# Part IV

# Evolution of tracers

Chapter 9

# Tracer advection-diffusion equation

In this chapter, we explain the advection-diffusion equation, which is the governing equation of the tracers including potential temperature and salinity.

## 9.1 The advection-diffusion equation

The equations for potential temperature and salinity, Eqs. (2.35) and (2.36), are re-written:

$$\frac{\partial(z_s\theta)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi u\theta)}{\partial\mu}+\frac{\partial(z_s h_\mu v\theta)}{\partial\psi}\right\}+\frac{\partial(z_s\dot{s}\theta)}{\partial s}=-z_s\nabla\cdot\mathbf{F}_\theta+z_sQ^\theta, \tag{9.1}$$

$$\frac{\partial(z_sS)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uS)}{\partial\mu}+\frac{\partial(z_s h_\mu vS)}{\partial\psi}\right\}+\frac{\partial(z_s\dot{s}S)}{\partial s}=-z_s\nabla\cdot\mathbf{F}_S+z_sQ^S. \tag{9.2}$$

Representing $\theta$ and $S$ with a general variable $T$, they become a common advection-diffusion equation,

$$\frac{\partial(z_sT)}{\partial t}+\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uT)}{\partial\mu}+\frac{\partial(z_s h_\mu vT)}{\partial\psi}\right\}+\frac{\partial(z_s\dot{s}T)}{\partial s}=-z_s\nabla\cdot\mathbf{F}_T+z_sQ^T, \tag{9.3}$$

where $Q^T$ indicates the source or sink term for $T$. If the advection term is expressed as $\mathcal{A}(T)$, the horizontal diffusion term $\mathcal{D}_H(T)$, and the vertical diffusion term $\mathcal{D}_V(T)$,

$$\frac{\partial(z_sT)}{\partial t}=\mathcal{A}(T)+\mathcal{D}_H(T)+\mathcal{D}_V(T)+z_sQ^T \tag{9.4}$$

$$\mathcal{A}(T)=-\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi uT)}{\partial\mu}+\frac{\partial(z_s h_\mu vT)}{\partial\psi}\right\}-\frac{\partial(z_s\dot{s}T)}{\partial s} \tag{9.5}$$

$$\mathcal{D}_H(T)=-\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_sF^T_\mu)}{\partial\mu}+\frac{\partial(h_\mu z_sF^T_\psi)}{\partial\psi}\right\} \tag{9.6}$$

$$\mathcal{D}_V(T)=-\frac{\partial F^T_z}{\partial s}. \tag{9.7}$$

Chapter 10, 11 and 12 explain the numerical treatments of $\mathcal{A}(T)$, $\mathcal{D}_H(T)$ and $\mathcal{D}_V(T)$, respectively. It should be noted that horizontal diffusion and vertical diffusion are treated together in the derivation of isopycnal diffusion (Sec. 11.3) as

$$\mathcal{D}_(T)=\mathcal{D}_H(T)+\mathcal{D}_v(T). \tag{9.8}$$

(In this case as well, diffusion is divided into $\mathcal{D}_H(T)$ and $\mathcal{D}_V(T)$ in the program code.)

Since the hydrostatic approximation is used, an unstable stratification should be removed somehow. Generally, we assume that vertical convection occurs instantaneously to remove unstable stratification. We call this convective adjustment, which is explained in Section 12.2. One might also choose to mix tracers by setting the local vertical diffusion coefficient to a large value such as 10 000 cm$^2$ s$^{-1}$ where stratification is unstable. In this case, the tracer equation should be solved using the partial implicit method, which is described in Section 23.5.

## 9.2 Boundary conditions

The boundary conditions of tracers can also be represented in common. The sea surface and bottom boundary conditions for $\theta$ and $S$ are shown by Eqs. (2.23), (2.24) and (2.25). These can be generally expressed as follows:

$$\kappa_V \left.\frac{\partial T}{\partial z}\right|_{z^*=0} = F^T_{\text{surf}}, \tag{9.9}$$

$$-\kappa_V \left.\frac{\partial T}{\partial z}\right|_{z^*=-H} = F^T_{\text{bottom}}, \tag{9.10}$$

where $F^T_{\text{surf}}$ is the surface flux (positive downward), while $F^T_{\text{bottom}}$ is the bottom flux (positive upward). At side walls, the boundary condition (Eq. 2.26) is given for the $n$ direction, which is perpendicular to the wall, as

$$\frac{\partial T}{\partial n} = 0. \tag{9.11}$$

## 9.3 Various tracers

Tracers governed by the advection-diffusion equation are not only $\theta$ and $S$. Passive tracers such as CFCs and age tracers (Chapter 20), and state variables of biogeochemical models (Chapter 19) are also tracers. Though time evolution of them is obtained by the common governing equation, Eq. (9.3), the boundary conditions and $Q^T$ are different. In addition, appropriate numerical schemes for advection and diffusion can change depending on tracer characteristics. MRI.COM allows individual flexible specifications for various tracers in the model. See Chapter 13 for details.

# Chapter 10

# Tracer advection schemes

This chapter describes the following tracer advection schemes available in MRI.COM.

- The weighted upcurrent scheme (Section 10.2)
- The Quadratic Upstream Interpolation for Convective Kinematics (QUICK; Leonard, 1979, Section 10.3)
- A combination of the QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979, Section 10.3) for vertical advection and the Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA; Leonard et al., 1993, Section 10.4) for horizontal advection
- The Second Order Moment (SOM; Prather, 1986, Section 10.5) scheme
- The Piecewise Parabolic Method (PPM; Colella and Woodward, 1984, Section 10.6) scheme
- The Multidimensional Positive Definite Advection Transport Algorithm (MPDATA; Smolarkiewicz and Margolin, 1998, Section 10.7) scheme

The default advection scheme for MRI.COM is a weighted upcurrent scheme, and the others are optional. Different advection schemes can be used for individual tracers. They should be chosen from among compiled schemes at run time (Chapter 13).

## 10.1 Finite volume or flux form method

Removing terms except for the time tendency and advection terms from Eq. (9.4), we gain the following three-dimensional advection equation:

$$\frac{\partial(z_s T)}{\partial t} = \mathcal{A}(T), \tag{10.1}$$

or

$$\frac{\partial(z_s T)}{\partial t} = -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(z_s h_\psi u T)}{\partial \mu} + \frac{\partial(z_s h_\mu v T)}{\partial \psi}\right\} - \frac{\partial(z_s \dot{s} T)}{\partial s}. \tag{10.2}$$

The finite difference form of (10.1) is given by first considering a control cell volume and then calculating fluxes through the cell faces, and setting their divergence and convergence to be the time change rate at the grid cell (Figure 10.1). In finite difference form, this is expressed as follows:

$$\begin{aligned} T^{n+1}_{i,j,k-\frac{1}{2}} \Delta V^{n+1}_{i,j,k-\frac{1}{2}} &= T^{n}_{i,j,k-\frac{1}{2}} \Delta V^{n}_{i,j,k-\frac{1}{2}} \\ &+ \Delta t\, \{FXA_{i-\frac{1}{2},j,k-\frac{1}{2}} - FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} + FYA_{i,j-\frac{1}{2},k-\frac{1}{2}} - FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} + FZA_{i,j,k} - FZA_{i,j,k-1}\}, \end{aligned} \tag{10.3}$$

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

As explained in Chapter 3, $\Delta V$ varies with time and is given by (3.28),

$$\begin{aligned} \Delta V_{i,j,k-\frac{1}{2}} \equiv (\texttt{volt})_{i,j,k-\frac{1}{2}} &= (\texttt{volu\_bl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{volu\_tl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ &+ (\texttt{volu\_br})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{volu\_tr})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}. \end{aligned} \tag{10.4}$$

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:

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} = U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{10.5}$$

$$FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} = V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{10.6}$$

$$FZA_{i,j,k} = W^T_{i,j,k} T_{i,j,k}, \tag{10.7}$$

where horizontal volume transport $U^T$ and $V^T$ are defined as follows,

$$U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta y_{i+\frac{1}{2},j}}{2}\left(u_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + u_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right), \tag{10.8}$$

$$V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{\Delta x_{i,j+\frac{1}{2}}}{2}\left(v_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + v_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right). \tag{10.9}$$

Vertical volume transport $W^T$ is then obtained by diagnostically solving (6.9) with (6.11). Moreover, the vertical velocity $w$, which is necessary for using QUICKEST, is calculated as follows ($w$ is not needed except for QUICKEST):

$$W^T_{i,j,k} = w_{i,j,k} \times (\texttt{areat})_{i,j,k-\frac{1}{2}}, \tag{10.10}$$

where

$$\begin{aligned}(\texttt{areat})_{i,j,k-\frac{1}{2}} = &(\texttt{a\_tr})_{i-\frac{1}{2},j-\frac{1}{2}} \times (\texttt{aexl})_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + (\texttt{a\_tl})_{i+\frac{1}{2},j-\frac{1}{2}} \times (\texttt{aexl})_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} \\ &+ (\texttt{a\_br})_{i-\frac{1}{2},j+\frac{1}{2}} \times (\texttt{aexl})_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + (\texttt{a\_bl})_{i+\frac{1}{2},j+\frac{1}{2}} \times (\texttt{aexl})_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}.\end{aligned} \tag{10.11}$$

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

## 10.2 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:

$$T^{upcurrent}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \begin{cases} T_{i,j,k-\frac{1}{2}}, & \text{if} \quad U_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\ T_{i+1,j,k-\frac{1}{2}}, & \text{if} \quad U_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0. \end{cases} \tag{10.12}$$

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

$$T^{center}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{T_{i+1,j,k-\frac{1}{2}} + T_{i,j,k-\frac{1}{2}}}{2}. \tag{10.13}$$

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

$$\begin{aligned} FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} =& \alpha\left\{\frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} + \left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}{2}T_i + \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} - \left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}{2}T_{i+1}\right\} \\ &+ (1-\alpha)U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\frac{T_{i+1,j,k-\frac{1}{2}} + T_{i,j,k-\frac{1}{2}}}{2} && (10.14) \\ =& U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\left\{\frac{1}{2}\left(1 + \alpha\frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}\right)T_i + \frac{1}{2}\left(1 - \alpha\frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}\right)T_{i+1}\right\} && (10.15) \\ =& U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\left\{\left[\frac{1}{2} + \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}\left(\beta - \frac{1}{2}\right)\right]T_i + \left[\frac{1}{2} - \frac{U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\left|U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|}\left(\beta - \frac{1}{2}\right)\right]T_{i+1}\right\}. && (10.16) \end{aligned}$$

Mainly for the sake of computational efficiency, we give a parameter $\beta = \frac{1}{2}(\alpha + 1)$ $(0.5 \le \beta \le 1)$ instead of $\alpha$ at run time. Different parameters for horizontal and vertical directions may be given, which are listed on Table 10.1.

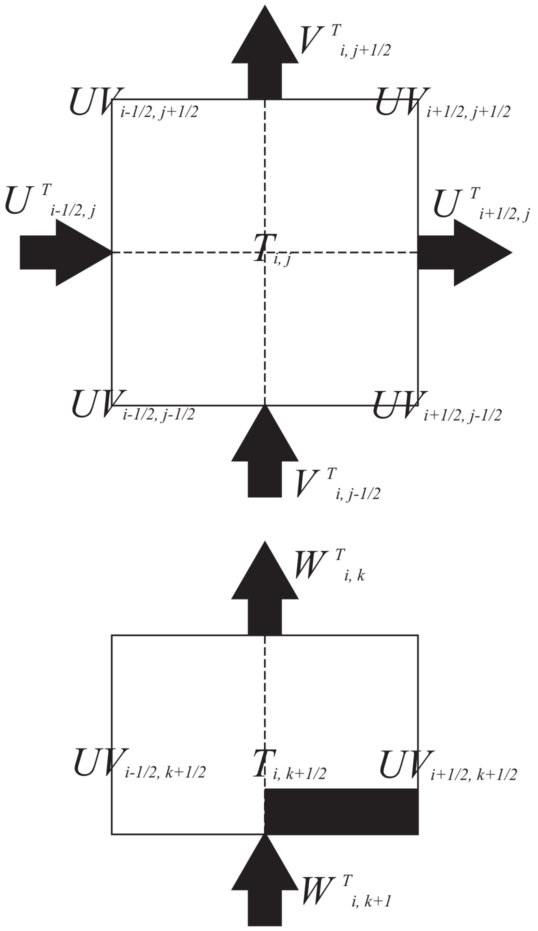

Figure10.1 Grid arrangement around TS-Box (Upper: Views from the upper, Below: views from the horizontal). Fluxes represented by an arrow are calculated.

Table10.1 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 |

## 10.3 QUICK and QUICKEST schemes

This section explains the QUICK and QUICKEST schemes. 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 10.2).

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

$$T_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta x_i T_{i+1,j,k-\frac{1}{2}} + \Delta x_{i+1} T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+1} + \Delta x_i} - \frac{\Delta x_{i+1}\Delta x_i}{4} c_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{10.17}$$

$$T_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{\Delta y_j T_{i,j+1,k-\frac{1}{2}} + \Delta y_{j+1} T_{i,j,k-\frac{1}{2}}}{\Delta y_{j+1} + \Delta y_j} - \frac{\Delta y_{j+1}\Delta y_j}{4} d_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{10.18}$$

$$T_{i,j,k} = \frac{\Delta z_{k-\frac{1}{2}} T_{i,j+1,k+\frac{1}{2}} + \Delta z_{k+\frac{1}{2}} T_{i,j,k-\frac{1}{2}}}{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}} - \frac{\Delta z_{k-\frac{1}{2}}\Delta z_{k+\frac{1}{2}}}{4} e_{i,j,k}, \tag{10.19}$$

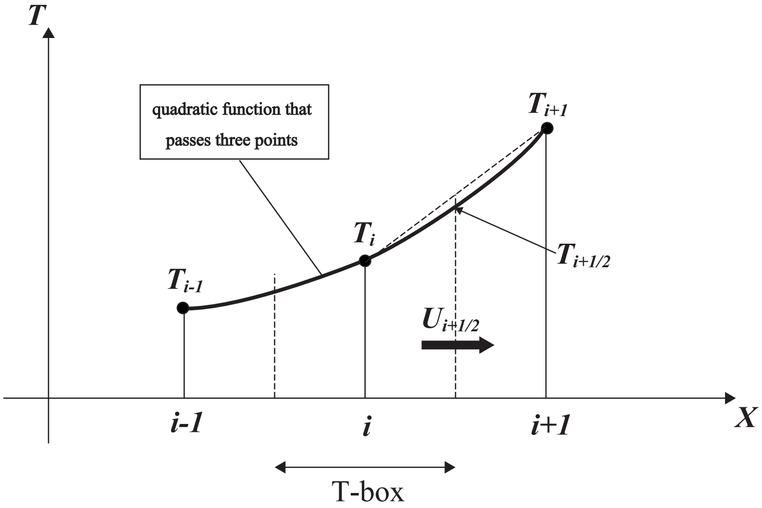

Figure10.2 Schematic for interpolation: $T_i$ represents the value of a tracer at T-point with index $i$, and $T_{i+\frac{1}{2}}$ is the cell boundary value. In the QUICK scheme, $T_{i+\frac{1}{2}}$ is interpolated by a quadratic function that passes $T_i$ and the neighboring T-point values, $T_{i-1}$ and $T_{i+1}$.

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

$$\begin{aligned}
c_{i+\frac{1}{2},j,k-\frac{1}{2}} &= \frac{\Delta x_i \delta_x \delta_x T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta x_i}^x} (\equiv c_p), && \text{if} \quad U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\
&= \frac{\Delta x_{i+1} \delta_x \delta_x T_{i+1,j,k-\frac{1}{2}}}{2\overline{\Delta x_{i+1}}^x} (\equiv c_m), && \text{if} \quad U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0, \\
d_{i,j+\frac{1}{2},k-\frac{1}{2}} &= \frac{\Delta y_j \delta_y \delta_y T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta y_j}^y} (\equiv d_p), && \text{if} \quad V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} > 0, \\
&= \frac{\Delta y_{j+1} \delta_y \delta_y T_{i,j+1,k-\frac{1}{2}}}{2\overline{\Delta y_{j+1}}^y} (\equiv d_m), && \text{if} \quad V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} < 0, \\
e_{i,j,k} &= \frac{\Delta z_{k+\frac{1}{2}} \delta_z \delta_z T_{i,j,k+\frac{1}{2}}}{2\overline{\Delta z_{i,j,k+\frac{1}{2}}}^z} (\equiv e_p), && \text{if} \quad W^T_{i,j,k} > 0, \\
&= \frac{\Delta z_{k-\frac{1}{2}} \delta_z \delta_z T_{i,j,k-\frac{1}{2}}}{2\overline{\Delta z_{i,j,k-\frac{1}{2}}}^z} (\equiv e_m), && \text{if} \quad W^T_{i,j,k} < 0.
\end{aligned} \tag{10.20}$$

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

$$\begin{aligned}
\delta_x A_i &\equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_i}, \quad \delta_x A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_i}{\Delta x_{i+\frac{1}{2}}}, \\
\overline{A_i}^x &\equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \quad \overline{A_{i+\frac{1}{2}}}^x \equiv \frac{A_{i+1} + A_i}{2}.
\end{aligned} \tag{10.21}$$

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

$$c_a = c_m + c_p \tag{10.22}$$

$$c_d = c_m - c_p \tag{10.23}$$

$$d_a = d_m + d_p \tag{10.24}$$

$$d_d = d_m - d_p \tag{10.25}$$

$$e_a = e_m + e_p \tag{10.26}$$

$$e_d = e_m - e_p, \tag{10.27}$$

we obtain

$$\begin{aligned}
FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} &= U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} \left[ \frac{\Delta x_{i,j} T_{i+1,j,k-\frac{1}{2}} + \Delta x_{i+1,j} T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+1,j} + \Delta x_{i,j}} - \frac{\Delta x_{i+1,j} \Delta x_{i,j}}{8} c_{a_{i+\frac{1}{2},j,k-\frac{1}{2}}} \right] \\
&+ |U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}| \frac{\Delta x_{i+1,j} \Delta x_{i,j}}{8} c_{d_{i+\frac{1}{2},j,k-\frac{1}{2}}}, \qquad (10.28) \\
FYA_{i,j+\frac{1}{2},k-\frac{1}{2}} &= V^T_{i,j+\frac{1}{2},k-\frac{1}{2}} \left[ \frac{\Delta y_{i,j} T_{i,j+1,k-\frac{1}{2}} + \Delta y_{i,j+1} T_{i,j,k-\frac{1}{2}}}{\Delta y_{i,j+1} + \Delta y_{i,j}} - \frac{\Delta y_{i,j+1} \Delta y_{i,j}}{8} d_{a_{i,j+\frac{1}{2},k-\frac{1}{2}}} \right] \\
&+ |V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}| \frac{\Delta y_{i,j+1} \Delta y_{i,j}}{8} d_{d_{i,j+\frac{1}{2},k-\frac{1}{2}}}, \qquad (10.29) \\
FZA_{i,j,k} &= W^T_{i,j,k} \left[ \frac{\Delta z_{i,j,k-\frac{1}{2}} T_{i,j,k+\frac{1}{2}} + \Delta z_{i,j,k+\frac{1}{2}} T_{i,j,k-\frac{1}{2}}}{\Delta z_{i,j,k+\frac{1}{2}} + \Delta z_{i,j,k-\frac{1}{2}}} - \frac{\Delta z_{i,j,k+\frac{1}{2}} \Delta z_{i,j,k-\frac{1}{2}}}{8} e_{a_{i,j,k}} \right] \\
&+ |W^T_{i,j,k-\frac{1}{2}}| \frac{\Delta z_{i,j,k+\frac{1}{2}} \Delta z_{i,j,k-\frac{1}{2}}}{8} e_{d_{i,j,k}}. \qquad (10.30)
\end{aligned}$$

Equation (10.28) can be rewritten as

$$FXA_{i+\frac{1}{2},j,k-\frac{1}{2}} \simeq U^T_{i+\frac{1}{2},j,k-\frac{1}{2}} \tilde{T}_{i+\frac{1}{2},j,k-\frac{1}{2}} + A_Q \frac{\partial^3 T_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\partial x^3}, \qquad (10.31)$$

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

$$A_Q = |U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}| \frac{\Delta x_{i+1} \Delta x_{i+\frac{1}{2}} \Delta x_i}{8}. \qquad (10.32)$$

Although the time integration for advection is done by the leap-frog scheme, the second term on the r.h.s. of (10.31) 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 10.1).

The following paragraphs describe the specific expression and the accuracy of the QUICK with Estimated Streaming Terms (QUICKEST; Leonard, 1979).

Consider a one-dimensional equation of advection for incompressible fluid

$$\frac{\partial T}{\partial t} + \frac{\partial}{\partial z}(wT) = 0, \qquad (10.33)$$

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:

$$\Delta z_k = \frac{\Delta z_{k+\frac{1}{2}} + \Delta z_{k-\frac{1}{2}}}{2}. \qquad (10.34)$$

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

$$T_{k-\frac{3}{2}} = c_0 + c_1\left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right) + c_2\left(\frac{\Delta z_{k-\frac{3}{2}}}{2} + \Delta z_{k-\frac{1}{2}}\right)^2 + O(\Delta z^3), \tag{10.35}$$

$$T_{k-\frac{1}{2}} = c_0 + c_1\frac{\Delta z_{k-\frac{1}{2}}}{2} + c_2\frac{\Delta z^2_{k-\frac{1}{2}}}{4} + O(\Delta z^3), \tag{10.36}$$

$$T_{k+\frac{1}{2}} = c_0 - c_1\frac{\Delta z_{k+\frac{1}{2}}}{2} + c_2\frac{\Delta z^2_{k+\frac{1}{2}}}{4} + O(\Delta z^3), \tag{10.37}$$

$$T_{k+\frac{3}{2}} = c_0 - c_1\left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right) + c_2\left(\frac{\Delta z_{k+\frac{3}{2}}}{2} + \Delta z_{k+\frac{1}{2}}\right)^2 + O(\Delta z^3). \tag{10.38}$$

Coefficients $c_0$, $c_1$, and $c_2$ can be solved using three of the four equations (10.35), (10.36), (10.37), and (10.38). The three upstream-side equations are chosen. When $w > 0$ ($w < 0$), equations (10.36), (10.37), and (10.38) ((10.35), (10.36), and (10.37)) are used. The solution is as follows:

$$c_0 = \frac{T_{k-\frac{1}{2}}\Delta z_{k+\frac{1}{2}} + T_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{2\Delta z_k} - \frac{\Delta z_{k+\frac{1}{2}}\Delta z_{k-\frac{1}{2}}}{4}c_2, \tag{10.39}$$

$$c_1 = \frac{T_{k-\frac{1}{2}} - T_{k+\frac{1}{2}}}{\Delta z_k} - \frac{\Delta z_{k-\frac{1}{2}} - \Delta z_{k+\frac{1}{2}}}{2}c_2, \tag{10.40}$$

$$c_2 = \begin{cases} \frac{1}{\Delta z_k+\Delta z_{k+1}}\left(\frac{T_{k-\frac{1}{2}}-T_{k+\frac{1}{2}}}{\Delta z_k} - \frac{T_{k+\frac{1}{2}}-T_{k+\frac{3}{2}}}{\Delta z_{k+1}}\right) & (w > 0), \\ \frac{1}{\Delta z_{k-1}+\Delta z_k}\left(\frac{T_{k-\frac{3}{2}}-T_{k-\frac{1}{2}}}{\Delta z_{k-1}} - \frac{T_{k-\frac{1}{2}}-T_{k+\frac{1}{2}}}{\Delta z_k}\right) & (w < 0). \end{cases} \tag{10.41}$$

Next, equation (10.33) is integrated over one time step and one grid cell.

$$\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz\frac{\partial T}{\partial t} = -\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz\frac{\partial}{\partial z}(wT). \tag{10.42}$$

The r.h.s. of (10.42) can be written as

$$-\int_{t^n}^{t^{n+1}} dt(w_uT_u - w_lT_l), \tag{10.43}$$

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

$$\int_{t^n}^{t^{n+1}} dtT_l = \int_{-w_l\Delta t}^{0} [c_0^n + c_1^n\xi + c_2^n\xi^2 + O(\Delta z^3)]\frac{d\xi}{w_l}. \tag{10.44}$$

Thus expression (10.43) becomes

$$-\Delta t(w_u\overline{T_u^n} - w_l\overline{T_l^n}) + O(\Delta z^3 w\Delta t), \tag{10.45}$$

where

$$\begin{aligned}\overline{T_l^n} &= \frac{1}{w_l\Delta t}\int_{-w_l\Delta t}^{0}(c_0^n + c_1^n\xi + c_2^n\xi^2)d\xi \\ &= c_0^n - \frac{c_1^n}{2}w_l\Delta t + \frac{c_2^n}{3}w_l^2\Delta t^2.\end{aligned} \tag{10.46}$$

Using up to the second order terms of a Taylor expansion, the l.h.s. of (10.42) can be written as follows:

$$\int_{t^n}^{t^{n+1}} dt \int_{z_k}^{z_{k-1}} dz\frac{\partial T}{\partial t} = \Delta z_{k-\frac{1}{2}}\left[T^{n+1}_{k-\frac{1}{2}} - T^n_{k-\frac{1}{2}} + \frac{\Delta z^2_{k-\frac{1}{2}}}{24}(T_{zz}{}^{n+1}_{k-\frac{1}{2}} - T_{zz}{}^{n}_{k-\frac{1}{2}}) + O(\Delta z^3)\right], \tag{10.47}$$

where

$$
\begin{aligned}
{T_{zz}}^{n+1}_{k-\frac{1}{2}} - {T_{zz}}^{n}_{k-\frac{1}{2}} &= \Delta t \left.\frac{\partial T_{zz}}{\partial t}\right|^{n}_{k-\frac{1}{2}} + O(\Delta t^2) \\
&= -\Delta t \frac{\partial^2}{\partial z^2}\left[\frac{\partial}{\partial z}(wT)\right]^{n}_{k-\frac{1}{2}} + O(\Delta t^2) \\
&= -\Delta t \frac{\partial}{\partial z}(wT_{zz})^{n}_{k-\frac{1}{2}} + O(\Delta t^2) \\
&= -\Delta t \left\{\frac{w_u {T_{zz}}^{n}_{u} - w_l {T_{zz}}^{n}_{l}}{\Delta z_{k-\frac{1}{2}}}\right\} + O(w\Delta t \Delta z) + O(\Delta t^2).
\end{aligned}
\tag{10.48}
$$

The expression for the r.h.s. of (10.47) becomes

$$
\Delta z_{k-\frac{1}{2}}\left[T^{n+1}_{k-\frac{1}{2}} - T^{n}_{k-\frac{1}{2}} - \frac{\Delta z^2_{k-\frac{1}{2}}}{24}\Delta t \frac{w_u {T_{zz}}^{n}_{u} - w_l {T_{zz}}^{n}_{l}}{\Delta z_{k-\frac{1}{2}}} + O(\Delta z^3)\right] + O(w\Delta t \Delta z^4) + O(\Delta z^3 \Delta t^2). \tag{10.49}
$$

Based on (10.45) and (10.49), the discretized forecasting equation is expressed as follows:

$$
T^{n+1}_{k-\frac{1}{2}} = T^{n}_{k-\frac{1}{2}} - \frac{\Delta t}{\Delta z_{k-\frac{1}{2}}}\left[w_u \overline{T^n_u} - w_l \overline{T^n_l} - \frac{\Delta z^2_{k-\frac{1}{2}}}{24}(w_u {T_{zz}}^{n}_{u} - w_l {T_{zz}}^{n}_{l})\right] + O(\alpha \Delta z^4) + O(\Delta z^2 \Delta t^2), \tag{10.50}
$$

where

$$
\alpha \equiv \frac{w\Delta t}{\Delta z} < 1, \tag{10.51}
$$

$$
{T_{zz}}^{n}_{l} = 2c_2 + O(\Delta z). \tag{10.52}
$$

The accuracy of equation (10.50) is $\max(O(\Delta z^4), O(\Delta z^2 \Delta t^2))$.

Note that MRI.COM adopts a flux limiter proposed by Leonard et al. (1994), which prevents unrealistic extrema, for the vertical QUICKEST scheme.

## 10.4 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:

$$
\frac{\partial T}{\partial t} + \frac{1}{h_\mu h_\psi}\frac{\partial}{\partial \mu}(h_\psi u T) + \frac{1}{h_\mu h_\psi}\frac{\partial}{\partial \psi}(h_\mu v T) = 0. \tag{10.53}
$$

Integrated over a tracer cell and for one time step,

$$
\int_{\psi_L-\Delta\psi_L/2}^{\psi_L+\Delta\psi_L/2} d\psi \int_{\mu_L-\Delta\mu_L/2}^{\mu_L+\Delta\mu_L/2} d\mu (\chi^{n+1} - \chi^n) = -\Delta t (u^n_r \overline{T^n_r}\Delta y_r - u^n_l \overline{T^n_l}\Delta y_l + v^n_u \overline{T^n_u}\Delta x_u - v^n_d \overline{T^n_d}\Delta x_d), \tag{10.54}
$$

where $\chi \equiv h_\mu h_\psi T$ and $\overline{T^n_r}$ etc. on the r.h.s. are the face values described later. On the l.h.s. of (10.54), the second-order interpolation of $\chi$ is used to integrate the terms. The Taylor expansion of $\chi$ about L is given as follows (see Figure 10.3 for the label of the point):

$$
\chi = \chi_L + a_{10}(\mu - \mu_L) + a_{20}(\mu - \mu_L)^2 + a_{01}(\psi - \psi_L) + a_{02}(\psi - \psi_L)^2 + a_{11}(\mu - \mu_L)(\psi - \psi_L). \tag{10.55}
$$

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

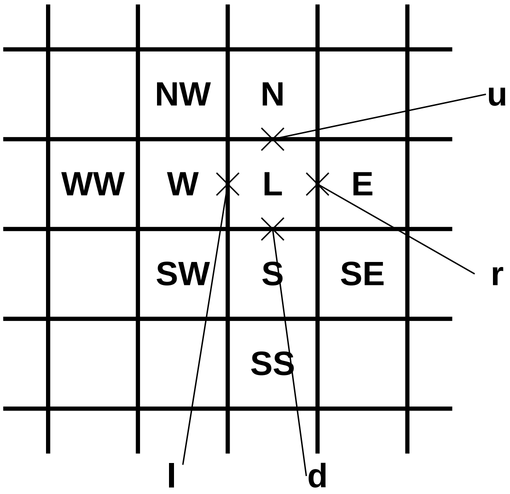

Figure10.3 Labels of tracer grid points (upper case characters) and faces (lower case characters).

$$\chi_E = \chi_L + a_{10}\Delta\mu_r + a_{20}\Delta\mu_r^2, \tag{10.56}$$

$$\chi_W = \chi_L - a_{10}\Delta\mu_l + a_{20}\Delta\mu_l^2, \tag{10.57}$$

$$\chi_N = \chi_L + a_{01}\Delta\psi_u + a_{02}\Delta\psi_u^2, \tag{10.58}$$

$$\chi_S = \chi_L - a_{01}\Delta\psi_d + a_{02}\Delta\psi_d^2, \tag{10.59}$$

where

$$\Delta\psi_u \equiv \frac{\Delta\psi_L + \Delta\psi_N}{2}, \tag{10.60}$$

$$\Delta\psi_d \equiv \frac{\Delta\psi_L + \Delta\psi_S}{2}, \tag{10.61}$$

$$\Delta\mu_r \equiv \frac{\Delta\mu_L + \Delta\mu_E}{2}, \tag{10.62}$$

$$\Delta\mu_l \equiv \frac{\Delta\mu_L + \Delta\mu_W}{2}. \tag{10.63}$$

Using these known values, the following parameters are obtained,

$$a_{10} = \frac{\Delta\mu_l \dfrac{\chi_E - \chi_L}{\Delta\mu_r} + \Delta\mu_r \dfrac{\chi_L - \chi_W}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}, \tag{10.64}$$

$$a_{20} = \frac{\dfrac{\chi_E - \chi_L}{\Delta\mu_r} - \dfrac{\chi_L - \chi_W}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}, \tag{10.65}$$

$$a_{01} = \frac{\Delta\psi_d \dfrac{\chi_N - \chi_L}{\Delta\psi_u} + \Delta\psi_u \dfrac{\chi_L - \chi_S}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}, \tag{10.66}$$

$$a_{02} = \frac{\dfrac{\chi_N - \chi_L}{\Delta\psi_u} - \dfrac{\chi_L - \chi_S}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{10.67}$$

Substituting (10.55) into the l.h.s. of (10.54) yields

$$\Delta\mu_L\Delta\psi_L \left[ \chi_L^{n+1} - \chi_L^n + \frac{\Delta\mu_L^2}{12}(a_{20}^{n+1} - a_{20}^n) + \frac{\Delta\psi_L^2}{12}(a_{02}^{n+1} - a_{02}^n) \right]. \tag{10.68}$$

Using equation (10.53), the following approximation is allowed:

$$a^{n+1}_{20} - a^n_{20} = -\Delta t \left[ \frac{h_{\psi r} u^n_r T^n_{\mu\mu r} - h_{\psi l} u^n_l T^n_{\mu\mu l}}{\Delta\mu_L} + \frac{h_{\mu u} v^n_u T^n_{\mu\mu u} - h_{\mu d} v^n_d T^n_{\mu\mu d}}{\Delta\psi_L} \right], \tag{10.69}$$

$$a^{n+1}_{02} - a^n_{02} = -\Delta t \left[ \frac{h_{\psi r} u^n_r T^n_{\psi\psi r} - h_{\psi l} u^n_l T^n_{\psi\psi l}}{\Delta\mu_L} + \frac{h_{\mu u} v^n_u T^n_{\psi\psi u} - h_{\mu d} v^n_d T^n_{\psi\psi d}}{\Delta\psi_L} \right], \tag{10.70}$$

where $T^n_{\mu\mu r}$ 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,

$$T^{n+1}_L = T^n_L - \frac{\Delta t}{\Delta S_L}(u^n_r \tilde{T}^n_r \Delta y_r - u^n_l \tilde{T}^n_l \Delta y_l + v^n_u \tilde{T}^n_u \Delta x_u - v^n_d \tilde{T}^n_d \Delta x_d), \tag{10.71}$$

where

$$\tilde{T}^n_l = \overline{T^n_l} - \frac{\Delta\mu_L^2}{24} T^n_{\mu\mu l} - \frac{\Delta\psi_L^2}{24} T^n_{\psi\psi l}, \tag{10.72}$$

$$\tilde{T}^n_d = \overline{T^n_d} - \frac{\Delta\mu_L^2}{24} T^n_{\mu\mu d} - \frac{\Delta\psi_L^2}{24} T^n_{\psi\psi d}. \tag{10.73}$$

Next, the expressions for $\overline{T^n_l}$ and $\overline{T^n_d}$ are required. The term $\overline{T^n_l}$ is the average over the hatched area of Figure 10.4, and the values of $T^n$ are given as the second order interpolation about **l** of Figure 10.3. Similar operations will be used to obtain the expression for $\overline{T^n_d}$.

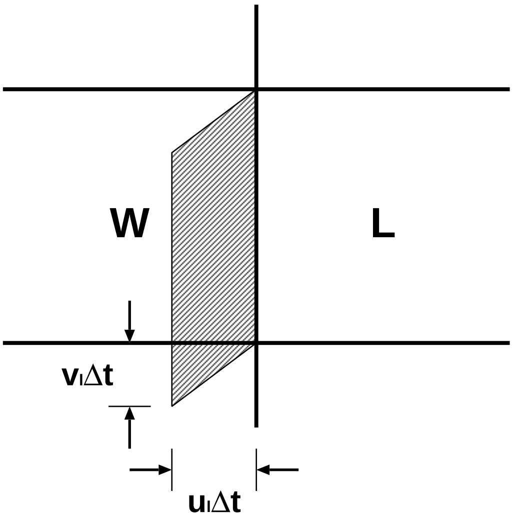

Figure10.4 Area used to average tracer values for the face **l**

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

$$T^n|_{\mathbf{l}} = c_{00} + c_{10}(\mu - \mu_l) + c_{20}(\mu - \mu_l)^2 + c_{01}(\psi - \psi_L) + c_{02}(\psi - \psi_L)^2 + c_{11}(\mu - \mu_l)(\psi - \psi_L), \tag{10.74}$$

$$T^n|_{\mathbf{d}} = d_{00} + d_{10}(\mu - \mu_l) + d_{20}(\mu - \mu_l)^2 + d_{01}(\psi - \psi_L) + d_{02}(\psi - \psi_L)^2 + d_{11}(\mu - \mu_l)(\psi - \psi_L). \tag{10.75}$$

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

$$\begin{aligned}
T_L^n &= c_{00} + c_{10}\frac{\Delta\mu_L}{2} + c_{20}\frac{\Delta\mu_L^2}{4}, && (10.76)\\
T_W^n &= c_{00} - c_{10}\frac{\Delta\mu_W}{2} + c_{20}\frac{\Delta\mu_W^2}{4}, && (10.77)\\
T_E^n &= c_{00} + c_{10}\left(\Delta\mu_L + \frac{\Delta\mu_E}{2}\right) + c_{20}\left(\Delta\mu_L + \frac{\Delta\mu_E}{2}\right)^2, && (10.78)\\
T_{WW}^n &= c_{00} - c_{10}\left(\Delta\mu_W + \frac{\Delta\mu_{WW}}{2}\right) + c_{20}\left(\Delta\mu_W + \frac{\Delta\mu_{WW}}{2}\right)^2, && (10.79)\\
T_N^n &= T_L^n + c_{01}\Delta\psi_u + c_{02}\Delta\psi_u^2 + c_{11}\frac{\Delta\mu_L}{2}\Delta\psi_u, && (10.80)\\
T_S^n &= T_L^n - c_{01}\Delta\psi_d + c_{02}\Delta\psi_d^2 - c_{11}\frac{\Delta\mu_L}{2}\Delta\psi_d, && (10.81)\\
T_{NW}^n &= T_W^n + c_{01}\Delta\psi_u + c_{02}\Delta\psi_u^2 - c_{11}\frac{\Delta\mu_W}{2}\Delta\psi_u, && (10.82)\\
T_{SW}^n &= T_W^n - c_{01}\Delta\psi_d + c_{02}\Delta\psi_d^2 + c_{11}\frac{\Delta\mu_W}{2}\Delta\psi_d. && (10.83)
\end{aligned}$$

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

$$\begin{aligned}
u_l^n > 0,\quad v_l^n > 0 &\Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{S}, \mathbf{NW}, \mathbf{SW} && (10.84)\\
u_l^n < 0,\quad v_l^n > 0 &\Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{SW} && (10.85)\\
u_l^n > 0,\quad v_l^n < 0 &\Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{WW}, \mathbf{N}, \mathbf{NW}, \mathbf{SW} && (10.86)\\
u_l^n < 0,\quad v_l^n < 0 &\Rightarrow \mathbf{L}, \mathbf{W}, \mathbf{E}, \mathbf{N}, \mathbf{S}, \mathbf{NW} && (10.87)
\end{aligned}$$

From equations (10.76) and (10.77),

$$\begin{aligned}
c_{00} &= \frac{\Delta\mu_W T_L^n + \Delta\mu_L T_W^n}{2\Delta\mu_l} - c_{20}\frac{\Delta\mu_L\Delta\mu_W}{4}, && (10.88)\\
c_{10} &= \frac{T_L^n - T_W^n}{\Delta\mu_l} - c_{20}\frac{\Delta\mu_L - \Delta\mu_W}{2}. && (10.89)
\end{aligned}$$

When $u_l^n > 0$, from (10.76) and (10.79),

$$c_{20} = \frac{\dfrac{T_L^n - T_W^n}{\Delta\mu_l} - \dfrac{T_W^n - T_{WW}^n}{\Delta\mu_{ll}}}{\Delta\mu_l + \Delta\mu_{ll}}, \qquad (10.90)$$

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

Using equations (10.82) and (10.83),

$$c_{02} = \frac{\dfrac{T_{NW}^n - T_W^n}{\Delta\psi_u} - \dfrac{T_W^n - T_{SW}^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \qquad (10.91)$$

When $u_l^n < 0$, from (10.77) and (10.78),

$$c_{20} = \frac{\dfrac{T_E^n - T_L^n}{\Delta\mu_r} - \dfrac{T_L^n - T_W^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \qquad (10.92)$$

Using equations (10.80) and (10.81),

$$c_{02} = \frac{\dfrac{T_N^n - T_L^n}{\Delta\psi_u} - \dfrac{T_L^n - T_S^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \qquad (10.93)$$

When $v_l^n > 0$, from (10.81) and (10.83),

$$c_{01} = \frac{\Delta\mu_W (T_L^n - T_S^n) + \Delta\mu_L (T_W^n - T_{SW}^n)}{2\Delta\mu_l \Delta\psi_d} + c_{02}\Delta\psi_d, \tag{10.94}$$

$$c_{11} = \frac{T_{SW}^n - T_W^n - T_S^n + T_L^n}{\Delta\mu_l \Delta\psi_d}. \tag{10.95}$$

When $v_l^n < 0$, from (10.80) and (10.82),

$$c_{01} = \frac{\Delta\mu_W (T_N^n - T_L^n) + \Delta\mu_L (T_{NW}^n - T_W^n)}{2\Delta\mu_l \Delta\psi_u} - c_{02}\Delta\psi_u, \tag{10.96}$$

$$c_{11} = \frac{T_N^n - T_L^n - T_{NW}^n + T_W^n}{\Delta\mu_l \Delta\psi_u}. \tag{10.97}$$

Next, using equation (10.75), the $T^n$ values at eight points around **d** are

$$T_L^n = d_{00} + d_{01}\frac{\Delta\psi_L}{2} + d_{02}\frac{\Delta\psi_L^2}{4}, \tag{10.98}$$

$$T_S^n = d_{00} - d_{01}\frac{\Delta\psi_S}{2} + d_{02}\frac{\Delta\psi_S^2}{4}, \tag{10.99}$$

$$T_N^n = d_{00} + d_{01}\left(\Delta\psi_L + \frac{\Delta\psi_N}{2}\right) + d_{02}\left(\Delta\psi_L + \frac{\Delta\psi_N}{2}\right)^2, \tag{10.100}$$

$$T_{SS}^n = d_{00} - d_{01}\left(\Delta\psi_S + \frac{\Delta\psi_{SS}}{2}\right) + d_{02}\left(\Delta\psi_S + \frac{\Delta\psi_{SS}}{2}\right)^2, \tag{10.101}$$

$$T_E^n = T_L^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 + d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_r, \tag{10.102}$$

$$T_W^n = T_L^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 - d_{11}\frac{\Delta\psi_L}{2}\Delta\mu_l, \tag{10.103}$$

$$T_{SE}^n = T_S^n + d_{10}\Delta\mu_r + d_{20}\Delta\mu_r^2 - d_{11}\frac{\Delta\psi_S}{2}\Delta\mu_r, \tag{10.104}$$

$$T_{SW}^n = T_S^n - d_{10}\Delta\mu_l + d_{20}\Delta\mu_l^2 + d_{11}\frac{\Delta\psi_S}{2}\Delta\mu_l. \tag{10.105}$$

From equations (10.98) and (10.99),

$$d_{00} = \frac{\Delta\psi_S T_L^n + \Delta\psi_L T_S^n}{2\Delta\psi_d} - d_{02}\frac{\Delta\psi_L \Delta\psi_S}{4}, \tag{10.106}$$

$$d_{01} = \frac{T_L^n - T_S^n}{\Delta\psi_d} - d_{02}\frac{\Delta\psi_L - \Delta\psi_S}{2}. \tag{10.107}$$

When $v_d^n > 0$, from (10.98) and (10.101),

$$d_{02} = \frac{\dfrac{T_L^n - T_S^n}{\Delta\psi_d} - \dfrac{T_S^n - T_{SS}^n}{\Delta\psi_{dd}}}{\Delta\psi_d + \Delta\psi_{dd}}, \tag{10.108}$$

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

From (10.104) and (10.105),

$$d_{20} = \frac{\dfrac{T_{SE}^n - T_S^n}{\Delta\mu_r} - \dfrac{T_S^n - T_{SW}^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{10.109}$$

When $v_d^n < 0$, from (10.99) and (10.100),

$$d_{02} = \frac{\dfrac{T_N^n - T_L}{\Delta\psi_u} - \dfrac{T_L^n - T_S^n}{\Delta\psi_d}}{\Delta\psi_u + \Delta\psi_d}. \tag{10.110}$$

From (10.102) and (10.103),

$$d_{20} = \frac{\dfrac{T_E^n - T_L^n}{\Delta\mu_r} - \dfrac{T_L^n - T_W^n}{\Delta\mu_l}}{\Delta\mu_r + \Delta\mu_l}. \tag{10.111}$$

When $u_d^n > 0$, from (10.103) and (10.105),

$$d_{10} = \frac{\Delta\psi_S(T_L^n - T_W^n) + \Delta\psi_L(T_S^n - T_{SW}^n)}{2\Delta\psi_d\Delta\mu_l} + d_{20}\Delta\mu_l, \tag{10.112}$$

$$d_{11} = \frac{T_L^n - T_S^n - T_W^n + T_{SW}^n}{\Delta\psi_d\Delta\mu_l}. \tag{10.113}$$

When $u_d^n < 0$, from (10.102) and (10.104),

$$d_{10} = \frac{\Delta\psi_S(T_E^n - T_L^n) + \Delta\psi_L(T_{SE}^n - T_S^n)}{2\Delta\psi_d\Delta\mu_r} - d_{20}\Delta\mu_r, \tag{10.114}$$

$$d_{11} = \frac{T_E^n - T_{SE}^n - T_L^n + T_S^n}{\Delta\psi_d\Delta\mu_r}. \tag{10.115}$$

The value of $\overline{T_l^n}$ is the average of $T^n$ over the hatched area of Figure 10.4. Defining

$$\xi_l^n = \frac{u_l^n}{h_{\mu l}}, \quad \eta_l^n = \frac{v_l^n}{h_{\psi l}}, \tag{10.116}$$

we have

$$\begin{aligned}
\overline{T_l^n} = & \frac{1}{\xi_l^n\Delta t\Delta\psi_L}\left[\int_{\psi_L-\Delta\psi_L/2}^{\psi_L+\Delta\psi_L/2}\int_{\mu_l-\xi_l^n\Delta t}^{\mu_l} T^n d\mu d\psi \right.\\
& \left. + \int_{\psi_L-\Delta\psi_L/2-\eta_l^n\Delta t}^{\psi_L-\Delta\psi_L/2}\int_{\mu_l-\xi_l^n\Delta t}^{\mu_l+\frac{\xi_l^n}{\eta_l^n}(\psi-\psi_L+\Delta\psi_L/2)} \{T^n(\psi) - T^n(\psi+\Delta\psi_L)\} d\mu d\psi\right]\\
= & c_{00} - \frac{1}{2}\eta_l^n\Delta t\, c_{01} + \left[\frac{1}{12}\Delta\psi_L^2 + \frac{1}{3}(\eta_l^n\Delta t)^2\right] c_{02}\\
& - \frac{1}{2}\xi_l^n\Delta t\, c_{10} + \frac{1}{3}(\xi_l^n\Delta t)^2 c_{20} + \frac{1}{3}\xi_l^n\eta_l^n\Delta t^2\, c_{11}.
\end{aligned} \tag{10.117}$$

Although the ranges of integrals in (10.117) assume $u_l^n > 0$ and $v_l^n > 0$, the result is independent of the signs of $u_l^n$ and $v_l^n$.

Similarly,

$$\begin{aligned}
\overline{T_d^n} = & d_{00} - \frac{1}{2}\eta_d^n\Delta t\, d_{01} + \frac{1}{3}(\eta_d^n\Delta t)^2\, d_{02}\\
& - \frac{1}{2}\xi_d^n\Delta t\, d_{10} + \left[\frac{1}{12}\Delta\mu_L^2 + \frac{1}{3}(\xi_d^n\Delta t)^2\right] d_{20} + \frac{1}{3}\xi_d^n\eta_d^n\Delta t^2 d_{11},
\end{aligned} \tag{10.118}$$

where

$$\xi_d^n = \frac{u_d^n}{h_{\mu d}}, \quad \eta_d^n = \frac{v_d^n}{h_{\psi d}}. \tag{10.119}$$

Therefore,

$$\begin{aligned}
\tilde{T_l^n} = & c_{00} - \frac{1}{2}\xi_l^n\Delta t\, c_{10} + \left[\frac{1}{3}(\xi_l^n\Delta t)^2 - \frac{\Delta\mu_L^2}{12}\right] c_{20}\\
& - \frac{1}{2}\eta_l^n\Delta t\, c_{01} + \frac{1}{3}(\eta_l^n\Delta t)^2 c_{02} + \frac{1}{3}\xi_l^n\eta_l^n\Delta t^2\, c_{11},
\end{aligned} \tag{10.120}$$

$$\begin{aligned}
\tilde{T_d^n} = & d_{00} - \frac{1}{2}\eta_d^n\Delta t\, d_{01} + \left[\frac{1}{3}(\eta_d^n\Delta t)^2 - \frac{\Delta\psi_L^2}{12}\right] d_{02}\\
& - \frac{1}{2}\xi_d^n\Delta t\, d_{10} + \frac{1}{3}(\xi_d^n\Delta t)^2 d_{20} + \frac{1}{3}\xi_d^n\eta_d^n\Delta t^2\, d_{11}.
\end{aligned} \tag{10.121}$$

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

$$(T_N^n - T_L^n)\Delta\psi_d^2 = (T_S^n - T_L^n)\Delta\psi_u^2. \tag{10.122}$$

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

$$(T_{NW}^n - T_W^n)\Delta\psi_d^2 = (T_{SW}^n - T_W^n)\Delta\psi_u^2 \tag{10.123}$$

When **WW** is a land grid,

$$(T_{WW}^n - T_W^n)\Delta\mu_l^2 = (T_L^n - T_W^n)\Delta\mu_{ll}^2. \tag{10.124}$$

When **E** is a land grid,

$$(T_E^n - T_L^n)\Delta\mu_l^2 = (T_W^n - T_L^n)\Delta\mu_r^2. \tag{10.125}$$

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

Note that MRI.COM adopts a flux limiter proposed by Leonard et al. (1994), which prevents unrealistic extrema, for the UTOPIA scheme.

## 10.5 Second Order Moment (SOM) scheme

### 10.5.1 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 ($0 \le x \le X, \quad 0 \le y \le Y, \quad 0 \le z \le Z$; volume $V = XYZ$) can be represented using second order functions as follows:

$$f(x,y,z) = a_0 + a_x x + a_{xx}x^2 + a_y y + a_{yy}y^2 + a_z z + a_{zz}z^2 + a_{xy}xy + a_{yz}yz + a_{zx}zx. \tag{10.126}$$

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

$$f(x,y,z) = m_0K_0 + m_xK_x + m_{xx}K_{xx} + m_yK_y + m_{yy}K_{yy} + m_zK_z + m_{zz}K_{zz} + m_{xy}K_{xy} + m_{yz}K_{yz} + m_{zx}K_{zx}, \tag{10.127}$$

where the orthogonal functions are given as follows:

$$\begin{aligned}
K_0 &= 1,\\
K_x(x) &= x - X/2,\\
K_{xx}(x) &= x^2 - Xx + X^2/6,\\
K_y(y) &= y - Y/2,\\
K_{yy}(y) &= y^2 - Yy + Y^2/6,\\
K_z(z) &= z - Z/2,\\
K_{zz}(z) &= z^2 - Zz + Z^2/6,\\
K_{xy}(x,y) &= (x - X/2)(y - Y/2),\\
K_{yz}(y,z) &= (y - Y/2)(z - Z/2),\\
K_{zx}(z,x) &= (z - Z/2)(x - X/2),
\end{aligned} \tag{10.128}$$

and

$$\int K_mK_n dV = 0 \quad (m \neq n). \tag{10.129}$$

The constants for normalization are determined using

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

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

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

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

Accordingly, the moments are set by the following expressions:

$$\begin{aligned}
S_0 &= \int f(x,y,z)K_0 dV = m_0 V,\\
S_x &= (6/X)\int f(x,y,z)K_x(x)dV = m_x VX/2,\\
S_{xx} &= (30/X^2)\int f(x,y,z)K_{xx}(x)dV = m_{xx}VX^2/6,\\
S_y &= (6/Y)\int f(x,y,z)K_y(y)dV = m_y VY/2,\\
S_{yy} &= (30/Y^2)\int f(x,y,z)K_{yy}(y)dV = m_{yy}VY^2/6,\\
S_z &= (6/Z)\int f(x,y,z)K_z(z)dV = m_z VZ/2,\\
S_{zz} &= (30/Z^2)\int f(x,y,z)K_{zz}(x)dV = m_{zz}VZ^2/6,\\
S_{xy} &= (36/XY)\int f(x,y,z)K_{xy}(x,y)dV = m_{xy}VXY/4,\\
S_{yz} &= (36/YZ)\int f(x,y,z)K_{yz}(y,z)dV = m_{yz}VYZ/4,\\
S_{zx} &= (36/ZX)\int f(x,y,z)K_{zx}(z,x)dV = m_{zx}VZX/4.
\end{aligned} \tag{10.130}$$

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 direction 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 $(z, Z)$ to derive a full set of formulas in the three dimensional space.

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

$$X - ct \le x \le X, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{10.131}$$

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

$$0 \le x \le X - ct, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{10.132}$$

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

$$K_0^T = K_0^R = 1,$$

$$K_x^T = x - (2X - ct)/2, \quad K_x^R = x - (X - ct)/2,$$

$$K_{xx}^T = x^2 - (2X - ct)x + (X - ct)X + (ct)^2/6,$$

$$K_{xx}^R = x^2 - (X - ct)x + (X - ct)^2/6,$$

$$\begin{aligned} K_y^T &= K_y^R = y - Y/2, \\ K_{yy}^T &= K_{yy}^R = y^2 - Yy + Y^2/6, \\ K_{xy}^T &= [x - (2X - ct)/2](y - Y/2). \\ K_{xy}^R &= [x - (X - ct)/2](y - Y/2), \end{aligned} \tag{10.133}$$

The basic quantities for calculating the moments are

$$\begin{aligned} m_0^T &= m_0 + \bar{K}_x^T m_x + \bar{K}_{xx}^T m_{xx}, \\ m_x^T &= m_x + 2\bar{K}_x^T m_{xx}, \\ m_{xx}^T &= m_{xx}, \\ m_y^T &= m_y + \bar{K}_x^T m_{xy}, \\ m_{yy}^T &= m_{yy}, \\ m_{xy}^T &= m_{xy}, \end{aligned} \tag{10.134}$$

and

$$\begin{aligned} m_0^R &= m_0 + \bar{K}_x^R m_x + \bar{K}_{xx}^R m_{xx}, \\ m_x^R &= m_x + 2\bar{K}_x^R m_{xx}, \\ m_{xx}^R &= m_{xx}, \\ m_y^R &= m_y + \bar{K}_x^R m_{xy}, \\ m_{yy}^R &= m_{yy}, \\ m_{xy}^R &= m_{xy}, \end{aligned} \tag{10.135}$$

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

$$\bar{K}_x^T = (X - ct)/2, \quad \bar{K}_x^R = -ct/2,$$

$$\bar{K}_{xx}^T = (X - ct)(X - 2ct)/6, \quad \bar{K}_{xx}^R = ct(2ct - X)/6.$$

The moments in the right part to be transported to the adjacent cell are expressed as follows:

$$\begin{aligned} S_0^T &= \alpha[S_0 + (1-\alpha)S_x + (1-\alpha)(1-2\alpha)S_{xx}], \\ S_x^T &= \alpha^2[S_x + 3(1-\alpha)S_{xx}], \\ S_{xx}^T &= \alpha^3 S_{xx}, \\ S_y^T &= \alpha[S_y + (1-\alpha)S_{xy}], \\ S_{yy}^T &= \alpha S_{yy}, \\ S_{xy}^T &= \alpha^2 S_{xy}, \end{aligned} \tag{10.136}$$

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

$$\begin{aligned} S_0^R &= (1-\alpha)[S_0 - \alpha S_x - \alpha(1-2\alpha)S_{xx}] = S_0 - S_0^T, \\ S_x^R &= (1-\alpha)^2(S_x - 3\alpha S_{xx}), \\ S_{xx}^R &= (1-\alpha)^3 S_{xx}, \\ S_y^R &= (1-\alpha)(S_y - \alpha S_{xy}) = S_y - S_y^T, \\ S_{yy}^R &= (1-\alpha)S_{yy} = S_{yy} - S_{yy}^T, \\ S_{xy}^R &= (1-\alpha)^2 S_{xy}. \end{aligned} \tag{10.137}$$

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. Here, we consider combinations of moments in the transported cell of the i-th grid point ($S^T_{n,i}$) and those in the remaining cell of the (i+1)-th grid point ($S^R_{n,i+1}$). The calculation is terribly complex, and only the results are presented:

$$S_0 = S^R_{0,i+1} + S^T_{0,i},$$

$$S_x = \alpha S^T_{x,i} + (1-\alpha)S^R_{x,i+1} + 3[\alpha S^R_{0,i+1} - (1-\alpha)S^T_{0,i}],$$

$$S_{xx} = \alpha^2 S^T_{xx,i} + (1-\alpha)^2 S^R_{xx,i+1} + 5\{\alpha(1-\alpha)(S^R_{x,i+1} - S^T_{x,i}) - (1-2\alpha)[\alpha S^R_{0,i+1} - (1-\alpha)S^T_{0,i}]\},$$

$$S_y = S^R_{y,i+1} + S^T_{y,i},$$

$$S_{yy} = S^R_{yy,i+1} + S^T_{yy,i},$$

$$S_{xy} = \alpha S^T_{xy,i} + (1-\alpha)S^R_{xy,i+1} + 3[-(1-\alpha)S^T_{y,i} + \alpha S^R_{y,i+1}],$$

where

$$\alpha = \alpha^T_i = V^T_i/(V^T_i + V^R_{i+1}). \tag{10.138}$$

When velocity $c$ in the $x$ direction is negative, the left part of the grid cell,

$$0 \le x \le -ct, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{10.139}$$

is removed (transported) from the cell and added to the adjacent cell on the left during time interval $t$. This part is now expressed using superscript $T$. The remaining part,

$$-ct \le x \le X, \quad 0 \le y \le Y, \quad 0 \le z \le Z, \tag{10.140}$$

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

$$K^T_0 = K^R_0 = 1,$$

$$K^T_x = x + ct/2, \quad K^R_x = x - (X-ct)/2,$$

$$K^T_{xx} = x^2 + ctx + (ct)^2/6,$$

$$K^R_{xx} = x^2 - (X-ct)x + (X^2 - 4Xct + c^2t^2)/6,$$

$$K^T_y = K^R_y = y - Y/2, \tag{10.141}$$

$$K^T_{yy} = K^R_{yy} = y^2 - Yy + Y^2/6,$$

$$K^T_{xy} = (x + ct/2)(y - Y/2),$$

$$K^R_{xy} = [x - (X-ct)/2](y - Y/2).$$

The moments in the left part to be transported to the adjacent cell are expressed as follows:

$$\begin{aligned}
S^T_0 &= \alpha[S_0 - (1-\alpha)S_x + (1-\alpha)(1-2\alpha)S_{xx}],\\
S^T_x &= \alpha^2[S_x - 3(1-\alpha)S_{xx}],\\
S^T_{xx} &= \alpha^3 S_{xx},\\
S^T_y &= \alpha[S_y - (1-\alpha)S_{xy}],\\
S^T_{yy} &= \alpha S_{yy},\\
S^T_{xy} &= \alpha^2 S_{xy},
\end{aligned} \tag{10.142}$$

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

$$\begin{aligned}
S_0^R &= (1-\alpha)[S_0 + \alpha S_x - \alpha(1-2\alpha)S_{xx}] = S_0 - S_0^T, \\
S_x^R &= (1-\alpha)^2(S_x + 3\alpha S_{xx}), \\
S_{xx}^R &= (1-\alpha)^3 S_{xx}, \\
S_y^R &= (1-\alpha)(S_y + \alpha S_{xy}) = S_y - S_y^T, \\
S_{yy}^R &= (1-\alpha)S_{yy} = S_{yy} - S_{yy}^T, \\
S_{xy}^R &= (1-\alpha)^2 S_{xy}.
\end{aligned} \tag{10.143}$$

The following equations give combinations of moments in the remaining cell of the i-th grid point ($S^R_{n,i}$) and those in the transported cell of the (i+1)-th grid point ($S^T_{n,i+1}$):

$$S_0 = S^R_{0,i} + S^T_{0,i+1},$$

$$S_x = \alpha S^T_{x,i+1} + (1-\alpha)S^R_{x,i} - 3[\alpha S^R_{0,i} - (1-\alpha)S^T_{0,i+1}],$$

$$S_{xx} = \alpha^2 S^T_{xx,i+1} + (1-\alpha)^2 S^R_{xx,i} + 5\{-\alpha(1-\alpha)(S^R_{x,i} - S^T_{x,i+1}) - (1-2\alpha)[\alpha S^R_{0,i} - (1-\alpha)S^T_{0,i+1}]\},$$

$$S_y = S^R_{y,i} + S^T_{y,i+1},$$

$$S_{yy} = S^R_{yy,i} + S^T_{yy,i+1},$$

$$S_{xy} = \alpha S^T_{xy,i+1} + (1-\alpha)S^R_{xy,i} - 3[-(1-\alpha)S^T_{y,i+1} + \alpha S^R_{y,i}],$$

where

$$\alpha = \alpha^T_{i+1} = V^T_{i+1}/(V^R_i + V^T_{i+1}). \tag{10.144}$$

The globally integrated tracer is conserved through these operations.

### 10.5.2 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,

$$\begin{aligned}
S_0 &\geq 0, \\
S'_x &= \min[+1.5S_0, \max(-1.5S_0, S_x)], \\
S'_{xx} &= \min[2S_0 - |S'_x|/3, \max(|S'_x| - S_0, S_{xx})], \\
S'_{xy} &= \min[+S_0, \max(-S_0, S_{xy})].
\end{aligned} \tag{10.145}$$

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).

$$\begin{aligned}
\bar{\tau}(x) &= \frac{1}{YZ}\int_0^Y \int_0^Z dy\, dz\, \tau(x, y, z) \\
&= \frac{1}{V}\left[S_0 - S_x + S_{xx} + 2(S_x - 3S_{xx})\frac{x}{X} + 6S_{xx}\left(\frac{x}{X}\right)^2\right],
\end{aligned} \tag{10.146}$$

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:

$$V \min \bar{\tau}(x) = \begin{cases} S_0 - |S_x| + S_{xx}, & \text{if } S_{xx} \le 0 \quad \text{(region I)}, \\ S_0 - |S_x| + S_{xx}, & \text{if } |S_x| \ge 3S_{xx} \ge 0 \quad \text{(region II)}, \\ S_0 - \frac{S_x^2 + 3S_{xx}^2}{6S_{xx}}, & \text{if } 3S_{xx} \ge |S_x| \quad \text{(region III)}. \end{cases} \tag{10.147}$$

The line of $V \min \bar{\tau}(x) = 0$ is plotted in the $(|S_x|, S_{xx})$ space in Figure 10.5. 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.

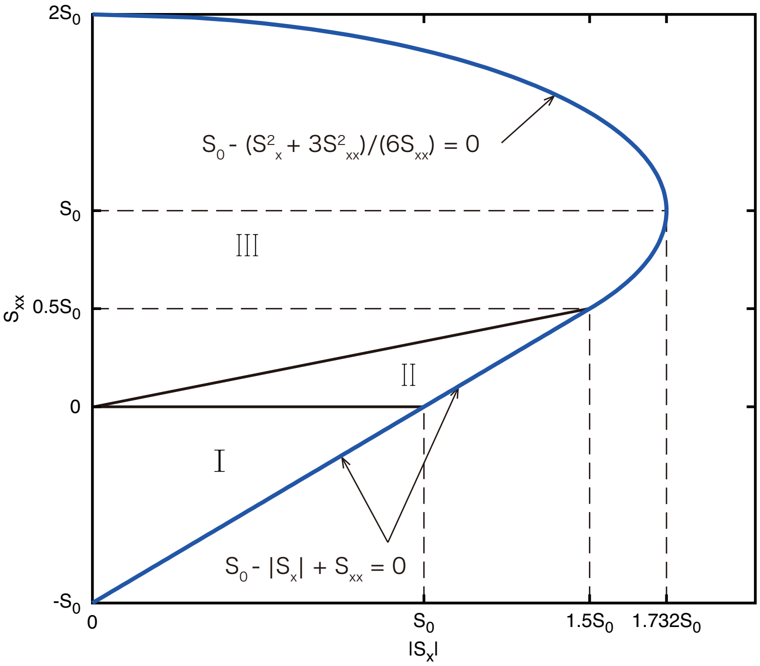

Figure10.5 Schematic to explain the flux limiter to avoid undershooting. This is a reproduction of Figure 3 of Morales Maqueda and Holloway (2006). See text for explanation.

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,

$$\begin{aligned} S'_x &= \min\left[3^{1/2}S_0,\ \max\left[-3^{1/2}S_0,\ S_x\right]\right], \\ S'_{xx} &= \begin{cases} \min\left[S_0 + \left(S_0^2 - \frac{S'^2_x}{3}\right)^{1/2},\ \max\left[\left[|S'_x| - S_0,\ S_{xx}\right]\right]\right], & \text{if } |S'_x| < \frac{3}{2}S_0, \\ \min\left[S_0 + \left(S_0^2 - \frac{S'^2_x}{3}\right)^{1/2},\ \max\left[S_0 - \left(S_0^2 - \frac{S'^2_x}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } |S'_x| \ge \frac{3}{2}S_0, \end{cases} \\ S'_{xy} &= \min\left[S_0,\ \max[-S_0,\ S_{xy}]\right], \\ S'_{xz} &= \min\left[S_0,\ \max[-S_0,\ S_{xz}]\right]. \end{aligned} \tag{10.148}$$

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 10.145) was used in PR86.

This modification is explained using Figure 10.5. 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 $\tau_a$ and $\tau_b$, 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 $\tau_b$" 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 (10.146), $\tau_b - \bar{\tau}(x)$ may be expressed as follows:

$$\begin{aligned}\tau_b - \bar{\tau}(x) &= \frac{1}{V}\left[V\tau_b - S_0 + S_x - S_{xx} - 2(S_x - 3S_{xx})\frac{x}{X} - 6S_{xx}\left(\frac{x}{X}\right)^2\right] \\ &= \frac{1}{V}\left[S_0^* + S_x - S_{xx} - 2(S_x - 3S_{xx})\frac{x}{X} - 6S_{xx}\left(\frac{x}{X}\right)^2\right], \end{aligned} \tag{10.149}$$

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

$$V\ \min\left[\tau_b - \bar{\tau}(x)\right] = \begin{cases} S_0^* - |S_x| - S_{xx}, & \text{if } S_{xx} \ge 0 \quad \text{(region I)}, \\ S_0^* - |S_x| - S_{xx}, & \text{if } |S_x| \ge -3S_{xx} \ge 0 \quad \text{(region II)}, \\ S_0^* + \frac{S_x^2 + 3S_{xx}^2}{6S_{xx}}, & \text{if } -3S_{xx} \ge |S_x| \quad \text{(region III)}. \end{cases} \tag{10.150}$$

Following Figure 10.5, this is visualized as Figure 10.6.

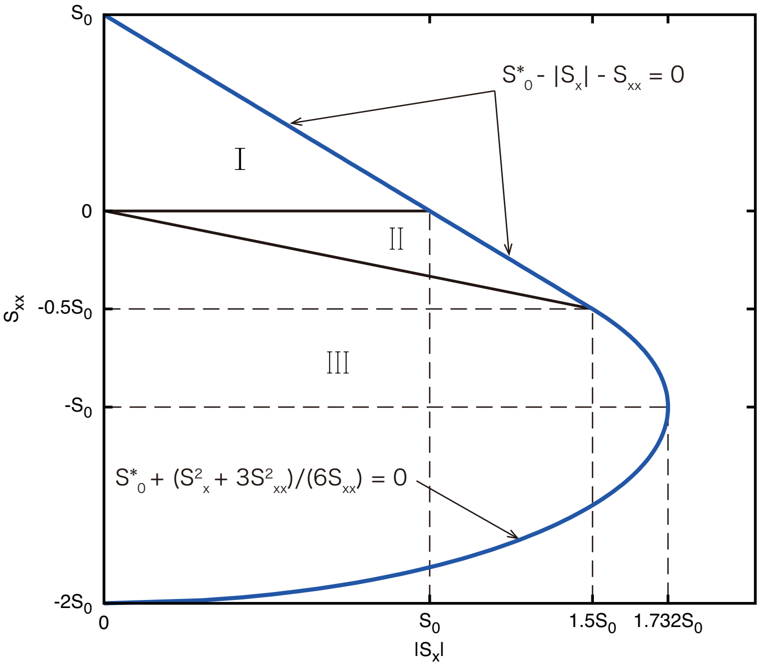

Figure10.6 Schematic to explain the flux limiter to avoid overshooting. The region in $(|S_x|, S_{xx})$ space that may set the upper bound on tracer distribution. The region bounded by the blue curve and the left vertical axis gives the tracer value that does not exceed the upper bound.

The modification for the moments corresponding to (10.148) is expressed as follows:

$$\begin{aligned} S'_x &= \min\left[3^{1/2}S_0^*,\ \max\left[-3^{1/2}S_0^*,\ S_x\right]\right], \\ S'_{xx} &= \begin{cases} \min\left[S_0^* - |S'_x|,\ \max\left[-S_0^* - \left(S_0^{*2} - \frac{S'^2_x}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } |S'_x| < \frac{3}{2}S_0^*, \\ \min\left[-S_0^* + \left(S_0^{*2} - \frac{S'^2_x}{3}\right)^{1/2},\ \max\left[-S_0^* - \left(S_0^{*2} - \frac{S'^2_x}{3}\right)^{1/2},\ S_{xx}\right]\right], & \text{if } |S'_x| \ge \frac{3}{2}S_0^*, \end{cases} \\ S'_{xy} &= \min\left[S_0^*,\ \max(-S_0^*,\ S_{xy})\right], \\ S'_{xz} &= \min\left[S_0^*,\ \max(-S_0^*,\ S_{xz})\right]. \end{aligned} \tag{10.151}$$

MH06 proposed three modification methods (Method A - C). 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. (10.148) and (10.151). This method does not guarantee exact monotonicity, but improves monotonicity of the zeroth moment.

### 10.5.3 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 ($\alpha$) are used in the SOM advection scheme. There are 10 moments for each tracer. The fraction $\alpha$ is calculated using volume transports, 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 10.2) 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.

Table10.2 Variables defined in namelist `nml_somadv`

| name | explanation | string or value |
|---|---|---|
| `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.` |

## 10.6 Piecewise Parabolic Method (PPM) scheme

### 10.6.1 Outline

The Piecewise Parabolic Method (PPM; Colella and Woodward, 1984) scheme first expresses a local tracer distribution within each model grid cell as a quadratic equation. These tracer distributions are then advected in a Lagrangian manner and redistributed to the fixed model grid cells (Figure 10.7).

Here we consider a one-dimensional advection equation in the $\xi$ direction and assume that we know the mean value of an arbitrary tracer in the $j$ grid at time $t^n$ ($a_j^n$). Then we consider a polynomial that represents the local distribution of the tracer in each grid cell and describes a set of these polynomials with $a(\xi)$. These polynomials satisfy the following equation:

$$a_j^n = \frac{1}{\Delta\xi_j}\int_{\xi_{j-\frac{1}{2}}}^{\xi_{j+\frac{1}{2}}} a(\xi)d\xi, \quad \Delta\xi_j = \xi_{j+\frac{1}{2}} - \xi_{j-\frac{1}{2}}, \tag{10.152}$$

where $\xi_{j-\frac{1}{2}}$ and $\xi_{j+\frac{1}{2}}$ denote cell boundaries between the $j-1$ and $j$ grid cells and between the $j$ and $j+1$ grid cells, respectively.

A solution of the advection equation at time $t^n + \Delta t$ is described as $a(\xi - u\Delta t)$ (Fig. 10.7b) with its initial distribution $a(\xi)$ (Fig. 10.7a). We can gain $a_j^{n+1}$ by integrating this solution as follows:

$$a_j^{n+1} = \frac{1}{\Delta\xi}\int_{\xi_{j-\frac{1}{2}}}^{\xi_{j+\frac{1}{2}}} a(\xi - u\Delta t)d\xi. \tag{10.153}$$

The specific form of the solution depends on a choice of polynomials. The piecewise parabolic method uses a locally continuous quadratic polynomial in each grid cell as follows:

$$\begin{aligned} a(\xi) &= a_{L,j} + x(\Delta a_j + a_{6,j}(1-x)), \\ x &= \frac{\xi - \xi_{j-\frac{1}{2}}}{\Delta\xi_j}, \quad \xi_{j-\frac{1}{2}} \le \xi \le \xi_{j+\frac{1}{2}}. \end{aligned} \tag{10.154}$$

The coefficients of this polynomial are defined with $a_j^n$, $\lim_{\xi\to\xi_{j-\frac{1}{2}}} a(\xi) = a_{L,j}$ and $\lim_{\xi\to\xi_{j+\frac{1}{2}}} a(\xi) = a_{R,j}$ as follows:

$$\Delta a_j = a_{R,j} - a_{L,j}, \quad a_{6,j} = 6(a_j^n - \frac{1}{2}(a_{L,j} + a_{R,j})). \tag{10.155}$$

The edge values such as $a_{L,j}$ and $a_{R,j}$ are calculated according to the following procedures. First, we interpolate a tracer value at a cell interface using the surrounding cell averages. We use the indefinite integral of $a$ as $A(\xi) = \int^{\xi} a(\xi', t^n) d\xi'$ for this interpolation. This integral at a cell boundary is calculated as follows:

$$A(\xi_{j+\frac{1}{2}}) = A_{j+\frac{1}{2}} = \sum_{k\le j} a_k^n \Delta\xi_k. \tag{10.156}$$

Then we find a quartic polynomial through $(A_{j+k+\frac{1}{2}}, \xi_{j+k+\frac{1}{2}})$, $(k = 0, \pm 1, \pm 2)$ and calculate a tracer value at the interface $a(\xi_{j+\frac{1}{2}}) = a_{j+\frac{1}{2}}$ by $a_{j+\frac{1}{2}} = dA/d\xi|_{\xi_{j+\frac{1}{2}}}$. The interpolated value $a_{j+\frac{1}{2}}$ is described as follows:

$$\begin{aligned} a_{j+\frac{1}{2}} =& a_j^n + \frac{\Delta\xi_j}{\Delta\xi_j + \Delta\xi_{j+1}}(a_{j+1}^n - a_j^n) + \frac{1}{\sum_{k=-1}^{2}\Delta\xi_{j+k}} \\ &\times \left\{ \frac{2\Delta\xi_{j+1}\Delta\xi_j}{\Delta\xi_j + \Delta\xi_{j+1}} \left[ \frac{\Delta\xi_{j-1} + \Delta\xi_j}{2\Delta\xi_j + \Delta\xi_{j+1}} - \frac{\Delta\xi_{j+2} + \Delta\xi_{j+1}}{2\Delta\xi_{j+1} + \Delta\xi_j} \right] (a_{j+1}^n - a_j^n) \right. \\ & \left. - \Delta\xi_j \frac{\Delta\xi_{j-1} + \Delta\xi_j}{2\Delta\xi_j + \Delta\xi_{j+1}} \delta a_{j+1} + \Delta\xi_{j+1} \frac{\Delta\xi_{j+1} + \Delta\xi_{j+2}}{\Delta\xi_j + 2\Delta\xi_{j+1}} \delta a_j \right\}, \end{aligned} \tag{10.157}$$

where the mean slope of tracer at the $j$ grid point $\delta a_j$ is

$$\delta a_j = \frac{\Delta\xi_j}{\Delta\xi_{j-1} + \Delta\xi_j + \Delta\xi_{j+1}} \left[ \frac{2\Delta\xi_{j-1} + \Delta\xi_j}{\Delta\xi_{j+1} + \Delta\xi_j}(a_{j+1}^n - a_j^n) + \frac{\Delta\xi_j + 2\Delta\xi_{j+1}}{\Delta\xi_{j-1} + \Delta\xi_j}(a_j^n - a_{j-1}^n) \right]. \tag{10.158}$$

This expression is also suitable for MRI.COM because its tracer point is defined at the center of each unit cell. The interpolation by Eq. (10.157) requires two effective tracer cells on both sides of the cell interface, respectively. For example, we need $[a_{j-1}, a_{j+2}]$ for $a_{j+\frac{1}{2}}$. If a land cell exists in this range, $a_{j+\frac{1}{2}}$ is calculated by distance weighted average of the adjacent cells.

This cell boundary value $a_{j+\frac{1}{2}}$ is usually modified by flux limiters described later to improve monotonicity, before it is used as the edge values $a_{L,j+1}$ and $a_{R,j}$. Because each edge value $a_{L,j+1}$ or $a_{R,j}$ is modified independently to each other, $a(\xi)$ is generally discontinuous at each cell boundary. If a lateral boundary of an ocean cell is completely in land (possible in MRI.COM; see Chapter 3) or a vertical cell interface is at the land-sea boundary, the edge value there is set to the cell average of the ocean cell.

Now we can calculate tracer fluxes at the cell boundaries and the cell average at the next time step with these edge values $a_{L,j}$ and $a_{R,j}$. Averages of advected tracer crossing lateral cell boundaries are calculated by the following equations:

$$f^a_{j+\frac{1}{2},L}(y) = \frac{1}{y}\int_{\xi_{j+\frac{1}{2}}-y}^{\xi_{j+\frac{1}{2}}} a(\xi)d\xi \tag{10.159}$$

$$f^a_{j+\frac{1}{2},R}(y) = \frac{1}{y}\int_{\xi_{j+\frac{1}{2}}}^{\xi_{j+\frac{1}{2}}+y} a(\xi)d\xi. \tag{10.160}$$

For the PPM scheme, these are described as follows:

$$\begin{aligned} f^a_{j+\frac{1}{2},L}(y) &= a_{R,j} - \frac{x}{2}\Big(\Delta a_j - \Big(1 - \frac{2}{3}x\Big)a_{6,j}\Big), \quad \text{for } x = \frac{y}{\Delta\xi_j} \\ f^a_{j+\frac{1}{2},R}(y) &= a_{L,j+1} + \frac{x}{2}\Big(\Delta a_{j+1} + \Big(1 - \frac{2}{3}x\Big)a_{6,j+1}\Big), \quad \text{for } x = \frac{y}{\Delta\xi_{j+1}} \end{aligned} \tag{10.161}$$

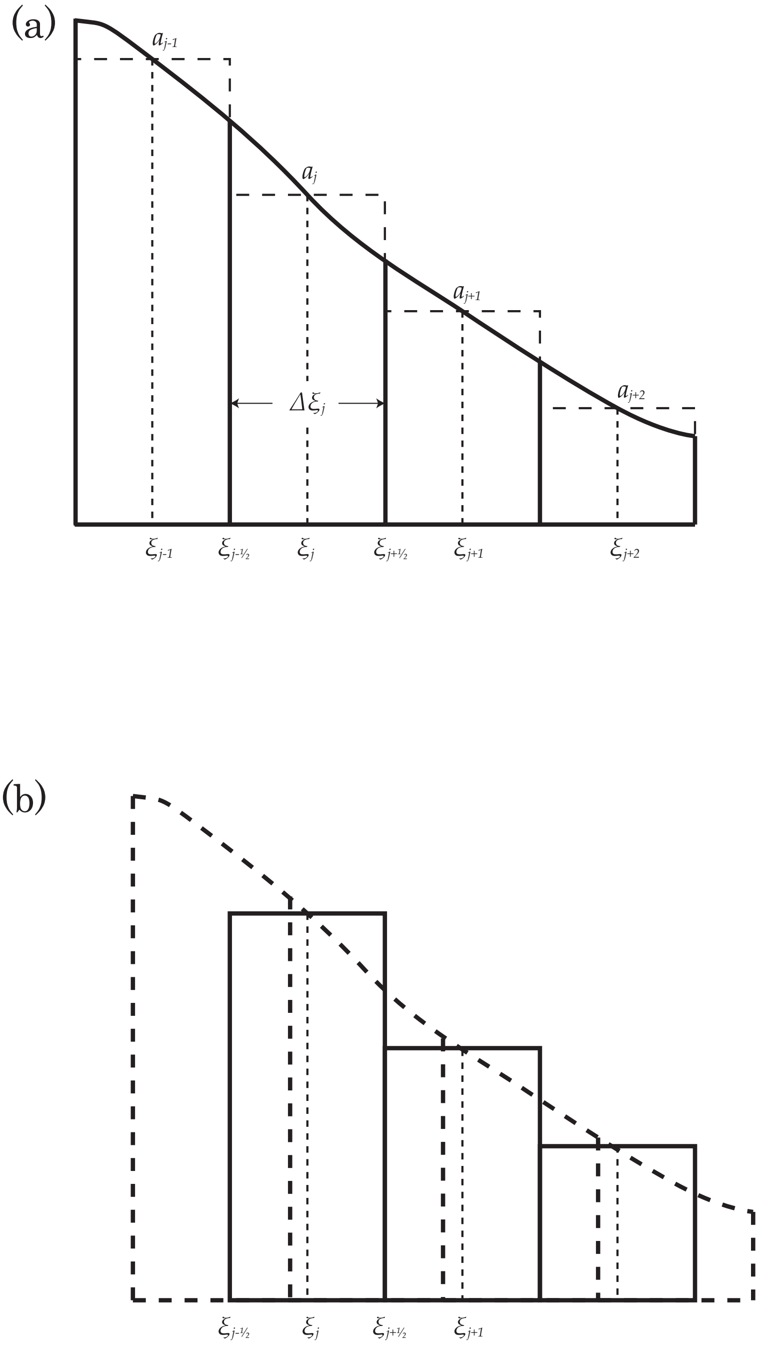

Figure10.7 Schematic of PPM scheme. (a) An initial state. (b) The model is advanced in time and the initial profile in (a) is shifted to the right.

If a lateral boundary is completely in land or a vertical cell interface is at the land-sea boundary, $\Delta a_j$ and $a_{6,j}$ in Eq. (10.161) in the corresponding direction are forced to zero. So, the equations above are simplified to

$$\begin{aligned} f^a_{j+\frac{1}{2},L}(y) &= a_{R,j}, \\ f^a_{j-\frac{1}{2},R}(y) &= a_{L,j}. \end{aligned} \tag{10.162}$$

The new cell average $a^{n+1}_j$ is described in the flux form that guarantees the tracer conservation as follows:

$$a^{n+1}_j = a^n_j + u\frac{\Delta t}{\Delta\xi_j}(\bar{a}_{j-\frac{1}{2}} - \bar{a}_{j+\frac{1}{2}}), \tag{10.163}$$

where

$$\bar{a}_{j+\frac{1}{2}} = \begin{cases} f^a_{j+\frac{1}{2},L}(u\Delta t), & \text{if} \quad u \geq 0 \\ f^a_{j+\frac{1}{2},R}(-u\Delta t), & \text{if} \quad u \leq 0. \end{cases} \tag{10.164}$$

### 10.6.2 Flux limiter

The following two flux limiters are implemented to enhance monotonicity.

#### a. Colella and Woodward (1984)

The flux limiter proposed by Colella and Woodward (1984) modifies the tracer gradient in Eq. (10.158) as follows:

$$\begin{aligned} \delta_m a_j =& \min(|\delta a_j|, 2|a^n_{j+1} - a^n_j|, 2|a^n_j - a^n_{j-1}|)\mathrm{sgn}(\delta a_j) \\ & \text{if} \quad (a^n_{j+1} - a^n_j)(a^n_j - a^n_{j-1}) > 0 \\ =& 0 \quad \text{otherwise} \end{aligned} \tag{10.165}$$

This allows us to express steeper gradients when we treat discontinuous tracer distributions.

The edge values $a_{L,j+1}$ and $a_{R,j}$ are normally assigned $a_{j+\frac{1}{2}}$. However, an interpolation function occasionally exceeds the range of $a_{L,j}$ and $a_{R,j}$ within the $j$-th grid cell. In that case, we make the following adjustments.

- Make the interpolation function a constant if $a^n_j$ exceeds the range of $a_{L,j}$ and $a_{R,j}$.
- Adjust edge values to satisfy a condition to prevent overshoot of the interpolation function, which may occur when $a^n_j$ is in the range of $a_{L,j}$ and $a_{R,j}$ but close enough to one side. The condition is $|\Delta a_j| \geq |a_{6,j}|$.

Specifically, edge values are modified as follows:

$$a_{L,j} \longrightarrow a^n_j, \quad a_{R,j} \longrightarrow a^n_j \quad \text{if} \quad (a_{R,j} - a^n_j)(a^n_j - a_{L,j}) < 0 \tag{10.166}$$

$$a_{L,j} \longrightarrow 3a^n_j - 2a_{R,j} \quad \text{if} \quad (a_{R,j} - a_{L,j})\left(a^n_j - \frac{1}{2}(a_{L,j} + a_{R,j})\right) > \frac{(a_{R,j} - a_{L,j})^2}{6} \tag{10.167}$$

$$a_{R,j} \longrightarrow 3a^n_j - 2a_{L,j} \quad \text{if} \quad -\frac{(a_{R,j} - a_{L,j})^2}{6} > (a_{R,j} - a_{L,j})\left(a^n_j - \frac{1}{2}(a_{L,j} + a_{R,j})\right) \tag{10.168}$$

Equation (10.167) is applied when $a_{R,j} \sim a^n_j$, and Eq. (10.168) is applied when $a_{L,j} \sim a^n_j$.

#### b. Lin et al. (1994)

The flux limiter proposed by Lin et al. (1994) adjust the tracer slope in Eq. (10.158) as follows:

$$\delta_m a_j = \min(|\delta a_j|, 2\delta a^{\min}_j, 2\delta a^{\max}_j)\ \mathrm{sgn}(\delta a_j), \tag{10.169}$$

where

$$\delta a^{\max}_j = \max(a_{j-1}, a_j, a_{j+1}) - a_j, \tag{10.170}$$

$$\delta a^{\min}_j = a_j - \min(a_{j-1}, a_j, a_{j+1}). \tag{10.171}$$

Edge values are modified as follows:

$$a_{L,j} \longleftarrow a_j - \min(|\delta_m a_j|, |a_{L,j} - a_j|)\ \mathrm{sgn}(\delta_m a_j), \tag{10.172}$$

$$a_{R,j} \longleftarrow a_j + \min(|\delta_m a_j|, |a_{R,j} - a_j|)\ \mathrm{sgn}(\delta_m a_j). \tag{10.173}$$

### 10.6.3 Extension to multiple dimensions with time-splitting algorithm

An advection scheme is commonly developed for one dimensional problems. There are two ways to extend it to multiple dimensions: one is based on a completely three-dimensional advection algorithm and the other is to apply the one-dimensional scheme to each dimensions in turn. The latter is called the time-splitting method. MRI.COM adopts the time-splitting method implemented in GFDL-MOM.

The following finite difference equations describe an one-dimensional advection equation for an arbitrary tracer $\phi$ and the equation of continuity in a generalized vertical coordinate $r$, which adopt the Euler forward scheme in time:

$$(z_r\phi)^{t+\Delta t} = (z_r\phi)^t - F_x(\phi^t) \tag{10.174}$$

$$(z_r)^{t+\Delta t} = (z_r)^t - F_x(I), \tag{10.175}$$

where $z_r$ is cell thickness, the right superscript denotes time and the vector $I \equiv 1$. The second term in the right hand side $F_x$ denotes flux divergence and is described as follows:

$$F_x(\phi) = (\Delta t/\Delta x)[f(\phi)_{x+\Delta x/2} - f(\phi)_{x-\Delta x/2}]. \tag{10.176}$$

The flux $f$ is calculated by a product of a volume transport $(z_r u)_{x\pm\Delta x/2}$ and a tracer value $\phi_{x\pm\Delta x/2}$ at a cell boundary. The tracer values at cell boundaries are calculated by interpolation or integration. Both the tracer advection equation and the equation of continuity must use the same volume transports at cell boundaries.

The volume conservation equation (the equation of continuity) is extended to three dimensions as follows:

$$(z_r)^{t+\Delta t} = (z_r)^t - F_x(I) - F_y(I) - F_z(I). \tag{10.177}$$

The finite difference equation of the one-dimensional tracer advection equation (10.174) is extended to multiple dimensions according to the following time-splitting method implemented in GFDL-MOM.

$$(z_r)^{t+\Delta t} = (z_r)^t - F_x(I) - F_y(I) - F_z(I) \tag{10.178}$$

$$(z_r\phi)^* = (z_r\phi)^t - F_x(\phi^t) + F_x(I)\phi^t, \tag{10.179}$$

$$\phi^* = (z_r\phi)^*/(z_r)^t, \tag{10.180}$$

$$(z_r\phi)^{**} = (z_r\phi)^* - F_y(\phi^*) + F_y(I)\phi^t, \tag{10.181}$$

$$\phi^{**} = (z_r\phi)^{**}/(z_r)^t, \tag{10.182}$$

$$(z_r\phi)^{t+\Delta t} = (z_r\phi)^{**} - F_z(\phi^{**}) - \{F_x(I) + F_x(I)\}\phi^t, \tag{10.183}$$

$$\phi^{t+\Delta t} = (z_r\phi)^{t+\Delta t}/(z_r)^{t+\Delta t}. \tag{10.184}$$

A feature of this method is that $F_x(I)\phi^t$ and $F_y(I)\phi^t$ are added in the right hand side of Eqs. (10.179) and (10.181). Each additional term puts back (removes) a change in the tracer content due to volume flux divergence (convergence) in the corresponding direction with a tracer value $\phi^t$ at the initial time ($t = t$). This process improves numerical stability when the volume flux divergence/convergence in each direction is large. The total tracer content in the cell is conserved by adding $-F_x(I)\phi^t - F_y(I)\phi^t$ in the right hand side of Eq. (10.183). We gain the following equation by taking the sum of Eqs. (10.179), (10.181) and (10.183):

$$(z_r\phi)^{t+\Delta t} = (z_r\phi)^t - F_x(\phi^t) - F_y(\phi^*) - F_z(\phi^{**}). \tag{10.185}$$

This clearly shows the tracer content conservation.

### 10.6.4 Usage

Users must compile the model with the option PPMADVEC and specify adv_scheme%name = ”ppm” in namelist nml_tracer_data of tracers to be advected with the PPM advection scheme. Two flux limiters are available as already noted; the monotonic scheme proposed by Lin et al. (1994) and the original limiter of Colella and Woodward (1984) (Table 10.3). Users may also specify whether to use the Strang splitting method which changes the order of horizontal advections every time step and whether to output a special monitor for PPM scheme (Table 10.4).

## 10.7 MPDATA scheme

### 10.7.1 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.

Table10.3 Namelist `nml_tracer_data`.

| variable name | units | description | usage |
|---|---|---|---|
| adv_scheme%name | character | select advection scheme for that tracer | = "ppm" |
| adv_scheme%limiter_ppm_org | logical | Use correction introduced by Colella and Woodward (1984) for monotonicity | default = .false. |
| adv_scheme%limiter_ppm_lin | logical | Use correction introduced by Lin et al. (1994) for monotonicity | default = .true. |

Table10.4 Namelist `nml_ppmadv`.

| variable name | units | description | usage |
|---|---|---|---|
| lppmmonitor | logical | monitor domain integrated volume and tracer | default = .false. |
| lppmstrang | logical | use Strang splitting | default = .false. |

Original dimensionless form of the anti-diffusive velocity given by Smolarkiewicz and Margolin (1998) is written as follows (their equation 16):

$$\mathcal{U}^{(1)} \equiv \frac{u^{(1)}\delta t}{\delta x} = |\mathcal{U}|(1-\mathcal{U})A^{(1)} - \mathcal{U}\mathcal{V}B^{(1)} - \mathcal{U}\mathcal{W}C^{(1)}, \tag{10.186}$$

$$\mathcal{V}^{(1)} \equiv \frac{v^{(1)}\delta t}{\delta y} = |\mathcal{V}|(1-\mathcal{V})B^{(1)} - \mathcal{V}\mathcal{W}C^{(1)} - \mathcal{V}\mathcal{U}A^{(1)}, \tag{10.187}$$

$$\mathcal{W}^{(1)} \equiv \frac{w^{(1)}\delta t}{\delta z} = |\mathcal{W}|(1-\mathcal{W})C^{(1)} - \mathcal{W}\mathcal{U}A^{(1)} - \mathcal{W}\mathcal{V}B^{(1)}, \tag{10.188}$$

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

$$A^{(1)} \equiv \left[\frac{\delta x}{2T}\frac{\partial T}{\partial x}\right]^{(1)}, \tag{10.189}$$

$$B^{(1)} \equiv \left[\frac{\delta y}{2T}\frac{\partial T}{\partial y}\right]^{(1)}, \tag{10.190}$$

$$C^{(1)} \equiv \left[\frac{\delta z}{2T}\frac{\partial T}{\partial z}\right]^{(1)}. \tag{10.191}$$

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 (10.8) to (10.10), the dimensionless form of volume transports can be written as

$$\mathcal{U}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{2U^T_{i+\frac{1}{2},j,k-\frac{1}{2}}\Delta t}{\Delta x_{i+\frac{1}{2}j}\Delta y_{i+\frac{1}{2},j}\left(\Delta z_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right)}, \tag{10.192}$$

$$\mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}} = \frac{2V^T_{i,j+\frac{1}{2},k-\frac{1}{2}}\Delta t}{\Delta x_{i,j+\frac{1}{2}}\Delta y_{i,j+\frac{1}{2}}\left(\Delta z_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \Delta z_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\right)}, \tag{10.193}$$

$$\mathcal{W}_{i,j,k} = \frac{W^T_{i,j,k}\Delta t}{\Delta z_{i,j,k}(\mathtt{areat})_{i,j,k+\frac{1}{2}}}. \tag{10.194}$$

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

$$A^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta x_{i+\frac{1}{2},j} \delta_x T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}{2\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x} + \epsilon} \tag{10.195}$$

$$B^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta y_{i+\frac{1}{2},j} \delta_y \overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{y}}{2\overline{\overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{y}}^{x}}^{y} + \epsilon} \tag{10.196}$$

$$C^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \frac{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}} \delta_z \overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{z}}{2\overline{\overline{\overline{T^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{z}}^{x}}^{z} + \epsilon} \tag{10.197}$$

where $\epsilon$ 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):

$$\begin{aligned} \delta_x A_i &\equiv \frac{A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}}{\Delta x_i}, \quad \delta_x A_{i+\frac{1}{2}} \equiv \frac{A_{i+1} - A_i}{\Delta x_{i+\frac{1}{2}}}, \\ \overline{A_i}^{x} &\equiv \frac{A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}}}{2}, \quad \overline{A_{i+\frac{1}{2}}}^{x} \equiv \frac{A_{i+1} + A_i}{2}. \end{aligned} \tag{10.198}$$

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

$$U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} = \left|U^{T}_{i+\frac{1}{2},j,k-\frac{1}{2}}\right|(1-\mathcal{U}_{i+\frac{1}{2},j,k-\frac{1}{2}})A^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} - U^{T}_{i+\frac{1}{2},j,k-\frac{1}{2}}\overline{\overline{\mathcal{V}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{y} B^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} - U^{T}_{i+\frac{1}{2},j,k-\frac{1}{2}}\overline{\overline{\mathcal{W}_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{x}}^{z} C^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{10.199}$$

$$V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} = \left|V^{T}_{i,j+\frac{1}{2},k-\frac{1}{2}}\right|(1-\mathcal{V}_{i,j+\frac{1}{2},k-\frac{1}{2}})B^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} - V^{T}_{i,j+\frac{1}{2},k-\frac{1}{2}}\overline{\overline{\mathcal{U}_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{x}}^{y} A^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} - V^{T}_{i,j+\frac{1}{2},k-\frac{1}{2}}\overline{\overline{\mathcal{W}_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{y}}^{z} C^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{10.200}$$

$$W^{(1)}_{i,j,k} = \left|W^{T}_{i,j,k}\right|(1-\mathcal{W}_{i,j,k})C^{(1)}_{i,j,k} - W^{T}_{i,j,k}\overline{\overline{\mathcal{U}_{i+\frac{1}{2},j,k}}^{x}}^{z} A^{(1)}_{i,j,k} - W^{T}_{i,j,k}\overline{\overline{\mathcal{V}_{i+\frac{1}{2},j,k}}^{y}}^{z} B^{(1)}_{i,j,k}. \tag{10.201}$$

### 10.7.2 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:

$$\left[U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i+1,j,k-\frac{1}{2}})U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} > 0, \\ \min(1, \beta^{out}_{i+1,j,k-\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad U^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} < 0, \end{cases} \tag{10.202}$$

$$\left[V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i,j+1,k-\frac{1}{2}})V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} & \text{if} \quad V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} > 0, \\ \min(1, \beta^{out}_{i,j+1,k-\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})V^{(1)}_{i+\frac{1}{2},j,k-\frac{1}{2}} & \text{if} \quad V^{(1)}_{i,j+\frac{1}{2},k-\frac{1}{2}} < 0, \end{cases} \tag{10.203}$$

$$\left[W^{(1)}_{i,j,k}\right]^{mon} = \begin{cases} \min(1, \beta^{out}_{i,j,k+\frac{1}{2}}, \beta^{in}_{i,j,k-\frac{1}{2}})W^{(1)}_{i,j,k} & \text{if} \quad W^{(1)}_{i,j,k} > 0, \\ \min(1, \beta^{out}_{i,j,k-\frac{1}{2}}, \beta^{in}_{i,j,k+\frac{1}{2}})W^{(1)}_{i,j,k} & \text{if} \quad W^{(1)}_{i,j,k} < 0, \end{cases} \tag{10.204}$$

where

$$\beta^{in}_{i,j,k-\frac{1}{2}} \equiv \frac{T^{max}_{i,j,k-\frac{1}{2}} - T^{(1)}_{i,j,k-\frac{1}{2}}}{A^{in}_{i,j,k-\frac{1}{2}} + \epsilon}, \tag{10.205}$$

$$\beta^{out}_{i,j,k-\frac{1}{2}} \equiv \frac{T^{(1)}_{i,j,k-\frac{1}{2}} - T^{min}_{i,j,k-\frac{1}{2}}}{A^{out}_{i,j,k-\frac{1}{2}} + \epsilon}, \tag{10.206}$$

$$\begin{aligned} T^{max}_{i,j,k-\frac{1}{2}} = \max\Big( & T_{i,j,k-\frac{1}{2}}, T_{i+1,j,k-\frac{1}{2}}, T_{i-1,j,k-\frac{1}{2}}, T_{i,j+1,k-\frac{1}{2}}, T_{i,j-1,k-\frac{1}{2}}, T_{i,j,k+\frac{1}{2}}, T_{i,j,k-\frac{3}{2}}, \\ & T^{(1)}_{i,j,k-\frac{1}{2}}, T^{(1)}_{i+1,j,k-\frac{1}{2}}, T^{(1)}_{i-1,j,k-\frac{1}{2}}, T^{(1)}_{i,j+1,k-\frac{1}{2}}, T^{(1)}_{i,j-1,k-\frac{1}{2}}, T^{(1)}_{i,j,k+\frac{1}{2}}, T^{(1)}_{i,j,k-\frac{3}{2}}\Big), \end{aligned} \tag{10.207}$$

$$\begin{aligned} T^{min}_{i,j,k-\frac{1}{2}} = \min\Big( & T_{i,j,k-\frac{1}{2}}, T_{i+1,j,k-\frac{1}{2}}, T_{i-1,j,k-\frac{1}{2}}, T_{i,j+1,k-\frac{1}{2}}, T_{i,j-1,k-\frac{1}{2}}, T_{i,j,k+\frac{1}{2}}, T_{i,j,k-\frac{3}{2}}, \\ & T^{(1)}_{i,j,k-\frac{1}{2}}, T^{(1)}_{i+1,j,k-\frac{1}{2}}, T^{(1)}_{i-1,j,k-\frac{1}{2}}, T^{(1)}_{i,j+1,k-\frac{1}{2}}, T^{(1)}_{i,j-1,k-\frac{1}{2}}, T^{(1)}_{i,j,k+\frac{1}{2}}, T^{(1)}_{i,j,k-\frac{3}{2}}\Big), \end{aligned} \tag{10.208}$$

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

### 10.7.3 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 11

# SGS parameterization of lateral mixing of tracers

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

## 11.1 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 (24.22) and (24.23), the harmonic-type diffusivity is represented for the horizontal diffusion, Eq. (9.6), as follows:

$$
\begin{aligned}
\mathcal{D}_H(T) &= -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial(h_\psi z_s F^T_\mu)}{\partial\mu}+\frac{\partial(h_\mu z_s F^T_\psi)}{\partial\psi}\right\}\\
&= \frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial\mu}\left(\frac{h_\psi z_s \kappa_H}{h_\mu}\frac{\partial T}{\partial\mu}\right)+\frac{\partial}{\partial\psi}\left(\frac{h_\mu z_s \kappa_H}{h_\psi}\frac{\partial T}{\partial\psi}\right)\right\},
\end{aligned}
\tag{11.1}
$$

where $\kappa_H$ is the horizontal diffusion coefficient. 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):

$$
\frac{DT}{Dt}+\nabla_H\cdot\left[T\frac{\partial}{\partial z}(\kappa_T\nabla_H\rho/\rho_z)\right]+\frac{\partial}{\partial z}\left[T\nabla_H\cdot(-\kappa_T\nabla_H\rho/\rho_z)\right]=\mathcal{D}(T)+Q^T,
\tag{11.2}
$$

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

$$
\mathcal{D}(T)=\nabla\cdot(\kappa_I\mathbf{K}\nabla T),
\tag{11.3}
$$

where

$$
\mathbf{K}=\frac{1}{\rho_x^2+\rho_y^2+\rho_z^2}\begin{pmatrix}\rho_y^2+\rho_z^2 & -\rho_x\rho_y & -\rho_x\rho_z\\ -\rho_x\rho_y & \rho_x^2+\rho_z^2 & -\rho_y\rho_z\\ -\rho_x\rho_z & -\rho_y\rho_z & \rho_x^2+\rho_y^2\end{pmatrix}
\tag{11.4}
$$

$$
=\frac{1}{1+(\rho_x/\rho_z)^2+(\rho_y/\rho_z)^2}\begin{pmatrix}1+(\rho_y/\rho_z)^2 & -(\rho_x/\rho_z)(\rho_y/\rho_z) & -\rho_x/\rho_z\\ -(\rho_x/\rho_z)(\rho_y/\rho_z) & 1+(\rho_x/\rho_z)^2 & -\rho_y/\rho_z\\ -\rho_x/\rho_z & -\rho_y/\rho_z & (\rho_x/\rho_z)^2+(\rho_y/\rho_z)^2\end{pmatrix},
\tag{11.5}
$$

(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 $\kappa_I$. Diapycnal diffusion is not considered here.

The second and third terms on the l.h.s. of (11.2) have the form of advection terms with a transport velocity vector

$(u_T, v_T, w_T)$:

$$u_T \equiv \frac{\partial}{\partial z}\left(\kappa_T \frac{1}{h_\mu}\frac{\partial \rho}{\partial \mu}\Big/\frac{\partial \rho}{\partial z}\right), \tag{11.6}$$

$$v_T \equiv \frac{\partial}{\partial z}\left(\kappa_T \frac{1}{h_\psi}\frac{\partial \rho}{\partial \psi}\Big/\frac{\partial \rho}{\partial z}\right), \tag{11.7}$$

$$w_T \equiv -\frac{1}{h_\mu h_\psi}\left\{\frac{\partial}{\partial \mu}\left(\kappa_T \frac{h_\psi}{h_\mu}\frac{\partial \rho}{\partial \mu}\Big/\frac{\partial \rho}{\partial z}\right) + \frac{\partial}{\partial \psi}\left(\kappa_T \frac{h_\mu}{h_\psi}\frac{\partial \rho}{\partial \psi}\Big/\frac{\partial \rho}{\partial z}\right)\right\}, \tag{11.8}$$

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

Note that these could be rewritten as

$$\mathcal{G}(T) = \nabla \cdot (\kappa_T \mathbf{A} \nabla T) \tag{11.9}$$

with

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\rho_x/\rho_z \\ 0 & 0 & -\rho_y/\rho_z \\ \rho_x/\rho_z & \rho_y/\rho_z & 0 \end{pmatrix}. \tag{11.10}$$

Comparing with (11.5), we notice that the isopycnal diffusion and the thickness diffusion terms are combined to yield a concise form (Griffies, 1998) and (11.2) can be rewritten as:

$$\frac{DT}{Dt} = \nabla \cdot \{(\kappa_I \mathbf{K} - \kappa_T \mathbf{A})\nabla T\} + Q^T. \tag{11.11}$$

Three types of lateral 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 11.4 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 (11.11) is given by taking the finite volume approach as follows:

$$\begin{aligned} T^{n+1}_{i,j,k-\frac{1}{2}} \Delta V^{n+1}_{i,j,k-\frac{1}{2}} = {} & T^{n-1}_{i,j,k-\frac{1}{2}} \Delta V^{n-1}_{i,j,k-\frac{1}{2}} + 2\,\Delta t\,\{FXD_{i-\frac{1}{2},j,k-\frac{1}{2}} - FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} + FYD_{i,j-\frac{1}{2},k-\frac{1}{2}} - FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} \\ & + FZD_{i,j,k} - FZD_{i,j,k-1}\} + (\text{other terms}). \end{aligned} \tag{11.12}$$

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 23.

## 11.2 Horizontal diffusion

### 11.2.1 Laplacian diffusion

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

$$FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} = -\kappa_H \Delta y_{i+\frac{1}{2},j} \overline{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{y} \delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{11.13}$$

$$FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} = -\kappa_H \Delta x_{i,j+\frac{1}{2}} \overline{\Delta z_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{x} \delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{11.14}$$

where

$$\delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}} \equiv \frac{T_{i+1,j,k-\frac{1}{2}} - T_{i,j,k-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2},j}}, \tag{11.15}$$

$$\delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}} \equiv \frac{T_{i,j+1,k-\frac{1}{2}} - T_{i,j,k-\frac{1}{2}}}{\Delta y_{i,j+\frac{1}{2}}}. \tag{11.16}$$

### 11.2.2 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:

$$FXD_{i+\frac{1}{2},j,k-\frac{1}{2}} = \sqrt{\kappa_b}\,\Delta y_{i+\frac{1}{2},j}\overline{\Delta z_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{y}\,\delta_x\,[\mathsf{Lap}(T)]_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{11.17}$$

$$FYD_{i,j+\frac{1}{2},k-\frac{1}{2}} = \sqrt{\kappa_b}\,\Delta x_{i,j+\frac{1}{2}}\overline{\Delta z_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{x}\,\delta_y\,[\mathsf{Lap}(T)]_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{11.18}$$

where $\kappa_b$ is diffusivity coefficient and

$$\mathsf{Lap}(T)_{i,j,k-\frac{1}{2}} = \frac{\Delta x_{i,j}\Delta y_{i,j}}{\Delta V_{i,j,k-\frac{1}{2}}}(\delta_x\sqrt{\kappa_b}\,\overline{\Delta z_{i,j,k-\frac{1}{2}}}^{y}\,\delta_x T_{i,j,k-\frac{1}{2}} + \delta_y\sqrt{\kappa_b}\,\overline{\Delta z_{i,j,k-\frac{1}{2}}}^{x}\,\delta_y T_{i,j,k-\frac{1}{2}}). \tag{11.19}$$

### 11.2.3 Specification of coefficient

The diffusion coefficient of horizontal diffusion is specified using the namelists listed on Tables 11.1 and 11.2. This must be zero if ISOPYCNAL option is selected, unless the horizontal diffusion is applied intentionally. By using SMAGHD option, the diffusion coefficient can be given based on the horizontal viscosity coefficient according to the Smagorinsky scheme (Section 8.3.3).

Table11.1 Namelist nml_tracer_diff_horz

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

Table11.2 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. | |

## 11.3 Isopycnal diffusion

In isopycnal diffusion, the diffusion flux is expressed by high diffusivity along the isopycnal surface $\kappa_I$, low diffusivity across the isopycnal surface $\kappa_D$, 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

$$\mathbf{K} = \frac{\kappa_I}{1+S^2}\begin{pmatrix} 1+S_y^2+\epsilon S_x^2 & (\epsilon-1)S_xS_y & (1-\epsilon)S_x \\ (\epsilon-1)S_xS_y & 1+S_x^2+\epsilon S_y^2 & (1-\epsilon)S_y \\ (1-\epsilon)S_x & (1-\epsilon)S_y & \epsilon+S^2 \end{pmatrix}, \tag{11.20}$$

where

$$\mathbf{S} = (S_x, S_y, 0) = \left( -\frac{\dfrac{\partial \rho}{\partial x}}{\dfrac{\partial \rho}{\partial z}}, -\frac{\dfrac{\partial \rho}{\partial y}}{\dfrac{\partial \rho}{\partial z}}, 0 \right), \tag{11.21}$$

$$S = \mid \mathbf{S} \mid, \tag{11.22}$$

$$\epsilon = \frac{\kappa_D}{\kappa_I} \tag{11.23}$$

(Redi, 1982). This diffusion tensor is rewritten as the sum of horizontal diffusion tensor and a difference from the horizontal diffusion:

$$\mathbf{K} = \kappa_I \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{\kappa_I}{1+S^2} \begin{pmatrix} (\epsilon-1)S_x^2 & (\epsilon-1)S_x S_y & (1-\epsilon)S_x \\ (\epsilon-1)S_x S_y & (\epsilon-1)S_y^2 & (1-\epsilon)S_y \\ (1-\epsilon)S_x & (1-\epsilon)S_y & \epsilon+S^2 \end{pmatrix}. \tag{11.24}$$

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 11.1):

$$(\delta_x T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \delta_x T_{i+\frac{1}{2},j,k-\frac{1}{2}}, \tag{11.25}$$

$$(\delta_y T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \overline{\delta_y T_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{xy}, \tag{11.26}$$

$$(\delta_z T)_{i+\frac{1}{2},j,k-\frac{1}{2}} = \overline{\delta_z T_{i+\frac{1}{2},j,k-\frac{1}{2}}}^{xz}. \tag{11.27}$$

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

$$(\delta_x T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \overline{\delta_x T_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{xy}, \tag{11.28}$$

$$(\delta_y T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \delta_y T_{i,j+\frac{1}{2},k-\frac{1}{2}}, \tag{11.29}$$

$$(\delta_z T)_{i,j+\frac{1}{2},k-\frac{1}{2}} = \overline{\delta_z T_{i,j+\frac{1}{2},k-\frac{1}{2}}}^{yz}, \tag{11.30}$$

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

$$(\delta_x T)_{i,j,k} = \overline{\delta_x T_{i,j,k}}^{xz}, \tag{11.31}$$

$$(\delta_y T)_{i,j,k} = \overline{\delta_y T_{i,j,k}}^{yz}, \tag{11.32}$$

$$(\delta_z T)_{i,j,k} = \delta_z T_{i,j,k}. \tag{11.33}$$

The density gradient in the calculation of each component of the diffusion tensor can be obtained by replacing $T$ in the above equations with $\rho$. 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 11.4) 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 (11.20) is

$$FZD_{i,j,k} = (\mathtt{areat})_{i,j,k} \frac{\kappa_I(\epsilon + S^2)}{1+S^2} \delta_z T_{i,j,k}. \tag{11.34}$$

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

$$\kappa_{\text{eff}} = \frac{\kappa_D + \kappa_I S^2}{1+S^2}. \tag{11.35}$$

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

$$\kappa_{\text{eff}} \sim 10^3\ \mathrm{cm^2\,s^{-1}}. \tag{11.36}$$

This is a fairly large value which warrants use of an implicit scheme (Section 23.5). In MRI.COM, this term is separated from other terms and solved with other vertical diffusion terms using an implicit method.

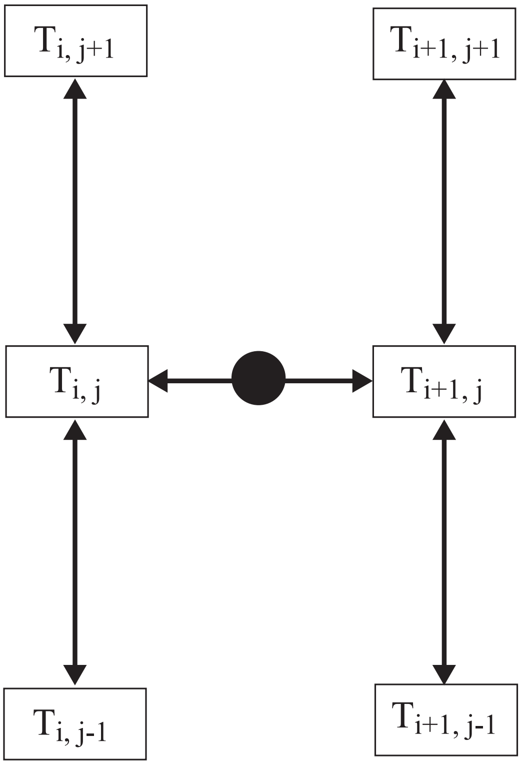

Figure11.1 The way of calculating the gradient at the circle $(i+\frac{1}{2}, j, k-\frac{1}{2})$ in isopycnal diffusion: the east-west gradient is indicated by an arrow through the circle, and the north-south gradient is given by averaging four arrows in the vertical direction.

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.

### 11.3.1 Tapering isopycnal diffusion tensor

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 as noted above. Griffies (2004) shows that such slope clipping could lead to an unrealistically large tracer flux. Another method to prevent numerical instability is to introduce a tapering factor to isopycnal diffusion tensor by specifying `ISOTAPER` option. This tapering factor $f$ is a product of two kinds of factors ($f = f_{\mathrm{steep}} f_{\mathrm{surface}}$) introduced in the subsequent sections. The equation 11.24 is rewritten as follows:

$$\mathbf{K} = f_{\mathrm{hdiag}}\kappa_I \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{f\kappa_I}{1+S^2} \begin{pmatrix} (\epsilon-1)S_x^2 & (\epsilon-1)S_xS_y & (1-\epsilon)S_x \\ (\epsilon-1)S_xS_y & (\epsilon-1)S_y^2 & (1-\epsilon)S_y \\ (1-\epsilon)S_x & (1-\epsilon)S_y & \epsilon+S^2 \end{pmatrix}. \tag{11.37}$$

By default, the factor $f_{\mathrm{hdiag}}$ is set to unity. This means that the isopycnal diffusion is rendered the horizontal diffusion as the factor $f$ reduces to zero. One may apply the tapering factor $f$ to the horizontal diagonal terms by setting `l_apply_hdiag = .true.` in namelist `nml_tracer_diff_isopy_taper` (Table 11.5). Note that the tapering factor is defined at each cell boundary where isopycnal diffusion flux is calculated and $\epsilon = \kappa_D/(f\kappa_I)$ in Eq. 11.37.

#### a. Around steep isopycnal slopes

Danabasoglu and McWilliams (1995) propose a factor that uses a hyperbolic tangent to exponentially taper isopycnal diffusion in steep slope regions as follows:

$$f_{\mathrm{steep}} = \frac{1}{2}\left\{1 + \tanh\left(\frac{S_{\mathrm{center}} - |\mathbf{S}|}{S_{\mathrm{width}}}\right)\right\}. \tag{11.38}$$

Two parameters $S_{\mathrm{center}}$ and $S_{\mathrm{width}}$, which determine a transitional region, are given by center_transition and width_transition in namelist nml_tracer_diff_isopy_taper, respectively (Table 11.5).

b. Around the sea surface

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

$$D = R|\mathbf{S}|, \tag{11.39}$$

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 other factor $f_{\mathrm{surface}}$ proposed by Large et al. (1997) introduces this constraint to the isopycnal diffusion tensor and is calculated as follows:

$$f_{\mathrm{surface}} = \frac{1}{2}\left\{1 + \sin\pi\left(r - \frac{1}{2}\right)\right\}, \tag{11.40}$$

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): $d = -z - \mathrm{BLD}$. 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 11.5).

### 11.3.2 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 11.3. We can use different slope maximal limits for isopycnal diffusion and GM parameterization that are specified using the namelist listed on Table 11.4. Configurations of surface tapering for the isopycnal diffusion scheme with ISOTAPER option are specified using the namelist listed on Table 11.5.

Table11.3 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 |

Table11.4 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 |

Table11.5 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 |

Continued on next page

Table 11.5 – continued from previous page

| variable name | units | description | usage |
|---|---|---|---|
| `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` |

## 11.4 Gent and McWilliams parameterization for eddy-induced transport

### 11.4.1 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:

$$R(T) = \partial_m (J^{mn} \partial_n T) \tag{11.41}$$

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 $F^m_{\text{skew}} = -A^{mn}\partial_n T$ is as follows:

$$\begin{aligned} R_A(T) &= \partial_m (A^{mn} \partial_n T) \\ &= \partial_m (A^{mn}) \partial_n T \\ &= \partial_n (\partial_m A^{mn} T), \end{aligned} \tag{11.42}$$

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 $\mathbf{F}_{\text{adv}} = \mathbf{u}_* T$ 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

$$\mathbf{u}_* = -\frac{\partial}{\partial \rho}(\kappa_{\text{GM}} \nabla_\rho h) \Big/ \frac{\partial h}{\partial \rho}, \tag{11.43}$$

where $h$ is the depth of the neutral surface ($\rho = \text{const}$). This velocity is expressed in the depth coordinate as

$$\mathbf{u}_* = \left(-\partial_z(\kappa_{\text{GM}} S_x),\ -\partial_z(\kappa_{\text{GM}} S_y),\ \nabla_h \cdot (\kappa_{\text{GM}} \mathbf{S})\right) \tag{11.44}$$

where

$$\mathbf{S} = (S_x, S_y, 0) = \left(-\rho_x/\rho_z, -\rho_y/\rho_z, 0\right). \tag{11.45}$$

(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 $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & -\kappa_{\text{GM}} S_x \\ 0 & 0 & -\kappa_{\text{GM}} S_y \\ \kappa_{\text{GM}} S_x & \kappa_{\text{GM}} S_y & 0 \end{pmatrix}, \tag{11.46}$$

so that

$$\frac{\partial T}{\partial t} = \cdots \quad + \nabla \cdot (\mathbf{A} \nabla T). \tag{11.47}$$

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

$$\Psi = \kappa_{\text{GM}} \hat{\mathbf{z}} \times \mathbf{S} = (-\kappa_{\text{GM}} S_y,\ \kappa_{\text{GM}} S_x,\ 0), \tag{11.48}$$

which produces $\mathbf{u}_*$ in (11.44):

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

The skew diffusive expression for the flux using tensor $\mathbf{A}$ in (11.46) is

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

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).

### 11.4.2 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, $[\kappa_{\text{GM}}]_{\text{ref}}$, is given by `diff_thick_cm2ps` in namelist `nml_tracer_diff_isopy` (Table 11.3). 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 11.8).

#### a. Simple scheme

If `l_grid_size_change_mix_coefs` = `.true.` in namelist `nml_grid_size_change_mix_coefs` (Table 11.7), the coefficient may be dependent on the horizontal grid size according to the following formula

$$\kappa_{\text{GM}} = [\kappa_{\text{GM}}]_{\text{ref}} \times \min(\Delta x, \Delta y)/(100\,\text{km}), \tag{11.49}$$

where $\Delta x$ and $\Delta y$ 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), specify `l_visbeck` = `.true.` in namelist `nml_gmvar_select`.

Visbeck et al. (1997) proposed to give the GM coefficient $\kappa_{\text{GM}}$ as

$$\kappa_{\text{GM}} = \alpha \frac{M^2}{N} l^2, \tag{11.50}$$

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, $\rho_0$ reference density.

Specifically in our model,

$$M^2 = \frac{g}{\rho_0 D}\left[\left(\int_{-H_1}^{-H_0} \frac{\partial \rho}{\partial x} dz\right)^2 + \left(\int_{-H_1}^{-H_0} \frac{\partial \rho}{\partial y} dz\right)^2\right]^{\frac{1}{2}}, \tag{11.51}$$

and

$$N^2 = \frac{g\left[\sigma_0(H_1) - \sigma_0(H_0)\right]}{\rho_0 D}, \tag{11.52}$$

where $H_0 = 100$ m, $H_1 = 2000$ m, $D = H_1 - H_0$, and $\sigma_0$ is the potential density. Lower limit for $N$ is set so that $N^2 \geq 10^{-9}\ \text{s}^{-1}$.

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

$$c_1 = \frac{1}{\pi}\int_{-H_B}^{0} \left(-\frac{g}{\rho_0}\frac{\partial \sigma_0}{\partial z}\right)^{\frac{1}{2}} dz, \tag{11.53}$$

where $H_B$ is the depth of sea floor, deformation radius $\lambda_1$ is calculated as follows:

$$\lambda_1 = \min\left(\left|\frac{c_1}{f}\right|, 4 \times 10^4\right) \quad \text{m}. \tag{11.54}$$

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

$$\kappa_{\text{GM}} = \alpha \frac{M^2}{N} (r\lambda_1)^2\,. \tag{11.55}$$

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

$$300 \leq \kappa_{\text{GM}} \leq 1500 \quad \text{m}^2\,\text{s}^{-1}. \tag{11.56}$$

### c. Eden and Greatbatch (2008)

To use the method proposed by Eden and Greatbatch (2008), specify `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

$$\kappa_{\mathrm{GM}} = cL^2\sigma. \tag{11.57}$$

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 theoretical considerations (Theiss, 2004). $L_{Rhi}$ is estimated from variables of the coarse resolution model as

$$L_{Rhi} = \frac{\sigma}{\beta} \tag{11.58}$$

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

$$L_r = \min\left[\frac{c_1}{|f|}, \sqrt{\frac{c_1}{2\beta}}\right], \tag{11.59}$$

where $c_1$ denotes the 1st baroclinic gravity wave speed calculated approximately as eq. (11.53). Considering the thermal wind relation in mid-latitudes, Eden and Greatbatch (2008) proposed that the inverse eddy time scale $\sigma$ is given by

$$\sigma = f(\mathrm{Ri} + \gamma)^{-\frac{1}{2}}. \tag{11.60}$$

Here, $\mathrm{Ri} = N^2|\partial_z u_h|^{-2}$ 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 $\sigma$ and consequently for $\kappa_{\mathrm{GM}}$. The default values of $\gamma$ and $c$ in eq. (11.57), 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 $\kappa_{\mathrm{GM}}$ using

$$\kappa_{\mathrm{GM}} = \left[\frac{N^2}{N^2_{\mathrm{ref}}}\right][\kappa_{\mathrm{GM}}]_{\mathrm{ref}}, \tag{11.61}$$

where $N$ is the local buoyancy frequency and $[\kappa_{\mathrm{GM}}]_{\mathrm{ref}}$ is the constant reference value of $\kappa_{\mathrm{GM}}$ within the surface diabatic layer. $N_{\mathrm{ref}}$ 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 $N^2/N^2_{\mathrm{ref}}$ 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

$$N_{\min} \leq \frac{N^2}{N^2_{\mathrm{ref}}} \leq 1.0, \tag{11.62}$$

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

## 11.4.3 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 ($\kappa_{\mathrm{GM}}$) is linearly reduced from the value at the base of mixed layer to zero at the sea surface within the mixed layer,

$$\kappa_{\mathrm{GM}}(z) = \kappa_{\mathrm{GM}}(z = -\mathrm{MLD}) \times (-z/\mathrm{MLD}) \quad \text{for} \quad -\mathrm{MLD} \leq z \leq 0, \tag{11.63}$$

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 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 ($|\mathbf{S}|$):

$$D = R|\mathbf{S}|. \tag{11.64}$$

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

$$\mathrm{DLD} = \mathrm{BLD} + D. \tag{11.65}$$

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

$$\Psi_{\mathrm{BL}} = \frac{\eta - z}{\eta + \mathrm{BLD}}\Psi_0 \quad \text{for} \quad -\mathrm{BLD} \le z \le \eta \tag{11.66}$$

and

$$\Psi_{\mathrm{TL}} = \left(\frac{z + \mathrm{BLD}}{\mathrm{TLT}}\right)^2 \Phi + \left(\frac{\eta - z}{\eta + \mathrm{BLD}}\right)\Psi_0 \quad \text{for} \quad -\mathrm{DLD} \le z < -\mathrm{BLD} \tag{11.67}$$

The two functions $\Psi_0$ and $\Phi$ are chosen such that $\Psi$ and its vertical derivative are continuous across the base of BLD and the base of TLT. These constraints then yield

$$\Psi_0 = \frac{\eta + \mathrm{BLD}}{2(\eta + \mathrm{BLD}) + \mathrm{TLT}}(2\Psi_I + \mathrm{TLT}\partial_z\Psi_I) \tag{11.68}$$

and

$$\Phi = -\frac{\mathrm{TLT}}{2(\eta + \mathrm{BLD}) + \mathrm{TLT}}(\Psi_I + (\eta + \mathrm{DLD})\partial_z\Psi_I), \tag{11.69}$$

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

$$\Psi_I = -\kappa_{\mathrm{GM}}\frac{\mathbf{z} \times \nabla_H \rho}{\partial_z \rho} \quad \text{at} \quad z = -\mathrm{DLD}. \tag{11.70}$$

In the implementation, to evaluate both $\Psi_I$ and $\partial_z\Psi_I$ at $z = -\mathrm{DLD}$, $\Psi_I$ 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`.

### 11.4.4 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:

$$\mathbf{K}_2 = \begin{pmatrix} L & M \\ M & N \end{pmatrix} = \begin{pmatrix} \kappa_A n_x^2 + \kappa_B n_y^2 & \kappa_B n_x n_y \\ \kappa_B n_x n_y & \kappa_B n_x^2 + \kappa_A n_y^2 \end{pmatrix}, \tag{11.71}$$

where $\kappa_A$ is the diffusivity in the $\hat{\mathbf{n}}$ direction, and $\kappa_B$ 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),

$$\mathbf{A}' = \begin{pmatrix} 0 & 0 & -LS_x - MS_y \\ 0 & 0 & -MS_x - NS_y \\ LS_x + MS_y & MS_x + NS_y & 0 \end{pmatrix}. \tag{11.72}$$

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

### 11.4.5 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 11.6

Table11.6 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 11.7.

Table11.7 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 11.8 through 11.12.

Table11.8 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 |

Table11.9 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 | $\mathrm{cm}^2\,\mathrm{sec}^{-1}$ | upper limit of thickness diffusivity | l_visbeck = .true. |
| lower_limit | $\mathrm{cm}^2\,\mathrm{sec}^{-1}$ | 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.<br>constant horizontal length scale when lcalc_defrad = .false. | l_visbeck = .true. |
| length_factor | 1 | [horizontal length scale] = [deformation radius] × length_factor | l_visbeck = .true. |

Table11.10 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. |

Table11.11 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. |

Table11.12 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 11.13.

Table11.13 Namelist nml_gm_anisotrop

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

## Chapter 12

# SGS parameterization of vertical mixing of tracers

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

## 12.1 Vertical diffusion

The following is the equation from the advection-diffusion equation (9.4) with only the time-varying term and vertical diffusion term ($\mathcal{D}_V(T)$),

$$\frac{\partial(z_s T)}{\partial t} = \mathcal{D}_V(T). \tag{12.1}$$

Vertical diffusion term takes the form of Laplacian and the vertical diffusion flux is proportional to the vertical gradient of tracer. The vertical diffusion, Eq. (9.6),

$$\mathcal{D}_V(T) = -\frac{\partial F_z^T}{\partial s} = \frac{\partial}{\partial s}\left(\frac{\kappa_V}{z_s}\frac{\partial T}{\partial s}\right). \tag{12.2}$$

Here, $\kappa_V$ is the vertical diffusion coefficient.

These equations yield the finite difference form as follows:

$$T^{n+1}_{i,j,k-\frac{1}{2}}\Delta V^{n+1}_{i,j,k-\frac{1}{2}} = T^{n-1}_{i,j,k-\frac{1}{2}}\Delta V^{n-1}_{i,j,k-\frac{1}{2}} + 2\Delta t(FZD_{i,j,k} - FZD_{i,j,k-1}), \tag{12.3}$$

$$FZD_{i,j,k} = -\kappa_z(\mathtt{areat})_{i,j,k+\frac{1}{2}}\delta_z T_{i,j,k}, \tag{12.4}$$

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

$$\delta_z T_{i,j,k} \equiv \frac{T_{i,j,k-\frac{1}{2}} - T_{i,j,k+\frac{1}{2}}}{\Delta z_k}. \tag{12.5}$$

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 $\Delta z_k$ when calculating the gradient.

In most realistic simulations, a backward (implicit) scheme is used in the time integration (VVDIMP option; Section 23.5) because high diffusivity is expected owing to the choice of parameterizations needed for realistic simulations. Otherwise, a forward scheme is used.

### 12.1.1 Specification of coefficient

Background vertical diffusivity, which is horizontally uniform, a function of depth, and fixed in time, should be always given. Additionally, non-time-varying, a three dimensional distribution can be set by selecting VMBG3D option to incorporate locally enhanced mixing processes in the climatology 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 12.1 and 12.2 summarizes how to give background vertical diffusivity.

Table12.1 Namelist nml_diff_vert_bg. Specify only one of the two variables

| variable name | units | description | usage |
|---|---|---|---|
| diff_vert_bg_cm2ps | $\mathrm{cm}^2\,\mathrm{s}^{-1}$ | 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 |

Table12.2 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 | |
| l_vmintpol | logical | interpolate input data to model grid points | |
| slatvm | | latitude of the southern end | only if l_vmintpol = .true. |
| slonvm | | longitude of the western 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 time-varying vertical diffusivity coefficients at every model time step.

- Surface mixed layer models (TURBULENCE option).
- Vertical component of isopycnal diffusion (ISOPYCNAL option).
- Enhanced diffusivity (= $1.0\,\mathrm{m}^2\,\mathrm{s}^{-1} = 10^4\,\mathrm{cm}^2\,\mathrm{s}^{-1}$) where the stratification is unstable (DIFAJS option).
- Enhanced diffusivity around rivermouths to avoid too low salinity if the model receives river run-off option (RUNOFF option). This scheme is especially needed when positive salinity is not guaranteed by a tracer advection algorithm. See Table 12.3 for how to specify the mixing.

The vertical diffusion for "this" time step is taken as the largest of the above estimations.（合計ではない).

Table12.3 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 | $\mathrm{cm}^2\,\mathrm{sec}^{-1}$ | maximum value of the enlarged diffusivity (= $\kappa_{\mathrm{rivmax}}$) | default = $1 \times 10^4\,\mathrm{cm}^2\,\mathrm{sec}^{-1}$ |
| depth_max_vmixriv_cm | cm | vertical diffusion is enlarged from surface to this depth, this is also used by subroutine salinity_limit | default = $30 \times 10^2$ 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 $W_{\mathrm{river}}$ is river discharge rate in $\mathrm{cm\,sec}^{-1}$ | default = 7 |

## 12.2 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 smoothes the distribution.

In general, there are three numerical schemes for convective adjustment.

1. 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. Therefore, this scheme is not used in MRI.COM.
2. 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 $10^4$ cm$^2$ s$^{-1}$. This scheme is the most standard used in ocean models worldwide.
3. 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).

### 12.2.1 Algorithm

In order to minimize the judgment process ("IF" statement) and replace it by arithmetic calculation, this scheme defines two integer indices, $\alpha_k$ and $\beta_k$, 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 12.1). The level at the vertical boundary of a T-cell corresponds to the integer $k$. The index $\alpha_k$ 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 $\beta_k$ 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 $\alpha$ above the level $k$ and below the level $k$, respectively, and are expressed by the following recursive relation.

$$\begin{aligned}
VU_1 &= \alpha_1 V_{\frac{1}{2}}, \\
VU_2 &= \alpha_2 (V_{1+\frac{1}{2}} + \alpha_1 V_{\frac{1}{2}}) = \alpha_2 (V_{1+\frac{1}{2}} + VU_1), \\
&\cdots, \\
VU_k &= \alpha_k (V_{k-\frac{1}{2}} + VU_{k-1}), \\
&\cdots, \\
VU_{KM-1} &= \alpha_{KM-1} (V_{KM-1-\frac{1}{2}} + VU_{KM-2}),
\end{aligned} \tag{12.6}$$

and

$$\begin{aligned}
VL_{KM-1} &= \alpha_{KM-1} V_{KM-\frac{1}{2}}, \\
VL_{KM-2} &= \alpha_{KM-2} (V_{KM-1-\frac{1}{2}} + \alpha_{KM-1} V_{KM-\frac{1}{2}}) = \alpha_{KM-2} (V_{KM-1-\frac{1}{2}} + VL_{KM-1}), \\
&\cdots, \\
VL_k &= \alpha_k (V_{k+\frac{1}{2}} + VL_{k+1}), \\
&\cdots, \\
VL_1 &= \alpha_1 (V_{1+\frac{1}{2}} + VL_2),
\end{aligned} \tag{12.7}$$

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:

$$\begin{aligned}
&TU_1 = \alpha_1 T_{\frac{1}{2}} V_{\frac{1}{2}}, \quad TU_k = \alpha_k (T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + TU_{k-1}),\\
&SU_1 = \alpha_1 S_{\frac{1}{2}} V_{\frac{1}{2}}, \quad SU_k = \alpha_k (S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + SU_{k-1}),\\
&TL_{KM-1} = \alpha_{KM-1} T_{KM-\frac{1}{2}} V_{KM-\frac{1}{2}}, \quad TL_k = \alpha_k (T_{k+\frac{1}{2}} V_{k+\frac{1}{2}} + TL_{k+1}),\\
&SL_{KM-1} = \alpha_{KM-1} S_{KM-\frac{1}{2}} V_{KM-\frac{1}{2}}, \quad SL_k = \alpha_k (S_{k+\frac{1}{2}} V_{k+\frac{1}{2}} + SL_{k+1}),
\end{aligned} \tag{12.8}$$

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

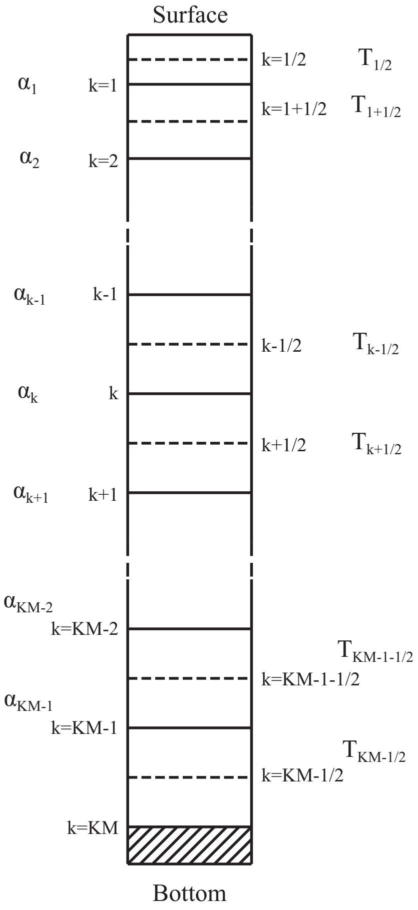

Figure12.1 Reference vertical grid points in Section 12.2

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

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

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

$$\begin{aligned}
TM_{k-\frac{1}{2},k+\frac{1}{2}} &= \frac{TU_k + TL_k}{VU_k + VL_k},\\
SM_{k-\frac{1}{2},k+\frac{1}{2}} &= \frac{SU_k + SL_k}{VU_k + VL_k},
\end{aligned} \tag{12.9}$$

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 $\alpha_{k-2} = \alpha_{k+1} = 0$,

$$
\begin{aligned}
VU_{k-1} + VL_{k-1} &= VU_k + VL_k \\
&= V_{k-1-\frac{1}{2}} + V_{k-\frac{1}{2}} + V_{k+\frac{1}{2}}, \\
TU_{k-1} + TL_{k-1} &= TU_k + TL_k \\
&= T_{k-1-\frac{1}{2}} V_{k-1-\frac{1}{2}} + T_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + T_{k+\frac{1}{2}} V_{k+\frac{1}{2}}, \\
SU_{k-1} + SL_{k-1} &= SU_k + SL_k \\
&= S_{k-1-\frac{1}{2}} V_{k-1-\frac{1}{2}} + S_{k-\frac{1}{2}} V_{k-\frac{1}{2}} + S_{k+\frac{1}{2}} V_{k+\frac{1}{2}},
\end{aligned}
\tag{12.10}
$$

and

$$
\begin{aligned}
TM_{k-1-\frac{1}{2},k+\frac{1}{2}} &= \frac{TU_{k-1} + TL_{k-1}}{VU_{k-1} + VL_{k-1}} = \frac{TU_k + TL_k}{VU_k + VL_k}, \\
SM_{k-1-\frac{1}{2},k+\frac{1}{2}} &= \frac{SU_{k-1} + SL_{k-1}}{VU_{k-1} + VL_{k-1}} = \frac{SU_k + SL_k}{VU_k + VL_k}.
\end{aligned}
\tag{12.11}
$$

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

### 12.2.2 Numerical procedure

In summary, numerical procedures are summarized as follows.

[1] Density is calculated at the intermediate depth between adjacent levels using (A) upper level temperature and salinity and (B) lower level ones. If the density using (A) is larger than using (B), $\alpha(\alpha^1)$ 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 (12.6) to (12.8), $VU$, $TU$, $SU$, $VL$, $TL$, and $SL$ are calculated using $\alpha^n$ for a water column that includes an unstable part.

[3] The vertical mean $TM$ and $SM$ are calculated for the unstable part using equation (12.9) 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 $\alpha^n$ is stored in $\beta^n$. $\beta^n = 1$ 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:

$$
\beta_k^n = \alpha_k^n + \beta_k^{n-1}(1 - \alpha_k^n). \tag{12.12}
$$

[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 $\alpha_k^{n+1}$ is done using $\beta_k^n$ 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.

$$
\begin{aligned}
&\gamma_1 = \alpha_1^{(n+1)}, \quad \gamma_k = \alpha_k^{(n+1)} + (1 - \alpha_k^{(n+1)})\beta_k^{(n)}\gamma_{k-1} \\
&\alpha_{KM-1}^{(n+1)} = \gamma_{KM-1}, \quad \alpha_k^{(n+1)} = \gamma_k + (1 - \gamma_k)\beta_k^{(n)}\alpha_{k+1}^{(n+1)},
\end{aligned}
\tag{12.13}
$$

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

Table 12.4 shows an example of the case with six levels. Static instability is removed after the three-time iteration. The second column of $\alpha$ in the table is the result of the corrected $\alpha_k^{n+1}$ using $\beta_k^n$ based on equation (12.13), as described in [6]. Note that $\beta_k^0 = 0$, though there is no description in the table.

Table12.4 Example of the convective adjustment procedure

| n | k | $\alpha$ | | VU | VL | VU+VL | TU+TL | $\beta$ |
|---|---|---|---|---|---|---|---|---|
| 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 13

# Tracer Package Structure and Usage

MRI.COM handles a wide variety of tracers, including physical variables for temperature and salinity, ecosystem model variables, and passive tracers. The user can individually specify initial values, advection schemes, body and surface forcings, etc. for each tracer. This chapter provides an overview of the program as well as common methods for their specification.

## 13.1 Program package structure

Program packages relevant to tracers are listed as follows.

### 13.1.1 Tracer equation

| | |
|---|---|
| Core/tracer_ctl.F90: | Controller of this package |
| Core/tracer.F90: | Main program of this package |
| Core/tracer_vars.F90: | Setting of tracer attributes |
| Core/upc_adv.F90: | Upcurrent advection scheme |
| Core/quick_adv.F90: | QUICK advection scheme (QUICKADVEC) |
| Core/utzq_adv.F90: | Combination of UTOPIA and QUICKEST advection scheme (UTZQADVEC) |
| Core/som_adv.F90: | Second order moment advection scheme (SOMADVEC) |
| Core/mpdata_adv.F90: | MPDATA advection scheme (MPDATAADVEC) |
| Core/ppm_adv.F90: | PPM advection scheme (PPMADVEC) |
| +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

| | |
|---|---|
| Core/vmixcoef_ctl.F90: | Controller of the vertical mixing package |
| Core/vmixcoef.F90: | Main program of the vertical mixing package |
| Core/vmixcoef_vmbg.F90: | Estimation of background vertical diffusion coefficient |
| Core/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

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

### 13.1.4 Reference state and restoring coefficient

| | |
|---|---|
| Core/restore_cond_ctl.F90: | Controller of reference state and restoring coefficient |
| Core/restore_cond.F90: | Main program of reference state and restoring coefficient |

| | |
|---|---|
| Core/force_data.F90: | Service package that handles external forcing data |

### 13.1.5 Passive tracers

| | |
|---|---|
| Core/ptrc_ctl.F90: | Controller of passive tracer evolution |
| Core/ptrc.F90: | Main program of passive tracer evolution that mainly treats surface sources and sinks |
| Core/ptrc_subp.F90: | Sub package of passive tracer evolution that describes internal sources and sinks |
| +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` and `nml_restart`. 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 `nml_restart` for the initial condition.<br>`.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`.<br>`.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`<br>`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 specified using a 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_vars.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)
- "ppm" : Picewise Parabolic advection scheme (PPMADVEC)
- "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 used for the corrector phase. | 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_ppm_org | logical | Use flux limiter for PPM by Colella and Woodward (1984) | PPMADVEC |
| adv_scheme%limiter_ppm_lin | logical | Use flux limiter for PPM by Lin et al. (1994) | PPMADVEC |
| adv_scheme%limiter_mpdata_nonoscillatory | logical | Apply flux limiter for MPDATA | MPDATAADVEC |
| adv_scheme%eps_lim_mpdata | same as tracer unit | Very small value to avoid zero division | MPDATAADVEC |
| adv_scheme%min_value_mpdata | same as tracer unit | Minimum value for tracer | MPDATAADVEC |
| adv_scheme_predictor%name | character | Name of the advection scheme used for the predictor phase. Only active tracers ($\theta$&S) need to be configured. | Since accuracy is not so demanding, QUICKADVEC is recommended due to its small computational cost. |

### 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<br>−1 : monthly<br>−999 : steady forcing |
| `trcref_conf%num_data_max` | integer | the number of record contained in the file | |
| `trcref_conf%ifstart` | integer, dimension (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` | logical | 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` | logical | 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<br>−1 : monthly<br>−999 : steady forcing |
| `trcref_surf_conf%num_data_max` | integer | the number of record contained in the file | |
| `trcref_surf_conf%ifstart` | integer, dimension (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<br>$-999$ : steady forcing |
| `rstcoef_conf%num_data_max` | integer | the number of record contained in the file | |
| `rstcoef_conf%ifstart` | integer, dimension (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 $\text{sec}^{-1}$.

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<br>$-1$ : monthly<br>$-999$ : steady forcing |
| `rstcoef_surf_conf%num_data_max` | | the number of record contained in the file | |
| `rstcoef_surf_conf%ifstart` | integer, dimension (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.,
 adv_scheme_predictor%name="quick",

! 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.,
/
```