to be consistent, phosphate is calculated in the ecosystem component of MRI.COM. Next, we elaborate on the above equations.

19.5.1 Description of each term

- Priprod = $J(I, N, P) * [\text{PhyPl}]$
  Primary production expresses photosynthesis (described in detail in the next subsection).
- MortP1 = $\phi_P * [\text{PhyPl}]$
  The conversion of mortality phytoplankton directly into nutrients. This term was introduced by Oschlies (2001) to increase the primary production of subtropical gyre, where the nutrient limit is severe.
- MortP2 = $\phi_{PP} * [\text{PhyPl}]^2$
  The conversion from phytoplankton to detritus (normal mortality of phytoplankton).
- GrP2Z = $G(P) * [\text{ZooPl}]$
  The grazing of zooplankton. There are a number of parameterizations of grazing. In this formulation, this is given as $G(P) = g * \epsilon * [\text{PhyPl}]^2/(g + \epsilon * [\text{PhyPl}]^2)$. Among the grazing, the ratio $assim$ is used for the growth of zooplankton, and the remainder $(1 - assim)$ is converted to detritus.
- Excrtn = $d * [\text{ZooPl}]$
  Excretion of zooplankton. The excretion is dissolute and directly returned to nutrients (NO_3).

- MortZ = $\phi_Z * [\text{ZooPl}]^2$
  The conversion from the zooplankton to detritus (mortality of zooplankton).
- RemiD = $\phi_D * [\text{Detri}]$
  Remineralization of detritus. This is converted to nutrients through the activity of bacteria.

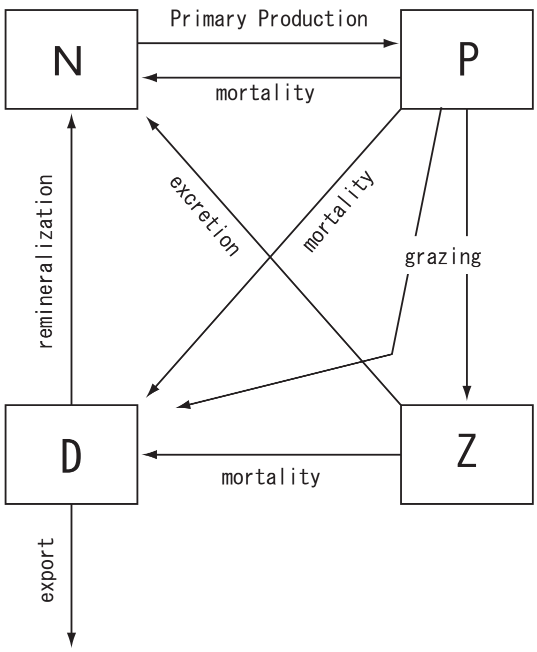

Figure19.1 Schematic of NPZD model

19.5.2 Primary Production

The growth rate of phytoplankton is limited by the irradiance (I) and nutrients. This limitation is expressed in several ways. Here we adopt an expression with a minimum function:

$$J(I, \mathrm{N}, \mathrm{P}) = \min(J_I, J_N, J_P), \tag{19.54}$$

where J_I denotes the purely light-limited growth rate, and J_N and J_P are nutrient-limited growth rates that are functions of nitrate or phosphate.

The light-limited growth is calculated as follows:

$$J_I = \frac{J_{max}\alpha I}{\left[J_{max}^2 + (\alpha I)^2\right]^{1/2}}. \tag{19.55}$$

Here, J_{max} is the light-saturated growth, which depends on temperature based on Eppley (1972) as

$$J_{max} = a \cdot b^{c\theta}, \tag{19.56}$$

where $a = 0.6\,\mathrm{day}^{-1}$, $b = 1.066$, and $c = 1\,(^\circ\mathrm{C})^{-1}$. Note that the default values in MRI.COM are based on Schmittner et al. (2008) and differ from these values (see Table 19.1). Equation (19.55) is called Smith-type growth. The coefficient α in the equation is "the initial slope of photosynthesis versus irradiance (P-I) curve," that is,

$$\alpha = \lim_{I \to 0} \frac{\partial J_I}{\partial I}. \tag{19.57}$$

Thus, it represents how sensitive J_I is to the irradiance when the light is weak.

Irradiance (I) depends on the angle of incidence and the refraction and absorption in the seawater.

$$I = I_{z=0}\ \mathrm{PAR}\ \exp\left(-k_w \tilde{z} - k_e \int_0^{\tilde{z}} P dz\right), \tag{19.58}$$

where $I_{z=0}$ denotes the downward shortwave radiation at the sea surface, PAR is the photosynthetically active radiation ratio (0.43) and $\tilde{z} = z/\cos\theta = z/\sqrt{1-\sin^2\theta/1.33^2}$ is the effective vertical coordinate (positive downward) with 1.33 as the refraction index according to Snell's law relating the zenith angle of incidence in air (θ) to the angle of incidence in water. The angle of incidence θ is a function of the latitude ϕ and declination δ.

For the nutrient-limited growth rate (J_N and J_P), we adopt the Optimal Uptake (OU) equation instead of the classic Michaelis-Menten (MM) equation. For the classic MM equation, the nitrate-limited growth rate is expressed as

$$J_N = J_{MM} = \frac{J_{max}N}{K_N + N}, \tag{19.59}$$

where K_N is a half-saturation constant for NO_3 uptake. In contrast, the Optimal Uptake (OU) equation for a nitrate is expressed as follows:

$$J_N = J_{OU} = \frac{V_0 N}{N + 2\sqrt{\frac{V_0}{A_0}N} + \frac{V_0}{A_0}}, \tag{19.60}$$

where A_0 and V_0 are the potential maximum values of affinity and uptake rate, respectively (see Smith et al. (2009) for details). Optimal Uptake (OU) kinetics assumes a physiological trade-off between the efficiency of nutrient encounter at the cell surface and the maximum rate at which a nutrient can be assimilated (Smith et al., 2009). The key idea is that phytoplankton alters the number of its surface uptake sites (or ion channels), which determines the encounter timescale, versus internal enzymes, which assimilate the nutrients once encountered.

We set parameters V_0 and A_0 so that the rates of uptake, J_{MM} and J_{OU}, are equal at $N = K_N$. In addition, we fix the ratio $V_0/A_0 = \alpha_{OU}$, where α_{OU} is determined from fitting the data. This requires

$$V_0 = 0.5\left(1 + \sqrt{\frac{\alpha_{OU}}{K_N}}\right)^2 J_{max}. \tag{19.61}$$

Finally, we obtain

$$J_{OU} = \frac{V_0 N}{N + 2\sqrt{\alpha_{OU} N} + \alpha_{OU}}. \tag{19.62}$$

We use $\alpha_{OU} = 0.19$, which is determined from the fitting of $\log K_N$ vs $\log N$ in the wide range of N by Smith et al. (2009).

19.5.3 Variation of DIC and Alk due to biological activity

Production of DIC and Alk is controlled by changes in inorganic nutrients and calcium carbonate ($CaCO_3$), in molar numbers according to

$$S_b(\mathrm{DIC}) = S_b([\mathrm{PO_4}])R_{cp} - S_b([\mathrm{CaCO_3}]), \tag{19.63}$$

$$S_b(\mathrm{Alk}) = -S_b([\mathrm{NO_3}]) - 2 \cdot S_b([\mathrm{CaCO_3}]). \tag{19.64}$$

Thus, only these source and sink terms of DIC and Alk are estimated. Since $[PO_4]$ and $[NO_3]$ are prognostic variables, their source and sink are explicitly calculated by the biological model. In contrast, the downward movement of $CaCO_3$ is much faster than the modeled downward velocity of water mass, so $[CaCO_3]$ is not a prognostic variable, and its source (Pr) and sink (Di) are diagnosed by the following equation,

$$S_b([\mathrm{CaCO_3}]) = Pr([\mathrm{CaCO_3}]) - Di([\mathrm{CaCO_3}]). \tag{19.65}$$

Following Schmittner et al. (2008), the source term ($Pr([CaCO_3])$) of calcium carbonate is determined by the production of detritus as follows:

$$Pr([\mathrm{CaCO_3}]) = [(1 - assim) * [\mathrm{GrP2Z}] + [\mathrm{MortP2}] + [\mathrm{MortZ}]]\, R_{\mathrm{CaCO_3/POC}} R_{C:N}, \tag{19.66}$$

where *assim*, GrP2Z, MortP2, and MortZ are as described above. The sink term ($Di([CaCO_3])$) of calcium carbonate is parameterized as

$$Di([\mathrm{CaCO_3}]) = \int Pr([\mathrm{CaCO_3}])dz \cdot \frac{d}{dz}\left(\exp\left(-z/D_{\mathrm{CaCO_3}}\right)\right), \tag{19.67}$$

which expresses an instantaneous sinking with an e-holding depth of $D_{CaCO_3} = 3500$ m. In this equation, z is positive downward. This depth of 3500 m was estimated by Yamanaka and Tajika (1996) to reproduce the observed nutrient profile.

This value is standard and also is used in the simple biological model in the protocol of OCMIP. The vertical integral of the source minus sink should be zero. Thus, when the sea bottom appears before the sum becomes zero, the remaining calcium carbonate is assumed to be dissolved in the lowermost layer. By using the ratio $R_{\mathrm{CaCO_3/POC}} = 0.035$ used by Schmittner et al. (2008), the resultant global mean Rain ratio should be roughly consistent with the recently estimated range (0.07 to 0.11) based on various observations.

19.6 Usage

In MRI.COM, carbon, dissolved oxygen, and ecosystem model as well as other passive tracers like CFCs can be calculated together or separately. Options described in `configure.in` determines what components should be used in MRI.COM. The combination of the options also determines the number of passive tracers that is necessary for calculation. See detail for the program `tracer_vars.F90`.

- When `CARBON` is used, carbon component is calculated in the model. For the ocean only model, the `CBNHSTRUN` option should be used to set the atmospheric CO2. Its internal source or sink is calculated in an ecosystem model.
- When `CARBON` is used, dissolved oxygen component is calculated.
- When `CBNHSTRUN` is used, the 'AtmosphericxCO2' in ppm should be applied as additional atmospheric forcing. This is the standard configuration for ocean only run. See Chapter 14 for the detailed setting.
- When `O2` is used, dissolved oxygen component is calculated. Its internal source or sink is calculated in an ecosystem model.

For ecosystem model, MRI.COM can use `NPZD` or `OBT` options.

- When `NPZD` is used, an NPZD model is used as the ecosystem component. In addition, when `CHLMA94` option is used, the chlorophyll concentration is considered to calculate the shortwave penetration following Morel and Antoine (1994).
  The parameters of NPZD are set in namelist `nml_bioNPZD`. The default values are based on Schmittner et al. (2008) and listed on Table 19.1. If the parameters of Oschlies (2001) are used, the high nutrient-low chlorophyll (HNLC) region in the North Pacific is not appropriately expressed. This may be because the parameters of Oschlies (2001) are calibrated for the North Atlantic biological model. The commonly used unit of time in biological models is [day]. Thus, in the namelist, the time unit of the biological parameter is specified by using the unit [day]. In the model, the time unit is converted to seconds, [sec].
- When `OBT` is used, a simple biological model of Obata and Kitamura (2003) is applied as the ecosystem component.

Restart files for the passive tracers should be specified by `nml_restart`. Following is an example when `CARBON`, `O2`, and `NPZD` options are chosen. In this case, `numtrc_p = 8` should be specified in `configure.in`. See `docs/README_Restart.md` for more details.

An example specification of restart files when `CARBON`, `O2`, and `NPZD` options are chosen.

```
&nml_restart name='DIC', file_base='result/rs_dic'/
&nml_restart name='Total Alkalinity', file_base='result/rs_alk'/
&nml_restart name='Dissolved Oxygen', file_base='result/rs_o2'  /
&nml_restart name='Phosphate', file_base='result/rs_po4' /
&nml_restart name='Nitrate', file_base='result/rs_no3' /
&nml_restart name='Phytoplankton', file_base='result/rs_PhyPl'/
&nml_restart name='Zooplankton', file_base='result/rs_ZooPl'/
&nml_restart name='Detritus', file_base='result/rs_Detri'/
```

Table19.1 Parameters used for the NPZD ecosystem component (NPZD).

variable name	description	units	default value
alphabio	Initial slope of P-I curve	$(\mathrm{W\,m^{-2}})^{-1}\,\mathrm{day}^{-1}$	0.1d0
abio	Maximum growth rate parameter	day^{-1}	0.2d0
bbio	Maximum growth rate = a * b ** (c * T)		1.066d0
cbio			1.d0
PARbio	Photosynthetically active radiation		0.43d0
dkcbio	Light attenuation due to phytoplankton	$\mathrm{m^{-1}}\,(\mathrm{mol\,m^{-3}})^{-1}$	0.03d3
dkwbio	Light attenuation in the water	$\mathrm{m^{-1}}$	0.04d0
rk1bioNO3	Half-saturation constant for NO_3 uptake	$\mathrm{mol\,m^{-3}}$	0.7d-3
rk1bioPO4	Half-saturation constant for PO_4 uptake	$\mathrm{mol\,m^{-3}}$	0.0d0
alpha_ou	Fitting constant for Optical Uptake kinetics		0.19d0
gbio	Maximum grazing rate	day^{-1}	1.575d0
epsbio	Prey capture rate	$(\mathrm{mol\,m^{-3}})^{-2}\,\mathrm{day}^{-1}$	1.6d6
phiphy	Specific mortality/recycling rate	s^{-1}	0.014d0
phiphyq	Quadratic mortality rate	$(\mathrm{mol\,m^{-3}})^{-1}\,\mathrm{day}^{-1}$	0.05d3
a_npz	Assimilation efficiency		0.925d0
phizoo	Quadratic mortality of zooplankton	$(\mathrm{mol\,m^{-3}})^{-1}\,\mathrm{day}^{-1}$	0.34d3
d_npz	Excretion	day^{-1}	0.01d0
remina	Remineralization rate	day^{-1}	0.048d0
w_detr	Sinking velocity	$\mathrm{m\,day^{-1}}$	2.0d0
fac_wdetr	Arbitrary parameter for numerical stability. When the concentration of detritus in the n+1 st level is higher than fac_wdetr times that in the n th level, w_detr is set to 0 between n and n+1 level.		3.d0
c_mrtn	Dimensionless scaling factor for Martin et al. (1987) Phi(z) = Phi(zo) * (z/dp_mrtn)**(-c_mrtn)		0.858d0
Rcn	Molar elemental ratio (C/N)		7.d0
Ron	Molar elemental ratio (O_2/N)		10.d0
Rnp	Molar elemental ratio (N/P)		16.d0
dp_euph	Maximum depth of euphotic zone	m	150.d0
dp_mrtn	Characteristic depth of martin curve	m	100.d0
dp_eprdc	The depth where the bio-export is diagnosed.	m	100.d0
	This value should be less than dp_mrtn.		
Rcaco3poc	$CaCO_3$ over nonphotosynthetic POC production ratio		0.03d0
Dcaco3	$CaCO_3$ remineralization e-folding depth	m	3500.d0
shwv_intv_sec	Interval for calculating the irradiance and light-limited growth rate. This must be a divisor of the time step for tracer.	min	600.d0

19.7 Program structure

```
ogcm__ini
  |
  +-- ptrc_ctl__ini
  |      |
  |      +-- ptrc__ini
  |             |
  |             +-- cbn__ini
  |             |
  |             +-- cbn__ini_history
  |
  +-- tracer__ctl__ini
         |
         +-- tracer__ini
                |
                +-- bio__ini
```

```
             |
             +-- bio__ini_history

ogcm__run
  |
  +-- part_1
  |     |
  |     +-- ptrc_ctl__main
  |     |     |
  |     |     +-- ptrc__surfflux
  |     |     |    |
  |     |     |    +-- cbn__set_xco2a
  |     |     |    |
  |     |     |    +-- cbn__calc_virtual_flux
  |     |     |    |
  |     |     |    +-- cbn__calc_co2_flux
  |     |     |          |
  |     |     |          +-- co2flux_mocsy
  |     |     |
  |     |     +-- ptrc__predict_surf
  |     |          |
  |     |          +-- cbn__predict_surf
  |     |
  |     +-- tracer_ctl__predict
  |          |
  |          +-- tracer__internal_source
  |                |
  |                +-- bio__predict
  |
  +-- hist_ctl
        |
        +-- hist_ctl__write
             |
             +-- cbn__write_history
```

Chapter 20

Inert tracers

This chapter explains inert tracers implemented in MRI.COM. In general, to include an inert tracer in a model integration, you need to specify the model options and tracer attributes corresponding to that tracer. General explanation on how to specify attributes of a tracer is given at Section 13.3.

20.1 Ideal age tracer

Ideal age of a water mass is the time in year since it last contacted with the sea surface. This tracer is introduced by England (1995).

20.1.1 Source / Sink term

Source and sink of an age tracer a [year], is expressed as follows. In the oceanic interior,

$$\frac{\partial a}{\partial t} = -\mathcal{A}(a) + \mathcal{D}(a) + \frac{1}{\text{the total number of seconds of this year}} \quad \text{year/sec}, \tag{20.1}$$

where $\mathcal{A}()$ and $\mathcal{D}()$ represent advection and diffusion operator, respectively. Note that the r. h. s. depends on whether it is leap year or not. The tracer ages slowly in leap years than in normal years.

At the sea surface, the ideal age is set to zero:

$$a(k = 1) = 0. \tag{20.2}$$

20.1.2 Usage

Required Model option `IDEALAGE`
Required Add one to `numtrc_p`
Required Restart file for passive tracer (`nmlrs_ptrc`) (See Section 19.6)
Required Name for `nml_tracer_data` is 'Ideal Age Tracer' (See Section 13.3)
Optional Namelist `nml_tracer_idealage_start` to specify the base date and time [year, month, day, hour, minute, second] for the age of water.

20.2 CFCs

MRI.COM can treat CFCs (CFC11 and CFC12) following the protocols of OMIP (Orr et al., 2017). There is no source and sink for CFCs in the interior.

20.2.1 Surface boundary condition

The CFCs have air-sea gas fluxes at the sea surface as the source. Their dissolved concentrations $[A]$ [mol m^{-3}] evolve according to the advection ($\mathcal{A}()$) and diffusion ($\mathcal{D}()$) in the oceanic interior,

$$\frac{\partial [A]}{\partial t} = -\mathcal{A}([A]) + \mathcal{D}([A]) + J_g([A]). \tag{20.3}$$

$J_g()$ represents source/sink due to the air-sea gas fluxes $F([A])$ at the sea surface,

$$J_g([A]) = \frac{F([A])}{\Delta z_{\frac{1}{2}}}, \tag{20.4}$$

with

$$F(A) = k_w([A]_{sat} - [A]), \tag{20.5}$$

where k_w is its gas transfer velocity, $[A]_{sat}$ is the surface gas concentration in equilibrium with the atmosphere at an atmospheric pressure at the surface (P_a). k_w is a function of wind velocity, surface temperature, and atmospheric pressure at the sea surface. $[A]_{sat}$ is given by

$$[A]_{sat} = \phi_A^0 x_A, \tag{20.6}$$

where x_A is its mole fraction in dry air. The combined solubility term ϕ_A^0 is computed using the empirical fit of temperature and salinity. See Orr et al. (2017) for further details.

20.2.2 Usage

Required Model option `CFC`
Required Add two to `numtrc_p`
Required Restart file for passive tracer (`nmlrs_ptrc`) (See Section 19.6)
Required Names for `nml_tracer_data` are 'CFC11' and 'CFC12' (See Section 13.3)
Required Namelist `nml_force_data` to specify partial gas pressure (See Section 14.10)
  file_data File that contains mole fraction in dry air of CFC11 (and CFC12) [ppt]
  name CFC11 (and 12)
  txyu 't'

20.3 SF_6

The calculation of SF_6 also follows the protocols of OMIP (Orr et al., 2017). However, the formula for k_w in SF6 is not given in (Orr et al., 2017), so we use (Wanninkhof, 2014) as the latest formula to replace it.

The formulation of SF_6 is nearly the same as CFCs. Only the coefficients of the empirical fit differ.

20.3.1 Surface boundary condition

See Section 20.2.1, where CFCs should read SF_6.

20.3.2 Usage

Required Model option `SF6`
Required Add one to `numtrc_p`
Required Restart file (`nmlrs_ptrc`) (See Section 19.6)
Required Names for `nml_tracer_data` is 'SF6' (See Section 13.3)
Required Namelist `nml_force_data` to specify partial gas pressure (See Section 14.10)
  file_data File that contains atmospheric SF_6 [ppt]
  name SF6
  txyu 't'

Chapter 21

Atmospheric model

21.1 Coupling with an atmospheric model

MRI.COM can be coupled with an atmospheric model (e.g., Yukimoto et al., 2019). MRI.COM exchanges information necessary to drive this coupled model with the atmospheric model at specified time intervals via Scup (Yoshimura and Yukimoto, 2008) in the coupled mode. The ocean model receives all kinds of surface fluxes from the atmospheric model and the surface oceanic state such as sea surface temperature, surface oceanic current, and sea ice state, are transferred to the atmospheric model. The coupling interval is usually longer than the time step of the ocean model and the fluxes are kept fixed in the ocean model during the coupling cycle. See Section 17.11 for details of flux adjustment for conserving heat and fresh water fluxes by correcting errors caused by interpolation and sea ice evolution during the coupling loop.

Work flow is summarized as follows:

1. Receive atmosphere-ocean and atmosphere-sea ice fluxes for the current coupling cycle from the atmospheric model.
2. Adjust the fluxes to compensate for errors associated with the interpolation between the numerical grids of the atmospheric model and the coupled sea ice-ocean model.
   (a) Adjust the fluxes to compensate for errors associated with the sea-ice time integration within the current coupling cycle.
   (b) Time integration of the coupled sea ice-ocean model.
   (c) Go to the next step 3 if the current coupling cycle has finished. Otherwise, go back to the step 2a.
3. Send the surface ocean and sea ice states to the atmospheric model and go back to the step 1.

21.1.1 Serial and parallel executions

The surface fluxes at air-sea and air-ice interfaces are calculated in the atmosphere model and sent to the ocean model via coupler in the standard way for the coupled atmosphere-ocean model in the Meteorological Research Institute. The atmosphere model uses an initial snapshot of each coupling cycle of ocean state variables such as SST and sea ice concentration required to calculate surface fluxes. In order to conserve the heat and freshwater of the atmosphere-ocean system, the sea surface heat and freshwater flux imposed on the ocean model must be the time mean of an coupling cycle. We need to serially integrate the atmosphere model and then the ocean model if we want to impose the surface fluxes in the same coupling cycle on the ocean model.

Such a serial execution tends to waste computational resources because one model must wait for the other model to finish its time integration. Therefore, the surface fluxes averaged over the previous coupling cycle are usually imposed on the ocean model. By doing so, the information required for time integration of both atmosphere and ocean models is known at the beginning of each coupling cycle, enabling efficient parallel computing. Strict heat and freshwater conservations are not available because of the delay of surface fluxes imposed on the ocean model, but we can minimize its undesirable effects by using the adequately small coupling interval. Users can specify either the serial execution or parallel execution in the Scup runtime options. Refer to Yoshimura and Yukimoto (2008) for more details of Scup.

21.1.2 Time integration from an initial state

When a coupled model is integrated from an initial state, the parallel execution noted in the previous section is not available because the previous coupling cycle does not exist there. We recommend that users serially run the atmosphere and ocean models for the first coupling cycle and run the models in parallel for the subsequent cycles in this case. The users need to specify an Scup run-time option for this function and also need to set `flg_send_initial_to_agcm` in namelist `nml_cgcm_scup` `.true.` (Table 21.1). The second and subsequent runs can be executed in the parallel mode from the first coupling cycle because Scup restart files contain the temporal mean of surface fluxes in the last coupling cycle of each preceding run. Note that `flg_send_initial_to_agcm` must be `.false.` in these restart runs.

21.2 Usage

21.2.1 Compilation

You must specify model options `SCUP` and `SCUPCGCM` in `configure.in`.

21.2.2 Job parameters (namelist)

MRI.COM parameters associated with the atmosphere-ocean coupling are specified by namelist `nml_cgcm_scup`. The coupled atmosphere-ocean model also requires a namelist file `NAMELIST_SCUP` that specifies configurations of variables exchanged between the atmosphere model and MRI.COM.

Table21.1 Namelist `nml_cgcm_scup`.

variable name	units	description	usage
`intkt_loop`	hour (integer)	Time interval of a coupling cycle	default = 1
`flg_get_l2o`	logical	Receive heat and freshwater fluxes from river and glacier runoff models or not	default = `.true.`
`flg_carbon`	logical	Couple carbon cycle	default = `.true.`
`flg_send_initial_to_agcm`	logical	Serial execution in the first coupling cycle	default = `.false.`

Chapter 22

Nesting

In MRI.COM, embedding of a fine-resolution regional model within a coarse-resolution model can be realized by nesting method. The available methods range from a serial execution of a coarse-resolution (parent) model writing a boundary data and a fine-resolution (child) model reading it (off-line one-way nesting) to a parallel execution of both models exchanging data in both directions (on-line two-way nesting), as summarized in Table 22.1.

This chapter explains facilities available in MRI.COM for nesting, and how to construct and run a set of coarse- and fine-resolution models. Section 22.1 explains outline of the nesting function, Sec. 22.2 how to create sub-model grid and topography, and Sec. 22.3 communication between the models. Section 22.4 describes optional features for conservation of variables in nesting experiments, and Sec. 22.5 optional features for improving stability near the side boundaries. Finally, Sec. 22.6 explains how to set configuration options for users. Usage of nesting in the sea ice model and the tidal scheme is described in Section 17.12 and 18.4, respectively.

22.1 Outline

In a set of nested grid models, a fine-resolution (child) model is embedded in a coarse-resolution (parent) model. In one-way nesting, values at the side boundary of the child model are given by the parent model. The side boundary data may be transferred both off-line and on-line. In off-line mode, the data needed to calculate side boundary values are written to files by first running the parent model, and the child model is executed reading these data and calculating the side boundary values by interpolation. In on-line mode, parent and child models are run in parallel. Simple coupler (Scup), which is originally developed by Yoshimura and Yukimoto (2008) for the communication among components of the MRI Earth System Model, is used to transfer data.

In two-way nesting, in addition to the transfer of the side boundary data from parent to child model, the result of the child model is reflected to the parent model in the embedded region. Data are exchanged on-line and Scup is used for data transfer.

The boundary between parent and child models are formed by connecting either the tracer points or the velocity points of the Arakawa B-grid arrangement (Figure 22.1). Table 22.1 shows the side boundaries suited for the available nesting methods. From experiences, we recommend setting the tracer points as the side boundary for one-way nesting (imposing Dirichlet boundary condition for the sea surface height and tracer equations) and the velocity points as the side boundary for two-way nesting (imposing Neumann boundary condition for the sea surface height and tracer equations). In one-way nesting, the child model is quite stable when the side boundary is set to the tracer points. In two-way nesting, imposing the conservation on water volume and tracer content for the set of parent and child models is more straightforward when the side boundary is set to the velocity (flux) points.

To be able to treat both side boundary conditions with a single topographic configuration, the domain of a child model is constructed by taking tracer points as the boundary of the main region (Figure 22.1a). Inside the boundary, the same topography as the parent model should be given for at least one parent grid cell, as written in Sec.22.2. By doing so, we may use the velocity points located inside the boundary by a half parent grid width as the parent-child boundary for two-way nesting (Figure 22.1b). The model domain needs additional two rows or columns of velocity cells outside the

Table22.1 Nesting methods available in MRI.COM and the recommended side boundary of a child model in each method. It should be noted that the region and the topography of a child model must be made as if tracer-points are used as the side boundary.

model run	on-line (coupler)	off-line (file I/O)
one-way nest	tracer point	tracer point
two-way nest	velocity point	unavailable

main region along each side boundary (hereinafter referred to as halo; the region shaded by dark green in Figure 22.1a).

Whether the side boundary of the child model is the tracer points or the velocity points, three columns or rows of the parent model are sufficient for a child model to fill both halo and boundary regions. Here, the boundary region refers to a region inside the boundary of the main region by one grid of the parent model. For example, $T_1 - T_3$ and $U_1 - U_3$ in Figure 22.1 are sufficient for filling the southern boundary (except for one-to-one nesting). In off-line one-way nesting, the parent model outputs the three columns or rows to files, and the child model reads them. In on-line nesting, the mapping table between parent and child grids should be prepared for the Scup library. The Scup library exchanges only necessary data among MPI processes of different models.

In this way, prognostic variables at the side boundary of the child model are directly replaced by those from the parent model. This is usually called the clamped method (Cailleau et al., 2008). This method does not guarantee conservation of tracers in contrast to the method where fluxes are imposed at the boundary, but the integration is quite stable.

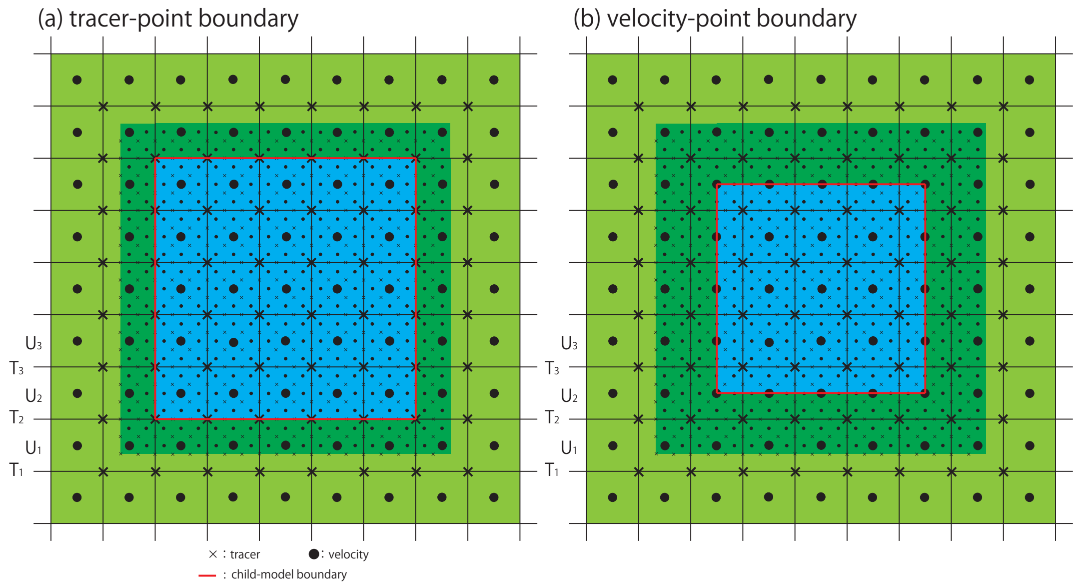

Figure22.1 Relation between parent (large symbols) and child (small symbols) grid points. The region shaded with blue (green) is the main region of the child (parent) model. The boundary of the main region is indicated by the red lines. In (a), the tracer points, and in (b), the velocity points, are set as the boundary between the two models. The domain of the child model should be constructed to include the dark green area, which is expanded outward by two velocity cells from the tracer point boundary (red line in (a)). This expanded region is used as halos for the main region. Variables in the halo region are obtained by using values of the parent model. Thus, for the southern boundary, three rows (T_1 - T_3 and U_1 - U_3) are sent to the child model from the parent model in both case (a) and (b). In two-way nesting, variables in the main region of the child model shown in (b) are sent to the parent model, except on the boundary line (red line).

Multistage nesting is available in MRI.COM, and on-line nesting and off-line nesting can be used together. This flexibility makes various kinds of experiments possible, e.g., a double on-line nesting experiment using global, North-Pacific, and Japanese coastal models, or an off-line one-way nesting experiment of a Japanese coastal model reading the side boundary values from an on-line nesting experiment for a set of global and North-Pacific models.

22.2 Grid and Topography

22.2.1 Parent (coarse-resolution) model

There is no particular issue in constructing the parent model. Because the grid size ratio between the child and parent models must be odd, the grid size of the child model may be taken into account in determining the grid size of the parent model. A grid size ratio of 1:3 or 1:5 is recommended.

22.2.2 Child (fine-resolution) model

a. Horizontal grid size

The domain of a child model is constructed by taking the tracer point as the boundary of the main region (the red line in Figure 22.1a) and adding the halo region (the dark green region), even if the velocity-point boundary will be used. Determining the grid arrangement of the child model and the corresponding parent model region (T_1 , U_1 in Figure 22.1a) is complicated, depending on presence or absence of the side boundaries and the ratio of grid division, as seen below. Users are advised to use the MXE package tool (`prep/nest/topo`, `prep/offnestsub`) to determine the grid size and other settings. With this tool, users can not only divide the parent model grid by a constant ratio, but also change the division ratio depending on the location.

Details of the horizontal grid division are as follows.

- the main internal region
  The distance between the velocity points of the parent model (i.e., T-cell) is divided equally in the main region of the child model (the blue region in Figure 22.1a). Thus, the T-cell arrangement is constructed, and then the velocity cell, which represents the model topography, is created. (Only when the parent model grid width is constant and the division ratio is constant, it can be considered that the velocity cell of the parent model is simply divided into equal parts.)
- the western and southern halo region
  There should be always two marginal velocity cells outside the main region, as shown by the dark green region in Figure 22.1a. The grid range of the parent model used to interpolate the values there depends on the division ratio.
- the eastern and northern halo region
  There should be two marginal velocity cells. However, if the boundary is filled by land and does not receive boundary data, one marginal velocity cell will be enough. The corresponding grid range of the parent model depends on the division ratio.

Our nesting scheme does not support division of the vertical layers. The same vertical grid intervals as the parent model must be used. The number of the model layers can be reduced, if the deepest topography of the child model is shallower than that of the parent model. (If a user prepares boundary data interpolated vertically, it is possible to run the child model with different vertical grid intervals by using the off-line nesting option.)

b. Topography of the child model

To achieve flux conservation, the child model should have the same topography as the parent model around the side boundary of the child model. It is recommended that the topography of the child model should have the same topography as the parent model for two parent velocity cells inward and one parent velocity cell outward relative to the boundary of the main region constructed by connecting tracer points (Figure 22.2).

c. Weighting ratio between parent and child models

For smoothness, prognostic values of the child model around the side boundary may be given as a weighted average of the solutions of the parent and child models. For variable ϕ at a grid point with index (i, j),

$$\phi_{ij}^{child} = \alpha_{ij}\phi_{ij}^{p} + (1 - \alpha_{ij})\phi_{ij}^{c}, \tag{22.1}$$

where ϕ^{child} is the value for the child model, ϕ^{p} is the parent model solution interpolated on the child grid, ϕ^{c} is the child model solution, and α is a spatially dependent weight between parent and child solutions.

Although this weight is introduced for a smooth transition from parent to child model solution, we recommend that the weight for the parent model to be unity in the halo and at the boundary of the main region and zero elsewhere. Instead, this weight is used to specify the boundary between the main regions of parent and child models. When the boundary is at the tracer points, the weight for the parent solution should be unity in the dark green region of Figure 22.1a. When the boundary is at the velocity points, the weight for the parent solution should be unity in the dark green region of Figure 22.1b. Weight must be prepared separately for the tracer and velocity points. (Incidentally, a radiation boundary condition may be imposed by elaborating weight values.)

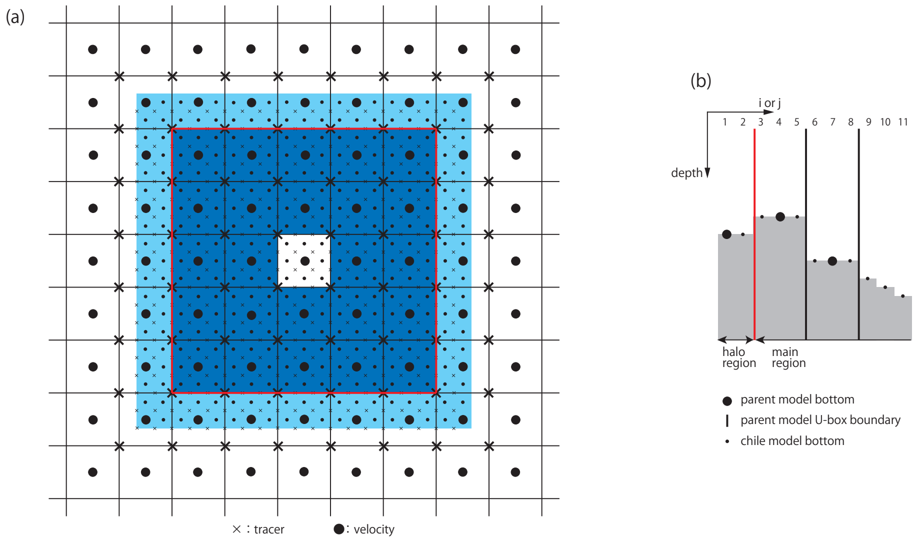

Figure22.2 (a) Condition on the topography of the child model around the side boundary. The topography of the child model should have the same topography as the parent model for two parent velocity cells inward (dark blue) and one parent velocity cell outward (light blue) relative to the boundary of the main region constructed by connecting tracer points (red rectangle). (b) Bathymetry near the boundary of the child model, when the grids of the parent model are divided by a factor of 3.

22.3 Intermodel transfer

Communication between models is both downscaling and upscaling in two-way nesting, whereas only downscaling in one-way nesting.

22.3.1 Downscaling

In the downscaling from parent to child model, the method of data transfer and interpolation is different between tracer point and velocity point.

a. Tracer points

It is straightforward to do a linear interpolation for variables on tracer points. Figure 22.3 shows the relative positions of tracer points of parent and child models. Because topography is common between parent and child models around the side boundary, all child tracer points in an ocean velocity cell of the parent model (bounded by the red rectangle) are ocean points. Values on these child tracer points are computed by using values at the four corner tracer points of the ocean velocity cell of the parent model (⊚ in Figure 22.3). All of the four corner points are always available for bilinear interpolation to compute the tracer values of the child model. Thus, assuming that parent and child grid points coincide at (i_p, j_p) and (i_c, j_c), the tracer value at $(i_c + m, j_c + n)$, where $(0 \leq m \leq M)$ and $(0 \leq n \leq N)$, and M and N are the zonal and meridional nesting ratios, is computed as follows:

$$\phi_{i_c+m,j_c+n} = (1-w_x)(1-w_y)\phi^p_{i_p,j_p} + w_x(1-w_y)\phi^p_{i_p+1,j_p} + (1-w_x)w_y\phi^p_{i_p,j_p+1} + w_x w_y \phi^p_{i_p+1,j_p+1}, \tag{22.2}$$

where $w_x = (x^t_{i_c+m} - x^t_{i_p})/(x^t_{i_p+1} - x^t_{i_p})$ and $w_y = (y^t_{j_c+n} - y^t_{j_p})/(y^t_{j_p+1} - x^t_{j_p})$, x^t and y^t represent the tracer point in zonal and meridional coordinates, respectively.

b. Velocity points

Figure 22.4 shows the relative positions of velocity points of parent and child models. Unlike the tracer points, all of the four parent velocity points surrounding a child velocity point are not necessarily ocean points. When all parent grid points are ocean, bi-linear interpolation can be applied.

A problem arises for velocity points of the child model between an ocean point and a land point of the parent model (shaded by blue in Figure 22.5a), because only data at a single point is available for interpolation. The same problem also

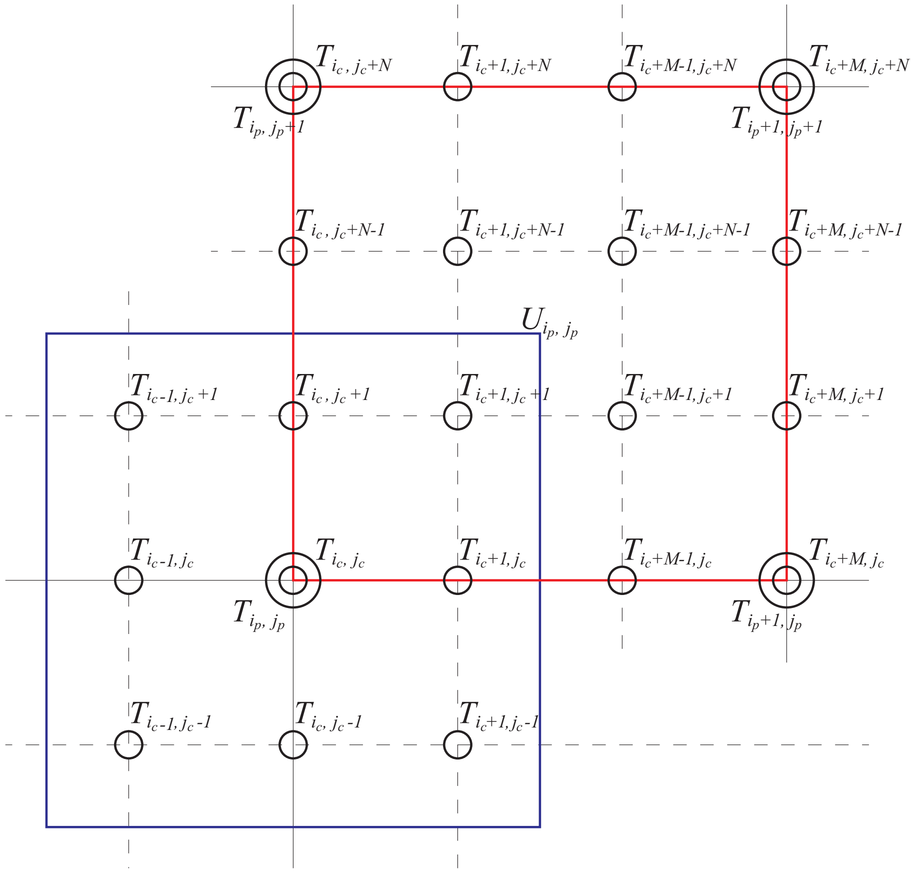

Figure22.3 Relative positions of tracer points of parent and child models (◎ and ∘, respectively), coinciding at (i_p, j_p) and (i_c, j_c). The red and blue frames indicate U- and T-boxes in the parent model, respectively. M and N are the zonal and meridional nesting ratios.

arises for a velocity cell that has a partial velocity cell at its neighbor. The velocity for the partially blocked part (shaded by purple in Figure 22.5a) should be treated separately from the upper unblocked part.

In this case, the child model velocity perpendicular to the coast line (u^c) is assumed to be zero at the coast and linearly interpolated from the nearest ocean grid point of the parent model (Figure 22.5b). The component tangent to the coast line (v^c) is set to the same value at the nearest ocean grid point of the parent model (Figure 22.5c). By doing so, the interpolated velocity field yields nearly the same transport along the coast line as the parent model. To be specific, when the parent model has a partially shaved bottom cell at (i_p-1, j_p, k), velocities of the child model between $x^u(i_p-1) \le x^u(i) \le x^u(i_p)$ are computed as follows:

$$u^c(i) = w_x u^p_{i_p} + (1-w_x) u^p_{i_p-1}, \tag{22.3}$$

$$v^c(i) = w_x v^p_{i_p} + (1-w_x) v^p_{i_p-1}, \qquad \text{if} \quad x^u(i) < x^t(i_p) \tag{22.4}$$

$$u^c(i) = \frac{\Delta z_{i_p-1,k}}{\Delta z_{i_p,k}} (w_x u^p_{i_p} + (1-w_x) u^p_{i_p-1}) + \frac{\Delta z_{i_p,k} - \Delta z_{i_p-1,k}}{\Delta z_{i_p,k}} w u^p_{i_p}, \tag{22.5}$$

$$v^c(i) = \frac{\Delta z_{i_p-1,k}}{\Delta z_{i_p,k}} (w_x v^p_{i_p} + (1-w_x) v^p_{i_p-1}) + \frac{\Delta z_{i_p,k} - \Delta z_{i_p-1,k}}{\Delta z_{i_p,k}} w v^p_{i_p}, \qquad \text{if} \quad x^t(i_p) < x^u(i) \tag{22.6}$$

where $w_x = (x^u(i) - x^u(i_p-1))/(x^u(i_p) - x^u(i_p-1))$ and $w = (x^u(i) - x^t(i_p))/(x^u(i_p) - x^t(i_p))$.

To apply this method to the barotropic transport, the barotropic transport is first divided by the height of the water column to obtain the vertically uniform barotropic velocity. This velocity is given to all the velocity points in the water column and then the same method as used for the 3-D velocity field is applied.

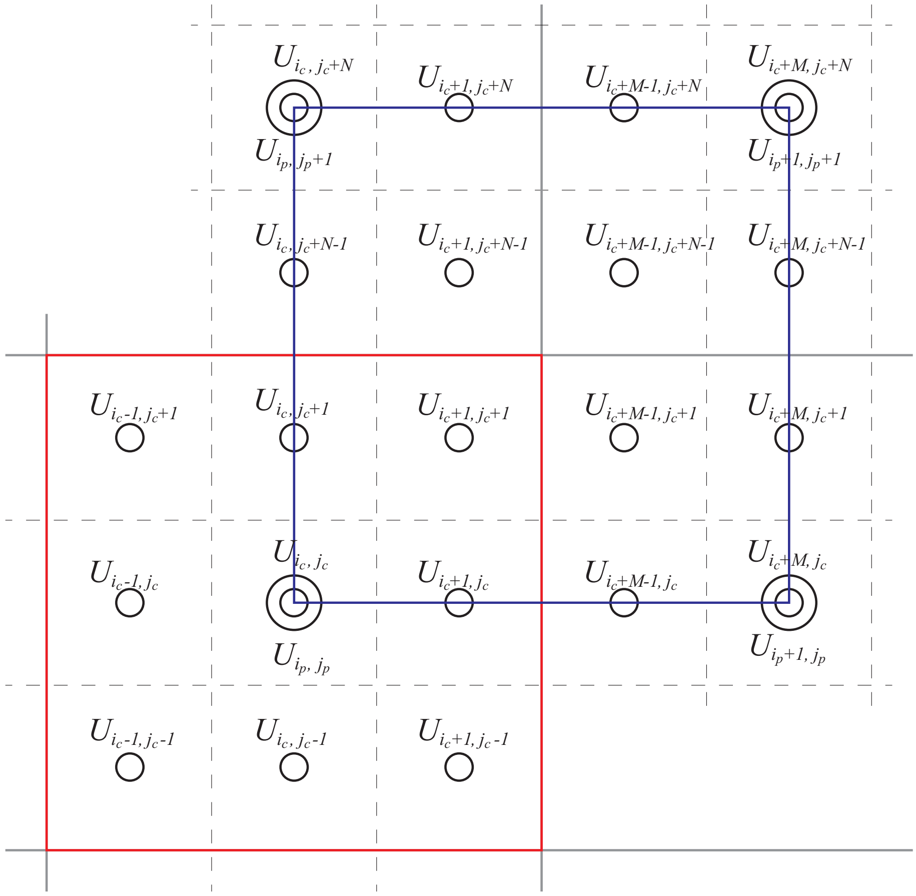

Figure22.4 Same as Figure 22.3 except for velocity point.

22.3.2 Upscaling

In two-way nesting, values of the parent model in the child model's main region are filled with those based on the child model. The region to receive feedback in the parent model is the main region of the child model excluding the boundary line (inside the red line of Figure 22.1b). MRI.COM has two options for upscaling, replacement and nudging.

a. Replacement

One option is to simply replace variables of the parent model by those of the child model. Ideally, fine scale features that cannot be resolved by the parent model should be removed in upscaling. In MRI.COM, simple averaging within a parent grid is used (averaging over blue rectangle in Figures 22.3 and 22.4), though more sophisticated filters have been proposed (e.g., Debreu et al., 2012). Because applying a full averaging for the entire embedded region is costly in terms of data transfer between models, a full averaging is only applied to the 2 to 3 grid points inside the parent-child boundary,

$$\phi^p_{i_p,j_p} = \frac{1}{M \times N} \sum_{i,j} \phi^c_{i,j}, \quad i_c - i_m \le i \le i_c + i_m \quad \text{and} \quad j_c - j_n \le j \le j_c + j_n, \tag{22.7}$$

where $i_m = (M-1)/2$ and $j_n = (N-1)/2$. For the region further inside the boundary, values on the child grid points that coincide with the parent grid points may be used without filtering,

$$\phi^p_{i_p,j_p} = \phi^c_{i_c,j_c}, \quad x^t(i_p) = x^t(i_c) \quad \text{and} \quad y^t(j_p) = y^t(j_c). \tag{22.8}$$

This treatment does not cause a problem because this inner region away from the boundary does not affect the temporal evolution of the physical field in the main region of the parent model. (There is an option to omit communication in this inner region for speeding up, `flg_feedback_repl_bnd`.)

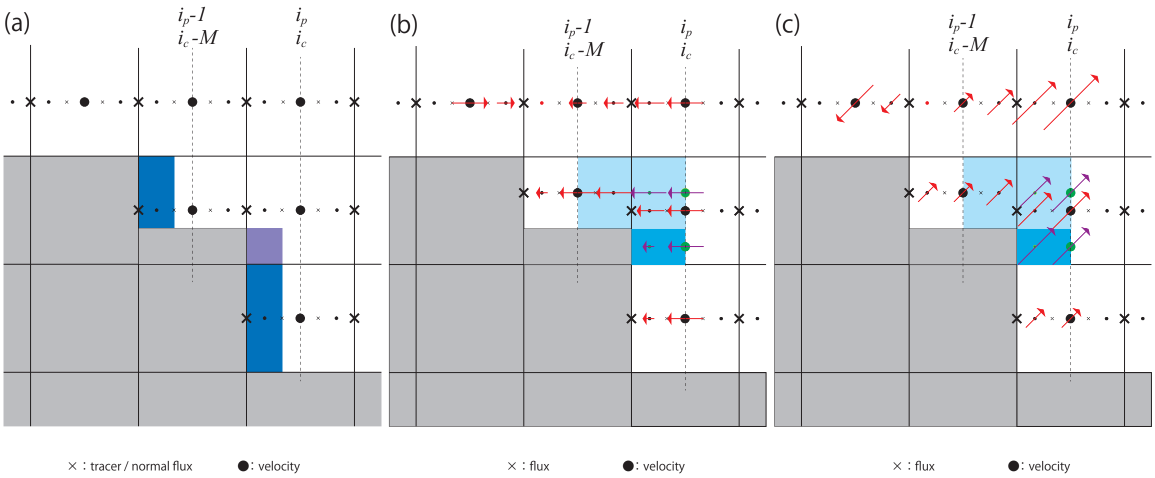

Figure22.5 Interpolation of the parent model velocity to the child grid with bottom topography. (a) For velocity points of the child model between an ocean point and a land point of the parent model (shaded with blue or purple), only one data is available for interpolation. Examples of (b) normal and (c) tangential component of velocity relative to the coast line (red arrows). For an ocean velocity cell next to a partial cell, the cell is divided into a part blocked by a wall and a part unblocked (green point). Linear interpolation is used for an unblocked part and special treatment is applied to the blocked part (purple arrows). The velocity is obtained as a weighted average of the two parts.

b. Nudging

Nudging is another option to reflect the child result to the parent model. In MRI.COM, tracer values of the child model may be used as reference values (ϕ_C^p) to which parent tracer fields (ϕ^p) are nudged,

$$\frac{\partial \phi^p}{\partial t} = -\frac{1}{\gamma}(\phi^p - \phi_C^p), \tag{22.9}$$

where γ is a restoring time scale usually taken to be very small, about 0.1 day. Only tracer fields are corrected in the parent model. Neither velocity nor sea surface height is modified. Our assumption lying behind this treatment is that reflecting only slowly evolving quasi-geostrophic baroclinic fields is enough for a parent model in many applications.

22.4 Conservation

Conservation of sea water volume and scalar quantities are generally preferable. This is especially so for the case of climate applications with long-term integration. To achieve this in nesting, MRI.COM has options to adjust side boundary and surface fluxes in the child model, or source / sink due to nudging in the parent model.

22.4.1 Side boundaries

Adjustment of the flux at the side boundary is performed as follows. Consider an evolution equation for a quantity ϕ in a simple flux form:

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot \mathbf{f}, \tag{22.10}$$

where $\mathbf{f} = (f_x, f_y)$, which are due to either advective or diffusive fluxes. Integration (22.10) over each model's main region (Figure 22.6) gives

$$\frac{\partial \Phi^p_{\Omega-\omega}}{\partial t} \equiv \int_{\Omega-\omega} \frac{\partial \phi}{\partial t} dS = -\int_{\partial\Omega} \mathbf{f}^p \cdot \mathbf{n} dl \tag{22.11}$$

for parent model's main region and

$$\frac{\partial \Phi^c_{\omega}}{\partial t} \equiv \int_{\omega} \frac{\partial \phi}{\partial t} dS = -\int_{\partial\omega} \mathbf{f}^c \cdot \mathbf{n} dl \tag{22.12}$$

for child model's main region. If integral values of the flux normal to the boundary (that is, r.h.s.) cancel between parent and child models, the sum of integral values of the tendency of ϕ in each model's main region is conserved,

$$\frac{\partial \Phi^p_{\Omega-\omega}}{\partial t} + \frac{\partial \Phi^c_{\omega}}{\partial t} = 0. \tag{22.13}$$

However, this conservation does not always hold owing to discretization. Some adjustment is needed.

In MRI.COM, conservation of water volume and tracers is achieved by adjusting boundary fluxes computed in each model. Different approaches are taken for advective and diffusive fluxes.

For advective fluxes, fluxes of the child model are adjusted to match those of the parent model. Fluxes for tracer cells of the child model (f^c_y) are area-integrated along the side boundary section over the width of a tracer cell of the parent model (Figure 22.7a),

$$F^c_y(i_p) = \sum_i f^c_y(i) \times \Delta s^c(i), \quad i_c - i_m \le i \le i_c + i_m, \tag{22.14}$$

where Δs^c is the area of the side boundary of a tracer cell of the child model and $i_m = (M-1)/2$. This is compared with the flux for the tracer cell of the parent model (f^p_y) (Figure 22.7b),

$$F^p_y(i_p) = f^p_y(i_p) \times \Delta s^p(i_p), \tag{22.15}$$

where Δs^p is the area of the side boundary of a tracer cell of the parent model. On the basis of this comparison, fluxes of the child model are adjusted using a quantity $f^c_{y\ \mathrm{adj}}(i_p)$ given by

$$f^c_{y\ \mathrm{adj}}(i_p) = (F^p_y(i_p) - F^c_y(i_p))/\Delta S^c(i_p), \tag{22.16}$$

where

$$\Delta S^c(i_p) = \sum_i \Delta s^c(i), \quad i_c - i_m \le i \le i_c + i_m. \tag{22.17}$$

The adjusted flux $f^c_{y\ \mathrm{mod}}$ is given by

$$f^c_{y\ \mathrm{mod}}(i) = f^c_y(i) + f^c_{y\ \mathrm{adj}}(i_p), \quad i_c - i_m \le i \le i_c + i_m, \tag{22.18}$$

so that the flux integral in the child model equals the parent model flux for parent's tracer cell.

For diffusive fluxes, the sum of fluxes in the child model (f^c_x) may be given to the parent model instead of adjusting the child model flux to the parent model flux as in the advective fluxes (Figure 22.8),

$$F^p_{x\ \mathrm{mod}}(j_p) = f^p_{x\ \mathrm{mod}}(j_p)\Delta s^p(j_p) = \sum_j f^c_x(j) \times \Delta s^c(j), \quad j_c - j_n \le j \le j_c + j_n. \tag{22.19}$$

$F^p_{x\ \mathrm{mod}}(j_p)$ replaces the flux calculated on the parent grid $F^p_x(j_p)$ at the boundary.

The flux adjustment should be applied to any horizontal flux of volume and tracers. Currently, this is implemented in the following fluxes:

- Volume flux in the 3-D continuity equation
- Advective volume flux in the 2-D continuity equation (both barotropic and baroclinic time step)
- Diffusive volume flux in the 2-D continuity equation (both barotropic and baroclinic time step)
- Tracer advection flux by Second Order Moment and MPDATA schemes
- Tracer diffusion flux by horizontal diffusion
- Tracer flux by isoneutral transport scheme (isoneutral diffusion plus Gent-McWilliams parameterization)
- Sea ice advection plus diffusion flux (area fraction, volume, thermal energy) with some special treatment. See Chapter 17 for details.

22.4.2 Surface fluxes

In order to impose conservation on some properties in a set of nested models, surface fluxes must be also taken into account.

In MRI.COM, you may impose conservation condition for the following cases by adjusting the net surface flux.

- Volume of sea water in an ocean-only mode (`WADJ` option).

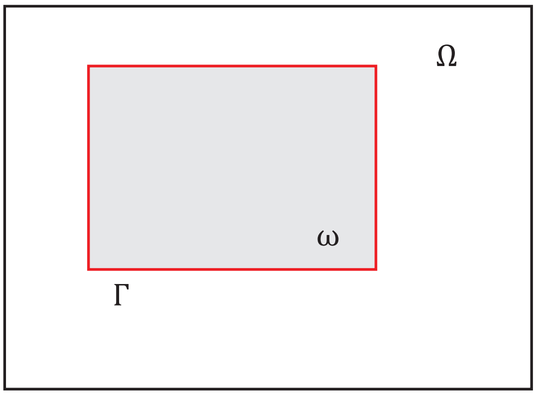

Figure22.6 Schematic for considering conservation in a system of coarse and fine grid models. The coarse grid (parent) model region is denoted by Ω and the fine grid (child) model region is denoted by ω. The boundary between models is denoted by Γ. These notations follow Figure 1 of Debreu et al. (2012).

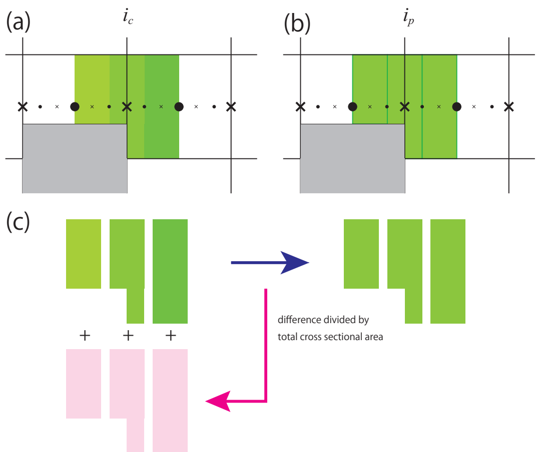

Figure22.7 Schematic explaining flux adjustment for a child model. In this zonal-vertical section, tracer points coincide at i_p and i_c. (a) Meridional fluxes are computed in the child model using the values interpolated from the parent model. Flux is integrated over the corresponding parent grid and this is compared with that of the parent model (b). (c) Difference between the integrated fluxes is divided by the integration area (unit parent grid area) to give flux correction for the child model (pink).

- Salt content in the ocean - sea ice system in any mode (`l_sss_rst_cnsv = .true.` in namelist `nml_sss_restore`)
- Heat content and volume of sea water in a coupled mode (`SCUPCGCM` and `NPUTFLUX` options for parent and `NGETFLUX` option for child)

For example, to conserve sea water volume in the system (`WADJ` option), a globally constant adjustment factor $f_{w\,\mathrm{adj}}$ is used so that

$$\int_{\Omega-\omega} (f_w^p + f_{w\,\mathrm{adj}})dS + \int_{\omega} (f_w^c + f_{w\,\mathrm{adj}})dS = 0, \tag{22.20}$$

where f_w^p and f_w^c represent total surface water fluxes of parent and child models, respectively. The adjustment factor is given by

$$f_{w\,\mathrm{adj}} = -(\int_{\Omega-\omega} f_w^p dS + \int_{\omega} f_w^c dS)/(S_{\Omega-\omega} + S_\omega), \tag{22.21}$$

where $S_{\Omega-\omega} = \int_{\Omega-\omega} dS$ and $S_\omega = \int_\omega dS$.

To avoid a drift of total salt content of the system (`l_sss_rst_cnsv = .true.` in namelist `nml_sss_restore`) in the case of surface salinity restoring, a globally constant adjustment factor $f_{s\,\mathrm{adj}}$ is used so that

$$\int_{\Omega-\omega} (f_s^p + f_{s\,\mathrm{adj}})dS + \int_{\omega} (f_s^c + f_{s\,\mathrm{adj}})dS = 0, \tag{22.22}$$

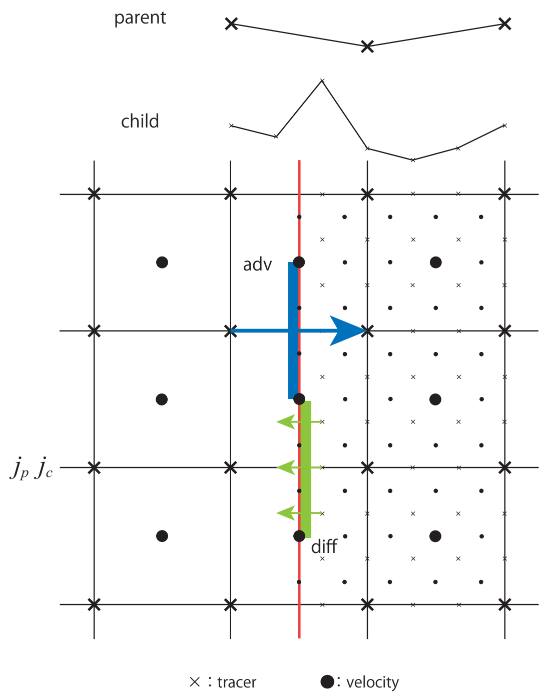

Figure22.8 Computation of parent model's zonal diffusive flux across the boundary, coinciding at j_p and j_c in terms of the tracer point. In this case, the sum of diffusive fluxes of the child model (green) is used as the diffusive flux for the parent model at j_p. This is contrasted with the advective flux (blue) where the parent model flux is used to adjust the child model flux.

where f_s^p and f_s^c represent surface salinity flux due to restoring in parent and child models, respectively. The adjustment factor is given by

$$f_{s\,\mathrm{adj}} = -\Big(\int_{\Omega-\omega} f_s^p\, dS + \int_\omega f_s^c\, dS\Big)/(S_{\Omega-\omega} + S_\omega). \tag{22.23}$$

When a set of nested grid models is coupled with an atmospheric model (`SCUPCGCM` and `NPUTFLUX` options for parent and `NGETFLUX` option for child), the sum of sea surface heat flux of the ocean model must equal that of the atmospheric model. To achieve this, a globally constant adjustment factor $f_{q\,\mathrm{adj}}$ is used so that

$$\int_{\Omega-\omega} (f_q^p + f_{q\,\mathrm{adj}})dS + \int_\omega (f_q^c + f_{q\,\mathrm{adj}})dS = \int_\Omega f_q^a\, dS, \tag{22.24}$$

where f_q^p, f_q^c, and f_q^a represent net surface heat flux in parent oceanic, child oceanic, and atmospheric models, respectively. The adjustment factor is given by

$$f_{q\,\mathrm{adj}} = \Big[\int_\Omega f_q^a\, dS - \Big(\int_{\Omega-\omega} f_q^p\, dS + \int_\omega f_q^c\, dS\Big)\Big]/(S_{\Omega-\omega} + S_\omega). \tag{22.25}$$

$f_{q\,\mathrm{adj}}$ is used to offset the net longwave flux in the ocean models.

These adjustments are applicable only to on-line two-way nesting by using pre-communicator of Scup.

22.4.3 Nudging

To conserve the total amount of tracer of the parent model when a tracer of the parent model is nudged toward that of the child model, an additional flux adjustment is needed. A globally constant adjustment factor S_{adj}^ϕ is added to the right hand side of (22.9) so that

$$\int_\omega [-\frac{1}{\gamma}(\phi^p - \phi_C^p) + S_{\mathrm{adj}}^\phi]dV = 0. \tag{22.26}$$

The adjustment factor is given by

$$S_{\mathrm{adj}}^\phi = \int_\omega \frac{1}{\gamma}(\phi^p - \phi_C^p)dV/V_\omega, \tag{22.27}$$

where $V_\omega = \int_\omega dV$. By doing so, nudging only changes the distribution of the tracer in the parent model without affecting its total amount. MRI.COM offers this adjustment as an option (`flg_feedback_cnsv_ndg` = `.true.` in namelist `nml_nest_par`).

22.5 Stabilization of parent-child boundaries

Appearance of noises and discontinuity around the parent-child boundary is almost unavoidable. It is a common practice to set a "sponge" region around the boundary to reduce them. (The word "sponge" is used because it works to relieve shock.) Several methods are available for MRI.COM.

22.5.1 Giving 2-D distribution of diffusion and viscosity coefficient

A simple choice would be to give 2-D distribution of time-invariant coefficient for diffusion and viscosity operators that have been selected for each model. For example, when a common operator is selected for both parent and child models, the coefficients of the child model should be the same as those of the parent model around the side boundary to obtain similar results there. The coefficients in the boundary region should be connected smoothly with the small coefficients in the interior.

Among the mixing operators available in MRI.COM, the additional harmonic viscosity is most effective for getting a smooth solution around the parent/child boundary. However, this scheme should be used with caution because the harmonic viscosity may even damp resolved processes of the parent model. These additional diffusivity and viscosity are available, whether nesting is one-way or two-way.

22.5.2 Scheme specialized for two-way nesting

MRI.COM has some schemes specialized for two-way nesting in order to make the model stable.

a. Parent model

A parent model can use an additional Laplacian operator for both velocity and tracer. This operator may be applied to the velocity points on the boundary (red line in Figure 22.9) and to the tracer point (blue line) just outside the feedback region where parent-model values are upscaled (replaced) by child-model values (light blue region). Laplacian is calculated in the direction perpendicular to the boundary.

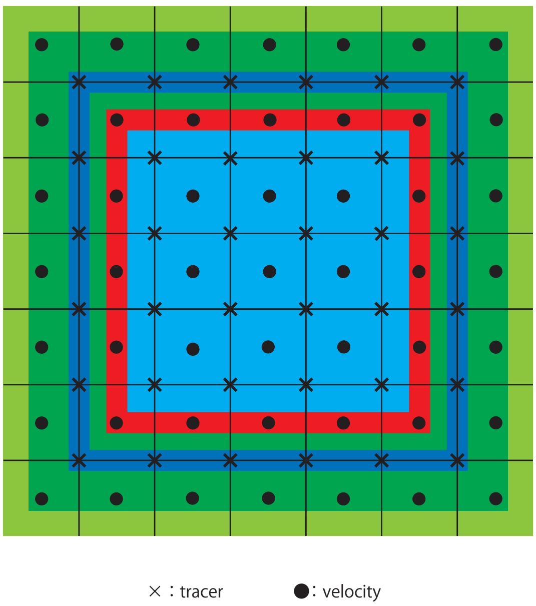

Figure22.9 Same as Figure 22.1b except for indicating of sponge regions for a parent model. Laplacian diffusion and viscosity operators are applied in the blue and red bands, respectively.

b. Child model

For a child model, Laplacian operator is applied to the difference between the child value and the parent value mapped on the child grid in the sponge region following Section 2.5 of Debreu et al. (2012),

$$\frac{\partial \phi^c}{\partial t} = \nabla \cdot \kappa \nabla (\phi^c - \phi_P^c), \tag{22.28}$$

where ϕ^c is the child model value, ϕ_P^c is the parent value mapped on the child grid, and κ is viscosity/diffusion coefficient. The operator is prepared for velocity and tracers. The width of the sponge region is two parent velocity cells from the boundary constructed by connecting tracer cells (region shaded with purple in Figure 22.10).

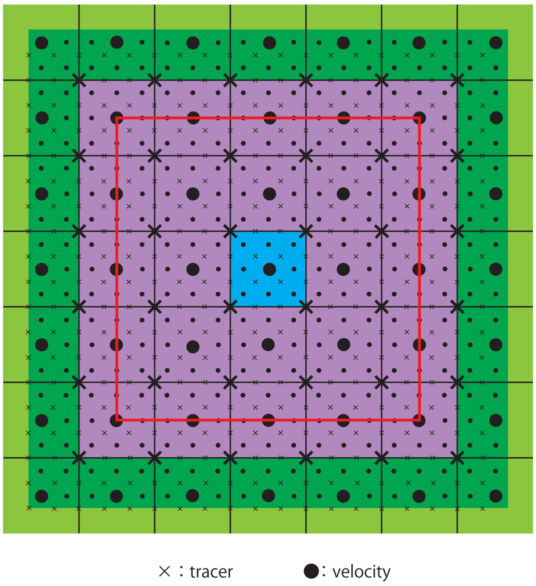

Figure22.10 Same as Figure 22.1b except for indicating of sponge region for child model. Laplacian diffusion/viscosity operator is applied to the difference between child and parent values in the region shaded with purple.

22.6 Usage

22.6.1 Offline one-way nesting (OFFNESTPAR, OFFNESTSUB option)

a. Compilation

The model option for the parent model is `OFFNESTPAR` (OFFline-NESTing PARent), and that for the child model is `OFFNESTSUB`. These model options and the model name should be specified in `configure.in`.

b. Runtime specification for the parent model

At run time, namelists `nml_parentmodel_grid` (Table 22.2) and `nml_parentmodel_package` (Table 22.3) must be specified. They are read by `Parent/parent_common.F90` on execution. Users should set the namelist based on the MXE tool (`prep/offnestsub`). See `docs/README_Namelist.md` for details.

Following files are output from the parent model to be read by the child model.

- Three rows or columns of data for each side boundary of specified packages. For package (`package_name`), ’uv’, ’barotropic’, ’active_tracer’, ’passive_tracer’, ’vmixcoef’, ’nohkim’, ’my25’, ’gls’, ’ice_thermodynamics’, ’ice_dynamics’, and ’tide’ are available.
- Land-sea index of the parent model (`file_parent_sea_index_out`).
- Latitude and longitude of the parent model (`file_parent_grid_out`).

The daily output of side boundary data will work in running the child model.

Table22.2 Namelist nml_parentmodel_grid for the off-line parent model

variable name	description	usage
sub_region_ifirst	the western end point in the parent model to be used to interpolate data (T-point)	
sub_region_ilast	the eastern end point in the parent model to be used to interpolate data (T-point)	
sub_region_jfirst	the southern end point in the parent model to be used to interpolate data (T-point)	
sub_region_jlast	the northern end point in the parent model to be used to interpolate data (T-point)	
dx_ratio_w	refinement ratio at west boundary	odd number, zero for no nesting along this side
dx_ratio_e	refinement ratio at east boundary	odd number, zero for no nesting along this side
dy_ratio_s	refinement ratio at south boundary	odd number, zero for no nesting along this side
dy_ratio_n	refinement ratio at north boundary	odd number, zero for no nesting along this side
file_parent_sea_index_out	land-sea index of the parent model	
file_parent_grid_out	latitude-longitude of the parent model	

Table22.3 Namelist nml_parentmodel_package for the off-line parent model

variable name	description	usage
package_name		
file_root_name	root name of the output boundary data files	
bndr_dt_sec	time interval [sec] of output	must be multiple of model time step
nrec_first	the first record number of the output files	= 1 : create new file > 1 : append to old file
l_write_initial_state		.true. : output initial state .false. : do not output initial state

c. Runtime specification for the child model

At run time, namelists nml_submodel_grid (Table 22.4), nml_submodel_bnd_cnd (Table 22.5), and nml_submodel_package (Table 22.6) must be specified. They are read by Sub/sub_common.F90 on execution. See docs/README_Namelist.md for details.

To run the child model, prepare the following two data files:

- file_sub_wgt_t: The contribution ratio between parent and child models around the boundary (T-point),
- file_sub_wgt_u: The contribution ratio between parent and child models around the boundary (U-point).

In addition, produce the following files by running the parent model:

- file_root_name in nml_submodel_package: Three rows or columns of the necessary elements at each side boundary,
- file_parent_sea_index: Land-sea index of the parent model,
- file_parent_grid: Latitude and longitude of the parent model.

Note the following when preparing barotropic boundary data. Since time filtering is used to feedback the result of the barotropic equations to the baroclinic mode, the barotropic equations are integrated over the end of the baroclinic time. Thus, the child model needs barotropic data longer than the original output from the parent model. To fulfill this need, prepare the barotropic data file to include one additional time record. For example, in a historical (yearly) run, the second record of the following year of the parent model should be appended to the present year data. If the additional record is not prepared, MRI.COM integrates the barotropic equation by repeating the final record just as needed.

Table22.4 Namelist `nml_submodel_grid` for the off-line child model

variable name	description	usage
`parent_grid_im`	Total number of grid points in X direction of the parent model	
`parent_grid_jm`	Total number of grid points in Y direction of the parent model	
`sub_region_ifirst`	the western end point in the parent model to be used to interpolate data (T-point)	
`sub_region_ilast`	the eastern end point in the parent model to be used to interpolate data (T-point)	
`sub_region_jfirst`	the southern end point in the parent model to be used to interpolate data (T-point)	
`sub_region_jlast`	the northern end point in the parent model to be used to interpolate data (T-point)	
`dx_ratio_w`	refinement ratio at west boundary	odd number, zero for no nesting along this side
`dx_ratio_e`	refinement ratio at east boundary	odd number, zero for no nesting along this side
`dy_ratio_s`	refinement ratio at south boundary	odd number, zero for no nesting along this side
`dy_ratio_n`	refinement ratio at north boundary	odd number, zero for no nesting along this side
`file_parent_grid`	latitude-longitude of the parent model	
`file_parent_sea_index`	land-sea index of the parent model	

Table22.5 Namelist `nml_submodel_bnd_cnd` for the off-line child model

variable name	description	usage
`file_sub_wgt_t`	filename of weighing factor for parent/child model around the boundary (T-points)	
`file_sub_wgt_u`	filename of weighing factor for parent/child model around the boundary (U-points)	

Table22.6 Namelist `nml_submodel_package` for the off-line child model

variable name	description	usage
`package_name`		
`file_root_name`	root name of the output boundary data files	
`bndr_dt_sec`	time interval [sec] of the input data	
`bndr_first_date`	integer array(6) indicating date and time (YMDMHS) of the first record	
`num_bndr_record`	the last record number of the input files	

22.6.2 On-line nesting (SCUPNEST option)

a. Compilation

For on-line mode, `SCUPNEST` option should be specified. Also, the name of the model should be explicitly specified such as `NAME_MODEL` = *modelname*. The model option for the parent model is `PARENT`, and that for the child model is `SUB`.

In ocean-only mode, surface fluxes are calculated in each model, preferably on the basis of common atmospheric state.

When the parent model is coupled with an atmospheric model (`SCUPCGCM`), the parent model receives surface fluxes from the atmospheric model. The surface flux may be sent from parent to child model. In the parent (child) model, `NPUTFLUX` (`NGETFLUX`) option must be selected.

These model options and the model name should be specified in `configure.in`.

b. Runtime specification: NAMELIST_SCUP

In on-line mode, parent and child models are run at the same time and Scup (simple coupler) by Yoshimura and Yukimoto (2008) is used to exchange data. User should tell the coupler how data are exchanged between parent and child models via a namelist file NAMELIST_SCUP. A template of NAMELIST_SCUP is available from MRI.COM execution environment (MXE). Mapping tables for data transfer used by Scup can be created in MXE. User may comment out unnecessary data exchange according to the model and runtime options chosen for a specific experiment. An example of NAMELIST_SCUP is listed in the following. In this example, the model name of the parent model is GLOBAL and that of the child model is NP01.

An example of NAMELIST_SCUP

```
# pre-communicator
#
# (parent -> child)
 &nam_scup_pre model_put='GLOBAL', model_get='NP01', type='REAL8' /
 &nam_scup_pre var_put='ALONTC',  var_get='ALONTC',  dst_get='ALL' /
 &nam_scup_pre var_put='ALATTC',  var_get='ALATTC',  dst_get='ALL' /
 &nam_scup_pre var_put='ALONUC',  var_get='ALONUC',  dst_get='ALL' /
 &nam_scup_pre var_put='ALATUC',  var_get='ALATUC',  dst_get='ALL' /
# activate if l_sss_rst_cnsv = .true.
 &nam_scup_pre var_put='SFLXADJ', var_get='SFLXADJ', dst_get='ALL' /
#
# (child -> parent)
 &nam_scup_pre model_put='NP01', model_get='GLOBAL', type='REAL8' /
# activate if l_sss_rst_cnsv = .true. and flg_trcflux_cnsv=.true.,
 &nam_scup_pre var_put='SFLXSUB', var_get='SFLXSUB', dst_get='ALL' /
#
# main-communicator
#
# (parent -> child) side boundary prognosticated variables (ALWAYS activate)
 &nam_scup model_put='GLOBAL'    , model_get='NP01'   ,
    comp_put ='BAROCLIP_GLOBAL' , comp_get ='BAROCLIS_NP01'   ,
    grid_put ='BCLI3DP_GLOBAL_T', grid_get ='BCLI3DS_NP01_T'  ,
    fl_remap='../data-np/rmp_tw_p2ct_3d.d' /
 &nam_scup
    var_put='TRC_SIDE01', var_get='TRC_SIDE01', intvl=-1, lag=0, flag='SNP'/
 &nam_scup
    var_put='TRC_SIDE02', var_get='TRC_SIDE02', intvl=-1, lag=0, flag='SNP'/
#
# (child -> parent) for replace (activate IF flg_feedback_replace=.true.)
 &nam_scup model_put='NP01'    , model_get='GLOBAL'   ,
    comp_put ='BAROCLIS_NP01'    , comp_get ='BAROCLIP_GLOBAL',
    grid_put ='BC3DS_NP01_TC'    , grid_get ='BC3DP_GLOBAL_TA',
    fl_remap='../data-np/rmp_c2pt_merge_3d.d' /
 &nam_scup
   var_put='TRC_REPL01', var_get='TRC_REPL01', intvl=-1, lag=0, flag='SNP'/
 &nam_scup
   var_put='TRC_REPL02', var_get='TRC_REPL02', intvl=-1, lag=0, flag='SNP'/
```

c. Runtime specification for parent model

At run time, namelist nml_nest_par (Table 22.7) must be specified. They are read by nest_scup_par.F90 on execution. See docs/README_Namelist.md for details. If sponge layer is used for the parent model (flg_sponge = .true.), the two-dimensional distribution of the diffusion and viscosity coefficient in the sponge layer is read from a file whose name is given by namelist nml_par_diff (Table 22.8) and nml_par_visc (Table 22.9).

Table22.7 Namelist `nml_nest_par` for the on-line parent model

variable name	units	description	usage
flg_carbon	logical	Send surface CO2 flux or not	
flg_verbose	logical	Output details of the processing	
flg_sponge	logical	Sponge region is set just outside the boundary	
flg_sponge_sub	logical	Send data to child model's sponge region	
flg_feedback_nudge	logical	Use nudging of tracers as two-way feedback	default = .false.
flg_feedback_replace	logical	Use replacing of all fields as two-way feedback	default = .false.
flg_feedback_repl_bnd	logical	Replace boundary region only in two-way feedback	default = .false.
flg_volflux_cnsv	logical	Impose conservation of volume flux at u(v)star for child model	default = .false.
flg_iceflux_cnsv	logical	Impose conservation of sea ice flux at u(v)star for child model	default = .false.
flg_send_sshhdif_c2p	logical	Send SSH diffusion flux from child to parent	default = .false.
flg_trcflux_cnsv	logical	Impose conservation of tracer flux at u(v)star for child model	default = .false.
flg_send_trchdif_p2c	logical	Send tracer lateral diffusion flux from parent to child	default = .false., choose either p2c or c2p or neither
flg_send_trchdif_c2p	logical	Send tracer lateral diffusion flux from child to parent	default = .false., choose either p2c or c2p or neither
flg_send_trcidif_p2c	logical	Send tracer isopycnal diffusion flux from parent to parent	default = .false., choose either p2c or c2p or neither
flg_send_trcidif_c2p	logical	Send tracer isopycnal diffusion flux from child to parent	default = .false., choose either p2c or c2p or neither
chfbc_sub_day	day	restoring time of received tracer in day	
scup_norecv	1	undefined value of received tracer given by scup	
nest_depth	integer	nesting is operated from surface to nest_depth level	
num_sub_models	integer	the number of child (sub) models	default = 1
file_upscale	character	Name of the file that contains mask of the upscale region	

Table22.8 Namelist `nml_par_diff` for the on-line parent model

variable name	description	usage
file_sponge_par_diff	2D distribution of horizontal diffusivity just outside the boundary	

Table22.9 Namelist `nml_par_visc` for the on-line parent model

variable name	description	usage
file_sponge_par_visc	2D distribution of horizontal viscocity just outside the boundary	

d. Runtime specification for child model

At run time, namelist `nml_submodel_grid` (Table 22.10) and `nml_nest_sub` (Table 22.11) must be specified. They are read by `nest_scup_sub.F90` on execution. See `docs/README_Namelist.md` for details.

The two dimensional distribution of the diffusion and viscosity coefficient in the sponge layer is optionally read from a file whose name is given by namelist `nml_sub_diff` (Table 22.12), `nml_sub_visc` (Table 22.13) (if `flg_sponge_sub` = `.true.`), `nml_sub_isopyc` (Table 22.14), and `nml_sub_ssh_diff` (Table 22.15).

Table22.10 Namelist `nml_submodel_grid` for the on-line child model

variable name	description	usage
`parent_grid_im`	Total number of grid points in X direction of the parent model	
`parent_grid_jm`	Total number of grid points in Y direction of the parent model	
`sub_region_ifirst`	the western end point in the parent model to be used to interpolate data (T-point)	
`sub_region_ilast`	the eastern end point in the parent model to be used to interpolate data (T-point)	
`sub_region_jfirst`	the southern end point in the parent model to be used to interpolate data (T-point)	
`sub_region_jlast`	the northern end point in the parent model to be used to interpolate data (T-point)	
`dx_ratio_w`	refinement ratio at west boundary	odd number, zero for no nesting along this side
`dx_ratio_e`	refinement ratio at east boundary	odd number, zero for no nesting along this side
`dy_ratio_s`	refinement ratio at south boundary	odd number, zero for no nesting along this side
`dy_ratio_n`	refinement ratio at north boundary	odd number, zero for no nesting along this side
`l_parent_tripolar`		`.true.` : The parent model uses `TRIPOLAR` grid

Table22.11 Namelist `nml_nest_sub` for the on-line child model

variable name	units	description	usage
`file_sub_wgt_t`	character	filename of weighing factor for parent/child model around the boundary (T-points)	
`file_sub_wgt_u`	character	filename of weighing factor for parent/child model around the boundary (U-points)	
`file_sub_wgt_x`	character	filename of weighing factor for parent/child model around the boundary (Ustar-points)	
`file_sub_wgt_y`	character	filename of weighing factor for parent/child model around the boundary (Vstar-points)	
`flg_carbon`	logical	Send surface CO2 flux or not	
`flg_verbose`	logical	Output details of the processing	
`flg_sponge_sub`	logical	Sponge region is set just inside the boundary	
`flg_feedback_nudge`	logical	Use nudging of tracers as two-way feedback	default = `.false.`
`flg_feedback_replace`	logical	Use replacing of all fields as two-way feedback	default = `.false.`
`flg_volflux_cnsv`	logical	Impose conservation of volume flux at u(v)star for the child model	default = `.false.`
`flg_iceflux_cnsv`	logical	Impose conservation of sea ice flux at u(v)star for the child model	default = `.false.`
`flg_send_sshhdif_c2p`	logical	Send SSH diffusion flux from child to parent	default = `.false.`
`flg_trcflux_cnsv`	logical	Impose conservation of tracer flux at u(v)star for the child model	default = `.false.`

Continued on next page

Table 22.11 – continued from previous page

variable name	units	description	usage
flg_send_trchdif_p2c	logical	Send tracer lateral diffusion flux from parent to child	default = .false., choose either p2c or c2p or neither
flg_send_trchdif_c2p	logical	Send tracer lateral diffusion flux from child to parent	default = .false., choose either p2c or c2p or neither
flg_send_trcidif_p2c	logical	Send tracer isopycnal diffusion flux from parent to parent	default = .false., choose either p2c or c2p or neither
flg_send_trcidif_c2p	logical	Send tracer isopycnal diffusion flux from child to parent	default = .false., choose either p2c or c2p or neither
nest_depth	integer	Nesting is operated from surface to nest_depth level	

Table22.12 Namelist nml_sub_diff for the on-line child model (tracer_nest.F90)

variable name	description	usage
file_sponge_sub_diff	3D distribution of horizontal diffusivity just inside the boundary	valid if flg_sponge_sub = .true.

Table22.13 Namelist nml_sub_visc for the on-line child model (clinic_nest.F90)

variable name	description	usage
file_sponge_sub_visc	3D distribution of horizontal diffusivity just inside the boundary	valid if flg_sponge_sub = .true.

Table22.14 Namelist nml_sub_isopyc for the on-line child model (ipcoef.F90)

variable name	description	usage
l_isopyc_mask_file	.true./.false 2D distribution of isopycnal diffusion mask is read from file	
file_isopyc_mask_sub	2D mask of isopycnal and GM diffusion	diffusion coefficients are determined as in the normal case

Table22.15 Namelist nml_sub_ssh_diff for the on-line child model (surface_integ.F90)

variable name	description	usage
l_ssh_diff_file	.true./.false 2D-distribution of SSH diffusion is read from file	
file_ssh_diff_sub	2D distribution of horizontal diffusivity of SSH	