UNIFIED MODEL DOCUMENTATION NO. 51

Isopycnal diffusion schemes

Unified Model Version 4.5

Malcolm J Roberts
Ocean Applications, Met. Office, U.K.

Reviewed by : Graham Rickard
June, 1999

Ocean Applications,
The Met. Office,
London Road,
Bracknell,
Berkshire,
RG12 2SY
United Kingdom

Modification History

Date	Author	UM Version	Change Description

Contents

1 Introduction

Mixing in the ocean occurs predominantly along, rather than across, isopycnal surfaces, which is why water masses tend to preserve their temperature and salinity characteristics over very large distances. This type of mixing is thought to be done by such mesoscale processes as ocean eddies, in contrast to cross-isopycnal mixing processes such as internal wave breaking and double-diffusion.

The easiest way to orient mixing in models where geopotential height is the vertical coordinate is horizontally and vertically - and indeed this is how mixing was originally implemented in the Cox (1984) model. However in regions of steeply-sloping isopycnals, horizontal diffusion of temperature and salinity causes large cross-isopycnal transports which are unrealistic. Indeed in such models there is usually large upwelling on the inshore side of currents such as the Gulf Stream, since a balance is set up between vertical advection and horizontal diffusion - this is the so-called Veronis effect (Veronis (1975)).

In order to represent interior ocean mixing more realistically, a diffusion scheme which orients the mixing tensor to lie on an isopycnal surface, rather than a horizontal surface, was introduced into the Cox model (Redi (1982), Cox (1987); hereafter C87 refers to this implementation). Before UM vn4.5, this was the only isopycnal diffusion scheme available in our ocean model. This scheme calculates the local isopycnal slope at each model point, and proceeds to mix all tracers along and across this slope (although strictly speaking the component directed across the isopycnal tends to be included as a purely vertical diffusion).

Although a great improvement over horizontal/vertical mixing, the Redi-Cox scheme cannot be used as the only parameterization of eddy mixing. This is because of its numerical discretisation - grid-scale noise tends to develop, and can only be controlled if additional background horizontal diffusion is added. Of course, this compromises the adiabatic properties of the scheme and is therefore unwelcome.

The reasons for the numerical difficulties of the Redi-Cox scheme have been clarified by Griffies *et al.* (1998) - they note that the scheme does averaging and differencing in the same direction, and hence allows a computational mode to develop. Starting from a basic functional formalism, they derive an improved numerical scheme which does not allow computational modes to develop, and which calculates the local isopycnal slope in a better way (by calculating temperature and salinity expansion coefficients), hence preventing upgradient transfers which were possible with C87. This scheme has been incorporated into the UM at vn4.5.

Another important process which mesoscale eddies perform is the removal of potential energy from frontal regions. Isopycnal mixing alone has no effect on the density field, and hence does not represent this process. The recent scheme of Gent and McWilliams (1990) attempts to parameterise this process - implementing this scheme in tandem with the isopycnal diffusion scheme of Griffies *et al.* (1998) allows one to have zero background horizontal diffusion, and hence allow the ocean interior to be realistically adiabatic. Implementation of this scheme will be described in a future UMDP paper.

This UMDP replaces a previous version by David Carrington. All the information in that former version is either contained here or in UMDP 59 by Graham Rickard. Some of the scientific detail in this paper is adapted from the old UMDP 51, and some from the Griffies *et al.* (1998) paper.

2 Isopycnal diffusion: basic derivation

The basic tracer equation for the Cox (1984) model, involving advection, horizontal and vertical diffusion is:

$$\frac{\partial T}{\partial t} + \Gamma(T) = \frac{\partial}{\partial z}\left(\frac{A_V}{\delta'}\frac{\partial T}{\partial z}\right) + A_H \nabla_H^2 T \tag{1}$$

where T is the tracer, $\Gamma(T)$ is the advective term, A_V is the vertical mixing coefficient, A_H the horizontal mixing coefficient, ∇_H^2 the horizontal Laplacian in spherical coordinates and

$$\delta' = \begin{cases} 1, \ \frac{\partial \rho}{\partial z} < 0 \\ \\ 0, \ \frac{\partial \rho}{\partial z} > 0 \text{ (unstable.)} \end{cases}$$

Thus the diffusion scheme for stable conditions is represented by $\nabla.(\mathbf{K}.\nabla\mathbf{T})$, where

$$\mathbf{K} = \begin{bmatrix} A_H & 0 & 0 \\ 0 & A_H & 0 \\ 0 & 0 & A_V \end{bmatrix} \tag{2}$$

One can consider a similar form but in isopycnal coordinates, i.e.,

$$\frac{\partial T}{\partial t} + \Gamma(T) = \nabla'.(\mathbf{K^I}.\nabla'\mathbf{T}) \tag{3}$$

where ∇' is the gradient operator relative to the isopycnal coordinates, and:

$$\mathbf{K}^I = \begin{bmatrix} A_I & 0 & 0 \\ 0 & A_I & 0 \\ 0 & 0 & A_D \end{bmatrix} = A_I \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \epsilon \end{bmatrix}$$

where $\epsilon = A_D/A_I$, A_I is the "along slope" mixing coefficient and A_D is the relatively weak "cross isopycnal" or diapycnal mixing coefficient.

This diffusion matrix can then be rotated into the geodesic coordinate system (details of the coordinate transformation can be found in Redi (1982) and Crawley and Johns (1987)) to give:

$$\mathbf{K}^g = \frac{A_I}{1+S^2} \begin{bmatrix} 1+S_y^2+\epsilon S_x^2 & (\epsilon-1)S_xS_y & (1-\epsilon)S_x \\ (\epsilon-1)S_xS_y & 1+S_x^2+\epsilon S_y^2 & (1-\epsilon)S_y \\ (1-\epsilon)S_x & (1-\epsilon)S_y & \epsilon+S^2 \end{bmatrix}$$

where

$$\mathbf{S} = (S_x, S_y, 0) = (-\frac{\partial_x \rho}{\partial_z \rho}, -\frac{\partial_y \rho}{\partial_z \rho}, 0)$$

is the isoneutral[1]slope vector with magnitude S.

Thus this matrix needs to be calculated at every timestep to implement the isopycnal diffusion.

Since values of ϵ ($\approx 10^{-7}$ - 10^{-8}) are very small in general compared to the other values with which it occurs in additive combination, ϵ can be set to zero in the off-diagonal terms of $\mathbf{K}^g$.

The matrix can be simplified further if we make the small-slope approximation - since horizontal density gradients are always much smaller than vertical gradients, then

$$S^2 << 1 \quad \Rightarrow \quad S_y^2 + \epsilon S_x^2 < S^2 << 1$$

and similarly for $S_x^2 + \epsilon S_y^2$. Hence $\mathbf{K}^g$ can be approximated by [2].

[1]For the purposes of this document, it is assumed that isoneutral and isopycnal are interchangeable descriptions of surfaces having the same density.

[2]Note that in the MOM code, the small-slope approximation makes the K_{12} and K_{21} elements of this martix zero as well - this approximation is implemented in the Griffies *et al.* (1998) scheme below

$$\mathbf{K}^g = A_I \begin{bmatrix} 1 & -S_x S_y & S_x \\ -S_x S_y & 1 & S_y \\ S_x & S_y & \epsilon + S^2 \end{bmatrix} \tag{4}$$

The diffusion equation for tracers (for stable conditions) may therefore be written as:

$$\frac{\partial T}{\partial t} + \Gamma(T) = \nabla.(\mathbf{K}^g.\nabla T) \tag{5}$$

Hence the component of the change in tracer over a timestep due to isopycnal diffusion is in full,

$$\begin{aligned} \frac{\partial T}{\partial t} = & \frac{\partial}{\partial x} A_I \left(T_x - S_x S_y T_y + S_x T_z \right) + \frac{\partial}{\partial y} A_I \left(-S_x S_y T_x + T_y + S_y T_z \right) \\ & + \frac{\partial}{\partial z} A_I \left(S_x T_x + S_y T_y + (\epsilon + S^2) T_z \right). \end{aligned} \tag{6}$$

It now remains to discretize the isopycnal diffusion matrix.

3 Redi-Cox implementation

The standard implementation of the C87 isopycnal diffusion uses the diffusion matrix (4). However, it is found to produce unrealistic noise in the tracer fields - this is generally alleviated by adding horizonal and vertical diffusion coefficients (in the form of (2)) to the diagonal terms of the isopycnal diffusion matrix:

$$\mathbf{K}^g = A_I \begin{bmatrix} 1 & -S_x S_y & S_x \\ -S_x S_y & 1 & S_y \\ S_x & S_y & \epsilon + S^2 \end{bmatrix} + \begin{bmatrix} A_H & 0 & 0 \\ 0 & A_H & 0 \\ 0 & 0 & A_V \end{bmatrix} \tag{7}$$

The value for the vertical diffusion coefficient may be derived from some type of vertical mixing scheme, be it the "K-theory" mixing scheme (Pacanowski and Philander (1981)) or a combination of the Large *et al.* (1994) and the Peters *et al* mixing schemes (see the UMDP 59 on Vertical Mixing Schemes by Rickard (1999) for further details).

With the above matrix, (5) could be solved explicitly in a finite difference scheme. However, the small vertical grid spacing Δz used for the upper layers means that the term $\partial^2/\partial^2 z((\epsilon + S^2)T_z)$ puts a severe constraint on the timestep.

To deal with this term, Cox (1987) implemented an implicit timestep, so that the timestep could remain relatively large. In practice, this component is added to the vertical diffusion coefficient used in the model (derived from Pacanowski and Philander (1981) or other scheme) and which must be solved implicitly if isopycnal diffusion is active. The implicit scheme is described in Rickard (1999). All the remaining terms are treated explicitly. The time constraint on them is now [3]

$$\Delta t \leq \frac{\Delta z \Delta x}{A_I}.$$

which arises from the cross-terms in the operator.

3.1 Problems with C87 implementation

Griffies *et al.* (1998) examined the numerics of the C87 scheme, and found several reasons why the model solutions become noisy. The details can be found in their paper - only a summary of the results will be presented here.

The first problem concerns the discretization of the off-diagonal components of the diffusive flux. Consider the x component of the isopycnal diffusive flux in a two-dimensional (x-z) model:

$$-F^x_{i,k} = A_I \left[\delta_x T_{i,k} - \left(\frac{\delta_x \rho_{i,k}}{\delta_z \overline{\rho}^{x,z}_{i,k-1}} \right) \delta_z \overline{T}^{x,z}_{i,k-1} \right] \tag{8}$$

Now the z-derivative of the tracer is discretized in C87 as

$$\delta_z \overline{T}^{x,z}_{i,k-1} = \frac{T_{i,k-1} - T_{i,k+1} + T_{i+1,k-1} - T_{i+1,k+1}}{4dz_k}, \tag{9}$$

and the z-derivative of the density appearing in the same term has the same form. The combination of both a z-average and a z-derivative in the same term can lead to computational modes, where the grid of even and odd points can become disconnected.

This problem is particularly serious if it can amplify, and in the C87 scheme it can. Consider a two-dimensional configuration with density containing the $2\Delta x$ form $\rho_{i+1,k} = \rho_{i-1,k}$. In this case the z-component of the discretized isoneutral flux is identically zero (as one can imagine

[3] correcting an error in Cox (1987)

from the equivalent form to (9) for the horizontal gradient). Therefore the projection of the diffusive flux onto the tracer gradient is entirely zonal and consists of

$$F_x \delta_x T = -A_I (\delta_x T)^2 \left(1 - \frac{\delta_z \overline{T}^{x,z} \delta \rho}{\delta_x T \delta \overline{\rho}^{x,z}} \right). \tag{10}$$

If the slopes of the isoneutrals S_ρ and tracers S_T satisfy

$$|S_\rho| \equiv \left\| \frac{\delta_x \rho}{\delta_z \overline{\rho}^{x,z}} \right\| > \left\| \frac{\delta_x T}{\delta_z \overline{T}^{x,z}} \right\| \equiv |S_T| \tag{11}$$

then the flux on the east face will be directed up the tracer gradient. Now a fundamental property of any diffusive scheme is that it should not increase tracer variance - the reason the C87 scheme violates this is that the z-diffusive flux, which should compensate the horizontal flux and make the total flux downgradient, is identically zero.

An additional problem comes from the calculation of density and tracer gradients at different reference depths. Consider a single active tracer for which the isoneutral diffusive flux should vanish. For example, the x-component of the flux in the small angle limit is

$$F_I^x(\theta) = -A_I \left(\delta_x \theta - \frac{\delta_x \rho}{\delta_z \rho} \delta_z \theta \right) \tag{12}$$

$$= -A_I \left(\delta_x \theta - \frac{\alpha \delta_x \theta}{\alpha \delta_z \theta} \delta_z \theta \right) = 0. \tag{13}$$

In the continuum the diffusive flux vanishes because the thermal expansion coefficient α cancels. However in the C87 scheme, density gradients are calculated by referencing each density to the same depth level, without using the expansion coefficients. In this way there is no exact cancellation, and hence the computed neutral directions are not aligned properly with the temperature isotherms. Fig. 6 of Griffies *et al.* (1998) shows how this problem causes the misalignment of density surfaces, and hence the fluxing of tracer along the wrong slope - this instability will gradually increase in amplitude and eventually make the model unstable.

4 Griffies *et al* implementation

The new isopycnal formulation of Griffies *et al.* (1998) is designed to cure the problems described above. The diffusion matrix implemented is the small-angle approximation to the full tensor [4]:

$$\mathbf{K}^{grif} = A_I \begin{bmatrix} 1 & 0 & S_x \\ 0 & 1 & S_y \\ S_x & S_y & \epsilon + S^2 \end{bmatrix} \tag{14}$$

The discretised x-component of the small-angle tensor isopycnal diffusive flux is now (see Fig. 1 and below for definitions):

$$-F^x_{i,k}(T) = \frac{1}{4dz_k} \sum_{kr=0}^{1} dzz_{k-1+kr} \sum_{ip=0}^{1} A^{(i+ip,k)}_{(i,k|i+ip,k-1+kr)} \\ (\delta_x T_{i,k} + Sx^{(i+ip,k)}_{(i,k|i+ip,k-1+kr)} \delta_z T_{i+ip,k-1+kr}) \tag{15}$$

Expanding this for clarity,

$$\begin{aligned} -F^x_{i,k}(T) = \frac{1}{4dz_k} \quad & dzz_{k-1} \left[A^{i,k}_{(i,k|i,k-1)} (\delta_x T_{i,k} + Sx^{i,k}_{(i,k|i,k-1)} \delta_z T_{i,k-1}) \right] \\ + \quad & dzz_k \left[A^{i,k}_{(i,k|i,k)} (\delta_x T_{i,k} + Sx^{i,k}_{(i,k|i,k)} \delta_z T_{i,k}) \right] \\ + \quad & dzz_{k-1} \left[A^{i+1,k}_{(i,k|i+1,k-1)} (\delta_x T_{i,k} + Sx^{i+1,k}_{(i,k|i+1,k-1)} \delta_z T_{i+1,k-1}) \right] \\ + \quad & dzz_k \left[A^{i+1,k}_{(i,k|i+1,k)} (\delta_x T_{i,k} + Sx^{i+1,k}_{(i,k|i+1,k)} \delta_z T_{i+1,k}) \right] \end{aligned} \tag{16}$$

where $\delta_x T_{i,k} = (T_{i+1,k} - T_{i,k})/dxu_i$ and $\delta_z T_{i,k} = (T_{i,k} - T_{i,k+1})/dzz_k$, and Sx and A are defined below.

The diffusive flux is now the sum of 4 contributions - the essence of the scheme can be revealed by its differences compared to the C87 formulation:

1. A different numerical discretization is used, where the flux through the face of each tracer point is calculated as the sum of 4 contributions. The grid is shown in Fig. 1 - to calculate the diffusive flux through the eastern face of point $T_{i,k}$, one must calculate the contributing

[4] Note that the small-angle approximation should make the K_{12} and K_{21} terms zero. In the Redi implementation in the UM, these were left in.

gradients in the boxes 5,6,9,10, as well the relevant isopycnal diffusion coefficients A at these points (which are each scaled as described in section 5).

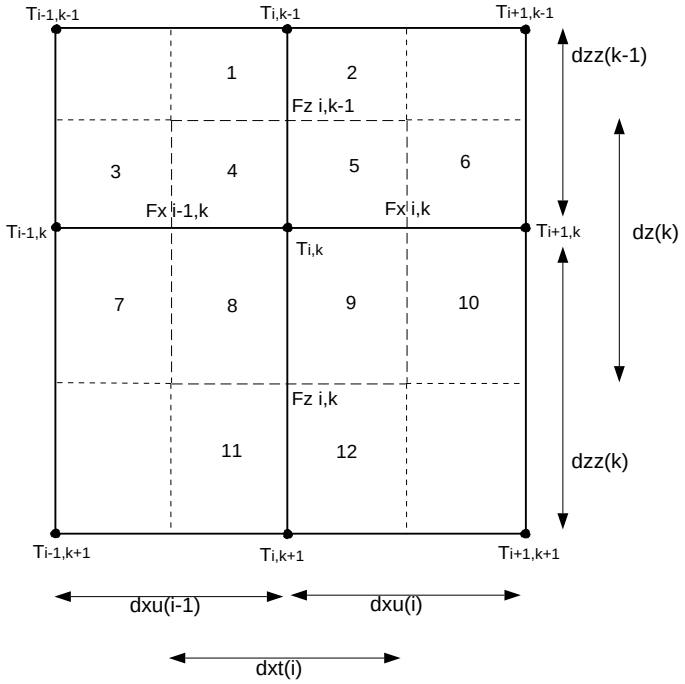

Figure 1: Illustration of finite difference grid (in the x-z plane) used in the Griffies *et al* isopycnal diffusion formulation.

2. The off-diagonal terms from the diffusion matrix are calculated using expansion coefficients so that the fluxes are properly aligned along the density surfaces. Hence each term S_x contributing to the flux through the eastern face of point $\mathrm{T}_{i,k}$ is evaluated as

$$Sx^{(i1,k1)}_{(i,k|i3,k3)} = -\frac{\delta_x \rho^{(i1,k1)}_{(i,k)}}{\delta_z \rho^{(i1,k1)}_{(i3,k3)}} \tag{17}$$

where $(i3, k3)$ references the 4 boxes involved and $(i1, k1)$ refers to the point where the four contributing boxes meet - the latter means that when each density gradient is calculated, it has a common reference level and hence will cancel exactly if required:

$$\delta_m \rho^{(i1,k1)}_{i,k} = (\rho_\theta)_{i1,k1} \delta_m \theta_{i,k} + (\rho_s)_{i1,k1} \delta_m s_{i,k} \tag{18}$$

where ρ_θ and ρ_s are the expansion coefficients for temperature and salinity respectively, and m is the derivative direction (i.e. x, y or z).

The vertical flux components follow in a similar way. In particular, the "zz" component of the diffusive flux, which is usually solved implicitly for stability reasons, can still be dealt with in the same way as in the C87 scheme.

In summary, the Griffies scheme takes the average of 4 fluxes to calculate each diffusive flux, using separately scaled diffusion coefficients and expanding density derivatives into temperature and salinity derivatives - in such a way, the computational mode and the misalignment of fluxes along density surfaces are both removed.

5 UMUI selection and relevant logicals

There are 5 logicals in the UM code at vn4.5 which control which isopycnal diffusion scheme is to be used and which aspects are enabled.

- L_OISOPYC - Overall control of any isopycnal diffusion - **true** is on and **false** is off.
- L_OISOMOM - If **true**, the new Griffies *et al.* (1998) scheme is selected; if **false** then the old Redi (1982) scheme is selected.
- L_SLOPEMAX - Applicable to Redi scheme only - if **true**, then should the isopycnal slope exceed its maximum allowable value, the slope is reset to its maximum value but the diffusive flux is unchanged - this causes a diapycnal component to the diffusive flux; if **false**, then the scheme of Gerdes *et al.* (1991) is used - this scales down the isopycnal diffusivity when the slope exceeds its maximum value, such that

$$A_I \longrightarrow A_I \left(\frac{slmx}{|S|} \right)^2 \tag{19}$$

  where $slmx$ is the maximum allowable slope and $|S|$ the magnitude of the isopycnal slope.
- L_OEXTRAP - Applicable to Redi scheme only - this controls the calculation of density gradients at the top and bottom boundaries of the model. If **true**, extrapolation is used at these boundaries to calculate vertical density gradients. (Note that this is not possible with the L_SLOPEMAX option, which calls a different subroutine (see subroutine TRACER)).
- L_OISOTAPER - Option to taper the isopycnal diffusivity with a *tanh* function, so that as the slope approaches its maximum allowable value, the isopycnal diffusivity tends to zero. This is essential for the Griffies scheme and advisable for the Redi scheme. The isopycnal slope is scaled such that:

$$A_I \longrightarrow 0.5 A_I \left(1 - tanh \left(\frac{|S| - \delta^{dm}}{S^{dm}} \right) \right) \tag{20}$$

  where S^{dm} is the half-width over which the transition from a diffusion coefficient near A_I to a near-zero diffusion coefficient occurs, and δ^{dm} is the slope at which the scaling is

0.5. The values are currently set to $S^{dm} = 0.001$ and $\delta^{dm} = 0.004$ - these are hard-wired in the UM code.

6 Structure of isopycnal diffusion code

Different sets of subroutines are used by the Redi and Griffies schemes - the main parts of these codes are called from within TRACER, with some initialisation of variables through BLOKINIT. The codes are as follows:

6.1 Redi/C87 code

The subroutines involved in the Redi scheme are:

- IPDCOFCO or IPDCOFCL: these routines are different versions of the same basic code. They calculate the isopycnal diffusion matrix components of (7); the former is used if one is using the old slopemax formulation (where the isopycnal slope is reset back to the maximum slope should this be exceeded), while the latter is more recent and can scale the diffusivity if the slope becomes large. Called from TRACER.
- IPDFLXCL: this calculates the actual isopycnal fluxes and updates the tracers. Called from TRACER.
- CALCESAV: this initialises the isopycnal flux through the south face of the first row in a block of rows. Called from BLOKINIT.

6.2 Griffies code

The subroutines involved in the Griffies code are:

- ISOPYC_M: calls routines which calculate isopycnal coefficients. Called from TRACER.
- ELEMENTS: Calculates expansion coefficients (for temperature and salinity) and temperature and salinity gradients. Called from ISOPYC_M.
- AI_CALC: Calculates the isopycnal diffusion coefficients to be used in ISOFLUX, scaling each as described by L_OISOTAPER above. Called from ISOPYC_M.
- ISOFLUX: Calculates the isopycnal diffusive fluxes and updates the tracer values. Called from TRACER.

- DRODT: Calculates the thermal expansion coefficient.[5]
- DRODS: Calculates the haline (salt) expansion coefficient.

7 Calculation of isopycnal diffusion coefficients

The values for the isopycnal diffusion coefficient A_I can be set through the UMUI to be either constant or have depth variation. The formula is

$$A_I(z) = AHI2 + (AHI1 - AHI2)\, exp(-z/AHI3)$$

where z is the depth. This makes the surface value equal to $AHI1$ and the bottom value $AHI2$, with a scale-depth of AHI3. Setting $AHI1 = AHI2$ clearly gives a constant diffusion coefficient with depth.

[5]NB: the sign convention for calculating the expansion coefficients in the UM code is the opposite of what might be expected - care is needed if these calculations are used elsewhere.

REFERENCES

Cox, M. D., 1984: A primitive equation, three dimensional model of the ocean. GFDL, Princeton.

——, 1987: Isopycnal diffusion in a z-coordinate model. *Ocean Modelling*, **74**, 1–5 (unpublished manuscript).

Crawley, S., and T. C. Johns, 1987: Implementation of an isopycnal diffusion scheme in the global ocean model. Met O 20 Internal note No. 76, UK Meteorological Office.

Gent, P. R., and J. C. McWilliams, 1990: Isopycnal mixing in ocean circulation models. *J. Phys. Oceanogr.*, **20**, 150–155.

Gerdes, R., C. Köberle, and J. Willebrand, 1991: The influence of numerical advection schemes on the results of ocean general circulation models. *Climate Dynamics*, **5**, 211–226.

Griffies, S. M., A. Gnanadesikan, R. C. Pacanowski, V. D. Larichev, J. K. Dukowicz, and R. D. Smith, 1998: Isoneutral diffusion in a z-coordinate ocean model. *J. Phys. Oceanogr.*, **28**(5), 805–830.

Large, W. G., J. C. McWilliams, and S. C. Doney, 1994: Ocean vertical mixing: A review and a nonlocal boundary layer parameterization. *rev. Geophys.*, **32**, 363–403.

Pacanowski, R. C., and S. G. Philander, 1981: Parametrization of vertical mixing in numerical models of tropical oceans. *J. Phys. Oceanogr.*, **11**, 1443–1451.

Redi, M. H., 1982: Oceanic isopycnal mixing by coordinate rotation. *J. Phys. Oceanogr.*, **12**, 1154–1158.

Rickard, G. J., 1999: Ocean models and the implementation of vertical diffusion and vertical mixing. Ocean UMDP 59, UK Met. Office.

Veronis, G., 1975: The role of models in tracer studies. *Numerical models of the ocean circulation.* National Academy of Sciences, Washington, D. C., 133–146.