JAMES Journal of Advances in Modeling Earth Systems

AGU ADVANCING EARTH AND SPACE SCIENCE

RESEARCH ARTICLE

10.1029/2019MS001992

Special Section:
Dynamical Cores of Oceanic Models Across All Scales and Their Evaluation

Key Points:

- A new discretization of a neutral diffusion operator is detailed, with the new method not based on a rotated tensor
- The new algorithm overcomes many numerical deficiencies of previous approaches
- Idealized test cases show minimal spurious dianeutral diffusion for linear and nonlinear equations of state

Correspondence to:
A. E. Shao,
aeshao@uvic.ca

Citation:
Shao, A. E., Adcroft, A., Hallberg, R., & Griffies, S. M. (2020). A general-coordinate, nonlocal neutral diffusion operator. *Journal of Advances in Modeling Earth Systems*, *12*, e2019MS001992. https://doi.org/10.1029/2019MS001992

Received 16 DEC 2019
Accepted 3 NOV 2020
Accepted article online 9 NOV 2020

A General-Coordinate, Nonlocal Neutral Diffusion Operator

Andrew E. Shao[1], Alistair Adcroft[2,3], Robert Hallberg[2,3], and Stephen M. Griffies[2,3]

[1]School of Earth and Ocean Sciences, University of Victoria, Victoria, BC, Canada, [2]Atmospheric and Oceanic Sciences Program, Princeton University, Princeton, NJ, USA, [3]NOAA/Geophysical Fluid Dynamics Laboratory, Princeton, NJ, USA

Abstract We present a neutral diffusion operator appropriate for an ocean model making use of general vertical coordinates. The diffusion scheme uses polynomial reconstructions in the vertical, along with a horizontally local but vertically nonlocal stencil for estimates of tracer fluxes. These fluxes are calculated on a vertical grid that is the superset of model columns in a neutral density space. Using flux-limiters, the algorithm dissipates tracer extrema locally, and no new extrema are created. A demonstration using a linear equation of state in an idealized configuration shows that the algorithm is perfectly neutral. When using the nonlinear TEOS-10 equation of state with a constant reference pressure, the algorithm compares nearly exactly to a case discretized onto isopycnal surfaces and using along-layer diffusion. The algorithm's cost is comparable to that of tracer advection and can be readily implemented into ocean general circulation models.

Plain Language Summary One prominent effect of ocean turbulence is the mixing of quantities such as temperature, salinity, and other tracers carried by the seawater. For turbulence associated with geostrophic motions, this mixing occurs along neutral directions, which are directions where buoyancy does not change. In computer models that do not explicitly represent such geostrophic eddy motions, this type of turbulent mixing is parameterized by diffusion oriented along the neutral directions. A common means to represent neutral diffusion is via rotating the diffusion fluxes to align with neutral directions (which are typically not aligned with the model grid lines). Unfortunately, this rotation can result in spurious numerical artifacts such as the creation of new minimum or maximum values of tracer concentration (referred to as extrema). In this paper, we propose a new algorithm for calculating and applying neutral diffusion in sublayers of a numerical model. We show that this new method does not create extrema, and it provides a physically accurate representation of turbulent diffusive mixing along neutral directions.

1. Introduction

Observational evidence supports the hypothesis that tracer stirring by mesoscale eddies preferentially acts along neutral directions (e.g., Iselin, 1939; Ledwell et al., 1993). This stirring leads to a downscale cascade of tracer variance. Ultimately, breaking gravity waves and molecular diffusion irreversibly mix tracers thus acting to homogenize tracers along neutral tangent planes (McDougall, 1987a). Neutral diffusion parameterizes this downgradient tracer mixing for ocean models that do not adequately resolve mesoscale stirring (McDougall & Church, 1986; Olbers et al., 1985; Redi, 1982; Solomon, 1971). Ocean and coupled climate simulations are sensitive to the strength of neutral diffusion given its ability to exchange tracers between the upper and abyssal oceans when neutral slopes are steep (Gnanadesikan et al., 2005, 2015; Gregory, 2000; Griffies et al., 2015; Hieronymus & Nycander, 2013; Lee & Nurser, 2012; Morrison et al., 2013; Sijp & England, 2009; Sijp et al., 2006; Wolfe et al., 2008; Zika et al., 2014). Neutral diffusion also affects the strength of water mass transformation through thermobaricity and cabbeling, both of which result from the nonlinear equation of state (Iudicone et al., 2008; Klocker & McDougall, 2010a, 2010b; McDougall, 1987b).

There are two general methods presently in use to realize neutral diffusion in ocean models. One method follows the ideas of Solomon (1971) and Redi (1982) by formulating a rotated diffusion operator oriented along the neutral tangent plane. This formulation is widely used for neutral diffusion in geopotential and terrain-following coordinate ocean models (Beckers et al., 2000; Cox, 1987; Griffies et al., 1998;

Lemarié et al., 2012). The second method is used in isopycnal models, whereby the neutral tangent plane is assumed to be locally parallel to isopycnals. The along-coordinate diffusion operator is then formulated as lateral diffusion within the projected nonorthogonal coordinate frame defined by isopycnals (e.g., Bleck & Smith, 1990; Hallberg, 2005).

Both of the above methods have their limitations. Griffies et al. (1998), Beckers et al. (1998), Gnanadesikan (1999a), Beckers et al. (2000), and Lemarié et al. (2012) noted that the downgradient property of neutral diffusive fluxes is not generally guaranteed by a discrete rotated operator. Such discrete rotated operator schemes can produce spurious extrema, especially when acting on passive tracers. In contrast, the discrete isopycnal diffusion operator in isopycnal coordinates is downgradient by construction and thus dissipates tracer extrema. However, no globally defined surface, such as an isopycnal, accurately approximates neutral tangent planes throughout the World Ocean (McDougall, 1987a; Klocker et al., 2009; Klocker & McDougall, 2010a, 2010b; McDougall & Jackett, 1988). Consequently, isopycnal diffusion does not generally admit the suite of physical processes possessed of neutral diffusion, including thermobaricity resulting from changes in pressure (see section A.27 of IOC et al., 2010).

In this study, we describe a new neutral diffusion scheme that overcomes the numerical issues associated with local, rotated operators. The algorithm is implemented in the Modular Ocean Model version 6 (MOM6), a general-coordinate, finite volume ocean general circulation model (whose application to a realistic, global configuration was detailed in Adcroft et al., 2019). The approach taken here constructs sublayers between two model columns whose interfaces are neutrally buoyant. Tracer fluxes are then calculated along these sublayers, and along with appropriate flux limiters, ensures the downgradient property for its discrete tracer fluxes. Hence, the scheme dissipates tracer extrema and is positive definite. In contrast to the along-layer diffusion approach used in isopycnal models, our method represents the physical processes of cabbeling and thermobaricity associated with neutral diffusion and a nonlinear equation of state. This property is ensured since the diffusive fluxes are oriented according to locally defined neutral directions rather than globally defined isopycnals.

The remainder of this paper consists of the following sections. In section 2, we summarize the basic concepts associated with neutral directions, neutral diffusive fluxes, and neutral diffusion operators. This section is largely a review, but it is essential to expose the elementary ideas to guide the numerical algorithm details. The discussion and background serve to motivate the three phases of a new neutral diffusion algorithm detailed in section 3 that calculates diffusive fluxes along sublayers of the model. In section 4, we describe the idealized numerical tests used to demonstrate that the algorithm results in minimal dianeutral diffusion for both linear and nonlinear equations of state. We close the main portion of the paper in section 5 with a discussion of extensions to the algorithm and other considerations when applied in realistic ocean configurations. We present in Appendix A diagrams demonstrating the expected positions of the diffusive sublayers between two columns for a variety of common oceanic scenarios.

2. Motivating a New Discretization of the Neutral Diffusion Operator

Here we provide a review of various physical and mathematical properties of neutral directions and neutral diffusive fluxes. This discussion serves as a guide to properties desired for a discrete neutral diffusion algorithm and introduces some of the equations that are used in the algorithm proposed in this paper.

2.1. Commonly Used Discretizations

The displacement of a water parcel with conservative temperature Θ, salinity S, and pressure P occurs along a “neutral direction” if no buoyancy force is experienced along the path of the displacement. When comparing two water parcels (subscripts 1 and 2), the difference in buoyancy can be calculated from the difference in the water parcels' densities displaced to the midpoint pressure $\bar{P}$

$$\bar{P} = \frac{P_1 + P_2}{2}. \tag{1}$$

Note here that the pressure used here arises from the weight of the water column above the water parcel (calculated differently if making the Boussinesq approximation or not).

The difference in density can then be approximated in terms of the partial derivatives of density with respect to Conservative Temperature, $\alpha = -(\partial\rho/\partial\Theta)_{S,P}$, and salinity, $\beta = (\partial\rho/\partial S)_{\Theta,P}$, as

$$\Delta\rho \approx \frac{1}{2}\left[\left(\alpha(\Theta_1,S_1,\bar{P})+\alpha(\Theta_2,S_2,\bar{P})\right)\left(\Theta_1-\Theta_2\right)-\left(\beta(\Theta_1,S_1,\bar{P})+\beta(\Theta_2,S_2,\bar{P})\right)\left(S_1-S_2\right)\right]. \tag{2}$$

Note that if a global reference pressure is used, $\Delta\rho$ represents a difference in potential density. A *neutral direction* is that direction along which a water parcel displacement satisfies the neutrality condition

$$\Delta\rho = 0. \tag{3}$$

The corresponding unit direction normal to the neutral direction is given by

$$\hat{\boldsymbol{\gamma}} = \frac{-\alpha\,\nabla\Theta + \beta\,\nabla S}{|-\alpha\,\nabla\Theta + \beta\nabla S|}. \tag{4}$$

Having defined the direction normal to the neutral tangent plane, we can thus express the time evolution of a tracer C within a layer due to neutral diffusion by

$$\frac{\partial C}{\partial t} = \frac{1}{h^\gamma}\nabla_\gamma \cdot (h^\gamma\,\kappa\,\nabla_\gamma C), \tag{5}$$

where h^γ is the layer thickness and ∇_γ is a horizontal gradient operator that computes the divergence of fluxes along the neutral tangent plane (see Figure 6.4 of Griffies, 2004, for details). Generally, layered isopycnal models assume that the neutral surface is aligned with isopycnal surfaces, in which case the right hand side of Equation 5 is downgradient diffusion with fluxes aligned along the coordinate surface.

For most other ocean models, an along-neutral flux is calculated using a rotated tensor assuming that the slope of the neutral surface with respect to the model surface is small. Following Griffies et al. (2004), the flux obtained via a coordinate transformation from the neutral tangent plane to a generalized vertical coordinate (denoted by subscript r) is thus

$$\mathbf{F} = -\kappa\left[\nabla_\gamma C + \hat{\mathbf{z}}(\boldsymbol{S}\cdot\nabla_\gamma C)\right], \tag{6}$$

where

$$\nabla_\gamma = \nabla_r + \boldsymbol{S}\partial_r \tag{7}$$

is the horizontal derivative operator computed on the neutral tangent plane; ∇_r is the corresponding operator defined according to the generalized vertical coordinate, r; $\partial_r = (\partial z/\partial r)\partial_z$ is the vertical derivative, and (as in Equation 5.28 of Griffies et al., 2004) $\mathbf{S}$ is the slope of the neutral direction relative to the r coordinate surface

$$\boldsymbol{S} = \nabla_\gamma r = -z_r r_\gamma \nabla_r \gamma = -\left[\frac{\alpha\,\nabla_r\Theta - \beta\,\nabla_r S}{\alpha\,\partial_z\Theta - \beta\,\partial_z S}\right], \tag{8}$$

where z_r is the depth of the general surface coordinate surface r and r_γ is the position of the neutral surface in the r coordinate.

Gent and McWilliams (1990) showed that, under the assumption that $\mathbf{S}$ is small, the neutral diffusive flux (Equation 6) can be written in the form of a matrix vector multiplication

$$\boldsymbol{F}^{\mathrm{redi}} = -\kappa\begin{pmatrix} 1 & 0 & S_x \\ 0 & 1 & S_y \\ S_x & S_y & \boldsymbol{S}\cdot\boldsymbol{S} \end{pmatrix}\begin{pmatrix} \partial C/\partial x \\ \partial C/\partial y \\ \partial C/\partial z \end{pmatrix}, \tag{9}$$

where S_x and S_y are calculated from horizontal density gradients along the model's surface. This form of the neutral diffusive flux has been utilized in both geopotential coordinate models (Griffies et al., 1998) and terrain-following models (Lemarié et al., 2012), with the resulting neutral diffusion operator given by the three-dimensional flux convergence

$$\mathcal{R} = -\nabla\cdot\boldsymbol{F}^{\mathrm{redi}} = \nabla_r\cdot\left(\kappa\,\nabla_\gamma C\right) + \partial_r\left(\kappa\,\boldsymbol{S}\cdot\nabla_\gamma C\right). \tag{10}$$

2.2. Known Shortcomings of Rotated Diffusion Operators

There are two key properties of diffusion in the continuum: (A) dissipation of local tracer extrema and (B) reduction of global, spatially integrated tracer variance (i.e., as defined in Equation 13.14 of Griffies, 2004). These two related properties follow from the downgradient nature of diffusive fluxes. They hold for diffusion oriented in any direction; that is, for any diffusive process characterized by a symmetric nonnegative diffusion tensor. However, these properties are not guaranteed when discretizing rotated diffusion operators.

Griffies et al. (1998) used this global tracer variance reduction as a framework to formulate a local, linear, consistent rotated neutral diffusion operator. The result is a weighted form of the traditional nine-point stencil employed for grid-oriented diffusion (see Figure 2 in Beckers et al., 1998). However, as noted by Beckers et al. (1998), any linear, consistent, and local (e.g., nine-point stencil) rotated diffusion scheme cannot guarantee that no new extrema are produced. This limitation largely follows from their proof showing that at least one of the coefficients of the stencil (other than the center value) must be negative to maintain consistency for any rotation of the tensor. Without being able to guarantee positive coefficients, any such scheme can produce new extrema (Beckers et al., 2000; Gnanadesikan, 1999b; Lemarié et al., 2012).

Various solutions have been proposed to address the problem of monotonicity. Constraints, such as the neutrality condition (Equation 3), can reduce the magnitude of extrema produced for temperature and salinity (Griffies et al., 1998). Thermodynamically passive tracers, such as biogeochemical tracers, are generally more problematic (Gnanadesikan, 1999b) since they do not have an analogous constraint. Applying one constraint to Θ and S, but not the passive tracers thus goes against the general principle that numerics should minimally impact T, S, and passive tracer relationships, that is, the same algorithms should be applied to active and passive tracers (though even this is not necessarily guaranteed if using monotonic limiters for advection and/or diffusion). Alternatively (as reviewed and evaluated by Beckers et al., 2000), nonmonotonicity of linear schemes can also be reduced by changing the stencil depending on the neutral slope; monotonicity can also be achieved by relaxing consistency or introducing nonlinearity to the operator.

Further problems arise near boundaries and in weakly stratified regions where the neutral slopes are very steep (Gerdes et al., 1991; Griffies et al., 1998) due to the vertical gradient of density in the denominator of the neutral slope calculation (Equation 8). In such cases, some form of regularization (slope tapering) is introduced to ensure that the resulting diffusive fluxes are numerically stable.

2.3. Vertically Nonlocal Implementation of Neutral Diffusion

The rotated operator approach estimates the neutral direction and diffusive fluxes using a locally connected grid stencil; that is, each member of the stencil is a neighbor of the original cell. Vertically nonlocal methods for estimating neutral directions for the rotated operator approach have been demonstrated in observational contexts (e.g., Groeskamp et al., 2019, and Jackett & McDougall, 1997); however, these studies still use a form of the rotated operator to calculate diffusive fluxes.

In this paper, we consider a new paradigm for the calculation of fluxes that is nonlocal in the vertical and fluxes can occur between any cells of an adjacent water column. The main idea of our method is to construct sublayers between columns whose interfaces are neutrally buoyant. This method is partially inspired by the algorithm in the GOLD isopycnal ocean model (Dunne et al., 2012). In the GOLD algorithm, there can be advective and isopycnal diffusive transport between interior layers with fixed potential density and layers within a bulk mixed layer possessing variable density (Hallberg, 1995).

As we show in this paper, two features of the new algorithm lead to a diffusion scheme that avoids the numerical issues with the rotated operator approach. The new algorithm does not calculate neutral slopes by computing the ratio of density derivatives as in Equation 8. Instead, it evaluates Equation 2 to determine the neutral position. Fluxes are calculated along the sublayers (without need for a rotated operator) and are guaranteed to be positive definite and extrema diminishing when employing flux limiters.

3. Neutral Diffusion Via Vertically Nonlocal Unidirectional Fluxes

Here we describe our new, nonlocal, finite-volume implementation of neutral diffusion, with four main elements of the algorithm described in separate subsections. Section 3.1 describes how a polynomial representation of the subgrid scale profiles of Θ and S is constructed. Section 3.2 details the construction of model

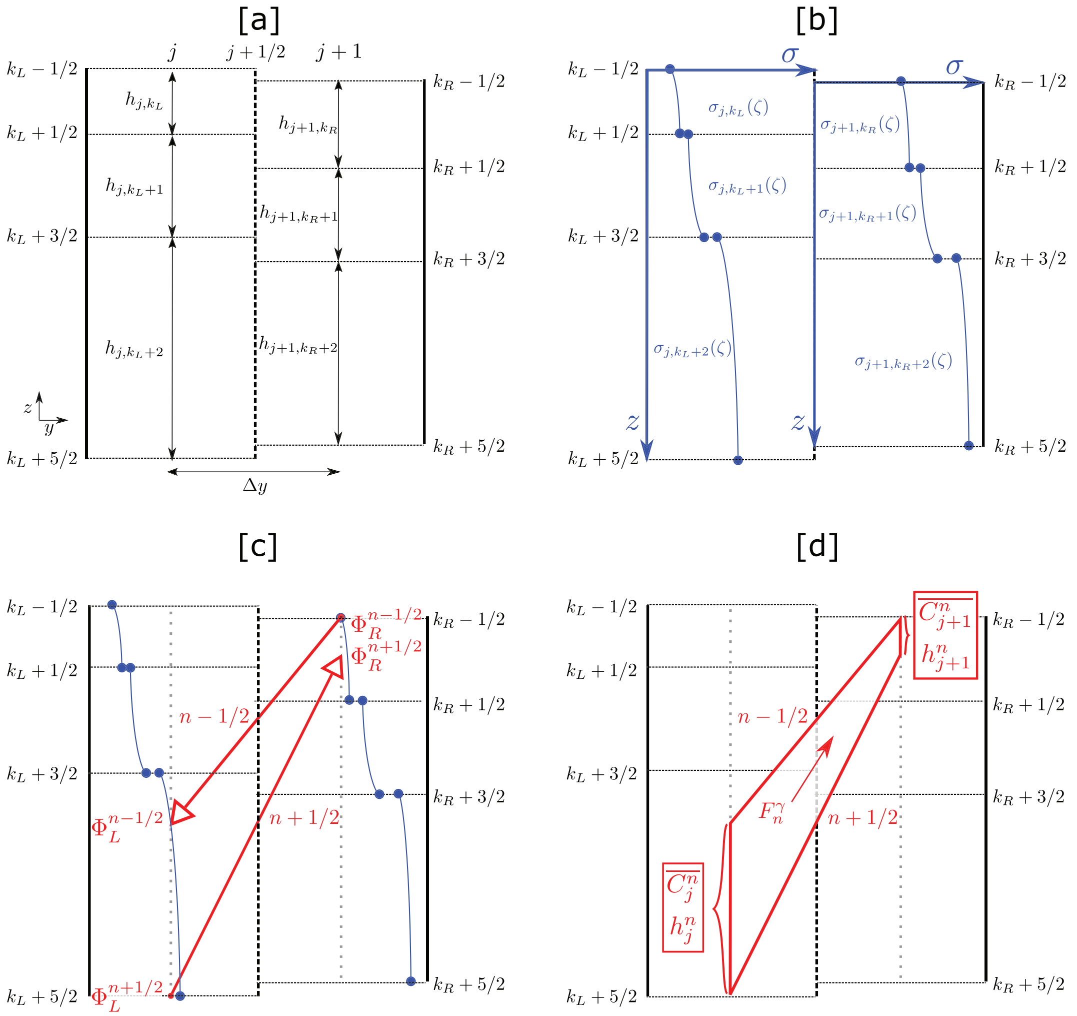

Figure 1. Diagram illustrating the quantities used in this paper to calculate diffusive fluxes within a layer bounded by neutral surfaces between two water columns. Two adjacent grid cells at tracer points j and $j+1$ with the velocity point between the two columns are denoted as vertical, dashed lines, and denoted by $j+1/2$. Each column has three cells in the vertical whose vertical interfaces (dashed, horizontal, black lines) are indicated using 1/2 notation. Panel (a) shows metrics of the two columns including the thickness of each cell (h) and the distance Δy between the two thickness points. Panel (b) shows hypothetical profiles of potential density σ (defined at the j and $j+1$ points) as a function of the intracell, nondimensional vertical position ζ using parabolic reconstructions of Θ and S. Strictly for this figure, a constant reference pressure is assumed so that two parcels are "neutrally buoyant" if they have the same potential density. Panel (c) shows positions (denoted by Φ) of interfaces $n-1/2$ and $n+1/2$ (red lines with arrows) whose endpoints are "neutrally buoyant" according to that simplification. Panel (d) shows quantities associated with the sublayer n bounded by $n-1/2$ and $n+1/2$. The thicknesses of the sublayer h_j^n and h_{j+1}^n are shown as vertical red lines. The sublayer-averaged concentrations of a tracer C are used to calculate the diffusive flux within this sublayer F_n^γ.

sublayers whose interfaces are neutrally buoyant between pairs of columns. Section 3.3 describes how to find a neutral position within a cell. Lastly, section 3.4 details the calculation of sublayer fluxes and the application of the accumulated fluxes to update the model state.

To aid in our description of the new algorithm, consider two neighboring vertical-meridional columns as depicted in Figure 1a. Layers of the model are denoted by k and interfaces denoted using a 1/2 convention. Hereafter, to distinguish between the sublayers along which diffusive fluxes are calculated and the model's native layers, the latter will be referred to as a "cell." The left column is denoted by index j and right column by $j+1$ at the interface $j+1/2$. Cell thicknesses and tracer concentrations are given at j and $j+1$ and separated by a horizontal distance Δy. Model cell thicknesses are denoted by $h_{j,k}$.

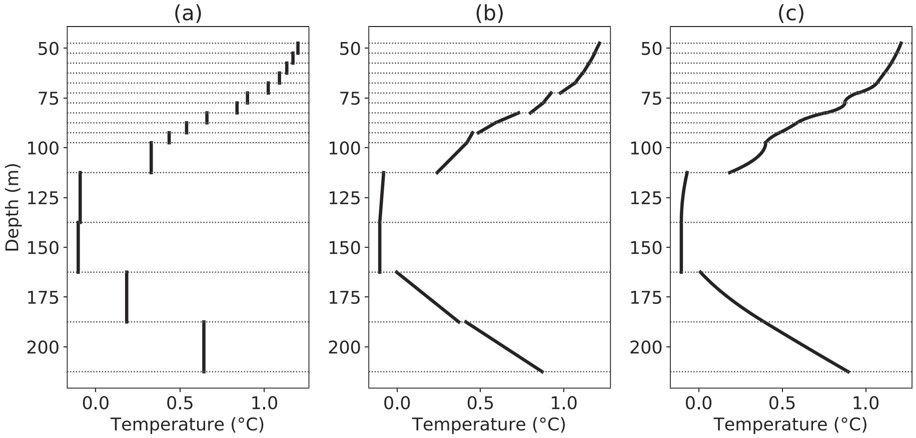

Figure 2. Reconstructions of intracell tracer profiles (lines) of temperature using piecewise constant (panel a), piecewise linear (panel b), and piecewise parabolic (panel c) methods. Horizontal dashed lines indicate the location of cell interfaces. Discontinuities at these interfaces in (b) and (c) arise when the polynomial coefficients need to be limited in order to satisfy the monotonicity constraint.

3.1. Reconstruction Phase

The representation of Θ and S in a finite-volume model is a set of discrete values, that is, the temperature of a cell is uniform across the depth range spanned by the cell's thickness. Finding a position of neutral buoyancy for an arbitrary water parcel is in general not possible except in the unlikely case that both the water parcel's and a layer's Θ, S, and P result in the neutrality condition, Equation 3, is met. For example, consider the case where the equation of state is only dependent on temperature and the modeled column has two cells with temperature 5°C and 10°C. A water parcel with temperature 8°C is neutrally buoyant with neither cell, even though its temperature falls within the range of modeled temperatures.

The purpose of the reconstruction phase described here creates vertical profiles of both Θ and S from their cell-averaged quantities. One approach (e.g., used in Groeskamp et al., 2019) is to assume that these quantities linearly vary between sampled points. In this algorithm for consistency between the reconstruction phases and the flux phase, we use high-order polynomial reconstructions to generate the vertical profiles.

The polynomial reconstruction method implemented in MOM6 (described in White & Adcroft, 2008) is conservative, monotonic, and extrema preserving. A reconstruction is conservative if the vertical average of the subcell profile returns the cell mean. Monotonicity means that there are no oscillations within the profile; that is, the profile increases or decreases monotonically between the cell boundaries. Extrema preserving means that at all points within a reconstruction fall within the range of tracer values spanned by the cell-mean values used in the reconstruction's stencil. The coefficients of the polynomial reconstructions may need to be limited to enforce the monotocity constraint. The limiting of the coefficients leads to discontinuities at the cell interfaces, though higher order polynomials can reduce the magnitude of these discontinuities (or remove them all together).

Any cells that represent a local extrema of the cell means are limited to a piecewise constant method (PCM) reconstruction to ensure no new-extrema are created as part of the reconstruction. Boundary cells are also typically limited to PCM. However, despite the potential for creating new extrema, we choose to extrapolate at the boundary cells in the test cases in this paper. The reason for doing so, as will be discussed, is that those boundary cells are essentially unstratified and would otherwise be effectively excluded from our new diffusion algorithm. The need to extrapolate at the boundary to ensure smooth solutions when remapping between grids is also noted in White and Adcroft (2008).

Figure 2 demonstrates these reconstructions, in order of increasing accuracy and polynomial degree, as applied to a temperature profile from the World Ocean Atlas 2018 (Locarnini et al., 2018) at a point (65°S, 120°W) in March. The piecewise constant reconstructions (Figure 2a) have discontinuities at every interface (dashed black lines) whereas the piecewise linear (Figure 2b) approach results in only six discontinuities (note that in MOM6, the slopes of these reconstructions are limited to ensure that the interface values are bounded by the cell means). The highest order scheme demonstrated here is the piecewise parabolic method

(Figure 2c) which further limits the interface discontinuties to just the portion of the water column with a temperature inversion. In both the linear and parabolic schemes, the extremum in temperature at 150 m results in a piecewise constant reconstruction for that cell.

For the reconstruction phase of the algorithm, the polynomial coefficients for all tracers, including Θ and S, are calculated and stored. MOM6 supports a hierarchy of polynomial-based reconstructions, but for ease of explaining the algorithm, piecewise linear interpolation is used in the algorithm description. For cases presented in section 4, piecewise parabolic reconstructions are used.

The following notation is used in this paper to distinguish quantities relating to the cell mean, interface values, or intracell reconstructions of an arbitrary tracer C. The vertical cell index k is 1 at the top of the column and increases with depth. Superscripts "top" and "bot" refer to the top and bottom interfaces of the cell. The polynomial reconstructions of degree d within a cell k are denoted as a function of the intracell coordinate ζ (that ranges from 0 at the top interface and 1 at the bottom) with the tracer concentration are written as

$$C_k(\zeta) = \sum_{m=0}^{d+1} c_m \zeta^m, \tag{11}$$

where c_m are the coefficients of the polynomial. Evaluating the reconstruction at a specific value of ζ yields the tracer's concentration for an infinitesimally thin layer of the cell. The average of the polynomial over a finite distance returns the average concentration over a finite thickness. Using this notation C_k^{top} refers to the concentration of the tracer at the top interface of the k cell of the model whereas $\overline{C_k}$ refers to the cell mean. The functional notation for the reconstructions means that $C_k(0) = C_k^{\mathrm{top}}$ and $C_k(1) = C_k^{\mathrm{bot}}$.

Using this notation, the neutrality condition (Equation 2) for a water parcel with a given Θ_1, S_1, and P_1 is satisfied within a cell j, k if there exists a position $\zeta_\gamma \in [0, 1]$ such that

$$\begin{aligned}\Delta\rho_{j,k}(\zeta_\gamma) = \frac{1}{2}\Big[&\left[\alpha\left(\Theta_1, S_1, \bar{P}(\zeta_\gamma)\right) + \alpha\left(\Theta_{j,k}(\zeta_\gamma), S_{j,k}(\zeta_\gamma), \bar{P}(\zeta_\gamma)\right)\right]\left[\Theta_1 - \Theta_{j,k}(\zeta_\gamma)\right] + \\ &\left[\beta\left(\Theta_1, S_1, \bar{P}(\zeta_\gamma)\right) + \beta\left(\Theta_{j,k}(\zeta_\gamma), S_{j,k}(\zeta_\gamma), \bar{P}(\zeta_\gamma)\right)\right]\left[S_1 - S_{j,k}(\zeta_\gamma)\right]\Big] = 0.\end{aligned} \tag{12}$$

Note that using reconstructions for Θ and S means that Equation 12 for an arbitrary water is a smoother function than if only the original finite-volume, cell-averaged quantities were used. This is illustrated in Figure 1b, where a constant reference pressure is used such that (Equation 12) simplifies to a difference in potential density σ (blue lines). While discontinuities exist at every interface, they are smaller in magnitude and more importantly the neighboring columns have finite regions where their density spaces overlap.

3.2. Search Phase

The search phase of the algorithm finds the position of neutral buoyancy for every cell interface in a pair of neighboring cells, that is, (using notation from the previous section), for an interface with a given Θ_1, S_1, and P_1, find the cell index k^γ and value ζ^γ within that cell such that $\Delta\rho_{k^\gamma}(\zeta^\gamma) = 0$. Because the polynomial reconstructions of Θ and S are discontinuous, a column with N cells has up to $2N$ cell interfaces with distinct values of Θ, S, and P. For a pair of columns, this means that a maximum of $4N$ positions may need to be found. These positions are then used to define subsurfaces of the model which form the interfaces of model sublayers along which diffusive fluxes are calculated.

One conceptually simple way of finding this k^γ and ζ^γ is to search every cell of the neighboring column for the position of neutral buoyancy for every interface value, resulting in $O(N^2)$ "searches." However, by assuming that columns are statically stable, the logic described in this section results in only $O(N)$ searches. To achieve this, we remove all cells whose interfaces are not dynamically stable from the remainder of the algorithm. In these cases, the neutral subsurfaces would cross and other physics (i.e., convection) are likely to dominate. The filtering stage of the algorithm also applies to cells that are unstratified. In this case, the top and bottom interfaces would have the same position of neutral buoyancy within the other column, meaning that the neutral sublayer spanned by the two interfaces would have zero thickness, and thus no exchange of tracer would occur.

We also make the choice to begin the "search algorithm" from the top cells of both column and search downward. In the presence of density inversions, multiple points of neutral buoyancy can exist meaning that searching downwards versus upwards can yield a different set of neutral sublayers. By searching downward, the shallower points of neutral buoyancy are preferred, resulting in more sublayers near the surface. Though subjective, this choice is motivated by the enhanced mesoscale turbulence in the upper ocean, thus prompting us to favor an accurate representation of neutral diffusion interactions in the upper ocean as opposed to the deeper ocean.

The following terminology and notation is used in the description of the search algorithm. To avoid subscripted indices and to aid the visual interpretation of the Figure 1, the quantities associated with the left cell j will be referred to with a subscript L (R is used similarly for the right cell $j+1$). The column containing the interface where the subsurface originates is referred to as the "pointed from" column and denoted by F. The other column being "searched" for a position of neutral buoyancy is referred to as being "pointed to" and denoted by T. The neutral subinterfaces of a sublayer n are referred to using 1/2 notation. The position in the column of the subinterface is denoted by the ordered pair $\Phi = [k^{*}, \zeta^{*}]$ where k^{*} refers to the vertical index of the cell where the surface connects and ζ^{*} is the nondimensional position within that cell. This notation also means that $S(\Phi)$ refers to the salinity within the cell k^{*} at a specific intracell location ζ^{*}.

We now describe the algorithm and logic involved in determining the positions of the neutral interfaces. Begin by examining the top interfaces of the top cell in each column (k_L, k_R). Two scenarios can occur in the first search stage:

1. The two interfaces are neutrally buoyant. In this case, the position of the neutral surface is $\Phi_L^{n-1/2} = [k_L, 0]$ and $\Phi_R^{n-1/2} = [k_R, 0]$.
2. One interface is lighter than the other interface. The cell with the lighter interface is the "pointed to" cell and the one with the denser interface is the "pointed from" cell.

In the second scenario, compare the top interface of the "pointed from" cell to the bottom interface of the "pointed to" cell (switching notation from L and R to T and F) and consider the following three outcomes in the second search stage:

a. The "pointed from" interface is neutrally buoyant with the bottom interface, then $\Phi_F^{n-1/2} = [k_F, 0]$ and $\Phi_T^{n-1/2} = [k_T, 1]$.
b. The "pointed from" top interface is lighter than the "pointed to" bottom interface. This implies that $\Delta\rho$ (Equation 2) between the "pointed from" top interface and the "pointed to" interfaces are of opposite sign. Therefore, there must be a position within the cell, $0 < \zeta_T < 1$, where the interface is neutrally buoyant. Section 3.3 details how to estimate this position. The position of this neutral surface is then $\Phi_F^{n+1/2} = [k_F, 0]$ and $\Phi_T^{n+1/2} = [k_T, \zeta_T]$.
c. The "pointed from interface" is denser than the "pointed to" bottom interface and thus denser than anything in the cell. No neutral surface can be found, so increment the cell index of the "pointed to" column, and iterate again to find the position of the $n-1/2$ interface (note if there are no cells remaining in both columns, the search phase of the algorithm completes).

For outcomes a and b, next compare the bottom interfaces. If the two are neutrally buoyant, then, the position of the neutral surface is $\Phi_L^{n+1/2} = [k_L, 1]$ and $\Phi_R^{n+1/2} = [k_R, 1]$. Increment k_L and k_R to the next stable cell in the respective columns. Otherwise, logic similar to the second search stage applies: find the lighter of the two bottom interfaces and find the position of neutral buoyancy in the other column.

Figure 1c shows the results of applying this logic to find two consecutive neutral surfaces. Strictly for the purpose of demonstrating the logic, Equation 2 is based solely on the difference in potential density σ (blue lines). Start with the top cells of each column k_L and k_R

- Iteration 1:
  1. Determine which of the top interfaces is denser. In this case, the top interface of k_R is denser than top interface of k_L. The top interface of k_R is designated the "pointed from" interface and k_L is the "pointed to" cell.
  2. Compare the "pointed from" interface to the bottom interface of the pointed to cell k_L. The top interface of k_R is denser than the bottom interface and is therefore denser than any point in cell k_L. No position of neutral buoyancy exists within that cell, so increment the cell index of the left column to $k_L + 1$ and search again.

- Iteration 2:
  1. Now compare the top interfaces of cells $k_L + 1$ and k_R. The top interface of k_R is the denser of the two and thus the "pointed from" interface and $k_L + 1$ is the "pointed to" cell.
  2. Compare the top interface of k_R to the bottom interface of k_L. The interface is still denser, so no position of neutral buoyancy can be found. Increment the cell index of the left column to $k_L + 2$ and search again.
- Iteration 3:
  1. The algorithm is now comparing the top interfaces of $k_L + 2$ and k_R. Again, the top interface of k_R is the denser one and thus the "pointed from" interface and $k_L + 2$ is to the "pointed to" cell.
  2. Compare the top interface of k_R to the bottom interface of $k_L + 2$. Now, the top interface of k_R is lighter of the two, so there is a position within cell $k_L + 2$ where the interface must be neutrally buoyant, denoted as ζ_1. This is the first neutral interface $\gamma_{-1/2}$ with $\Phi_L^{-1/2} = [k_L + 2, \zeta_1]$ and $\Phi_R^{-1/2} = [k_R, 0]$.
  3. Compare the bottom two interfaces of $k_L + 2$ and k_R and designate the denser one as the "pointed from" interface, that is, k_L.
  4. Comparing the bottom interface of k_L to the top interface of k_R shows that the former is lighter. Thus, somewhere within the k_R cell, there is a position ζ_2 where the bottom interface $k_L + 2$ will be neutrally buoyant. This is the second neutral interface $\gamma_{1/2}$ with position $\Phi_L^{1/2} = [k_L + 2,1]$ and $\Phi_R^{1/2} = [k_R, \zeta_2]$. These two interfaces, $\gamma_{-1/2}$ and $\gamma_{1/2}$, bound a sublayer of the model along which diffusive fluxes will be calculated.
  5. Because the bottom interface of $k_L + 2$ was the "pointed from" interface, increment the cell index to be searched to $k_L + 3$. No such cell exists, so the algorithm is now complete.

The search algorithm here essentially replaces the slope calculations required for rotated diffusion and defines a new vertically nonlocal space along which the diffusion operator is already aligned in the neutral direction. Because this is a crucial aspect of the overall algorithm, the sublayers and their interfaces for twelve oceanographically relevant scenarios (two of which are described in detail) are depicted in Appendix A.

3.3. Finding a Neutral Position within a Cell

When a neutral surface is denser than the top interface of a cell and lighter than the bottom interface, some form of interpolation must be done to find the position of neutral density for the subsurface within the cell denoted as ζ_γ. This is equivalent to satisfying the neutrality condition, $\Delta\rho_{j,k}(\zeta_\gamma) = 0$ (Equation 12) which is a potentially expensive operation due to calls to the equation of state. We consider three forms of the equation of state in order of decreasing accuracy and decreasing computational cost:

- Method 1: The original form where α and β are fully nonlinear.
- Method 2: Assume that α and β vary linearly across the cell being searched.
- Method 3: Assume that $\Delta\rho(\zeta)$ is a linear function.

Method 1 is potentially expensive because the neutrality condition (Equation 12) would generally require a recalculation of the density derivatives for every iteration of a root finding algorithm. For each of these three approaches, we discuss the form of Equation 12 and provide guidance on how to calculate the zero crossing (i.e., the position within a cell where the *searched from* interface is neutrally buoyant). For simplicity of notation, we define the following quantities where again F refers to the quantities defined at the searched from interface and T refers to the *pointed to* cell:

$$\bar{P}_{\text{top}} = \frac{P_T^{\text{top}} + P_F}{2}, \tag{13a}$$

$$\bar{P}_{\text{bot}} = \frac{P_T^{\text{bot}} + P_F}{2}, \tag{13b}$$

$$\alpha_T^{\text{top}} = \alpha(\Theta_T^{\text{top}}, S_T^{\text{top}}, \bar{P}_{\text{top}}), \tag{13c}$$

$$\beta_T^{\text{top}} = \beta(\Theta_T^{\text{top}}, S_T^{\text{top}}, \bar{P}_{\text{top}}), \tag{13d}$$

$$\alpha_T^{\text{bot}} = \alpha(\Theta_T^{\text{bot}}, S_T^{\text{bot}}, \bar{P}_{\text{bot}}), \tag{13e}$$

$$\beta_T^{\mathrm{bot}} = \beta(\Theta_T^{\mathrm{bot}}, S_T^{\mathrm{bot}}, \bar{P}_{\mathrm{bot}}), \tag{13f}$$

$$\alpha_F^{\mathrm{top}} = \alpha(\Theta_F, S_F, \bar{P}_{\mathrm{top}}), \tag{13g}$$

$$\beta_F^{\mathrm{top}} = \beta(\Theta_F, S_F, \bar{P}_{\mathrm{top}}), \tag{13h}$$

$$\alpha_F^{\mathrm{bot}} = \alpha(\Theta_F, S_F, \bar{P}_{\mathrm{bot}}), \tag{13i}$$

$$\beta_F^{\mathrm{bot}} = \beta(\Theta_F, S_F, \bar{P}_{\mathrm{bot}}). \tag{13j}$$

The difference between these three methods of finding the neutral position are demonstrated in section 4 for both linear and nonlinear equations of state.

Method 1 evaluates the neutrality condition (Equation 12) as written requiring that α and β are recalculated every iteration. This approach is the most accurate method, but it can be computationally expensive for a realistic equation of state.

Method 2 linearizes α and β within a grid cell according to

$$\alpha_T(\zeta) = (1-\zeta)\alpha_T^{\mathrm{top}} + \zeta\alpha_T^{\mathrm{bot}}, \tag{14a}$$

$$\beta_T(\zeta) = (1-\zeta)\beta_T^{\mathrm{top}} + \zeta\beta_T^{\mathrm{bot}}, \tag{14b}$$

$$\alpha_F(\zeta) = (1-\zeta)\alpha_F^{\mathrm{top}} + \zeta\alpha_F^{\mathrm{bot}}, \tag{14c}$$

$$\beta_F(\zeta) = (1-\zeta)\beta_F^{\mathrm{top}} + \zeta\beta_F^{\mathrm{bot}}. \tag{14d}$$

The $\alpha + \alpha_{j,k}$ and $\beta + \beta_{j,k}$ terms in Equation 12 can thus be expressed as a linear function of ζ,

$$\tilde{\alpha}(\zeta) = \tilde{\alpha}_{top} + (\tilde{\alpha}_{bot} - \tilde{\alpha}_{top})\zeta, \tag{15a}$$

$$\tilde{\beta}(\zeta) = \tilde{\alpha}_{top} + (\tilde{\alpha}_{bot} - \tilde{\alpha}_{top})\zeta, \tag{15b}$$

where

$$\tilde{\alpha}_{top} = \alpha_F^{\mathrm{top}} + \alpha_T^{\mathrm{top}}, \tag{16a}$$

$$\tilde{\alpha}_{bot} = \alpha_F^{\mathrm{bot}} + \alpha_T^{\mathrm{bot}}, \tag{16b}$$

$$\tilde{\beta}_{top} = \beta_F^{\mathrm{top}} + \beta_T^{\mathrm{top}}, \tag{16c}$$

$$\tilde{\beta}_{bot} = \beta_F^{\mathrm{bot}} + \beta_T^{\mathrm{bot}}. \tag{16d}$$

Substituting these expressions into Equation 12 yields a simpler expression for the neutrality condition

$$\Delta\rho(\zeta_\gamma) = \frac{1}{2}\left[\tilde{\alpha}(\zeta_\gamma)\left(\Theta_T(\zeta_\gamma) - \Theta_F\right) - \tilde{\beta}(\zeta_\gamma)\left(S_T(\zeta_\gamma) - S_F\right)\right] = 0. \tag{17}$$

Note here that α and β are both linear functions of ζ meaning that $\Delta\rho(\zeta)$ is a polynomial whose degree is one higher than the ones used for the reconstructions of Θ and S. Solutions for the roots of this equation can be found directly for cases where the reconstructions are linear or parabolic because Equation 12 becomes a quadratic and cubic equation respectively. If $\Theta(\zeta)$ and $S(\zeta)$ are linear, the equation is

$$0 = a_0 + a_1\zeta_\gamma + a_2\zeta_\gamma^2, \tag{18}$$

$$a_0 = (s_0 - s_F)\beta_{top} + (\theta_0 - \Theta_F)\alpha_{top}, \tag{19}$$

$$a_1 = s_F(\beta_{top} - \beta_{bot}) + s_0(\beta_{bot} - \beta_{top}) + s_1(\beta_{top}) + \Theta_F(\alpha_{top} - \alpha_{bot}) + \theta_0(\alpha_{bot} - \alpha_{top}) + \theta_1(\alpha_{top}), \tag{20}$$

$$a_2 = s_1(\beta_{bot} - \beta_{top}) + \theta_1(\alpha_{bot} - \alpha_{top}), \tag{21}$$

where we use s_n and θ_n are the coefficient of the nth term of the polynomial reconstruction of Θ and S.

Parabolic reconstructions result in a cubic equation

$$0 = a_0 + a_1\zeta_\gamma + a_2\zeta_\gamma^2 + a_3\zeta_\gamma^3, \tag{22}$$

$$a_0 = (s_0 - S_F)\beta_{top} + (\theta_0 - S_F)\alpha_{top}, \tag{23}$$

$$a_1 = S_F(\beta_{top} - \beta_{bot}) + s_0(\beta_{bot} - \beta_{top}) + s_1(\beta_{top}) + \Theta_F(\alpha_{top} - \alpha_{bot}) + \theta_0(\alpha_{bot} - \alpha_{top}) + \theta_1(\alpha_{top}), \tag{24}$$

$$a_2 = s_1(\beta_{bot} - \beta_{top}) + s_2\beta_{top} + \theta_1(\alpha_{bot} - \alpha_{top}) + \theta_2\alpha_{top}, \tag{25}$$

$$a_3 = s_2(\beta_{bot} - \beta_{top}) + \theta_2(\alpha_{bot} - \alpha_{top}). \tag{26}$$

This approximation has the added benefit of being computationally cheaper than Method 1 because α and β are not recalculated at every iteration.

For Methods 1 and 2, $\Delta\rho$ is not guaranteed to be monotonic, even with monotonic reconstructions of Θ and S and a linear equation of state. In Method 1, to ensure that the neutral interfaces never cross when multiple cells connect to a single cell, the zero crossing is restricted to lie between ζ of the last interface to connect within the cell and the bottom of the cell. In Method 2, all the zero crossings within the interval $[0, 1]$ can be calculated analytically. If there is more than one zero crossing, the one closest to the bottom of the cell is chosen. In both Method 1 and Method 2, if all zero crossings would intersect another sublayer, no neutral interface is stored and the algorithm iterates to the next cell.

Method 3 is equivalent to making the assumption that the density difference in Equation 12 is a linear function across the cell. Given the differences in density at the top and bottom of the cell, we have

$$\Delta\rho_{\mathrm{top}} = \frac{1}{2}\left[(\alpha_T^{\mathrm{top}} + \alpha_F^{\mathrm{top}})(\Theta_T^{\mathrm{top}} - \Theta_F) - (\beta_T^{\mathrm{top}} + \beta_F^{\mathrm{top}})(S_T^{\mathrm{top}} - S_F)\right], \tag{27a}$$

$$\Delta\rho_{\mathrm{bot}} = \frac{1}{2}\left[(\alpha_T^{\mathrm{bot}} + \alpha_F^{\mathrm{bot}})(\Theta_T^{\mathrm{bot}} - \Theta_F) - (\beta_T^{\mathrm{bot}} + \beta_F^{\mathrm{bot}})(S_T^{\mathrm{bot}} - S_F)\right], \tag{27b}$$

$$\Delta\rho(\zeta) = (1 - \zeta)\Delta\rho_{\mathrm{top}} + \zeta\Delta\rho_{\mathrm{bot}}. \tag{27c}$$

The density difference in Equation 12 is thus a linear equation in ζ so that the zero crossing is given by

$$\zeta_\gamma = \frac{\Delta\rho_{\mathrm{top}}}{\Delta\rho_{\mathrm{top}} - \Delta\rho_{\mathrm{bot}}}. \tag{28}$$

This is in general a poor approximation even for a linear equation of state because Equation 12 is only linear if the polynomial reconstructions of Θ and S are also linear.

3.4. Flux Calculation Step

The search phase described in the previous subsection defined the sublayers along which tracers are diffused. This subsection describes how the neutral diffusive fluxes of a generic tracer C are calculated and then aggregated to update the model state. Figure 1d serves as a visual reference for quantities described here.

Pairs of successive interfaces $\gamma_{n-1/2}$ and $\gamma_{n+1/2}$ define a sublayer (denoted by index n) along which tracers can be diffused (analogous to the layers of isopycnal models). The thickness of a sublayer within a cell, h_j^n, is calculated as the difference in depth between the locations of the neutral interfaces $\Phi_j^{n-1/2}$ and $\Phi_j^{n+1/2}$. Recalling that Φ describes both the cell index k^* and the intracell position ζ^* where the neutral surface connects within column j, the concentration of the tracer within the sublayer, $\overline{C_j^n}$, is calculated as the average of the polynomial reconstructions

$$\overline{C_j^n} = \frac{1}{\zeta_j^{n+1/2} - \zeta_j^{n-1/2}} \int_{\zeta_j^{n-1/2}}^{\zeta_j^{n+1/2}} C_{j,k^*}(\zeta)\, d\zeta. \tag{29}$$

The thickness of the sublayer at the $j+1/2$ interface is calculated as the harmonic mean of the two thicknesses (a second-order estimate)

$$h_{j+1/2}^n = 2\frac{h_j^n h_{j+1}^n}{h_j^n + h_{j+1}^n}. \tag{30}$$

This definition for the effective thickness results in zero diffusive flux whenever one side of a subcell has a vanishing thickness.

The thickness-weighted flux is calculated for sublayers where $h_{j+1/2}^n > 0$. The thickness weighted meridional neutral diffusive flux along this sublayer, F_n^γ, based on the discretized gradient is then

$$F_n^\gamma = -\kappa\, h_{j+1/2}^n \left[\frac{\overline{C_{j+1}^n} - \overline{C_j^n}}{\Delta y} \right], \tag{31}$$

where Δy is the horizontal distance between the centers of the neighboring cells. Notably, there is no vertical flux component in our scheme since it is based on a layered approach.

This notional flux must be limited to ensure that the flux is downgradient and thus not antidiffusive. The flux is guaranteed to be downgradient if all of the following differences are either of the same sign as Equation 31 or zero

- $C_{j+1}(\Phi_{j+1}^{n-1/2}) - C_j(\Phi_j^{n-1/2})$
- $C_{j+1}(\Phi_{j+1}^{n+1/2}) - C_j(\Phi_j^{n+1/2})$
- $\overline{C}_{j+1}^n - \overline{C_j^n}$
- $\overline{C_{j+1}^{k_R^*}} - \overline{C_j^{k_L^*}}$,

where $\overline{C_{j+1}^{k_R^*}}$ and $\overline{C_j^{k_L^*}}$ represent the concentration of the tracer over the entirety of the cell in which the sublayer connects. The last difference arises from the necessity that the prognostic concentrations used elsewhere in the model are assumed to be piecewise constant. Any higher order fluxes must thus be of the same sign to maintain discretization consistency.

The sublayer fluxes can now be accumulated to provide the total flux through the vertical face at the U-point. For a given vertical cell k and sublayer fluxes defined in Equation 31, the total flux is written as

$$F_j^k = \sum_l F_l^\gamma, \tag{32}$$

for all sublayers l that connect within cell k. Because the sublayer fluxes themselves are nonlocal, F_j^k represents the diffusive exchange of tracer from that individual grid cell to cells in the adjacent column. The tendency for the cell-averaged tracer concentration due to the meridional convergence of neutral diffusive fluxes is then given by

$$\left[\frac{\partial C_{j,k}}{\partial t} h_{j,k}\right]_{\text{merid}} = -\frac{1}{\Delta y}\left(F_{j+1/2}^k - F_{j-1/2}^k\right), \tag{33}$$

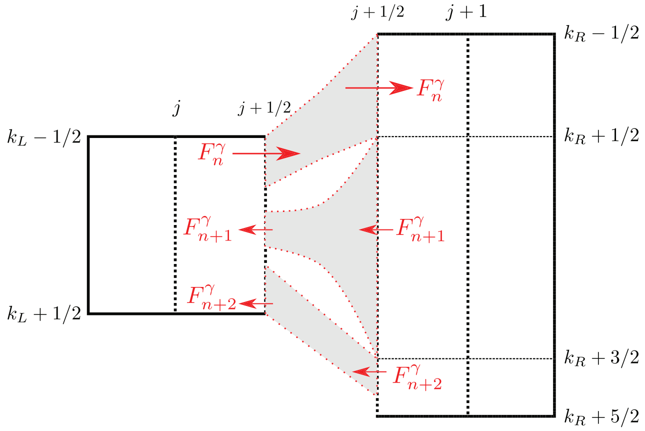

Figure 3. A hypothetical case where a single cell in the left column is neutrally connected to three cells in the neighboring column to demonstrate how sublayer fluxes, F^γ_n, F^γ_{n+1}, and F^γ_{n+2} fluxes are aggregated in the vertical. Arrows show the magnitude and direction of the flux with all other notation similar to those used in Figure 1. These cells are adjacent (in that they share the same velocity point); however, to show the sublayers more effectively, the two columns are separated visually. The distortion of the sublayer is used to emphasize that the thickness used to calculate the diffusive flux is based on harmonic means of the sublayer thicknesses.

with an analogous contribution from zonal fluxes. It is crucial to note that, unlike vertically local algorithms, $F^k_{j+1/2}$ does not apply to both adjacent thickness cells, j and $j+1$. Instead, $j-1/2$ and $j+1/2$ fluxes, accumulated from the sublayers, must be recalculated for each column individually.

Figure 3 is an example of how the diffusive fluxes are accumulated and applied. In this case, three sublayers (n, $n+1$, $n+2$) connect cell k_L on the left with cells (k_R, k_R+1, k_R+2) on the right. Discontinuities at the interfaces on the right column mean that there a gaps on the left cell in between the sublayers which will not diffuse with any cell in the right column. Fluxes $F^\gamma_n, F^\gamma_{n+1}, F^\gamma_{n+2}$ are calculated in each of these three sublayers. Using Equation 32, the total flux going into the left cell $F^{k_L}_{j+1/2}$ is the sum of all three sublayer fluxes

$$F^{k_L}_{j+1/2} = -\left(F^\gamma_n + F^\gamma_{n+1} + F^\gamma_{n+2}\right). \tag{34}$$

For each of the cells of the right column, only one sublayer flux applies

$$F^{k_R}_{j+1/2} = F^\gamma_n, \tag{35}$$

$$F^{k_R+1}_{j+1/2} = F^\gamma_{n+1}, \tag{36}$$

$$F^{k_R+2}_{j+1/2} = F^\gamma_{n+2}. \tag{37}$$

The resulting tendencies for each of the four cells written in terms of the sublayer fluxes are

$$\frac{\partial C_{j,k_L}}{\partial t} h_{j,k_L} = -\frac{1}{\Delta y}\left(F^\gamma_n + F^\gamma_{n+1} + F^\gamma_{n+2}\right), \tag{38}$$

$$\frac{\partial C_{j+1,k_R}}{\partial t} h_{j+1,k_R} = \frac{1}{\Delta y} F^\gamma_n, \tag{39}$$

$$\frac{\partial C_{j+1,k_R+1}}{\partial t} h_{j+1,k_R+1} = \frac{1}{\Delta y} F^\gamma_{n+1}, \tag{40}$$

$$\frac{\partial C_{j+1,k_R+2}}{\partial t} h_{j+1,k_R+2} = \frac{1}{\Delta y} F^\gamma_{n+2}. \tag{41}$$

This example demonstrates several key aspects of the algorithm:

- The entirety of a sublayer flux is applied to only one cell in each column.
- A flux can transfer tracer between two different layers of the model.
- Total flux through a cell face may not be related to the neighboring cell, that is, it is not strictly necessary that for $k_L = k_R$, $F^{k_L}_{j+1/2} = F^{k_R}_{j+1/2}$.
- The tracer contents of the j, k_L cell and $j+1, k_R+2$ cell can still change despite neutral sublayers not spanning the entire cell thicknesses.
- The vertically integrated, thickness-weighted change in tracer content in both columns are opposite in sign, but equal in magnitude, that is, the sum of the right hand sides of Equations 38–41 is zero.

4. Demonstration and Evaluation of the Algorithm

In this section, idealized test cases in MOM6 are used to demonstrate the behavior of the algorithm. No physical processes other than neutral diffusion are used and the fluid velocity is set to zero. The model domain is 200 km in the horizontal with 4 km spacing, and 200 m in the vertical. The initial temperature

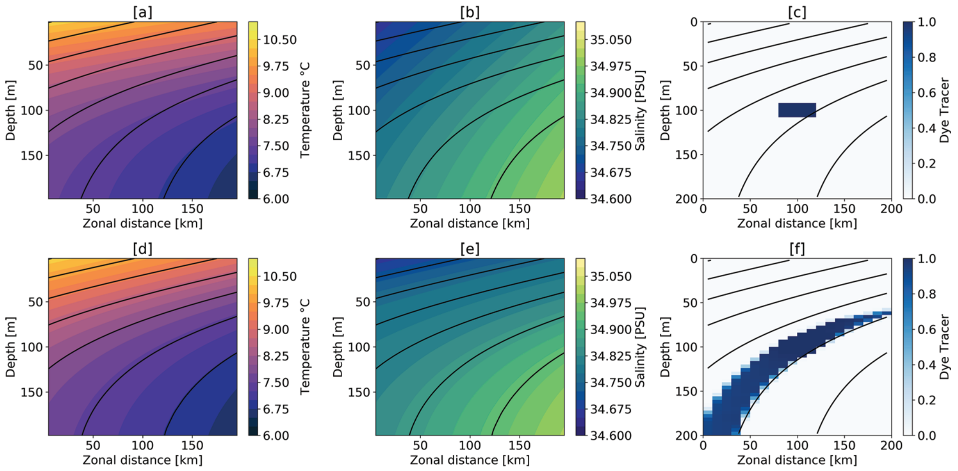

Figure 4. Temperature (a,d), salinity (b,e), and a passive dye tracer (c,f) at model initialization (top row) and 40 days (bottom row). Potential density contours are shown as solid black lines.

and salinity (Figures 4 and 5) represent a region stratified by both temperature and salinity. Both active tracers have a hyperbolic tangent profile and have nonlinear horizontal gradients. A passive dye tracer is also continuously injected in the center of the domain and diffused. Test cases are integrated in time for 40 days with a constant and uniform neutral diffusivity of $\kappa = 4{,}000\,\mathrm{m^2\,s^{-1}}$.

The neutral diffusion algorithm is used throughout the domain. However, in order to maintain a smooth field, tracer profiles are extrapolated at the boundary cells and this extrapolation can lead to new extrema. Section 5.3.2 discusses the implications in more detail. For this test case, a new maximum (a diminishing minimum) is created, but is negligibly small (an increase of 0.004°C) over the course of the 40-day simulation.

Three test cases are used to evaluate the algorithm. The first test case uses a linear equation of state with $\alpha = -0.2\,\mathrm{kg\,m^{-3}\,K^{-1}}$ and $\beta = 0.8\,\mathrm{kg\,m^{-3}}$ and uniform vertical spacing of 8 m on a z* grid. This case is used to describe the evolution of the active tracers and the passive dye tracer.

The second and third test cases both use the TEOS-10 nonlinear equation of state for seawater (IOC et al., 2010) but differ in their choice of vertical coordinate. One of these cases is discretized using the continuous isopycnal coordinate available in MOM6 and diffuses tracers along the model layers (i.e., isolayer diffusion). The other uses uniformly spaced z* levels and uses the neutral diffusion algorithm. The isopycnal test case serves as an independent verification of our diffusion method. However, in order to fairly compare the two, the z* case uses a fixed reference pressure of 2,000 dbar when evaluating the neutrality condition (Equation 12). As a full proof-of-concept demonstration of neutral diffusion, a variant of the z* case using the midpoint pressures is also discussed. The number of model levels is varied from between 5 and 200 to test the algorithm across a range of vertical grid spacing.

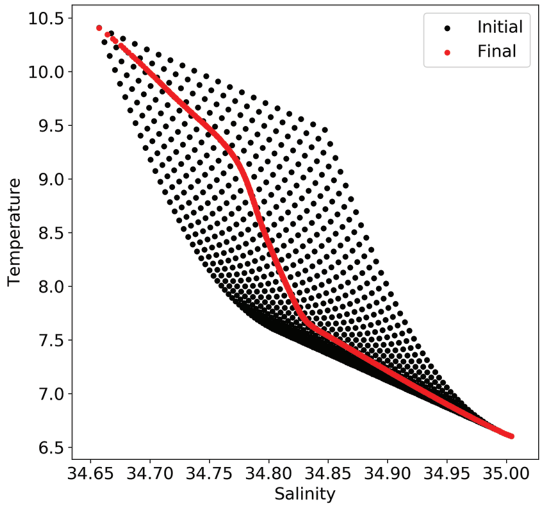

Figure 5. Temperature and salinity diagram for model state shown at initialization (black dots) and after 40 days of integration (red dots). Data are from the test case with a linear equation of state, though a visually identical result holds for other test cases with a nonlinear equation of state.

For the nonlinear equation of state, we assess how truly neutral this new algorithm is by inferring a vertical diffusivity based on a change in available potential energy (APE) (Ilicak et al., 2012; Winters et al., 1995). Due to the test cases' configurations (e.g., a resting fluid with flat topography), the difference in APE is strictly due to a change in stratification. A resulting change in energy per unit mass, ΔAPE, in our two-dimensional test case after one timestep can be calculated as

$$\Delta \mathrm{APE} = \frac{g\int_0^D [\rho(t_1,x,z) - \rho(t_0,x,z)]\, z\, dz\, dx}{\int_0^D \rho(t_0,x,z)\, dz}, \qquad (42)$$

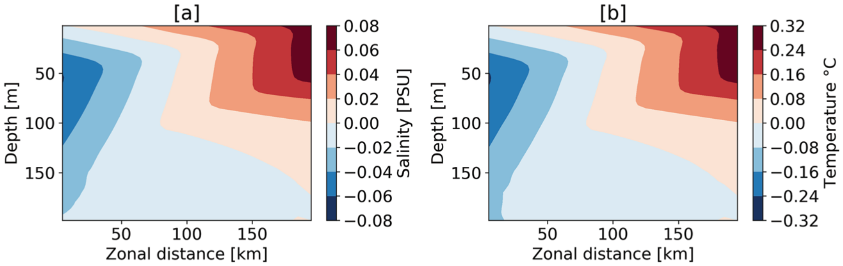

Figure 6. Difference between the final and initial salinity (panel a) and temperature (panel b) in the linear test case.

where g is the acceleration due to gravity (9.81 m s^{-2}), t_0 is at the beginning of a time step, t_1 is the end of a time step, and z is height above the flat-bottomed topography D. Assuming that any nonneutral mixing done by the algorithm decreases the stratification, the change in APE due to spurious processes $\Delta\text{APE}_{\text{spurious}}$ can be related to an effective nonneutral diffusivity κ_δ by

$$\frac{\Delta\text{APE}_{\text{spurious}}}{\Delta t} = \kappa_\delta N^2, \tag{43}$$

where Δt is the timestep of the model (3,600 s) and N is the buoyancy frequency ($\approx 3.3\times10^{-3}$ s^{-1} for these test cases). In both the isopycnal and z* cases, ΔAPE from Equation 42 is calculated after the first timestep of the model.

The spurious diffusivity diagnosed here is not necessarily limited to that of the neutral diffusion scheme itself. Any change in APE due to cabbeling or thermobaricity is also included. Additionally, roundoff errors during any part of the integration will be subsumed in this diagnostic. Nevertheless, this approach is preferred over that used by Groeskamp et al. (2019), which is based on the net buoyancy flux from the calculated salinity and temperature fluxes. Difficulty arises from that method because it is ambiguous what values for α and β to use with our sublayer approach. Namely, α and β could be calculated in at least two ways: from either the cell averaged or the sublayer averaged Θ, S, and P. Given a nonlinear equation of state, these two calculations cannot both simultaneously yield a nonzero diffusivity. It is thus unclear how to interpret the inferred spurious diffusivity. The energetic approached we use is a straightforward way of ensuring consistency between the two separate algorithms.

4.1. Linear Equation of State

A linear equation of state eliminates much of the difficulty with calculating neutral directions, with the relative simplicity helping to understand the more complex algorithm needed with a nonlinear equation of state. In addition, a linear equation of state can be found in a variety of idealized and semirealistic cases in which neutral diffusion may need to be applied.

The temperature and salinity at the beginning and end of the test case are show two different ways in Figure 4 and as a T-S diagram in Figure 5). In the beginning of the simulation, isohalines intersect isopycnals (Figure 4) whereas isotherms are almost aligned with the isopycnals. The passive dye tracer is confined to the interior of the domain and intersects one of the plotted isopycnals. In the final state of the model, the isohalines now lie parallel to the isopycnals of the model (which have not significantly changed). The isotherms do not change as perceptibly since they were nearly parallel to isopycnals from the start. As can be seen in Figure 5, Θ and S have collapsed onto a single line as a result of neutral diffusion. By the end of the simulation, the passive dye tracer has filled the entire spatial extent of its initial isopycnal range. As the domain becomes more strongly stratified near the top-right of the domain, the vertical extent of the dye thins. Near the bottom left of the domain, where the model is more weakly stratified, the dyed portion of the model is thicker.

The differences between initial and final distributions of Θ and S show the compensating buoyancy fluxes due to neutral diffusion. For example, those parts of the model domain that are saltier at the end of the model integration (red contours, left panel in Figure 6) also become warmer (red contours, right panel). Conversely, the fresher regions also become cooler. No evidence of grid-scale noise can be seen in this example. Domain inventories of Θ and S are within numerical roundoff.

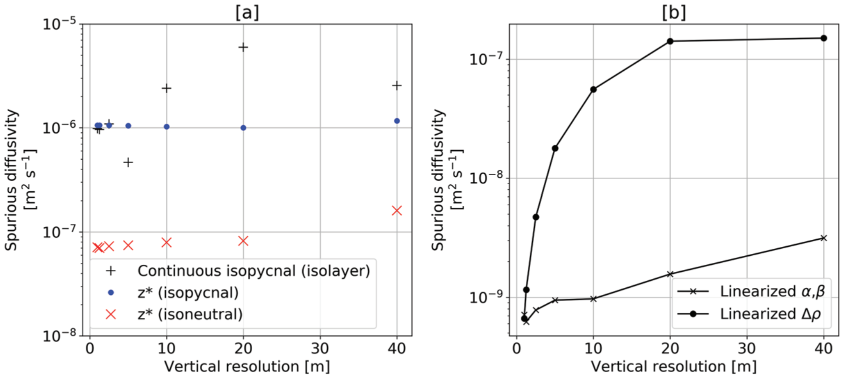

Figure 7. Spurious diffusivities when (panel a) using isolayer diffusion and a continuous isopycnal coordinate (black +), using a z* vertical coordinate and assuming a constant 2,000 dbar reference pressure in the new diffusion algorithm, i.e., isopycnal diffusion (blue dot), and the fully neutral, z* vertical case (red "x"). Panel (b) shows the additional spurious mixing when Equation 2 is assumed to vary linearly from the top and bottom interfaces (black line with circles) and when α and β vary linearly (black line with "x").

4.2. TEOS-10 Equation of State

In the TEOS-10 test-case, α and β are calculated from polynomial expressions that are fifth order in Θ and S. Evolution of Θ and S in this test case is similar to the linear test case. Hence, we focus on diagnosing spurious diffusivity and evaluating the different methods for finding the neutral position as described in section 3.3. The nominal resolution of the isopycnal test case is calculated as the average thickness of all nonzero isopycnal layers (Note: Plots of the beginning and end states of the test case have been omitted because they are visually indistinguishable from the linear test case).

Despite the two cases using different lateral diffusion schemes (i.e., our neutral diffusion versus isolayer diffusion) and vertical discretizations, the spurious diffusivity of both algorithms is within the same order of magnitude ($\approx 10^{-6}$ m^2 s^{-1}) (Figure 7). The continuous isopycnal case exhibits more variance in the diagnosed diffusivity as a function of resolution due to the difficulty of generating the model grid over a linear isopycnal spacing. Nevertheless, it serves as a reasonable independent verification that the diagnosed levels of spurious diffusivity in the z*, isopycnal diffusion case is reasonable. This metric seems to be insensitive to model resolution. The z* case using isoneutral diffusion exhibits a full order of magnitude less spurious diffusivity as compared to its isopycnal diffusion variant due to thermobaric effects being taken into account and weakly increases as the vertical resolution becomes coarser. We suspect that the latter result is likely due to increasingly larger differences between the parabolic reconstructions and the analytic hyperbolic tangent profiles of Θ and S as the resolution is coarsened.

The neutral diffusion test case is also used to compare the various methods for finding the position when a neutral surface connects in between two interfaces, as described in section 3.3 (Figure 7b). Method 1, as described in that section, is the most accurate of the three because of the recalculation of α and β every iteration. Excess spurious diffusivities are calculated as the difference between κ_δ of Method 1 and Method 2 (α and β vary linearly from the top and bottom of the cell) or Method 3 (the difference in density from Equation 12) (Figure 7b). While both Methods 2 and 3 have similar levels of excess diffusivity at the finest resolutions, Method 3 rapidly increases before asymptoting to $\approx 10^{-7}$. In contrast, excess diffusivities seen in Method 2 increase slowly as resolution decreases.

Based on these results, we recommend that Method 2, despite being less accurate than Method 1, should be adopted for most numerical models. When using the analytic solution to find the roots, it runs about 8 times more quickly than Method 1 for this test case and is comparable to the expense of Method 3.

5. Discussion

In this section, we emphasize the numerical benefits of our approach to neutral diffusion as compared to the widely used rotated tensor approach. We also briefly discuss some additional considerations when using the algorithm in realistic ocean configurations.

5.1. Absence of Slope Calculations

The new algorithm never estimates neutral slopes, hence, there is no need to provide a regularization of the neutral diffusive fluxes when the neutral slope is large. The diffusive fluxes in our algorithm are exactly the same even if one cell is at the top of its column and the other is at the bottom of its column. Indeed the only limit on the upper bound of the representable slope is one that arises from the ratio of the water column depth to the model's horizontal discretization.

In contrast, slope tapering methods must be used in the rotated operator approach for numerical stability in realistic settings where neutral slopes can steepen. Ocean models differ in the details of how slope tapering is implemented, and the resulting simulations can be quite sensitive to details, as illustrated in Gnanadesikan et al. (2007) and Ferrari et al. (2008). In general, the specifics of slope tapering are strongly tied to details of the numerics (e.g., time step and grid spacing). Dispensing with slope-tapering all together reduces complexity in understanding how the direction of horizontal mixing affects realistic and idealized simulations.

5.2. Rotated Operator Instability

Griffies et al. (1998) identified a growing instability associated with the Cox (1987) implementation of the rotated operator approach due to misalignment of the approximate slope with the true neutral direction. The algorithm developed in that study ameliorated this instability by calculating the diffusive fluxes of salinity and temperature to be explicitly neutrally buoyant. It is likely that any approach which calculates Θ and S fluxes separately, with no such thermodynamic constraints, also leads to nonzero buoyancy fluxes. This is true of both along-layer diffusion in isopycnal models as well as our neutral diffusion algorithm. Without incurring the expense of iteratively modifying the position of the sublayers, it is unclear how we could enforce an explicit zero-buoyancy flux condition.

Given that the algorithm here is monotonic and avoids slope limiters, we conjecture that these nonzero fluxes may be smaller than those of the original Cox (1987) scheme. The comparison of spurious diffusivities between the along-layer mixing of the isopycnal test case and our neutral diffusion suggests that these errors are smaller than those of isopycnal models. Nevertheless, it is possible that in very weakly stratified regions (e.g., sites of deep-water formation) that this imbalance could cause a column to destabilize. If these imbalances remained in the next timestep, the filtering portion of the search phase ensures that no fluxes would be calculated to or from the unstable portion of the water column, guaranteeing that the instability does not grow in magnitude due to neutral diffusion.

5.3. Practical Considerations in Realistic Configurations

A preliminary, lower-order implementation of this scheme (based on a continuous reconstruction of polynomials and a linearization of Equation 2) was included in the Geophysical Fluid Dynamics Laboratory's earth system model configuration (Adcroft et al., 2019). MOM6 does not have a rotated operator implementation of neutral diffusion. Hence, a comparison to previous methods is precluded pending implementation of the current algorithm into another ocean model with the rotated approach. Nevertheless, based on our initial implementation, we share some insights gained when applying this scheme in the global MOM6 configurations.

5.3.1. Computational Cost

The neutral diffusion algorithm scales linearly with the number of tracers. As long as the positions of the sublayers are stored, only the flux and reconstruction stages of the algorithm are dependent on the number of tracers.

The most expensive part of the algorithm is the searching phase. This is partly due to the branching logic utilized during the search phase of the algorithm which inhibits vectorization. The search phase also requires multiple calls to the EOS in order to calculate the differences in density. The worst case scenario arises when two columns are identical meaning that Equation 2 must be evaluated for all six pairs of interfaces. In this case, the algorithm would iterate down N times leading to $6N$ calls to the EOS. For comparison, the original Cox (1987) operator requires $4N$ calls for a pair of columns (two calls for each of the $2N$ cells to calculate the horizontal and vertical gradients of density). Reductions to the number of calls to the equation of state could be achieved by reusing the values α and β when interface values are continuous.

To assess the computational cost in a realistic configuration, we ran Method 2 (neutral position by linearization of α and β) of the neutral diffusion in a 10-day integration of the $\frac{1}{2}$-degree variant of MOM6 described by Adcroft et al. (2019). Of this, the neutral diffusion comprises 26% of the integration time, comparable to the 3-D tracer advection (horizontal advection, 29%, plus vertical regridding and remapping, 9%), but considerably more than the calculation of the Gent-McWilliams (GM) parameterized mass transports (1%) (Gent & McWilliams, 1990) (note that horizontal advection applies the net mass transport from all sources including GM). Method 3 (linearization of $\Delta\rho$) is slightly faster, comprising 22% of the total time, however as demonstrated in section 4 also results in slightly more spurious mixing.

5.3.2. Treatment of Boundary Cells

Another consideration is whether extrapolation should be used for the polynomial reconstructions in the boundary cells (i.e., the topmost and bottommost cells). Extrapolation generally results in smoother solutions (White & Adcroft, 2008), but raises may create new extrema. In the idealized test case shown here, the extrapolated values of Θ and S remained within physically realistic values. However, passive tracers can have sharp gradients near the surface, in which case extrapolation at the boundaries can yield unrealistic values. Extrapolation also represents an additional unconstrained boundary flux that may aggravate model biases. In these cases, we recommend against boundary extrapolation. Instead, we recommend reconstructing the top and bottom cells as piecewise constant to essentially isolate these cells from neutral diffusion fluxes. In this case, so long as the vertical resolution near the boundaries is fine enough, only a very small portion of the water column would be occluded from the neutral diffusion scheme.

The question of reconstructions within boundary cells is distinct from the question of the direction of eddy mixing Within the oceanic boundary layers. Treguier et al. (1997) and Ferrari et al. (2008) suggest that mesoscale eddy fluxes in these boundary layers are not oriented along the neutral direction, but are influenced by the shape of the surface and bottom. Studies that use the rotated tensor formulations of neutral diffusion taper the slopes near the surface so that fluxes smoothly tilt towards the horizontal (e.g., Danabasoglu et al., 2008; Griffies et al., 1998). Our approach has no slopes to be tapered accordingly. One way to incorporate this physical regime within our neutral diffusion scheme is to add an artificial compressibility such that Equation 2 transitions from being neutral below the boundary layer to fully pressure dependent within the boundary layer. Alternatively, if another scheme were to be implemented that handled this horizontal, diabatic mixing, the neutral diffusion algorithm could also be modified to eliminate any neutral sublayers within the boundary layer.

5.3.3. Necessity of Neutral Diffusion

Hybrid coordinates attempt to construct a vertical grid that employs the coordinate most appropriate to the local physics. Bleck (2002) describes an approach that is z-like near the surface boundary layer, isopycnal-like within the ocean interior, and terrain-following near the bottom boundary layer. This approach is generally followed for the hybrid coordinate within MOM6, where the model transitions from a z* coordinate near the surface to an isopycnic interior given a set of user-specified paramaters. Hofmeister et al. (2010) introduced a framework for an adaptive coordinate extending this idea of a hybrid coordinate to adjust to local shear and stratification. Given that along-layer diffusion would thus follow the geometrically constrained direction of eddy effects near boundaries and neutral directions within the interior, a natural question is whether a neutral diffusion operator is necessary.

Here we argue in favor of including our neutral diffusion algorithm when using a hybrid coordinate in a model like MOM6 that follows a regridding/remapping implementation of the arbitrary Lagrangian-Eulerian framework (Griffies et al., 2020). First, when using along-layer diffusion, the direction of diffusion depends on the point within the timestepping that neutral diffusive fluxes are calculated. In MOM6, layer thicknesses are updated after every physical process (e.g., dynamics and boundary fluxes). This means that unless neutral diffusion is calculated directly after a regridding/remapping step, diffusion will be calculated along an intermediate state of the layers. Second, hybrid coordinates are not generally available or even desirable across all configurations for which GCMs are used. Additionally, it can be crucially important to use the same operators, discretizations, and parameterizations when investigating the sensitivity of a configuration to the choice of vertical coordinate (e.g., the investigation of spurious heat uptake in Adcroft et al., 2019). Having a discretized operator that explicitly diffuses along neutral directions, thus remains important for GCMs.

5.3.4. Diagnostics

The calculation of neutral diffusive fluxes along nonlocal sublayers complicates the diagnostic analysis of diffusive tracer fluxes. Diffusive fluxes in local schemes are of equal magnitude but of opposite sign between adjacent cells. In contrast, the vertically nonlocal nature of our sublayer fluxes breaks this symmetry because tracer can be exchanged from any part of the water column. Reconstructing the 3-D tracer budget based on the fluxes through the faces of a cell requires 2N fluxes per cell, none of which are symmetric with their neighbors. However, column-integrated fluxes collapse the nonlocality distinction and so can be used and interpreted in the same manner as fluxes from local schemes. Furthermore, if diagnosing the grid cell tracer budget via tracer tendencies, the convergence of the neutral diffusive fluxes remains readily accessible.

5.3.5. Flux Limiters on Θ and S Fluxes

One last consideration is the potential need for an additional flux limiter. As described in section 3.4, some kind of gradient-based flux limiting is needed to enforce numerical consistency and avoid antidiffusive fluxes due to different levels of accuracy between the original model representation of tracers and the polynomial reconstructions of tracers. The need for limiters raises challenges for equations of state where both Θ and S are active tracers, but their respective neutral diffusive fluxes are computed separately. For example, in the case where a Θ flux in a sublayer was limited, but S was not, there would be a net buoyancy flux of $g\beta\kappa \nabla S$, likely manifesting itself as noise in the density field. To avoid this problem, another limiter could be applied to all tracer fluxes such that if either Θ or S is limited within a sublayer, all tracer fluxes are also limited. Such a limiter would be *ad hoc* because it is not based on physical or fundamental numerical arguments. In the test cases used here, Θ, S, and σ_2 all remain smooth in the absence of such a limiter. It is unclear how frequently this hypothetical limiter would need to be applied and what impacts it would have in realistic ocean configurations.

6. Summary

We have presented a new algorithm for diffusive transport of tracers along neutral directions that relies on well-established algorithms to create polynomial reconstructions of Θ and S. The method incorporates the idea of a vertically nonlocal search for neutral tangent plane as discussed by Groeskamp et al. (2019) and Jackett and McDougall (1997), with our work extending those ideas for use in a finite volume ocean model. The resulting discretized operator is fundamentally different from commonly used rotated operator methods such as Griffies et al. (1998). Instead of calculating neutral slopes as in the rotated operator approach, neutral surfaces are here found that connect neighboring columns. With appropriate flux-limiting, downgradient neutral diffusive fluxes of tracers are calculated along sublayers bounded by the neutral surfaces.

The resulting algorithm was demonstrated in an idealized, baroclinic zone test case using both linear and nonlinear equations of state. Emphasis was placed on diagnosing the spurious dianeutral mixing arising from the algorithm, particularly from different methods of calculating the position of a neutral surface. Spurious diffusivities ($\approx 10^{-6}$ m^2 s^{-1}) in a test case using our new algorithm with constant reference pressure were similar to those diagnosed when using an isopycnal coordinate and along-layer diffusion. The test case using fully neutral calculations exhibited spurious diffusivities ($\approx 10^{-7}$ m^2 s^{-1}), an order of magnitude or two smaller than the background levels used in realistic ocean configurations.

Some practical considerations when using the algorithm in realistic configurations were also discussed including computational cost, how the surface and bottom boundaries could be dealt with, and how to interpret the fluxes diagnostically. The new diffusion scheme is about 30% cheaper than the tracer advection portions of the code. Lastly, to aid those aiming to implement this algorithm, Appendix A contains schematics showing the position of the neutral surfaces for twelve oceanographically relevant cases.

Appendix A: Test Cases Demonstrating the Sorting Algorithm

Here we provide a number of cases with the expected results of the sorting algorithm (Figure A1). These cases assume that both columns have three layers of equal thickness and a linear equation of state dependent only on temperature. The intracell reconstructions are assumed to vary linearly from the given interface values (note: The numbers presented here are for demonstration purposes only and not a result of a formal PLM reconstruction). As before, these columns do share the same U-points, but are separated visually to see the shape of the sublayers. These cases are included to help readers understand the searching part of the algorithm and to test implementations of the algorithm.

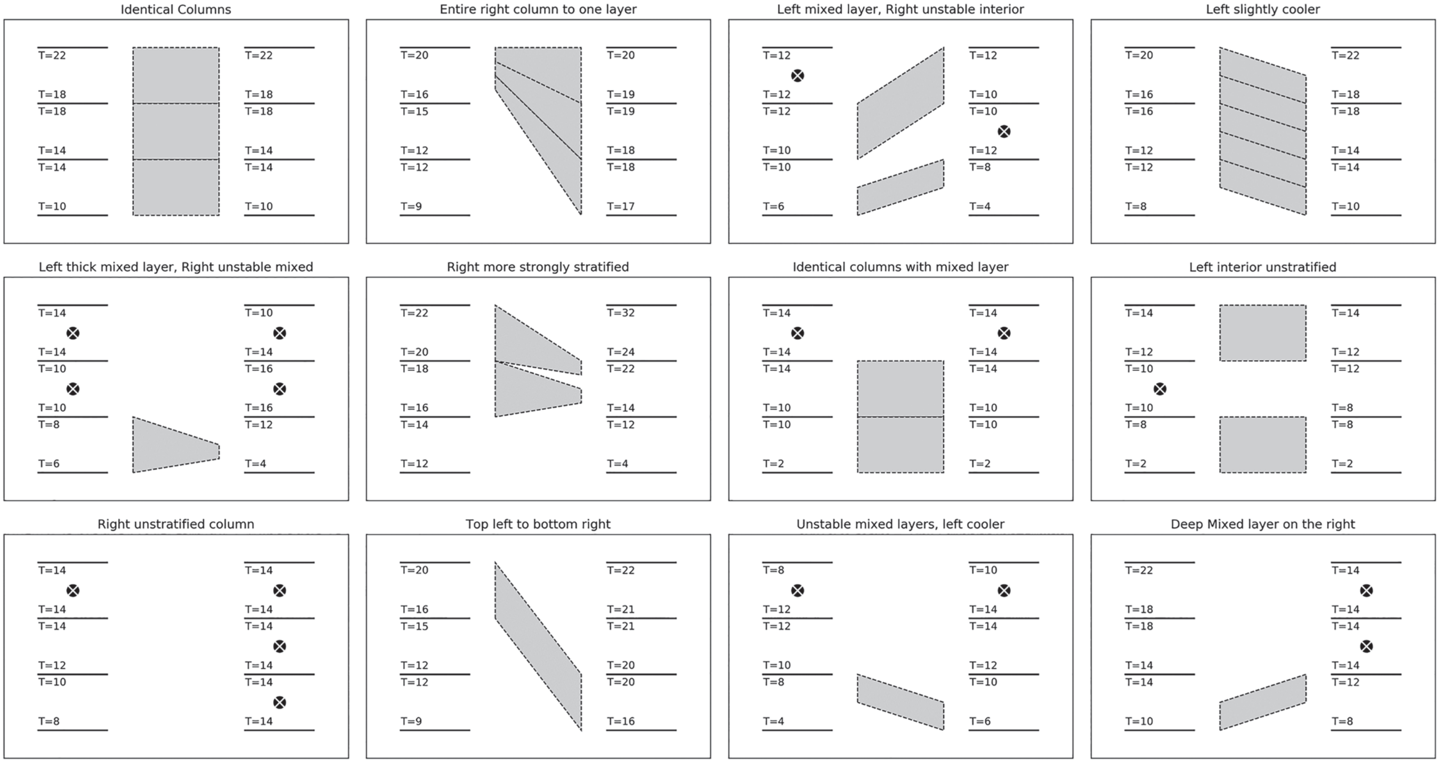

Figure A1. The neutral sublayers (filled gray polygons) found by the searching algorithm using a linear equation of state dependent only on temperature. Temperatures are shown at cell interfaces where multiple values can exist at the interface if the profiles are discontinuous. Numbers above the interface line correspond to value at the bottom interface of one cell and the number below the line indicates the top interface value of the next deeper cell. A circled "x" indicates that the layer is unstratified or unstably stratified.

Data Availability Statement

The idealized test cases used here can be found at https://doi.org/10.5281/zenodo.3911765 based on a Fortran implementation of the code in MOM6 found at https://doi.org/10.5281/zenodo.3911779.

Acknowledgments

We thank Sjoerd Groeskamp for discussions about their nonlocal observation-oriented approach in Groeskamp et al. (2019) and Bill Large for discussions on how to estimate the spurious diffusivity. We thank Brandon Reichl and Marshall Ward for providing feedback on an early draft of the manuscript. Lastly, we express our gratitude to George Nurser and two other anonymous reviewers for their constructive comments that greatly improved the manuscript. AS, AA, RH, and SG were supported under award NA18OAR4320123 from the National Oceanic and Atmospheric Administration, U.S. Department of Commerce. The statements, findings, conclusions, and recommendations are those of the authors and do not necessarily reflect the views of the National Oceanic and Atmospheric Administration, or the U.S. Department of Commerce.

References

Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., et al. (2019). The GFDL global ocean and sea ice model OM4.0: Model description and simulation features. *Journal of Advances in Modeling Earth Systems*, *11*, 3167–3211. https://doi.org/10.1029/2019MS001726

Beckers, J. M., Burchard, H., Campin, J. M., Deleersnijder, E., & Mathieu, P. P. (1998). Another reason why simple discretizations of rotated diffusion operators cause problems in ocean models: Comments on "isoneutral diffusion in a z-coordinate ocean model". *Journal of Physical Oceanography*, *28*(7), 1552–1559. https://doi.org/10.1175/1520-0485(1998)028<1552:ARWSDO>2.0.CO;2

Beckers, J.-M., Burchard, H., Deleersnijder, E., & Mathieu, P. P. (2000). Numerical discretization of rotated diffusion operators in ocean models. *Monthly Weather Review*, *128*(8), 2711–2733. https://doi.org/10.1175/1520-0493(2000)128<2711:NDORDO>2.0.CO;2

Bleck, R. (2002). An oceanic general circulation model framed in hybrid isopycnic-cartesian coordinates. *Ocean Modelling*, *4*, 55–88.

Bleck, R., & Smith, L. T. (1990). A wind-driven isopycnic coordinate model of the north and equatorial Atlantic Ocean: 1. Model development and supporting experiments. *Journal of Geophysical Research*, *95*(C3), 3273–3285. https://doi.org/10.1029/JC095iC03p03273

Cox, M. D. (1987). Isopycnal diffusion in a z-coordinate ocean model. *Ocean Modelling*, *74*, 1–5.

Danabasoglu, G., Ferrari, R., & McWilliams, J. C. (2008). Sensitivity of an ocean general circulation model to a parameterization of near-surface eddy fluxes. *Journal of Climate*, *21*, 1192–1208.

Dunne, J. P., John, J. G., Hallberg, R. W., Griffies, S. M., Shevliakova, E. N., Stouffer, R. J., et al. (2012). GFDLs ESM2 global coupled climate-carbon Earth System Models Part I: Physical formulation and baseline simulation characteristics. *Journal of Climate*, *25*, 6646–6665.

Ferrari, R., McWilliams, J. C., Canuto, V. M., & Dubovikov, M. (2008). Parameterization of eddy fluxes near oceanic boundaries. *Journal of Climate*, *21*(12), 2770–2789. https://doi.org/10.1175/2007JCLI1510.1

Gent, P. R., & McWilliams, J. C. (1990). Isopycnal mixing in ocean circulation models. *Journal of Physical Oceanography*, *20*, 150–155.

Gerdes, R., Köberle, C., & Willebrand, J. (1991). The influence of numerical advection schemes on the results of ocean general circulation models. *Climate Dynamics*, *5*(4), 211–226. https://doi.org/10.1007/BF00210006

Gnanadesikan, A. (1999a). A global model of silicon cycling: Sensitivity to eddy parameterization and dissolution. *Global Biogeochemical Cycles*, *13*(1), 199–220. https://doi.org/10.1029/1998GB900013

Gnanadesikan, A. (1999b). Numerical issues for coupling biological models with isopycnal mixing schemes. *Ocean Modelling*, *1*(1), 1–15. https://doi.org/10.1016/S1463-5003(99)00002-5

Gnanadesikan, A., Griffies, S. M., & Samuels, B. L. (2007). Effects in a climate model of slope tapering in neutral physics schemes. *Ocean Modelling*, *16*(1), 1–16. https://doi.org/10.1016/j.ocemod.2006.06.004

Gnanadesikan, A., Pradal, M.-A., & Abernathey, R. (2015). Isopycnal mixing by mesoscale eddies significantly impacts oceanic anthropogenic carbon uptake. *Geophysical Research Letters*, *42*, 4249–4255. https://doi.org/10.1002/2015GL064100

Gnanadesikan, A., Slater, R. D., Swathi, P. S., & Vallis, G. K. (2005). The energetics of ocean heat transport. *Journal of Climate*, *18*(14), 2604–2616. https://doi.org/10.1175/JCLI3436.1

Gregory, J. M. (2000). Vertical heat transports in the ocean and their effect on time-dependent climate change. *Climate Dynamics*, *16*(7), 501–515. https://doi.org/10.1007/s003820000059

Griffies, S. M. (2004). *Fundamentals of ocean climate models*. Princeton, USA: Princeton University Press. 518+xxxiv pages.

Griffies, S. M., Adcroft, A., & Hallberg, R. W. (2020). A primer on the vertical lagrangian-remap method in ocean models based on finite volume generalized vertical coordinates. *Journal of Advances in Modeling Earth Systems*, *12*, e2019MS001954. https://doi.org/10.1029/2019MS001954

Griffies, S. M., Gnanadesikan, A., Pacanowski, R. C., Larichev, V., Dukowicz, J. K., & Smith, R. D. (1998). Isoneutral diffusion in a *z*-coordinate ocean model. *Journal of Physical Oceanography*, *28*, 805–830.

Griffies, S. M., Harrison, M. J., Pacanowski, R. C., & Rosati, A. (2004). *A technical guide to MOM4* (p. 337). Princeton, USA: NOAA/Geophysical Fluid Dynamics Laboratory.

Griffies, S. M., Winton, M., Anderson, W. G., Benson, R., Delworth, T. L., Dufour, C., et al. (2015). Impacts on ocean heat from transient mesoscale eddies in a hierarchy of climate models. *Journal of Climate*, *28*, 952–977. https://doi.org/10.1175/JCLI-D-14-00353.1

Groeskamp, S., Barker, P. M., McDougall, T. J., Abernathey, R. P., & Griffies, S. M. (2019). VENM: An algorithm to accurately calculate neutral slopes and gradients. *Journal of Advances in Modeling Earth Systems*, *11*, 1917–1939. https://doi.org/10.1029/2019MS001613

Hallberg, R. (1995). Some aspects of the circulation in ocean basins with isopycnals intersecting the sloping boundaries (PhD Thesis). University of Washington Oceanography, Seattle, USA.

Hallberg, R. (2005). A thermobaric instability of Lagrangian vertical coordinate ocean models. *Ocean Modelling*, *8*(3), 279–300. https://doi.org/10.1016/j.ocemod.2004.01.001

Hieronymus, M., & Nycander, J. (2013). The budgets of heat and salinity in NEMO. *Ocean Modelling*, *67*, 28–38. https://doi.org/10.1016/j.ocemod.2013.03.006

Hofmeister, R., Burchard, H., & Beckers, J.-M. (2010). Non-uniform adaptive vertical grids for 3d numerical ocean models. *Ocean Modelling*, *33*(1), 70–86. https://doi.org/10.1016/j.ocemod.2009.12.003

IOC, SCOR, & IAPSO (2010). The international thermodynamic equation of seawater-2010: calculation and use of thermodynamic properties. Intergovernmental Oceanographic Commission, Manuals and Guides No. 56, UNESCO 196pp.

Ilicak, M., Adcroft, A. J., Griffies, S. M., & Hallberg, R. W. (2012). Spurious dianeutral mixing and the role of momentum dissipation. *Ocean Modelling*, *45–46*, 37–58. https://doi.org/10.1016/j.ocemod.2011.10.003

Iselin, C. O. D. (1939). The influence of vertical and lateral turbulence on the characteristics of the waters at mid-depth. *Eos, Transactions American Geophysical Union*, *3*, 414–417.

Iudicone, D., Madec, G., & McDougall, T. J. (2008). Water-mass transformations in a neutral density framework and the key role of light penetration. *Journal of Physical Oceanography*, *38*, 1357–1376.

Jackett, D. R., & McDougall, T. J. (1997). A neutral density variable for the world's oceans. *Journal of Physical Oceanography*, *27*(2), 237–263. https://doi.org/10.1175/1520-0485(1997)027<0237:ANDVFT>2.0.CO;2

Klocker, A., & McDougall, T. J. (2010a). Influence of the nonlinear equation of state on global estimates of dianeutral advection and diffusion. *Journal of Physical Oceanography*, *40*(8), 1690–1709. https://doi.org/10.1175/2010JPO4303.1

Klocker, A., & McDougall, T. J. (2010b). Quantifying the consequences of the ill-defined nature of neutral surfaces. *Journal of Physical Oceanography*, *40*(8), 1866–1880. https://doi.org/10.1175/2009JPO4212.1

Klocker, A., McDougall, T. J., & Jackett, D. R. (2009). A new method for forming approximately neutral surfaces. *Ocean Science*, *5*(2), 155–172. https://doi.org/10.5194/os-5-155-2009

Ledwell, J. R., Watson, A. J., & Law, C. S. (1993). Evidence for slow mixing across the pycnocline from an open-ocean tracer-release experiment. *Nature*, *364*, 701–703.

Lee, M.-M., & Nurser, A. J. G. (2012). Eddy subduction and the vertical transport streamfunction. *Journal of Physical Oceanography*, *42*(11), 1762–1780. https://doi.org/10.1175/JPO-D-11-0219.1

Lemarié, F., Debreu, L., Shchepetkin, A. F., & McWilliams, J. C. (2012). On the stability and accuracy of the harmonic and biharmonic isoneutral mixing operators in ocean models. *Ocean Modelling*, *52–53*, 9–35. https://doi.org/10.1016/j.ocemod.2012.04.007

Locarnini, R. A., Mishonov, A. V., Baranova, O. K., Boyer, T. P., Zweng, M. M., Garcia, H. E., et al. (2018). World ocean atlas 2018, volume 1: Temperature (*Report*).

McDougall, T. J. (1987a). Neutral surfaces. *Journal of Physical Oceanography*, *17*(11), 1950–1964. https://doi.org/10.1175/1520-0485(1987)017<1950:NS>2.0.CO;2

McDougall, T. J. (1987b). Thermobaricity, cabbeling, and water-mass conversion. *Journal of Geophysical Research*, *92*(C5), 5448–5464. https://doi.org/10.1029/JC092iC05p05448

McDougall, T. J., & Church, J. A. (1986). Pitfalls with the numerical representation of isopycnal diapycnal mixing. *Journal of Physical Oceanography*, *16*(1), 196–199. https://doi.org/10.1175/1520-0485(1986)016<0196:PWTNRO>2.0.CO;2

McDougall, T. J., & Jackett, D. R. (1988). On the helical nature of neutral trajectories in the ocean. *Progress in Oceanography*, *20*(3), 153–183. https://doi.org/10.1016/0079-6611(88)90001-8

Morrison, A. K., Saenko, O. A., Hogg, A. M., & Spence, P. (2013). The role of vertical eddy flux in Southern Ocean heat uptake. *Geophysical Research Letters*, *40*, 5445–5450. https://doi.org/10.1002/2013GL057706

Olbers, D. J., Wenzel, M., & Willebrand, J. (1985). The inference of North Atlantic circulation patterns from climatological hydrographic data. *Reviews of Geophysics*, *23*(4), 313–356. https://doi.org/10.1029/RG023i004p00313

Redi, M. H. (1982). Oceanic isopycnal mixing by coordinate rotation. *Journal of Physical Oceanography*, *12*, 1154–1158.

Sijp, W. P., Bates, M., & England, M. H. (2006). Can Isopycnal mixing control the stability of the thermohaline circulation in ocean climate models? *Journal of Climate*, *19*(21), 5637–5651. https://doi.org/10.1175/JCLI3890.1

Sijp, W. P., & England, M. H. (2009). The control of polar haloclines by along-isopycnal diffusion in climate models. *Journal of Climate*, *22*(3), 486–498. https://doi.org/10.1175/2008JCLI2513.1

Solomon, H. (1971). On the Representation of Isentropic Mixing in Ocean circulation models. *Journal of Physical Oceanography*, *1*(3), 233–234. https://doi.org/10.1175/1520-0485(1971)001<0233:OTROIM>2.0.CO;2

Treguier, A. M., Held, I. M., & Larichev, V. D. (1997). On the parameterization of quasi-geostrophic eddies in primitive equation ocean models. *Journal of Physical Oceanography*, *27*, 567–580.

White, L., & Adcroft, A. (2008). A high-order finite volume remapping scheme for nonuniform grids: The piecewise quartic method (PQM). *Journal of Computational Physics*, *227*(15), 7394–7422. https://doi.org/10.1016/j.jcp.2008.04.026

Winters, K. B., Lombard, P. N., Riley, J. J., & D'Asaro, E. A. (1995). Available potential energy and mixing in density-stratified fluids. *Journal of Fluid Mechanics*, *289*, 115–128. https://doi.org/10.1017/S002211209500125X

Wolfe, C. L., Cessi, P., McClean, J. L., & Maltrud, M. E. (2008). Vertical heat transport in eddying ocean models. *Geophysical Research Letters*, *35*, L23605. https://doi.org/10.1029/2008GL036138

Zika, J. D., Le Sommer, J., Dufour, C. O., Molines, J.-M., Barnier, B., Brasseur, P., et al. (2014). Vertical eddy fluxes in the southern ocean. *Journal of Physical Oceanography*, *43*, 941–955.