The Computation of Diapycnal Diffusive and Adveotive Scalar Fluxes in Multilayer Isopycnic-Coordinate Ocean Models*

DINGMING HU

Joint Institute for the Study of the Atmosphere and Ocean, University of Washington, Seattle, Washington

(Manuscript received 17 March 1995, in final form 31 January 1996)

ABSTRACT

This paper addresses the problem of how to numerically incorporate diapycnal mixing and advection processes into isopycnal ocean general circulation models. A general expression of diapycnal velocity is derived for the case in which density is a function of both potential temperature and salinity. The expression is independent of turbulence closure parameterizations and thus can be viewed as a generalized definition of diapycnal velocity. With this definition, it is simple to derive expressions of diapycnal velocity for different parameterizations of turbulent mixing in isopycnal ocean models. Numerical algorithms are developed to overcome difficulties associated with massless layers in computing diapycnal diffusive and advective scalar fluxes in isopycnic-coordinate ocean models. In the diapycnal diffusive flux computation, a mass-weighted averaging is used to prevent linear instability in the time integration. In diapycnal advective flux computation, it is shown that the diapycnal mass exchange associated with density coordinate restoration should be consistent with that caused by diapycnal velocity. This consistency is achieved in the developed algorithms. The algorithms are tested and verified in one-dimensional Dirichlet and Neumann boundary value problems. Furthermore, through a simulation of a realistic ocean profile diffusion process, the author shows that the algorithms not only have the ability to simulate vertical mixing processes in the real ocean but also have computational efficiency good enough for application to three-dimensional isopycnal ocean models.

1. Introduction

In the early stages of development of multilayer isopycnic-coordinate ocean general circulation models (OGCMs), density was either regarded as an independent thermodynamic variable or as a function of potential temperature only, and flow in the interior ocean was usually assumed to be adiabatic. For this kind of idealized ocean model, one of the major difficulties in the numerical implementation is isopycnal outcropping. This difficulty may be overcome by using positive-definite transport schemes like the flux-corrected transport (FCT) algorithm (Zalesak 1979). By skillfully blending low-order and high-order advective flux estimates ("order" here is the order of the numerical truncation error), the FCT algorithm prevents the occurrence of negative layer thickness values associated with higher-order computational overshooting but retains steep layer thickness gradients such as those found in the outcropping zone. Development of the FCT algorithm cleared the way for the development of the wind-driven OGCMs in isopycnic coordinates (Bleck and Boudra 1986; Bleck and Smith 1990; Smith et al. 1990). However, much work remains to be done in expanding the wind-driven isopycnal ocean models into fullfledged OGCMs. An important component that is missing in the wind-driven isopycnal models is diapycnal mixing and advection. However, how diapycnal mixing and advection should be incorporated into isopycnic coordinate OGCMs remains unclear in both theoretical formulation and numerical implementation.

One problem facing us arises from the constraint that diapycnal diffusive and advective density fluxes must be mass conservative and in the mean time maintain the density coordinates. This difficulty was first pointed out by Huang and Bryan (1987, HB hereafter). In HB's multilayer isopycnal model, diapycnal turbulent mixing flux of density is parameterized by vertical diffusivity and diapycnal advective velocity is calculated based on its definition. In order to conserve buoyancy globally and maintain the density coordinates in the isopycnal layers underlying the surface mixed layer, HB modified the turbulent density flux across the bottom of the mixed layer to transfer the excess buoyancy in the isopycnal layers into the surface mixed layer. In HB, density is an independent thermodynamic variable and thus no equation of state is needed. When density is a function of potential temperature and salinity, the problem becomes more com-

* JISAO Contribution 318.

Corresponding author address: Dr. Dingming Hu, JISAO, University of Washington, Box 354235, Seattle, WA 98195.

© 1996 American Meteorological Society
licated. In the model developed by Bleck et al. (1992), vertical mixing is only considered in the surface mixed layer, and interlayer mass exchanges in the isopycnic interior are only caused by the cabbeling effect of isopycnic mixing of temperature and salinity. In the model developed by Oberhuber (1993), interior interlayer mass exchange rate is calculated based on a parameterization analogous to that of the surface mixed layer and on the constraint of density coordinate maintenance. However, consistency of such determined interlayer mass exchange rate with the definition of diapycnal velocity was not addressed. Based on eddy diffusion coefficient parameterization of interior mixing, McDougall (1987a) derived an expression for a locally defined diapycnal velocity (dianeutral velocity) in which density is a function of potential temperature, salinity, and pressure. Since different parameterization of interior diapycnal mixing is used in Oberhuber’s model, McDougall’s expression cannot apply to this model. In this case, what we need is an expression of diapycnal velocity that is independent of turbulent closure parameterization. Another problem in incorporating diapycnal mixing and advection into isopycnic OGCMs is associated with the vanishing layers, or the so-called massless layers, existing in the models. This problem is handled differently in different models. Usually a criterion is set to judge if the layer is massless or not so that different schemes can apply. This raises a question “are the diapycnal diffusive and advective fluxes mathematically continuous?”

Since diapycnal mixing and advection are very important processes that determine the thermohaline circulation and the vertical stratification in OGCMs, clearly addressing the above mentioned problems is a key step for the development of a fullfledged isopycnic OGCM. This is our attempt in the present study. The layout of the present paper is as follows. Section 2 discusses the definition of diapycnal velocity and its expressions used in OGCMs. Section 3 discusses the algorithm for the objective diagnosis of diapycnal diffusive scalar fluxes in isopycnic coordinates. Section 4 discusses the relationship between diapycnal velocity and density coordinate restoration in multilayer isopycnic models and derives the algorithms that objectively diagnose diapycnal velocity and maintain the isopycnic coordinate. The algorithms are tested and verified in section 5. Section 6 gives a summary and conclusions of this study.

2. Definition of diapycnal velocity

Although the conventional expression for diapycnal velocity, \( e = \frac{\partial t}{\partial \rho} \left( K T \rho \partial \rho / \partial z \right) \), is well known, it is not clear what the expression should be when density is a function of potential temperature and salinity. Further, as mentioned in the introduction, when different parameterizations for diapycnal mixing are used, what are the expressions for the diapycnal velocity? To answer these questions, one has to be based on the definition of diapycnal velocity. In this section we shall give a discussion of the definition of diapycnal velocity and derive its expressions used in OGCMs. As will be shown in section 4, such a discussion will help to understand the relationship between diapycnal velocity and density coordinate restoration in isopycnic ocean models and thus set the stage for the development of numerical algorithms for computing diapycnal velocity in multilayer isopycnic-coordinate OGCMs.

For analogy, we start with the definition of vertical velocity. Mathematically vertical velocity is defined as \( w = \dot{z} \), where the dot represents the substantial derivative. Thus, vertical velocity is the rate at which a fluid parcel crosses \( z \)-coordinate surfaces. Similarly, diapycnal velocity can be defined as the rate at which a fluid parcel crosses \( \rho \)-coordinate surfaces (i.e., isopycnals). In this sense, \( \dot{\rho} \) is the diapycnal "velocity." However, \( \dot{\rho} \) does not have the dimensions of velocity. The relation of \( \dot{\rho} \) to vertical velocity can be obtained by expressing \( w \) in \( x, y, \rho \) coordinates, that is,

\[
\begin{align*}
w = \dot{z} &= \left( \frac{\partial z}{\partial t} \right)_\rho + u \left( \frac{\partial z}{\partial x} \right)_\rho + v \left( \frac{\partial z}{\partial y} \right)_\rho + \dot{\rho} \left( \frac{\partial z}{\partial \rho} \right)_\rho.
\end{align*}
\]

It is clearly seen in (1) that in \( \rho \) coordinates the vertical velocity of a fluid parcel is caused by 1) vertical displacement of the isopycnic surface in which the parcel lies, 2) horizontal advection of the parcel over the isopycnic topography, and 3) vertical displacement of the parcel relative to its original isopycnic surface. Due to its clear indication of cross-isopycnic displacement and due to its velocity dimension, the fourth term in (1) is in practice defined as diapycnal velocity, that is

\[
\begin{align*}
e &= \dot{\rho} \left( \frac{\partial z}{\partial \rho} \right)_\rho.
\end{align*}
\]

The physical meaning of definition (2) is illustrated in Fig. 1. In Fig. 1, the solid curve shows the initial vertical profile of density in a water column, and the dashed curve shows the vertical profile after a time interval \( \Delta t \). We assume the vertical profile variation is due to diapycnal mixing only. As can be seen in Fig. 1, the vertical coordinate of parcel \( A \) remains \( z_A \) but its density has decreased to \( \rho_A' \) after time interval \( \Delta t \). In the mean time, isopycnic \( \rho = \rho_A \), where parcel \( A \) was initially located, has displaced downward to \( z_A' \). In other words, parcel \( A \) has moved upward a vertical distance of \( \Delta z = (z_A - z_A') \) relative to its original isopycnic during time interval \( \Delta t \). The relationship of the parcel’s relative displacement to its original isopycnic and its density change \( \Delta \rho = (\rho_A' - \rho_A) \) due to diapycnal mixing can be expressed as \( \Delta z = \Delta \rho \tan \alpha \).

The vertical velocity at which parcel \( A \) moves away from its original isopycnic due to diapycnal mixing then is
and salinity only. In simple isopycnal ocean models, potential density itself is often used as an independent thermodynamic variable. For convenience we shall interchangeably use density and potential density. When used as vertical coordinate, therefore, \( \rho \) can be treated as an independent thermodynamic variable, or as a function of potential temperature and salinity. For the later case \( \rho \) can be any one of the potential densities used in physical oceanography practice, such as \( \sigma_0 \), \( \sigma_2 \), and \( \sigma_z \).

In the case that \( \rho \) is an independent thermodynamic variable, density mixing along isopycnal surfaces vanishes, and small-scale turbulent mixing is the only interior diabatic process that causes a nonzero \( \dot{\rho} \). If small-scale turbulent mixing is parameterized by vertical eddy diffusivity \( K_d \), then

\[
\dot{\rho} = \frac{\partial}{\partial z} \left( K_d \frac{\partial \rho}{\partial z} \right),
\]

and diapycnal velocity is given as

\[
e = \dot{\rho} \frac{\partial \rho}{\partial \rho} = \frac{\partial}{\partial \rho} \left( \frac{\partial \rho}{\partial \rho} \right) = \frac{\partial \rho N^2}{g}.
\]

Here one obtains the conventional expression for diapycnal velocity.

In the case that \( \rho \) is a function of potential temperature \( \theta \) and salinity \( S \), from the equation of state, \( \rho = \rho(\theta, S) \), we have

\[
\dot{\rho} = \rho_0 (-\alpha \dot{\theta} + \beta \dot{S}),
\]

\[
\frac{\partial \rho}{\partial z} = \rho_0 \left( \left( -\alpha \frac{\partial \theta}{\partial z} + \beta \frac{\partial S}{\partial z} \right) - \frac{\rho_0 N^2}{g} \right),
\]

where \( \alpha = -\rho_0 \frac{\partial \theta}{\partial \rho} \) and \( \beta = \rho_0 \frac{\partial S}{\partial \rho} \) are, respectively, thermal expansion and saline contraction coefficients, and \( N \) is the buoyancy frequency. The diapycnal velocity definition (1) can then be rewritten as

\[
e = \dot{\rho} \frac{\partial z}{\partial \rho} = g \left( \alpha \theta - \beta S \right).
\]

Note that (5) is a general expression of diapycnal velocity in the case that density is a function of \( \theta \) and \( S \). Since (5) is independent of turbulence closure parameterization, it provides a theoretical basis for deriving expressions of diapycnal velocity for different parameterizations of interior turbulent mixing in isopycnal OGCMs. In this sense, (5) can be viewed as a generalized definition of diapycnal velocity. According to (5), diapycnal velocity is caused by temperature and salinity transformations of fluid parcels, that is, \( \dot{\theta} \) and \( \dot{S} \), which, in turn, are caused by turbulent mixing processes in the interior ocean. In modeling practice, therefore, a turbulence closure parameterization must be used. Certainly, different parameterizations will result in different expressions of diapycnal velocity, but the
corresponding expressions must be consistent with (5). If isopycnal mixing due to mesoscale eddies and dia-
pycnal mixing associated with small-scale turbulent mixing are parameterized by eddy diffusion coeffi-
cients as in most OGCMs, following McDougall (1987a)\(^1\), \(\dot{\theta}\) and \(\dot{S}\) can be expressed as

\[
\dot{\theta} = \nabla_{\rho} \cdot (K_i \nabla_{\rho} \theta) + \frac{\partial}{\partial z} \left( K_d \frac{\partial \theta}{\partial z} \right) + \frac{\partial F^\theta}{\partial z}, \quad (6)
\]

\[
\dot{S} = \nabla_{\rho} \cdot (K_i \nabla_{\rho} S) + \frac{\partial}{\partial z} \left( K_d \frac{\partial S}{\partial z} \right) + \frac{\partial F^S}{\partial z}. \quad (7)
\]

Here \(K_i\) and \(K_d\) are, respectively, isopycnal and diapycnal eddy diffusivities, and \(F^\theta\) and \(F^S\) are, respectively, heat and salt fluxes caused by double diffusion. Substituting (6) and (7) into (5) gives

\[
e = \rho \frac{\partial z}{\partial p} = \frac{g}{N^2} (\alpha \dot{\theta} - \beta \dot{S})
\]

\[
= \frac{g}{N^2} \left\{ \alpha \left[ \nabla_{\rho} \cdot (K_i \nabla_{\rho} \theta) + \frac{\partial}{\partial z} \left( K_d \frac{\partial \theta}{\partial z} \right) + \frac{\partial F^\theta}{\partial z} \right] \right. \\
- \beta \left[ \nabla_{\rho} \cdot (K_i \nabla_{\rho} S) + \frac{\partial}{\partial z} \left( K_d \frac{\partial S}{\partial z} \right) + \frac{\partial F^S}{\partial z} \right] \right\}. \quad (8)
\]

Equation (8) can be rewritten in a form the same as the expression for diapycnal velocity derived by Mc-
dougall (1987a). The only difference between (8) and McDougall's diapycnal velocity expression is that \(\alpha\) and \(\beta\) are functions of \(\theta\) and \(S\) only for diapycnal velocity but are functions of \(\theta, S,\) and pressure \(p\) for dia-
pycnal velocity. It should be pointed out that diapycnal velocity does not include all the effects of in-
terior mixing. Decomposing the substantial derivatives on the left-hand sides of (6) and (7) in isopycnic co-
ordinates, we have

\[
\left( \frac{\partial \theta}{\partial t} \right)_\rho + \mathbf{v} \cdot \nabla_{\rho} \theta + e \frac{\partial \theta}{\partial z} = \nabla_{\rho} \cdot (K_i \nabla_{\rho} \theta)
\]

\[
+ \frac{\partial}{\partial z} \left( K_d \frac{\partial \theta}{\partial z} \right) + \frac{\partial F^\theta}{\partial z},
\]

\[
\left( \frac{\partial S}{\partial t} \right)_\rho + \mathbf{v} \cdot \nabla_{\rho} S + e \frac{\partial S}{\partial z} = \nabla_{\rho} \cdot (K_i \nabla_{\rho} S)
\]

\[
+ \frac{\partial}{\partial z} \left( K_d \frac{\partial S}{\partial z} \right) + \frac{\partial F^S}{\partial z}.
\]

Obviously, in addition to the diapycnal advection terms, all mixing terms are explicitly included in the con-
servation equations. Therefore, it is inconsistent to include diapycnal advection without including diapyc-
nal mixing in the conservation equations (also see McDougall 1987a). The point is that diapycnal advection
is only one of the results of interior mixing, and thus neither isopycnal mixing nor diapycnal mixing can be replaced by diapycnal advection.

3. Algorithm for computing diapycnal diffusive scalar fluxes in isopycnic coordinates

In this and next sections, we confine ourselves to the diapycnal mixing and advection processes in the inter-
rior ocean and their computation in the multilayer iso-
pycnic coordinate ocean models. By interior ocean we mean the region underlying the oceanic mixed layer. In
other words, diapycnal mixing and advection processes associated with the mixed-layer entrainment/detrain-
ment as well as free convection are not taken into account.

In OGCMs, diapycnal mixing is usually parameterized by the so-called \(K\) theory, in which the diapycnal diffusive flux of a tracer is set proportional to the grid-
scale diapycnal gradient of the tracer. The proportion-
ality coefficient is referred to as diapycnal diffusivity. In modeling practice, diapycnal diffusivity is either re-
garded as constant or as a function of stratification. Thus, the computation of subgrid-scale diapycnal diffu-
sive scalar fluxes becomes a matter of diagnosing grid-scale diapycnal scalar gradients. As pointed out by
McDougall (1987a), the slope of isopycnal surfaces in the interior ocean is very small, and the vertical gra-
dient of a scalar is a good approximation to the one perpendicu-
lar to the isopycnal surface. Thus diapycnal diffusivity can be used interchangeably with vertical diffusivity. The problem is then further simplified to vertical gradient computation. For convenience, let us focus on discussing the finite-difference representation of the vertical potential temperature gradient \(\partial \theta/\partial z\) in isopycnic coordinates. One can use the same algorithm for other scalars, including the horizontal velocity com-
ponents.

In the following discussion, we associate index \(k\) with coordinate layers, and index \(k + 1/2\) with the in-
terface between layer \(k\) and layer \(k + 1\) (see Fig. 2). Density and temperature are carried at the mass points of
the vertical grid, that is, the central points of iso-
pycnic layers. Using centered finite differences, one can express \(\partial \theta/\partial z\) at depth \(Z_{k+1/2}\) as

\[
\left( \frac{\partial \theta}{\partial z} \right)_{k+1/2} = \frac{\theta_k - \theta_{k+1}}{0.5(\Delta Z_k + \Delta Z_{k+1})}, \quad (9)
\]

where the layer thickness is defined as \(\Delta Z_k = Z_{k+1/2} - Z_{k+1/2}\).

It is straightforward to apply (9) to \(z\) coordinates where layer thickness is fixed and the vertical gradient

\[1\) Note that (6) and (7) are written in the reference frame of a glo-
ally defined potential density surface, while the corresponding equa-
tions in McDougall (1987a) are written in the reference frame of a neutral surface or a locally referenced potential density surface (Mc-
dougall 1987b). In the interior ocean, mixing and flow predomin-
nantly occur along neutral surfaces rather than surfaces of a globally defined potential density. How to achieve best approximation of mixing and flow along neutral surfaces in OGCMs is beyond the scope of the present study.
computed from (9) is always finite. In isopycnic coordinates, however, layer thickness is allowed to vanish or become massless, while temperature/density difference between the vanishing layers remains finite because of the prescribed finite-density coordinate intervals. As a result, an infinite gradient may arise. Although one can get around this problem by setting either a finite minimum value to the denominator or a finite maximum value to the gradient itself, neither option is free of subjective judgment. Huang and Bryan (1987) used a scheme in which whenever an isopycnic layer outcrops, the quantities associated with the outcropping layer are replaced by those of the mixed layer in the vertical gradient computation. However, any criterion for declaring a layer "outcropped" introduces a discontinuity in the vertical gradient calculation as the layer changes from massless to mass-containing, or vice versa.

In trying to develop a method for objectively diagnosing the vertical temperature gradient in \( \rho \)-coordinate models, it is helpful to take a close look at the difference between \( z \) and \( \rho \) coordinates in vertical grid resolution. In any gridpoint model, the vertical temperature field is represented by its values at a set of discrete grid points, and the vertical temperature gradient depends strongly on the vertical grid resolution. Since isopycnic models adjust their vertical resolution to the vertical density gradient, \( \rho \) coordinates generally outperform \( z \) coordinates in objectively representing the vertical temperature field and its gradient for a given number of grid points. In prognostic numerical models, the vertical grid size \( \Delta Z \) must satisfy a linear stability criterion if an explicit scheme is used for time integration of the vertical diffusion equation, that is,

\[
\Delta Z \geq (2K_D \Delta t)^{1/2}, \tag{10}
\]

where \( \Delta t \) is the time step. In \( z \) coordinates, \( z \) is an independent spatial variable, and \( \Delta Z, \Delta t \) can be chosen to satisfy the above criterion. In \( \rho \) coordinates, however, \( z \) is a function of density and \( \Delta Z \) is dependent on the vertical stratification. Thus, while the \( \rho \) grid represents the temperature field more accurately than the \( z \) grid, it may not satisfy the stability criterion in regions of strong stratification where \( \Delta Z \) goes to zero. So \( z \) coordinates outperform \( \rho \) coordinates in numerical stability. Note that on a \( z \) grid, layer depth is finite [not less than \( (2K_D \Delta t)^{1/2} \) and layer temperature is actually the mass-weighted average temperature of all isopycnic layers embedded in the space between two grid points. It is this mass weighting over a finite depth that limits the magnitude of the computed vertical temperature gradient and prevents linear instability in time integrations on the \( z \) grid. Thus, one can get around the numerical difficulty in computing vertical gradients in multilayer isopycnic models by mass-weighted averaging over a finite depth, which filters out variations with length scale less than \( (2K_D \Delta t)^{1/2} \). On the other hand, one certainly does not want to sacrifice the advantage of high spatial resolution of isopycnic models. Therefore, the vertical averaging operation is confined in the model to the vertical gradient computation. The basic idea here is to replace depths and temperatures of the two adjacent isopycnic layers in (9) by those of two adjacent layers of finite depth. In other words, (9) is revised as

\[
\left( \frac{\delta \theta}{\delta z} \right)_{k+1/2} = \frac{\tilde{\theta}^{\text{up}}(k + \frac{1}{2}) - \tilde{\theta}^{\text{low}}(k + \frac{1}{2})}{0.5 \left[ H^{\text{up}}(k + \frac{1}{2}) + H^{\text{low}}(k + \frac{1}{2}) \right]}, \tag{11}
\]

where \( H^{\text{up}}(k + 1/2) \) and \( H^{\text{low}}(k + 1/2) \) are, respectively, the depths of the two finite-depth layers above and below interface \( Z_{k+1/2} \), and \( \theta^{\text{up}}(k + 1/2) \) and \( \theta^{\text{low}}(k + 1/2) \) are the corresponding mass-weighted average temperatures in the two layers. The problem is how to choose the depths of the two finite layers. Obviously, the choice is not unique. However, a good algorithm should be able to produce diffusive scalar fluxes that are physically reasonable and mathematically continuous in time and in space. Based on these considerations, \( H^{\text{up}}(k + 1/2) \) and \( H^{\text{low}}(k + 1/2) \) can be calculated as follows:

\[
H^{\text{up}}(k + \frac{1}{2}) = h^{\text{up}}(k + \frac{1}{2}) + \left\{ 1 - \min \left[ \gamma^{\text{up}}(k + \frac{1}{2}) , \gamma^{\text{low}}(k + \frac{1}{2}) \right] \right\} \Delta Z_{k+1/2}, \tag{12}
\]
\[ H_{\text{low}}^{\text{up}}(k + \frac{1}{2}) = h_{\text{low}}^{\text{up}}(k + \frac{1}{2}) + \left\{1 - \min \left[ \gamma_{1}^{\text{up}}(k + \frac{1}{2}), \gamma_{2}^{\text{up}}(k + \frac{1}{2}) \right] \right\} \Delta Z_{KL(k+\frac{1}{2})}, \]  \tag{13}

where

\[ \gamma_{1}^{\text{up}}(k + \frac{1}{2}) = \frac{h_{\text{up}}^{\text{high}}(k + \frac{1}{2})}{H_0}, \]

\[ \gamma_{2}^{\text{up}}(k + \frac{1}{2}) = \frac{h_{\text{up}}^{\text{high}}(k + \frac{1}{2}) + \Delta Z_{KL(k+1/2)} - H_0}{\Delta Z_{KL(k+1/2)}}. \]  \tag{14}

\[ \gamma_{1}^{\text{low}}(k + \frac{1}{2}) = \frac{h_{\text{low}}^{\text{high}}(k + \frac{1}{2})}{H_0}, \]

\[ \gamma_{2}^{\text{low}}(k + \frac{1}{2}) = \frac{h_{\text{low}}^{\text{high}}(k + \frac{1}{2}) + \Delta Z_{KL(k+1/2)} - H_0}{\Delta Z_{KL(k+1/2)}}. \]  \tag{15}

As illustrated in Fig. 3, \( H_0 \) is a prescribed depth, \( h_{\text{up}}^{\text{high}}(k + \frac{1}{2}) \) is the total depth of those isopycnic layers that are completely contained in the vertical interval \((Z_{k+1/2}, Z_{k+1/2} + H_0)\), and \( KL(k + \frac{1}{2}) \) is the index for the isopycnic layer that contains \( Z_{k+1/2} + H_0 \). One can similarly define \( h_{\text{low}}^{\text{high}}(k + \frac{1}{2}) \) and \( KL(k + \frac{1}{2}) \) from Fig. 3. The mass-weighted average temperatures in \( H_{\text{up}}^{k+1/2} \) and \( H_{\text{low}}^{k+1/2} \) are then respectively given as

\[ \bar{\theta}_{\text{up}}^{\text{high}}(k + \frac{1}{2}) = \sum_{l=k}^{KU(k+1/2)} \alpha_{l}^{\text{up}}(k + \frac{1}{2}) \theta_l, \]

\[ \bar{\theta}_{\text{low}}^{\text{low}}(k + \frac{1}{2}) = \sum_{l=k}^{KL(k+1/2)} \alpha_{l}^{\text{low}}(k + \frac{1}{2}) \theta_l, \]  \tag{16}

where \( \alpha_{l}^{\text{up}}(k + \frac{1}{2}) \) and \( \alpha_{l}^{\text{low}}(k + \frac{1}{2}) \) are respectively the weight coefficients for the layers that are embedded in the finite-depth layers \( H_{\text{up}}^{k+1/2} \) and \( H_{\text{low}}^{k+1/2} \). The weight coefficient is simply the ratio of the embedded layer depth to the total depth of the finite layer.

Our algorithm for computing vertical scalar gradient consists of (11) – (16). Although it appears to be complicated, it can be easily understood through the following discussions.

1) It can be shown that

\[ H_{\text{up}}^{\text{up}}(k + \frac{1}{2}) = H_0 \]

when

\[ \gamma_{1}^{\text{up}}(k + \frac{1}{2}) \geq \gamma_{2}^{\text{up}}(k + \frac{1}{2}), \]

\[ H_{\text{up}}^{\text{up}}(k + \frac{1}{2}) \geq H_0 \]

Thus the minimum value of \( H_{\text{up}}^{\text{up}}(k + \frac{1}{2}) \) is \( H_0 \). The same is true for \( H_{\text{up}}^{\text{low}}(k + \frac{1}{2}) \). Since \( H_0 \) determines the extent to which the temperature field is smoothed in the vertical direction, it must satisfy the linear stability criterion (10). In other words, \( H_0 \) is determined by \( H_0 \geq (2K_p\Delta t)^{1/2} \). Note that (10) must be satisfied by z-coordinate models, too. The counterpart of \( H_0 \) in z-coordinate models is the minimum vertical grid size. Therefore, \( H_0 \) is no more “ad hoc” in isopycnic models than in z-coordinate models.

2) In the case that the depth of each isopycnic layer is finite (\( \geq H_0 \)), the vertical gradient should be simply given by (9). In this case \( h_{\text{up}}^{\text{up}}(k + \frac{1}{2}) \) and \( h_{\text{low}}^{\text{low}}(k + \frac{1}{2}) \) are zero, and \( KL(k + \frac{1}{2}) \) and \( KL(k + \frac{1}{2}) \) are, respectively, \( k + 1 \) and \( k + 1 \). According to (12) – (16), \( H_{\text{up}}^{\text{up}}(k + \frac{1}{2}) \), \( h_{\text{up}}^{\text{low}}(k + \frac{1}{2}) \), \( \theta_{l}^{\text{up}}(k + \frac{1}{2}) \), and \( \theta_{l}^{\text{low}}(k + \frac{1}{2}) \) become \( \Delta Z_k \), \( \Delta Z_{k+1} \), \( \theta_k \), and \( \theta_{k+1} \), respectively, and (11) becomes the standard finite difference expression (9). Therefore, the algorithm is reasonable.

3) A natural question to ask is “do \( H_{\text{up}}^{\text{up}}(k + \frac{1}{2}) \) and \( H_{\text{up}}^{\text{low}}(k + \frac{1}{2}) \) vary continuously?” For simplicity, let us consider \( H_{\text{up}}^{\text{up}}(k + \frac{1}{2}) \). If \( KL(k + \frac{1}{2}) \) does not change, \( H_{\text{up}}^{\text{up}}(k + \frac{1}{2}) \) will vary continuously as long as depth of each layer varies continuously. The only pos-
Fig. 4. A schematic illustration of cases in which \( KU(k + 1/2) \) will change. When interface \( Z_{k+1/2}^* \) is deeper than \( Z_{k+1/2} + H_0 \), \( KU(k + 1/2) \) is \( k^* \). When it moves upward across \( Z_{k+1/2} + H_0 \), \( KU(k + 1/2) \) becomes \( k^* + 1 \). Any time an interface moves upward/downward across \( Z_{k+1/2} + H_0 \), \( KU(k + 1/2) \) will increase/decrease by 1.

Possibly that \( KU(k + 1/2) \) changes is when a layer interface moves across \( Z_{k+1/2} + H_0 \) (see Fig. 4), which may result in discontinuity in \( H^{\text{op}}(k + 1/2) \). The discontinuity is prevented by \( y^{\text{op}}(k + 1/2) \) and \( y^{\text{op}}(k + 1/2) \).

4) If \( \Delta Z_k \) goes to zero, its top interface \( Z_{k-1/2} \) approaches its bottom interface \( Z_{k+1/2} \), and thus \( H^{\text{op}}(k - 1/2), H^{\text{low}}(k - 1/2), \theta^{\text{op}}(k - 1/2), \text{ and } \theta^{\text{low}}(k - 1/2) \) respectively, approach \( H^{\text{op}}(k + 1/2), H^{\text{low}}(k + 1/2), \theta^{\text{op}}(k + 1/2), \text{ and } \theta^{\text{low}}(k + 1/2) \). As a result, we have

\[
\left( \frac{\partial \theta}{\partial z} \right)_{k-1/2} \rightarrow \left( \frac{\partial \theta}{\partial z} \right)_{k+1/2} \quad \text{as } \Delta Z_k \rightarrow 0.
\]

This feature allows us to solve the vertical diffusion equation correctly without resorting to any criterion for “massless” layers in that the vertical diffusive flux convergence within an isopycnic layer vanishes when the layer becomes massless. As a result, temperatures in massless layers are not changed by the vertical diffusion.

4. Algorithms for computing diapycnal velocity in isopycnic coordinates

With the diapycnal diffusive fluxes reasonably diagnosed, we now proceed to derive numerical algorithms for computing diapycnal velocity in multilayer isopycnic models. Again, we confine ourselves to the case in which interior mixing is parameterized by the eddy diffusion coefficients.

We shall start with the case where potential density is an independent thermodynamic variable and then proceed to the more general case where potential density is a function of both potential temperature and salinity. Although analytical expressions for both cases are given by (4) and (8), respectively, we are going to derive the finite-difference schemes directly from the definition \( e = \dot{\rho} \partial z / \partial \rho \). The reason is, as will be shown, that diapycnal velocity cannot be simply given by a finite-difference expression when there exist massless layers.

a. Diapycnal velocity when density is an independent thermodynamic variable

1) Finite-layer-thickness cases

In finite-difference representation, diapycnal velocity is defined at interfaces, that is, the top and bottom boundaries of an isopycnic layer. We therefore need to diagnose \( \dot{\rho} \) at the interfaces. As a starting point, we assume that thicknesses of isopycnic layers are finite. When density \( \rho \) is an independent thermodynamic variable, \( \dot{\rho} \) is analytically given by (3). A simple finite-difference representation of (3) is

\[
\dot{\rho}_{k+1/2} = \frac{\left( K_d \frac{\partial \rho}{\partial z} \right)_{k-1/2} - \left( K_d \frac{\partial \rho}{\partial z} \right)_{k+3/2}}{\Delta Z_k + \Delta Z_{k+1}},
\]

where

\[
\left( K_d \frac{\partial \rho}{\partial z} \right)_{k-1/2} = K_d \frac{\rho_{k-1} - \rho_k}{0.5(\Delta Z_{k-1} + \Delta Z_k)}.
\]

In (17) \( \dot{\rho} \) at interface \( Z_{k+1/2} \) is determined by density diffusive fluxes at interfaces. However, (17) and (18) do not assure correct diagnosis of \( \dot{\rho} \) at the interfaces. For instance, consider a multilayer water column with constant density increments and prescribed densities at the top and bottom boundaries. In this case, a vertical diffusion process in the water column will lead to an

Fig. 5. Equilibrium states of vertical diffusion in a multilayer water column with constant density interval and prescribed densities at the top and bottom boundaries (a) when \( \dot{\rho} \) is computed by Eqs. (17) and (18) and (b) when \( \dot{\rho} \) is computed by Eqs. (19) and (20).
equilibrium state in which each layer has equal thickness (see Fig. 5a). When (17) is used for diagnosing \( \hat{\rho}_{k+1/2} \), however, any state in which layer thickness takes the same value in every other layer (see Fig. 5b) is an equilibrium state. In Fig. 5b, density diffusive fluxes at interfaces are in fact the same in that \( \rho_k - \rho_{k+1} \) and \( \Delta Z_k + \Delta Z_{k+1} \) in (18) are the same for any index \( k \). Thus \( \hat{\rho} \) is zero at all interfaces and so is diapycnal velocity. As a result, the vertical density profile is invariant in time, that is, reaches an equilibrium state. Obviously, there is an infinite number of equilibrium states consistent with (17) and (18). This indicates that (17) and (18) are an unsuitable finite-difference approximation to \( \hat{\rho} \) at the interfaces. As an alternative, \( \hat{\rho}_{k+1/2} \) can be given as

\[
\hat{\rho}_{k+1/2} = \frac{\left( K_d \frac{\delta \rho}{\delta z} \right)_k - \left( K_d \frac{\delta \rho}{\delta z} \right)_{k+1}}{\Delta Z_k + \Delta Z_{k+1}}, \quad (19)
\]

where

\[
\left( K_d \frac{\delta \rho}{\delta z} \right)_k = K_d \frac{\rho_k - \rho_{k+1}}{2 \Delta Z_k}. \quad (20)
\]

In (19) \( \hat{\rho}_{k+1/2} \) is determined by density diffusive fluxes at mass points of the vertical grid. Obviously, density diffusive fluxes calculated from (20) are not the same for any two adjacent mass points in Fig. 5b, and thus \( \hat{\rho}_{k+1/2} \) will not vanish until thicknesses in all isopycnal layers become the same. Therefore, Fig. 5a is the only equilibrium vertical profile of the water column. In other words, (19) and (20) assure the uniqueness of the equilibrium state for the above diffusion process.

With \( \hat{\rho}_{k+1/2} \) determined by (19) and (20), we choose the finite-difference representation of \( \delta z / \delta \rho \) at interface \( Z_{k+1/2} \) as

\[
\left( \delta z / \delta \rho \right)_{k+1/2} = \frac{\Delta Z_k + \Delta Z_{k+1}}{2(\rho_k - \rho_{k+1})}. \quad (21)
\]

It follows from (19) and (21) that

\[
\epsilon_{k+1/2} = \hat{\rho}_{k+1/2} \left( \delta z / \delta \rho \right)_{k+1/2} = \frac{\left( K_d \frac{\delta \rho}{\delta z} \right)_k - \left( K_d \frac{\delta \rho}{\delta z} \right)_{k+1}}{\rho_k - \rho_{k+1}}. \quad (22)
\]

Our next concern is whether the diapycnal velocity given by (22) removes the density coordinate drifts caused by the small-scale turbulent mixing. As discussed in section 2, diapycnal velocity virtually measures the rate an isopycnal moves vertically away from its original position due to interior mixing. On the other hand, density coordinate restoration is essentially a process of converting density coordinate drifts in isopycnal layers to diapycnal mass exchanges. The rate of diapycnal mass exchange across an isopycnal interface determines the rate at which the interface moves away from its initial position. Therefore, isopycnel coordinate restoration is physically consistent with diagnosis of diapycnal velocity. In order to show the consistency of (22) with density coordinate restoration we have to convert density drifts at the interfaces to those within the isopycnal layers.

From (19) one notes that \( \hat{\rho}_{k+1/2} \) is controlled by \( (K_d \frac{\delta \rho}{\delta z})_k - (K_d \frac{\delta \rho}{\delta z})_{k+1/2} \), that is, by convergence of density diffusive fluxes in the layer bounded by isopycnals \( \rho = \rho_k \) and \( \rho = \rho_{k+1} \), which can be expressed as the sum of the diffusive flux convergence in the bottom half of layer \( k \), \( (K_d \frac{\delta \rho}{\delta z})_k - (K_d \frac{\delta \rho}{\delta z})_{k+1/2} \), and that in the top half of layer \( k + 1 \), \( (K_d \frac{\delta \rho}{\delta z})_{k+1/2} - (K_d \frac{\delta \rho}{\delta z})_{k+1} \). Initially, densities in the bottom half of layer \( k \) and the top half of layer \( k + 1 \) are, respectively, \( \rho_k \) and \( \rho_{k+1} \), the reference densities for layers \( k \) and \( k + 1 \) (see the solid curve in Fig. 6a). Due to the diffusive flux convergence, densities in the bottom half of layer \( k \) and the top half of layer \( k + 1 \) will drift away from the reference densities. Let \( \hat{\rho}^{\text{bot}}_k \) and \( \hat{\rho}^{\text{top}}_{k+1} \) be, respectively, the actual densities in the bottom half of layer \( k \) and the top half of layer \( k + 1 \) at the end of time interval \( \Delta t \) (see the dashed curve in Fig. 6a).

From density conservation, one has

\[
\hat{\rho}^{\text{bot}}_k = \frac{\hat{\rho}^{\text{bot}}_k - \rho_k}{\Delta t} = \frac{\left( K_d \frac{\delta \rho}{\delta z} \right)_k - \left( K_d \frac{\delta \rho}{\delta z} \right)_{k+1/2}}{0.5 \Delta Z_k}, \quad (23)
\]

\[
\hat{\rho}^{\text{top}}_{k+1} = \frac{\hat{\rho}^{\text{top}}_{k+1} - \rho_{k+1}}{\Delta t} = \frac{\left( K_d \frac{\delta \rho}{\delta z} \right)_{k+1/2} - \left( K_d \frac{\delta \rho}{\delta z} \right)_{k+1}}{0.5 \Delta Z_{k+1}}. \quad (24)
\]

Our task is to restore \( \hat{\rho}^{\text{bot}}_k \) and \( \hat{\rho}^{\text{top}}_{k+1} \) to their reference densities. In Fig. 6a the actual density in the top half of layer \( \hat{\rho}^{\text{top}}_{k+1} \) is lighter than the reference density \( \rho_{k+1} \). To restore \( \hat{\rho}^{\text{bot}}_k \), we distribute a slab of water of thickness \( \Delta z^{\text{bot}}_{k+1} \) to layer \( k \) and in the meantime transfer the excess buoyancy in the top half of layer \( k + 1 \), \( \Delta Z_{k+1}(\hat{\rho}^{\text{top}}_{k+1} - \rho_{k+1}) \), to water column \( \delta z^{\text{top}}_{k+1} \), so that the density in \( \delta z^{\text{top}}_{k+1} \) matches the reference density of layer \( k \). Therefore, \( \delta z^{\text{top}}_{k+1} \) is given as

\[
\delta z^{\text{top}}_{k+1} = \frac{\Delta Z_{k+1}(\hat{\rho}^{\text{bot}}_k - \rho_k)}{2(\rho_k - \rho_{k+1})} = \Delta Z_{k+1} \hat{\rho}^{\text{bot}}_k \frac{\Delta t}{2(\rho_k - \rho_{k+1})}. \quad (25)
\]

After transferring water column \( \delta z^{\text{top}}_{k+1} \) from layer \( k + 1 \) to layer \( k \), the vertical profile becomes the dashed curve in Fig. 6b. In the bottom half of layer \( k \), \( \hat{\rho}^{\text{bot}}_k \) is lighter than \( \rho_k \). The density coordinate is restored by entraining a slab of water of thickness \( \delta z^{\text{bot}}_k \) from the top half of layer \( k + 1 \). We have

\[
\delta z^{\text{bot}}_k = \frac{\Delta Z_k(\hat{\rho}^{\text{bot}}_k - \rho_k)}{2(\rho_k - \rho_{k+1})} = \frac{\Delta Z_k \hat{\rho}^{\text{bot}}_k \Delta t}{2(\rho_k - \rho_{k+1})}. \quad (26)
\]
The dashed curve in Fig. 6c indicates the final vertical profile. The total mass exchange across interface \( Z_{k+1/2} \) during time period \( \Delta t \) is \( \delta Z_{k+1/2}^{\text{bot}} + \delta Z_{k+1/2}^{\text{top}} \). The diapycnal velocity at interface \( Z_{k+1/2} \) required for density coordinate restoration is thus

\[
e_{k+1/2} = \frac{\delta Z_{k+1/2}^{\text{top}} + \delta Z_{k+1/2}^{\text{bot}}}{\Delta t} = \frac{\rho_k^{\text{bot}} \Delta Z_k + \rho_{k+1}^{\text{top}} \Delta Z_{k+1}}{2(\rho_k - \rho_{k+1})},
\]

(27)

According to (27), diapycnal velocity at interface \( Z_{k+1/2} \) is obtained by converting \( \rho_k^{\text{bot}} \) to vertical displacement with mapping coefficient \( \Delta Z_k/2(\rho_k - \rho_{k+1}) \), and \( \rho_{k+1}^{\text{top}} \) with \( \Delta Z_{k+1}/2(\rho_k - \rho_{k+1}) \). Substituting (23) and (24) into (27) for \( \rho_k^{\text{bot}} \) and \( \rho_{k+1}^{\text{top}} \), one finds that \( e_{k+1/2} \) given by (27) is identical to that given by (22). Thus we have shown that (22) assures restoration of the density coordinates. In other words, the above restoration scheme yields correct diapycnal velocities.

2) Cases with Massless and Quasi-Massless Layers

In the above discussion, however, if \( \Delta Z_{k+1} \) is massless, or less than \( \delta Z_k^{\text{bot}} \), neither (22) nor the scheme for coordinate restoration can apply since layer \( k + 1 \) does not have enough mass to support the upward diapycnal advection at interface \( Z_{k+1/2} \). In this subsection, we hence consider the general case where several massless or quasi-massless layers are sandwiched between mass-containing layers (see Fig. 7a). Again density is assumed to be an independent thermodynamic variable. Instead of resorting to a general finite-difference expression for diapycnal velocity, we prefer to generalize the above algorithm for restoring density coordinates to the general case of vertical stratification.

According to the discussions at the end of section 3, density drift rates in the quasi-massless layers sandwiched between layers \( k \) and \( k + 5 \) in Fig. 7a are negligibly small. On the other hand, diapycnal mass exchanges associated with coordinate restoration in a layer is proportional to the layer thickness [see (25) and (26)]. This is also true in the existence of quasi-massless layers [see (28) derived below]. This means that diapycnal mass exchanges associated with density coordinate restoration in quasi-massless layers are also negligible. Therefore, we only focus on coordinate restoration in the bottom half of layer \( k \) and the top half of layer \( k + 5 \). Restoring density drift in the top half of layer \( k + 5 \) is simple. We distribute a slab of water \( \delta Z_{k+1}^{\text{bot}} \) from layer \( k + 5 \) to layer \( k + 4 \) and transfer the excess buoyancy in the top half of layer \( k + 5 \) to \( \delta Z_{k+1}^{\text{bot}} \) [\( \delta Z_{k+1}^{\text{bot}} \) can be obtained by replacing index \( k \) in (25) by \( k + 4 \)]. As a result, density drift in the top half of layer \( k + 5 \) is restored and layer \( k + 4 \) is inflated (Fig. 7b).

Restoring density drift in the bottom half of layer \( k \) requires entrainment of denser water from layers below, which is not simple to implement due to the existence of quasi-massless layers. One approach is to entrain denser water layer by layer from below until density coordinate in the bottom half of layer \( k \) is restored. The disadvantage is that all of the quasi-massless layers, including the inflated layer \( k + 4 \), may be drained up. Note that the effect of diapycnal diffusion is to diffuse strong density gradients associated with massless or quasi-massless layers. In other words,
quasi-massless layers should be inflated rather than deflated. Thus, the above scheme is not physically reasonable. An alternative is to entrain water directly from the layer that is thick enough to supply denser water to layer \( k \). The problem is that, except for layer \( k + 5 \), layer \( k + 4 \) is probably also such a candidate, and thus subject to drain-up.

To get around this problem, we turn to the finite-depth layer approach used in section 3. Instead of dealing with individual isopycnal layers, we entrain fluid from \( H^{\text{low}}(k + 1/2) \), a finite-depth layer underlying interface \( Z_{k+1/2} \) [see (13)]. The mass-weighted average density in \( H^{\text{low}}(k + 1/2) \), \( \bar{\rho}^{\text{low}}(k + 1/2) \), can be calculated from (16). Thus thickness of the water slab to be entrained to layer \( k \) can be given as (see Fig. 7c)

\[
\delta z_k^{\text{bot}} = \frac{\Delta Z_k (\bar{\rho}_k^{\text{bot}} - \rho_k)}{2 \left[ \rho_k - \bar{\rho}^{\text{low}}(k + 1/2) \right]},
\]  

and the mass transferred from \( H^{\text{low}}(k + 1/2) \) to layer \( k \) is

\[
\bar{\rho}^{\text{low}}(k + 1/2) \delta z_k^{\text{bot}} = \sum_{j=k+1}^{K/(k+1/2)} \alpha_j^{\text{low}}(k + 1/2) \rho_j \delta z_k^{\text{bot}}.
\]  

According to (29), the amount of water taken from each layer embedded in \( H^{\text{low}}(k + 1/2) \) is \( \alpha_j^{\text{low}}(k + 1/2) \delta z_k^{\text{bot}} \). Since the weighting coefficient \( \alpha_j^{\text{low}}(k + 1/2) \) is proportional to the thickness of each embedded layer, the main contribution to \( \delta z_k^{\text{bot}} \) is from the thickest layer in \( H^{\text{low}}(k + 1/2) \), that is, layer \( k + 5 \), and the mass loss in quasi-massless layers is negligibly small. Especially, there is a net mass gain of \( \delta z_k^{\text{top}} - \alpha_j^{\text{low}}(k + 1/2) \delta z_k^{\text{bot}} \) in layer \( k + 4 \) after the density coordinate restoration. Note that the net mass gain is of the same order as \( \delta z_k^{\text{top}} \). Therefore, layer \( k + 4 \) is

Fig. 7. Isopycnic coordinate restoration in the case in which several massless and quasi-massless layers are sandwiched between two thick layers. Solid curves: initial density profile. Dashed curves: actual density profiles due to vertical diffusion and coordinate restoration.
inflated, basically due to the density coordinate restoration in the adjacent layer \( k + 5 \). The final density profile is shown by the dashed curve in Fig. 7d. Since the total mass exchange across interface \( Z_{k+4+1/2} \) is equal to \( \delta z_{k}^{\text{bot}} + \delta z_{k}^{\text{top}} \), diapycnal velocity at the interface is given as

\[
e_{k+4+1/2} = \frac{\delta z_{k}^{\text{bot}} + \delta z_{k}^{\text{top}}}{\Delta t} - \frac{\Delta Z_{k}(\hat{\rho}_{k}^{\text{bot}} - \rho_{k})}{2\Delta t(\rho_{k} - \rho_{k}^{\text{low}}(k + 1/2))} + \frac{\Delta Z_{k+5}(\hat{\rho}_{k+5}^{\text{top}} - \rho_{k+5})}{2\Delta t(\rho_{k+4} - \rho_{k+5})} \]
\[
= \hat{\rho}_{k}^{\text{bot}} \frac{\Delta Z_{k}}{2(\rho_{k} - \rho_{k}^{\text{low}}(k + 1/2))} + \hat{\rho}_{k+5}^{\text{top}} \frac{\Delta Z_{k+5}}{2(\rho_{k+4} - \rho_{k+5})} . \tag{30}
\]

Comparing (30) to (27), one notes that the difference made by massless and quasi-massless layers is mainly the change in the mapping coefficient associated with entrainment.

Before going on to discuss the case where density is a function of \( \theta \) and \( S \), it is helpful to summarize the algorithm as follows.

1) Theoretically, one can either derive finite-difference expressions for diapycnal velocity that assure density coordinate restoration or develop algorithms for density coordinate restoration which produce the correct diapycnal velocity. For the former approach, the task is to correctly diagnose \( \hat{\rho} \partial \theta / \partial \rho \) at interfaces. To this end, \( \hat{\rho} \) at the interfaces must be computed from density diffusive fluxes at the vertical mass points rather than interfaces. Consistent with this, it is important in the approach of coordinate restoration to diagnose density drift rates for the top and bottom halves of each isopycnal layer and convert both to diapycnal mass exchanges. We have shown that the two approaches produce the same diapycnal velocity and restore density coordinates in the case where the depth of each isopycnal layer is finite (thick enough to supply mass needed for the associated diapycnal mass exchange). In the general case where massless and quasi-massless layers exist, the approach based on coordinate restoration is preferable because any correct expression for diapycnal velocity involves mass exchange among several layers.

2) The basic effect of vertical diffusion is to diffuse density discontinuities or strong density gradients associated with massless or quasi-massless layers. Diapycnal mass exchange associated with density coordinate restoration must be consistent with this effect. To this end, the following principle is applied. If entrainment is required, always detrain the water column to the adjacent isopycnal layer, which assures thin-layer inflation. If entrainment is needed, always entrain water from an adjacent layer of finite depth, as defined in section 1, so that deflation mainly occurs in the thickest isopycnal layer in the finite-depth layer.

b. Diapycnal velocity when density is a function of temperature and salinity

We now consider the case in which density is a function of potential temperature and salinity. Although diagnosis of diapycnal velocity and coordinate restoration become more complicated in this case, the idea is the same: density coordinate drifts are being diagnosed in the top and bottom halves of each layer and are then converted into diapycnal mass exchanges. In three-dimensional isopycnal ocean models, density coordinate drift is caused by isopycnal and diapycnal mixing. In addition, numerical diffusion during heat and salt advection also causes density coordinate drift. In order to diagnose the density coordinate drifts, we first compute \( \theta_{k}^{*} \) and \( S_{k}^{*} \), the bulk potential temperature and salinity in layer \( k \) after isopycnal advection and diffusion. Then we add the diapycnal temperature and salinity diffusive fluxes and compute potential temperature and salinity in the top and bottom half of layer \( k \), which are determined by the following equations:

\[
\frac{\overline{\theta}_{k}^{\text{top}} - \theta_{k}^{*}}{\Delta t} = \frac{1}{0.5\Delta Z_{k}} \left( \frac{\partial \theta}{\partial z} \right)_{k+1/2} - \left( K_{d} \frac{\partial \theta}{\partial z} \right)_{k} ,
\]

\[
\frac{\overline{S}_{k}^{\text{top}} - S_{k}^{*}}{\Delta t} = \frac{1}{0.5\Delta Z_{k}} \left( K_{d} \frac{\partial S}{\partial z} \right)_{k+1/2} - \left( K_{d} \frac{\partial S}{\partial z} \right)_{k} , \tag{31}
\]

\[
\frac{\overline{\theta}_{k}^{\text{bot}} - \theta_{k}^{*}}{\Delta t} = \frac{1}{0.5\Delta Z_{k}} \left( K_{d} \frac{\partial \theta}{\partial z} \right)_{k-1/2} - \left( K_{d} \frac{\partial \theta}{\partial z} \right)_{k} ,
\]

\[
\frac{\overline{S}_{k}^{\text{bot}} - S_{k}^{*}}{\Delta t} = \frac{1}{0.5\Delta Z_{k}} \left( K_{d} \frac{\partial S}{\partial z} \right)_{k-1/2} - \left( K_{d} \frac{\partial S}{\partial z} \right)_{k} . \tag{32}
\]

Diffusive fluxes at interfaces and mass points in (31) and (32) are computed by the algorithm expressed by (11)—(16). Solving (31) and (32) for \( \overline{\theta}_{k}^{\text{top}}, \overline{\theta}_{k}^{\text{bot}}, \overline{S}_{k}^{\text{bot}}, \overline{S}_{k}^{\text{top}} \), and \( S_{k}^{*} \) and using the equation of state gives

\[
\overline{\rho}_{k}^{\text{top}} = \rho \overline{\theta}_{k}^{\text{top}}, \overline{S}_{k}^{\text{top}} , \overline{\rho}_{k}^{\text{bot}} = \rho \overline{\theta}_{k}^{\text{bot}}, \overline{S}_{k}^{\text{bot}} .
\]

Thus, we obtain density drifts in the top and bottom halves of layer \( k \), \( \overline{\rho}_{k}^{\text{top}} - \rho_{k} \), and \( \overline{\rho}_{k}^{\text{bot}} - \rho_{k} \).

The basic steps for restoring isopycnic coordinates are the same as discussed in the previous subsection; the difference now is that one has to deal with heat and salt transfer as well as nonlinearity of the equation of state. The steps are given as follows.

1) Based on the density drift in the top (bottom) half of layer \( k \), determine the amount of fluid that is needed
to exit/enter the top (bottom) half layer through the top (bottom) interface in order to restore the density coordinate there.

2) When detrainment is needed, the detrained water column $\delta z_k^{\text{top}}(\delta z_k^{\text{m}})$ always goes to the adjacent single layer, and $\theta$, $S$, and $\rho$ in $\delta z_k^{\text{top}}(\delta z_k^{\text{m}})$ should match those of the receiving layer. The matching requires heat and salt transfer from top (bottom) half of layer $k$ to $\delta z_k^{\text{top}}(\delta z_k^{\text{m}})$. If the equation of state is linear, the heat and salt transfer completely restores the density coordinate in the top (bottom) half of the layer. If the equation of state is nonlinear, there will be some drift remaining. If the remaining drift is not negligibly small, we then calculate the new $\delta z_k^{\text{top}}(\delta z_k^{\text{m}})$ associated with the remaining drift for further density coordinate restoration. In practice, however, the further restoration is not needed in that new $\delta z_k^{\text{top}}(\delta z_k^{\text{m}})$ is virtually several orders of magnitude smaller than the initial one and thus is negligible. The reason is that density drift caused by the interior mixing is several orders of magnitude smaller than the density-coordinate interval between layers, and thus the original $\delta z_k^{\text{top}}(\delta z_k^{\text{m}})$ is much smaller than the layer thickness [see (25) and (28)]. Because of the large volume difference between $\delta z_k^{\text{top}}(\delta z_k^{\text{m}})$ and layer $k$, the nonlinear effect becomes negligible.

3) When entrainment is needed, always entrain the fluid from a layer of finite depth. The same arguments for negligible nonlinear effect also hold true in this case.

4) Restoring density coordinates in the top and bottom half of each layer may result in different temperatures and salinities in the two halves of the layer. To avoid this, we restore density coordinate in the top (bottom) half of each layer first and then get the bulk temperature and salinity for the entire layer. The bulk temperature and salinity are then used to diagnose the density drift for the whole layer. The density drift, in turn, is used to diagnose the amount of fluid that is needed to cross the bottom (top) interface in order to restore the density coordinate of the entire layer.

5. Verifications in one-dimensional vertical diffusion problems

It has been asserted that isopycnal models cannot simulate the full range of vertical diffusion processes $z$-coordinate models can simulate. For example, a statically stable two-layer water column with insulated top and bottom boundaries will eventually become vertically homogeneous due to buoyancy diffusion. This diffusion process can be easily simulated in a two-level $z$-coordinate model. However, a two-layer isopycnal model cannot simulate this process because the interface must remain at its initial position in order to conserve overall density while maintaining the reference densities in the two layers. The question is challenging. If an isopycnal model cannot simulate such a simple one-dimensional diffusion process, its ability to simulate diapycnal mixing in the real ocean is certainly doubtful. Having derived algorithms for computing diapycnal diffusive and advective scalar fluxes, we feel that it is necessary to address this issue in one-dimensional diapycnal diffusion models. The objectives are, on the one hand, to demonstrate the ability of isopycnal models to simulate diapycnal diffusion processes, and on the other hand, to verify the derived algorithms.

In this section two kinds of one-dimensional boundary condition problems are solved: 1) Dirichlet boundary condition problems, in which values of the variables themselves are specified at the boundaries, and 2) Neumann boundary condition problems, in which gradients of the variables are given at the boundaries. Dirichlet problems will be used to test the ability of the derived algorithms to inflate massless layers or to diffuse strong discontinuities. Therefore, initially some layers are specified as massless. For Neumann problems, we will confine ourselves to the insulated water column diffusion problems mentioned above, or homogeneous boundary condition problems, where we can test conservative properties of the derived algorithms in long time integration.

a. Dirichlet boundary value problems

Three experiments are carried out to simulate vertical diffusion in a water column with fixed densities at the top and bottom boundaries. In the first two experiments density is an independent variable, while in the third experiment density is a function of temperature and salinity. The one-dimensional model consists of ten isopycnal layers with equal density intervals. The depth of the water column is 50 m. In order to see how well the derived algorithms work in dealing with massless layers, layer thicknesses of three layers near the top boundary (layers 1–3) and two layers in the interior (layers 6–7) are initially set to zero. For all three experiments, $H_0$ is set to 0.1 m.

For the cases in which density is an independent thermodynamic variable, one can predict that the equilibrium vertical density profile will be linear with depth and the thickness of each isopycnal layer will be the same. Figure 8 shows the time evolution of the solution for $K_e = 0.005$ cm$^2$ s$^{-1}$. The initial stage of the process is characterized by rapid inflation of the massless layers. Since vertical diffusion is the only process that cause vertical displacements of isopycnals, diapycnal velocity is equal to $-\partial Z/\partial t$. Figure 8 shows that inflation of massless layers at the top boundary is due to convergence of upward diapycnal velocities, while interior massless layer inflation is associated with convergence of upward diapycnal velocity from below and downward diapycnal velocity from above. After the initial stage, the diapycnal velocities at the interfaces become small and gradually approach zero. At year 100, the thicknesses of all isopycnal layers are virtually
equal. For comparison, we carried out another experiment using a stability-dependent vertical diffusivity, $K_d = 3 \times 10^{-4} N^{-1}$. The time evolution is shown in Fig. 9. Compared to the constant diffusivity case, inflation of massless layers is slower and diapycnal velocities are smaller in the initial stage. Later, however, diapycnal mass exchange is more pronounced than in the constant-diffusivity experiment. Therefore, the stability-dependent diffusivity spreads diapycnal mass exchange more evenly over time. Although different in the early stages, the diffusion processes associated with both the stability-dependent and constant diffusivity reach the same steady state: uniform stratification.

In the third experiment, density is determined by the United Nations Educational, Scientific and Cultural Organization (UNESCO) equation of state, and temperatures and salinities are fixed at the top and bottom boundaries. In this case, one can predict that the equilibrium vertical temperature and salinity profiles will be both linear with depth. Figure 10 displays the evolution of layer thickness with time in the water column for $K_d = 0.005 \text{ cm}^2 \text{s}^{-1}$. The initial stage is still characterized by inflation of massless layers near the top boundary and in the interior. A equilibrium state is reached at the end of 100-yr time integration. The equilibrium temperature and salinity profiles are, as expected, linear with depth (not shown). Due to the non-linearity of the equation of state, however, the equilibrium layer thickness increases with depth.

b. Homogeneous Neumann boundary value problems

We now turn our attention to diffusion problems in insulated water columns. For simplicity, density is assumed to be an independent thermodynamic variable in the following experiments. First let us look at the time evolution of layer thickness solutions in a ten-layer insulated water column shown in Fig. 11. The water column is discretized by a constant density interval. Initially the thickness of each isopycnic layer is 5 m. Note that the stratification of the water column is initially symmetric about the middle interface and that the top and bottom boundaries are insulated. Thus, the stratification should remain symmetric about the middle interface in the course of diffusion. This symmetry can be clearly seen in Fig. 11. Figure 11 also shows that the diapycnal velocity ($= -\delta Z/\delta t$) is downward in the top half of the water column and upward in the bottom half. As a result, isopycnic layers in the interior become inflated at the expense of the layers adjacent to the top and bottom boundaries, and thus the overall density gradient in the water column decreases with time. Note that every time a pair of interior layers
achieves the status of bounding layers, mass convergence in those layers turns into mass divergence, resulting in kinks in the time plot. Furthermore, diapycnal velocities at all interfaces decrease in magnitude with time, which indicates that the diffusion process slows down as the column becomes more and more homogeneous.

Although the model simulates the diffusion process rather successfully, it fails to reach the final state of a homogeneous water column. The final state is a twolayer water column. Obviously this is due to insufficient density coordinate resolution: the model does not have an isopycnic layer whose reference density equals the mean density of the water column. Theoretically, as density grid size goes to zero (or as the number of layers becomes infinite), such a mean density isopycnic layer will be included in the model and thus the complete diffusion process can be simulated. If the mean density of the water column is known in advance, however, one can always include a layer of this coordinate value in the model and thus simulate the complete diffusion process with a finite number of layers. For instance, the two-layer insulated water column diffusion problem mentioned at the beginning of this section can be solved in a three-layer isopycnic model (see Fig. 12). In Fig. 12 the water column initially consists of two layers of equal depth and a massless isopycnic layer whose associated density equals the mean density of the water column. As time goes on, this layer is inflated at the expense of the top and bottom layers. After roughly 8.4 years of integration, the model reaches the final state: a homogeneous water column.

It has been shown that with appropriate density coordinate discretization any insulated water column diffusion process can be simulated satisfactorily in isopycnic models. However, one must determine the mean density of the water column in advance, which is not always convenient. This inconvenience can be removed if hybrid coordinates are used. Figure 13 shows the solution to the above two-layer water column diffusion problem in a two-layer hybrid-coordinate model. Density is allowed to vary in the top layer but is fixed in the bottom layer. As time goes on, buoyancy is transferred from the top layer to the bottom layer. To maintain its reference density, the bottom layer must continue to transfer mass to the top layer. After roughly 8.4 years of integration, the bottom layer becomes massless and the density of the top layer becomes equal to the mean density of the water column. In the three-dimensional isopycnic OGCMs, isopycnic layers are
Table 1. Potential density, temperature, salinity, and thickness of a water column at 25°S, 30°W.

<table>
<thead>
<tr>
<th>Layer</th>
<th>( \sigma_0 )</th>
<th>( \theta ) (°C)</th>
<th>( S ) (psu)</th>
<th>( \Delta Z ) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24.9026</td>
<td>23.780</td>
<td>36.5765</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>25.0106</td>
<td>23.620</td>
<td>36.6565</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>25.0785</td>
<td>23.410</td>
<td>36.6645</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>25.2195</td>
<td>23.940</td>
<td>36.6695</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>25.4449</td>
<td>21.955</td>
<td>36.5955</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>25.6644</td>
<td>20.865</td>
<td>36.4865</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>25.8964</td>
<td>19.460</td>
<td>36.2990</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>26.1152</td>
<td>17.600</td>
<td>35.9705</td>
<td>50</td>
</tr>
<tr>
<td>9</td>
<td>26.3242</td>
<td>15.525</td>
<td>35.6075</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>26.5279</td>
<td>13.425</td>
<td>35.2850</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>26.8041</td>
<td>10.345</td>
<td>34.8865</td>
<td>200</td>
</tr>
<tr>
<td>12</td>
<td>27.0875</td>
<td>6.525</td>
<td>34.4975</td>
<td>200</td>
</tr>
<tr>
<td>13</td>
<td>27.2731</td>
<td>4.195</td>
<td>34.3805</td>
<td>200</td>
</tr>
<tr>
<td>14</td>
<td>27.4232</td>
<td>3.360</td>
<td>34.4635</td>
<td>200</td>
</tr>
<tr>
<td>15</td>
<td>27.5890</td>
<td>3.125</td>
<td>34.6435</td>
<td>300</td>
</tr>
<tr>
<td>16</td>
<td>27.7395</td>
<td>3.090</td>
<td>34.8280</td>
<td>500</td>
</tr>
<tr>
<td>17</td>
<td>27.8225</td>
<td>2.935</td>
<td>34.9140</td>
<td>1000</td>
</tr>
<tr>
<td>18</td>
<td>27.8486</td>
<td>2.205</td>
<td>34.8670</td>
<td>1000</td>
</tr>
<tr>
<td>19</td>
<td>27.8744</td>
<td>0.700</td>
<td>34.7630</td>
<td>1000</td>
</tr>
</tbody>
</table>

The problem under consideration is a combination of the Dirichlet and Neumann boundary value problems. We fix \( \theta \) and \( S \) at the top boundary of the water column and specify no flux boundary conditions for \( \theta \) and \( S \) at the bottom. The vertical diffusivity is set to 0.5 cm² s⁻¹, and potential density is computed from the UNESCO equation of state. The problem is intended to model the process of the oceanic vertical-stratification destruction by vertical turbulent mixing. As shown in Fig. 14, the model successfully simulate the complete vertical homogenization process. Note that the initial salinity profile is not monotonic (see Table 1). One purpose of this experiment is to test the algorithms' ability to diffuse a realistic ocean profile without creating spurious \( \theta \) and \( S \) extremes. Figure 15 shows the solutions of \( \theta \) and \( S \) as functions of depth and time. The solutions were interpolated to the original 19-level \( z \) grid, in which \( \theta \) and \( S \) are carried at depths of 5, 15, 25, 40, 62.5, 87.5, 125, 175, 250, 350,

For this modification is again to test the algorithms' ability to handle massless layers in OGCMs. In the original data, potential temperature in the bottom level is 0.72°C and the corresponding potential density is smaller than that in the level above it. Since potential density, as a vertical coordinate, must be vertically monotonic, some modification is needed. One way around it is to use \( \sigma_1 \) or \( \sigma_2 \), both of which are vertically monotonic. In practice, however, virtually all isopycncal OGCMs use \( \sigma_0 \) as vertical coordinates. Therefore, we still keep \( \sigma_0 \) as vertical coordinates in our one-dimensional isopycnal model and change the potential temperature value in the bottom layer to 0.7°C to maintain a monotonic initial profile of \( \sigma_0 \).

c. An example for the real ocean

In the previous two subsections, we verified the algorithms in one-dimensional Dirichlet and Neumann boundary value problems. The dimensions and vertical diffusivities used in the one-dimensional models were arbitrarily chosen to show the general applicability of the algorithms to vertical diffusion problems. Since the algorithms are to be used in isopycncal OGCMs, we shall test them in this subsection on a one-dimensional problem using the water column layout and vertical diffusivity value appropriate to the real ocean. To this end, we choose a water column from a three-dimensional \( \theta \) and \( S \) initial condition derived from the Levitus (1982) dataset for a 19-level GFDL ocean model. The water column is located at about 25°S, 30°W. Based on the data of the 19-level water column, we constructed a 19-layer water column, whose potential density, temperature, salinity, and layer thickness are summarized in Table 1. In our 19-layer one-dimensional isopycnal model, the density coordinate values and the initial values of potential temperature, salinity, and layer thickness are prescribed, respectively, by the values of \( \sigma_0 \), \( \theta \), \( S \), and \( \Delta Z \) in Table 1. Note that the data in Table 1 are slightly different from the original data. In the original 19-level water column, thickness is 10 m for levels 1–3 and 20 m for level 4, while in Table 1 thickness is 50 m for layer 1 and 0 m for layers 2–4. The reason for this modification is again to test the algorithms' ability to handle massless layers in OGCMs. In the original data, potential temperature in the bottom level is 0.72°C and the corresponding potential density is smaller than that in the level above it. Since potential density, as a vertical coordinate, must be vertically monotonic, some modification is needed. One way around it is to use \( \sigma_1 \) or \( \sigma_2 \), both of which are vertically monotonic. In practice, however, virtually all isopycncal OGCMs use \( \sigma_0 \) as vertical coordinates. Therefore, we still keep \( \sigma_0 \) as vertical coordinates in our one-dimensional isopycnal model and change the potential temperature value in the bottom layer to 0.7°C to maintain a monotonic initial profile of \( \sigma_0 \).

![Fig. 14. Solution of isopycnal layer thickness during vertical homogenization of an oceanic water column at 25°S, 30°W by vertical turbulent mixing in a 19-layer one-dimensional isopycnal model. The solution is sampled in a 50-yr time interval. Potential density is calculated by the UNESCO equation of state. Vertical diffusivity is 0.5 cm² s⁻¹. Vertical axis: depth (m). Horizontal axis: time (unit: 100 yr).](image-url)
Fig. 15. As in Fig. 14 but for solutions of potential temperature and salinity. The solutions are interpolated to a 19-level z grid. The solutions for the entire vertical diffusion process are shown on the right panels, which are sampled in a time interval of 50 yr. The left panels show the solutions for the first 300 yr, which are sampled every 5 yr. Vertical axis: depth (m). Horizontal axis: time (unit: years for the left panels and 100 yr for the right panels).

Fig. 16. As in Fig. 15 but for solutions of potential temperature and salinity of a 19-level z-coordinate model.
500, 700, 900, 1100, 1350, 1750, 2500, 3500, and 4500 m. In order to show the details of $\theta$ and $S$ time variations in the initial stage of vertical diffusion, the model solutions for the first 300 years are shown in the left panels of Fig. 15. As can be seen, the model successfully diffuses the realistic $\theta$ and $S$ profiles without creating any spurious extremes. In order to verify the model results, we solved the same initial and boundary value problem using the 19-level z-coordinate grid. The solutions of $\theta$ and $S$ are shown in Fig. 16. Although the solutions of the z-grid model are very smooth, kinks occur at the depths where $\theta$ and $S$ are carried in the model. Especially, kinks are obviously seen at depths of 2500 and 3500 m. In general, the solutions of the isopycnal model are not as smooth as those of the z-coordinate model. The main reason is that, except for the kinks at the z grid points, there are kinks associated with those in the interface-depth field shown in Fig. 14. As noted in section 5b, the kinks in the interface-depth field occur whenever a layer physically vanishes at the boundary. In other words, the kinks only occur when the active vertical grid points of the model are reduced. Note that vertical resolution is time invariant in the z-coordinate model and decreases with time in the isopycnal model. The decreasing vertical resolution in the isopycnal model may be responsible for the fact that the differences between the two model solutions become larger with time. Nevertheless, the $\theta$ and $S$ solutions in the two models are in reasonably good agreement. Especially in the initial stage of massless layer inflation, the two solutions agree very well. The slight differences in the initial stage solutions may be due to the difference in the vertical resolution between the two models. Initially there is only one grid point in the top 50 m in the isopycnal model, while in the z-grid model there are four grid points for the same depth range. (The initial values of $\theta$ and $S$ in the top four levels are set to those of layer 1 in the isopycnal model.) Finally, we note that the 23°C and 36.5-mil contour lines reach 4500-m depth at about the same time in the two models. This indicates that despite the differences in the time evolution due to the difference in vertical resolution, the time period for the vertical homogenization is about the same in the two models.

By carrying out this one-dimensional real ocean experiment, we also obtain an estimate of the algorithms’ computational cost in OGCMs. In the time integration of the 19-layer one-dimensional model, we used a time step of 4 days. It took about 0.1-s CPU time for one model year integration on a Dec-alpha 3000 workstation. For a 19-layer global isopycnal model with a horizontal resolution of $4^\circ$(longitude) $\times$ $3^\circ$(latitude), that is, with 5400 horizontal grid points, the cost of the algorithms is then about 9 min for one model year on the Dec-alpha. Therefore, the algorithms’ cost is even affordable on a Dec-alpha machine when a large time step is used. But for short time steps the cost is still high. According to our ongoing coarse-resolution $[4^\circ$(longitude) $\times$ $3^\circ$(latitude)] run of a 10-layer global isopycnal ocean model on the Dec-alpha, the algorithms take about 17.8% of the total CPU time when the model is integrated synchronously with a time step of 5 h. To accelerate the time integration, we do the diapycnal computation twice every 10 steps (50 h). As a result, the cost of the algorithms is reduced to 4.2% of the total CPU time.

6. Summary and conclusions

In this paper, we have addressed the problem of how to numerically incorporate diapycnal mixing and advection processes into ocean general circulation models. A general expression of diapycnal velocity is derived for the case that density is a function of both potential temperature and salinity. Since the expression is independent of turbulence closure schemes, it can be viewed as a generalized definition of diapycnal velocity. Although different turbulence closure schemes result in different expressions of diapycnal velocity, the expressions must be consistent with the definition.

The basic ideas behind the algorithms derived in this study can be summarized as follows. The difference between $z$ and $\rho$ coordinates is that variations of length scale less than $(2K_c\Delta)^{-1/2}$ in a tracer field are filtered out in advance in the design of the $z$ grid, and thus the linear stability criterion (10) is satisfied once and for all. This is not the case in $\rho$ coordinates. The approach adopted here is to smooth the tracer field by mass-weighted vertical averaging so that diagnosed vertical tracer gradients do not cause linear instability in the model time integration. Since the mass-weighted average is only used for vertical gradient computation, the resolution advantage of $\rho$ coordinates in regions of strong vertical gradients is retained. Diagnosis of diapycnal velocity and restoration of density coordinates are physically consistent. The two tasks are accomplished by diagnosing density variation rates in the upper and lower halves of each isopycnic layer and then converting them into diapycnal mass exchange rates using appropriate mapping coefficients.

The algorithms are verified in one-dimensional isopycnal models of diapycnal diffusion for Dirichlet and homogeneous Neumann boundary value problems. The results confirm that the algorithms are conservative and are able to produce diapycnal diffusive and advective scalar fluxes that are physically reasonable and mathematically continuous. With the algorithms, we have shown that isopycnal models can simulate vertical mixing processes leading to complete homogenization in an insulated water column if density coordinate values are properly chosen. We show that the latter restrictions can be easily circumvented by including a model layer with variable density and thickness. Finally, through a simulation of a realistic ocean profile diffusion process, we show that the algorithms not only have the ability to simulate vertical mixing processes in the real ocean.
but also have computational efficiency good enough for application to three-dimensional isopycnal OGCMs.

The algorithms have been tested in parameter studies of three-dimensional isopycnal OGCMs for box basins and used in global isopycnal OGCM simulations. The results will be reported in separate papers.

**Acknowledgments.** The algorithms discussed in this paper were first developed and tested by the author in 1991 for a simple nonlinear equation of state from which temperature can be analytically solved as a function of potential density and salinity, which was part of the author’s dissertation research at the Rosenstiel School of Marine and Atmospheric Science of the University of Miami. The author would like to thank his adviser, Dr. Rainer Bleck, for his advise, encouragement, and many helpful discussions. The author succeeded in extending the algorithms to the UNESCO equation of state at the Geophysical Fluid Dynamics Laboratory, which was part of the author’s work of developing a global isopycnal ocean general circulation model during 1992 and 1993. The author would like to thank the UCAR Ocean Modeling Postdoctoral Program and Dr. Kirk Bryan for their support. The algorithms were improved and written in the present form at the Joint Institute for the Study of Atmosphere and Ocean (JISAO) of the University of Washington. The experiment in section 5c was suggested by an anonymous reviewer. The comments of this reviewer were also helpful in improving the text. All computations in this paper were carried out on the Dec-alpha work stations at the JISAO. The author would like to thank Dr. Edward Sarachick for encouraging the author to publish the algorithms. This research was supported 50% by a grant from the Department of Energy National Institute for Global Environmental Change and 50% by a grant from the National Oceanic and Atmospheric Administration Office of Global Programs to the Hayes Center at JISAO.

**REFERENCES**


