Part IV

Evolution of tracers

Chapter 8

Tracer advection schemes

This chapter describes tracer advection schemes available in MRI.COM. The tracer advection schemes available in MRI.COM are: the weighted upcurrent scheme (Section 8.2), the Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979) as described in Section 8.3 (`QUICKADVEC` option), a combination of the QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979) for vertical advection (Section 8.4) and the Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993), which is a two-dimensional generalization of the QUICKEST scheme, for horizontal advection (Section 8.5) with `UTZQADVEC` option. For QUICKEST and UTOPIA, a flux limiter that prevents unrealistic extrema is applied (Leonard et al., 1994).

The above schemes seek to improve the accuracy by refining the finite-difference expression at the cell faces. There are other approaches. The Second Order Moment (SOM; Prather, 1986) scheme seeks to improve the accuracy by considering the distribution within the cell and is available in MRI.COM through `SOMADVEC` option (Section 8.6). The Multidimensional Positive Definite Advection Transport Algorithm (MPDATA; Smolarkiewicz and Margolin, 1998) scheme seeks to cancel the numerical diffusion of the lower order scheme by using anti-diffusive velocities and is available in MRI.COM through `MPDATAADVEC` option (Section 8.7).

Different advection schemes can be used for individual tracers. They should be chosen from among compiled schemes at run time (Chapter 13).

8.1 Finite volume or flux form method

The equation governing the time change of tracer T is

$$\frac{\partial(z_s T)}{\partial t} + \frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi u T)}{\partial \mu} + \frac{\partial(z_s h_\mu v T)}{\partial \psi}\right\} + \frac{\partial(z_s \dot{s} T)}{\partial s} = \frac{\partial(z_s T)}{\partial t} + \mathcal{A}(T) = -z_s \nabla \cdot \mathbf{F}_T + z_s Q^T, \tag{8.1}$$

where $\mathcal{A}(T)$ represents the advection operator for T. The finite difference form of the l.h.s. of (8.1) is given by first considering a control cell volume and then calculating fluxes through each cell face, and setting their divergence and convergence to be the time change rate at the grid cell (Figure 8.1). In finite difference form, this is expressed as follows:

$$\begin{aligned} T^{n+1}_{i,j,k-\frac{1}{2}} \Delta V^{n+1}_{i,j,k-\frac{1}{2}} &= T^{n-1}_{i,j,k-\frac{1}{2}} \Delta V^{n-1}_{i,j,k-\frac{1}{2}} \\ &\quad + 2\,\Delta t \left\{ FXA_{i-\frac{1}{2},j,k-\frac{1}{2}} - FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} + FYA_{i,j-\frac{1}{2},k-\frac{1}{2}} - FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} + FZA_{i,j,k} - FZA_{i,j,k-1} \right\}, \end{aligned} \tag{8.2}$$

where ΔV is the volume of the grid cell, and FXA, FYA, and FZA represent (*flux due to advection*) $\times$ (*area of the cell boundary*). The same consideration may be applied for the discretization of the diffusion term.

As explained in Chapter 3, ΔV varies with time and is given by (3.28),

$$\begin{aligned} \Delta V_{i,j,k-\frac{1}{2}} \equiv (\texttt{volt})_{i,j,k-\frac{1}{2}} &= (\texttt{volu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{volu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ &\quad + (\texttt{volu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{volu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}. \end{aligned} \tag{8.3}$$

The volume of the left-lower quarter of the T-cells at $(i,\ j)$ (corresponding to the southwestern part in geographic coordinates) is represented by $(\texttt{volt\_tr})_{i-\frac{1}{2},j-\frac{1}{2}}$. Similarly, $(\texttt{volt\_tl})_{i+\frac{1}{2},j-\frac{1}{2}}$ is the right-lower (southeastern), $(\texttt{volt\_br})_{i-\frac{1}{2},j+\frac{1}{2}}$ is the left-upper (northwestern), and $(\texttt{volt\_bl})_{i+\frac{1}{2},j+\frac{1}{2}}$ is the right-upper (northeastern) quarter of a T-cell at $(i,\ j)$.

Fluxes due to advection are given as follows:

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} = U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{8.4}$$

$$FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} = V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{8.5}$$

$$FZA_{i,j,k} = W^T_{i,j,k} T_{i,j,k}, \tag{8.6}$$

where horizontal volume transport U^T and V^T are defined as follows, using (5.13) to (5.18):

$$U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta y_{i+\frac{1}{2},j}}{2}\left(u_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right), \tag{8.7}$$

$$V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{\Delta x_{i,j+\frac{1}{2}}}{2}\left(v_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + v_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right). \tag{8.8}$$

Vertical volume transport W^T is then obtained by diagnostically solving (5.12). Moreover, the vertical velocity w, which is necessary for using QUICKEST, is calculated as follows (w is not needed except for QUICKEST):

$$W^T_{i,j,k} = w_{i,j,k} \times (\texttt{areat})_{i,j,k-\frac{1}{2}}, \tag{8.9}$$

where

$$\begin{aligned}(\texttt{areat})_{i,j,k-\frac{1}{2}} = {} & (\texttt{a\_tr})_{i-\frac{1}{2},j-\frac{1}{2}} \times (\texttt{aexl})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (\texttt{a\_tl})_{i+\frac{1}{2},j-\frac{1}{2}} \times (\texttt{aexl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ & + (\texttt{a\_br})_{i-\frac{1}{2},j+\frac{1}{2}} \times (\texttt{aexl})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{a\_bl})_{i+\frac{1}{2},j+\frac{1}{2}} \times (\texttt{aexl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}.\end{aligned} \tag{8.10}$$

An array `aexl` is set to be unity if the corresponding U-cell is a sea cell and set to be zero otherwise.

This formulation does not depend on the choice of the advection scheme. The difference arises from the way of determining cell boundary values of tracer, $T_{i+\frac{1}{2},j,k-\frac{1}{2}}$, $T_{i,j+\frac{1}{2},k-\frac{1}{2}}$, and $T_{i,j,k}$. There are several choices for tracer advection schemes, which are explained in the following sections.

- Weighted upcurrent scheme (`adv_scheme%name` = "`upc`"). This scheme is always available for use. Specify the weighting ratio listed on Table 8.1.
- Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979) for both horizontal and vertical direction (`QUICKADVEC` option; `adv_scheme%name` = "`quick`").
- The combination of the Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993) for horizontal direction and QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979) for the vertical direction (`UTZQADVEC` option; `adv_scheme%name` = "`utzqadv`").
- Second Order Moment (SOM; Prather, 1986) schemes for both horizontal and vertical direction (`SOMADVEC` option; `adv_scheme%name` = "`som`").
- Multidimensional Positive Definite Advection Transport Algorithm (MPDATA; Smolarkiewicz and Margolin, 1998) for all directions (`MPDATAADVEC` option; `adv_scheme%name` = "`mpdata`").

8.2 Weighted upcurrent scheme

In the weighted upcurrent scheme (e.g., Suginohara and Aoki, 1991; Yamanaka et al., 2000), the cell boundary value is determined by a weighted average of the upcurrent scheme and the centered finite difference scheme.

The upcurrent scheme employs the upstream value as the cell boundary value:

$$T^{upcurrent}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \begin{cases} T_{i,j,k-\frac{1}{2}}, & \text{if} \quad U_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\ T_{i+1,j,k-\frac{1}{2}}, & \text{if} \quad U_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0. \end{cases} \tag{8.11}$$

The centered finite difference scheme uses the average between the two neighboring points of tracer as the cell boundary value:

$$T^{center}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{T_{i+1,j,k-\frac{1}{2}} + T_{i,j,k-\frac{1}{2}}}{2}. \tag{8.12}$$

Chapter 8 Tracer advection schemes

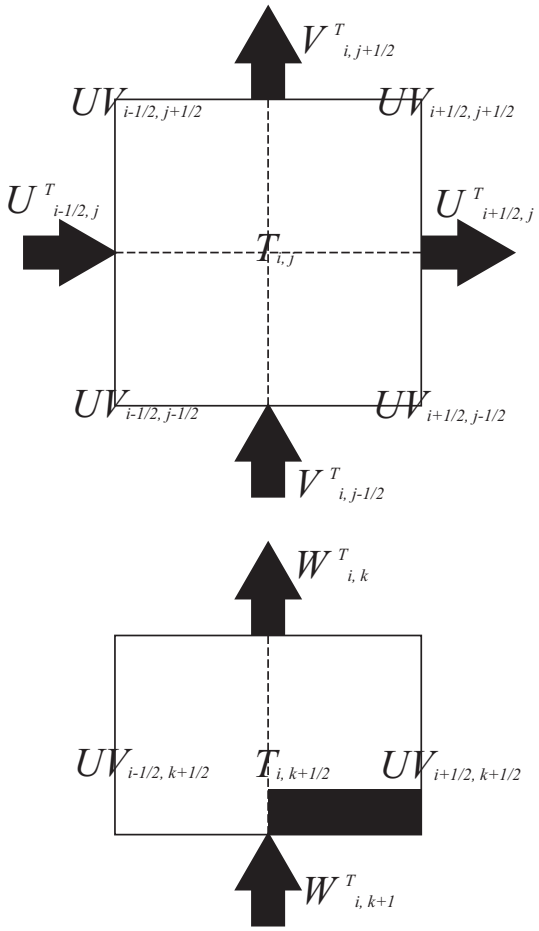

Figure8.1 Grid arrangement around TS-Box (Upper: Views from the upper, Below: views from the horizontal). Fluxes represented by an arrow are calculated.

Table8.1 Namelist `nml_tracer_adv` for weighted upcurrent scheme

variable name	units	description	usage
`weight_upcurrent_horz`	1	upstream-side weighting ratio for the horizontal advection	1.0: up-current, 0.5: centered difference
`weight_upcurrent_vert`	1	upstream-side weighting ratio for the vertical advection	1.0: up-current, 0.5: centered difference

Taking the ratio of the upcurrent scheme to be α $(0 \leq \alpha \leq 1)$, the tracer flux at the eastern face of a grid cell is expressed as follows:

$$
\begin{aligned}
FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} =& \alpha \left\{ \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} + |U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}|}{2} T_i + \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} - |U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}|}{2} T_{i+1} \right\} \\
&+ (1-\alpha) U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} \frac{T_{i+1,j,k-\frac{1}{2}} + T_{i,j,k-\frac{1}{2}}}{2} \qquad (8.13) \\
=& U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} \left\{ \frac{1}{2}\left(1 + \alpha \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}|}\right) T_i + \frac{1}{2}\left(1 - \alpha \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}|}\right) T_{i+1} \right\}. \qquad (8.14)
\end{aligned}
$$

Mainly for the sake of computational efficiency, we give a parameter $\beta = \frac{1}{2}(\alpha + 1)$ instead of α at run time. Different parameters for horizontal and vertical directions may be given, which are listed on Table 8.1.

8.3 QUICK scheme

This section explains the QUICK scheme. In the QUICK scheme, the cell boundary value is interpolated by a quadratic function, using three points, with one of them added from the upstream side (Figure 8.2).

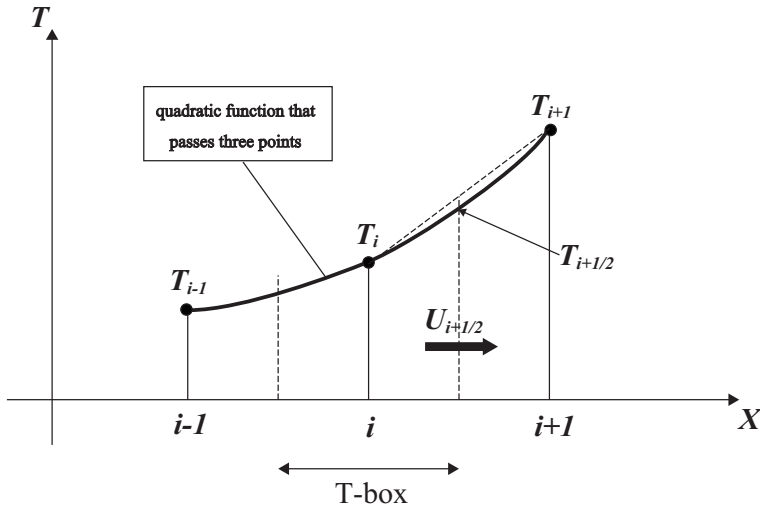

Figure8.2 Schematic for interpolation: T_i represents the value of a tracer at T-point with index i, and $T_{i+\frac{1}{2}}$ is the cell boundary value. In the QUICK scheme, $T_{i+\frac{1}{2}}$ is interpolated by a quadratic function that passes T_i and the neighboring T-point values, T_{i-1} and T_{i+1}.

Originally, cell boundary values in the QUICK scheme are given as follows:

$$T_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta x_i T_{i+1,j,k-\frac{1}{2}} + \Delta x_{i+1} T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+1} + \Delta x_i} - \frac{\Delta x_{i+1}\Delta x_i}{4} c_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{8.15}$$

$$T_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{\Delta y_j T_{i,j+1,k-\frac{1}{2}} + \Delta y_{j+1} T_{i,j,k-\frac{1}{2}}}{\Delta y_{j+1} + \Delta y_j} - \frac{\Delta y_{j+1}\Delta y_j}{4} d_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{8.16}$$

$$T_{i,j,k} = \frac{\Delta z_{k-\frac{1}{2}} T_{i,j+1,k+\frac{1}{2}} + \Delta z_{k+\frac{1}{2}} T_{i,j,k-\frac{1}{2}}}{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}} - \frac{\Delta z_{k-\frac{1}{2}}\Delta z_{k+\frac{1}{2}}}{4} e_{i,j,k}, \tag{8.17}$$

where c, d, and e are defined depending on the direction of the mass flux as follows:

$$\begin{aligned}
c_{i+\frac{1}{2},j,k-\frac{1}{2}} &= \frac{\Delta x_i \delta_x \delta_x T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta x_i}^{x}} (\equiv c_p), \quad \text{if} \quad U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\
&= \frac{\Delta x_{i+1} \delta_x \delta_x T_{i+1,j,k-\frac{1}{2}}}{2\overline{\Delta x_{i+1}}^{x}} (\equiv c_m), \quad \text{if} \quad U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0, \\
d_{i,j+\frac{1}{2},k-\frac{1}{2}} &= \frac{\Delta y_j \delta_y \delta_y T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta y_j}^{y}} (\equiv d_p), \quad \text{if} \quad V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} > 0, \\
&= \frac{\Delta y_{j+1} \delta_y \delta_y T_{i,j+1,k-\frac{1}{2}}}{2\overline{\Delta y_{j+1}}^{y}} (\equiv d_m), \quad \text{if} \quad V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} < 0, \\
e_{i,j,k} &= \frac{\Delta z_{k+\frac{1}{2}} \delta_z \delta_z T_{i,j,k+\frac{1}{2}}}{2\overline{\Delta z_{i,j,k+\frac{1}{2}}}^{z}} (\equiv e_p), \quad \text{if} \quad W^T_{i,j,k} > 0, \\
&= \frac{\Delta z_{k-\frac{1}{2}} \delta_z \delta_z T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta z_{i,j,k-\frac{1}{2}}}^{z}} (\equiv e_m), \quad \text{if} \quad W^T_{i,j,k} < 0.
\end{aligned} \tag{8.18}$$

The finite difference operators are defined as follows (definitions in y and z directions are the same):

$$\delta_x A_i \equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_i}, \quad \delta_x A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_i}{\Delta x_{i+\frac{1}{2}}},$$
$$\overline{A_i}^x \equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \quad \overline{A_{i+\frac{1}{2}}}^x \equiv \frac{A_{i+1} + A_i}{2}. \tag{8.19}$$

Letting c_p, d_p, and e_p represent their values for positive velocity at the cell boundary and c_m, d_m, and e_m represent their values for negative velocity at cell boundary and taking

$$c_a = c_m + c_p \tag{8.20}$$
$$c_d = c_m - c_p \tag{8.21}$$
$$d_a = d_m + d_p \tag{8.22}$$
$$d_d = d_m - d_p \tag{8.23}$$
$$e_a = e_m + e_p \tag{8.24}$$
$$e_d = e_m - e_p, \tag{8.25}$$

we obtain

$$\begin{aligned} FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} &= U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} \left[\frac{\Delta x_{i,j} T_{i+1,j,k-\frac{1}{2}} + \Delta x_{i+1,j} T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+1,j} + \Delta x_{i,j}} - \frac{\Delta x_{i+1,j} \Delta x_{i,j}}{8} c_{a\,i+\frac{1}{2},j,k-\frac{1}{2}} \right] \\ &\quad + |U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}| \frac{\Delta x_{i+1,j} \Delta x_{i,j}}{8} c_{d\,i+\frac{1}{2},j,k-\frac{1}{2}}, \end{aligned} \tag{8.26}$$

$$\begin{aligned} FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} &= V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} \left[\frac{\Delta y_{i,j} T_{i,j+1,k-\frac{1}{2}} + \Delta y_{i,j+1} T_{i,j,k-\frac{1}{2}}}{\Delta y_{i,j+1} + \Delta y_{i,j}} - \frac{\Delta y_{i,j+1} \Delta y_{i,j}}{8} d_{a\,i,j+\frac{1}{2},k-\frac{1}{2}} \right] \\ &\quad + |V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}| \frac{\Delta y_{i,j+1} \Delta y_{i,j}}{8} d_{d\,i,j+\frac{1}{2},k-\frac{1}{2}}, \end{aligned} \tag{8.27}$$

$$\begin{aligned} FZA_{i,j,k} &= W^T_{i,j,k} \left[\frac{\Delta z_{i,j,k-\frac{1}{2}} T_{i,j,k+\frac{1}{2}} + \Delta z_{i,j,k+\frac{1}{2}} T_{i,j,k-\frac{1}{2}}}{\Delta z_{i,j,k+\frac{1}{2}} + \Delta z_{i,j,k-\frac{1}{2}}} - \frac{\Delta z_{i,j,k+\frac{1}{2}} \Delta z_{i,j,k-\frac{1}{2}}}{8} e_{a\,i,j,k} \right] \\ &\quad + |W^T_{i,j,k-\frac{1}{2}}| \frac{\Delta z_{i,j,k+\frac{1}{2}} \Delta z_{i,j,k-\frac{1}{2}}}{8} e_{d\,i,j,k}. \end{aligned} \tag{8.28}$$

Equation (8.26) can be rewritten as

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} \simeq U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} \tilde{T}_{i+\frac{1}{2},j,k-\frac{1}{2}} + A_Q \frac{\partial^3 T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\partial x^3}, \tag{8.29}$$

where $\tilde{T}_{i+\frac{1}{2},j,k-\frac{1}{2}}$ is the value of T at the cell boundary interpolated by the cubic polynomial, and

$$A_Q = |U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}| \frac{\Delta x_{i+1} \Delta x_{i+\frac{1}{2}} \Delta x_i}{8}. \tag{8.30}$$

Although the time integration for advection is done by the leap-frog scheme, the second term on the r.h.s. of (8.29) has a biharmonic diffusion form, and thus the forward scheme is used to achieve calculation stability (Holland et al., 1998).

A similar procedure is applied for the north-south and vertical directions.

The weighted up-current scheme is used for vertical direction if $w_{i,j,k-1} > 0$ and the T-point at $(i, j, k + \frac{1}{2})$ is below the bottom. The upstream-side weighting ratio is given by the user as the namelist parameter specified for the up-current scheme (Table 8.1).

8.4 QUICKEST for vertical advection

This section describes the specific expression and the accuracy of the QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979) for vertical advection.

Consider a one-dimensional equation of advection for incompressible fluid

$$\frac{\partial T}{\partial t} + \frac{\partial}{\partial z}(wT) = 0, \tag{8.31}$$

where w is a constant. Although the velocities are not uniform in the real three dimensional ocean, we assume a constant velocity for simplicity.

Following the notation of vertical grid points and their indices (Section 3.2), tracers are defined at the center $(k - \frac{1}{2})$ of the vertical cells and vertical velocities are defined at the top $(k - 1)$ and bottom (k) faces of the vertical cells. The following relation holds for the vertical grid spacing:

$$\Delta z_k = \frac{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}}{2}. \tag{8.32}$$

In QUICKEST, the distribution of tracer T is defined using the second order interpolations, and the mean value during a time step at the cell face (boundary of two adjacent tracer cells) is calculated. The coefficients for the second order interpolation are calculated first. A Taylor expansion of T about point z_k gives

$$T_{k-\frac{3}{2}} = c_0 + c_1\left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right) + c_2\left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right)^2 + O(\Delta z^3), \tag{8.33}$$

$$T_{k-\frac{1}{2}} = c_0 + c_1\frac{\Delta z_{k-\frac{1}{2}}}{2} + c_2\frac{\Delta z_{k-\frac{1}{2}}^2}{4} + O(\Delta z^3), \tag{8.34}$$

$$T_{k+\frac{1}{2}} = c_0 - c_1\frac{\Delta z_{k+\frac{1}{2}}}{2} + c_2\frac{\Delta z_{k+\frac{1}{2}}^2}{4} + O(\Delta z^3), \tag{8.35}$$

$$T_{k+\frac{3}{2}} = c_0 - c_1\left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right) + c_2\left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right)^2 + O(\Delta z^3). \tag{8.36}$$

Coefficients c_0, c_1, and c_2 can be solved using three of the four equations (8.33), (8.34), (8.35), and (8.36). The three upstream-side equations are chosen. When $w > 0$ ($w < 0$), equations (8.34), (8.35), and (8.36) ((8.33), (8.34), and (8.35)) are used. The solution is as follows:

$$c_0 = \frac{T_{k-\frac{1}{2}}\Delta z_{k+\frac{1}{2}} + T_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{2\Delta z_k} - \frac{\Delta z_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{4}c_2, \tag{8.37}$$

$$c_1 = \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k} - \frac{\Delta z_{k-\frac{1}{2}} - \Delta z_{k+\frac{1}{2}}}{2}c_2, \tag{8.38}$$

$$c_2 = \begin{cases} \frac{1}{\Delta z_k + \Delta z_{k+1}}\left(\frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k} - \frac{T_{k+\frac{1}{2}} - T_{k+\frac{3}{2}}}{\Delta z_{k+1}}\right) & (w > 0), \\ \frac{1}{\Delta z_{k-1} + \Delta z_k}\left(\frac{T_{k-\frac{3}{2}} - T_{k-\frac{1}{2}}}{\Delta z_{k-1}} - \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k}\right) & (w < 0). \end{cases} \tag{8.39}$$

Next, equation (8.31) is integrated over one time step and one grid cell.

$$\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz \frac{\partial T}{\partial t} = -\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz \frac{\partial}{\partial z}(wT). \tag{8.40}$$

The r.h.s. of (8.40) can be written as

$$-\int_{t^n}^{t^{n+1}} dt(w_u T_u - w_l T_l), \tag{8.41}$$

where subscript u (l) denotes $z = z_{k-1}$ ($z = z_k$). Assuming that w does not depend on time,

$$\int_{t^n}^{t^{n+1}} dt T_l = \int_{-w_l\Delta t}^{0} [c_0^n + c_1^n\xi + c_2^n\xi^2 + O(\Delta z^3)]\frac{d\xi}{w_l}. \tag{8.42}$$

Thus expression (8.41) becomes

$$-\Delta t(w_u\overline{T_u^n} - w_l\overline{T_l^n}) + O(\Delta z^3 w\Delta t), \tag{8.43}$$

where

$$\begin{aligned}\overline{T_l^n} &= \frac{1}{w_l \Delta t}\int_{-w_l\Delta t}^{0}(c_0^n + c_1^n\xi + c_2^n\xi^2)d\xi \\ &= c_0^n - \frac{c_1^n}{2}w_l\Delta t + \frac{c_2^n}{3}w_l^2\Delta t^2. \end{aligned} \tag{8.44}$$

Using up to the second order terms of a Taylor expansion, the l.h.s. of (8.40) can be written as follows:

$$\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz \frac{\partial T}{\partial t} = \Delta z_{k-\frac{1}{2}}\left[T_{k-\frac{1}{2}}^{n+1} - T_{k-\frac{1}{2}}^{n} + \frac{\Delta z_{k-\frac{1}{2}}^2}{24}(Tzz_{k-\frac{1}{2}}^{n+1} - Tzz_{k-\frac{1}{2}}^{n}) + O(\Delta z^3)\right], \tag{8.45}$$

where

$$\begin{aligned} Tzz_{k-\frac{1}{2}}^{n+1} - Tzz_{k-\frac{1}{2}}^{n} &= \Delta t \left.\frac{\partial T_{zz}}{\partial t}\right|_{k-\frac{1}{2}}^{n} + O(\Delta t^2) \\ &= -\Delta t \frac{\partial^2}{\partial z^2}\left[\frac{\partial}{\partial z}(wT)\right]_{k-\frac{1}{2}}^{n} + O(\Delta t^2) \\ &= -\Delta t \frac{\partial}{\partial z}(wT_{zz})_{k-\frac{1}{2}}^{n} + O(\Delta t^2) \\ &= -\Delta t \left\{\frac{w_u Tzz_u^n - w_l Tzz_l^n}{\Delta z_{k-\frac{1}{2}}}\right\} + O(w\Delta t\Delta z) + O(\Delta t^2). \end{aligned} \tag{8.46}$$

The expression for the r.h.s. of (8.45) becomes

$$\Delta z_{k-\frac{1}{2}}\left[T_{k-\frac{1}{2}}^{n+1} - T_{k-\frac{1}{2}}^{n} - \frac{\Delta z_{k-\frac{1}{2}}^2}{24}\Delta t \frac{w_u Tzz_u^n - w_l Tzz_l^n}{\Delta z_{k-\frac{1}{2}}} + O(\Delta z^3)\right] + O(w\Delta t\Delta z^4) + O(\Delta z^3\Delta t^2). \tag{8.47}$$

Based on (8.43) and (8.47), the discretized forecasting equation is expressed as follows:

$$T_{k-\frac{1}{2}}^{n+1} = T_{k-\frac{1}{2}}^{n} - \frac{\Delta t}{\Delta z_{k-\frac{1}{2}}}\left[w_u\overline{T_u^n} - w_l\overline{T_l^n} - \frac{\Delta z_{k-\frac{1}{2}}^2}{24}(w_u Tzz_u^n - w_l Tzz_l^n)\right] + O(\alpha\Delta z^4) + O(\Delta z^2\Delta t^2), \tag{8.48}$$

where

$$\alpha \equiv \frac{w\Delta t}{\Delta z} < 1, \tag{8.49}$$

$$Tzz_l^n = 2c_2 + O(\Delta z). \tag{8.50}$$

The accuracy of equation (8.48) is $\max(O(\Delta z^4), O(\Delta z^2\Delta t^2))$.

8.5 UTOPIA for horizontal advection

The Uniformly Third Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993) is an advection scheme that can be regarded as a multi-dimensional version of QUICKEST. In `UTZQADVEC` option, horizontally two-dimensional advection is calculated using UTOPIA. Vertical advection is calculated separately using QUICKEST.

Since grid intervals could be variable in both zonal and meridional directions in MRI.COM, UTOPIA is formulated based on a variable grid interval. It is assumed that the tracer cell is subdivided by the borderlines of the velocity cells into four boxes with (almost) identical area.

Consider an equation of advection:

$$\frac{\partial T}{\partial t} + \frac{1}{h_\mu h_\psi}\frac{\partial}{\partial \mu}(h_\psi uT) + \frac{1}{h_\mu h_\psi}\frac{\partial}{\partial \psi}(h_\mu vT) = 0. \tag{8.51}$$

Integrated over a tracer cell and for one time step,

$$\int_{\psi_L-\Delta\psi_L/2}^{\psi_L+\Delta\psi_L/2} d\psi \int_{\mu_L-\Delta\mu_L/2}^{\mu_L+\Delta\mu_L/2} d\mu(\chi^{n+1} - \chi^n) = -\Delta t(u_r^n\overline{T_r^n}\Delta y_r - u_l^n\overline{T_l^n}\Delta y_l + v_u^n\overline{T_u^n}\Delta x_u - v_d^n\overline{T_d^n}\Delta x_d), \tag{8.52}$$

where $\chi \equiv h_\mu h_\psi T$ and $\overline{T_r^n}$ etc. on the r.h.s. are the face values described later. On the l.h.s. of (8.52), the second-order interpolation of χ is used to integrate the terms. The Taylor expansion of χ about L is given as follows (see Figure 8.3 for the label of the point):

$$\chi = \chi_L + a_{10}(\mu - \mu_L) + a_{20}(\mu - \mu_L)^2 + a_{01}(\psi - \psi_L) + a_{02}(\psi - \psi_L)^2 + a_{11}(\mu - \mu_L)(\psi - \psi_L). \tag{8.53}$$

Then values at points E, W, N, and S are

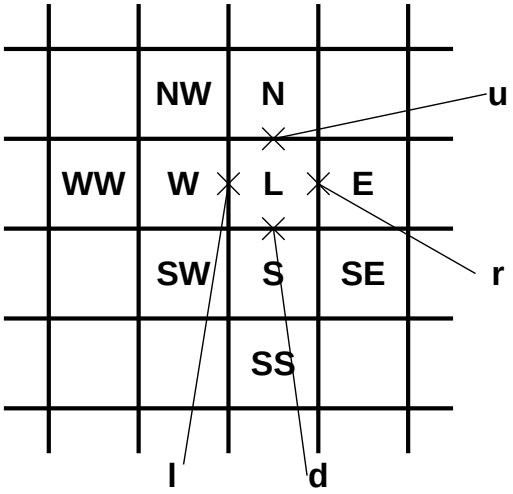

Figure8.3 Labels of tracer grid points (upper case characters) and faces (lower case characters).

$$\chi_E = \chi_L + a_{10}\Delta\mu_r + a_{20}\Delta\mu_r^2, \tag{8.54}$$
$$\chi_W = \chi_L - a_{10}\Delta\mu_l + a_{20}\Delta\mu_l^2, \tag{8.55}$$
$$\chi_N = \chi_L + a_{01}\Delta\psi_u + a_{02}\Delta\psi_u^2, \tag{8.56}$$
$$\chi_S = \chi_L - a_{01}\Delta\psi_d + a_{02}\Delta\psi_d^2, \tag{8.57}$$

where

$$\Delta\psi_u \equiv \frac{\Delta\psi_L + \Delta\psi_N}{2}, \tag{8.58}$$
$$\Delta\psi_d \equiv \frac{\Delta\psi_L + \Delta\psi_S}{2}, \tag{8.59}$$
$$\Delta\mu_r \equiv \frac{\Delta\mu_L + \Delta\mu_E}{2}, \tag{8.60}$$
$$\Delta\mu_l \equiv \frac{\Delta\mu_L + \Delta\mu_W}{2}. \tag{8.61}$$

Using these known values, the following parameters are obtained,

$$a_{10} = \frac{\Delta\mu_l \dfrac{\chi_E - \chi_L}{\Delta\mu_r} + \Delta\mu_r \dfrac{\chi_L - \chi_W}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}, \tag{8.62}$$
$$a_{20} = \frac{\dfrac{\chi_E - \chi_L}{\Delta\mu_r} - \dfrac{\chi_L - \chi_W}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}, \tag{8.63}$$
$$a_{01} = \frac{\Delta\psi_d \dfrac{\chi_N - \chi_L}{\Delta\psi_u} + \Delta\psi_u \dfrac{\chi_L - \chi_S}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}, \tag{8.64}$$
$$a_{02} = \frac{\dfrac{\chi_N - \chi_L}{\Delta\psi_u} - \dfrac{\chi_L - \chi_S}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{8.65}$$

Chapter 8 Tracer advection schemes

Substituting (8.53) into the l.h.s. of (8.52) yields

$$\Delta\mu_L\Delta\psi_L\left[\chi_L^{n+1}-\chi_L^n+\frac{\Delta\mu_L^2}{12}(a_{20}^{n+1}-a_{20}^n)+\frac{\Delta\psi_L^2}{12}(a_{02}^{n+1}-a_{02}^n)\right]. \tag{8.66}$$

Using equation (8.51), the following approximation is allowed:

$$a_{20}^{n+1}-a_{20}^n=-\Delta t\left[\frac{h_{\psi r}u_r^nT_{\mu\mu r}^n-h_{\psi l}u_l^nT_{\mu\mu l}^n}{\Delta\mu_L}+\frac{h_{\mu u}v_u^nT_{\mu\mu u}^n-h_{\mu d}v_d^nT_{\mu\mu d}^n}{\Delta\psi_L}\right], \tag{8.67}$$

$$a_{02}^{n+1}-a_{02}^n=-\Delta t\left[\frac{h_{\psi r}u_r^nT_{\psi\psi r}^n-h_{\psi l}u_l^nT_{\psi\psi l}^n}{\Delta\mu_L}+\frac{h_{\mu u}v_u^nT_{\psi\psi u}^n-h_{\mu d}v_d^nT_{\psi\psi d}^n}{\Delta\psi_L}\right], \tag{8.68}$$

where $T_{\mu\mu r}^n$ is the value of the second order derivative at the right face **r**, whose expression is similar to that of c_{20} described later. Therefore, under a suitable approximation,

$$T_L^{n+1}=T_L^n-\frac{\Delta t}{\Delta S_L}(u_r^n\tilde{T_r^n}\Delta y_r-u_l^n\tilde{T_l^n}\Delta y_l+v_u^n\tilde{T_u^n}\Delta x_u-v_d^n\tilde{T_d^n}\Delta x_d), \tag{8.69}$$

where

$$\tilde{T_l^n}=\overline{T_l^n}-\frac{\Delta\mu_L^2}{24}T_{\mu\mu l}^n-\frac{\Delta\psi_L^2}{24}T_{\psi\psi l}^n, \tag{8.70}$$

$$\tilde{T_d^n}=\overline{T_d^n}-\frac{\Delta\mu_L^2}{24}T_{\mu\mu d}^n-\frac{\Delta\psi_L^2}{24}T_{\psi\psi d}^n. \tag{8.71}$$

Next, the expressions for $\overline{T_l^n}$ and $\overline{T_d^n}$ are required. The term $\overline{T_l^n}$ is the average over the hatched area of Figure 8.4, and the values of T^n are given as the second order interpolation about **l** of Figure 8.3. Similar operations will be used to obtain the expression for $\overline{T_d^n}$.

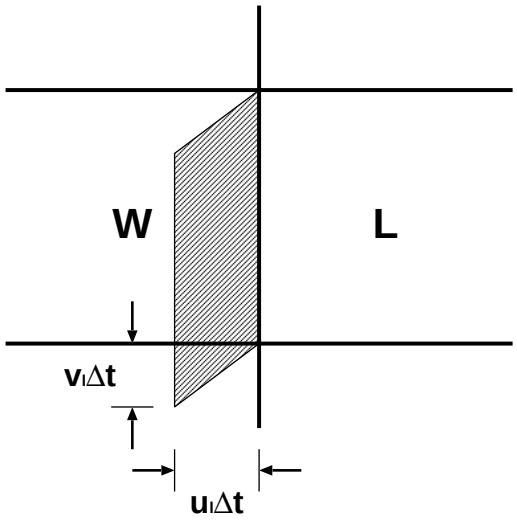

Figure8.4 Area used to average tracer values for the face **l**

First, Taylor expansions of T^n about **l** and **d** are written as follows:

$$T^n|_{\mathbf{l}}=c_{00}+c_{10}(\mu-\mu_l)+c_{20}(\mu-\mu_l)^2+c_{01}(\psi-\psi_L)+c_{02}(\psi-\psi_L)^2+c_{11}(\mu-\mu_l)(\psi-\psi_L), \tag{8.72}$$

$$T^n|_{\mathbf{d}}=d_{00}+d_{10}(\mu-\mu_l)+d_{20}(\mu-\mu_l)^2+d_{01}(\psi-\psi_L)+d_{02}(\psi-\psi_L)^2+d_{11}(\mu-\mu_l)(\psi-\psi_L). \tag{8.73}$$

The T^n values at eight points around **l** are,

$$T_L^n = c_{00} + c_{10}\frac{\Delta\mu_L}{2} + c_{20}\frac{\Delta\mu_L^2}{4}, \tag{8.74}$$

$$T_W^n = c_{00} - c_{10}\frac{\Delta\mu_W}{2} + c_{20}\frac{\Delta\mu_W^2}{4}, \tag{8.75}$$

$$T_E^n = c_{00} + c_{10}\left(\Delta\mu_L + \frac{\Delta\mu_E}{2}\right) + c_{20}\left(\Delta\mu_L + \frac{\Delta\mu_E}{2}\right)^2, \tag{8.76}$$

$$T_{WW}^n = c_{00} - c_{10}\left(\Delta\mu_W + \frac{\Delta\mu_{WW}}{2}\right) + c_{20}\left(\Delta\mu_W + \frac{\Delta\mu_{WW}}{2}\right)^2, \tag{8.77}$$

$$T_N^n = T_L^n + c_{01}\Delta\psi_u + c_{02}\Delta\psi_u^2 + c_{11}\frac{\Delta\mu_L}{2}\Delta\psi_u, \tag{8.78}$$

$$T_S^n = T_L^n - c_{01}\Delta\psi_d + c_{02}\Delta\psi_d^2 - c_{11}\frac{\Delta\mu_L}{2}\Delta\psi_d, \tag{8.79}$$

$$T_{NW}^n = T_W^n + c_{01}\Delta\psi_u + c_{02}\Delta\psi_u^2 - c_{11}\frac{\Delta\mu_W}{2}\Delta\psi_u, \tag{8.80}$$

$$T_{SW}^n = T_W^n - c_{01}\Delta\psi_d + c_{02}\Delta\psi_d^2 + c_{11}\frac{\Delta\mu_W}{2}\Delta\psi_d. \tag{8.81}$$

To obtain all six coefficients, six of these equations (points) are used. The equations are chosen according to the following flow direction.

$$u_l^n > 0, \quad v_l^n > 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{S}, \mathbf{NW}, \mathbf{SW} \tag{8.82}$$

$$u_l^n < 0, \quad v_l^n > 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{SW} \tag{8.83}$$

$$u_l^n > 0, \quad v_l^n < 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{N}, \mathbf{NW}, \mathbf{SW} \tag{8.84}$$

$$u_l^n < 0, \quad v_l^n < 0 \Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{NW} \tag{8.85}$$

From equations (8.74) and (8.75),

$$c_{00} = \frac{\Delta\mu_W T_L^n + \Delta\mu_L T_W^n}{2\Delta\mu_l} - c_{20}\frac{\Delta\mu_L\Delta\mu_W}{4}, \tag{8.86}$$

$$c_{10} = \frac{T_L^n - T_W^n}{\Delta\mu_l} - c_{20}\frac{\Delta\mu_L - \Delta\mu_W}{2}. \tag{8.87}$$

When $u_l^n > 0$, from (8.74) and (8.77),

$$c_{20} = \frac{\dfrac{T_L^n - T_W^n}{\Delta\mu_l} - \dfrac{T_W^n - T_{WW}^n}{\Delta\mu_{ll}}}{\Delta\mu_l + \Delta\mu_{ll}}, \tag{8.88}$$

$$\text{where} \quad \Delta\mu_{ll} \equiv \frac{\Delta\mu_W + \Delta\mu_{WW}}{2}.$$

Using equations (8.80) and (8.81),

$$c_{02} = \frac{\dfrac{T_{NW}^n - T_W^n}{\Delta\psi_u} - \dfrac{T_W^n - T_{SW}^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{8.89}$$

When $u_l^n < 0$, from (8.75) and (8.76),

$$c_{20} = \frac{\dfrac{T_E^n - T_L^n}{\Delta\mu_r} - \dfrac{T_L^n - T_W^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{8.90}$$

Using equations (8.78) and (8.79),

$$c_{02} = \frac{\dfrac{T_N^n - T_L^n}{\Delta\psi_u} - \dfrac{T_L^n - T_S^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{8.91}$$

When $v_l^n > 0$, from (8.79) and (8.81),

$$c_{01} = \frac{\Delta\mu_W(T_L^n - T_S^n) + \Delta\mu_L(T_W^n - T_{SW}^n)}{2\Delta\mu_l\Delta\psi_d} + c_{02}\Delta\psi_d, \tag{8.92}$$

$$c_{11} = \frac{T_{SW}^n - T_W^n - T_S^n + T_L^n}{\Delta\mu_l\Delta\psi_d}. \tag{8.93}$$

When $v_l^n < 0$, from (8.78) and (8.80),

$$c_{01} = \frac{\Delta\mu_W(T_N^n - T_L^n) + \Delta\mu_L(T_{NW}^n - T_W^n)}{2\Delta\mu_l\Delta\psi_u} - c_{02}\Delta\psi_u, \tag{8.94}$$

$$c_{11} = \frac{T_N^n - T_L^n - T_{NW}^n + T_W^n}{\Delta\mu_l\Delta\psi_u}. \tag{8.95}$$

Next, using equation (8.73), the T^n values at eight points around **d** are

$$T_L^n = d_{00} + d_{01}\frac{\Delta\psi_L}{2} + d_{02}\frac{\Delta\psi_L^2}{4}, \tag{8.96}$$

$$T_S^n = d_{00} - d_{01}\frac{\Delta\psi_S}{2} + d_{02}\frac{\Delta\psi_S^2}{4}, \tag{8.97}$$

$$T_N^n = d_{00} + d_{01}\left(\Delta\psi_L + \frac{\Delta\psi_N}{2}\right) + d_{02}\left(\Delta\psi_L + \frac{\Delta\psi_N}{2}\right)^2, \tag{8.98}$$

$$T_{SS}^n = d_{00} - d_{01}\left(\Delta\psi_S + \frac{\Delta\psi_{SS}}{2}\right) + d_{02}\left(\Delta\psi_S + \frac{\Delta\psi_{SS}}{2}\right)^2, \tag{8.99}$$

$$T_E^n = T_L^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 + d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_r, \tag{8.100}$$

$$T_W^n = T_L^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 - d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_l, \tag{8.101}$$

$$T_{SE}^n = T_S^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 - d_{11}\frac{\Delta\psi_S}{2}\Delta\mu_r, \tag{8.102}$$

$$T_{SW}^n = T_S^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 + d_{11}\frac{\Delta\psi_S}{2}\Delta\mu_l. \tag{8.103}$$

From equations (8.96) and (8.97),

$$d_{00} = \frac{\Delta\psi_S T_L^n + \Delta\psi_L T_S^n}{2\Delta\psi_d} - d_{02}\frac{\Delta\psi_L\Delta\psi_S}{4}, \tag{8.104}$$

$$d_{01} = \frac{T_L^n - T_S^n}{\Delta\psi_d} - d_{02}\frac{\Delta\psi_L - \Delta\psi_S}{2}. \tag{8.105}$$

When $v_d^n > 0$, from (8.96) and (8.99),

$$d_{02} = \frac{\dfrac{T_L^n - T_S^n}{\Delta\psi_d} - \dfrac{T_S^n - T_{SS}^n}{\Delta\psi_{dd}}}{\Delta\psi_d + \Delta\psi_{dd}}, \tag{8.106}$$

$$\text{where} \quad \Delta\psi_{dd} \equiv \frac{\Delta\psi_S + \Delta\psi_{SS}}{2}.$$

From (8.102) and (8.103),

$$d_{20} = \frac{\dfrac{T_{SE}^n - T_S^n}{\Delta\mu_r} - \dfrac{T_S^n - T_{SW}^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{8.107}$$

When $v_d^n < 0$, from (8.97) and (8.98),

$$d_{02} = \frac{\dfrac{T_N^n - T_L}{\Delta\psi_u} - \dfrac{T_L^n - T_S^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{8.108}$$

From (8.100) and (8.101),

$$d_{20} = \frac{\dfrac{T_E^n - T_L^n}{\Delta\mu_r} - \dfrac{T_L^n - T_W^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{8.109}$$

When $u_d^n > 0$, from (8.101) and (8.103),

$$d_{10} = \frac{\Delta\psi_S(T_L^n - T_W^n) + \Delta\psi_L(T_S^n - T_{SW}^n)}{2\Delta\psi_d\Delta\mu_l} + d_{20}\Delta\mu_l, \tag{8.110}$$

$$d_{11} = \frac{T_L^n - T_S^n - T_W^n + T_{SW}^n}{\Delta\psi_d\Delta\mu_l}. \tag{8.111}$$

When $u_d^n < 0$, from (8.100) and (8.102),

$$d_{10} = \frac{\Delta\psi_S(T_E^n - T_L^n) + \Delta\psi_L(T_{SE}^n - T_S^n)}{2\Delta\psi_d\Delta\mu_r} - d_{20}\Delta\mu_r, \tag{8.112}$$

$$d_{11} = \frac{T_E^n - T_{SE}^n - T_L^n + T_S^n}{\Delta\psi_d\Delta\mu_r}. \tag{8.113}$$

The value of $\overline{T_l^n}$ is the average of T^n over the hatched area of Figure 8.4. Defining

$$\xi_l^n = \frac{u_l^n}{h_{\mu l}}, \quad \eta_l^n = \frac{v_l^n}{h_{\psi l}}, \tag{8.114}$$

we have

$$\begin{aligned}\overline{T_l^n} = {} & \frac{1}{\xi_l^n\Delta t\Delta\psi_L}\left[\int_{\psi_L-\Delta\psi_L/2}^{\psi_L+\Delta\psi_L/2}\int_{\mu_l-\xi_l^n\Delta t}^{\mu_l} T^n d\mu d\psi\right. \\ & \left. + \int_{\psi_L-\Delta\psi_L/2-\eta_l^n\Delta t}^{\psi_L-\Delta\psi_L/2}\int_{\mu_l-\xi_l^n\Delta t}^{\mu_l+\frac{\xi_l^n}{\eta_l^n}(\psi-\psi_L+\Delta\psi_L/2)} \{T^n(\psi) - T^n(\psi+\Delta\psi_L)\} d\mu d\psi\right] \\ = {} & c_{00} - \frac{1}{2}\eta_l^n\Delta t\, c_{01} + \left[\frac{1}{12}\Delta\psi_L^2 + \frac{1}{3}{}^{*}(\eta_l^n\Delta t)^2\right] c_{02} \\ & - \frac{1}{2}\xi_l^n\Delta t\, c_{10} + \frac{1}{3}(\xi_l^n\Delta t)^2 c_{20} + \frac{1}{3}{}^{\dagger}\xi_l^n\eta_l^n\Delta t^2\, c_{11}.\end{aligned} \tag{8.115}$$

This is the result for $u_l^n > 0$ and $v_l^n > 0$. The result is the same independent of the sign of u_l^n and v_l^n.

Similarly,

$$\begin{aligned}\overline{T_d^n} = {} & d_{00} - \frac{1}{2}\eta_d^n\Delta t\, d_{01} + \frac{1}{3}(\eta_d^n\Delta t)^2\, d_{02} \\ & - \frac{1}{2}\xi_d^n\Delta t\, d_{10} + \left[\frac{1}{12}\Delta\mu_L^2 + \frac{1}{3}{}^{\ddagger}(\xi_d^n\Delta t)^2\right] d_{20} + \frac{1}{3}{}^{\S}\xi_d^n\eta_d^n\Delta t^2 d_{11},\end{aligned} \tag{8.116}$$

where

$$\xi_d^n = \frac{u_d^n}{h_{\mu d}}, \quad \eta_d^n = \frac{v_d^n}{h_{\psi d}}. \tag{8.117}$$

* $\frac{1}{2}$ in COCO
† $\frac{1}{4}$ in COCO
‡ $\frac{1}{2}$ in COCO
§ $\frac{1}{4}$ in COCO

Therefore,

$$\tilde{T_l^n} = c_{00} - \frac{1}{2}\xi_l^n \Delta t\, c_{10} + \left[\frac{1}{3}(\xi_l^n \Delta t)^2 - \frac{\Delta\mu_L^2}{12}\right] c_{20} - \frac{1}{2}\eta_l^n \Delta t\, c_{01} + \frac{1}{3}(\eta_l^n \Delta t)^2 c_{02} + \frac{1}{3}\xi_l^n \eta_l^n \Delta t^2\, c_{11}, \tag{8.118}$$

$$\tilde{T_d^n} = d_{00} - \frac{1}{2}\eta_d^n \Delta t\, d_{01} + \left[\frac{1}{3}(\eta_d^n \Delta t)^2 - \frac{\Delta\psi_L^2}{12}\right] d_{02} - \frac{1}{2}\xi_d^n \Delta t\, d_{10} + \frac{1}{3}(\xi_d^n \Delta t)^2 d_{20} + \frac{1}{3}\xi_d^n \eta_d^n \Delta t^2\, d_{11}. \tag{8.119}$$

Finally, we describe how to derive the boundary conditions. Since the face values of the tracers are calculated through the second order interpolation, the value of a tracer at a point over land is sometimes necessary. For that case, the value should be appropriately decided by using the tracer values at the neighboring points in the sea. Since ocean models generally assume that there is no flux of tracers across land-sea boundary, the provisional value over land should be given so as not to create a normal gradient at the boundary.

When the face value of a tracer at boundary **l** is calculated, **W** and **L** are not land, but either **N** or **S** may be land, and either **NW** or **SW** may be land. When **N** or **S** is land, the land-sea boundary runs at the center of **L** in the zonal direction. It is reasonable to assume that the value of land grid **N** or **S** must not cause any meridional tracer gradient at **L** set by second order interpolation using the values at grids **N**, **L**, and **S**. Thus, we set

$$(T_N^n - T_L^n)\Delta\psi_d^2 = (T_S^n - T_L^n)\Delta\psi_u^2. \tag{8.120}$$

When **NW** or **SW** is a land grid, the following should be assumed.

$$(T_{NW}^n - T_W^n)\Delta\psi_d^2 = (T_{SW}^n - T_W^n)\Delta\psi_u^2 \tag{8.121}$$

When **WW** is a land grid,

$$(T_{WW}^n - T_W^n)\Delta\mu_l^2 = (T_L^n - T_W^n)\Delta\mu_{ll}^2. \tag{8.122}$$

When **E** is a land grid,

$$(T_E^n - T_L^n)\Delta\mu_l^2 = (T_W^n - T_L^n)\Delta\mu_r^2. \tag{8.123}$$

Similar boundary conditions are specified for face **d**.

8.6 Second Order Moment (SOM) scheme

8.6.1 Outline

The Second Order Moment (SOM) advection scheme by Prather (1986) seeks to improve the accuracy by treating the tracer distribution within a grid cell, unlike the scheme that aims to calculate the tracer flux at the boundary of grid cells with high accuracy. It is assumed that the distribution of tracer f in a grid cell ($0 \le x \le X$, $0 \le y \le Y$, $0 \le z \le Z$; volume $V = XYZ$) can be represented using second order functions as follows:

$$f(x,y,z) = a_0 + a_x x + a_{xx}x^2 + a_y y + a_{yy}y^2 + a_z z + a_{zz}z^2 + a_{xy}xy + a_{yz}yz + a_{zx}zx. \tag{8.124}$$

Prather (1986) expressed the above as a sum of orthogonal functions $K_i(x,y,z)$;

$$f(x,y,z) = m_0K_0 + m_xK_x + m_{xx}K_{xx} + m_yK_y + m_{yy}K_{yy} + m_zK_z + m_{zz}K_{zz} + m_{xy}K_{xy} + m_{yz}K_{yz} + m_{zx}K_{zx}, \tag{8.125}$$

where the orthogonal functions are given as follows:

$$\begin{aligned}
K_0 &= 1, \\
K_x(x) &= x - X/2, \\
K_{xx}(x) &= x^2 - Xx + X^2/6, \\
K_y(y) &= y - Y/2, \\
K_{yy}(y) &= y^2 - Yy + Y^2/6, \\
K_z(z) &= z - Z/2, \\
K_{zz}(z) &= z^2 - Zz + Z^2/6, \\
K_{xy}(x, y) &= (x - X/2)(y - Y/2), \\
K_{yz}(y, z) &= (y - Y/2)(z - Z/2), \\
K_{zx}(z, x) &= (z - Z/2)(x - X/2),
\end{aligned} \tag{8.126}$$

and

$$\int K_i K_j dV = 0 \quad (i \neq j). \tag{8.127}$$

The constants for normalization are decided using

$$\int K_x^2 dV = VX^2/12, \quad \int K_{xx}^2 dV = VX^4/180,$$

$$\int K_y^2 dV = VY^2/12, \quad \int K_{yy}^2 dV = VY^4/180,$$

$$\int K_z^2 dV = VZ^2/12, \quad \int K_{zz}^2 dV = VZ^4/180,$$

$$\int K_{xy}^2 dV = VX^2Y^2/144, \quad \int K_{yz}^2 dV = VY^2Z^2/144, \quad \int K_{zx}^2 dV = VZ^2X^2/144.$$

The moments are set by the following expressions:

$$\begin{aligned}
S_0 &= \int f(x, y, z) K_0 dV = m_0 V, \\
S_x &= (6/X) \int f(x, y, z) K_x(x) dV = m_x VX/2, \\
S_{xx} &= (30/X^2) \int f(x, y, z) K_{xx}(x) dV = m_{xx} VX^2/6, \\
S_y &= (6/Y) \int f(x, y, z) K_y(y) dV = m_y VY/2, \\
S_{yy} &= (30/Y^2) \int f(x, y, z) K_{yy}(y) dV = m_{yy} VY^2/6, \\
S_z &= (6/Z) \int f(x, y, z) K_z(z) dV = m_z VZ/2, \\
S_{zz} &= (30/Z^2) \int f(x, y, z) K_{zz}(x) dV = m_{zz} VZ^2/6, \\
S_{xy} &= (36/XY) \int f(x, y, z) K_{xy}(x, y) dV = m_{xy} VXY/4, \\
S_{yz} &= (36/YZ) \int f(x, y, z) K_{yz}(y, z) dV = m_{yz} VYZ/4, \\
S_{zx} &= (36/ZX) \int f(x, y, z) K_{zx}(z, x) dV = m_{zx} VZX/4.
\end{aligned} \tag{8.128}$$

All these moments are transported with the upstream advection scheme. The procedure is carried out in one direction at a time. The second and third procedures use the results of the last procedure. For simplicity, we describe the change

of each moment caused by an advection procedure in one direction (x) in a two dimensional plane (xy) in the following. You may replace (y, Y) with (z, Z).

When velocity c in the x direction is positive, the right part of the grid cell,

$$X - ct \le x \le X, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{8.129}$$

is removed from the cell and added to the adjacent cell on the right during time interval t. This part is expressed using superscript R. The remaining part,

$$0 \le x \le X - ct, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{8.130}$$

is expressed by superscript L. New orthogonal functions K_i^R (K_i^L) are calculated in the part R (L) with the volume $V^R = ctYZ$ ($V^L = (X - ct)YZ$). The orthogonal functions are given as follows:

$$\begin{gathered}
K_0^L = K_0^R = 1, \\
K_x^L = x - (X - ct)/2, \quad K_x^R = x - (2X - ct)/2, \\
K_{xx}^L = x^2 - (X - ct)x + (X - ct)^2/6, \\
K_{xx}^R = x^2 - (2X - ct)x + (X - ct)X + (ct)^2/6, \\
\\
K_y^L = K_y^R = y - Y/2, \\
K_{yy}^L = K_{yy}^R = y^2 - Yy + Y^2/6, \\
K_{xy}^L = [x - (X - ct)/2](y - Y/2), \\
K_{xy}^R = [x - (2X - ct)/2](y - Y/2).
\end{gathered} \tag{8.131}$$

The basic quantities for calculating the moments are

$$\begin{aligned}
m_0^R &= m_0 + \bar{K}_x^R m_x + \bar{K}_{xx}^R m_{xx}, \\
m_x^R &= m_x + 2\bar{K}_x^R m_{xx}, \\
m_{xx}^R &= m_{xx}, \\
m_y^R &= m_y + 2\bar{K}_x^R m_{xy}, \\
m_{yy}^R &= m_{yy}, \\
m_{xy}^R &= m_{xy},
\end{aligned} \tag{8.132}$$

where $\bar{K}$ is the average of the new orthogonal function:

$$\begin{gathered}
\bar{K}_x^L = -ct/2, \quad \bar{K}_x^R = (X - ct)/2, \\
\bar{K}_{xx}^L = ct(2ct - X)/6, \quad \bar{K}_{xx}^R = (X - ct)(X - 2ct)/6.
\end{gathered}$$

The moments in the right part to be removed are expressed as follows:

$$\begin{aligned}
S_0^R &= \alpha[S_0 + (1 - \alpha)S_x + (1 - \alpha)(1 - 2\alpha)S_{xx}], \\
S_x^R &= \alpha^2[S_x + 3(1 - \alpha)S_{xx}], \\
S_{xx}^R &= \alpha^3 S_{xx}, \\
S_y^R &= \alpha[S_y + (1 - \alpha)S_{xy}], \\
S_{yy}^R &= \alpha S_{yy}, \\
S_{xy}^R &= \alpha^2 S_{xy},
\end{aligned} \tag{8.133}$$

where $\alpha = \alpha^R = ct/X = V^R/V$. The moments in the remaining part are expressed as follows:

$$\begin{aligned} S_0^L &= (1-\alpha)[S_0 - \alpha S_x - \alpha(1-2\alpha)S_{xx}], \\ S_x^L &= (1-\alpha)^2(S_x - 3\alpha S_{xx}), \\ S_{xx}^L &= (1-\alpha)^3 S_{xx}, \\ S_y^L &= (1-\alpha)(S_y - \alpha S_{xy}), \\ S_{yy}^L &= (1-\alpha)S_{yy}, \\ S_{xy}^L &= (1-\alpha)^2 S_{xy}. \end{aligned} \tag{8.134}$$

As the final step of the procedure, the orthogonal functions and moments transported from the adjacent cell and those in the remaining part of the original cell are combined to create new united moments in the cell. The calculation is terribly complex, and only the results are presented:

$$S_0 = S_0^R + S_0^L,$$

$$S_x = \alpha S_x^R + (1-\alpha)S_x^L + 3[(1-\alpha)S_0^R - \alpha S_0^L],$$

$$S_{xx} = \alpha^2 S_{xx}^R + (1-\alpha)^2 S_{xx}^L + 5\{\alpha(1-\alpha)(S_x^R - S_x^L) + (1-2\alpha)[(1-\alpha)S_0^R - \alpha S_0^L]\},$$

$$S_y = S_y^R + S_y^L,$$

$$S_{yy} = S_{yy}^R + S_{yy}^L,$$

$$S_{xy} = \alpha S_{xy}^R + (1-\alpha)S_{xy}^L + 3[(1-\alpha)S_y^R - \alpha S_y^L],$$

where

$$\alpha = \alpha^R = V^R/(V^R + V^L). \tag{8.135}$$

The globally integrated tracer is conserved through these operations.

8.6.2 Flux limiter

Some limiters are necessary to guarantee that a tracer is positive (negative) definite.

a. Prather (1986)

Prather (1986) proposed to set limits for the moments related to the direction of advection. For instance, when the moments are advected in the x direction,

$$\begin{aligned} S_0 &\geq 0, \\ S_x' &= \min[+1.5S_0,\ \max(-1.5S_0, S_x)], \\ S_{xx}' &= \min[2S_0 - |S_x'|/3, \max(|S_x'| - S_0, S_{xx})], \\ S_{xy}' &= \min[+S_0, \max(-S_0, S_{xy})]. \end{aligned} \tag{8.136}$$

It should be noted that the application of this limiter does not completely guarantee that the tracer will be positive (negative) definite.

b. Merryfield and Holloway (2003)

Flux limiters introduced by Morales Maqueda and Holloway (2006) (hereinafter MH06) extended that of Prather (1986) (hereinafter PR86).

Let us consider the advection in the x direction. As in PR86, MH06 consider the mean tracer distribution ($\bar{\tau}(x)$) in the x direction by integrating in the y and z direction within the grid cell (Equation (2.15) of MH06).

$$\begin{aligned} \bar{\tau}(x) &= \frac{1}{YZ}\int_0^Y \int_0^Z dy\, dz\, \tau(x,y,z) \\ &= \frac{1}{V}\left[S_0 - S_x + S_{xx} + 2(S_x - 3S_{xx})\frac{x}{X} + 6S_{xx}\left(\frac{x}{X}\right)^2\right], \end{aligned} \tag{8.137}$$

where x, y, z represent the position relative to the lower south-western corner of the grid cell and X, Y, and Z are grid widths in the x, y, and z directions, respectively. V is the volume of the grid cell.

First we consider the minimum value of $\bar{\tau}(x)$. When S_{xx} is negative (region I), the minimum is taken either at $x = 0$ or $x = X$ because $\bar{\tau}(x)$ is convex upward. Even when S_{xx} is positive, if the value of x that gives the global minimum of the quadratic function is not within $0 \le x \le X$ (region II), the minimum is taken either at $x = 0$ or $x = X$. Otherwise (region III), the minimum value is given as the global minimum of the quadratic function. These are summarized as follows:

$$V \min \bar{\tau}(x) = \begin{cases} S_0 - |S_x| + S_{xx}, & \text{if } S_{xx} \le 0 & \text{(region I)}, \\ S_0 - |S_x| + S_{xx}, & \text{if } |S_x| \ge 3S_{xx} \ge 0 & \text{(region II)}, \\ S_0 - \frac{S_x^2 + 3S_{xx}^2}{6S_{xx}}, & \text{if } 3S_{xx} \ge |S_x| & \text{(region III)}. \end{cases} \tag{8.138}$$

The line of $V \min \bar{\tau}(x) = 0$ is plotted in the $(|S_x|, S_{xx})$ space in Figure 8.5. If a pair of $(|S_x|, S_{xx})$ resides in the region bounded by the blue line and the left vertical axis, the minimum value of $\bar{\tau}(x)$ is positive.

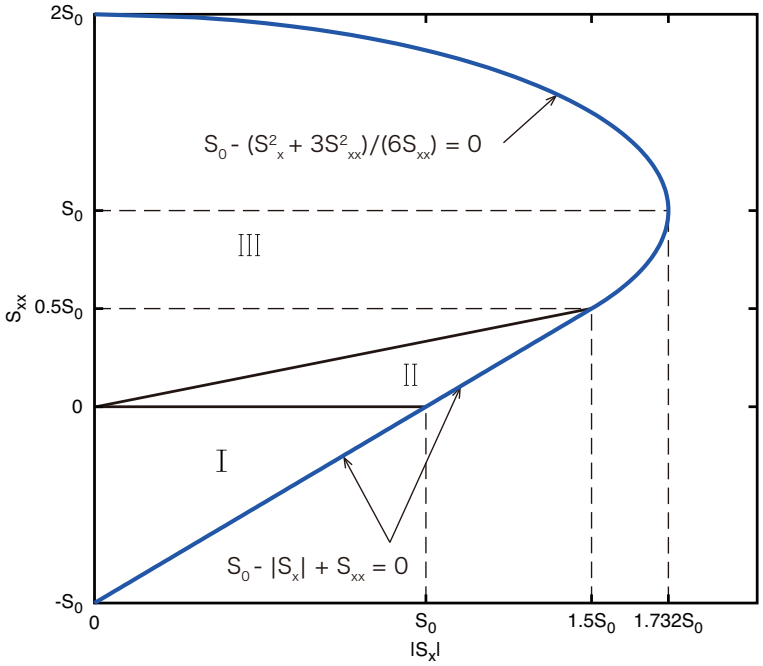

Figure8.5 Schematic to explain the flux limiter to avoid undershooting. This is a reproduction of Figure 3 of Morales Maqueda and Holloway (2006). See text for explanation.

PR86 proposed to modify the moments so that the minimum of $\bar{\tau}$ is positive. This is to improve the sign-definiteness of the tracer. The limiters of MH06 are also based on this strategy. They proposed to modify the moments in the following way,

$$\begin{aligned} S'_x &= \min\left[3^{1/2}S_0,\ \max\left[-3^{1/2}S_0,\ S_x\right]\right], \\ S'_{xx} &= \begin{cases} \min\left[S_0 + \left(S_0^2 - \frac{S_x'^2}{3}\right)^{1/2},\ \max\left[|S'_x| - S_0,\ S_{xx}\right]\right], & \text{if } |S'_x| < \frac{3}{2}S_0, \\ \min\left[S_0 + \left(S_0^2 - \frac{S_x'^2}{3}\right)^{1/2},\ \max\left[S_0 - \left(S_0^2 - \frac{S_x'^2}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } |S'_x| \ge \frac{3}{2}S_0, \end{cases} \\ S'_{xy} &= \min\left[S_0,\ \max[-S_0,\ S_{xy}]\right], \\ S'_{xz} &= \min\left[S_0,\ \max[-S_0,\ S_{xz}]\right]. \end{aligned} \tag{8.139}$$

Here, S'_x, S'_{xx}, S'_{xy} are modified moments and it is assumed that $S_0 \ge 0$. This is slightly different from PR86 because an approximation $\sqrt{3} \sim 1.5$ (see 8.136) was used in PR86.

This modification is explained using Figure 8.5. If a pair of $(|S_x|, S_{xx})$ resides in the region that gives negative minimum of $\bar{\tau}(x)$, $|S_x|$ is shifted along the horizontal axis so that it is less than $3^{1/2}S_0$. Then S_{xx} is shifted along the vertical axis so that the point is within the region that gives the minimum of $\bar{\tau}(x)$ to be positive. The modification to S_{xy}, S_{xz} is designed so that a negative value of the tracer is avoided by these moments.

MH06 proposed to extend the above modifications for setting the lower and upper bounds for the tracers to improve the monotonicity. The lower and upper limit for $\tau(x)$ are set as τ_a and τ_b, respectively ($\tau_a \le \bar{\tau}(x) \le \tau_b$). The setting of the lower limit is achieved by replacing S_0 with $S_0^* = S_0 - V\tau_a$ in the above modification. For the upper limit, we interpret the condition "the maximum of $\bar{\tau}(x)$ must be less than τ_b" as "the minimum of $-\bar{\tau}(x)$ must be more than $-\tau_b$." In other words, the condition is "the minimum of $\tau_b - \bar{\tau}(x)$ must be positive."

Using (8.137), $\tau_b - \bar{\tau}(x)$ may be expressed as follows:

$$\begin{aligned}\tau_b - \bar{\tau}(x) &= \frac{1}{V}\left[V\tau_b - S_0 + S_x - S_{xx} - 2(S_x - 3S_{xx})\frac{x}{X} - 6S_{xx}\left(\frac{x}{X}\right)^2\right] \\ &= \frac{1}{V}\left[S_0^* + S_x - S_{xx} - 2(S_x - 3S_{xx})\frac{x}{X} - 6S_{xx}\left(\frac{x}{X}\right)^2\right],\end{aligned} \tag{8.140}$$

where $S_0^* = V\tau_b - S_0$. Its minimum is obtained as follows:

$$V\ \min\left[\tau_b - \bar{\tau}(x)\right] = \begin{cases} S_0^* - |S_x| - S_{xx}, & \text{if } S_{xx} \ge 0 \quad \text{(region I)}, \\ S_0^* - |S_x| - S_{xx}, & \text{if } |S_x| \ge -3S_{xx} \ge 0 \quad \text{(region II)}, \\ S_0^* + \frac{S_x^2 + 3S_{xx}^2}{6S_{xx}}, & \text{if } -3S_{xx} \ge |S_x| \quad \text{(region III)}. \end{cases} \tag{8.141}$$

Following Figure 8.5, this is visualized as Figure 8.6.

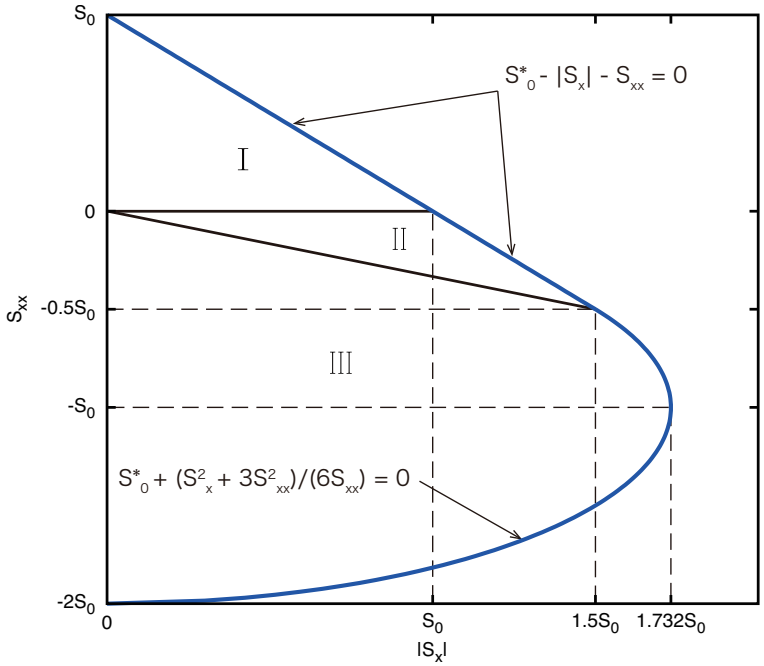

Figure8.6 Schematic to explain the flux limiter to avoid overshooting. The region in $(|S_x|, S_{xx})$ space that may set the upper bound on tracer distribution. The region bounded by the blue curve and the left vertical axis gives the tracer value that does not exceed the upper bound.

The modification for the moments corresponding to (8.139) is expressed as follows:

$$\begin{aligned} S'_x &= \min\left[3^{1/2}S_0^*,\ \max\left[-3^{1/2}S_0^*,\ S_x\right]\right], \\ S'_{xx} &= \begin{cases} \min\left[S_0^* - |S'_x|,\ \max\left[-S_0^* - \left(S_0^{*2} - \frac{S_x'^2}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } |S'_x| < \frac{3}{2}S_0^*, \\ \min\left[-S_0^* + \left(S_0^{*2} - \frac{S_x'^2}{3}\right)^{1/2},\ \max\left[-S_0^* - \left(S_0^{*2} - \frac{S_x'^2}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } |S'_x| \ge \frac{3}{2}S_0^*, \end{cases} \\ S'_{xy} &= \min\left[S_0^*,\ \max(-S_0^*,\ S_{xy})\right], \\ S'_{xz} &= \min\left[S_0^*,\ \max(-S_0^*,\ S_{xz})\right]. \end{aligned} \tag{8.142}$$

MH06 proposed three modification methods (Method A - C). Following COCO, MRI.COM adopts "Method B" of Morales Maqueda and Holloway (2006) as proposed by Merryfield and Holloway (2003), which lays emphasis on

monotonicity and suppresses numerical diffusion as well. In this method, the minimum and the maximum value among the three grid cells $(i-1, i, i+1)$ are set as the lower and the upper limit, respectively. Moments are adjusted using Eqs. (8.139) and (8.142). This method does not guarantee exact monotonicity, but improves monotonicity of the zeroth moment.

8.6.3 Calculating SOM advection in MRI.COM

It should be noted that the coordinate system is not Cartesian in ocean models. Since the coordinate system covers a spherical surface, the x direction in a grid cell is not identical to that in the adjacent cell, for instance. Thus, the exact conservation of moments cannot be realized. In addition, a tracer-cell including solid earth (sea floor or lateral boundary) is not a cuboid, so the orthogonal functions cannot be defined precisely for such a cell. Nevertheless, the procedures described in the previous subsection can be carried out using the volume of seawater in the non-cuboid grid cell, and the zeroth moment S_0 (total amount of the tracer) is conserved.

As indicated in the expressions in the previous subsections, the volume-integrated moments (S_i) and the fraction of volume to be removed (α) are used in the SOM advection scheme. There are 10 moments for each tracer. The fraction α is calculated using volume transports (`USTARL`, `VSTARL`, and `WLWL`), which are calculated in the subroutines `continuity__diagnose_horizontal` and `continuity__diagnose_vertical`. Following Prather (1986), the procedures in three directions are executed in order, not simultaneously. By default, the procedure in the meridional direction (`advec_y`) is called first, the zonal direction (`advec_x`) next, and lastly the vertical direction (`advec_z`). The order of operations in the horizontal direction may be flipped every time step if runtime option `lsomstrang` is set to be true (Strang splitting). This would improve the overall accuracy of the serially executed advection operations (Skamarock, 2006). The change in the tracer value caused by SOM advection is estimated in the subroutine `tracer__adv` and added to the variable `trcal` directly.

Usage

Model option `SOMADVEC` must be specified for compilation when the SOM advection scheme will be used for any tracer. Namelist `nml_somadv` is required (see Table 8.2) at run time. You must also specify which tracer will use this scheme as well as what kind of specifications will be used for that tracer. See Section 13.3.2 and Table 13.3 for details.

Table8.2 Variables defined in namelist `nml_somadv`

name	explanation	string or value
`file_som_in`	base name of the restart file to be input	{`file_som_in`}_i_n *
`file_som_out`	base name of the restart file to be output	{`file_som_out`}_i_n *
`nstep_som_restart_interval`	interval in time steps with which restart files are output	default is 0 (at the end of this job)
`lsomstrang`	The order of calling x and y direction is reversed every time step	`.true.` / `.false.`
`lsommonitor`	flag to monitor the conservation of the moments	`.true.` / `.false.`

* i={x, xx, y, yy, z, zz, xy, yz, zx}, n={01, 02, ..., numtrc}; tracer number

8.7 MPDATA scheme

8.7.1 Outline

This section explains the MPDATA scheme. In the MPDATA scheme (e.g., Smolarkiewicz and Margolin, 1998), advection for T is first solved by using the original volume transport, U^T, V^T, W^T, to obtain a temporary value ($T^{(1)}$) using the upstream scheme. Using this temporary value, an anti-diffusive volume transport ($U^{(1)}$, $V^{(1)}$, $W^{(1)}$) is computed. Here, the superscript $^{(1)}$ means that it is the first approximation to the error to be subtracted. This set of transports is used to compute a value of the next time step using the upstream scheme starting from the above temporary value.

Original dimensionless form of the anti-diffusive velocity given by Smolarkiewicz and Margolin (1998) is written as

follows (their equation 16):

$$\mathcal{U}^{(1)} \equiv \frac{u^{(1)}\delta t}{\delta x} = |\mathcal{U}|(1-\mathcal{U})A^{(1)} - \mathcal{U}\mathcal{V}B^{(1)} - \mathcal{U}\mathcal{W}C^{(1)}, \tag{8.143}$$

$$\mathcal{V}^{(1)} \equiv \frac{v^{(1)}\delta t}{\delta y} = |\mathcal{V}|(1-\mathcal{V})B^{(1)} - \mathcal{V}\mathcal{W}C^{(1)} - \mathcal{V}\mathcal{U}A^{(1)}, \tag{8.144}$$

$$\mathcal{W}^{(1)} \equiv \frac{w^{(1)}\delta t}{\delta z} = |\mathcal{W}|(1-\mathcal{W})C^{(1)} - \mathcal{W}\mathcal{U}A^{(1)} - \mathcal{W}\mathcal{V}B^{(1)}, \tag{8.145}$$

where $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{W}$ are the dimensionless velocity based on the original flow field and $A^{(1)}$, $B^{(1)}$, and $C^{(1)}$ are defined as

$$A^{(1)} \equiv \left[\frac{\delta x}{2T}\frac{\partial T}{\partial x}\right]^{(1)}, \tag{8.146}$$

$$B^{(1)} \equiv \left[\frac{\delta y}{2T}\frac{\partial T}{\partial y}\right]^{(1)}, \tag{8.147}$$

$$C^{(1)} \equiv \left[\frac{\delta z}{2T}\frac{\partial T}{\partial z}\right]^{(1)}. \tag{8.148}$$

In MRI.COM, we formulate them by using the set of original volume transports U^T, V^T, W^T. Considering the definition of the volume transport (8.7) to (8.9), the dimensionless form of volume transports can be written as

$$\mathcal{U}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{2U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\Delta t}{\Delta x_{i+\frac{1}{2}j}\Delta y_{i+\frac{1}{2},j}\left(\Delta z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right)}, \tag{8.149}$$

$$\mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{2V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\Delta t}{\Delta x_{i,j+\frac{1}{2}}\Delta y_{i,j+\frac{1}{2}}\left(\Delta z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right)}, \tag{8.150}$$

$$\mathcal{W}_{i,j,k} = \frac{W^T_{i,j,k}\Delta t}{\Delta z_{i,j,k}(\texttt{areat})_{i,j,k+\frac{1}{2}}}. \tag{8.151}$$

The $A^{(1)}$ through $C^{(1)}$ terms at $(i+\frac{1}{2}, j, k-\frac{1}{2})$, which are needed to calculate $U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}$, are

$$A^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta x_{i+\frac{1}{2},j}\delta_x T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}{2\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x} + \epsilon} \tag{8.152}$$

$$B^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta y_{i+\frac{1}{2},j}\delta_y T^{(1)}_{i+\frac{1}{2},j}}{2\overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{y} + \epsilon} \tag{8.153}$$

$$C^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta z_{i+\frac{1}{2},j}\delta_z T^{(1)}_{i+\frac{1}{2},j}}{2\overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{z} + \epsilon} \tag{8.154}$$

where ϵ is a small number that prevents zero division when tracer values reach zero. The finite difference and averaging operators are defined as follows (definitions in y and z directions are the same):

$$\delta_x A_i \equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_i}, \quad \delta_x A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_i}{\Delta x_{i+\frac{1}{2}}},$$

$$\overline{A_i}^{x} \equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \quad \overline{A_{i+\frac{1}{2}}}^{x} \equiv \frac{A_{i+1} + A_i}{2}. \tag{8.155}$$

The $A^{(1)}$ through $C^{(1)}$ terms should be estimated at $(i, j+\frac{1}{2}, k-\frac{1}{2})$ for $V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}}$ and (i, j, k) for $W^{(1)}_{i,j,k}$.

Then anti-diffusive volume transport is written as

$$U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|(1-\mathcal{U}_{i+\frac{1}{2},j,k-\frac{1}{2}})A^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} - U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\overline{\overline{\mathcal{V}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{y} B^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} - U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\overline{\overline{\mathcal{W}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{z} C^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{8.156}$$

$$V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} = \left|V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\right|(1-\mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}})B^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} - V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\overline{\overline{\mathcal{U}_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{x}}^{y} A^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} - V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\overline{\overline{\mathcal{W}_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{y}}^{z} C^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{8.157}$$

$$W^{(1)}_{i,j,k} = \left|W^T_{i,j,k}\right|(1-\mathcal{W}_{i,j,k})C^{(1)}_{i,j,k} - W^T_{i,j,k}\overline{\overline{\mathcal{U}_{i+\frac{1}{2},j,k}}^{x}}^{z} A^{(1)}_{i,j,k} - W^T_{i,j,k}\overline{\overline{\mathcal{V}_{i+\frac{1}{2},j,k}}^{y}}^{z} B^{(1)}_{i,j,k}. \tag{8.158}$$

8.7.2 Gauge transformation and flux limiter

Original MPDATA scheme is positive definite and cannot be used for tracers that take negative value. This can be avoided by adding some constant for the temporary value, $T^{(1)}$.

The monotonicity of MPDATA scheme can be obtained by using a non-oscillatory option proposed by Smolarkiewicz and Grabowski (1990). In this option, an anti-diffusive volume transport has the upper limit given as follows:

$$\left[U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i+1,j,k-\frac{1}{2}})U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\ \min(1, \beta^{out}_{i+1,j,k-\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0, \end{cases} \tag{8.159}$$

$$\left[V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i,j+1,k-\frac{1}{2}})V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} & \text{if} \quad V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} > 0, \\ \min(1, \beta^{out}_{i,j+1,k-\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})V^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} < 0, \end{cases} \tag{8.160}$$

$$\left[W^{(1)}_{i,j,k}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k+\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})W^{(1)}_{i,j,k} & \text{if} \quad W^{(1)}_{i,j,k} > 0, \\ \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i,j,k+\frac{1}{2}})W^{(1)}_{i,j,k} & \text{if} \quad W^{(1)}_{i,j,k} < 0, \end{cases} \tag{8.161}$$

where

$$\beta^{in}_{i,j,k-\frac{1}{2}} \equiv \frac{T^{max}_{i,j,k-\frac{1}{2}} - T^{(1)}_{i,j,k-\frac{1}{2}}}{A^{in}_{i,j,k-\frac{1}{2}} + \epsilon}, \tag{8.162}$$

$$\beta^{out}_{i,j,k-\frac{1}{2}} \equiv \frac{T^{(1)}_{i,j,k-\frac{1}{2}} - T^{min}_{i,j,k-\frac{1}{2}}}{A^{out}_{i,j,k-\frac{1}{2}} + \epsilon}, \tag{8.163}$$

$$\begin{aligned} T^{max}_{i,j,k-\frac{1}{2}} = \max\Big(& T_{i,j,k-\frac{1}{2}}, T_{i+1,j,k-\frac{1}{2}}, T_{i-1,j,k-\frac{1}{2}}, T_{i,j+1,k-\frac{1}{2}}, T_{i,j-1,k-\frac{1}{2}}, T_{i,j,k+\frac{1}{2}}, T_{i,j,k-\frac{3}{2}}, \\ & T^{(1)}_{i,j,k-\frac{1}{2}}, T^{(1)}_{i+1,j,k-\frac{1}{2}}, T^{(1)}_{i-1,j,k-\frac{1}{2}}, T^{(1)}_{i,j+1,k-\frac{1}{2}}, T^{(1)}_{i,j-1,k-\frac{1}{2}}, T^{(1)}_{i,j,k+\frac{1}{2}}, T^{(1)}_{i,j,k-\frac{3}{2}}\Big), \end{aligned} \tag{8.164}$$

$$\begin{aligned} T^{min}_{i,j,k-\frac{1}{2}} = \min\Big(& T_{i,j,k-\frac{1}{2}}, T_{i+1,j,k-\frac{1}{2}}, T_{i-1,j,k-\frac{1}{2}}, T_{i,j+1,k-\frac{1}{2}}, T_{i,j-1,k-\frac{1}{2}}, T_{i,j,k+\frac{1}{2}}, T_{i,j,k-\frac{3}{2}}, \\ & T^{(1)}_{i,j,k-\frac{1}{2}}, T^{(1)}_{i+1,j,k-\frac{1}{2}}, T^{(1)}_{i-1,j,k-\frac{1}{2}}, T^{(1)}_{i,j+1,k-\frac{1}{2}}, T^{(1)}_{i,j-1,k-\frac{1}{2}}, T^{(1)}_{i,j,k+\frac{1}{2}}, T^{(1)}_{i,j,k-\frac{3}{2}}\Big), \end{aligned} \tag{8.165}$$

and A^{in} and A^{out} are the absolute values of the total incoming and outgoing advection flux at the T-box.

8.7.3 Usage

Model option `MPDATAADVEC` must be specified for compilation when MPDATA will be used for any tracer. At run time, you must specify which tracer will use this scheme as well as what kind of specifications will be used for that tracer. See Section 13.3.2 and Table 13.3 for details.

Chapter 9

SGS parameterization of lateral mixing of tracers

This chapter explains subgrid-scale parameterizations for horizontal mixing of tracers.

9.1 Introduction and Formulation

Historically, a harmonic diffusion operator is applied in each direction of the model coordinates to express mixing of tracers. In the real ocean, transport and mixing would occur dominantly along neutral (isopycnal) surfaces. Thus, horizontal mixing along a constant depth surface is generally inappropriate since neutral surfaces are generally slanting relative to a constant depth surface. Neutral physics schemes are devised as substitutes for the harmonic scheme in the horizontal direction, while the harmonic scheme continues to be used for vertical diffusion.

By default, the diffusion operator mixes a tracer in each direction of the model coordinates with the harmonic scheme. For horizontal diffusion, the biharmonic scheme can be used instead of the harmonic scheme. Using (20.22) and (20.23), the harmonic-type diffusivity is represented for any tracer (T) as follows:

$$
\begin{aligned}
-z_s \nabla \cdot \mathbf{F}_T = \mathcal{D}(T) &= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial \mu}\left(\frac{h_\psi z_s \kappa_H}{h_\mu}\frac{\partial T}{\partial \mu}\right) + \frac{\partial}{\partial \psi}\left(\frac{h_\mu z_s \kappa_H}{h_\psi}\frac{\partial T}{\partial \psi}\right)\right\} + \frac{\partial}{\partial s}\left(\frac{\kappa_V}{z_s}\frac{\partial T}{\partial s}\right) \\
&= -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial (h_\psi z_s F^T_\mu)}{\partial \mu} + \frac{\partial (h_\mu z_s F^T_\psi)}{\partial \psi}\right\} - \frac{\partial F^T_z}{\partial s},
\end{aligned}
\tag{9.1}
$$

where κ_H and κ_V are the horizontal and vertical diffusion coefficients and the diffusive fluxes are represented by F. $\mathcal{D}(T)$ represents the diffusion operator for T. When the biharmonic-type is selected for horizontal diffusion, the above Laplacian operation is repeated twice reversing its sign after the first operation.

When the neutral physics schemes are selected, the advection-diffusion equation for any tracer (T) is expressed as follows (Gent and McWilliams, 1990):

$$
\frac{DT}{Dt} + \nabla_H \cdot \left[T\frac{\partial}{\partial z}(\kappa_T \nabla_H \rho/\rho_z)\right] + \frac{\partial}{\partial z}\left[T\nabla_H \cdot (-\kappa_T \nabla_H \rho/\rho_z)\right] = \mathcal{D}(T) + Q^T, \tag{9.2}
$$

where the first term on the r.h.s. is the isopycnal diffusion, whose form is given by

$$
\mathcal{D}(T) = \nabla \cdot (\kappa_I \mathbf{K} \nabla T), \tag{9.3}
$$

where

$$
\mathbf{K} = \frac{1}{\rho_x^2 + \rho_y^2 + \rho_z^2}\begin{pmatrix} \rho_y^2 + \rho_z^2 & -\rho_x\rho_y & -\rho_x\rho_z \\ -\rho_x\rho_y & \rho_x^2 + \rho_z^2 & -\rho_y\rho_z \\ -\rho_x\rho_z & -\rho_y\rho_z & \rho_x^2 + \rho_y^2 \end{pmatrix} \tag{9.4}
$$

$$
= \frac{1}{1 + (\rho_x/\rho_z)^2 + (\rho_y/\rho_z)^2}\begin{pmatrix} 1 + (\rho_y/\rho_z)^2 & -(\rho_x/\rho_z)(\rho_y/\rho_z) & -\rho_x/\rho_z \\ -(\rho_x/\rho_z)(\rho_y/\rho_z) & 1 + (\rho_x/\rho_z)^2 & -\rho_y/\rho_z \\ -\rho_x/\rho_z & -\rho_y/\rho_z & (\rho_x/\rho_z)^2 + (\rho_y/\rho_z)^2 \end{pmatrix}, \tag{9.5}
$$

(Redi, 1982). In the above, the Cartesian notation is used for simplicity. The subscript x represents $\partial/(h_\mu \partial\mu)$ y represents $\partial/(h_\psi \partial\psi)$, and z represents $\partial/(z_s \partial s)$, The isopycnal diffusion coefficient is represented by κ_I. Diapycnal diffusion is not considered here.

The second and third terms on the l.h.s. of (9.2) have the form of advection terms with a transport velocity vector (u_T, v_T, w_T):

$$u_T \equiv \frac{\partial}{\partial z}\Big(\kappa_T \frac{1}{h_\mu}\frac{\partial \rho}{\partial \mu}\Big/\frac{\partial \rho}{\partial z}\Big), \tag{9.6}$$

$$v_T \equiv \frac{\partial}{\partial z}\Big(\kappa_T \frac{1}{h_\psi}\frac{\partial \rho}{\partial \psi}\Big/\frac{\partial \rho}{\partial z}\Big), \tag{9.7}$$

$$w_T \equiv -\frac{1}{h_\mu h_\psi}\Big\{\frac{\partial}{\partial \mu}\Big(\kappa_T \frac{h_\psi}{h_\mu}\frac{\partial \rho}{\partial \mu}\Big/\frac{\partial \rho}{\partial z}\Big) + \frac{\partial}{\partial \psi}\Big(\kappa_T \frac{h_\mu}{h_\psi}\frac{\partial \rho}{\partial \psi}\Big/\frac{\partial \rho}{\partial z}\Big)\Big\}, \tag{9.8}$$

(Gent and McWilliams, 1990). This velocity can be understood as the advection caused by the thickness diffusion of an isopycnal layer, where κ_T is often referred to as thickness diffusivity.

Note that these could be rewritten as

$$\mathcal{G}(T) = \nabla \cdot (\kappa_T \mathbf{A} \nabla T) \tag{9.9}$$

with

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\rho_x/\rho_z \\ 0 & 0 & -\rho_y/\rho_z \\ \rho_x/\rho_z & \rho_y/\rho_z & 0 \end{pmatrix}. \tag{9.10}$$

Comparing with (9.5), we notice that the isopycnal diffusion and the thickness diffusion terms are combined to yield a concise form (Griffies, 1998) and (9.2) can be rewritten as:

$$\frac{DT}{Dt} = \nabla \cdot \{(\kappa_I \mathbf{K} - \kappa_T \mathbf{A})\nabla T\} + Q^T. \tag{9.11}$$

Three types of horizontal diffusion, harmonic horizontal diffusion (default), biharmonic horizontal diffusion (`TRCBIHARM` option), and isopycnal diffusion (`ISOPYCNAL` option), are available in MRI.COM. When isopycnal diffusion (Redi, 1982) is selected, the parameterization scheme for eddy induced advection by Gent and McWilliams (1990) (GM scheme) is used with it (see Section 9.4 for details of this parameterization).

The following is a guide to selecting a horizontal diffusion scheme. Biharmonic diffusion is appropriate for a high resolution model that can resolve eddies because it is more scale-selective than harmonic diffusion and does not unnecessarily suppress disturbances in resolved scales. On the other hand, biharmonic diffusion is not recommended in eddy-less models because this would result in numerical instability. Harmonic horizontal diffusion is also not recommended because this scheme would cause unrealistic cross-isopycnal (diapycnal) mixing as mentioned above. Instead, both isopycnal diffusion and the GM scheme should be used there. Using an anisotropic GM scheme can maintain the meso-scale eddy structures and swift currents by restricting the direction of diffusion, and thus may be usable even for a high resolution model.

The finite difference expression of (9.11) is given by taking the finite volume approach as follows:

$$\begin{aligned} T^{n+1}_{i,j,k-\frac{1}{2}}\Delta V^{n+1}_{i,j,k-\frac{1}{2}} = {} & T^{n-1}_{i,j,k-\frac{1}{2}}\Delta V^{n-1}_{i,j,k-\frac{1}{2}} + 2\,\Delta t\,\{FXD_{i-\frac{1}{2},j,k-\frac{1}{2}} - FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} + FYD_{i,j-\frac{1}{2},k-\frac{1}{2}} - FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} \\ & + FZD_{i,j,k} - FZD_{i,j,k-1}\} + (\text{other terms}). \end{aligned} \tag{9.12}$$

Usually, fluxes due to diffusion are explicitly represented using a starting time level of the temporal discretization. However, when the flux is very large relative to the grid size and the time step interval, which would often occur for vertical fluxes, an implicit scheme is used. These issues are discussed in Chapter 19.

9.2 Horizontal diffusion

9.2.1 Laplacian diffusion

Harmonic horizontal diffusion assumes that the diffusion flux is a product of the gradient of tracer and the diffusion coefficient (κ_H). The finite difference expressions for the fluxes are given as follows:

$$FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} = -\kappa_H \Delta y_{i+\frac{1}{2},j}\overline{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{\,y}\delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{9.13}$$

$$FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} = -\kappa_H \Delta x_{i,j+\frac{1}{2}}\overline{\Delta z_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{\,x}\delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{9.14}$$

where

$$\delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}} \equiv \frac{T_{i+1,j,k-\frac{1}{2}} - T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j}}, \tag{9.15}$$

$$\delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}} \equiv \frac{T_{i,j+1,k-\frac{1}{2}} - T_{i,j,k-\frac{1}{2}}}{\Delta y_{i,j+\frac{1}{2}}}. \tag{9.16}$$

9.2.2 Biharmonic diffusion

Biharmonic horizontal diffusion (`TRCBIHARM` option) assumes that the diffusion flux is proportional to the gradient of the Laplacian of tracer. The finite difference expressions for the fluxes are given as follows:

$$FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} = \sqrt{\kappa_b}\,\Delta y_{i+\frac{1}{2},j}\,\overline{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{\,y}\,\delta_x[\mathrm{Lap}(T)]_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{9.17}$$

$$FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} = \sqrt{\kappa_b}\,\Delta x_{i,j+\frac{1}{2}}\,\overline{\Delta z_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{\,x}\,\delta_y[\mathrm{Lap}(T)]_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{9.18}$$

where κ_b is diffusivity coefficient and

$$\mathrm{Lap}(T)_{i,j,k-\frac{1}{2}} = \frac{\Delta x_{i,j}\Delta y_{i,j}}{\Delta V_{i,j,k-\frac{1}{2}}}(\delta_x\sqrt{\kappa_b}\,\overline{\Delta z_{i,j,k-\frac{1}{2}}}^{\,y}\,\delta_x T_{i,j,k-\frac{1}{2}} + \delta_y\sqrt{\kappa_b}\,\overline{\Delta z_{i,j,k-\frac{1}{2}}}^{\,x}\,\delta_y T_{i,j,k-\frac{1}{2}}). \tag{9.19}$$

9.2.3 Specification of coefficient

The diffusion coefficient of horizontal diffusion is specified using the namelists listed on Tables 9.1 and 9.2. This must be zero if `ISOPYCNAL` option is selected, unless the horizontal diffusion is applied intentionally.

Table9.1 Namelist `nml_tracer_diff_horz`

variable name	units	description	usage
`diff_horz_cm2ps`	$\mathrm{cm^2\ s^{-1}}$	Laplacian diffusion coefficient (κ_H)	if not `TRCBIHARM`
`diff_horz_cm4ps`	$\mathrm{cm^4\ s^{-1}}$	Biharmonic diffusion coefficient (κ_b)	if `TRCBIHARM`
`file_diff_horz_2d`	character	2D distribution of diffusion coefficient is read from this file	

Table9.2 Namelist `nml_grid_size_change_mix_coefs`

variable name	units	description	usage
`l_grid_size_change_mix_coefs`	logical	the given coefficient is multiplied by the fraction of the local grid size to 100 km.	

9.3 Isopycnal diffusion

In isopycnal diffusion, the diffusion flux is expressed by high diffusivity along the isopycnal surface κ_I, low diffusivity across the isopycnal surface κ_D, and their product with the gradient of tracer in each direction. Using diffusion tensor $\mathbf{K}$, each flux component is written as $F^m(T) = -K^{mn}\partial_n T$, and then

$$\mathbf{K} = \frac{\kappa_I}{1+S^2}\begin{pmatrix} 1+S_y^2+\epsilon S_x^2 & (\epsilon-1)S_xS_y & (1-\epsilon)S_x \\ (\epsilon-1)S_xS_y & 1+S_x^2+\epsilon S_y^2 & (1-\epsilon)S_y \\ (1-\epsilon)S_x & (1-\epsilon)S_y & \epsilon+S^2 \end{pmatrix}, \tag{9.20}$$

where

$$\mathbf{S} = (S_x, S_y, 0) = \left(-\frac{\frac{\partial\rho}{\partial x}}{\frac{\partial\rho}{\partial z}}, -\frac{\frac{\partial\rho}{\partial y}}{\frac{\partial\rho}{\partial z}}, 0\right), \tag{9.21}$$

$$S = |\,\mathbf{S}\,|, \tag{9.22}$$

$$\epsilon = \frac{\kappa_D}{\kappa_I} \tag{9.23}$$

(Redi, 1982).

The finite difference method is based on Cox (1987) except for the small isopycnal slope approximation. The finite difference form of three components of the gradient of tracer in calculating the east-west component of flux $FXD_{i+\frac{1}{2},j,k-\frac{1}{2}}$ is given as follows (see Figure 9.1):

$$(\delta_x T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{9.24}$$

$$(\delta_y T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \overline{\delta_y T_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{xy}, \tag{9.25}$$

$$(\delta_z T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \overline{\delta_z T_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{xz}. \tag{9.26}$$

Similarly, the north-south component $FYD_{i,j+\frac{1}{2},k-\frac{1}{2}}$ is given by

$$(\delta_x T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \overline{\delta_x T_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{xy}, \tag{9.27}$$

$$(\delta_y T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{9.28}$$

$$(\delta_z T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \overline{\delta_z T_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{yz}, \tag{9.29}$$

and the vertical component $FZD_{i,j,k}$ is given by

$$(\delta_x T)_{i,j,k} = \overline{\delta_x T_{i,j,k}}^{xz}, \tag{9.30}$$

$$(\delta_y T)_{i,j,k} = \overline{\delta_y T_{i,j,k}}^{yz}, \tag{9.31}$$

$$(\delta_z T)_{i,j,k} = \delta_z T_{i,j,k}. \tag{9.32}$$

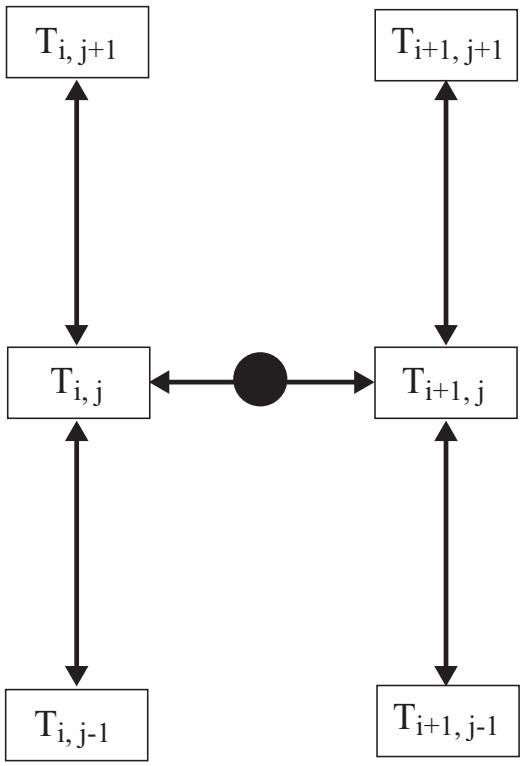

Figure9.1 The way of calculating the gradient at the circle $(i+\frac{1}{2}, j, k-\frac{1}{2})$ in isopycnal diffusion: the east-west gradient is indicated by an arrow through the circle, and the north-south gradient is given by averaging four arrows in the vertical direction.

The density gradient in the calculation of each component of the diffusion tensor can be obtained by replacing T in the above equations with ρ. However, density is calculated at the reference level $k-\frac{1}{2}$ for the east-west and north-south components, and at the reference level k for the vertical component.

In addition, the upper bound on the isopycnal slope $S_{\max}$ (`slope_clip_iso` in namelist `nml_isopy_slope_clip`, Table 9.4) is set because a nearly vertical isopycnal slope and the resultant low horizontal diffusivity could cause numerical instability. If $|\mathbf{S}| > S_{\max}$, $\partial_z \rho$ in all flux components is replaced so as to satisfy $|\mathbf{S}| = S_{\max}$.

The vertical flux due to the third diagonal component of the diffusion tensor (9.20) is

$$FZD_{i,j,k} = (\texttt{areat})_{i,j,k} \frac{\kappa_I(\epsilon + S^2)}{1 + S^2} \delta_z T_{i,j,k}. \tag{9.33}$$

Thus the effective vertical diffusivity $\kappa_{\rm eff}$ is

$$\kappa_{\rm eff} = \frac{\kappa_D + \kappa_I S^2}{1 + S^2}. \tag{9.34}$$

For a steep isopycnal slope $S \sim 1/100$ and a canonical value of isopycnal diffusion coefficient $\kappa_I \sim 10^7\,{\rm cm^2\,s^{-1}}$ and a typical value of diapycnal diffusion coefficient $\kappa_D \sim 10^{-1}\,{\rm cm^2\,s^{-1}}$,

$$\kappa_{\rm eff} \sim 10^3\,{\rm cm^2\,s^{-1}}. \tag{9.35}$$

This is a fairly large value which warrants use of an implicit scheme (Section 19.5). In MRI.COM, this term is separated from other terms and solved with other vertical diffusion terms using an implicit method.

Griffies et al. (1998) noted a problem in the finite difference expression of the isopycnal diffusion as implemented in the GFDL-model by Cox (1987). The problem is that the down-gradient orientation of the diffusive fluxes along the neutral directions does not necessarily guarantee the zero isoneutral diffusive flux of locally referenced density (e.g., potential temperature when it is the only active tracer). This is caused by the nature of the finite difference method and the non-linearity of the equation of state. Griffies et al. (1998) proposed a remedy, but this remains to be implemented in MRI.COM.

9.3.1 Tapering around steep isopycnal slopes

As noted above, we set an upper bound on the isopycnal slope used to evaluate isopycnal tracer diffusion terms in MRI.COM in order to prevent numerical instability around steep isopycnal slopes. Griffies (2004) shows that such slope clipping could lead to an unrealistically large tracer flux. Danabasoglu and McWilliams (1995) propose another method that uses a hyperbolic tangent to exponentially taper isopycnal diffusion in steep slope regions. The following factor is multiplied to the isopycnal diffusion coefficient there:

$$f_{\rm steep} = \frac{1}{2}\left\{1 + \tanh\left(\frac{S_{\rm center} - |\mathbf{S}|}{S_{\rm width}}\right)\right\}. \tag{9.36}$$

Two parameters $S_{\rm center}$ and $S_{\rm width}$, which determine a transitional region, are given by `center_transition` and `width_transition` in namelist `nml_tracer_diff_isopy_taper`, respectively (Table 9.5). This tapering method is valid when `ISOTAPER` option is chosen and used with the other factor $f_{\rm surface}$ introduced in the subsequent section.

9.3.2 Surface tapering

The vertical displacement of a water parcel due to mesoscale eddy stirring (D) is approximately calculated as follows:

$$D = R|\mathbf{S}|, \tag{9.37}$$

where R is the internal deformation radius and $|\mathbf{S}|$ is the isopycnal slope. If the depth of the water parcel (d) is shallower than D, the boundary constrains its displacement and eddy diffusive fluxes (Griffies, 2004). The `ISOTAPER` option enables the use of the following tapering scheme proposed by Large et al. (1997):

$$f_{\rm surface} = \frac{1}{2}\left\{1 + \sin\pi\left(r - \frac{1}{2}\right)\right\}, \tag{9.38}$$

where $r = \max(0, \min(1, d/D))$. Eddy diffusivity is tapered off to zero toward the sea surface in the region where $0 \le d \le D$. In MRI.COM, the reference level of the depth d is set at the boundary layer depth (BLD): $d = -z - {\rm BLD}$. In MRI.COM, the surface mixed layer depth (MLD) is treated as the BLD (see also the next section). This means that the eddy diffusivity is tapered to zero within the mixed layer. The upper boundary of this tapering region can be changed to a constant level by setting `upper_level_isotaper_m` in namelist `nml_tracer_diff_isopy_taper` (Table 9.5).

By default, the horizontal diagonal terms of the isopycnal diffusion tensor (9.20) are not multiplied by the two factors noted above. This means that the isopycnal diffusion is rendered the horizontal diffusion within the boundary layer and in the steep slope region. One may apply these tapering methods to the horizontal diagonal terms by setting `l_apply_hdiag` = `.true.` in namelist `nml_tracer_diff_isopy_taper` (Table 9.5).

9.3.3 Specification of coefficient

The diffusion coefficients of isopycnal diffusion and GM parameterization explained in the next section are specified using the namelist listed on Table 9.3. We can use different slope maximal limits for isopycnal diffusion and GM parameterization that are specified using the namelist listed on Table 9.4. Configurations of surface tapering for the isopycnal diffusion scheme with `ISOTAPER` option are specified using the namelist listed on Table 9.5.

Table9.3 Namelist `nml_tracer_diff_isopy`

variable name	units	description	usage
`diff_isopy_cm2ps`	$cm^2\ sec^{-1}$	isopycnal diffusion coefficient	if `ISOPYCNAL`
`diff_diapy_cm2ps`	$cm^2\ sec^{-1}$	diapycnal diffusion coefficient	if `ISOPYCNAL`
`diff_thick_cm2ps`	$cm^2\ sec^{-1}$	coefficient of GM parameterization	if `ISOPYCNAL`

Table9.4 Namelist `nml_isopy_slope_clip`

variable name	units	description	usage
`slope_clip_iso`	1	maximum slope of isopycnal surface for isopycnal diffusion	if `ISOPYCNAL`
`slope_clip_gm`	1	maximum slope of isopycnal surface for GM parameterization	if `ISOPYCNAL`

Table9.5 Namelist `nml_tracer_diff_isopy_taper`

variable name	units	description	usage
`l_apply_hdiag`	logical	apply tapering factors to horizontal diagonal terms (default = `.false.`)	if `ISOTAPER`
`center_transition`	1	center of the transition region of tapering with hyperbolic tangent (default = 0.005)	if `ISOTAPER`
`width_transition`	1	width of the transition region of tapering with hyperbolic tangent (default = 0.001)	if `ISOTAPER`
`ai_min`	$cm^2\ sec^{-1}$	lower limit of horizontal isopycnal diffusion coefficient when `l_apply_hdiag` = `.true.` (default = 0.0)	if `ISOTAPER`
`upper_level_isotaper_m`	m	BLD for the sine taper (default = mixed layer depth (−1.0))	if `ISOTAPER`
`l_explicit_vdif`	logical	handle the vertical diffusion term explicitly (default = `.false.`)	if `ISOTAPER`

9.4 Gent and McWilliams parameterization for eddy-induced transport

9.4.1 General features

The Gent and McWilliams (1990) parameterization represents transports of tracers due to disturbances smaller than the grid size, assuming that a flux proportional to the gradient of the layer thickness exists along the isopycnal surface. The isopycnal diffusion stated above does not produce any flux when the isopycnal surface coincides with the isotherm and isohaline surface. This parameterization, however, produces fluxes in such a case, and acts to decrease the isopycnal slope.

Flux convergence due to diffusion is expressed as follows:

$$R(T) = \partial_m (J^{mn} \partial_n T) \tag{9.39}$$

Diffusion tensor J^{mn} is expressed as the sum of the symmetric component $K^{mn} = (J^{mn} + J^{nm})/2$ and the anti-symmetric component $A^{mn} = (J^{mn} - J^{nm})/2$. Isopycnal diffusion has the form of a symmetric diffusion tensor. Convergence of a

skew flux caused by the anti-symmetric component $F^m_{\rm skew} = -A^{mn}\partial_n T$ is as follows:

$$\begin{aligned} R_A(T) &= \partial_m(A^{mn}\partial_n T) \\ &= \partial_m(A^{mn})\partial_n T \\ &= \partial_n(\partial_m A^{mn} T), \end{aligned} \tag{9.40}$$

where $A^{mn}\partial_m\partial_n T = 0$ and $\partial_m\partial_n A^{mn} = 0$ are used. If we set a virtual velocity $u^n_* \equiv -\partial_m A^{mn}$, then the flux due to the anti-symmetric component could be regarded as the advection due to this virtual velocity $\mathbf{u}_*$. In this case, the flux is $\mathbf{F}_{\rm adv} = \mathbf{u}_* T$ and $R_A(T) = -\mathbf{u}_* \cdot \nabla T$ since $\mathbf{u}_*$ is divergence free by definition.

The Gent and McWilliams parameterization for eddy-induced transport velocity is given by

$$\mathbf{u}_* = -\frac{\partial}{\partial\rho}(\kappa_{\rm GM}\nabla_\rho h) \Big/ \frac{\partial h}{\partial\rho}, \tag{9.41}$$

where h is the depth of the neutral surface (ρ = const). This velocity is expressed in the depth coordinate as

$$\mathbf{u}_* = \left(-\partial_z(\kappa_{\rm GM}S_x),\ -\partial_z(\kappa_{\rm GM}S_y),\ \nabla_h\cdot(\kappa_{\rm GM}\mathbf{S})\right) \tag{9.42}$$

where

$$\mathbf{S} = (S_x, S_y, 0) = \left(-\rho_x/\rho_z, -\rho_y/\rho_z, 0\right). \tag{9.43}$$

(Gent et al., 1995).

Griffies (1998) showed that the tendency of a tracer due to this parameterization might be expressed using an anti-symmetric diffusion tensor $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\kappa_{\rm GM}S_x \\ 0 & 0 & -\kappa_{\rm GM}S_y \\ \kappa_{\rm GM}S_x & \kappa_{\rm GM}S_y & 0 \end{pmatrix}, \tag{9.44}$$

so that

$$\frac{\partial T}{\partial t} = \cdots + \nabla\cdot(\mathbf{A}\nabla T). \tag{9.45}$$

The flux due to advection can be expressed using a vector streamfunction,

$$\Psi = \kappa_{\rm GM}\hat{\mathbf{z}}\times\mathbf{S} = (-\kappa_{\rm GM}S_y,\ \kappa_{\rm GM}S_x,\ 0), \tag{9.46}$$

which produces $\mathbf{u}_*$ in (9.42):

$$\mathbf{F}_{\rm adv} = \mathbf{u}_* T = T(\nabla\times\Psi).$$

The skew diffusive expression for the flux using tensor $\mathbf{A}$ in (9.44) is

$$\mathbf{F}_{\rm skew} = -\mathbf{A}\nabla T = -(\nabla T)\times\Psi = \mathbf{F}_{\rm adv} - \nabla\times(T\Psi).$$

Thus, the convergence of the flux expressed in tensorial form matches that of the advective expression. In other words, the Gent and McWilliams parameterization is realized by only adding $\mathbf{A}$ to the tensor of the isopycnal diffusion $\mathbf{K}$ (Griffies, 1998).

9.4.2 Dependency of coefficient on space and time

By default, the diffusivity coefficient for the Gent-McWilliams parameterization is constant both in time and space, whose value, $[\kappa_{\rm GM}]_{\rm ref}$, is given by `diff_thick_cm2ps` in namelist `nml_tracer_diff_isopy` (Table 9.3). However, it may be dependent on local horizontal grid size by specifying a namelist (see the next paragraph). Several parameterization may be used by choosing `GMVAR` option. User should specify one of `l_visbeck`, `l_eden`, `l_danabasoglu` to be `.true.` in namelist `nml_gmvar_select` (Table 9.8).

a. Simple scheme

If `l_grid_size_change_mix_coefs` = `.true.` in namelist `nml_grid_size_change_mix_coefs` (Table 9.7), the coefficient may be dependent on the horizontal grid size according to the following formula

$$\kappa_{\rm GM} = [\kappa_{\rm GM}]_{\rm ref} \times \min(\Delta x, \Delta y)/(100\,{\rm km}), \tag{9.47}$$

where Δx and Δy are local zonal and meridional grid sizes of a U-cell, respectively.

b. Visbeck et al. (1997)

To use the method proposed by Visbeck et al. (1997), `l_visbeck` = `.true.` in namelist `nml_gmvar_select`.

Visbeck et al. (1997) proposed to give the GM coefficient κ_{GM} as

$$\kappa_{\mathrm{GM}} = \alpha \frac{M^2}{N} l^2, \tag{9.48}$$

where $\alpha = 0.015$,

$$M^2 = \frac{g}{\rho_0} \left| \nabla_H \rho \right|, \qquad N^2 = -\frac{g}{\rho_0} \frac{\partial \rho}{\partial z},$$

and l the horizontal length scale of the baroclinic zone, g acceleration of gravity, ρ_0 reference density.

Specifically in our model,

$$M^2 = \frac{g}{\rho_0 D} \left[\left(\int_{-H_1}^{-H_0} \frac{\partial \rho}{\partial x} dz \right)^2 + \left(\int_{-H_1}^{-H_0} \frac{\partial \rho}{\partial y} dz \right)^2 \right]^{\frac{1}{2}}, \tag{9.49}$$

and

$$N^2 = \frac{g \left[\sigma_0(H_1) - \sigma_0(H_0) \right]}{\rho_0 D}, \tag{9.50}$$

where $H_0 = 100$ m, $H_1 = 2000$ m, $D = H_1 - H_0$, and σ_0 is the potential density. Lower limit for N is set so that $N^2 \geq 10^{-9}\,\mathrm{s}^{-1}$.

Using the following formula for the phase speed of the 1st baroclinic mode gravity wave (Sueyoshi and Yasuda, 2009)

$$c_1 = \frac{1}{\pi} \int_{-H_B}^{0} \left(-\frac{g}{\rho_0} \frac{\partial \sigma_0}{\partial z} \right)^{\frac{1}{2}} dz, \tag{9.51}$$

where H_B is the depth of sea floor, deformation radius λ_1 is calculated as follows:

$$\lambda_1 = \min \left(\left| \frac{c_1}{f} \right|, 4 \times 10^4 \right) \quad \mathrm{m}. \tag{9.52}$$

Using a factor $r = 7$, GM coefficient is determined as follows:

$$\kappa_{\mathrm{GM}} = \alpha \frac{M^2}{N} \left(r \lambda_1 \right)^2. \tag{9.53}$$

Lower and upper limits for the coefficient are set as follows:

$$300 \leq \kappa_{\mathrm{GM}} \leq 1500 \quad \mathrm{m}^2\,\mathrm{s}^{-1}. \tag{9.54}$$

c. Eden and Greatbatch (2008)

To use the method proposed by Eden and Greatbatch (2008), `l_eden` = `.true.` in namelist `nml_gmvar_select`.

Eden and Greatbatch (2008) and Eden et al. (2009) proposed that the thickness diffusivity is given by

$$\kappa_{\mathrm{GM}} = c L^2 \sigma \tag{9.55}$$

The eddy length scale L is given as the minimum of the Rossby radius L_r and Rhines scale L_{Rhi}. This choice for L was found to be consistent with independent estimates of eddy length scales from satellite observations and high-resolution model results (Eden, 2007) and with recent theoretical considerations (Theiss, 2004). L_{Rhi} is estimated from variables of the coarse resolution model as

$$L_{Rhi} = \frac{\sigma}{\beta} \tag{9.56}$$

(Eden and Greatbatch, 2008), while L_r is given by

$$L_r = \min \left[\frac{c_1}{|f|}, \sqrt{\frac{c_1}{2\beta}} \right], \tag{9.57}$$

where c_1 denotes the 1st baroclinic gravity wave speed calculated approximately as eq. (9.51). Considering the thermal wind relation in mid-latitudes, Eden and Greatbatch (2008) proposed that the inverse eddy time scale σ is given by

$$\sigma = f(\mathrm{Ri} + \gamma)^{-\frac{1}{2}}. \tag{9.58}$$

Here, $\mathrm{Ri} = N^2|\frac{\partial}{\partial z}u_h|^{-2}$ denotes the local Richardson number. $\gamma(>0)$ is introduced to prevent the singularity as $N \to 0$, which acts effectively as an upper limit for σ and consequently for κ_{GM}. The default values of γ and c in eq. (9.55), are 200 and 2, respectively.

d. Danabasoglu and Marshall (2007)

To use the method proposed by Danabasoglu and Marshall (2007), `l_danabasoglu = .true.` in namelist `nml_gmvar_select`.

Danabasoglu and Marshall (2007), guided by Ferreira et al. (2005) and Ferreira and Marshall (2006), proposed to specify the vertical variation of κ_{GM} using

$$\kappa_{\mathrm{GM}} = \Big[\frac{N^2}{N^2_{\mathrm{ref}}}\Big][\kappa_{\mathrm{GM}}]_{\mathrm{ref}}, \tag{9.59}$$

where N is the local buoyancy frequency and $[\kappa_{\mathrm{GM}}]_{\mathrm{ref}}$ is the constant reference value of κ_{GM} within the surface diabatic layer. N_{ref} is the reference buoyancy frequency obtained just below the diabatic layer, in other words, the first stable N^2 below surface diabatic layer. The ratio N^2/N^2_{ref} is set to 1 for all shallower depths. Between the depth at which $N^2 = N^2_{\mathrm{ref}}$ and the ocean bottom, we also ensure that

$$N_{\mathrm{min}} \le \frac{N^2}{N^2_{\mathrm{ref}}} \le 1.0, \tag{9.60}$$

where N_{min} is the lower limit specified by the user (`ratio_bvf_min` in namelist `nml_gmvar_danabasoglu`).

9.4.3 Surface tapering

By default, no specific modification is applied to the eddy-induced transport velocity of the Gent-McWilliams parameterization near the surface and the bottom, except for limiting the isopycnal slope to a specified value (`slope_clip_gm` in namelist `nml_isopy_slope_clip`). This may result in too strong transport velocity in the first vertical level of the model (sea surface). The problem may be overcome by tapering the transport in the surface mixed layer, where the transport is made nearly or completely uniform in the vertical direction. This is realized by choosing either `SLIMIT` or `GMTAPER` option.

a. Simple scheme

By choosing `SLIMIT` option, the Gent-McWilliams coefficient (κ_{GM}) is linearly reduced from the value at the base of mixed layer to zero at the sea surface within the mixed layer,

$$\kappa_{\mathrm{GM}}(z) = \kappa_{\mathrm{GM}}(z = -\mathrm{MLD}) \times (-z/\mathrm{MLD}) \quad \text{for} \quad -\mathrm{MLD} \le z \le 0, \tag{9.61}$$

where MLD is the depth of the mixed layer. The MLD is defined as the level at which the local potential density is larger than the surface density by a specified value, given by the user (default value is $0.03\,\mathrm{kg\,m^{-3}}$).

b. Danabasoglu et al. (2008)

By choosing `GMTAPER` option, a practical scheme proposed by Danabasoglu et al. (2008) is used. This scheme modifies the Gent-McWilliams vector stream function for eddy induced transport velocity near the surface, aiming to implement a near-surface parameterization proposed by Ferrari et al. (2008). Concept of the near-surface parameterization is as follows (Danabasoglu et al., 2008):

- In the turbulent boundary layer (BL), the eddy-induced velocity is set parallel to the boundary and has no vertical shear, as expected in the mixed layer.
- There is an eddy diffusion of buoyancy along the boundary as well as along isopycnals.
- In the interior the parameterization satisfies the adiabatic constraint as in the original scheme.
- The two forms are matched through a transition layer that separates the quasi-adiabatic interior with isopycnically oriented eddy fluxes from the near boundary regions.

Two vertical length scales must be estimated to implement this parameterization: the boundary layer depth (BLD) and the transition layer thickness (TLT). Their sum is defined as the diabatic layer depth (DLD), over which the upper-ocean eddy fluxes depart from their interior formulas. In MRI.COM, the surface mixed layer depth (MLD) is treated as the BLD. The MLD is defined as the level at which the local potential density is larger than the surface density by a specified value, given by the user (default value is $0.03\,\mathrm{kg\,m^{-3}}$). The TLT is defined by the range of isopycnals that can be lifted into the boundary layer by eddy heaving, which is given by the product of the internal deformation radius (R) and the isopycnal slope ($|\mathbf{S}|$):

$$D = R|\mathbf{S}|. \tag{9.62}$$

Thus we calculate D at each grid point and the DLD is obtained as follows:

$$\mathrm{DLD} = \mathrm{BLD} + D. \tag{9.63}$$

Now the near-surface expression for the eddy-induced vector streamfunction is given in the following. The streamfunction is split into its boundary layer, Ψ_{BL}, and transition layer, Ψ_{TL}, expression as follows:

$$\Psi_{\mathrm{BL}} = \frac{\eta - z}{\eta + \mathrm{BLD}}\Psi_0 \quad \text{for} \quad -\mathrm{BLD} \le z \le \eta \tag{9.64}$$

and

$$\Psi_{\mathrm{TL}} = \Big(\frac{z + \mathrm{BLD}}{\mathrm{TLT}}\Big)^2\Phi + \Big(\frac{\eta - z}{\eta + \mathrm{BLD}}\Big)\Psi_0 \quad \text{for} \quad -\mathrm{DLD} \le z < -\mathrm{BLD} \tag{9.65}$$

The two functions Ψ_0 and Φ are chosen such that Ψ and its vertical derivative are continuous across the base of BLD and the base of TLT. These constraints then yield

$$\Psi_0 = \frac{\eta + \mathrm{BLD}}{2(\eta + \mathrm{BLD}) + \mathrm{TLT}}(2\Psi_I + \mathrm{TLT}\partial_z\Psi_I) \tag{9.66}$$

and

$$\Phi = -\frac{\mathrm{TLT}}{2(\eta + \mathrm{BLD}) + \mathrm{TLT}}(\Psi_I + (\eta + \mathrm{DLD})\partial_z\Psi_I), \tag{9.67}$$

where Ψ_I is the interior eddy-induced streamfunction at the base of the transition layer given by the Gent-McWilliams parameterization,

$$\Psi_I = -\kappa_{\mathrm{GM}}\frac{\mathbf{z}\times\nabla_H\rho}{\partial_z\rho} \quad \text{at} \quad z = -\mathrm{DLD}. \tag{9.68}$$

In the implementation, to evaluate both Ψ_I and $\partial_z\Psi_I$ at $z = -\mathrm{DLD}$, Ψ_I are evaluated at the vertical grid points that straddle $z = -\mathrm{DLD}$.

Danabasoglu et al. (2008) also showed that the model solutions are not very sensitive to their transition layer thickness. Whether the transition layer is included or not may be specified by `l_transition_layer` in namelist `nml_gm_transition`.

9.4.4 Anisotropic Gent-McWilliams scheme

An anisotropic GM scheme (Smith and Gent (2004), `GMANISOTROP` option), which gives greater diffusivity only in the direction of the current vector, is also available. Using unit vector $\hat{\mathbf{n}} = (n_x, n_y)$ in an arbitrary direction, the two-dimensional anisotropic diffusion tensor is defined as follows:

$$\mathbf{K}_2 = \begin{pmatrix} L & M \\ M & N \end{pmatrix} = \begin{pmatrix} \kappa_A n_x^2 + \kappa_B n_y^2 & \kappa_B n_x n_y \\ \kappa_B n_x n_y & \kappa_B n_x^2 + \kappa_A n_y^2 \end{pmatrix}, \tag{9.69}$$

where κ_A is the diffusivity in the $\hat{\mathbf{n}}$ direction, and κ_B is that in the direction normal to $\hat{\mathbf{n}}$. This is applied to the anti-symmetric tensor in the Gent-McWilliams scheme, and the following expression is obtained (Smith and Gent, 2004),

$$\mathbf{A}' = \begin{pmatrix} 0 & 0 & -LS_x - MS_y \\ 0 & 0 & -MS_x - NS_y \\ LS_x + MS_y & MS_x + NS_y & 0 \end{pmatrix}. \tag{9.70}$$

In the choice of `GMANISOTROP` option, $\hat{\mathbf{n}}$ is set in the direction of the local horizontal velocity. The value of κ_A is read from namelist `nml_tracer_diff_isopy` (variable name `diff_thick_cm2ps`). The ratio of κ_B/κ_A is read from namelist `nml_gmanisotrop` (variable name `cscl_isotrop`). The default value of `cscl_isotrop` is set to 1/2.

9.4.5 Usage Summary

How to specify the overall behavior of the Gent-McWilliams parameterization is summarized as follows.

a. Model options

Model options related to the GM parameterization are listed on Table 9.6

Table9.6 List of model options related to GM parameterization.

option name	description	usage
GMVAR	Coefficient of GM parameterization is allowed to vary	specify nml_gmvar
SLIMIT	Linearly reduce the coefficient of GM parameterization from the bottom of the mixed layer to the sea surface	cannot be used with GMTAPER
GMTAPER	Taper GM vector stream function near the sea surface	cannot be used with SLIMIT, GMANISOTROP, AFC
GMANISOTROP	An-isotropic horizontal variation of GM parameterization	specify nml_gmanisotrop
AFC	Calculate additional flux by using horizontal gradients of density and velocity (Hirabara et al., 2010)	cannot be used with TRCBIHARM

b. Spatial dependency

The diffusion coefficient of GM parameterization may be grid size dependent by using the namelist listed on Table 9.7.

Table9.7 Namelist nml_grid_size_change_mix_coefs

variable name	units	description	usage
l_grid_size_change_mix_coefs	logical	the given coefficient is multiplied by the fraction of the local grid size to 100 km.	default = .false.

Overall behavior of GM parameterization with GMVAR option should be specified by using the namelists listed on Tables 9.8 through 9.12.

Table9.8 Namelist nml_gmvar_select for GMVAR

variable name	units	description	usage
l_visbeck	logical	use Visbeck et al. (1997)	choose only one of the three options
l_eden	logical	use Eden and Greatbatch (2008)	choose only one of the three options
l_danabasoglu	logical	use Danabasoglu and Marshall (2007)	choose only one of the three options

Table9.9 Namelist nml_gmvar_visbeck for GMVAR

variable name	units	description	usage
start_depth	cm	density gradients are averaged from start_depth	l_visbeck = .true.
base_depth	cm	density gradients are averaged to base_depth	l_visbeck = .true.
cscl_gmvar	1	parameter for GM diffusivity calculation	l_visbeck = .true.
upper_limit	$cm^2\ sec^{-1}$	upper limit of thickness diffusivity	l_visbeck = .true.
lower_limit	$cm^2\ sec^{-1}$	lower limit of thickness diffusivity	l_visbeck = .true.
lcalc_defrad	logical	flag whether deformation radius is calculated or not	l_visbeck = .true.
defrad_const	cm	upper limit of deformation radius when lcalc_defrad = .true. constant horizontal length scale when lcalc_defrad = .false.	l_visbeck = .true.
length_factor	1	[horizontal length scale] = [deformation radius] × length_factor	l_visbeck = .true.

Table9.10 Namelist nml_gmvar_eden for GMVAR

variable name	units	description	usage
c_EG	1	In this parameterization, Thickness diffusivity is parameterized as c_EG $\times L^2 \times$ sigma_EG.	l_eden = .true.
gamma_EG	1	sigma_EG = $f/(\mathrm{Ri} +$ gamma_EG$)$	l_eden = .true.
upper_limit	cm^2 sec^{-1}	upper limit of thickness diffusivity	l_eden = .true.
lower_limit	cm^2 sec^{-1}	lower limit of thickness diffusivity	l_eden = .true.

Table9.11 Namelist nml_gmvar_danabasoglu for GMVAR

variable name	units	description	usage
ratio_bvf_min	1	Lower bound for the squared buoyancy frequency relative to the reference value (default = 0.1)	l_danabasoglu = .true.
ratio_bvf_max	1	Upper bound for the squared buoyancy frequency relative to the reference value (default = 1.0)	l_danabasoglu = .true.

Table9.12 Namelist nml_gm_transition

variable name	units	description	usage
l_transition_layer	logical	include transition layer into diabatic layer (default = .false.)	effective when l_danabasoglu = .true.

c. Anisotropic scheme

Behavior of GMANISOTROP option should be specified using the namelist listed on Table 9.13.

Table9.13 Namelist nml_gm_anisotrop

variable name	units	description	usage
cscl_isotrop	1	factor for anisotropy in GM diffusivity	if GMANISOTROP

Chapter 10

SGS parameterization of vertical mixing of tracers

This chapter explains subgrid-scale parameterizations of vertical mixing of tracers.

10.1 Vertical diffusion

Vertical diffusion term takes the form of Laplacian and the vertical diffusion flux is proportional to the vertical gradient of tracer. The finite difference form is as follows:

$$FZD_{i,j,k} = -\kappa_z \, (\texttt{areat})_{i,j,k+\frac{1}{2}} \, \delta_z T_{i,j,k}, \tag{10.1}$$

where the use of $(\texttt{areat})_{i,j,k+\frac{1}{2}}$ implies that the flux occurs only through the oceanic part of the grid interface and

$$\delta_z T_{i,j,k} \equiv \frac{T_{i,j,k-\frac{1}{2}} - T_{i,j,k+\frac{1}{2}}}{\Delta z_k}. \tag{10.2}$$

Note that, for simplicity, the change of the grid thickness at the bottom and fluctuations of the surface height are not considered in the grid distance Δz_k when calculating the gradient.

In most realistic simulations, a backward (implicit) scheme is used in the time integration (`VVDIMP` option; Section 19.5) because high diffusivity is expected owing to the choice of parameterizations needed for realistic simulations. Otherwise, a forward scheme is used.

10.1.1 Specification of coefficient

Background vertical diffusivity, which is horizontally uniform, a function of depth, and fixed in time, should be always given. Additionally, a three dimensional distribution can be set by selecting `VMBG3D` option to incorporate locally enhanced mixing processes induced by interaction between the bottom topography and tidal currents (e.g., St. Laurent et al., 2002). With this choice, three dimensional distributions for vertical diffusivity and viscosity should be prepared in advance. Tables 10.1 and 10.2 summarizes how to give background vertical diffusivity.

Table10.1 Namelist `nml_diff_vert_bg`. Specify only one of the two variables

variable name	units	description	usage
`diff_vert_bg_cm2ps`	$\mathrm{cm^2\ s^{-1}}$	vertically uniform value of background vertical diffusivity	Usable only if a vertically uniform value is intended
`file_diff_vert_1d_cm2ps`		file having vertical 1D distribution	cannot be specified with the above

Table10.2 Namelist `nml_vmbg3d`. Specify when `VMBG3D` is selected

variable name	units	description	usage
`file_vmix_3d`		file having 3D distribution	
`imvm`		east-west data size	
`jmvm`		north-south data size	

Continued on next page

Table 10.2 – continued from previous page

variable name	units	description	usage
l_vmintpol	logical	interpolate input data to model grid points	
slatvm		latitude of the western end	only if l_vmintpol = .true.
slonvm		longitude of the southern end	only if l_vmintpol = .true.
dlatvm		uniform grid spacing in the meridional direction	only if l_vmintpol = .true.
dlonvm		uniform grid spacing in the zonal direction	only if l_vmintpol = .true.

In addition to the static background profiles, the following processes give varying vertical diffusivity coefficients at every model time step.

- Surface mixed layer models (TURBULENCE option).
- Vertical component of isopycnal diffusion (ISOPYCNAL option).
- Enhanced diffusivity (= $1.0\,\mathrm{m^2\,s^{-1}} = 10^4\,\mathrm{cm^2\,s^{-1}}$) where the stratification is unstable (DIFAJS option).
- Enhanced diffusivity around rivermouths to avoid too low salinity if river run-off is received by the model (RUNOFF option). This scheme is especially needed when positive definiteness is not guaranteed by a tracer advection algorithm. See Table 10.3 for how to specify the mixing.

The vertical diffusion for "this" time step is taken as the largest of the above estimations.

Table10.3 Namelist nml_vmix_river. Specify when RUNOFF is selected

variable name	units	description	usage
l_enhance_vmix_rivmouth	logical	diffusivity is enlarged around the rivermouth	default = .false.
diff_max_vmixriv_cm2ps	$\mathrm{cm^2\,sec^{-1}}$	maximum value of the enlarged diffusivity (= κ_{rivmax})	default = $1 \times 10^4\,\mathrm{cm^2\,sec^{-1}}$
depth_max_vmixriv_cm	cm	vertical diffusion is enlarged from surface to this depth, this is also used by subroutine salinity_limit	default = 30×10^2 cm
para_vmixriv_1	1	parameter for the enlarged vertical diffusion formula (= a). See below.	default = 10
para_vmixriv_2	1	parameter for the enlarged vertical diffusion formula (= b). Enlarged diffusion is calculated as $\kappa_{\mathrm{riv}} = \min(a^{\log_{10} W_{\mathrm{river}}+b}, \kappa_{\mathrm{rivmax}})$, where W_{river} is river discharge rate in $\mathrm{cm\,sec^{-1}}$	default = 7

10.2 Convective adjustment

Convective adjustment is realized by replacing the density (temperature and salinity) that is statically unstable (the upper density exceeds the lower density) in a water column with the averaged density between neighboring levels (neutralization), considering that interior convection occurs in that place. Most of the realistic phenomena represented by the convective adjustment include the developing mixed layer due to surface cooling during winter. Convective adjustment also includes the case in which dense bottom water flows out the sill and flows down along the slope. Moreover, the convective adjustment includes the practical effect that it suppresses disturbances caused by the numerical calculation error and smooths the distribution.

In general, there are three numerical schemes for convective adjustment. In the simplest one, adjustment is done for a pair of two neighboring levels, and then for a pair of another two neighboring levels. By repeating this procedure, it attempts to neutralize the density in the unstable part. This procedure is simple at each step, but it has a defect that the finite-time repetition does not necessarily guarantee reaching the complete averaged value. In the second scheme, adjustment is done by assigning a high vertical diffusivity between the two levels that are statically unstable and by solving the vertical diffusion term using an implicit method. This method cannot remove the unstable condition completely in one procedure. However, it has good calculation efficiency for the case where the model has a high vertical diffusivity already due to the mixed layer or isopycnal diffusion schemes and thus needs an implicit method to solve it. In MRI.COM, this scheme is invoked by specifying DIFAJS option. The vertical diffusivity between the unstable grid points is set to $10\,000\,\mathrm{cm^2\,s^{-1}}$.

In the third scheme, the unstable part is first neutralized. The stability at the top and bottom of the neutralized column is then reexamined. If the unstable condition remains, the part including the already-neutralized column is re-neutralized. This procedure continues until the instability at the top and bottom of the neutralized column disappears. This method can remove the unstable part completely and thus is called "Complete Convection," but it requires a number of iterations, the vertical level size minus one, at maximum. The third method, which is the default scheme in MRI.COM, is explained below (Ishizaki, 1997).

10.2.1 Algorithm

In order to minimize the judgment process ("IF" statement) and replace it by arithmetic calculation, this scheme defines two integer indices, α_k and β_k, at the layer boundaries, and six real variables TU_k, TL_k, SU_k, SL_k, VU_k, and VL_k, $(k = 1, KM-1)$, in addition to the vertical rows of temperature, salinity, and density $T_k, S_k, R_k, (k = \frac{1}{2}, KM-\frac{1}{2})$ (KM is the number of levels; see Figure 10.1). The level at the vertical boundary of a T-cell corresponds to the integer k. The index α_k indicates an unstable part within a water column: $\alpha_k = 1$ if it is unstable at the level between $k-\frac{1}{2}$ and $k+\frac{1}{2}$, and $\alpha_k = 0$ if it is neutral or stable. The index β_k memorizes the mixed part: $\beta_k = 1$ at the boundary where it is neutral as a result of mixing, and $\beta_k = 0$ elsewhere. Variables TU_k, SU_k, and VU_k and TL_k, SL_k, and VL_k are temperature, salinity and volume accumulated by multiplying α above the level k and below the level k, respectively, and are expressed by the following recursive relation.

$$
\begin{aligned}
VU_1 &= \alpha_1 V_{\frac{1}{2}},\\
VU_2 &= \alpha_2(V_{1+\frac{1}{2}} + \alpha_1 V_{\frac{1}{2}}) = \alpha_2(V_{1+\frac{1}{2}} + VU_1),\\
&\cdots,\\
VU_k &= \alpha_k(V_{k-\frac{1}{2}} + VU_{k-1}),\\
&\cdots,\\
VU_{KM-1} &= \alpha_{KM-1}(V_{KM-1-\frac{1}{2}} + VU_{KM-2}),
\end{aligned}
\tag{10.3}
$$

and

$$
\begin{aligned}
VL_{KM-1} &= \alpha_{KM-1} V_{KM-\frac{1}{2}},\\
VL_{KM-2} &= \alpha_{KM-2}(V_{KM-1-\frac{1}{2}} + \alpha_{KM-1} V_{KM-\frac{1}{2}}) = \alpha_{KM-2}(V_{KM-1-\frac{1}{2}} + VL_{KM-1}),\\
&\cdots,\\
VL_k &= \alpha_k(V_{k+\frac{1}{2}} + VL_{k+1}),\\
&\cdots,\\
VL_1 &= \alpha_1(V_{1+\frac{1}{2}} + VL_2),
\end{aligned}
\tag{10.4}
$$

where $V_{k+\frac{1}{2}}$ denotes a volume of the cell at the level $k+\frac{1}{2}$. In a similar way, other quantities are expressed as follows:

$$
\begin{aligned}
TU_1 &= \alpha_1 T_{\frac{1}{2}} V_{\frac{1}{2}}, \quad TU_k = \alpha_k(T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + TU_{k-1}),\\
SU_1 &= \alpha_1 S_{\frac{1}{2}} V_{\frac{1}{2}}, \quad SU_k = \alpha_k(S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + SU_{k-1}),\\
TL_{KM-1} &= \alpha_{KM-1} T_{KM-\frac{1}{2}} V_{KM-\frac{1}{2}}, \quad TL_k = \alpha_k(T_{k+\frac{1}{2}} V_{k+\frac{1}{2}} + TL_{k+1}),\\
SL_{KM-1} &= \alpha_{KM-1} S_{KM-\frac{1}{2}} V_{KM-\frac{1}{2}}, \quad SL_k = \alpha_k(S_{k+\frac{1}{2}} V_{k+\frac{1}{2}} + SL_{k+1}),
\end{aligned}
\tag{10.5}
$$

where $T_{k+\frac{1}{2}}$ and $S_{k+\frac{1}{2}}$ are temperature and salinity at the level $k+\frac{1}{2}$.

According to this definition, if $\alpha_k = 1$ and elsewhere 0, we get

$$
\begin{aligned}
VU_k + VL_k &= V_{k-\frac{1}{2}} + V_{k+\frac{1}{2}},\\
TU_k + TL_k &= T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + T_{k+\frac{1}{2}} V_{k+\frac{1}{2}},\\
SU_k + SL_k &= S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + S_{k+\frac{1}{2}} V_{k+\frac{1}{2}},
\end{aligned}
$$

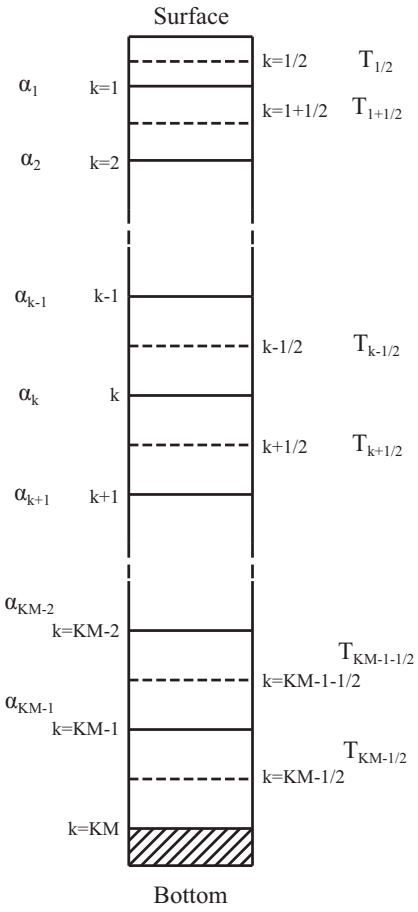

Figure10.1 Reference vertical grid points in Section 10.2

indicating a volume and accumulated temperature and salinity in an unstable part and

$$\begin{aligned} TM_{k-\frac{1}{2},k+\frac{1}{2}} &= \frac{TU_k + TL_k}{VU_k + VL_k}, \\ SM_{k-\frac{1}{2},k+\frac{1}{2}} &= \frac{SU_k + SL_k}{VU_k + VL_k}, \end{aligned} \tag{10.6}$$

are volume averaged temperature and salinity, respectively.

If the level k constitutes a series of the unstable part, the same equation holds for the averaged temperature and salinity. For example, let $\alpha_{k-1} = \alpha_k = 1$ and $\alpha_{k-2} = \alpha_{k+1} = 0$,

$$\begin{aligned} VU_{k-1} + VL_{k-1} &= VU_k + VL_k \\ &= V_{k-1-\frac{1}{2}} + V_{k-\frac{1}{2}} + V_{k+\frac{1}{2}}, \\ TU_{k-1} + TL_{k-1} &= TU_k + TL_k \\ &= T_{k-1-\frac{1}{2}} V_{k-1-\frac{1}{2}} + T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + T_{k+\frac{1}{2}} V_{k+\frac{1}{2}}, \\ SU_{k-1} + SL_{k-1} &= SU_k + SL_k \\ &= S_{k-1-\frac{1}{2}} V_{k-1-\frac{1}{2}} + S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + S_{k+\frac{1}{2}} V_{k+\frac{1}{2}}, \end{aligned} \tag{10.7}$$

and

$$\begin{aligned} TM_{k-1-\frac{1}{2},k+\frac{1}{2}} &= \frac{TU_{k-1} + TL_{k-1}}{VU_{k-1} + VL_{k-1}} = \frac{TU_k + TL_k}{VU_k + VL_k}, \\ SM_{k-1-\frac{1}{2},k+\frac{1}{2}} &= \frac{SU_{k-1} + SL_{k-1}}{VU_{k-1} + VL_{k-1}} = \frac{SU_k + SL_k}{VU_k + VL_k}. \end{aligned} \tag{10.8}$$

These are averages of the three layer, $k - 1 - \frac{1}{2}$, $k - \frac{1}{2}$, and $k + \frac{1}{2}$.

10.2.2 Numerical procedure

In summary, numerical procedures are summarized as follows.

[1] Density is calculated at the intermediate depth between adjacent levels using a pair of temperature and salinity and is judged to be statically stable or unstable. If an instability occurs, $\alpha(\alpha^1)$ is replaced by 1, otherwise by 0. At this stage, $\beta(\beta^0)$ is set to 0, where the superscript denotes the number of the iteration.

After this preprocessing, the following procedure (represented by n-th) is repeated until the instability is removed.

[2] Based on equations (10.3) to (10.5), VU, TU, SU, VL, TL, and SL are calculated using α^n for a water column that includes an unstable part.

[3] The vertical mean TM and SM are calculated for the unstable part using equation (10.6) and substituted for the original values of T and S. This change modifies the density at the intermediate depth in [1].

[4] The value of α^n is stored in β^n. $\beta^n = 1$ is set if $\alpha^n = 1$, or $\alpha^n = 0$ and $\beta^{n-1} = 1$, and otherwise $\beta^n = 0$. This is presented by the following:

$$\beta_k^n = \alpha_k^n + \beta_k^{n-1}(1 - \alpha_k^n). \tag{10.9}$$

[5] The static stability is judged only for $\beta_k^n = 0$. Let $\alpha_k^{n+1} = 1$ if statically unstable, and 0 otherwise. If there is no unstable part, the procedure for that water column is completed.

[6] For a water column which still includes an unstable part, modification for α_k^{n+1} is done using β_k^n by the following. After the procedure [2], any instability will be found only at the bottom of the part that is neutral as a result of prior mixing. In that case, the neutral part must be treated as an unstable part, that is, $\alpha_k^{n+1} = 1$. On the other hand, no more procedure is needed if the upper and lower end is stable, giving $\alpha_k^{n+1} = 0$. This is done by a recursive formula going down and up in the following.

$$\begin{aligned} &\gamma_1 = \alpha_1^{(n+1)}, \quad \gamma_k = \alpha_k^{(n+1)} + (1 - \alpha_k^{(n+1)})\beta_k^{(n)}\gamma_{k-1} \\ &\alpha_{KM-1}^{(n+1)} = \gamma_{KM-1}, \quad \alpha_k^{(n+1)} = \gamma_k + (1 - \gamma_k)\beta_k^{(n)}\alpha_{k+1}^{(n+1)}, \end{aligned} \tag{10.10}$$

where γ is a work variable, but may be treated as α itself in a FORTRAN program. Then, the procedure goes back to [2].

Table 10.4 shows an example of the case with six levels. Static instability is removed after the three-time iteration. The second column of α in the table is the result of the corrected α_k^{n+1} using β_k^n based on equation (10.10), as described in [6]. Note that $\beta_k^0 = 0$, though there is no description in the table.

Table10.4 Example of the convective adjustment procedure

n	k	α		VU	VL	VU+VL	TU+TL	β
1	1	1	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{T}_{\frac{1}{2}}\mathbf{V}_{\frac{1}{2}}+\mathbf{T}_{1\frac{1}{2}}\mathbf{V}_{1\frac{1}{2}}+\mathbf{T}_{2\frac{1}{2}}\mathbf{V}_{2\frac{1}{2}}$	1
	2	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{T}_{\frac{1}{2}}\mathbf{V}_{\frac{1}{2}}+\mathbf{T}_{1\frac{1}{2}}\mathbf{V}_{1\frac{1}{2}}+\mathbf{T}_{2\frac{1}{2}}\mathbf{V}_{2\frac{1}{2}}$	1
	3	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	5	1	1	$\mathbf{V}_{4\frac{1}{2}}$	$\mathbf{V}_{5\frac{1}{2}}$	$\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\mathbf{T}_{4\frac{1}{2}}\mathbf{V}_{4\frac{1}{2}}+\mathbf{T}_{5\frac{1}{2}}\mathbf{V}_{5\frac{1}{2}}$	1
2	1	0	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	2	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	3	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\sum_{k=0}^{3}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	4	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	1
3	1	0	1	$\mathbf{V}_{\frac{1}{2}}$	$\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	2	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}$	$\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	3	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}$	$\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	4	1	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}$	$\mathbf{V}_{4\frac{1}{2}}+\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1
	5	0	1	$\mathbf{V}_{\frac{1}{2}}+\mathbf{V}_{1\frac{1}{2}}+\mathbf{V}_{2\frac{1}{2}}+\mathbf{V}_{3\frac{1}{2}}+\mathbf{V}_{4\frac{1}{2}}$	$\mathbf{V}_{5\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{V}_{k+\frac{1}{2}}$	$\sum_{k=0}^{5}\mathbf{T}_{k+\frac{1}{2}}\mathbf{V}_{k+\frac{1}{2}}$	1

Chapter 11

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.

11.1 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 uptake or release of CO_2 from the ocean 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 these cases, 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., 2016). 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.

11.2 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.

$$\frac{\partial \mathrm{DIC}}{\partial t} = -\mathcal{A}(\mathrm{DIC}) + \mathcal{D}(\mathrm{DIC}) + S_b(\mathrm{DIC}) + J_v(\mathrm{DIC}) + J_g(\mathrm{DIC}), \tag{11.1}$$

$$\frac{\partial \mathrm{Alk}}{\partial t} = -\mathcal{A}(\mathrm{Alk}) + \mathcal{D}(\mathrm{Alk}) + S_b(\mathrm{Alk}) + J_v(\mathrm{Alk}), \tag{11.2}$$

$$\frac{\partial [\mathrm{PO_4}]}{\partial t} = -\mathcal{A}([\mathrm{PO_4}]) + \mathcal{D}([\mathrm{PO_4}]) + S_b([\mathrm{PO_4}]), \tag{11.3}$$

$$\frac{\partial [\mathrm{O_2}]}{\partial t} = -\mathcal{A}([\mathrm{O_2}]) + \mathcal{D}([\mathrm{O_2}]) + S_b([\mathrm{O_2}]) + J_g([\mathrm{O_2}]), \tag{11.4}$$

$$\frac{\partial [\mathrm{NO_3}]}{\partial t} = -\mathcal{A}([\mathrm{NO_3}]) + \mathcal{D}([\mathrm{NO_3}]) + S_b([\mathrm{NO_3}]), \tag{11.5}$$

$$\frac{\partial [\mathrm{PhyPl}]}{\partial t} = -\mathcal{A}([\mathrm{PhyPl}]) + \mathcal{D}([\mathrm{PhyPl}]) + S_b([\mathrm{PhyPl}]), \tag{11.6}$$

$$\frac{\partial [\mathrm{ZooPl}]}{\partial t} = -\mathcal{A}([\mathrm{ZooPl}]) + \mathcal{D}([\mathrm{ZooPl}]) + S_b([\mathrm{ZooPl}]), \tag{11.7}$$

$$\frac{\partial [\mathrm{Detri}]}{\partial t} = -\mathcal{A}([\mathrm{Detri}]) + \mathcal{D}([\mathrm{Detri}]) + S_b([\mathrm{Detri}]), \tag{11.8}$$

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 mol/m^3 of the substance within them. The terms represented by $J_g()$ and $J_v()$ are the air-sea gas fluxes (Section 11.3.1) and the dilution and concentration effects of evaporation and precipitation on DIC and Alk (Section 11.3.2), 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.

11.3 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. (2016). 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.

11.3.1 Air-sea gas exchange fluxes at the sea surface (J_g)

The air-sea gas transfer must be calculated for DIC and $[O_2]$. The terms $J_g(\mathrm{DIC})$ and $J_g([O_2])$ appear only in the uppermost layer. When these fluxes are expressed as $F_g(\mathrm{DIC})$ and $F_g([O_2])$, $J_g(\mathrm{DIC})$ and $J_g([O_2])$ are given as follows:

$$J_g(\mathrm{DIC}) = \frac{F_g(\mathrm{DIC})}{\Delta z_{\frac{1}{2}}}, \tag{11.9}$$

$$J_g([\mathrm{O_2}]) = \frac{F_g([\mathrm{O_2}])}{\Delta z_{\frac{1}{2}}}, \tag{11.10}$$

where

$$F_g(\mathrm{DIC}) = K_w^{\mathrm{CO_2}} * ([\mathrm{CO_2}]_{\mathrm{sat}} - [\mathrm{CO_2}]_{\mathrm{surf}}), \tag{11.11}$$

$$F_g([\mathrm{O_2}]) = K_w^{\mathrm{O_2}} * ([\mathrm{O_2}]_{\mathrm{sat}} - [\mathrm{O_2}]_{\mathrm{surf}}), \tag{11.12}$$

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.

Chapter 11 Biogeochemical model

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:

$$K_w^{CO_2} = a\left(\frac{Sc^{CO_2}}{660}\right)^{-1/2} U_{10}^2(1-f_i), \tag{11.13}$$

$$K_w^{O_2} = a\left(\frac{Sc^{O_2}}{660}\right)^{-1/2} U_{10}^2(1-f_i), \tag{11.14}$$

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 cm hr^{-1}/(m s^{-1})^2, which will give K_w in units of cm hr^{-1} when winds speeds (U_{10}) are in 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.

$$[A]_{\rm sat} = K_0 f_A \tag{11.15}$$

$$= K_0 C_f p_A \tag{11.16}$$

$$= K_0 C_f (P_a - p{\rm H_2O}) x_A \tag{11.17}$$

$$= \phi_A x_A \tag{11.18}$$

$$\simeq \frac{P_a}{P_a^\circ}\phi_A^0 x_A \qquad \text{(with errors less than 0.1\%)} \tag{11.19}$$

$$= \frac{P_a}{P_a^\circ}[A]_{\rm sat}^0, \tag{11.20}$$

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), $p{\rm H_2O}$ is water vapor pressure at saturation, ϕ_A is its solubility function, ϕ_A^0 and $[A]_{\rm sat}^0$ are its solubility function and saturation concentration at the reference pressure.

Orr et al. (2016) explain two approaches to calculate the saturation concentration.

The first approach uses equations (11.19) or (11.20), computing solubility function ϕ_A^0 or saturation concentration $[A]_{\rm sat}^0$ 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 $[O_2]_{\rm sat}^0$ is available. This is a rather conventional approach, and is used in MRI.COM except for carbon.

The other approach uses equations (11.15)-(11.17), computing its partial pressure $p_A \equiv (P_a - p{\rm H_2O})x_A$, 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, $[CO_2]_{\rm surf}$ is diagnosed every step from DIC, Alk, temperature, salinity, $[PO_4]$, and silicate concentration at the surface. After $[CO_2]_{\rm surf}$ is computed, fugacity (f_{CO_2}) and partial pressure (pCO_2) of the ocean surface are computed as

$$f_{CO_2} = [CO_2]_{\rm surf}/K_0, \tag{11.21}$$

$$pCO_2 = f_{CO_2}/C_f, \tag{11.22}$$

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:

$$\mathrm{DIC} = [CO_2] + [H_2CO_3] + [HCO_3^-] + [CO_3^{2-}] \tag{11.23}$$

$$= [CO_2^*] \qquad + [HCO_3^-] + [CO_3^{2-}]. \tag{11.24}$$

In the OMIP protocol, the following equations are solved to obtain $[CO_2^*]$.

The equilibrium constants for dissociation reactions are:

$$K_1 = \frac{[H^+][HCO_3^-]}{[CO_2^*]} \qquad K_2 = \frac{[H^+][CO_3^{2-}]}{[HCO_3^-]}, \tag{11.25}$$

$$K_B = \frac{[H^+][B(OH)_4^-]}{[B(OH)_3]}, \tag{11.26}$$

$$K_{1P} = \frac{[H^+][H_2PO_4^-]}{[H_3PO_4]} \qquad K_{2P} = \frac{[H^+][HPO_4^{2-}]}{[H_2PO_4^-]} \qquad K_{3P} = \frac{[H^+][PO_4^{3-}]}{[HPO_4^{2-}]}, \tag{11.27}$$

$$K_{Si} = \frac{[H^+][SiO(OH)_3^-]}{[Si(OH)_4]}, \tag{11.28}$$

$$K_W = [H^+][OH^-], \tag{11.29}$$

$$K_S = \frac{[H^+]_F[SO_4^{2-}]}{[H\,SO_4^-]}, \tag{11.30}$$

and

$$K_F = \frac{[H^+]_F[F^-]}{[HF]}, \tag{11.31}$$

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:

$$[H^+] = [H^+]_F \left(1 + \frac{S_T}{K_S} + \frac{F_T}{K_F}\right), \tag{11.32}$$

$$\text{and} \qquad [H^+]_T = [H^+]_F \left(1 + \frac{S_T}{K_S}\right). \tag{11.33}$$

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:

$$\mathrm{DIC} = [CO_2^*] + [HCO_3^-] + [CO_3^{2-}], \tag{11.34}$$

$$B_T = [B(OH)_3] + [B(OH)_4^-], \tag{11.35}$$

$$P_T = [H_3PO_4] + [H_2PO_4^-] + [HPO_4^{2-}] + [PO_4^{3-}], \tag{11.36}$$

$$Si_T = [Si(OH)_4] + [Si(OH)_3^-], \tag{11.37}$$

$$S_{\mathrm{T}} = [\mathrm{HSO_4^-}] + [\mathrm{SO_4^{2-}}], \tag{11.38}$$

and

$$F_{\mathrm{T}} = [\mathrm{HF}] + [\mathrm{F^-}]. \tag{11.39}$$

Alkalinity used in this calculation is defined as follows:

$$\begin{aligned}\mathrm{Alk} = {} & [\mathrm{HCO_3^-}] + 2[\mathrm{CO_3^{2-}}] + [\mathrm{B(OH)_4^-}] + [\mathrm{OH^-}] + [\mathrm{HPO_4^{2-}}] + 2[\mathrm{PO_4^{3-}}] + [\mathrm{SiO(OH)_3^-}] \\ & - [\mathrm{H^+}]_F - [\mathrm{HSO_4^-}] - [\mathrm{HF}] - [\mathrm{H_3PO_4}].\end{aligned} \tag{11.40}$$

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.

11.3.2 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 unless `SFLUXW` or `SFLUXR` option is used. In these options, 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

$$\mathtt{sflux}(i,j) = -(P-E) * S(i,j,1)/\Delta z, \tag{11.41}$$

where $S(i,j,1)$ and Δz are the salinity and thickness of the uppermost layer. Note that the variable $\mathtt{sflux}(i,j)$ 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:

$$J_v(\mathrm{DIC}(i,j,1)) = \mathtt{sflux}(i,j)/S(i,j,1) * \mathrm{DIC}(i,j,1), \tag{11.42}$$

$$J_v(\mathrm{Alk}(i,j,1)) = \mathtt{sflux}(i,j)/S(i,j,1) * \mathrm{Alk}(i,j,1). \tag{11.43}$$

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(11.42,11.43). In addition, globally integrated J_v(DIC) and J_v(Alk) are set to 0. In MRI.COM, these modifications are not the default considering the use in regional ocean models.

11.4 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 $rL[\mathrm{PO_4}]^2/([\mathrm{PO_4}]+k)$, 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 $(z/100\mathrm{m})^{-0.9}$ and $\exp(-z/3500\mathrm{m})$ (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:

$$S_b(\mathrm{DIC}) = R_{cp} * \mathrm{RemiP} + \mathrm{SolnCa} - R_{cp} * \mathrm{ExprodP} \tag{11.44}$$
$$S_b(\mathrm{Alk}) = 2 * \mathrm{SolnCa} + R_{np} * \mathrm{ExprodP} - R_{np} * \mathrm{RemiP} \tag{11.45}$$
$$S_b([\mathrm{PO_4}]) = \mathrm{RemiP} - \mathrm{ExprodP} \tag{11.46}$$
$$S_b([\mathrm{O_2}]) = -R_{op} * S_b([\mathrm{PO_4}]) \tag{11.47}$$

11.5 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 ($\mathrm{NO_3}$), 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, $S_b(\mathrm{DIC})$ and $S_b(\mathrm{Alk})$, used for calculating the carbon cycle, are described later in this section.

$$S_b([\mathrm{PhyPl}]) = \mathrm{Priprod} - \mathrm{MortP1} - \mathrm{MortP2} - \mathrm{GrP2Z} \tag{11.48}$$
$$S_b([\mathrm{ZooPl}]) = assim * \mathrm{GrP2Z} - \mathrm{Excrtn} - \mathrm{MortZ} \tag{11.49}$$
$$S_b([\mathrm{Detri}]) = [(1 - assim) * \mathrm{GrP2Z} + \mathrm{MortP2} + \mathrm{MortZ}] - \mathrm{RemiD} - w_{detri}\frac{\partial \mathrm{Detri}}{\partial z} \tag{11.50}$$
$$S_b([\mathrm{NO_3}]) = \mathrm{MortP1} + \mathrm{Excrtn} + \mathrm{RmeiD} - \mathrm{PriProd} \tag{11.51}$$
$$S_b([\mathrm{PO_4}]) = R_{pn} * S_b([\mathrm{NO_3}]) \tag{11.52}$$
$$S_b([\mathrm{O_2}]) = -R_{on} * R_{np} * S_b([\mathrm{PO_4}]) \tag{11.53}$$

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 ($-w_{detri}\frac{\partial \mathrm{Detri}}{\partial z}$). 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.

11.5.1 Description of each term

- Priprod = $J(I, N, P) * [\mathrm{PhyPl}]$
  Primary production expresses photosynthesis (described in detail in the next subsection).
- MortP1 = $\phi_P * [\mathrm{PhyPl}]$
  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 = $\phi_{PP} * [\mathrm{PhyPl}]^2$
  The conversion from phytoplankton to detritus (normal mortality of phytoplankton).
- GrP2Z = $G(P) * [\mathrm{ZooPl}]^2$
  The grazing of zooplankton. There are a number of parameterizations of grazing. In this formulation, this is given as $G(P) = g * \epsilon * [\mathrm{PhyPl}]^2/(g + \epsilon * [\mathrm{PhyPl}]^2)$. Among the grazing, the ratio $assim$ is used for the growth of zooplankton, and the remainder $(1 - assim)$ is converted to detritus.
- Excrtn = $d * [\mathrm{ZooPl}]$
  Excretion of zooplankton. The excretion is dissolute and directly returned to nutrients ($\mathrm{NO_3}$).

- MortZ = $\phi_Z *$ [ZooPl]
  The conversion from the zooplankton to detritus (mortality of zooplankton).
- RemiD = $\phi_D *$ [Detri]
  Remineralization of detritus. This is converted to nutrients through the activity of bacteria.

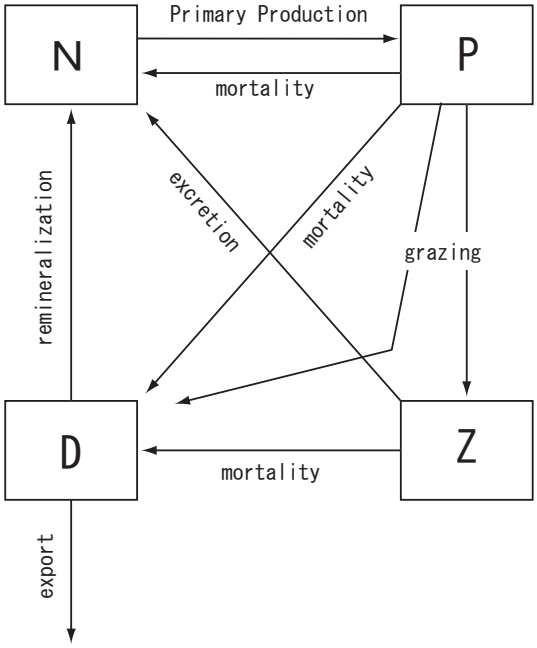

Figure11.1 Schematic of NPZD model

11.5.2 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:

$$J(I, \mathrm{N}, \mathrm{P}) = \min\left(J_I, J_N, J_P\right), \tag{11.54}$$

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:

$$J_I = \frac{J_{max}\alpha I}{\left[J_{max}^2 + (\alpha I)^2\right]^{1/2}}. \tag{11.55}$$

Here, J_{max} is the light-saturated growth, which depends on temperature based on Eppley (1972) as

$$J_{max} = a \cdot b^{c\theta}, \tag{11.56}$$

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 11.1). Equation (11.55) is called Smith-type growth. The coefficient α in the equation is "the initial slope of photosynthesis versus irradiance (P-I) curve," that is,

$$\alpha = \lim_{I \to 0} \frac{\partial J_I}{\partial I}. \tag{11.57}$$

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.

$$I = I_{z=0}\,\mathrm{PAR}\,\exp\left(-k_w\tilde{z} - k_e\int_0^{\tilde{z}} P dz\right), \tag{11.58}$$

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

$$J_N = J_{MM} = \frac{J_{max}N}{K_N + N}, \tag{11.59}$$

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:

$$J_N = J_{OU} = \frac{V_0 N}{N + 2\sqrt{\frac{V_0}{A_0}N} + \frac{V_0}{A_0}}, \tag{11.60}$$

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

$$V_0 = 0.5\left(1 + \sqrt{\frac{\alpha_{OU}}{K_N}}\right)^2 J_{max}. \tag{11.61}$$

Finally, we obtain

$$J_{OU} = \frac{V_0 N}{N + 2\sqrt{\alpha_{OU}N} + \alpha_{OU}}. \tag{11.62}$$

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).

11.5.3 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

$$S_b(\mathrm{DIC}) = S_b([PO_4])R_{cp} - S_b([CaCO_3]), \tag{11.63}$$

$$S_b(\mathrm{Alk}) = -S_b([NO_3]) - 2 \cdot S_b([CaCO_3]). \tag{11.64}$$

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,

$$S_b([CaCO_3]) = Pr([CaCO_3]) - Di([CaCO_3]). \tag{11.65}$$

Following Schmittner et al. (2008), the source term ($Pr([CaCO_3])$) of calcium carbonate is determined by the production of detritus as follows:

$$Pr([CaCO_3]) = [(1 - assim) * [\mathrm{GrP2Z}] + [\mathrm{MortP2}] + [\mathrm{MortZ}]]\, R_{CaCO_3/POC} R_{C:N}, \tag{11.66}$$

where *assim*, GrP2Z, MortP2, and MortZ are as described above. The sink term ($Di([CaCO_3])$) of calcium carbonate is parameterized as

$$Di([CaCO_3]) = \int Pr([CaCO_3])dz \cdot \frac{d}{dz}\left(\exp\left(-z/D_{CaCO_3}\right)\right), \tag{11.67}$$

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 $R_{CaCO_3/POC} = 0.035$ 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.

11.6 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 11.1. 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 repeatedly writing `nmlrs_ptrc`. The order of passive tracers is determined in `tracer_vars.F90`. Following is an example when `CARBON`, `O2`, and `NPZD` options are chosen. In this case, `numtrc_p = 8` should be specified in `configure.in`.

An example specification of restart files when `CARBON`, `O2`, and `NPZD` options are chosen.

```
&nmlrs_ptrc fname='result/rs_dic'/
&nmlrs_ptrc fname='result/rs_alk'/
&nmlrs_ptrc fname='result/rs_o2'  /
&nmlrs_ptrc fname='result/rs_po4' /
&nmlrs_ptrc fname='result/rs_no3' /
&nmlrs_ptrc fname='result/rs_PhyPl'/
&nmlrs_ptrc fname='result/rs_ZooPl'/
&nmlrs_ptrc fname='result/rs_Detri'/
```

Table11.1 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	$\mathrm{day^{-1}}$	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	$\mathrm{m^{-1}\,(mol\,m^{-3})^{-1}}$	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	$\mathrm{day^{-1}}$	1.575d0
epsbio	Prey capture rate	$\mathrm{(mol\,m^{-3})^{-2}\,day^{-1}}$	1.6d6
phiphy	Specific mortality/recycling rate	$\mathrm{s^{-1}}$	0.014d0
phiphyq	Quadratic mortality rate	$\mathrm{(mol\,m^{-3})^{-1}\,day^{-1}}$	0.05d3
a_npz	Assimilation efficiency		0.925d0
phizoo	Quadratic mortality of zooplankton	$\mathrm{(mol\,m^{-3})^{-1}\,day^{-1}}$	0.34d3
d_npz	Excretion	$\mathrm{day^{-1}}$	0.01d0
remina	Remineralization rate	$\mathrm{day^{-1}}$	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	400.d0
dp_eprdc	The depth where the bio-export is diagnosed.	m	126.d0
	This value should be less than dp_mrtn.		
Rcaco3poc	$CaCO_3$ over nonphotosynthetic POC production ratio		0.05d0
Dcaco3	$CaCO_3$ remineralization e-folding depth	m	3500.d0
shwv_intv	Interval for calculating the irradiance and light-limited growth rate. This must be a divisor of the time step for tracer.	min	10.d0

11.7 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 12

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.

12.1 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).

12.1.1 Source / Sink term

Source and sink of an age tracer a [year], is expressed as follows. In the oceanic interior,

$$\frac{\partial a}{\partial t} = -\mathcal{A}(a) + \mathcal{D}(a) + \frac{1}{\text{the total number of seconds of this year}} \quad \text{year/sec}, \tag{12.1}$$

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:

$$a(k = 1) = 0. \tag{12.2}$$

12.1.2 Usage

Required Model option `IDEALAGE`
Required Add one to `numtrc_p`
Required Restart file for passive tracer (`nmlrs_ptrc`) (See Section 11.6)
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.

12.2 CFCs

MRI.COM can treat CFCs (CFC11 and CFC12) following the protocols of OMIP (Orr et al., 2016). There is no source and sink for CFCs in the interior.

12.2.1 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,

$$\frac{\partial [A]}{\partial t} = -\mathcal{A}([A]) + \mathcal{D}([A]) + J_g([A]). \tag{12.3}$$

$J_g()$ represents source/sink due to the air-sea gas fluxes $F([A])$ at the sea surface,

$$J_g([A]) = \frac{F([A])}{\Delta z_{\frac{1}{2}}}, \tag{12.4}$$

with

$$F(A) = k_w([A]_{sat} - [A]), \tag{12.5}$$

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

$$[A]_{sat} = \phi^0_A x_A, \tag{12.6}$$

where x_A is its mole fraction in dry air. The combined solubility term ϕ^0_A is computed using the empirical fit of temperature and salinity. See Orr et al. (2016) for further details.

12.2.2 Usage

Required Model option `CFC`
Required Add two to `numtrc_p`
Required Restart file for passive tracer (`nmlrs_ptrc`) (See Section 11.6)
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'

12.3 SF_6

The calculation of SF_6 also follows the protocols of OMIP (Orr et al., 2016). The formulation of SF_6 is nearly the same as CFCs. Only the coefficients of the empirical fit differ.

12.3.1 Surface boundary condition

See Section 12.2.1, where CFCs should read SF_6.

12.3.2 Usage

Required Model option `SF6`
Required Add one to `numtrc_p`
Required Restart file (`nmlrs_ptrc`) (See Section 11.6)
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 13

Package Structure and Usage

13.1 Package structure

Packages relevant to tracers are listed as follows.

13.1.1 Tracer equation

tracer_ctl.F90:	Controller of this package
tracer.F90:	Main program of this package
upc_adv.F90:	Upcurrent advection scheme
quick_adv.F90:	QUICK advection scheme (`QUICKADVEC`)
utzq_adv.F90:	Combination of UTOPIA and QUICKEST advection scheme (`UTZQADVEC`)
som_adv.F90:	Second order moment advection scheme (`SOMADVEC`)
mpdata_adv.F90:	MPDATA advection scheme (`MPDATAADVEC`)
tracer_vars.F90:	Setting of tracer attributes
+vvdimp/trcimp.F90:	Solver of the vertical diffusion part using the implicit method (`VVDIMP`)
+isopycnal/ipcoef.F90:	Calculation of tensor components of neutral physics parameterization (`ISOPYCNAL`)
+isopycnal/ipycmix.F90:	Calculation of tendency due to neutral physics parameterization (`ISOPYCNAL`)

13.1.2 Vertical mixing coefficients

vmixcoef_ctl.F90:	Controller of the vertical mixing package
vmixcoef.F90:	Main program of the vertical mixing package
vmixcoef_vmbg.F90:	Estimation of background vertical diffusion coefficient
vmixcoef_vars.F90:	Declaration of variables
+runoff/vmixcoef_rivermouth.F90:	Estimation of vertical mixing coefficient around the river mouth

13.1.3 Stratification and convective adjustment

strat_adjust_ctl.F90:	Controller of stratification and adjustment package
stratification.F90:	Main program of calculation of stratification
cnvajs.F90:	Main program of convective adjustment
strat_adjust_vars.F90:	Declaration of variables

13.1.4 Reference state and restoring coefficient

restore_cond_ctl.F90:	Controller of reference state and restoring coefficient
restore_cond.F90:	Main program of reference state and restoring coefficient
force_data.F90:	Service package that handles external forcing data

13.1.5 Passive tracers

ptrc_ctl.F90:	Controller of passive tracer evolution
ptrc.F90:	Main program of passive tracer evolution that mainly treats surface sources and sinks

ptrc_subp.F90:	Sub package of passive tracer evolution that describes internal sources and sinks (called from `tracer.F90`)
+ptrc/cfc.F90, sf6.F90, etc.	Sub package of passive tracer evolution that describes specialized processes of particular tracers

13.2 Handling the initial state

How to determine the initial state for temperature and salinity is specified in namelist `nml_tracer_run`. Parameters are listed on Table 13.1.

Table13.1 Namelist `nml_tracer_run`

variable name	units	description	usage
`l_rst_tracer_in`	logical	`.true.` : Read restart files specified by `nmlrs_t` and `nmlrs_s` for the initial condition. `.false.`: Start condition depends on the `l_rst_uni_strati`	Default is the same as `l_rst_in` of `nml_run_ini_state`.
`l_rst_uni_strati`	logical	`.false.`: Start from 3D-distribution at the starting time of reference data following `nml_tracer_data`. `.true.` : Start from uniform stratification created by the reference data following `nml_tracer_data`. Time average is conducted based on `start_rec_uni_strati` and `end_rec_uni_strati`.	if `l_rst_tracer_in` = `.false.`
`start_rec_uni_strati` `end_rec_uni_strati`	integer	uniform stratification is created by the average from `start_rec_uni_strati` data record to `end_rec_uni_strati` record.	if `l_rst_uni_strati` = `.true.`

13.3 Configuration of tracers

The attributes of each tracer such as name, advection scheme, restoring condition, reference data, and restoring coefficients, are stored in the structural type `type_tracer_data` defined in `tracer_vars.F90`. The contents of this structural type are specified by namelist `nml_tracer_data`, which should be repeatedly defined as many times as the number of tracers that should be calculated. Tables 13.2 through 13.8 list the variables.

13.3.1 Name

List of effective names is found in subroutine `tracer_vars__set_num_and_name` of `tracer_data.F90`

Table13.2 Namelist `nml_tracer_data`

variable name	units	description	usage
`name`	character	Name of tracer. Two tracers are necessary: "Potential Temperature" and "Salinity."	Case sensitive. For example, "potential temperature" is not correct.

13.3.2 Advection scheme

Following can be specified as the name of the advection scheme (`adv_scheme%name`).

- "upc" : weighted UP-Current advection scheme (always available)
- "quick" : QUICK advection scheme (`QUICKADVEC`)
- "utzq" : UTOPIA + ZQUICKEST schemes with ultimate limiter (`UTZQADVEC`)
- "som" : Second-Order Moment advection scheme (`SOMADVEC`)
- "mpdata" : MPDATA advection scheme (`MPDATAADVEC`)

Table13.3 Namelist nml_tracer_data related to advection scheme

variable name	units	description	usage
adv_scheme%name	character	Name of the advection scheme.	Different advection schemes can be set for individual tracers.
adv_scheme%limiter_som_org	logical	Use flux limiter for SOM by Prather (1986)	SOMADVEC
adv_scheme%limiter_som_Merryfield03	logical	Use flux limiter for som by Merryfield and Holloway (2003)	SOMADVEC
adv_scheme%lrstin_som	logical	The SOM initial state of moments is read from file	SOMADVEC
adv_scheme%lrstout_som	logical	The SOM final state of moments is written to file	SOMADVEC
adv_scheme%limiter_mpdata_nonoscillatory	logical	Apply flux limiter for MPDATA	MPDATAADVEC
adv_scheme%eps_lim_mpdata	that of tracer	Very small value to avoid zero division	MPDATAADVEC
adv_scheme%min_value_mpdata	that of tracer	Minimum value for tracer	MPDATAADVEC

13.3.3 Restoring condition

The following are variables related to the restoring condition for a tracer.

Table13.4 Namelist nml_tracer_data related to restoring condition

variable name	units	description	usage
restore_conf%l_surf_restore	logical	restore condition at the surface is applied or not	default = .false.
restore_conf%l_body_restore	logical	restore condition in the interior is applied or not	default = .false.

13.3.4 Reference data

a. Three dimensional reference state for restoring

When the field of a tracer is restored to a reference state, the attributes of the reference state should be given by the variables listed on Table 13.5. This reference state is also used to produce an initial state for that tracer when its restart file is not available (See Table 13.1).

Table13.5 Namelist nml_tracer_data related to reference values for body forcing and initial condition

variable name	units	description	usage
trcref_conf%file_data	character	a file name that contains reference values for body forcing and initial condition.	
trcref_conf%file_data_grid	character	a file name of grid	needed if linterp = .true.
trcref_conf%imfrc	integer	grid size of data in x direction	
trcref_conf%jmfrc	integer	grid size of data in y direction	
trcref_conf%kmfrc	integer	grid size of data in z direction	
trcref_conf%interval	integer	regular time interval of data	positive value : unit is sec −1 : monthly −999 : steady forcing

Continued on next page

Table 13.5 – continued from previous page

variable name	units	description	usage
trcref_conf%num_data_max	integer	the number of record contained in the file	
trcref_conf%ifstart	ifstart(6)	[ymdhms] of the first record of the input file	1999,1,1,0,0,0 when the first record is the average value of Jan 1999 and its data interval is monthly.
trcref_conf%lrepeat	logical	climatological data is repeatedly used	default = .false.
trcref_conf%linterp	logical	interpolate horizontally or not	default = .false.
trcref_conf%linterp_v	logical	interpolate vertically or not	default = .false.
trcref_conf%ilinear	integer	interpolation method	1 : linear, 2 : spline
trcref_conf%luniform	integer	data is horizontally uniform or not	default = .false.
trcref_conf%luniform_v	logical	data is vertically uniform or not	default = .false.
trcref_conf%ldouble	logical	input data is double or not	default = .false.
trcref_conf%iverbose	integer	standard output of progress	1 : extensive, 0 : minimum
trcref_conf%ldefined	integer	the input data is defined or not	default = .false.

Format of tracer reference / restoring data is shown in the following.

Format of tracer reference / restoring data (trcref(_surf)_conf%file_data)

```
integer(4), parameter :: imn = 12, nu = 99
integer(4) :: imfrc, jmftc, kmfrc           ! data size
character(128) :: file_data, fname_grid
real(4) :: ttlev(imfrc,jmfrc,kmfrc,imn)
real(8) :: alonf(imfrc), alatf(jmfrc), dpf(kmfrc)
logical :: linterp, linterp_v

! main data
open (unit=nu,file=file_data,access=direct,recl=4*imfrc*jmfrc*kmfrc)
do m = 1, imn
  write(unit=nu,rec=m) ttlev(:,:,:,m)
end do
close(nu)

! longitude/latitude of main data
if (linterp) then  ! If input data is horizontally interpolated in the model.
  open (unit=nu,file=file_grid)
  write(nu) alonf, alatf
  if (linterp_v) then ! If input data is vertically interpolated in the model.
    write(nu) dpf
  end if
  close(nu)
end if
```

b. Two dimensional reference state for surface restoring

When the surface field of a tracer is intended to be restored to a reference state, the attributes of the surface reference state should be given by the variables listed on Table 13.6.

Table13.6 Namelist `nml_tracer_data` related to reference values for surface restoring forcing.

variable name	units	description	usage
`trcref_surf_conf%file_data`	character	a file name that contains reference values for surface restoring forcing	
`trcref_surf_conf%file_data_grid`	character	a file name of grid	needed if `linterp` = `.true.`
`trcref_surf_conf%imfrc`	integer	grid size of data in x direction	
`trcref_surf_conf%jmfrc`	integer	grid size of data in y direction	
`trcref_surf_conf%interval`	integer	regular time interval of data	positive value : unit is sec −1 : monthly −999 : steady forcing
`trcref_surf_conf%num_data_max`	integer	the number of record contained in the file	
`trcref_surf_conf%ifstart`	ifstart(6)	[ymdhms] of the first record of the input file	1999,1,1,0,0,0 when the first record is the average value of Jan 1999 and its data interval is monthly.
`trcref_surf_conf%lrepeat`	logical	climatological data is repeatedly used	default = `.false.`
`trcref_surf_conf%linterp`	logical	interpolate horizontally or not	default = `.false.`
`trcref_surf_conf%ilinear`	integer	interpolation method	1 : linear, 2 : spline
`trcref_surf_conf%luniform`	logical	data is horizontally uniform or not	default = `.false.`
`trcref_surf_conf%ldouble`	logical	input data is double or not	default = `.false.`
`trcref_surf_conf%iverbose`	integer	standard output of progress	1 : extensive, 0 : minimum
`trcref_surf_conf%ldefined`	logical	the input data is defined or not	default = `.false.`

13.3.5 Restoring coefficient

a. Coefficient for three dimensional restoring

When the field of a tracer is restored to a reference state, the attributes of the file that contains restoring coefficients should be given by the variables listed on Table 13.7. Units of restoring coefficient is sec^{-1}.

Table13.7 Namelist `nml_tracer_data` related to restoring coefficient for body forcing.

variable name	units	description	usage
`rstcoef_conf%file_data`	character	a file name that contains restoring coefficient for body forcing.	
`rstcoef_conf%file_data_grid`	character	a file name of grid	needed if `linterp` = `.true.`
`rstcoef_conf%imfrc`	integer	grid size of data in x direction	
`rstcoef_conf%jmfrc`	integer	grid size of data in y direction	
`rstcoef_conf%kmfrc`	integer	grid size of data in z direction	
`rstcoef_conf%interval`	integer	regular time interval of data	positive value : unit is sec

Continued on next page

Table 13.7 – continued from previous page

variable name	units	description	usage
			−1 : monthly −999 : steady forcing
rstcoef_conf%num_data_max	integer	the number of record contained in the file	
rstcoef_conf%ifstart	ifstart(6)	[ymdhms] of the first record of the input file	1999,1,1,0,0,0 when the first record is the average value of Jan 1999 and its data interval is monthly.
rstcoef_conf%lrepeat	logical	climatological data is repeatedly used	default = .false.
rstcoef_conf%linterp	logical	interpolate horizontally or not	default = .false.
rstcoef_conf%linterp_v	logical	interpolate vertically or not	default = .false.
rstcoef_conf%ilinear	integer	interpolation method	1 : linear, 2 : spline
rstcoef_conf%luniform	logical	data is horizontally uniform or not	default = .true.
rstcoef_conf%luniform_v	logical	data is vertically uniform or not	default = .false.
rstcoef_conf%ldouble	logical	input data is double or not	default = .true.
rstcoef_conf%iverbose	logical	standard output of progress	1 : extensive 0 : minimum
rstcoef_conf%ldefined	logical	the input data is defined or not	default = .false.

Note that the default settings for luniform and ldouble are differ from those of the other attributes.

b. Coefficient for surface restoring

When the surface field of a tracer is intended to be restored to a reference state, the attributes of the file that contains surface restoring coefficients should be given by the variables listed on Table 13.6. Units of the surface restoring coefficient is sec^{-1}.

Table13.8 Namelist nml_tracer_data related to restoring coefficient for surface restoring forcing.

variable name	units	description	usage
rstcoef_surf_conf%file_data	character	a file name that contains restoring coefficient for for surface restoring forcing.	
rstcoef_surf_conf%file_data_grid	character	a file name of grid	needed if linterp = .true.
rstcoef_surf_conf%imfrc	integer	grid size of data in x direction	
rstcoef_surf_conf%jmfrc	integer	grid size of data in y direction	
rstcoef_surf_conf%interval	integer	regular time interval of data	positive value : unit is sec −1 : monthly −999 : steady forcing
rstcoef_surf_conf%num_data_max		the number of record contained in the file	
rstcoef_surf_conf%ifstart	ifstart(6)	[ymdhms] of the first record of the input file	1999,1,1,0,0,0 when the first record is the average value of Jan 1999 and its data interval is monthly.
rstcoef_surf_conf%lrepeat	logical	climatological data is repeatedly used	default = .false.
rstcoef_surf_conf%linterp	logical	interpolate horizontally or not	default = .false.
rstcoef_surf_conf%ilinear	integer	interpolation method	1 : linear, 2 : spline
rstcoef_surf_conf%luniform	logical	data is horizontally uniform or not	default = .false.
rstcoef_surf_conf%ldouble	logical	input data is double or not	default = .false.
rstcoef_surf_conf%iverbose	logical	standard output of progress	1 : extensive, 0 : minimum
rstcoef_surf_conf%ldefined	logical	the input data is defined or not	default = .false.

13.3.6 Example

Following is an example of namelist `nml_tracer_data` for Salinity. Some systems may not allow blank lines or comment lines in a namelist. In this case, you should delete them.

```
&nml_tracer_data
 name="Salinity",

! advection scheme
 adv_scheme%name="som",
 adv_scheme%limiter_som_org=.false.,
 adv_scheme%limiter_som_Merryfield03=.true.,
 adv_scheme%lrstin_som=.false.,
 adv_scheme%lrstout_som=.true.,

! restore_condition
 restore_conf%l_surf_restore=.true.
 restore_conf%l_body_restore=.false.

! trcref
 trcref_conf%file_data='../data/file_sclim.grd',
 trcref_conf%file_data_grid='dummy.d',
 trcref_conf%imfrc=184,
 trcref_conf%jmfrc=152,
 trcref_conf%kmfrc=51,
 trcref_conf%interval=-1,
 trcref_conf%ifstart=1947,12,1,0,0,0,
 trcref_conf%num_data_max=14,
 trcref_conf%lrepeat=.false.,
 trcref_conf%linterp=.false.,
 trcref_conf%ilinear=1,
 trcref_conf%iverbose=1,

! rstcoef
 rstcoef_conf%ldefined=.false.,

! trcref_surf
 trcref_surf_conf%file_data='../data/file_ssurf.grd',
 trcref_surf_conf%file_data_grid='dummy.d',
 trcref_surf_conf%imfrc=184,
 trcref_surf_conf%jmfrc=152,
 trcref_surf_conf%interval=-1,
 trcref_surf_conf%ifstart=1947,12,1,0,0,0,
 trcref_surf_conf%num_data_max=14,
 trcref_surf_conf%lrepeat=.false.,
 trcref_surf_conf%linterp=.false.,
 trcref_surf_conf%ilinear=1,
 trcref_surf_conf%iverbose=1

! rstcoef_surf
 rstcoef_surf_conf%file_data='../data/rstcoef_surf_s.grd',
 rstcoef_surf_conf%file_data_grid='dummy.d',
 rstcoef_surf_conf%imfrc=1,
 rstcoef_surf_conf%jmfrc=1,
 rstcoef_surf_conf%interval=-999,
 rstcoef_surf_conf%num_data_max=1,
 rstcoef_surf_conf%iverbose=1,
 rstcoef_surf_conf%luniform=.true.,
 rstcoef_surf_conf%ldouble=.true.,
/
```