LAUR-02-2484

Reference Manual for the Parallel Ocean Program (POP)

Ocean Component of the Community Climate System Model (CCSM2.0)

Rick Smith and Peter Gent, Eds.

Los Alamos National Laboratory
Los Alamos, New Mexico

National Center for Atmospheric Research
Boulder, Colorado

May, 2002

Contents

1. Introduction

This manual describes the details of the numerical methods and discretization used in the Parallel Ocean Program (POP), a level-coordinate ocean general circulation model that solves the 3-dimensional primitive equations for ocean dynamics. It is designed for users who want an in-depth description of the numerics in the model, including details of the computational grid, the space and time discretization of the hydrodynamical core and subgridscale parameterizations, and other features of the model. Two other manuals associated with POP are also available: The *POP Users's Guide* (available online at www.acl.lanl.gov/climate) contains detailed instructions for setting up and running the POP code, including how to compile and run the code, and how to setup input files that specify the model configuration, diagnostics and ouput. A modified version of this manual is also available from NCAR as part of the Spring 2002 public release of CCSM2.0. The *POP Installation Guide* (also available online) shows how to set up POP on any computer system that supports Fortran 90, and, with multiprocessor systems, MPI or SHMEM.

The POP model is a descendant of the Bryan-Cox-Semtner class of models (Semtner, 1986). In the early 1990's it was written for the Connection Machine by Rick Smith, John Dukowicz and Bob Malone (Smith et al., 1992). An implicit free surface formulation and other numerical improvements were added by Dukowicz and Smith (1994). Later, the capability for general orthogonal coordinates for the horizontal mesh was implemented (Smith et al., 1995). Since then many new features and physics packages have been added by various people, including Dukowicz and Smith, Wooyoung Choi, Phil Jones, Bob Malone and Matt Maltrud, (all from Los Alamos National Laboratory (LANL)), and Frank Bryan, Gokhan Danabasoglu, Peter Gent, Matthew Hecht Bill Large, and Nancy Norton (from the National Center for Atmopheric Research (NCAR)). Over the last several years, Phil Jones has revamped and rewritten the model for improved portability and performance on a variety of machines and architectures, and substantially improved the user interface.

POP has a large base of users in the international community. In 2001, POP was officially adopted as the ocean component of the Community Climate System Model (CCSM), based at NCAR. Substantial effort at both LANL and NCAR has gone into adding various features to meet the needs of the CCSM coupled model. This manual includes descriptions of several of the features and options used in the ocean model configuration of the Spring 2002 release of CCSM2.0.

Acknowledgements. This manual was largely written by Rick Smith, edited by Peter Gent, with substantial contributions from several people for various sections: John Dukowicz (Equation of State, Boussinesq Approximation, Expansion Diagnostic), Phil Jones (Parameterizations of Vertical Mixing), Wooyoung Choi (Partial Bottom Cells, GM Parameterization), Peter Gent (GM Parameterization), Gokhan Danabasoglu (Forcing and Coupling Issues, KPP), Bill Large (KPP), Frank Bryan (Equation of State), and Matthew Hecht (Third-Order Upwind Advection). The work at LANL has been supported by the Department of Energy through the CHAMMP Program and later the Climate Change Prediction Program. The work at NCAR has been supported by the National Science Foundation.

Send comments, typos or other errors to rdsmith@lanl.gov.

2. The Primitive Equations in General Coordinates

Ocean dynamics are described by the 3-D primitive equations for a thin stratified fluid using the hydrostatic and Boussinesq approximations. Before deriving the equations in general coordinates, we first present, as a reference point, the continuous equations in spherical polar coordinates with vertical z-coordinate (these are standard in Bryan-Cox models, see, for example, Semtner 1986 and Pacanowski and Griffies, 2000).

momentum equations:

$$\frac{\partial}{\partial t}u + \mathcal{L}(u) - (uv\tan\phi)/a - fv = -\frac{1}{\rho_0 a\cos\phi}\frac{\partial p}{\partial \lambda} + \mathcal{F}_{Hx}(u,v) + \mathcal{F}_V(u) \tag{1}$$

$$\frac{\partial}{\partial t}v + \mathcal{L}(v) + (u^2\tan\phi)/a + fu = -\frac{1}{\rho_0 a}\frac{\partial p}{\partial \phi} + \mathcal{F}_{Hy}(u,v) + \mathcal{F}_V(v) \tag{2}$$

$$\mathcal{L}(\alpha) = \frac{1}{a\cos\phi}\left[\frac{\partial}{\partial\lambda}(u\alpha) + \frac{\partial}{\partial\phi}(\cos\phi\; v\alpha)\right] + \frac{\partial}{\partial z}(w\alpha) \tag{3}$$

$$\mathcal{F}_{Hx}(u,v) = A_M\left\{\nabla^2 u + u(1-\tan^2\phi)/a^2 - \frac{2\sin\phi}{a^2\cos^2\phi}\frac{\partial v}{\partial\lambda}\right\} \tag{4}$$

$$\mathcal{F}_{Hy}(u,v) = A_M\left\{\nabla^2 v + v(1-\tan^2\phi)/a^2 + \frac{2\sin\phi}{a^2\cos^2\phi}\frac{\partial u}{\partial\lambda}\right\} \tag{5}$$

$$\nabla^2\alpha = \frac{1}{a^2\cos^2\phi}\frac{\partial^2\alpha}{\partial\lambda^2} + \frac{1}{a^2\cos\phi}\frac{\partial}{\partial\phi}\left(\cos\phi\frac{\partial\alpha}{\partial\phi}\right) \tag{6}$$

$$\mathcal{F}_V(\alpha) = \frac{\partial}{\partial z}\mu\frac{\partial}{\partial z}\alpha \tag{7}$$

continuity equation:

$$\mathcal{L}(1) = 0 \tag{8}$$

hydrostatic equation:

$$\frac{\partial p}{\partial z} = -\rho g \tag{9}$$

equation of state:

$$\rho = \rho(\Theta, S, p) \;\rightarrow\; \rho(\Theta, S, z) \tag{10}$$

tracer transport:

$$\frac{\partial}{\partial t}\varphi + \mathcal{L}(\varphi) = \mathcal{D}_H(\varphi) + \mathcal{D}_V(\varphi) \tag{11}$$

$$\mathcal{D}_H(\varphi) = A_H\nabla^2\varphi \tag{12}$$

$$\mathcal{D}_V(\varphi) = \frac{\partial}{\partial z}\kappa\frac{\partial}{\partial z}\varphi\;, \tag{13}$$

where λ, ϕ, $z = r - a$ are longitude, latitude, and depth relative to mean sea level $r = a$; g is the acceleration due to gravity, $f = 2\Omega \sin\phi$ is the Coriolis parameter, and ρ_0 is the background density of seawater. The prognostic variables in these equations are the eastward and northward velocity components (u, v), the vertical velocity w, the pressure p, the density ρ, and the potential temperature Θ and salinity S. In (11) φ represents Θ, S or a passive tracer. The pressure dependence of the equation of state is usually approximated to be a function of depth only (see Sec. 3.5). A_H and A_M are the coefficients (here assumed to be spatially constant) for horizontal diffusion and viscosity, respectively, and κ and μ are the corresponding vertical mixing coefficients which typically depend on the local Richardson number (Pacanowski and Philander, 1981). The third terms on the left-hand side in Eqs. (1), (2) are metric terms due to the convective derivatives in $d\mathbf{u}/dt$ acting on the unit vectors in the λ, ϕ directions, and the second and third terms in brackets in Eqs. (4), (5) ensure that no stresses are generated due to solid-body rotation (Williams, 1972). The forcing terms due to wind stress and heat and fresh water fluxes are applied as surface boundary conditions to the friction and diffusive terms $\mathcal{F}_V$ and $\mathcal{D}_V$. The bottom and lateral boundary conditions applied in POP (and in most other Bryan-Cox models), are no-flux for tracers (zero tracer gradient normal to boundaries), and no-slip for velocities (both components of velocity zero on bottom and lateral boundaries).

To derive the primitive equations in general coordinates, consider the transformation from Cartesian coordinates (ξ_1, ξ_2, ξ_3 with origin at the center of the Earth) to general horizontal coordinates (q_x, q_y, z), where q_x and q_y are arbitrary curvilinear coordinates in the horizontal directions, and $z = r - a$ is again the vertical coordinate normal to the surface of the sphere. The actual distances along the curvilinear coordinates are denoted x and y, which typically lie along the circumpolar (longitude-like) and azimuthal (latitude-like) coordinate lines, respectively, on dipole grids with two arbitrarily located poles (see Sec. 7). The differential length element ds is given by

$$ds^2 = d\xi_1^2 + d\xi_2^2 + d\xi_3^2 = h_{ij}^2 dq_i dq_j + dz^2 \tag{14}$$

$$h_{ij}^2 = \frac{\partial \xi_k}{\partial q_i} \frac{\partial \xi_k}{\partial q_j} , \tag{15}$$

(where $i, j = x, y$ and repeated indices are summed). The metric coefficients h_{ij} depend on the local curvature of the coordinates. Differential lengths in the z direction are assumed independent of x and y, so no metric coefficients involving z appear. Further restricting ourselves to orthogonal grids, the cross terms vanish, and we have

$$\begin{aligned} h_i &\equiv h_{ii} , \quad h_{ij} = 0 \; (i \neq j) \\ ds_i &= h_i dq_i . \end{aligned} \tag{16}$$

For the purpose of constructing horizontal finite-difference operators corresponding to the various terms in the primitive equations, define:

$$\begin{aligned} \Delta_i &\equiv ds_i \\ \delta_i &\equiv \frac{\partial}{\partial s_i} = \frac{1}{h_i} \frac{\partial}{\partial q_i} , \end{aligned} \tag{17}$$

where Δ_i and δ_i can be interpreted as either infinitesimal or finite differences and their associated derivatives. Formulas for the basic horizontal operators (gradient, divergence, curl) can be found in standard textbooks (e.g., Arfken, 1970). The gradient is

$$\begin{aligned} \nabla\psi &= \hat{\mathbf{x}}\frac{1}{h_x}\frac{\partial\psi}{\partial q_x} + \hat{\mathbf{y}}\frac{1}{h_y}\frac{\partial\psi}{\partial q_y} \\ &= \hat{\mathbf{x}}\delta_x\psi + \hat{\mathbf{y}}\delta_y\psi \ . \end{aligned} \tag{18}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are unit vectors in the x, y directions. The horizontal divergence is:

$$\begin{aligned} \nabla\cdot\mathbf{u} &= \frac{1}{h_x h_y}\frac{\partial}{\partial q_x}(h_y u_x) + \frac{1}{h_x h_y}\frac{\partial}{\partial q_y}(h_x u_y) \\ &= \frac{1}{\Delta_y}\delta_x(\Delta_y u_x) + \frac{1}{\Delta_x}\delta_y(\Delta_x u_y) \ , \end{aligned} \tag{19}$$

where u_x and u_y are the velocity components along the x and y directions. The advection operator (Eq. 3) is similar:

$$\mathcal{L}(\alpha) = \frac{1}{\Delta_y}\delta_x(\Delta_y u_x\alpha) + \frac{1}{\Delta_x}\delta_y(\Delta_x u_y\alpha) + \delta_z(w\alpha) \ . \tag{20}$$

The vertical component of the curl operator is

$$\begin{aligned} \hat{\mathbf{z}}\cdot\nabla\times\mathbf{u} &= \frac{1}{h_x h_y}\frac{\partial}{\partial q_x}(h_y u_y) - \frac{1}{h_x h_y}\frac{\partial}{\partial q_y}(h_x u_x) \\ &= \frac{1}{\Delta_y}\delta_x(\Delta_y u_y) - \frac{1}{\Delta_x}\delta_y(\Delta_x u_x) \ . \end{aligned} \tag{21}$$

Laplacian-type operators, which appear in the viscous and diffusive terms, have the form

$$\nabla\cdot G\nabla\psi = \frac{1}{\Delta_y}\delta_x(\Delta_y G\delta_x\psi) + \frac{1}{\Delta_x}\delta_y(\Delta_x G\delta_y\psi) \ . \tag{22}$$

where G is an arbitrary scalar function of x and y. Note that these oprators have been expressed in terms of the differences and derivatives Δ_i and δ_i, and hence there is no explicit dependence on the new coordinates q_i or the metric coefficients h_i. In the discrete operators the same is true: it is not necessary to have an analytic transformation with metric coefficients describing the new coordinate system, it is only necessary to know the location of the discrete grid points and the distances between neighboring gridpoints along the coordinate directions.

The other horizontal finite difference operators appearing in the primitive equations can also be derived in general coordinates. The Coriolis terms are simply given by

$$2\Omega\times\mathbf{u} = -\hat{\mathbf{x}}fu_y + \hat{\mathbf{y}}fu_x \ . \tag{23}$$

The metric momentum advection terms corresponding to the third terms on the left in Eqs. (1,2) are given by (Haltiner and Williams, 1980, p. 442):

$$(uv\tan\phi)/a \rightarrow u_x u_y k_y - u_y^2 k_x \tag{24}$$

$$(u^2\tan\phi)/a \rightarrow u_x u_y k_x - u_x^2 k_y \tag{25}$$

$$k_x \equiv \frac{1}{h_x h_y}\frac{\partial}{\partial q_x}h_y = \frac{1}{\Delta_y}\delta_x\Delta_y \tag{26}$$

$$k_y \equiv \frac{1}{h_x h_y}\frac{\partial}{\partial q_y}h_x = \frac{1}{\Delta_x}\delta_y\Delta_x \tag{27}$$

Note that these revert to the standard forms (left of arrows) in spherical polar coordinates, where $h_x = a\cos\phi$, $h_y = a$, $u_x = u$ and $u_y = v$. The metric terms in the viscous operators (second and third terms on the right in Eqs. (4), (5) require a more careful treatment. These terms were derived by Williams (1972) in spherical coordinates, by applying the thin-shell approximation ($r \rightarrow a$) to the viscous terms expressed as the divergence of a stress tensor whose components are linearly proportional to the components of the rate-of-strain tensor. This form is transversely isotropic and ensures that for solid rotation the fluid is stress-free. The general coordinate versions of these terms are derived in Smith et al. (1995). The results are

$$\begin{aligned}\mathcal{F}_{Hx}(u_x,u_y) \;=\;& A_M\{\nabla^2 u_x - u_x(\delta_x k_x + \delta_y k_y + 2k_x^2 + 2k_y^2) \\ &+ u_y(\delta_x k_y - \delta_y k_x) + 2k_y(\delta_x u_y) - 2k_x(\delta_y u_y)\}\end{aligned} \tag{28}$$

The formula for $\mathcal{F}_{Hy}(u_x,u_y)$ is the same with x and y interchanged everywhere on the r.h.s. It is straightforward to show that these also reduce to the correct form in the spherical polar limit (Eqs. 4, 5). The above forms assume a spatially constant viscosity A_M. More terms appear if A_M is allowed to vary spatially. Wajsowicz (1993) derives the extra terms for spherical polar coordinates. In general orthogonal coordinates they take the form:

$$\begin{aligned}\mathcal{F}_{Hx}(u_x,u_y) \;=\;& \nabla\cdot A_M\nabla u_x - u_x\{\delta_x A_M k_x + \delta_y A_M k_y + 2A_M(k_x^2 + k_y^2)\} \\ &+ u_y(\delta_x A_M k_y - \delta_y A_M k_x) \\ &+ (2A_M k_y + \delta_y A_M)(\delta_x u_y) - (2A_M k_x + \delta_x A_M)(\delta_y u_y)\ .\end{aligned} \tag{29}$$

The formula for $\mathcal{F}_{Hy}(u_x,u_y)$ is again the same with x and y interchanged everywhere.

The general coordinate forms of the anisotropic and biharmonic viscous operators are given in Sec. 5.2 below, and the Gent-McWilliams and biharmonic forms of the horizontal diffusion terms are given in Sec. 5.1.

3. Spatial Discretization

3.1. Discrete horizontal and vertical grids

The placement of model variables on the horizontal B-grid is illustrated in Figure 1. The solid lines enclose a "T-cell" and the hatched lines enclose a "U-cell". Scalars (T, S, p, ρ) are located at

"T-points" (dots) at the centers of T-cells, and horizontal vectors (u_x, v_x) are located at "U-points" (crosses) at the corners of T-cells. The indexing for points (i, j) in the logically-rectangular 2-D horizontal grid is such that i increases in the x direction (eastward for spherical polar coordinates), and j increases in the y direction (northward for spherical polar coordinates). A U-point with logical indices (i, j) lies to the upper right ($\sim$ northeast) of the T-point with the same indices (i, j). The index for the vertical dimension k increases with depth, although the vertical coordinate z, measured from the mean surface level $z = 0$, decreases with depth.

When the horizontal grid is generated, the latitude and longitude of each U-point and the distances HTN and HTE (see Fig. 1) along the coordinates between adjacent U-points are first constructed. Then the latitude and longitude of T-points are computed as the straight average of the latitude and longitude of the four surrounding U-points, and the along-coordinate distances HUW (HUS) between adjacent T-points are computed as the straight average of the four surrounding values of HTE (HTN). Thus T-points are located exactly in the middle of the T-cell, but because the grid spacing in either direction may be non-uniform, the U-points are not located exactly in the middle of the U-cell.

In addition to the grid spacings HTN, HTE, HUS, HUW, several other lengths and areas are also used in the code. These are defined as follows (see Fig. 1):

$$
\begin{aligned}
\mathrm{DXU}_{i,j} &= 0.5[\mathrm{HTN}_{i,j} + \mathrm{HTN}_{i+1,j}] \\
\mathrm{DYU}_{i,j} &= 0.5[\mathrm{HTE}_{i,j} + \mathrm{HTE}_{i,j+1}] \\
\mathrm{DXT}_{i,j} &= 0.5[\mathrm{HTN}_{i,j} + \mathrm{HTN}_{i,j-1}] \\
\mathrm{DYT}_{i,j} &= 0.5[\mathrm{HTE}_{i,j} + \mathrm{HTE}_{i-1,j}] \\
\mathrm{UAREA}_{i,j} &= \mathrm{DXU}_{i,j}\mathrm{DYU}_{i,j} \\
\mathrm{TAREA}_{i,j} &= \mathrm{DXT}_{i,j}\mathrm{DYT}_{i,j} \qquad (30)
\end{aligned}
$$

DXU and DYU are the grid lengths centered on U-points, and DXT and DYT are centered on T-points. TAREA and UAREA are the horizontal areas of T-cells and U-cells, respectively.

The construction of the semi-analytic dipole and tripole grids commonly used in POP is described in detail in Smith et al. 1995, and is briefly reviewed in Sec. 7. These grids are based on an underlying orthogonal curvilinear coordinate system with the one or two singularities in the northern hemisphere displaced into a land masses (typically North America, Asia, or Greenland). The equator is usually retained as a grid line and the southern hemisphere is a standard spherical polar grid with the southern grid pole located at the true South Pole. These grids are topologically equivalent to a cylinder (periodic in x but not in y), and therefore can be mapped onto a logically-rectangular 2-D array (i, j) which is cyclic in i. Tripole grids require additional comunication along the northern boundary of the grid in order to "sew up" the grid along the line between the two northern grid poles. The grids are constructed off line and a file is generated which contains the following 2-D fields: ULAT,ULONG,HTN,HTE,HUS,HUW,ANGLE, where $\mathrm{ULAT}_{i,j}$ and $\mathrm{ULONG}_{i,j}$ are the true latitude and longitude of U-points, and $\mathrm{ANGLE}_{i,j}$ is the angle between the x-direction and true east at the U-point (i, j).

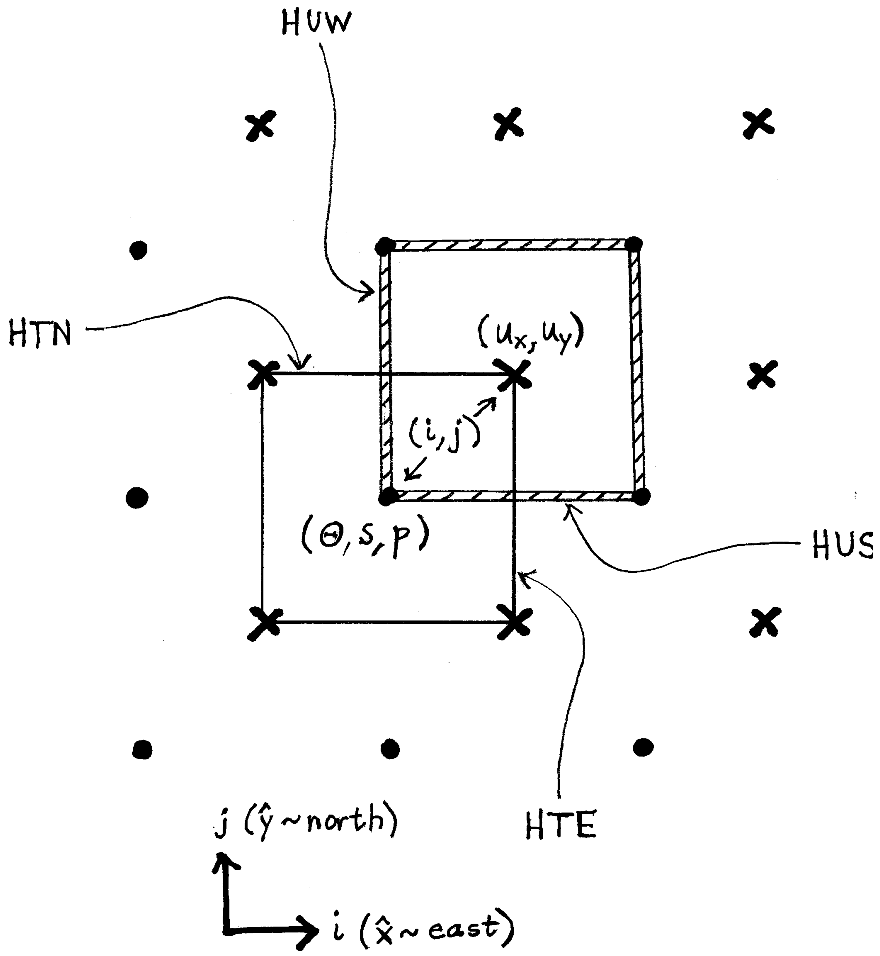

Figure 1. The staggered horizontal B-grid. The x-coordinate grid index i increases to right (generally eastward), and the y-coordinate index j increases upward (generally northward). Solid lines enclose a T-cell, hatched lines a U-cell. The quantities labeled HTN, HTE, HUW, HUS, as well as the model prognostic variables (Θ, S, p, u_x, u_y) at the locations shown all have grid indices (i, j).

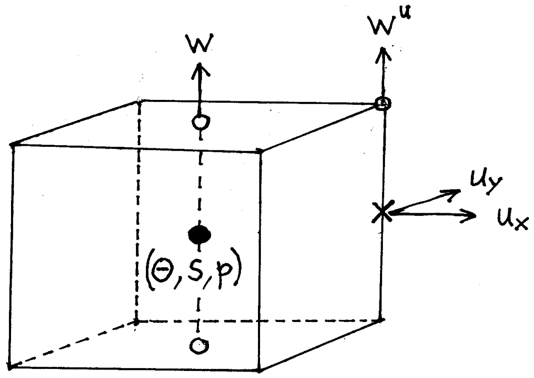

Figure 2: The 3-D T-cell, showing the location of the vertical velocity in T-columns and U-columns.

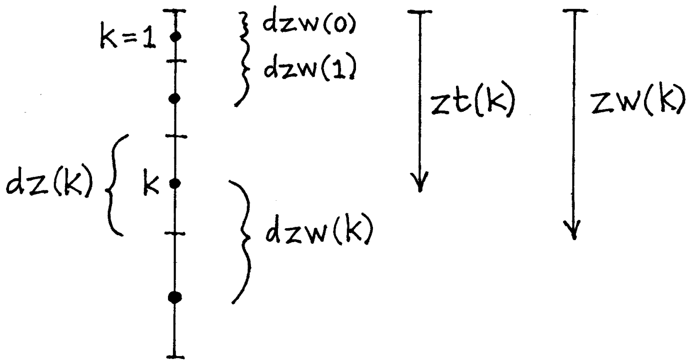

Figure 3: The vertical grid.

The fields ULAT, ULONG and ANGLE are used primarily to interpolate the wind stress fields from a latitude-longitude grid to the model grid if needed. ULAT is used to compute the Coriolis parameter f at each model grid point.

Figure 2 is a diagram of the full 3-dimensional T-cell, showing the location of the vertical velocities w and w^U, which advect tracers and momentum, respectively. Note that the vertical velocites w are located in the middle of the top and bottom faces of the T-cell, while the horizontal velocities are located at the midpoints of the vertical edges.

Since POP is a z-level model, the depth of each point (i, j, k) is independent of its horizontal location (unless partial bottom cells are used, see Sec. 6.4). The vertical discretization is illustrated in Figure 3. The discrete index k increases from the surface ($k = 1$) to the deepest level ($k = km$). The thickness of cells at level k is dz_k. T-points are located exactly in the middle of each level, but since the vertical grid may be non-uniform ($dz_k \neq dz_{k+1}$), the interfaces where the vertical velocities w lie are not exactly halfway between the T-points. The vertical distances between T-points dzw_k are just the average $dzw_k = 0.5(dz_k + dz_{k+1})$, except at the surface where $dzw_0 = 0.5dz_1$. The depth of a T-point at level k is zt_k, and zw_k is the depth of the bottom of cells at level k. Note that while the coordinate z is positive upward, zt_k and zw_k are positive depths. Vertical profiles of dz_k are usually generated offline and read in by the code, but there is an option for generating this profile internally. Usually dz_k is small in the upper ocean and increases with depth according to a smooth analytic function describing the thickness as a function of depth. This is necessary in order to maintain the formal second-order accuracy of the vertical discretization; if the vertical spacing changes suddenly the scheme reverts to first order accuracy (see for example Smith et al., 1995, Sec. 4).

The topography is defined in the T-cells, which are completely filled with either land or ocean (except when optional partial bottom cells are used; see Sec. 6.4). Thus U-points lie exactly on the lateral boundaries between land and ocean, and w points lie exactly on the ocean floor. These boundary velocites are always set to zero due to the no-slip boundary conditions, however, the vertical velocities w^U along the rims in the stairstep topography may be nonzero (see the discussion of velocity boundary conditions in Sec. 3.4.1). The topography is determined by the 2-D integer field $\mathrm{KMT}_{i,j}$ which gives the number of open ocean points in each vertical column of T-cells. The KMT field is usually generated offline and read in from a file in the code. Thus $0 \leq \mathrm{KMT} \leq km$, and $\mathrm{KMT} = 0$ indicates a surface land point. In some situations the ocean depth in a column of U-points is needed, and this is defined by the field $\mathrm{KMU}_{i,j}$, which is just the minimum of the four surrounding values of KMT:

$$\mathrm{KMU}_{i,j} = \min\{\mathrm{KMT}_{i,j}, \mathrm{KMT}_{i-1,j}, \mathrm{KMT}_{i,j-1}, \mathrm{KMT}_{i-1,j-1}\} \tag{31}$$

The depths of columns of ocean T-points and U-points are given, respectively, by:

$$\begin{aligned} H_T &= zw(\mathrm{KMT}) \; , \\ H_U &= zw(\mathrm{KMU}) \; . \end{aligned} \tag{32}$$

With partial bottom cells the depth of the deepest ocean cell in each column has variable thickenss, and the above formulas are modified accordingly (see Sec. 6.4).

3.2. Finite-difference operators

The exact finite-difference versions of the differential operators can be easily derived for the various types of staggered horizontal grids A,B,C,D,E (Arakawa and Lamb, 1977) given only the forms of the fundamental operators: divergence, gradient, and curl for that type of mesh. POP employs a B-grid (scalars at cell centers, vectors at cell corners) while some OGCM's use a C-grid (scalars at cell centers, vector components normal to cell faces). We will use standard notation (Semtner, 1986) for finite-difference derivatives and averages:

$$\begin{aligned} \delta_x\psi &= \left[\psi\left(x+\Delta_x/2\right)-\psi\left(x-\Delta_x/2\right)\right]/\Delta_x && (33)\\ \overline{\psi}^x &= \left[\psi\left(x+\Delta_x/2\right)+\psi\left(x-\Delta_x/2\right)\right]/2\ , && (34)\end{aligned}$$

with similar definitions for differences and averages in the y and z directions. These formulas strictly apply for uniform grid spacing; where, for example, if ψ is a tracer located at T-points, then $\psi\left(x+\Delta_x/2\right)$ is located on the east face of a T-cell. For nonuniform grid spacing, the above definitions should be interpreted such that variables lie exactly at T- or U-cell centers and faces, as appropriate.

The fundamental operators on C-grids have the same form as Eqs. (18)-(22). On B-grids the derivatives involve transverse averaging, and the fundamental operators are given by:

$$\begin{aligned} \nabla\psi &= \hat{\mathbf{x}}\delta_x\overline{\psi}^y+\hat{\mathbf{y}}\delta_y\overline{\psi}^x && (35)\\ \nabla\cdot\mathbf{u} &= \frac{1}{\Delta_y}\delta_x\overline{\Delta_y u_x}^y+\frac{1}{\Delta_x}\delta_y\overline{\Delta_x u_y}^x && (36)\\ \hat{\mathbf{z}}\cdot\nabla\times\mathbf{u} &= \frac{1}{\Delta_y}\delta_x\overline{\Delta_y u_y}^y-\frac{1}{\Delta_x}\delta_y\overline{\Delta_x u_x}^x && (37)\\ \nabla\cdot G\nabla\psi &= \frac{1}{\Delta_y}\delta_x\overline{[\Delta_y G\delta_x\overline{\psi}^y]}^y+\frac{1}{\Delta_x}\delta_y\overline{[\Delta_x G\delta_y\overline{\psi}^x]}^x\ . && (38)\end{aligned}$$

The gradient is located at U-points and the divergence, curl and Laplacian are located at T-points. In the Laplacian operator G must also be defined at U-points. The factors Δ_x, Δ_y inside the difference operators δ_x, δ_y are located at U-points and are given by DXU, DYU, respectively, while the factors $1/\Delta_x$, $1/\Delta_y$ outside the difference operators, as well as similar factors in the denominators of the difference operators δ_x, δ_y, are evaluated at T-points. For example, the first term on the r.h.s. of the divergence (36) at the T-point (i,j) is given by

$$\begin{aligned} &0.5[\mathrm{DYU}_{i,j}(u_x)_{i,j}+\mathrm{DYU}_{i,j-1}(u_x)_{i,j-1} && (39)\\ &-\mathrm{DYU}_{i-1,j}(u_x)_{i-1,j}-\mathrm{DYU}_{i-1,j-1}(u_x)_{i-1,j-1}]/\mathrm{TAREA}_{i,j} \end{aligned}$$

In POP (and in other Bryan-Cox models which use a B-grid formulation) all viscous and diffusive terms are given in terms of an approximate C-grid discretization in order to ensure they will damp checkerboard oscillations on the scale of the grid spacing (see Secs. 3.3.2, 5.1, and 5.2).

3.3. Discrete Tracer Transport Equations

The discrete tracer transport equations are:

$$\frac{\partial}{\partial t}(1+\xi)\varphi + \mathcal{L}_T(\varphi) = \mathcal{D}_H(\varphi) + \mathcal{D}_V(\varphi) + \mathcal{F}_W(\varphi) \tag{40}$$

where $\mathcal{L}_T$ is the advection operator in T-cells, and $\mathcal{D}_H$, $\mathcal{D}_V$ are the horizontal and vertical diffusion operators, respectively. The factor $(1+\xi)$ is associated with the change in volume of the surface layer due to undulations of the free surface, and $\mathcal{F}_W(\varphi)$ is the change in tracer concentration associated with the freshwater flux. ξ and $\mathcal{F}_W(\varphi)$ are given by

$$\xi = \frac{\delta_{k1}}{dz_1}\eta \tag{41}$$

$$\mathcal{F}_W(\varphi) = \frac{\delta_{k1}}{dz_1}\sum_m q_W^{(m)}\varphi_W^{(m)} \tag{42}$$

where δ_{k1} is the Kronecker delta, equal to 1 for $k=1$ and zero otherwise. η is the displacement of the free surface relative to $z=0$, $q_W^{(m)}$ is the freshwater flux per unit area associated with a specific source (labeled m) of freshwater. Thus, $q_W = \sum_m q_W^{(m)} = P - E + R - F_{ice} + M_{ice}$ is the total freshwater flux per unit area associated with precipitation P, evaporation E, river runoff R, freezing F_{ice} and melting M_{ice} of sea ice, and $\varphi_W^{(m)}$ is the tracer concentration in the freshwater associated with source m. The change in volume of the surface layer due to the freshwater flux is discussed in Sec. 4.5.1, and the natural boundary conditions for tracers associated with freshwater flux are discussed in Sec. 6.3. The boundary conditions on tracers are no-flux normal to bottom and lateral boundaries.

3.3.1. Tracer Advection. POP currently has two options for tracer advection, a standard 2nd-order centered advection scheme and a 3rd-order upwind scheme (see Sec. 6.1). In the standard 2nd-order scheme, the advection operator is given by:

$$\mathcal{L}_T(\varphi) = \frac{1}{\Delta_y}\delta_x(\overline{\Delta_y u_x}^y\,\overline{\varphi}^x) + \frac{1}{\Delta_x}\delta_y(\overline{\Delta_x u_y}^x\,\overline{\varphi}^y) + \delta_z(w\overline{\varphi}^z) \tag{43}$$

Again, Δ_x and Δ_y inside the difference operator are located at U-points, and the mass fluxes $\overline{\Delta_y u_x}^y$, $\overline{\Delta_x u_y}^x$ are located on the lateral faces of T-cells. $\overline{\varphi}^x$ and $\overline{\varphi}^y$ are also located on the lateral faces of T-cells, while $\overline{\varphi}^z$ is located on the top and bottom faces of T-cells. At the surface, $\overline{\varphi}^z$ is set equal to zero since there is no advection of tracers across the surface. The vertical velocity w at T-points is determined from the solution of the continuity equation

$$\frac{1}{\Delta_y}\delta_x(\overline{\Delta_y u_x}^y) + \frac{1}{\Delta_x}\delta_y(\overline{\Delta_x u_y}^x) + \delta_z(w) = 0 \tag{44}$$

which is integrated in a column of T-cells downward from the top with the boundary conditions:

$$w = \frac{\partial}{\partial t}\eta + q_W \quad \text{at} \quad z = 0 \tag{45}$$

$$w = 0 \quad \text{at} \quad z = -H_T\ . \tag{46}$$

The boundary condition (45) is discussed in more detail in Sec. 4.5.1.

When integrating from the top down using (44) and (45), the bottom boundary condition (46) will only be satisfied to the extent that the solution of the elliptic system of barotropic equations for the surface height η has converged exactly (see Sec. 4.5). In practice, this exact convergence is not achieved, so the bottom velocity in T-columns is set to zero to ensure no tracers are fluxed through the bottom. This amounts to allowing a very small divergence of velocity in the bottom cell.

In versions of POP prior to 1.4.3, the volume of the surface cells is assumed constant, and no account is taken of the change in volume of the surface cells when the free surface height changes. Thus $\xi = 0$ and $q_W = 0$ in Eqs. (40), (41) and (42), and the freshwater flux is approximated as a virtual salinity flux and imposed as a boundary condition on the vertical diffusion operator. In addition, the tracers are advected through the surface in this formulation using (43) with the vertical velocity given by (45) and $\overline{\varphi}^z = \varphi_1$ at the surface. One problem with this approximation is that the advective flux of tracers through the surface is not zero in global average. The globally-integrated vertical mass flux vanishes, but the integrated tracer flux does not. In practice, we have found that the residual surface tracer fluxes associated with this are usually small, but in some situations they may be nonnegligible. (Note: the global mean residual surface tracer fluxes are standard diagnostic model output in the earlier versions of POP without the variable thickness surface layer.) Because the residual surface flux is nonzero, the global mean tracers are not conserved in the absence of surface forcing. In the newer versions of the code based on (40), (41) and (42), global mean tracers (in particular total salt) are conserved exactly. The differences between the old version and the new version with the variable-thickess surface layer are detailed in Sec. 6.3.

3.3.2. Horizontal Tracer Diffusion. POP has three options for horizontal tracer diffusion: 1) horizontal Laplacian diffusion, 2) horizontal biharmonic diffusion, 3) the Gent-McWilliams parameterization, which includes along-isopycnal tracer diffusion and tracer advection with an additional eddy-induced transport velocity. All of these are implemented for a spatially-varying diffusivity. The first of these options (Laplacian diffusion) is described here, the other two options are discussed in Sec. 5.1. The discrete horizontal Laplacian diffusion operator is given by:

$$\mathcal{D}_H(\varphi) = \frac{1}{\Delta_y}\delta_x(\overline{A_H}^x \Delta_y \delta_x \varphi) + \frac{1}{\Delta_x}\delta_y(\overline{A_H}^y \Delta_x \delta_y \varphi) \tag{47}$$

Note that no lateral averaging is involved as in Eq. (38). Thus the Laplacian is approximated as a 5-point stencil as would be used on a C-grid. The factors Δ_x, Δ_y inside the difference operator are given by HTN, HTE, respectively (see Fig. 1), and A_H, defined at T-points, is averaged across the T-cell faces. As mentioned above, all the horizontal diffusive operators in POP that would normally involve 9-point operators (including the GM parameterization and the horizontal friction operators), are approximated by 5-point C-grid operators in order to ensure that they damp checkerboard noise on the grid scale. B-grid Laplacian-type operators like (38) have a checkerboard null space, i.e., they yield zero when applied to a +/− checkerboard field, and thus cannot damp noise of this character (see Sec. 6.3.1). The only Laplacian-type operator which uses a B-grid discretization is the elliptic operator in the implicit barotropic system. There the B-grid discretization is required in order to maintain energetic consistency (Smith et al., 1992, Dukowicz et al., 1993, see also Secs. 3.6 and 4.5). The boundary conditions on the diffusive operator (47) are that tracer gradients $\delta_x\varphi$, $\delta_y\varphi$ are zero normal to lateral boundaries.

3.3.3. Vertical Tracer Diffusion. The spatial discretization of the vertical diffusion operator is given by:

$$\begin{aligned}\mathcal{D}_V(\varphi) &= \delta_z(\kappa\delta_z\varphi)\\ &= \frac{1}{dz_k}\left(\frac{\kappa_{k-\frac{1}{2}}(\varphi_{k-1}-\varphi_k)}{dz_{k-\frac{1}{2}}} - \frac{\kappa_{k+\frac{1}{2}}(\varphi_k-\varphi_{k+1})}{dz_{k+\frac{1}{2}}}\right).\end{aligned} \tag{48}$$

where φ_k is a tracer at level k, and $\kappa_{k-\frac{1}{2}}$, $\kappa_{k+\frac{1}{2}}$ are evaluated on the top and bottom faces, respectively, of the T-cell at level k, and $dz_{k-\frac{1}{2}} = dzw_{k-1}$, $dz_{k+\frac{1}{2}} = dzw_k$. The boundary conditions at the top and bottom of the column are

$$\begin{aligned}\kappa\delta_z\varphi &\to Q_\varphi \quad \text{at} \quad z=0\\ \kappa\delta_z\varphi &\to 0 \quad \text{at} \quad z=-H_T\end{aligned} \tag{49}$$

where Q_φ is the surface flux of tracer φ (e.g., heat flux for temperature and equivalent salt flux associated with freshwater flux for salinity). The modifications to this discretization when partial bottom cells are used is described in Sec. 6.4. The diffusive term may either be evaluated explicitly or implicitly. The implicit treatment is described in Sec. 4.2.2. With explicit mixing, a convective adjustment routine may also be used to more efficiently mix tracers when the column is unstable (see Sec. 5.3.1). Various subgrid-scale parameterizations for the vertical diffusivity are discussed in Sec. 5.3.

3.4. Discrete Momentum Equations.

The momentum equations discretized on the B-grid are given by:

$$\frac{\partial}{\partial t}u_x + \mathcal{L}_U(u_x) + u_xu_yk_y - u_y^2k_x - fu_y = -\frac{1}{\rho_0}\,\delta_x\overline{p}^y + \mathcal{F}_{Hx}(u_x,u_y) + \mathcal{F}_V(u_x) \tag{50}$$

$$\frac{\partial}{\partial t}u_y + \mathcal{L}_U(u_y) + u_xu_yk_x - u_x^2k_y + fu_x = -\frac{1}{\rho_0}\,\delta_y\overline{p}^x + \mathcal{F}_{Hy}(u_y,u_x) + \mathcal{F}_V(u_y) \tag{51}$$

In these equations no account has been taken of the change in volume of the surface layer due to undulations of the free surface. Therefore, no terms involving ξ appear as in the tracer transport equation (40). The justification for this is that the global mean momentum, unlike the global mean tracers, is not conserved in the absence of forcing, so there is less motivation to correct for momentum nonconservation due to surface height fluctuations. Furthermore, the error introduced is typically small compared to the uncertainty in the applied wind stress.

Note: Currently the code is in cgs units and it is assumed that $\rho_0 = 1.0\text{gm cm}^{-3}$, so it never explicitly appears. If the Boussinesq correction (Sec. 3.5.1) is used, then this factor is already taken into account, because the factor $r(p)$ in (65) is normalized such that the pressure gradient should be divided by $\rho_0 = 1.0$.

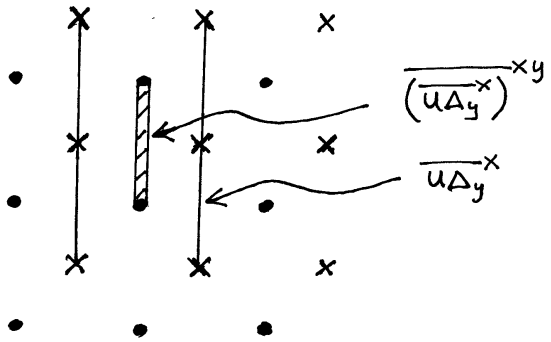

Figure 4: Advective mass fluxes through lateral faces of T-cells (solid lines) and U-cells (hatched line). U-cell fluxes are the average of the four surrounding T-cell fluxes. A similar averaging applies to the vertical fluxes.

3.4.1. Momentum Advection. The nonlinear momentum advection term is discretized as:

$$\mathcal{L}_U(\alpha) = \frac{1}{\Delta_y}\delta_x[\overline{(\overline{\Delta_y u_x}^{y})}^{xy}\overline{\alpha}^{x}] + \frac{1}{\Delta_x}\delta_y[\overline{(\overline{\Delta_x u_y}^{x})}^{xy}\overline{\alpha}^{y}] + \delta_z(w^U\overline{\alpha}^{z}) \ . \qquad (52)$$

This is a second-order centered advection scheme and is currently the only option available in POP for momentum advection. It has the property that global mean kinetic energy is conserved by advection. Momentum is conserved in the interior by advection, but not on the boundaries (see Sec. 3.6 on energetic consistency). The mass fluxes in the operator include an extra average in both horizontal directions, denoted $\overline{(..)}^{xy}$. Both the horizontal and vertical mass fluxes in a U-cell are the average of the four surrounding T-cell mass fluxes. This is illustrated in Figure 4 for the mass flux in the x-direction. As a consequence, the vertical velocity in a U-cell is exactly the area-weighted average of the four surrounding T-cell vertical velocities w. This averaging is necessary in order to maintain the energetic balance between the global mean work done by the pressure gradient and the change in gravitational potential energy (Smith et al., 1992, see also Sec. 3.6 on energetic consistency). An additional advantage of this flux averaging in U-cells is that it substantially reduces noise in the vertical velocity field compared to other approaches (Webb, 1995). At the bottom of a column of ocean U-cells w^U is not necessarily zero, since it is the weighted average of the surrounding w's, some of which may be nonzero if it is a “rim” point where $k =$ KMU but at least one of the surrounding T-points has $k <$ KMT. It can be shown that the value of w^U at these points approximates the boundary condition for tangential flow along the sloping bottom $w = -\mathbf{u} \cdot \nabla H$ (Semtner, 1986, Eq. 34). Momentum is advected through the surface using the vertical velocity (45) averaged to U-points and with $\overline{\alpha}^z = \alpha_1$ (where $\alpha = u_x$ or u_y) in (52).

3.4.2. Metric Terms. The metric terms (third and fourth terms on the l.h.s. in Eq. 51) are constructed using a simple C-grid discretization of the metric factors (Eqs. 26, 27). Specifically, k_x and k_y at U-points are given by:

$$\begin{aligned} \mathrm{KXU}_{i,j} &= [\mathrm{HUW}_{i+1,j} - \mathrm{HUW}_{i,j}]/\mathrm{UAREA}_{i,j} \\ \mathrm{KYU}_{i,j} &= [\mathrm{HUS}_{i,j+1} - \mathrm{HUS}_{i,j}]/\mathrm{UAREA}_{i,j} \end{aligned} \tag{53}$$

3.4.3. Horizontal Friction. Several options for horizontal friction are available in the code. In this section a simple spatial discretization of the Laplacian-type formulation with a spatially varying viscous coefficient as given by (29) is presented. The biharmonic friction operator obtained by applying (28) twice is described in Sec. 5.2.2. More sophisticated formulations of the viscosity based on a functional discretization of the friction operator formulated as the divergence of a viscous stress that is linearly related to the components of the strain-rate tensor are described in Sections 5.2.3 and 5.2.4. These include an anisotropic formulation of the viscosity and the use of Smagorinksy-type nonlinear viscous coefficients.

The Laplacian horizontal friction terms (28) and (29) are constructed from a C-grid discretization of both the Laplacian and the metric terms. The discrete Laplacian terms are given by:

$$\nabla \cdot A_M \nabla \mathbf{u} = \frac{1}{\Delta_y} \delta_x (\overline{A_M}^x \Delta_y \delta_x \mathbf{u}) + \frac{1}{\Delta_x} \delta_y (\overline{A_M}^y \Delta_x \delta_y \mathbf{u}) \tag{54}$$

where A_M, defined at U-points, is averaged across cell faces inside the divergence. Terms proportional to k_x and k_y in (28) and (29) are evaluated using (53). Terms involving derivatives of k_x and k_y are evaluated with k_x, k_y defined at T-points and averaged along U-cell faces. For example, the term $\delta_x A_M k_x$ is evaluated as:

$$\begin{aligned} &\delta_x \overline{A_M}^x \overline{k_x}^y = \\ &0.25\{[(A_M)_{i+1,j} + (A_M)_{i,j}][\mathrm{KXT}_{i+1,j+1} + \mathrm{KXT}_{i+1,j}] \\ &-[(A_M)_{i,j} + (A_M)_{i-1,j}][\mathrm{KXT}_{i,j+1} + \mathrm{KXT}_{i,j}]\}/\mathrm{DXU}_{i,j} \end{aligned} \tag{55}$$

where A_M is averaged across the U-cell faces, and k_x, k_y at T-points are given by

$$\begin{aligned} \mathrm{KXT}_{i,j} &= [\mathrm{HTE}_{i,j} - \mathrm{HTE}_{i-1,j}]/\mathrm{TAREA}_{i,j} \\ \mathrm{KYT}_{i,j} &= [\mathrm{HTN}_{i,j} - \mathrm{HTN}_{i,j-1}]/\mathrm{TAREA}_{i,j} \end{aligned} \tag{56}$$

Finally, in those terms in (28,29) that involve single derivatives of the velocities or viscosities (e.g. $\delta_x u_y$ or, $\delta_x A_M$) the derivatives are evaluated as differences across the cell without using the central value. For example, $\delta_x u_y$ at point (i, j) is evaluated as:

$$[(u_y)_{i+1,j} - (u_y)_{i-1,j}]/[\mathrm{HTN}_{i,j} + \mathrm{HTN}_{i+1,j}] \tag{57}$$

The no-slip boundary conditions are implemented by simply setting $u_x = u_y = 0$ on all lateral boundary points. Modifications to the operator when parital bottom cells are used are described in Sec. 6.4.

3.4.4. Vertical Friction. The spatial discretization of the vertical friction terms is essentially identical to that of the vertical diffusion described in Sec. 3.3.3. The spatial discretization of both $\mathcal{F}_V(u_x)$ and $\mathcal{F}_V(u_y)$ is identical to Eq. (48) with κ replaced by the vertical viscosity μ and φ replaced by one of the velocity components u_x or u_y. Modifications for partial bottom cells are discussed in Sec. 6.4. The boundary conditions at the top and bottom of the column of U-points are

$$\begin{aligned} \mu\delta_z(u_x, u_y) &\rightarrow (\tau_x, \tau_y) \quad \text{at} \quad z = 0 \\ \mu\delta_z(u_x, u_y) &\rightarrow c|\mathbf{u}|(u_x, u_y) \quad \text{at} \quad z = -H_U \end{aligned} \tag{58}$$

where (τ_x, τ_y) are the components of the surface wind stress along the coordinate directions, and a quadratic drag term is applied at the bottom of the column. The dimensionless constant c is typically chosen to be of order 10^{-3}. The semi-implicit treatment of these terms is described in Sec. 4.4.2. Various subgrid-scale parameterizations for the vertical viscosity are described in Sec. 5.3.

3.5. Equation of State

POP requires an equation of state in the form

$$\rho = \rho(\Theta, S, p) \ , \tag{59}$$

to relate density, ρ, to the prognostic variables Θ and S; see (10). To aviod a nonlinear integration of the hydrostatic equation (9), the pressure in the equation of state is approximately evaluated as a time-independent function of depth by means of the equation

$$p = p_0(z) = 0.059808[\exp(-0.025z) - 1] + 0.100766z + 2.28405 \times 10^{-7} z^2 \ , \tag{60}$$

where pressure p is in bars and depth z is in meters. This formula is derived in Dukowicz (2001) from the latest Levitus global mean climatology. At a number of places in the code partial derivatives of ρ with respect to Θ and S may also be necessary.

There are four options in the code for the equation of state:

(A) The density may be obtained from the formula of Jackett and McDougall (1995).

$$\rho = \frac{\rho(\Theta, S, 0)}{1 - p/K(\Theta, S, p)}, \tag{61}$$

The numerator, $\rho(\Theta, S, 0)$, is a 15-term equation in powers of S and Θ with coefficients given by the UNESCO international standard equation of state (Fofonoff and Millard, 1983). The secant bulk modulus, $K(\Theta, S, p)$, is a 26-term equation in powers of Θ, S and p, obtained from a least-squres fit to the expression for the secant bulk modulus as a function of the in situ temperature (Fofonoff and Millard, 1983). This density equation is valid in the range $0 \le S \le 42psu$, $-2^\circ \le \Theta \le 40^\circ C$, and $0 \le p \le 1000bar$. The total of 41 terms in this equation of state makes it the most expensive of the four available options.

(B) A cubic polynomial fit to the UNESCO international standard equation of state (Fofonoff and Millard, 1983) which has the form

$$\rho(\Theta, S, z_k) = \rho_k(\Theta - \Theta_{ref}(k), S - S_{ref}(k)) \;, \tag{62}$$

where $\Theta_{ref}(k)$ and $S_{ref}(k)$ are reference values of the potential temperature and salinity for each model level. The set of nine coefficients in the cubic polynomial for each level are pre-calculated based on a least-squres fit to the international standard equation of state as prescribed by Bryan and Cox (1972). The range of validity of this equation is depth dependent and is specified during the fitting procedure to encompass typically observed oceanographic conditions. This results in a fairly narrow range of validity at depth which may not be appropriate for certain simulations or for evaluating density resulting from large adiabatic displacements. The reduction from 41 terms to 9, however, represents a considerable cost savings over the full UNESCO equation of state.

(C) A 25-coefficient approximation of the Feistel and Hagen (1995) equation of state (which is more accurate relative to laboratory data than the UNESCO form), derived by McDougall et al., (2002).

$$\rho(\Theta, S, p) = \frac{P_1(\Theta, S, p)}{P_2(\Theta, S, p)} \;, \tag{63}$$

where P_1 is a 12 term polynomial and P_2 is a 13-term polynomial. This density equation is valid in the range $0 \le S \le 40psu$, $-2^\circ \le \Theta \le 33^\circ C$, at the surface, diminishing to $30 \le S \le 40psu$, $-2^\circ \le \Theta \le 12^\circ C$ at 550 bar. However, the authors report that the equation is well behaved in the range $0 \le S \le 50psu$, $-10^\circ \le \Theta \le 50^\circ C$, and $0 \le p \le 1000bar$ This equation of state is intermediate in cost between the UNESCO form (A) and the cubic polynomial form (B). It is currently the preferred option for CCSM integrations.

(D) A simple linear equation of state given by

$$\rho = \rho_0 - 2.5^{-4}\Theta + 7.6 \times 10^{-4} S \;, \tag{64}$$

where ρ is in gm/cm^3, Θ is in °C, and S is in practical salinity units (psu). The ρ_0 term is not included in the code for the full density with the linear EOS.

3.5.1. Boussinesq Correction. The pressure-gradient terms in the momentum equations (50) and (51) are an approximation of $\rho^{-1}\nabla p$, introduced as part of the Boussinesq approximation. Furthermore, the conversion of pressure to depth in the equation of state by means of (60) can have significant dynamic consequences through its effect on the pressure gradient (Dewar et al., 1998). Both of these errors can be greatly reduced by transforming the density as follows (Dukowicz 2001):

$$\rho = r(p)\rho^* \;, \tag{65}$$

where ρ^* is termed the thermobaric density, and $r(p)$ is a nondimensional function of pressure that extracts the pressure-dependent part of the adiabatic compressibility from the density along the global-mean Levitus climatology:

$$r(p) = 1.02819 - 2.93161 \times 10^{-4}\exp(-0.05p) + 4.4004 \times 10^{-5}p \ , \tag{66}$$

where p is in bars. This leads to the definition of an associated thermobaric pressure p^*:

$$p^*(p) = \int_0^p \frac{dp'}{r(p')} \ , \tag{67}$$

such that the hydrostatic equation (9) becomes

$$\frac{\partial p^*}{\partial z} = -\rho^* g \ , \tag{68}$$

and the pressure gradient force is transformed into

$$\frac{1}{\rho}\nabla p = \frac{1}{\rho^*}\nabla p^* \approx \frac{1}{\rho_0}\nabla p^* \ . \tag{69}$$

The effective equation of state in terms of these new variables becomes

$$\rho^* = \rho(\Theta, S, p(p^*))/r(p(p^*)) = \rho^*(\Theta, S, p^*) \ . \tag{70}$$

The advantage of this transformation is that the effective adiabatic compressibility associated with the equation of state is now at least an order of magnitude lower. This means that the variation of density with depth is much lower and that fluctuations of pressure in the equation of state have a much smaller effect on density. Thus, the errors associated with the Boussinesq approximation in the pressure gradient force and the linearization of the equation of state in the transformed variables are at least an order of magnitude smaller than without the transformation, as explained in Dukowicz (2001).

The pressure gradient force is approximated by $\rho_0^{-1}\nabla p^*$, and the hydrostatic equation becomes

$$\frac{\partial p^*}{\partial z} = -\frac{\rho(\Theta, S, p_0(z))g}{r(p_0(z))} \ . \tag{71}$$

This implies that the pressure variable handled in POP is the thermobaric pressure p^*, not p. The pressure enters only in the pressure gradient and in the equation of state and effects nothing else. Should the true pressure ever be required it is easily obtainable from p^* using the relation (67). Given this, we shall drop the notation p^* and henceforth interpret p to imply the thermobaric pressure.

3.5.2. Hydrostatic Pressure. The pressure at depth z is obtained by integrating the hydrostatic equation from $z = 0$ (the "hydrostatic pressure" p_h) and adding the contribution from the surface pressure p_s associated with undulations of the free surface:

$$\begin{aligned} p(x, y, z) &= p_s(x, y) + p_h(x, y, z) \\ p_h(x, y, z) &= \int_z^0 dz' g\rho(x, y, z') \end{aligned} \tag{72}$$

In the code only the horizontal pressure gradients are needed, and these are evaluated by integrating the horizontal gradients of density in (72). The discrete formulas for the horizontal pressure gradients at level k are:

$$\begin{aligned}
\delta_x \overline{p_k}^y &= \delta_x \overline{p_s}^y + g \sum_{m=1}^{k} \frac{1}{2}[\delta_x \overline{\rho^*_{m-1}}^y + \delta_x \overline{\rho^*_m}^y]\, dz_{m-\frac{1}{2}} \\
\delta_y \overline{p_k}^x &= \delta_y \overline{p_s}^x + g \sum_{m=1}^{k} \frac{1}{2}[\delta_y \overline{\rho^*_{m-1}}^x + \delta_y \overline{\rho^*_m}^x]\, dz_{m-\frac{1}{2}}
\end{aligned} \tag{73}$$

where $dz_{m-\frac{1}{2}} = dzw_{m-1}$, and $dz_{-\frac{1}{2}} = 0.5dz_1$. Here $\rho^*_m = \rho_m / r(p_0 zt_m)$ if the Boussinesq correction is used, otherwise $\rho^*_m = \rho_m$, where ρ_m is the density at level m, and $\rho_m = \rho_1$ when $m = 0$.

3.5.3. Expansion Diagnostic. This is a feature that is currently not in the code but will be added later. The expansion diagnositc is supposed to account for the change in global mean sea level due to two effects: 1) the change in volume associated with the net accumulated freshwater flux into the ocean (from precipitation, evaporation, melting and freezing of sea ice, and river runnoff); and 2) the change in volume associated with steric expansion due to changes in global mean density. If the model employs virtual salinity fluxes, either from restoring to climatological surface salinity or from the conversion of actual freshwater fluxes to virtual salinity fluxes (e.g., when the model is run without using the natural boundary conditions for freshwater flux as discussed in Sec. 6.3) then the expansion effects due to 1) and 2) are not easily separated: a virtual salinity flux will change the salinity and hence the density through the equation of state, on the other hand, a virtual salinity flux is usually associated with an actual freshwater flux which changes the volume as well, and it is not clear how to cleanly separate these volume and density changes. However, in the steric effect the change in density is primarily associated with thermal expansion due to heating or cooling, rather than changes in salinity. Hence, we expect that computing the steric effect due to density changes without including volume changes due to freshwater flux provides a useful diagnositic. Therefore, in the code the change in surface elevation due to the steric effect can be computed assuming there is no change in total volume. The change in surface elevation due to steric expansion is computed as:

$$\langle \eta^n \rangle = \langle H \rangle \left(\frac{\langle \rho^0 \rangle}{\langle \rho^n \rangle} - 1 \right) \tag{74}$$

where $\langle H \rangle$, the ratio of ocean volume to surface area, is the mean ocean depth, $\langle \rho^0 \rangle$ is the initial global mean density, $\langle \rho^n \rangle$ is the global mean density at the n^{th} model timestep computed assuming there is no change in total volume, and $\langle \eta^n \rangle$ is the estimated change in sea-surface elevation at the n^{th} timestep due to steric expansion only. If natural boundary conditions for freshwater flux are employed, then the actual volume of the ocean in the model does change (see Sec. 6.3), and the corresponding change in mean sea-level is diagnosed as the actual global-mean sea level in the model. This change can be added to the change due to steric expansion to obtain an approximate total change in global mean sea level due to the combined effects of 1) and 2).

3.6. Energetic Consistency

This section is not yet comlete. The energetic balances in the free-surface formulation are described in detail in Dukowicz and Smith (1994). However, this reference predates the variable-surface thickness layer (Sec. 6.3) and partial bottom cells (Sec. 6.4), both of which influence the energetic balances.

3.7. CFL Diagnostics

This section is not yet complete. The POP code prints several global CFL diagnostics to standard output. These are associated with restrictions on numerical stability asociated with the explicit integration of advection of tracers and momentum (both horizontal and vertical), viscosity, and diffusion. The CFL diagnostics depend on both the space and time discretization. They are computed in the POP source file `diagnostics.F`.

4. Time Discretization

The POP model uses a 3-time-level second-order-accurate modified leapfrog scheme for stepping forward in time. It is modified in the sense that some terms are evaluated semi-implicitly, and of the terms that are treated explicitly, only the advection operators are actually evaluated at the central time level, as in a pure leapfrog scheme. The diffusive terms are evaluated using a forward step. The reason for this is that the centered advection scheme is unstable for forward steps, and the diffusive scheme is unstable for leapfrog steps.

4.1. Filtering Timesteps

Leapfrog schemes can develop computational noise due to the partial decoupling of even and odd timesteps. In a pure leapfrog scheme they are completely decoupled and the solutions on the even and odd steps can evolve independently, leading to $2\Delta t$ oscillations in time. There are several methods to damp the leapfrog computational mode, two of which are currently implemented in POP. One is to occasionally take a forward step or an Euler forward-backward step (sometimes called a 'Matsuno' timestep, Haltiner and Williams, 1980). The Matsuno step is more expensive than a forward step, but it is stable for advection. The other method is to occasionally perform an averaging of the solution at three successive time levels to the two intermediate times, back up half a timestep and proceed. The later procedure is refered to as an "averaging timestep" (Dukowicz and Smith 1994) and is the recommended method for eliminating the leapfrog computational mode. The leapfrog scheme generates two different "trajectories" of the solution, one corresponding to even and one to odd steps. The advantage of the averaging step is that it places the solution on the average trajectory, whereas the forward and Matsuno steps select only one trajectory, corresponding to either the even or the odd solution. Experience has shown that some model configurations are not stable using Matsuno filtering timesteps, and this is especially true with the variable-thickness surface layer (Sec. 6.3). The Matsuno step is a forward predictor step followed by a "backward" step which is essentially a repeat of the forward step but using the predicted prognostic variables from the first pass to evaluate all terms except the time-tendency term (see

Sec. 4.2). On the very first time step of a spinup from rest a forward step is taken to avoid immediately exciting a leapfrog computational mode (this feature is hardwired into the code). In the presentation below, the time discretization of various terms will first be presented for a regular leapfrog step, then the discretization for the forward and backward steps will be given.

4.2. Tracer Transport Equations

Labeling the three time levels on a given step as $n-1$, n, and $n+1$, the tracer transport equation (40) is discretized in time as follows:

$$
\begin{aligned}
(1+\xi^{n+1})\varphi^{n+1} - (1+\xi^{n-1})\varphi^{n-1} &= \tau[-\mathcal{L}_T^{(n)}(\varphi^n) + \mathcal{D}_H(\varphi^{n-1}) + \mathcal{D}_V(\overline{\varphi}^\lambda) + \mathcal{F}_W^n] && (75)\\
\overline{\varphi}^\lambda &= \lambda\varphi^{(n+1)} + (1-\lambda)\varphi^{(n-1)} && (76)
\end{aligned}
$$

where Δt is the timestep and $\tau = 2\Delta t$. $\mathcal{F}_W^n$ is given by (42) with q_W and φ evaluated at time n, and ξ^n is given by (41) with η evaluated at time n. The superscript (n) on the advection operator indicates that the advective mass fluxes are evaluated using the time n velocities. The vertical diffusion term may be evaluated either explicitly or semi-implicitly. In the explicit case $\lambda = 0$. For semi-implicit diffusion $\lambda \geq 1/2$ is required for stability, and the code is usually run with $\lambda = 1$. The surface forcing (49), applied as a boundary condition on the vertical diffusion operator, is evaluated at time n for both explicit and semi-implicit mixing. The modifications to the tracer transport equations with implicit vertical mixing are decribed in Sec. 4.2.2.

With pressure averaging (see Sec. 4.4.1) the potential temperature and salinity at the new time are needed to evaluate the pressure gradient at the new time. This is required in the baroclinic momentum equations before the barotropic equations have been solved for the surface height at the new time. Therefore, (75) cannot be used to predict the new tracers since ξ^{n+1} is not yet known. Instead an approximation is made to predict the new temperature and salinity which are then used to evaluate the pressure gradient at the new time. After the barotropic equations have been solved, the new potential temperature and salinity are corrected so that they satisfy (75) exactly. The equations for predicting and correcting the tracers at the new time are obtained as follows. The l.h.s. of (75), which involves the unknown surface height η^{n+1}, is approximated as:

$$
\begin{aligned}
(1+\xi^{n+1})\varphi^{n+1} - (1+\xi^{n-1})\varphi^{n-1} &= (\varphi^{n+1} - \varphi^{n-1})\\
&+ \left(\frac{\xi^{n+1}+\xi^{n-1}}{2}\right)(\varphi^{n+1} - \varphi^{n-1})\\
&+ (\xi^{n+1} - \xi^{n-1})\left(\frac{\varphi^{n+1}+\varphi^{n-1}}{2}\right)\\
&\approx (1+\xi^n)(\varphi^{n+1} - \varphi^{n-1}) + 2(\xi^n - \xi^{n-1})\varphi^n && (77)
\end{aligned}
$$

With this approximation, the equations for predicting and correcting Θ and S are given by:

Predictor:

$$(1+\xi^n)(\widehat{\varphi}^{n+1} - \varphi^{n-1}) = -2(\xi^n - \xi^{n-1})\varphi^n + \tau F \qquad (78)$$

Corrector:

$$(1+\xi^{n+1})\varphi^{n+1} = (1+\xi^n)\widehat{\varphi}^{n+1} + (\xi^n - \xi^{n-1})(2\varphi^n - \varphi^{n-1}) \qquad (79)$$

where $\widehat{\varphi}^{n+1}$ is the predicted tracer at the new time, and F represents all terms in brackets on the r.h.s. of (75). Note that these equations are used only to predict and correct Θ and S; all passive tracers are updated directly using (75).

The time discretization for the two-time-level forward and backward steps is given by

Forward step:

$$(1+\xi^*)\varphi^* - (1+\xi^n)\varphi^n = \tau[-\mathcal{L}_T^{(n)}(\varphi^n) + \mathcal{D}_H(\varphi^n) + \mathcal{D}_V(\overline{\varphi}^\lambda) + \mathcal{F}_W^n]$$
$$\overline{\varphi}^\lambda = \lambda\varphi^* + (1-\lambda)\varphi^n \tag{80}$$

Backward step:

$$(1+\xi^{n+1})\varphi^{n+1} - (1+\xi^n)\varphi^n = \tau[-\mathcal{L}_T^{(*)}(\varphi^*) + \mathcal{D}_H(\varphi^*) + \mathcal{D}_V(\overline{\varphi}^\lambda) + \mathcal{F}_W^n]$$
$$\overline{\varphi}^\lambda = \lambda\varphi^{n+1} + (1-\lambda)\varphi^n \tag{81}$$

Here $\tau = \Delta t$ and φ^*, the predicted tracer at the new time from the forward step, is used to evaluate the r.h.s. in the backward step. ξ^* is given by (41) with $\eta = \eta^*$. $\mathcal{L}_T^{(*)}$ is the advection operator evaluated using the predicted velocities from the forward step (see Sec. 4.4). For a forward step only, $\varphi^* = \varphi^{n+1}$. The surface forcing applied to $\mathcal{D}_V$, as well as the freshwater tracer flux $\mathcal{F}_W$, are evaluated at time n in both forward and backward steps.

4.2.1. Tracer Acceleration. The acceleration technique of Bryan (1984) can be used to more quickly spin up the model to a steady-state by modifying the timestep for tracers in the deep ocean. The basic method is to modify the time step τ as follows: in the baroclinic and barotropic momentum equations (94) and (113)

$$\tau \rightarrow \tau_k^{(u)} = \tau^{(u)}/\alpha \tag{82}$$

and in the tracer transport equation (75)

$$\tau \rightarrow \tau_k^{(\varphi)} = \tau^{(\varphi)}/\gamma_k \tag{83}$$

where superscripts denote timesteps for tracers (φ), and momentum and continuity (u). α and γ_k are acceleration factors that are specified by the namelist model input: $dtuxcel = \alpha^{-1}$, $dttxcel(k) = (\gamma_k)^{-1}$.

A couple of points should be made about the use of acceleration. First, the method is only guaranteed to reach the correct steady state solution if the external forcing includes some kind of restoring term for the tracer (this will drive the solution toward a unique steady state with or without acceleration). Second, if in the surface layer $\alpha \neq \gamma_1$, the tracer transport and barotropic continuity equations will be inconsistent, and this can cause the model to blow up with the variable-thickness surface layer (see the discussion at the end of Sec. 6.3). Furthermore, if $\alpha \neq 1$ or $\gamma_k \neq 1$ then the surface forcing for momentum and tracers will not occur at the correct time (the model calender and all the forcing fields are based on the unaccelerated surface tracer time step). For this reason we recommend that momentum acceleration not be used, and that the tracer acceleration only be applied at subsurface levels so that $\alpha = \gamma_1 = 1$. A disadvantage of depth-dependent tracer acceleration is that it leads to non-conservative advective and diffusive fluxes when $\gamma_k \neq \gamma_{k+1}$. For this reason it is recommended that profiles of γ_k be smooth functions of depth, and that the largest vertical discontinuities in γ_k be restricted to depths where fluxes are small enough that tracers are well conserved. Nevertheless, it should be noted that (with restoring boundary conditions) the correct steady-state solution will be reached even when γ_k varies with depth. The depth-dependent tracer timestep must also be accounted for in the convective adjustment and vertical mixing routines These changes are described in Secs. 5.3.1, 4.2.3, and 4.4.2.

4.2.2. Implicit Vertical Diffusion. The tracer equations with implicit vertical diffusion involve the solution of a tridiagonal system in each vertical column of grid points. This is a relatively easy thing to do since the system does not involve any coupling with neighboring points in the horizontal direction. The equations solved in the code are

$$\begin{aligned}[1+\xi^{n+1}-\lambda\tau\mathsf{A}(\kappa)]\Delta\varphi \;&=\; -(\xi^{n+1}-\xi^{n-1})\varphi^{n-1}+\tau[-\mathcal{L}_T^{(n)}(\varphi^n)+\mathcal{D}_H(\varphi^{n-1})+\mathsf{A}(\kappa)\varphi^{n-1}+\mathcal{F}_W^n] \\ \mathsf{A}(\kappa)=\delta_z\kappa\delta_z \quad &, \qquad \Delta\varphi\equiv\varphi^{n+1}-\varphi^{n-1}\end{aligned} \tag{84}$$

where $\tau = 2\Delta t$. For explicit diffusion $\lambda = 0$, so the term in brackets on the right corresponds to the exact r.h.s. in the explicit case. The system is solved for the change in tracer $\Delta\varphi$, subject to the no-flux boundary conditions

$$\delta_z\Delta\varphi = 0 \quad \text{at both } \; z=0 \; \text{ and } \quad z=-H_T\;. \tag{85}$$

Note that since the surface forcing and freshwater tracer fluxes are evaluated at time n they are entirely contained in the r.h.s. of (84), and hence are not directly imposed as a boundary condition on the operator.

The predictor and corrector steps for updating Θ and S with pressure averaging (Sec. 4.4.1) are given by:

Predictor:

$$\begin{aligned}&[1+\xi^n-\lambda\tau\mathsf{A}(\kappa)]\Delta\varphi=-2(\xi^n-\xi^{n-1})\varphi^n+\tau F \\ &\Delta\varphi\equiv\widehat{\varphi}^{n+1}-\varphi^{n-1}\end{aligned} \tag{86}$$

Corrector:

$$\begin{aligned}&[1+\xi^{n+1}-\lambda\tau\mathsf{A}(\kappa)]\Delta\varphi=(\xi^n-\xi^{n-1})(2\varphi^n-\varphi^{n-1})-(\xi^{n+1}-\xi^n)\widehat{\varphi}^{n+1} \\ &\Delta\varphi\equiv\varphi^{n+1}-\widehat{\varphi}^{n+1}\end{aligned} \tag{87}$$

where F represents all terms in brackets on the r.h.s. of (84). These reduce to (78) and (79) in the limit $\lambda = 0$. Note that both the predictor and corrector steps involve the solution of a tridiagonal system.

In forward and backward timesteps the implicit equations have the form:

Forward step:

$$\begin{aligned}&[1+\xi^*-\lambda\tau\mathsf{A}(\kappa)]\Delta\varphi=-(\xi^*-\xi^n)\varphi^n+\tau[-\mathcal{L}_T^{(n)}(\varphi^n)+\mathcal{D}_H(\varphi^n)+\mathsf{A}(\kappa)\varphi^n+\mathcal{F}_W^n] \\ &\Delta\varphi\equiv\varphi^*-\varphi^n\end{aligned} \tag{88}$$

Backward step:

$$\begin{aligned}&[1+\xi^{n+1}-\lambda\tau\mathsf{A}(\kappa)]\Delta\varphi=-(\xi^{n+1}-\xi^n)\varphi^n+\tau[-\mathcal{L}_T^{(*)}(\varphi^*)+\mathcal{D}_H(\varphi^*)+\mathsf{A}(\kappa)\varphi^n+\mathcal{F}_W^n] \\ &\Delta\varphi\equiv\varphi^{n+1}-\varphi^n\end{aligned} \tag{89}$$

where $\tau = \Delta t$.

4.2.3. Tridiagonal Solver. For the tridiagonal solution of (84), (86), (87), (88) or (89), a new algorithm is used (Schopf and Loughe, 1995) which is more accurate and stable than traditional methods. These equations have the form:

$$(1+\xi)X_k - \lambda \frac{\tau_k}{dz_k}\left[\frac{\kappa_{k-\frac{1}{2}}}{dz_{k-\frac{1}{2}}}(X_{k-1} - X_k) - \frac{\kappa_{k+\frac{1}{2}}}{dz_{k+\frac{1}{2}}}(X_k - X_{k+1})\right] = R_k \tag{90}$$

where $X_k = \Delta\varphi$ at level k, R_k is the r.h.s., and a depth-dependent timestep Δt_k ($\tau_k = 2\Delta t_k$ for leapfrog and $\tau_k = \Delta t_k$ for forward/backward timesteps) is used with tracer acceleration (see Sec. 4.2.1). Here $\xi = \xi^{n+1}$, except in the predictor step (86) where $\xi = \xi^n$. (90) can be rewritten in the form of a tridiagonal system:

$$\begin{aligned}
&-A_k X_{k+1} + C_k X_k - A_{k-1} X_{k-1} = R'_k \\
&A_k = \lambda \frac{\kappa_{k+\frac{1}{2}}}{dz_{k+\frac{1}{2}}} \\
&C_k = h'_k + A_k + A_{k-1} \\
&R'_k = h_k R_k \\
&h_k = \frac{dz_k}{\tau_k} \\
&h'_k = h_k(1+\xi)
\end{aligned} \tag{91}$$

The algorithm for the solution of this system involves a loop over vertical levels to determine the coefficients:

$$\begin{aligned}
&E_k = A_k / D_k \\
&F_k = (R'_k + A_{k-1} F_{k-1}) / D_k \\
&\text{where} \\
&B_k = A_k (h'_k + B_{k-1}) / D_k \\
&D_k = h'_k + A_k + B_{k-1}
\end{aligned} \tag{92}$$

The loop begins at $k = 1$ with $A_0 = B_0 = F_0 = 0$. This is followed by another vertical loop to determine the solution by back substitution:

$$X_k = E_k X_{k+1} + F_k \tag{93}$$

This loop begins at the bottom with $X_{k+1} = 0$ when $k = \mathrm{KMT}$.

4.3. Splitting of the Barotropic and Baroclinic Modes

The barotropic mode of the primitive equations supports fast gravity waves with speeds of $\sqrt{gH} \sim 200$ m s^{-1}. If resolved numerically, these waves impose a severe restriction on the model timestep. However, they have little effect on the dynamics, especially on timescales longer than a day or so. To overcome this severe limitation on the timestep, the barotropic mode is split off and solved as a separate 2-D system (see Sec. 4.5).

The barotropic equations are taken to be the vertically-integrated momentum and continuity equations. The true barotropic mode is not exactly isolated by the vertically-integrated system, except in the limit of a flat-bottom topography, a rigid lid, and a depth-independent buoyancy frequency. Nevertheless, it suffices in practice to isolate and treat the vertically-integrated system. Then, provided advective and diffusive CFL limits do not control the timestep, the baroclinic equations can be integrated with a timestep that is controlled by the gravity-wave speed of the first internal mode, which is of order 2 m s^{-1}, two orders of magnitude smaller than the barotropic wave speed. The procedure for solving the split barotropic-baroclinic system is as follows.

1) First the momentum equations are solved, without including the surface pressure gradient, for an auxiliary velocity $\mathbf{u}'$. This is what the momentum at the new time would be in the absence of the surface pressure gradient, which is depth-independent and hence determined only by the solution of the vertically-integrated system. The time discretization of the resulting "baroclinic momentum equations" written in vector form is:

$$\frac{\mathbf{u}' - \mathbf{u}^{n-1}}{\tau} + \tilde{\mathcal{L}}_U(\mathbf{u}^n) - f\hat{\mathbf{z}} \times \overline{\mathbf{u}}^{\alpha\gamma} = -\frac{1}{\rho_o}\nabla\overline{p_h} + \mathcal{F}_H(\mathbf{u}^{n-1}) + \mathcal{F}_V(\overline{\mathbf{u}}^{\lambda}) \tag{94}$$

where $\tau = 2\Delta t$, $\tilde{\mathcal{L}}_U$ represents the advection operator plus metric terms acting on both components of velocity, and $\mathcal{F}_H$ and $\mathcal{F}_V$ are now horizontal vectors. The overbars indicate various averages over the three time levels for the semi-implicit treatment of the Coriolis, hydrostatic pressure gradient, and vertical mixing terms. The velocities are averaged as:

$$\overline{\mathbf{u}}^{\alpha\gamma} = \alpha\mathbf{u}' + \gamma\mathbf{u}^n + (1-\alpha-\gamma)\mathbf{u}^{n-1} \tag{95}$$

$$\overline{\mathbf{u}}^{\lambda} = \lambda\mathbf{u}' + (1-\lambda)\mathbf{u}^{n-1} \tag{96}$$

where α, γ, λ are coefficients used to vary the time-centering of the velocities. The averaging for the hydrostatic pressure $\overline{p_h}$ is discussed in Sec. 3.5.2. To maintain an accurate time discretization of geostrophic balance, it is important that, in the averaging over the three time levels, the velocity and pressure are centered at the same time, i.e. if they are centered at time (n), then the coefficients for the variables at times $(n+1)$ and $(n-1)$ must be equal.

2) Next the vertical average of $\mathbf{u}'$ is subtracted. The result is the baroclinic velocity:

$$\tilde{\mathbf{u}}'_k = \mathbf{u}'_k - \frac{1}{H_U}\sum_{k'=1}^{km} dz_{k'}\ \mathbf{u}'_{k'} \tag{97}$$

3) Finally, the barotropic system is solved for the barotropic velocity at the new time $\mathbf{U}^{n+1}$ (see Sec. 4.5), and this is added to the normalized auxiliary velocities to obtain the full velocities at

the new time:

$$\mathbf{u}_k^{n+1} = \tilde{\mathbf{u}}_k' + \mathbf{U}^{n+1} \tag{98}$$

The barotropic velocity is defined by

$$\begin{aligned} \mathbf{U} &= \frac{1}{H+\eta}\int_{-H}^{\eta} dz\ \mathbf{u}(z) \\ &\approx \frac{1}{H}\int_{-H}^{0} dz\ \mathbf{u}(z) = \frac{1}{H_U}\sum_{k=1}^{km} dz_k\ \mathbf{u}_k\ . \end{aligned} \tag{99}$$

where η is the displacement of the free surface relative to $z = 0$. As discussed in Secs. 4.5 and 6.3, in the current version of the POP model we assume $\eta \ll dz_1$ and approximate the barotropic velocity as the vertical integral from $z = -H$ to $z = 0$ in (99).

4.4. Baroclinic Momentum Equations

The time discretization of the baroclinic momentum equations for leapfrog steps is given by (94). For forward and backward steps, the time discretization is

Forward steps:

$$\begin{aligned} &\frac{\mathbf{u}'^{*} - \mathbf{u}^n}{\tau} + \tilde{\mathcal{L}}_U(\mathbf{u}^n) - f\hat{\mathbf{z}}\times\overline{\mathbf{u}}^{\theta} = -\frac{1}{\rho_o}\nabla p_h^n + \mathcal{F}_H(\mathbf{u}^n) + \mathcal{F}_V(\overline{\mathbf{u}}^{\lambda'}) \\ &\overline{\mathbf{u}}^{\theta} = \theta\mathbf{u}'^{*} + (1-\theta)\mathbf{u}^n \\ &\overline{\mathbf{u}}^{\lambda'} = \lambda'\mathbf{u}'^{*} + (1-\lambda')\mathbf{u}^n \end{aligned} \tag{100}$$

Backward steps:

$$\begin{aligned} &\frac{\mathbf{u}' - \mathbf{u}^n}{\tau} + \tilde{\mathcal{L}}_U(\mathbf{u}'^{*}) - f\hat{\mathbf{z}}\times\overline{\mathbf{u}}^{\theta} = -\frac{1}{\rho_o}\nabla p_h^{*} + \mathcal{F}_H(\mathbf{u}'^{*}) + \mathcal{F}_V(\overline{\mathbf{u}}^{\lambda'}) \\ &\overline{\mathbf{u}}^{\theta} = \theta\mathbf{u}' + (1-\theta)\mathbf{u}^n \\ &\overline{\mathbf{u}}^{\lambda'} = \lambda'\mathbf{u}' + (1-\lambda')\mathbf{u}^n \end{aligned} \tag{101}$$

where $\tau = \Delta t$, and p_h^* is the predicted hydrostatic pressure from the forward step. $\mathbf{u}'$ is the unnormalized momentum, as in (94) and (97), and $\mathbf{u}'^*$ is the same quantity predicted by the forward step only. $\theta = 0.5$ is hardwired into the code. This choice was made (rather than choosing $\theta = 0$ where both Coriolis and pressure gradient would be centered and time n) because the forward step is unstable if the Coriolis terms is treated explicitly.

4.4.1. Pressure Averaging. The method of Brown and Campana (1978) is used for the semi-implicit treatment of the hydrostatic pressure gradient on leapfrog timesteps, where the averaging of the pressure over the three time levels in (94) is given by

$$\frac{1}{\rho_o}\nabla\overline{p_h} = \frac{1}{\rho_o}\nabla\left(\frac{1}{4}p^{n+1} + \frac{1}{2}p^n + \frac{1}{4}p^{n-1}\right) \tag{102}$$

This choice of coefficients allows for a factor of two increase in the CFL limit associated with internal gravity waves, and hence a factor of two increase in timestep if internal gravity waves are the controlling factor (see Maltrud et al., 1998, Sec. 2.3 for a simple proof of this property).

The new pressure at time $(n+1)$ is obtained by updating the tracer transport equations for the temperature Θ and salinity S before the baroclinic momentum equations are solved. Then the density and hydrostatic pressure at the new time can be diagnosed from the equation of state and the hydrostatic equation. An approximation is used in the code for efficiency if explicit vertical mixing is used with convective adjustment. With explicit vertical mixing the loops over vertical levels in the tracer and momentum updates are fused, (Note: this will no longer be true in POP 2.0) but the new pressure can still be evaluated if the tracers are updated first at each level. However, the convective adjustment is done afterwards, outside the k loop, and this modifies the new Θ and S and hence the new pressure. So, in this case the new pressure before convective adjustment is used in (102). With implicit vertical mixing the vertical loops over tracer and momentum are separated, and the implicit vertical diffusion of tracers is done after the tracer loop and before the momentum loop, so in that case the exact pressure at the new time is used in (102). Equation (102) applies to leapfrog timesteps. On forward steps the time n pressure is used, and on backward steps the predicted pressure from the first pass is used (see (100), (101)).

4.4.2. Semi-Implicit Treatment of Coriolis and Vertical Friction Terms. A semi-implicit treatment of the Coriolis terms can allow a somewhat longer timestep due to filtering of inertial waves and barotropic Rossby waves, but the main motivation in POP is to damp the Rossby-wave computational mode which appears in the implicit free-surface formulation of the barotropic equations (see Sec. 4.5 and Dukowicz and Smith 1994, Eqn. 43). Since the barotropic equations are the exact vertical average of the momentum equations, the Coriolis terms in the baroclinic equations must also be treated semi-implicitly. While it is possible to run POP with explicit treatment of the Coriolis terms, we strongly recommend against this because of the above-mentioned computational mode. The following discussion assumes the Coriolis terms are treated semi-implicitly.

The simultaneous semi-implicit treatment of both Coriolis and vertical mixing terms leads to a coupled system where both components of velocity must be solved for simultaneously. To avoid this, we employ an operator splitting which maintains the second-order accuracy of the time discretization. To illustrate this splitting, we write the momentum equations in the matrix form

$$\begin{aligned}
&\mathbf{u}' - \mathbf{u}^{n-1} + \tau \mathsf{B}\overline{\mathbf{u}}^{\alpha\gamma} = \tau \mathbf{F} + \tau \mathsf{A}(\mu)\overline{\mathbf{u}}^{\lambda} \\
&\tau = 2\Delta t \ , \quad \mathbf{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} \ , \quad \mathsf{B} = \begin{pmatrix} 0 & -f \\ f & 0 \end{pmatrix} \ , \quad \mathsf{A}(\mu) = \delta_z \mu \delta_z
\end{aligned} \tag{103}$$

where $\mathbf{u}$ is the velocity vector organized in block form and B is a 2×2 matrix in the space of the two velocity components. $\mathbf{F}$ represents all other terms in the momentum equation that are treated explicitly (advection, metric, pressure gradient and horizontal diffusion terms). The time-averaging of the velocities $\overline{\mathbf{u}}^{\alpha\gamma}$ and $\overline{\mathbf{u}}^{\lambda}$ are given by (95) and (96). Equation (103) can be rewritten as

$$\begin{aligned}
[\mathbf{I} - \lambda\tau \mathsf{A}(\mu) + \alpha\tau \mathsf{B}]\Delta\mathbf{u} &= \tau\{\mathbf{F} - \mathsf{B}[\gamma \mathbf{u}^n + (1-\gamma)\mathbf{u}^{n-1}] + \mathsf{A}(\mu)\mathbf{u}^{n-1}\} \equiv \tau \mathbf{F}' \\
\Delta\mathbf{u} &\equiv \mathbf{u}' - \mathbf{u}^{n-1}
\end{aligned} \tag{104}$$

where $\mathbf{I}$ is the identity matrix. The operator splitting is given by

$$[\mathbf{I} - \lambda\tau \mathsf{A}(\mu) + \alpha\tau \mathsf{B}] = [\mathbf{I} + \alpha\tau \mathsf{B}][\mathbf{I} - \lambda\tau \mathsf{A}(\mu)] + \mathcal{O}(\tau^2) \tag{105}$$

and thus second-order accuracy in time is maintained by dropping the $\mathcal{O}(\tau^2)$ terms and solving the simpler system

$$[\mathbf{I} - \lambda\tau \mathsf{A}(\mu)]\Delta\mathbf{u} = (\mathbf{I} + \alpha\tau\mathsf{B})^{-1}\tau\mathbf{F}' \tag{106}$$

The r.h.s. of (106) can be evaluated analytically using

$$(\mathbf{I} + \alpha\tau\mathsf{B})^{-1} = \frac{1}{1 + (\alpha\tau f)^2}(\mathbf{I} - \alpha\tau\mathsf{B}) \ . \tag{107}$$

Note again that the surface forcing is contained only on the r.h.s. of (106). Furthermore, the quadratic bottom drag term (58) is evaluated at the lagged time $(n-1)$ and also appears only on the r.h.s. By lagging the bottom drag term, the tridiagonal systems for u_x and u_y are linearized and decoupled, which greatly facilites their solution. Again, $\lambda = 0$ is the explicit vertical mixing case. For implicit vertical mixing, $\lambda > 0.5$ is required, and the code is typically run with $\lambda = 1$. In the barotropic equations (Sec. 4.5) we choose $\alpha = \gamma = 1/3$, which is hardwired in the code, and it is clear from (95) that the Coriolis terms are centered at time n.

On forward and backward steps the splitting is given by

Forward step:

$$\mathbf{u}'^{*} - \mathbf{u}^n + \tau\mathsf{B}\overline{\mathbf{u}}^{\theta} = \tau\mathbf{F} + \tau\mathsf{A}(\mu)\overline{\mathbf{u}}^{\lambda'}$$

or

$$\begin{aligned} &[\mathbf{I} - \lambda'\tau\mathsf{A}(\mu) + \theta\tau\mathsf{B}]\Delta\mathbf{u} = \tau[\mathbf{F}^n - \mathsf{B}\mathbf{u}^n + \mathsf{A}(\mu)\mathbf{u}^n] \\ &\Delta\mathbf{u} \equiv \mathbf{u}'^{*} - \mathbf{u}^n \end{aligned} \tag{108}$$

Backward step:

$$\mathbf{u}' - \mathbf{u}^n + \tau\mathsf{B}\overline{\mathbf{u}}^{\theta} = \tau\mathbf{F} + \tau\mathsf{A}(\mu)\overline{\mathbf{u}}^{\lambda'}$$

or

$$\begin{aligned} &[\mathbf{I} - \lambda'\tau\mathsf{A}(\mu) + \theta\tau\mathsf{B}]\Delta\mathbf{u} = \tau[\mathbf{F}^{*} - \mathsf{B}\mathbf{u}^n + \mathsf{A}(\mu)\mathbf{u}^n] \\ &\Delta\mathbf{u} \equiv \mathbf{u}' - \mathbf{u}^n \end{aligned} \tag{109}$$

where $\tau = \Delta t$, $\mathbf{F}^n$ represents the advection and hydrostatic pressure gradient terms evaluated at time n, and $\mathbf{F}^{*}$ represents the same terms evaluated using the predicted variables from the forward step. In (108) and (109) $\overline{\mathbf{u}}^{\lambda'}$ is defined as in (100) and (101), respectively. Now employing the operator splitting

$$[\mathbf{I} - \lambda'\tau\mathsf{A}(\mu) + \theta\tau\mathsf{B}] = [\mathbf{I} + \theta\tau\mathsf{B}][\mathbf{I} - \lambda'\tau\mathsf{A}(\mu)] + \mathcal{O}(\tau^2) \tag{110}$$

the tridiagonal systems analogous to (106) are:

Forward step:

$$\begin{aligned} &[\mathbf{I} - \lambda'\tau\mathsf{A}(\mu)]\Delta\mathbf{u} = (\mathbf{I} + \theta\tau\mathsf{B})^{-1}\tau[\mathbf{F}^n - \mathsf{B}\mathbf{u}^n + \mathsf{A}(\mu)\mathbf{u}^n] \\ &\Delta\mathbf{u} \equiv \mathbf{u}'^{*} - \mathbf{u}^n \end{aligned} \tag{111}$$

Backward step:

$$\begin{aligned} &[\mathbf{I} - \lambda'\tau\mathsf{A}(\mu)]\Delta\mathbf{u} = (\mathbf{I} + \theta\tau\mathsf{B})^{-1}\tau[\mathbf{F}^{*} - \mathsf{B}\mathbf{u}^n + \mathsf{A}(\mu)\mathbf{u}^n] \\ &\Delta\mathbf{u} \equiv \mathbf{u}' - \mathbf{u}^n \end{aligned} \tag{112}$$

Again, $\theta = 0.5$ is hardwired into the code.

4.5. Barotropic Equations

POP uses the implicit free-surface formulation of the barotropic equations developed by Dukowicz and Smith (1994), and this formulation is presented here. Other possible options are the rigid-lid streamfunction approach (Bryan 1969), the rigid-lid surface pressure approach (Smith et al., 1992, Dukowicz et al., 1993), and the explicit free-surface method (Killworth et al., 1991), which involves subcycling the barotropic mode with a smaller timestep than that used in the baroclinic equations.

The prognostic equation for the barotropic velocity, defined by (99), is obtained by vertically integrating the momentum and continuity equations. The barotropic momentum equation in block form, analogous to (103), is given by:

$$\begin{aligned} &\mathbf{U}^{n+1} - \mathbf{U}^{n-1} + \tau \mathsf{B} \overline{\mathbf{U}}^{\alpha\gamma} = -\tau g \nabla \overline{\eta}^{\alpha\gamma} + \tau \mathbf{F}_B \\ &\overline{\mathbf{U}}^{\alpha\gamma} = \alpha \mathbf{U}^{n+1} + \gamma \mathbf{U}^n + (1-\alpha-\gamma)\mathbf{U}^{n-1} \\ &\overline{\eta}^{\alpha\gamma} = \alpha \eta^{n+1} + \gamma \eta^n + (1-\alpha-\gamma)\eta^{n-1} \end{aligned} \tag{113}$$

where $\eta = p_s/\rho_0 g$ is the surface height, and $\mathbf{F}_B$ is the vertical integral of all terms besides the time-tendency, Coriolis, and surface pressure gradient terms in the momentum equation.

4.5.1. Linear Free Surface Model. A prognostic equation for the free surface height η is obtained by vertically integrating the continuity equation

$$\int_{-H}^{\eta} dz\, (\nabla \cdot \mathbf{u} + \frac{\partial w}{\partial z}) \;=\; \frac{\partial}{\partial t}\eta + \nabla \cdot (H+\eta)\mathbf{U} - q_W = 0\,, \tag{114}$$

where we have used the surface boundary condition on the vertical velocity:

$$w(\eta) \;=\; d\eta/dt - q_W = \frac{\partial}{\partial t}\eta + \mathbf{u}(\eta)\cdot\nabla\eta - q_W\,, \tag{115}$$

and $\mathbf{U}$ is the vertically-averaged horizontal velocity:

$$\mathbf{U} \;=\; \frac{1}{H+\eta}\int_{-H}^{\eta} dz\, \mathbf{u}(z) \tag{116}$$

where $w(\eta)$ and $\mathbf{u}(\eta)$ are the vertical and horizontal velocities at the surface, and q_W is freshwater flux. This result is derived using Leibnitz's Theorem to interchange the order of integration and differentiation. A difficulty with this form of the barotropic continuity equation is that, in the implicit time discretization of the barotropic equations, the term proportional to η inside the divergence in (114) leads to a nonlinear elliptic system, and standard solution methods such as conjugate gradient algoithms cannot be directly applied to it. To avoid this, POP uses a linearized form of the barotropic continuity equation which is derived as follows. Integrating the continuity equation over depth as before, but modifying the boundary condition (115) by dropping the term involving $\nabla\eta$ (which can be shown to be of order $|\eta|/dz_1$ relative to the time tendency term), we find:

$$\int_{-H}^{\eta} dz\,(\nabla\cdot\mathbf{u}+\frac{\partial w}{\partial z}) \;=\; \frac{\partial}{\partial t}\eta+\nabla\cdot H\mathbf{U}+\int_0^{\eta} dz\;\nabla\cdot\mathbf{u}-q_W=0 \qquad (117)$$

$$w(\eta) \;=\; \frac{\partial}{\partial t}\eta-q_W \qquad (118)$$

$$\mathbf{U} \;=\; \frac{1}{H}\int_{-H}^{0} dz\;\mathbf{u}(z) \qquad (119)$$

To obtain the linearized barotropic continuity equation we drop the term $\int_0^\eta dz\;\nabla\cdot\mathbf{u}$ in (117), which corresponds to neglecting the horizontal mass flux between $z=0$ and $z=\eta$:

$$\frac{\partial}{\partial t}\eta+\nabla\cdot H\mathbf{U}=q_W \quad . \qquad (120)$$

This derivation makes it clear that in the advection operators the horizontal mass flux between $z=0$ and $z=\eta$ must be neglected in order to be consistent with (120). So there are four ingredients to the linear free surface model: 1) the barotropic continuity equation is given by (120) instead of (114); 2) the barotropic velocity is given by (119) instead of (116); 3) the vertical velocity at the surface, which is used to integrate the continuity equation from the top down, is given by (118) instead of (115); and 4) the horizontal mass fluxes between $z=0$ and $z=\eta$ should be neglected in the advection operators and when integrating the continuity equation to find the vertical velocites. In the discrete equations this means that the horizontal mass fluxes in the surface cells are proportional to the full cell height dz_1, rather than $dz_1+\eta$ (see Sec. 6.3). This linear approximation is valid provided the surface displacement is small compared to the depth of the full ocean: $|\eta|\ll H$, and in the discrete equations the surface displacement must also be small compared to the depth of the surface layer: $|\eta|\ll dz_1$.

With a non-zero freshwater flux q_W the mean volume of the ocean is not constant, even though the velocity field is divergence free. Integrating (120) over horizontal area, we find

$$\frac{d}{dt}\int dV=\int da\;\frac{\partial}{\partial t}\eta=\int da\;q_W \qquad (121)$$

where $da=dxdy$ is the horizontal area element. So the total volume of the ocean will change unless the surface-integrated freshwater flux vanishes.

4.5.2. Time Discretization of the Barotropic Equations. In the implicit free-surface formulation the continuity equation (120) is discretized in time using a forward step as follows:

$$\frac{\eta^{n+1}-\eta^n}{\Delta t}+\nabla\cdot H\overline{\mathbf{U}}^{\alpha'}=q_W^n$$
$$\overline{\mathbf{U}}^{\alpha'}=\alpha'\mathbf{U}^{n+1}+(1-\alpha')\mathbf{U}^n \qquad (122)$$

where α' is a coefficient used to vary the time centering of the velocity in the continuity equation. The barotropic equations support three types of linear waves: two gravity waves (one in each horizontal direction) and one Rossby wave. In a pure leapfrog discretization, there would be three computational modes, one associated with each of these waves. By choosing the forward discretization (122), one computational mode is eliminated, leaving one gravity-like and one Rossby-like computational mode. The Rossby computational mode is damped by the discretization scheme if the Coriolis terms are treated implicitly, and the gravity-wave computational mode is strongly damped if α' is close to 1. In Dukowicz and Smith (1994) it was shown that the optimal set of time-centering coefficients which maximally damps the computational modes, minimally damps the physical modes, and minimally distorts the phase speed of the physical modes is given by the parameter set:

$$\alpha = \gamma = \frac{1}{3}\,, \quad \alpha' = 1\,. \tag{123}$$

Thus in (113)

$$\overline{\eta}^{\alpha\gamma} = \frac{1}{3}(\eta^{n+1} + \eta^n + \eta^{n-1})$$

and similarly for $\overline{\mathbf{u}}^{\alpha\gamma}$. These choices are hardwired into the code. The physical gravity waves are damped at small space and time scales in this implicit scheme, but the physical Rossby waves are essentially unaffected.

By inserting the barotropic momentum equation (113) into the continuity equation (122) we obtain an elliptic equation for the surface height at the new time η^{n+1}. However, due to the presence of the Coriolis terms, the resulting elliptic operator is not symmetric, making it much more difficult to invert, because standard solvers such as conjugate gradient methods require symmetric positive-definite linear operators. This problem can be overcome by using an operator splitting technique that maintains the second-order accuracy of the time discretization scheme (Smith et al., 1992; Dukowicz et al., 1993). Defining an auxiliary velocity

$$\hat{\mathbf{U}} \equiv \mathbf{U}^{n+1} + \alpha\tau g\nabla(\eta^{n+1} - \eta^{n-1}) \tag{124}$$

equation (113) can be written:

$$(\mathbf{I} + \alpha\tau\mathsf{B})(\hat{\mathbf{U}} - \mathbf{U}^{n-1}) = \tau\{\mathbf{F}_B - \mathsf{B}[\gamma\mathbf{U}^n + (1-\gamma)\mathbf{U}^{n-1}] - g\nabla[\gamma\eta^n + (1-\gamma)\eta^{n-1}]\} + \mathcal{O}(f\tau^3) \tag{125}$$

Dropping the $\mathcal{O}(f\tau^3)$ terms (which are the same order as the time truncation error in this second-order scheme), and using the continuity equation (122) with $\alpha' = 1$, we arrive at the elliptic system

$$[\nabla\cdot H\nabla - \frac{1}{g\alpha\tau\Delta t}]\eta^{n+1} = \nabla\cdot H[\frac{\hat{\mathbf{U}}}{g\alpha\tau} + \nabla\eta^{n-1}] - \frac{\eta^n}{g\alpha\tau\Delta t} - \frac{q_W^n}{g\alpha\tau} \tag{126}$$

The procedure is to first solve (125) for $\hat{\mathbf{U}}$, then solve (126) for η^{n+1} (POP actually solves for the surface pressure $p_s = \rho_0 g\eta$), and finally use (124) to obtain $\mathbf{U}^{n+1}$. This system is solved in POP on leapfrog timesteps with $\alpha = 1/3$ using a diagonally preconditioned conjugate gradient algorithm described in Sec. 4.5.3. Note that the terms dropped in (125) are only $\mathcal{O}(\tau^3)$ if the timestep is small compared to the inertial period $1/f$. We therefore recommend that the model timestep be no greater than about two hours. The scheme is stable for longer timesteps, but the barotropic mode will start to become inaccurate as the timestep is increased above two hours. The divergence on the r.h.s. in (126) is discretized with the correct B-grid discretization, i.e., the quantity in brackets is transversely averaged as in (36). The Laplacian-like operator on the l.h.s. in (126) is a nine-point stencil with the correct B-grid discretization:

$$\nabla \cdot H\nabla\eta = \frac{1}{\Delta_y}\delta_x\overline{[\Delta_y H_U \delta_x \overline{\eta}^y]}^y + \frac{1}{\Delta_x}\delta_y\overline{[\Delta_x H_U \delta_y \overline{\eta}^x]}^x . \tag{127}$$

Unlike the friction and diffusion operators, this operator cannot be approximated as a five-point stencil, because doing so would violate the energetic balance between pressure work and change in potential energy (see Dukowicz and Smith, 1994, and Sec. 3.6). Since the 9-point B-grid operator in (127) has a checkerboard null space (see Sec. 6.3.1), the solution of (126) is prone to have local patches of checkerboard noise. Strictly speaking, the operator on the l.h.s. of (126) has no checkerboard null space due to the presence of the extra diagonal term (second term in brackets). However, if the solution in some region is in a near steady-state, this diagonal term is cancelled by the next to last term on the r.h.s., and checkerboard noise may appear. This is particularly true in isolated bays where the solution is only weakly coupled to the interior. On the other hand, the checkerboard noise has little effect on the dynamics because only the gradient of surface height enters the barotropic momentum equation, and the B-grid gradient operator (35) does not see a checkerboard field. The only way a checkerboard surface height field can affect the dynamics is through the vertical velocity at the surface (45), which depends on the change in surface height, and is used as the surface boundary condition to integrate the continuity equation to find the advection velocities. However, experience has so far shown that this does not lead to serious problems with the model simulations.

It should be noted that the solution for surface height will only satisfy the continuity equation (122) to the extent that the solution of (126) has converged in the iterative solution of the elliptic solver. As discussed in Sec. 3.3.1, this can lead to a small non-divergent mass flux in the bottom-most ocean cell when the 3-D continuity equation is integrated from the top down with w given by (118) at the surface, as discussed in Sec. 3.3.1. In practice, we suggest that the convergence criterion for the iterative solver be chosen so that the global mean balance between pressure work and change in potential energy is accurate to within about three significant digits (see Sec. 3.6 on energetic consistency).

The time discretization for forward and backwards steps, corresponding to (113) and (122) are given by:

Forward step:

$$\begin{aligned}
&\mathbf{U}^* - \mathbf{U}^n + \tau \mathsf{B}\overline{\mathbf{U}}^\theta = -\tau g \nabla \overline{\eta}^\theta + \tau \mathbf{F}_B^n \\
&\frac{\eta^* - \eta^n}{\Delta t} + \nabla \cdot H\mathbf{U}^* = q_W^n \\
&\overline{\mathbf{U}}^\theta = \theta \mathbf{U}^* + (1-\theta)\mathbf{U}^n \\
&\overline{\eta}^\theta = \theta \eta^* + (1-\theta)\eta^n
\end{aligned} \tag{128}$$

Backward step:

$$\begin{aligned}
&\mathbf{U}^{n+1} - \mathbf{U}^{n} + \tau \mathsf{B}\overline{\mathbf{U}}^{\theta} = -\tau g \nabla \overline{\eta}^{\theta} + \tau \mathbf{F}_B^{*} \\
&\frac{\eta^{n+1} - \eta^{n}}{\Delta t} + \nabla \cdot H \mathbf{U}^{n+1} = q_W^{n} \\
&\overline{\mathbf{U}}^{\theta} = \theta \mathbf{U}^{n+1} + (1-\theta)\mathbf{U}^{n} \\
&\overline{\eta}^{\theta} = \theta \eta^{n+1} + (1-\theta)\eta^{n}
\end{aligned} \tag{129}$$

where we have assumed $\alpha' = 1$ as in (122) and (123). $\mathbf{U}^*$ and η^* are the predicted barotropic velocity and surface height from the forward step. $\mathbf{F}_B^n$ contains all terms other than the Coriolis, surface pressure gradient and time-tendency terms in the barotropic momentum equation, all evaluated at time n, and $\mathbf{F}_B^*$ contains the same terms but evaluated using the prognostic variables predicted by the forward step. The operator splittings for the forward and backward steps, analogous to (124) and (125) are:

Forward step:

$$\begin{aligned}
&\hat{\mathbf{U}} \equiv \mathbf{U}^{*} + \theta \tau g \nabla (\eta^{*} - \eta^{n}) \\
&(\mathbf{I} + \theta \tau \mathsf{B})(\hat{\mathbf{U}} - \mathbf{U}^{n}) = \tau \{\mathbf{F}_B^{n} - \mathsf{B}\mathbf{u}^{n} - g \nabla \eta^{n}\} + \mathcal{O}(f\tau^{3})
\end{aligned} \tag{130}$$

Backward step:

$$\begin{aligned}
&\hat{\mathbf{U}} \equiv \mathbf{U}^{n+1} + \theta \tau g \nabla (\eta^{n+1} - \eta^{*}) \\
&(\mathbf{I} + \theta \tau \mathsf{B})(\hat{\mathbf{U}} - \mathbf{U}^{n}) = \tau \{\mathbf{F}_B^{*} - \mathsf{B}\mathbf{u}^{n} - g[\theta \nabla \eta^{*} + (1-\theta)\nabla \eta^{n}]\} + \mathcal{O}(f\tau^{3})
\end{aligned} \tag{131}$$

Finally, the elliptic equations for the forward and backward steps analogous to (126) are given by:

Forward step:

$$[\nabla \cdot H \nabla - \frac{1}{g\theta\tau\Delta t}]\eta^{*} = \nabla \cdot H[\frac{\hat{\mathbf{U}}}{g\theta\tau} + \nabla \eta^{n}] - \frac{\eta^{n}}{g\theta\tau\Delta t} - \frac{q_W^{n}}{g\theta\tau} \tag{132}$$

Backward step:

$$[\nabla \cdot H \nabla - \frac{1}{g\theta\tau\Delta t}]\eta^{n+1} = \nabla \cdot H[\frac{\hat{\mathbf{U}}}{g\theta\tau} + \nabla \eta^{*}] - \frac{\eta^{n}}{g\theta\tau\Delta t} - \frac{q_W^{n}}{g\theta\tau} \tag{133}$$

4.5.3. Conjugate Gradient Algorithm. The most efficient method we have found for solving the elliptic system of barotropic equations (126) is to use a standard Preconditioned Conjugate Gradient solver. The algorithm consists of the following steps to solve the system $Ax = b$ given by (126) multiplied by the T-cell area. A is the symmetric positive-definite Laplacian type operator on the l.h.s. in (126) (it is important to note that if (126) is not multiplied by the T-cell area then the operator is not symmetric), b is the r.h.s., and x is the solution. Lower-case Greek and Roman letters are used, respectively for scalars and 2-D arrays, and (x, y) denotes a dot-product: $(x, y) = \sum_{ij} x_{ij} y_{ij}$.

PCG Algorithm:

$$\text{Initial guess } x_o \tag{134}$$

Compute initial residual $r_o = b - Ax_o$
Set $\beta_o = 1, s_o = 0$
For $k = 1, 2, ..., k_{max}$ do
  $r'_{k-1} = Zr_{k-1}$
  $\beta_k = (r_{k-1}, r'_{k-1})$
  $s_k = r'_{k-1} + (\beta_k/\beta_{k-1})s_{k-1}$
  $s'_k = As_k$
  $\alpha_k = \beta_k/(s_k, s'_k)$
  $x_k = x_{k-1} + \alpha_k s_k$
  $r_k = r_{k-1} - \alpha_k s'_k$
  Exit if converged: $(r_k, r_k)^{\frac{1}{2}} < \epsilon\,\overline{a}$
End do

Here Z is a preconditioning matrix. This is usually taken to be a diagonal preconditioner $Z = C_0^{-1}$, where C_0 is the diagonal matrix composed of the central coefficient in the 9-point stencil corresponding to the operator in (126). The code is set up to use a more sophisticated preconditioner consisiting of a 9-point operator which is a local approximate inverse of the true operator (Smith et al., 1992). This preconditioner is expensive to compute but is time independent and can be computed offline and read in by the code. To improve the initial guess, x_o is linearly extrapolated in time from the solutions at the two previous timesteps. In the convergence criterion $(r_k, r_k)^{\frac{1}{2}} < \epsilon\,\overline{a}$, $\overline{a}$ is the rms T-cell area. This normalization is somewhat arbitrary, but is chosen so that ϵ has the same dimensions as the operator (126) before it is multiplied by the T-cell area averaged over surface ocean points. As discussed in Sec. 4.5.2, we suggest that ϵ be chosen such that the global mean balance between pressure work and change in potential energy is accurate to within about three significant digits. Typically this is achieved for values of ϵ between 10^{-12} and 10^{-13}.

5. Subgrid-scale Parameterizations

5.1. Parameterizations of Horizontal Tracer Diffusion

5.1.1. Laplacian Horizontal Diffusivity. The spatial discretization of the standard Laplacian horizontal tracer diffusion with a spatially varying tracer diffusivity A_H is described in Sec. 3.3.2.

5.1.2. Biharmonic Horizontal Diffusivity. Biharmonic horizontal tracer diffusion is implemented by applying the Laplacian operator (47) twice. Specifically,

$$\mathcal{D}_H^{(4)}(\varphi) = \mathcal{D}_H^{(2)}(A_H^{(4)}\mathcal{D}_H^{(2)}(\varphi)) \tag{135}$$

where the superscripts (4) and (2) refer to biharmonic and harmonic (Laplacian) operators, and $\mathcal{D}_H^{(2)}$ is given by (47) with $A_H = 1$. $A_H^{(4)}$ is the biharmonic diffusivity, which may be spatially varying, and should be negative for positive-definite diffusion. Here the biharmonic diffusivity is sandwiched between the two applications of the Laplacian operator. An alternate approach (not implemented in the code), following Griffies and Hallberg (2000), is to include the square-root of $-A_H$ inside the second derivative of each $\mathcal{D}_H^{(2)}$ operator.

5.1.3. The Gent-McWilliams Parameterization. The transport equation of tracer φ is given by

$$\frac{\partial}{\partial t}\varphi + (\mathbf{u}+\mathbf{u}^*)\cdot\nabla\,\varphi + (w+w^*)\frac{\partial}{\partial z}\,\varphi = R(\varphi) + \mathcal{D}_V(\varphi), \tag{136}$$

where the bolus velocity induced by mesoscale eddies is parameterized, from Gent and McWilliams (1990), as

$$\mathbf{u}^* = \left(\nu\frac{\nabla\rho}{\rho_z}\right)_z, \qquad w^* = -\nabla\cdot\left(\nu\frac{\nabla\rho}{\rho_z}\right), \tag{137}$$

(henceforth subscripts x, y, z on ρ and tracers φ denote partial derivatives with respect to those variables). The Redi isoneutral diffusion operator (Redi, 1982) for small slope can be written as

$$R(\varphi) = \nabla_3\cdot\left(\mathbf{K}\cdot\nabla_3\,\varphi\right), \tag{138}$$

where the subscript 3 indicates the 3-dimensional gradient or divergence, i.e., $\nabla_3 = (\nabla, \frac{\partial}{\partial z})$. The symmetric tensor $\mathbf{K}$ is defined as

$$\mathbf{K} = \kappa_I\begin{pmatrix} 1 & 0 & -\rho_x/\rho_z \\ 0 & 1 & -\rho_y/\rho_z \\ -\rho_x/\rho_z & -\rho_y/\rho_z & |\nabla\rho|^2/\rho_z^2 \end{pmatrix}, \tag{139}$$

This tensor describes along-isopycnal diffusion that is isotropic in the two horizontal dimensions. The general anisotropic form of (139) is given in Smith (1999, Appendix B). The diffusivity κ_I is in general a function of space and time, and a parameterization for variable κ_I will be described at the end of this section. In POP, we write the bolus velocity in the skew-flux form (Griffies, 1998):

$$\mathbf{u}_3^*\cdot\nabla_3\,\varphi = \nabla_3\cdot\left(\mathbf{u}_3^*\,\varphi\right) = -\nabla_3\cdot\left(\mathbf{B}\cdot\nabla_3\,\varphi\right), \tag{140}$$

where we have used $\nabla_3\cdot\mathbf{u}_3^* = 0$. The subscript 3 on the velocity indicates the 3-dimensional velocity, i.e., $\mathbf{u}_3^* = (\mathbf{u}^*, w^*)$. The antisymmetric tensor $\mathbf{B}$ is given by

$$\mathbf{B} = \nu\begin{pmatrix} 0 & 0 & \rho_x/\rho_z \\ 0 & 0 & \rho_y/\rho_z \\ -\rho_x/\rho_z & -\rho_y/\rho_z & 0 \end{pmatrix}. \tag{141}$$

By substituting (141), the transport equation (136) can be written

$$\frac{\partial}{\partial t}\,\varphi + \mathbf{u}\cdot\nabla\varphi + w\frac{\partial}{\partial z}\,\varphi = \overline{R}(\varphi)\ + \mathcal{D}_V(\varphi),, \qquad \overline{R}(\varphi) = \nabla_3\cdot(\mathbf{K}+\mathbf{B})\cdot\nabla_3\,\varphi\,. \tag{142}$$

With a nonlinear equation of state, the isopycnal slopes $(L_x, L_y) = (-\rho_x/\rho_z, -\rho_y/\rho_z)$ that appear in (139) and (141) should be replaced with slopes along the local neutral surfaces (McDougall, 1987), which are given by:

$$L_x = -\frac{\alpha_p\Theta_x - \beta_p S_x}{\alpha_p\Theta_z - \beta_p S_z} \tag{143}$$

with a similar definition for L_y. Here $\alpha_p = -\partial\,\rho_p/\partial\Theta$ and $\beta_p = \partial\,\rho_p/\partial S$, where Θ and S are the model potential temperature and salinity, and ρ_p is the potential density referenced to the local pressure (or depth).

Discretization of the diffusion operator

The functional formalism for deriving positive-definite diffusive operators in ocean models was originally used in POP to implement the Gent-McWilliams parameterization. Later, a collaboration between GFDL and LANL led to an improved implementation that is described in Griffies et al, (1998). The main change in the new formulation is to discretize the neutral slopes (143) in a particular way that avoids exciting a nonlinear instability.

The discrete functional formalism is based on the property of the continuous dissipation operator (142) that it can be expressed as the functional derivative of the integral of a positive-definite functional $\mathcal{G}$:

$$
\begin{aligned}
\mathcal{G}[\varphi] &= \frac{1}{2}\int dV\,(\nabla_3\,\varphi)\cdot\mathbf{K}\cdot(\nabla_3\,\varphi) \\
&= \frac{1}{2}\int dV\,\kappa_I\left[(\varphi_x + L_x\varphi_z)^2 + (\varphi_y + L_y\varphi_z)^2\right] \qquad (144) \\
R(\varphi) &= -\frac{\delta}{\delta\varphi}\mathcal{G}[\varphi] \qquad (145)
\end{aligned}
$$

where δ denotes a functional derivative. That is, $\mathcal{G}$ is functional whose derivative with respect to a tracer field at a given point in space yields the isoneutral diffusive operator R at that point. Note that, as a consequence of the assumption of isotropic diffusion (139), the 3-D functional splits into two terms which are effectively 2-D, one in the $x-z$ plane and one in the $y-z$ plane.

The procedure for deriving the discrete operator is to first discretize the functional and then take its derivatives with respect to the velocity components at a given point on the compuational grid; this yields the positive-definite friction operator at that point. In order to discretize the functional on the POP grid, we need to define all the quantities appearing in $\mathcal{G}$ within each cell. Because gradients involve differences across cell faces, it is convenient to subdivide each T-cell into four subcells, as shown in Figure 5 (see caption), which shows a central T-cell, its four interior subcells, and the 8 surrounding subcells that contribute to the friction operator when the discrete functional derivative is taken with respect to the tracer value at the central point. A unique value of the tracer gradients and isoneutral slopes can be assigned to each subcell, and from these the complete discrete functional $\mathcal{G}$ can be constructed and summed over the grid. Associated with each subcell is the set of its three nearest points (known as the "triad" of points associated with that subcell), namely, the central point of the full cell to which it belongs, and the two nearest points in the horizontal and vertical directions. For example, the triad associated with the subcell in Fig. 5 labeled (o, ne) is the set of points labeled (o), (n), (e). All the derivatives of the tracer fields needed to construct the functional in a given subcell only involve the tracer values at its triad points. For example, the x and z derivatives of φ in the subcell labeled (o, ne), are given by $\varphi_x^{(o,ne)} = (\varphi^{(e)} - \varphi^{(o)})/h_x$ and $\varphi_z^{(o,ne)} = (\varphi^{(n)} - \varphi^{(o)})/h_z$, where h_x and h_z are defined in each

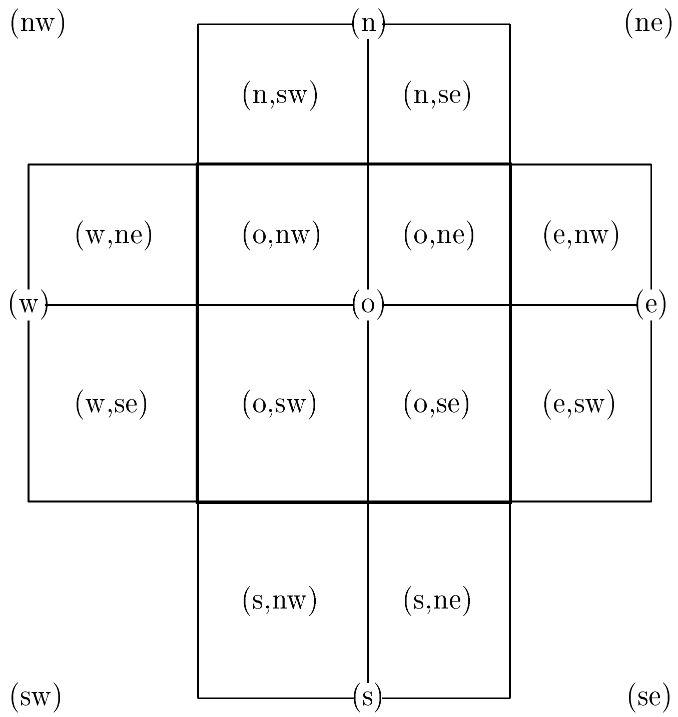

Figure 5: The discretization of the $x - z$ part of the isoneutral diffusion functional is accomplished by subdividing each T-cell (shown bounded by thick lines) into four subcells. The central tracer point (T-point) is denoted by "(o)" and the surrounding T-points by "(n)", "(s)", "(ne)", etc. (this is for notation only, points "(n)" and "(s)" are displaced in the vertical direction.) The four subcells surrounding the central point are labeled "(o,ne)", "(o,nw)", etc., and the subcells surrounding the T-cell to the east are labeled "(e,nw)", etc. Within each subcell a unique value is defined for all quantities (e.g., the gradients and slope factors appearing in (144) that are needed to evaluate the functional. Only the twelve subcells shown contribute to the friction operator at the central point (see text). (Note: this figure shows a case where the fundamental grid is constructed with U-points in the exact middle of T-cells in a vgrid with variable horizontal spacing. More commonly T-points are located at the exact center of U-cells. This choice is made during grid generation and is only determined in the code through the input grid file.) This figure can also be used for constructing the anisotropic dissipation functional (see Sec. 5.2.3) where the central point represents a U-point.

subcell as the distances between the central point (o) and the other two triad points in the x and z directions, respectively. Similarly, the derivatives in the subcell labeled (o, nw) are given by $\varphi_x^{(o,nw)} = (\varphi^{(o)} - \varphi^{(w)})/h_x$ and $\varphi_z^{(o,nw)} = (\varphi^{(n)} - \varphi^{(o)})/h_z$. The diffusion coefficient is assumed to have a constant value across all four subcells within a given full cell, but can vary from full-cell to full cell.

Using these definitions of the quantities in each subcell appearing in (144), it can be written in the discrete form:

$$\mathcal{G}[\varphi] = \frac{1}{2} \sum_{ij} \sum_{n=1}^{4} (\kappa_I)_{ij} V_{ijn} \left[(\varphi_x + L_x \varphi_z)^2 + (\varphi_y + L_y \varphi_z)^2 \right] \tag{146}$$

where subscripts ij run over all full cells, and subscripts n over subcells within a given full cell. $V_{ijn} = \frac{1}{4} h_x h_y dz(k)$ is the subcell volume (where h_y is the full cell width centered at the same point as h_x, and note that the height of each subcell is taken to be half its full-cell height $dz(k)$.)

The diffusion operator is then given by the derivative of the discrete functional with respect to the velocity at a given point (i', j'):

$$R_{i'j'} = -\frac{1}{V_{i'j'}^T} \frac{\partial}{\partial \varphi_{i'j'}} G[\varphi] \tag{147}$$

where $V_{i'j'}^T = A_{i'j'}^T \, dz(k)$ full-cell volume of cell (i, j, k), and $A_{i'j'}^T$ is the full-cell area. (Note: in the code $A^T = \text{TAREA}$, which is not exactly equal to the sum of the four subcell areas given above, however, the difference is small and is absorbed in an overall normalization factor.) When taking the derivative with respect to φ at the central point, only those subcells in the full functional whose triad points contain the central point will contribute to the derivative, and these are just the 12 subcells shown in Fig. 5. It is straightforward to take the derivative in (147) in these 12 subcells and assemble the results into a compact form for the diffusion operator. The result is:

$$\begin{aligned} V_{ij}^T \, R_{ij}(\varphi) \; &= \; \overline{(V/h_x)\kappa_I(\varphi_x + L_x \varphi_z)}^e - \overline{(V/h_x)\kappa_I(\varphi_x + L_x \varphi_z)}^w \\ &+ \; \overline{(V/h_z)\kappa_I(L_x \varphi_x + L_x^2 \varphi_z)}^n - \overline{(V/h_z)\kappa_I(L_x \varphi_x + L_x^2 \varphi_z)}^s \\ &+ \; \text{similar terms with } x \text{ replaced by } y \text{ everywhere.} \end{aligned} \tag{148}$$

Here overbars with superscripts e, w, etc., denote straight averages of the four subcells surrounding the east, west, etc., faces of the central full cell. In computing the slopes L_x and L_y given by (143) it is important that the coefficients α_p and β_p be evaluated at the pressure corresponding to the level of the central point of the triad in a given subcell (Griffies et al., 1998); this avoids exciting the nonlinear instability mentioned above. No-flux boundary conditions on the tracers are implemented by setting $\kappa_I = 0$ in all subcells adjacent to lateral and bottom boundaries. There is no way to derive the bolus transport term in (142) and (141) from a functional like the diffusive term, but the bolus term in (142) involves terms proportional to the off-diagonal terms in the diffusion tensor, and these are discretized in the same way. Thus we find for the full skew flux ($\overline{R}(\varphi)$ in Eq. 142)

$$\begin{aligned} V_{ij}^T \, \overline{R}_{ij}(\varphi) \; &= \; \overline{(V/h_x)[\kappa_I \varphi_x + (\kappa_I - \nu) L_x \varphi_z]}^e - \overline{(V/h_x)[\kappa_I \varphi_x + (\kappa_I - \nu) L_x \varphi_z]}^w \\ &+ \; \overline{(V/h_z)[(\kappa_I + \nu) L_x \varphi_x + \kappa_I L_x^2 \varphi_z]}^n - \overline{(V/h_z)[(\kappa_I + \nu) L_x \varphi_x + \kappa_I L_x^2 \varphi_z)}^s \\ &+ \; \text{similar terms with } x \text{ replaced by } y \text{ everywhere.} \end{aligned} \tag{149}$$

If the tracer and thickness diffusion coefficients are chosen equal ($\kappa_I = \nu$), then the second terms in brackets in the first line of (149) vanish and need not be computed. (Note: in the code, the above formulae are evaluated using the slope functions SLX $= -(h_x/h_z)L_x$, SLY $= -(h_y/h_z)L_y$, and the differences $\Delta_x\varphi = h_x\varphi_x$, etc.) Finally, it should be noted that the last terms in (149) involving L_x^2 (and L_y^2) are separated and treated implicitly, because the effective viscous coefficient can be large enough to violate explicit CFL conditions (Cox, 1987). These terms can be written

$$\begin{aligned}&\overline{(V/h_z)\kappa_I L_x^2\varphi_z}^n - \overline{(V/h_z)\kappa_I L_x^2\varphi_z}^s = \delta_z(\kappa'\delta_z\varphi)\ , \\ &\text{where}\quad \kappa' = (A^T)^{-1}\,\overline{(V/h_z)\kappa_I(L_x^2+L_y^2)}^n\end{aligned} \tag{150}$$

and the last line gives κ' on the top of the T-cell. κ' is then added to the vertical diffusivity κ in (48) and the vertical diffusion is solved implicitly. For this reason implicit vertical mixing must be used with the Gent-McWilliams parameterization. One of the design features of the diffusive operator is that with a linear equaton of state it should be zero when applied to $\varphi = \rho$. This condition is violated when the above term is treated implicitly while the remaining terms are treated explicitly, but in practice this leads to a very small error and is ignored.

Two tapering functions are often applied to the GM coefficient ν and the Redi mixing coefficient κ_I. The first is for physical reasons and reduces the GM coefficient in the upper boundary layer, see Large et al., (1997). Here the ocean is not nearly adiabatic, and so the GM parameterization does not apply; instead the isopynals tend to become more vertical because of the strong vertical mixing in the upper boundary layer. The tapering occurs where the depth is less than the local isopycnal slope multiplied by the local Rossby radius L_R defined by $L_R = |f|/c_0$ with $c_0 = 2$m/s and L_R is bounded by 15km $\leq L_R \leq$ 100km. Two forms for the factor are in the code, with the second being faster because there is no sin function.

$$\textit{tanh option}: f_1 = \frac{1}{2}\Big[1 + \sin\,\Big(\pi(\min(1, -z/(|\mathbf{L}|L_R)) - 0.5)\Big)\Big] \tag{151}$$

$$\textit{else}: f_1 = \frac{1}{2} + 2\Big(min(1, -z/(|\mathbf{L}|L_R)) - 0.5\Big)\Big(1 - |min(1, -z/(|\mathbf{L}|L_R)) - 0.5|\Big) \tag{152}$$

where $|\mathbf{L}| = (L_x^2 + L_y^2)^{\frac{1}{2}}$. The second factor is to ensure numerical stability when the isopycnals become too steep. There are four options to reduce the coefficients κ_I and ν for $|\mathbf{L}| \leq S_M^{(\kappa_I,\nu)}$, where the maximum slope allowed can be different for $S_M^{\kappa_I}$ and S_M^{ν}. The default values are $S_M^{\kappa_I} = S_M^{\nu} = 0.01$. The first option was described in Danabasoglu and McWilliams (1995), and the second is a faster version because there is no tanh function.

$$\textit{tanh option}: f_2 = \frac{1}{2}\Big[1 - \tanh(10|\mathbf{L}|/S_M^{(\kappa_I,\nu)} - 4)\Big] \tag{153}$$

for $|\mathbf{L}| > S_M^{(\kappa_I,\nu)}$, $\kappa_I = \nu = 0$.

$$\textit{no tanh option}: f_2 = \frac{1}{2}\Big[1 - \Big(2.5|\mathbf{L}|/S_M^{(\kappa_I,\nu)} - 1\Big)\Big(4 - |10|\mathbf{L}|/S_M^{(\kappa_I,\nu)} - 4|\Big)\Big] \tag{154}$$

with $f_2 = 1$ for $|\mathbf{L}| \leq 0.2 S_M^{(\kappa_I,\nu)}$ and $f_2 = 0$ for $|\mathbf{L}| \geq 0.6 S_M^{(\kappa_I,\nu)}$.

The third option is the clipping option described in Cox (1987) where the slopes are set to the maximum if they are diagnosed as larger. In this case, mixing occurs along the maximum allowed slope. This implies some cross isopycnal mixing when the slopes are steep. The default values are $S_M^{\kappa_I} = S_M^{\nu} = 0.01$.

The fourth option is described in Gerdes et al. (1991) and the factor is

$$\textit{Gerdes option} : f_2 = (S_M^{(\kappa_I,\nu)}/|\mathbf{L}|)^2 \tag{155}$$

when the slope is larger than the maximum slope. This option and the first two options retain the adiabatic nature of the parameterization. When $\kappa_I = \nu$ and $S_M^{\kappa_I} = S_M^{\nu}$, the code runs much faster because the terms proportional to $\kappa_I - \nu$ disappear. This is used in the CCSM ocean component. There, the KPP vertical mixing option is used, and the second option for f_1 is used, but only down to the first model level below the diagnosed depth of the boundary layer. This leaves the GM mixing as large as possible below the boundary layer. The second option for f_2 is used in the CCSM, but the maximum slopes allowed are much larger than the default values with $S_M^{\kappa_I} = S_M^{\nu} = 0.3$.

Variable GM diffusivity

Following Visbeck et al. (1997), it is assumed that

$$\kappa_I = \alpha \, \frac{L_E^2}{T_E} \, . \tag{156}$$

where the default value of the constant α is 0.13. In (156), the baroclinic length scale L_E and the Eady time scale T_E are defined as

$$L_E = \min\left(c/|f|, \sqrt{c/2\beta}\right) , \quad \frac{1}{T_E} = \frac{\max(|f|, \sqrt{2c\beta})}{\sqrt{Ri}} , \quad Ri = -\frac{g}{\rho_0 H}\int_{z_2}^{z_1} \frac{\rho_z}{|\mathbf{u}_z|^2}\,\mathrm{d}z \, , \tag{157}$$

where β is the meridional gradient of the Coriolis parameter and c is the first baroclinic wave speed given by $c^2 = -gH\int_{z_2}^{z_1} \rho_z \, \mathrm{d}z/\rho_0$, $z_1 = -50$m and $z_2 = -1000$m. L_E is also bounded below by the smaller horizontal side of each grid box. In the code, κ_I is set to be bounded by $3 \times 10^6 \leq \kappa_I \leq 2 \times 10^7 cm^2/s$ and κ_I is set to the lower bounding value where the model depth is less than -z_1. All values given here are presently hard-wired in the code.

This option is not presently used in the CCSM ocean configuration.

5.2. Parameterizations of Horizontal Viscosity

5.2.1. Laplacian Horizontal Viscosity. The spatial discretization of the standard Laplacian horizontal friction terms with a spatially varying viscosity A_M given by (29) is described in Sec. 3.4.3.

5.2.2. Biharmonic Horizontal Viscosity. In analogy to the biharmonic diffusion operator (135), the biharmonic viscosity is constructed by applying the Laplacian-like operator (28) twice. Specifically,

$$\mathcal{F}^{(4)}_{Hx}(u_x, u_y) = \mathcal{F}^{(2)}_{Hx}(A^{(4)}_M \mathcal{F}^{(2)}_{Hx}(u_x, u_y), A^{(4)}_M \mathcal{F}^{(2)}_{Hy}(u_x, u_y)) \tag{158}$$

with a similar expression for $\mathcal{F}^{(4)}_{Hy}$. The superscripts (4) and (2) refer to biharmonic and harmonic operators, and $\mathcal{F}^{(2)}_{Hx}$ is given by (28) with $A_M = 1$. $A^{(4)}_M$ is the biharmonic viscosity, which may be spatially varying, and should be negative for positive-definite dissipation of kinetic energy. Here the biharmonic viscosity is sandwiched between the two applications of the harmonic operator. An alternate approach (not implemented in the code) is to include the square-root of $-A_M$ inside the second derivative of each harmonic operator (Griffies and Hallberg, 2000).

5.2.3. Anisotropic Horizontal Viscosity. An anisotropic formulation of the friction terms in the momentum equation was first introduced into an ocean GCM by Large et al., (2001). This was implemented in a model with a spherical polar grid, and their formulation was specifically tied to that coordinate system. A more general formulation of anisotropic viscosity that can be applied in any general orthogonal coordinate system was developed by Smith and McWilliams (2002, hereafter SM), and is implemented in the POP code. The friction operator is formulated as the divergence of a viscous stress tensor which is linearly related to the velocity gradients. The general anisotropic formulation of the stress involves four independent viscous coefficients, as well as a unit vector $\hat{\mathbf{n}}$ that specifies a preferred horizontal direction which breaks the transverse isotropy. The coefficients and the direction vector may vary in both space and time. SM discuss two different reduced forms involving only two viscous coefficients, A and B. In both of these forms, the friction operator in Cartesian coordinates has the following approximate form if the x-coordinate is aligned with $\hat{\mathbf{n}}$:

$$\begin{aligned} F_x &= A\ \partial_x^2 u + B\ \partial_y^2 u \\ F_y &= B\ \partial_x^2 v + A\ \partial_y^2 v\ . \end{aligned} \tag{159}$$

Thus, the friction acts to diffuse momentum along the direction $\hat{\mathbf{n}}$ with viscosity A, and perpendicular to $\hat{\mathbf{n}}$ with viscosity B; this is true for the general operator, even when the coordinates are not aligned with $\hat{\mathbf{n}}$.

The three independent elements σ_{11}, σ_{22} and σ_{12} of the symmetric transverse stress tensor σ_{ij} are linearly related to the elements of the rate-of-strain tensor $\dot{e}_{ij}$ (where the subscripts 1 and 2 refer to the x and y coordinates, respectively). In the first form discussed by SM, which is currently implemented in the code, the stress and strain-rate are related by

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{pmatrix} = \left[\begin{pmatrix} A & -B & 0 \\ -B & A & 0 \\ 0 & 0 & B \end{pmatrix} + (A-B) n_1 n_2 \begin{pmatrix} -2n_1n_2 & 2n_1n_2 & n_1^2-n_2^2 \\ 2n_1n_2 & -2n_1n_2 & n_2^2-n_1^2 \\ n_1^2-n_2^2 & n_2^2-n_1^2 & 2n_1n_2 \end{pmatrix} \right] \begin{pmatrix} \dot{e}_{11} \\ \dot{e}_{22} \\ 2\dot{e}_{12} \end{pmatrix} \tag{160}$$

where n_1 and n_2 are the components of $\hat{\mathbf{n}}$ along the x and y coordinates, respectively. In the second two-coefficient form, the relation between the stress and strain-rate is also given by (160), except that the first matrix inside the brackets on the right is replaced by

$$\begin{pmatrix} \frac{1}{2}(A'+B') & -\frac{1}{2}(A'+B') & 0 \\ -\frac{1}{2}(A'+B') & \frac{1}{2}(A'+B') & 0 \\ 0 & 0 & B' \end{pmatrix} .$$

The second form is actually independent of the horizontal divergence, and can be written in the more compact form:

$$\begin{pmatrix} \sigma_T \\ \sigma_S \end{pmatrix} = \left[\begin{pmatrix} A+B & 0 \\ 0 & 2B \end{pmatrix} + 2(A-B)n_1 n_2 \begin{pmatrix} -2n_1 n_2 & n_1^2 - n_2^2 \\ n_1^2 - n_2^2 & 2n_1 n_2 \end{pmatrix} \right] \begin{pmatrix} \dot{e}_T \\ \dot{e}_S \end{pmatrix} . \qquad (161)$$

where $\sigma_T = \sigma_{11} - \sigma_{22}$, $\sigma_S = 2\sigma_{12}$, $\dot{e}_T = \dot{e}_{11} - \dot{e}_{22}$ and $\dot{e}_S = 2\dot{e}_{12}$. The second form is currently not in the code but will be implemented in a future version. The two forms are equivalent up to terms proportional to the horizontal velocity divergence $\nabla \cdot \mathbf{u} = \dot{e}_{11} + \dot{e}_{22}$ which is very small compared to typical velocity gradients in geophysical flows, as discussed by SM. The first form was essentially constructed so that it exactly produces the Cartesian friction operator (159), whereas the second form was derived as the limit $\nabla \cdot \mathbf{u} \to 0$ of the general 4-coefficient form of the stress. For this reason, the second form has a more sound physical basis, however, the results using the two forms should in most cases be very similar.

In both forms the stress is invariant under a rotation of $\hat{\mathbf{n}}$ by 90°. Currently there are three options in the code for the field of unit vectors $\hat{\mathbf{n}}$: (1) aligned along the flow direction ($\hat{\mathbf{n}} = \mathbf{u}/|\mathbf{u}|$); (2) aligned with due East (or, equivalently, with any of the four cardinal directions); and (3) aligned with the local grid coordinates (in polar coordinates this is equaivalent to (2), but it is different in general orthogonal coordinate systems such as the displaced-pole grids described in Sec. 7). With any of these choices, the isotropic limit is obtained by equating the two coefficients $A = B$. The first form reduces to the original anisotropic formulation of Large et al. (2001) when $\hat{\mathbf{n}}$ is oriented eastward.

The spatial discretization of the anisotropic friction operator is described in detail by SM, and will only be briefly reviewed here. The basic idea of this approach is to take advantage of the fact that the friction operator can be written as the functional derivative of the area-integrated energy dissipation rate. Using the no-slip lateral boundary conditions on the velocity, it can be shown that the domain-averaged dissipation of kinetic energy due to friction is given by

$$\int da\, \mathbf{u} \cdot \mathbf{F} = \int da\, \mathcal{D} \qquad (162)$$

where $\mathbf{F}$ is the friction vector and $\mathcal{D}$ is the energy dissipation rate:

$$\mathcal{D} = \frac{1}{2} \dot{e} : \sigma . \qquad (163)$$

To ensure positive-definite dissipation of kinetic energy, the viscous coefficients must be chosen to satisfy $\mathcal{D} \geq 0$. It can be shown that in the first form this leads to the constraint $A \geq B \geq 0$, whereas the second form has the less restrictive constraint $A \geq 0$, $B \geq 0$. The i^{th} component of the friction is given by

$$F_i = -\frac{\delta}{\delta u_i} \int dV\, \mathcal{D} \qquad (164)$$

where $\frac{\delta}{\delta u_i}$ denotes a functional derivative with respect to the i^{th} component of the velocity. The procedure for deriving the discrete operator is to first discretize the functional, making sure that the discrete form is positive definite, and then take its derivatives with respect to the velocity components at a given point on the compuational grid; this yields a friction operator at that point that is guaranteed to dissipate kinetic energy. As was done in the case of the GM operator described in Sec. 5.1.3, the method of discretizing the functional is to subdivide each horizontal cell into four subcells (see Fig. 5, with the central point interpreted as a U-point). This allows all quantities (such as viscous coefficients, gradients of velocities, etc.) required to construct the dissipation rate to be assigned a unique value in each subcell. In this manner the components of the strain-rate tensor are computed in each subcell, and from these the stresses are computed using (160). The specific details of the derivation are given by SM, and we will only quote the final result here. The discrete friction operator has the form

$$
\begin{aligned}
F_x = \; & \frac{1}{V} \, [\; \overline{h_2\sigma_{11}}^{e} - \overline{h_2\sigma_{11}}^{w} + \overline{h_1\sigma_{12}}^{n} - \overline{h_1\sigma_{12}}^{s} \\
& + \frac{1}{2} \, (h_2^n k_2^n \; \overline{h_1\sigma_{12}}^{n} + h_2^s k_2^s \; \overline{h_1\sigma_{12}}^{s}) \\
& - \frac{1}{2} \, (h_1^e k_1^e \; \overline{h_2\sigma_{22}}^{e} + h_1^w k_1^w \; \overline{h_2\sigma_{22}}^{w}) \;] \\
\\
F_y = \; & \frac{1}{V} \, [\; \overline{h_1\sigma_{22}}^{n} - \overline{h_1\sigma_{22}}^{s} + \overline{h_2\sigma_{12}}^{e} - \overline{h_2\sigma_{12}}^{w} \\
& + \frac{1}{2} \, (h_1^e k_1^e \; \overline{h_2\sigma_{12}}^{e} + h_1^w k_1^w \; \overline{h_2\sigma_{12}}^{w}) \\
& - \frac{1}{2} \, (h_2^n k_2^n \; \overline{h_1\sigma_{11}}^{n} + h_2^s k_2^s \; \overline{h_1\sigma_{11}}^{s}) \;]
\end{aligned}
\tag{165}
$$

where F_x and F_y are the components of the friction operator along the two general orthogonal coordinate directions x and y. The factors h_1 and h_2 are distances between grid points along the two coordinate directions, and k_1 and k_2 are meric factors proportional to the derivatives of h_2 and h_1 in the x and y directions, respectively. The overbars denote an average over the four subcells surrounding a given face (denoted by the superscript n, s, e, or w, for the north, south, east and west faces), and V is the full cell area. For a more detailed description of the quanities in (165), see the appendix of SM.

The viscous coefficients are assumed to have a constant value within a given full cell, i,e., all four subcells within each full cell have the same values of the coefficients A and B. There are currently three options in the code for the form of the viscous coeffients: (1) constant values of A and B in both space and time, (2) spatially varying but constant in time; and (3) Smagorinsky-type nonlinear coefficients that depend on the local deformation rate and hence vary in both space and time. In options (2) and (3), the viscosities are tapered if they exceed one-half the viscous CFL limit. Specifically, both A and B are tapered as follows:

$$
\begin{aligned}
A &= \min(A, \frac{1}{2} A_{\text{cfl}}) \\
A_{\text{cfl}} &= \frac{(dx)^2 + (dy)^2}{4 \, dt}
\end{aligned}
\tag{166}
$$

where dt is the model momentum timestep, and dx and dy are the grid spacings in the two coordinate directions. For option (1), A and B are simply set to different, constant values. The model does not enforce the limit, if they exceed A_{cfl}; only a warning message is printed. If $A < B$, the integration is terminated with an error exit.

Option (2) is used in the coarse resolution configuration of the CCSM2.0, which has a resolution of about 3°. The spatial dependence of the coefficients is computed following Large et al. (2001) with very minor modifications. The two primary design criteria are to have viscosity at values appropriate for the parameterization of missing meso-scale eddies wherever possible and to use other values only where required by the numerics. The *parallel* coefficient A_1 is given by

$$A_1 = \max\left\{\frac{1}{2}\ V(z)\ F(\phi)\ \max[dx,\ dy],\quad A_{eddy}\right\}, \tag{167}$$

where the first part represents the grid Reynolds number constraint, and A_{eddy} is a physical, lower bound set to account for all the missing mesoscale eddy activity. The product $V(z)F(\phi)$ produces a reasonable, *a priori* velocity distribution with large values near the equator, diminishing both with depth and polewards. Thus,

$$V(z) = V_o\ exp(z/D), \tag{168}$$

with $V_o = 1$ m s^{-1} and the e-folding depth $D = 1500$ m, and

$$\begin{aligned} F(\phi) &= 0.425\ cos(6\phi)\ +\ 0.575 \quad \text{for } |\phi| \le 30°, \\ F(\phi) &= 0.15 \quad \text{for } |\phi| > 30°. \end{aligned} \tag{169}$$

The *perpendicular* coefficient B_1 is constructed as

$$B_1 = \max\left\{c_3\ \beta\ dx^3\ exp(-p(x)^2),\ B_{eddy}\ (1 + c_2(1 + cos[2\phi + \pi]))\right\}. \tag{170}$$

The first part represents a numerical, Munk viscous western boundary constraint with

$$p(x) = L_M^{-1}\ \max(0, x - x_N) \tag{171}$$

where $p(x)$ causes B_1 to fall off as fast as possible away from the western boundaries. Here, x_N is the zonal coordinate of the N^{th} grid point east of the nearest western boundary and L_M is a length scale. Viscosity is not similarly increased near zonal and eastern boundaries because doing so does not reduce numerical noise. Also, $c_3(\approx 0.2)$ is a dimensionless coefficient, and β is the meridional gradient of the Coriolis parameter. In the second part, B_{eddy} represents a physical lower bound which can be increased away from the equator to control possible numerical noise, using the dimensionless coefficient c_2. In most applications, $B_{eddy} = A_{eddy}$.

To obtain the final distributions of A and B, the viscosities are tapered if they exceed one-half of the viscous CFL limit, and the positive-definite dissipation of kinetic energy constraint is enforced:

$$\begin{aligned} A &= \min(A_1, \frac{1}{2}A_{\text{cfl}}), \\ B &= \min(B_1, \frac{1}{2}A_{\text{cfl}}), \\ A &= \min(A, B). \end{aligned} \tag{172}$$

5.2.4. Smagorinsky Nonlinear Viscous Coefficients. The code is set up to optionally allow for Smagorinksy-type nonlinear dependence of the coefficients on the deformation rate (Smagorinsky, 1993) if the anisotropic functional discretization described in Sec. 5.2.3 is used. This option is not available with the simple Laplacian-like friction operator given by (29). An isotropic form with the same Smagorinsky-type nonlinear viscosity can be obtained by simply setting the two anisotropic viscous coeffieicnts A and B equal to one another.

The deformation rate $\dot{D}$ is proportional to the norm of the strain-rate tensor:

$$\frac{1}{2}\dot{D}^2 = \|\dot{e}\|^2 = \dot{e}_{11}^2 + \dot{e}_{22}^2 + 2\dot{e}_{12}^2 \approx \frac{1}{2}(\dot{e}_{_T}^2 + \dot{e}_{_S}^2) \tag{173}$$

This is particularly easy to evaluate since the strain-rate tensor is already computed in the code. Specifically, the nonlinear coefficients A and B are given by

$$\begin{aligned} A &= \min[\max\left(C_{_A}\dot{D}\,ds^2,\ V_{_A}\,ds\right),\ \frac{1}{2}A_{\text{cfl}}] \\ B &= \min[\max\left(C_{_B}\dot{D}\,ds^2,\ V_{_B}\,ds\right),\ \frac{1}{2}A_{\text{cfl}}] \end{aligned} \tag{174}$$

where $ds = \min(dx, dy)$ is the minimum grid spacing in the two coordinate directions. $C_{_A}$ and $C_{_B}$ are dimensionless coefficients of order 1, and $V_{_A}$ and $V_{_B}$ are velocity scales associated with the grid Reynolds number which determine a minimum background viscosity in regions where the nonlinear viscosity is too small to control grid-point noise. Typically $V_{_A}$ and $V_{_B}$ are order 1 cm s^{-1}. A_{cfl} is the maximum viscosity allowed by the viscous CFL limit, see Eq. 166. The advantage of the nonlinear viscosities (174) is that they selectively apply the friction operator in regions of strong shear.

In the high resolution configuration of the CCSM2.0, which has a resolution of about 1°, the Smagorinsky coefficients have been modified in a similar manner to the anisotropic coefficients in Large et al. (2001) described above. Again the reason is to control numerical noise in the solutions. Eq. (174) has been modified such that

$$\begin{aligned} A &= \min\Big[\max\Big(C_A\ F_A(\phi)\ \dot{D}\ ds^2,\ A_1\Big),\ \tfrac{1}{2}A_{\text{cfl}}\Big], \\ B &= \min\Big[\max\Big(C_B\ F_B(\phi)\ \dot{D}\ ds^2,\ B_1\Big),\ \tfrac{1}{2}A_{\text{cfl}}\Big], \end{aligned} \tag{175}$$

with

$$\begin{aligned} F_A(\phi) &= 1, && \\ F_B(\phi) &= 0.02 && \text{for } |\phi| \le 20°, \\ &= 1\ -\ 0.98\ exp(-(\phi-20)^2/98) && \text{for } |\phi| > 20°. \end{aligned} \tag{176}$$

Here, $ds = \min(dx, dy)$ is the minimum grid spacing in the two coordinate directions. C_A and C_B are dimensionless coefficients of order 1 whose latitudinal dependencies are given by F_A and F_B, respectively. The positive-definite dissipation of kinetic energy is enforced by the last equation in (172). Where these nonlinear viscosities are too small, A_1 and B_1, with the same formulations as in the previous section, provide lower limits to control any grid-point noise.

5.3. Parameterizations of Vertical Mixing

Vertical mixing is treated using the spatial discretizations presented in sections 3.3.3 and 3.4.4. The implicit time discretizations are presented in Secs. 4.2.2, 4.2.3, and 4.4.2. POP contains three different parameterizations for computing the vertical diffusivity κ (VDC in the code) and viscosity μ (VVC in the code): constant coefficients, a parameterization based on the gradient Richardson number by Pacanowski and Philander (1981) and the K-profile parameterization (KPP) of Large et al. (1994).

5.3.1. Convection. Convective instability can be handled in two ways within the code. If implicit vertical mixing is used, convection is generally treated by assigning a very large diffusivity and viscosity when the density profile between two cells in the vertical direction is statically unstable. The diffusivity and viscosity in each case is determined at run time through namelist inputs (convect_diff, convect_visc).

If explicit vertical mixing is used, convection is usually treated using convective adjustment. During convective adjustment, multiple passes through the water column are performed to check for stability and to adjust tracer values. Each pass in turn consists of two sweeps. The first sweep checks odd-valued vertical levels; the second checks even-valued levels. At each level k, the stability between level k and level $k+1$ is checked (see below for stability criterion). If the two cells are found to be unstably-stratified, the tracer in each level is adjusted using

$$\phi_k = \phi_{k+1} = \frac{(dz_k\phi_k + dz_{k+1}\phi_{k+1})}{2dz_{k+\frac{1}{2}}}. \tag{177}$$

If tracer acceleration is being used, the thicknesses dz are adjusted by the acceleration factor

$$dz_k^* = dz_k/\gamma_k \tag{178}$$

$$dz_{k+\frac{1}{2}}* = \frac{1}{2}(dz_k^* + dz_{k+1}^*). \tag{179}$$

Instability can be determined using one of two methods. In both the Richardson number parameterization and KPP, the Richardson number is computed and the column is unstable when the Richardson number is less than zero. The calculation of Richardson number differs slightly between the two and is shown in each section below.

If the Richardson number is not computed, stability is determined by adiabatically displacing a water mass from the current vertical level k to the level $k+1$ just below. The density of the parcel after this displacement, denoted ρ_k^*, is computed by calling the equation of state with the temperature and salinity from level k, but evaluating the equation of state at level $k+1$. If the density after the displacement is greater than the actual density at the level $k+1$, the column is unstable.

5.3.2. Constant Vertical Viscosity. If this option is chosen, the vertical diffusivity VDC and vertical viscosity VVC are simply constant at all levels and for all tracers. The scalar constant values const_vdc and const_vvc are determined at run time through namelist input.

5.3.3. Richardson Number Dependent Mixing. In this parameterization, the vertical diffusivity and viscosity are functions of the Richardson number,

$$
\begin{aligned}
Ri &= -g\frac{(\rho_k^* - \rho_{k+1})}{\frac{1}{2}(dz(k) + dz(k+1))}\left(\frac{\partial V}{\partial z}\right)^{-2} \\
&= \frac{-g dz_{k+\frac{1}{2}}(\rho_k^* - \rho_{k+1})}{((u_k - u_{k+1})^2 + (v_k - v_{k+1})^2 + \epsilon)} \qquad (180)
\end{aligned}
$$

where the velocities are evaluated at tracer points and ϵ is simply a small number to avoid dividing by zero.

The particular functional forms for the diffusivity κ (VDC) and viscosity μ (VVC) used in the code are

$$\kappa = \kappa_{bkgrnd} + (\mu_{bckgrnd} + Rich\_mix/(1+5Ri)^2)/(1+5Ri) \qquad (181)$$

$$\mu = \mu_{bckgrnd} + Rich\_mix/(1+5Ri)^2 \qquad (182)$$

where the background values κ_{bkgrnd} and $\mu_{bckgrnd}$ and Richardson mixing coefficient $Rich\_mix$ are determined at run time through namelist input. Note that the Richardson number used for the viscosity μ has been averaged from tracer points to velocity points.

5.3.4. The KPP Boundary-Layer Parameterization. The full KPP parameterization is detailed in Large et al. (1994), hereafter LMD, so only a general overview plus specifics of the current POP implementation are described here. The scheme provides all the coefficients required to compute the vertical mixing contributions to both $\mathcal{F}_V(\alpha)$ in Eq. (7), and $\mathcal{D}_V(\varphi)$ in Eq. (11). As such, it should not be used in conjunction with any other parameterization of vertical mixing. Implicit vertical mixing is required to solve Eq. (7) and a modified version of Eq. (13); namely,

$$\mathcal{D}_V(\varphi) = \frac{\partial}{\partial z}\,\kappa\,\Big(\frac{\partial}{\partial z}\varphi \; - \; \gamma_\varphi\Big), \qquad (183)$$

where the additional non-local transport term, γ_φ is non-zero only in convective (unstable) forcing, and above a diagnosed boundary layer depth, h. In the ocean interior below h, the viscosity, μ_I, and diffusivity, κ_I, represent very different physics than their boundary layer, $z < -h$, counterparts, denoted μ_B and κ_B, respectively.

The inputs to the scheme on the tracer grid are the surface forcing and vertical profiles of temperature and salinity. The specific forcings are the surface friction velocity, u^*, both the solar, B_{sol}, and non-solar, B_{ns}, buoyancy fluxes, and the kinematic surface tracer fluxes, $\overline{w\varphi}_o$. From the equation of state and its buoyancy form, $b(\Theta, S, p)$, derived profiles are the thermal and haline Boussinesq coefficients, α_T and β_S, the local buoyancy difference at each interface, $\Delta b_{k+.5}$, and the excess buoyancy of first level water when moved down to each grid level,

$$\Delta B_k = b(\Theta_1, S_1, -z_k) \; - \; b(\Theta_k, S_k, -z_k)\;. \qquad (184)$$

The local vertical shear at each interface is squared before being averaged onto the tracer grid, as $Sh^2_{k+.5}$. Similarly, velocity differences with the first level are squared, then averaged to the tracer grid to give

$$|\Delta V|^2_k \;=\; |(u_1, v_1) \; - \; (u_k, v_k)|^2\;. \qquad (185)$$

For vertical resolutions of $1m$ or less, excessive sensitivity can be avoided if Θ_1, S_1, u_1 and v_1 in the above equations are replaced by averages over the upper 10% of the boundary layer (LMD). A local gradient Richardson number, Ri_g, buoyancy frequency, N, and density ratio, R_ρ, on the tracer grid are computed as

$$\begin{aligned} N^2 &= \Delta B_{k+.5} \;/\; (z_k - z_{k+1}) \\ Ri_g &= N^2 \;/\; Sh^2_{k+.5} \\ R_\rho &= \alpha_T\ \partial_z \Theta \;/\; (\beta_S\ \partial_z S\ ). \end{aligned} \tag{186}$$

It is then a four step process to complete the algorithm, as described below. First, the interior mixing coefficients are computed at all model interfaces, on the tracer grid, as if there were no boundary layer scheme. These are the sum of individual viscosities and diffusivities corresponding to a number (currently up to four) of different physical processes:

$$\begin{aligned} \mu_I &= \mu_w + \mu_s + \mu_c \\ \kappa_I &= \kappa_w + \kappa_s + \kappa_c + \kappa_d \end{aligned} \tag{187}$$

The first coefficients on the right hand side are background values associated with internal waves. Their vertical variation has the general form

$$\begin{aligned} \kappa_w &= vdc1\ +\ vdc2\ \ tan^{-1}((|z| - dpth)\ linv), \\ \mu_w &= Pr_w\ \ \kappa_w \end{aligned} \tag{188}$$

where $vdc1$ equals the vertical diffusivity at $|z| = D$, $vdc2$ is the amplitude of variation , $linv$ is the inverse length scale of the transition region, and $dpth$ is the depth where diffusivity equals $vdc1$. These constants are namelist specified as `bckgrnd_vdc1` (cm^2/s), `bckgrnd_vdc2` (cm^2/s), `bckgrnd_vdc_linv` (cm^{-1}), and `bckgrnd_vdc_dpth` (cm), respectively. The form allows for an increase in diffusivity with depth, as a crude parameterization of the observed increase in deep mixing over rough topography. The viscosity has the same form, but multiplied by a constant Prandtl number, Pr_w (`Prandtl` in namelist).

The viscosity and diffusivity associated with shear instability mixing are equal ($Pr_s = 1$) and parameterized as a function of Ri_g:

$$\mu_s = \kappa_s = \kappa_s^0\ \ [1 - (\ \frac{Ri_g}{Ri_0}\ )^2\ ]^3, \qquad\qquad 0 < Ri_g < Ri_0. \tag{189}$$

This computation is enabled with a namelist flag (`lrich` = .true.). For an unstable profile with negative Ri_g the coefficients remain constant at κ_s^0 (`rich_mix` (cm^2/s) in namelist), and are zero for all $Ri_g \geq Ri_0$. This function falls most rapidly near $Ri_g = 0.4Ri_0$, where it approximates the onset of shear instability. In this neighborhood rapid changes in Ri_g can cause instabilities to develop in the vertical, but these are largely controlled by vertical smoothing Ri_g profiles. A $1-2-1$ smoother is repeated a specified number `num_v_smooth_Ri` of times.

Convective instability in the interior ocean is relieved by setting the mixing coefficients, μ_c and κ_c, to large values whenever the density profile is unstable, $N < 0$. Otherwise, they are set to zero.

Double diffusion processes have the potential to enhance diffusivities, but not viscosity, with R_ρ the governing parameter. They are enabled or disabled with a namelist flag (`ldbl_diff` = .true. or false.). In the salt fingering regime (destabilizing salinity profile and $1 < R_\rho < R_\rho^0 < 2$), the thermal diffusivity is less than that of salt (LMD). For salt

$$\kappa_d \;=\; \kappa_d^0 \, [1 - (\frac{R_\rho - 1}{R_\rho^0} - \; 1 \;)^2 \;]^3. \tag{190}$$

The magnitude is internally set by κ_d^0, which should be less than $1cm^2/s$. Diffusive convective instability occurs where the temperature is destabilizing and $0 < R_\rho < 1$. For temperature

$$\kappa_d = VISCM \quad .909 \;\; exp(4.6 \; exp[-.54(R_\rho^{-1} - 1)]), \tag{191}$$

where $VISCM$ is molecular viscosity. Multiplying this diffusivity by a factor

$$\begin{array}{ll} (1.85 \; - \; 0.85R_\rho^{-1}) \; R_\rho & \qquad 0.5 \le R_\rho < 1 \\ 0.15 \;\; R_\rho & \qquad R_\rho < 0.5, \end{array} \tag{192}$$

gives diffusivities for other tracers, including salt.

Second, the diagnostic boundary layer depth, h, is determined on the tracer grid. A bulk Richardson number relative to the surface is defined at each vertical level as

$$Ri_b \;=\; \frac{-z_k \;\; \Delta B_k}{|\Delta V|_k^2 + V_t^2} \tag{193}$$

LMD gives the rationale and expression for the shear contribution from unresolved turbulence, V_t/d. The boundary layer depth is equated to the shallowest depth, where Ri_b equals an internally specified critical value, $Ri_{cr} = 0.3$ (LMD). With stabilizing surface forcing, options exist (namelist flag `lcheckekmo` = .true.) to limit h to be no greater than either the Ekman depth ($0.7u^*/f$), or the Monin-Obukhov depth ($L_{mo} = u^{*3}/(vonk \;\; B_f)$), where $vonk$ is the von Karman constant, and the surface buoyancy flux, B_f, is B_{ns} plus a fraction of B_{sol} (LMD). The grid level immediately below h is denoted as k_b.

Third, the boundary layer mixing coefficients are computed on the tracer grid and replace the interior values from step 1, for $1.5 \le k \le k_b - \frac{1}{2}$. The analytic expression is

$$\begin{array}{ll} \mu_B(\sigma) & = \; vonk \; w_\alpha \; h \; G(\sigma) \\ \kappa_B(\sigma) & = \; vonk \; w_\varphi \; h \; G(\sigma) \;\; , \end{array} \tag{194}$$

where $\sigma = -z/h$ varies from 0 to 1 over the boundary layer. The turbulent velocity scales, w_α and w_φ are usually proportional to u^*, but become proportional to the convective velocity scale as $u^* \to 0$ in convective forcing (see LMD for details). The shape function is a cubic polynomial whose coefficients are chosen such that $G(0) = 0$, fluxes vary linearly near the surface, and interior and boundary layer coefficients, and their first vertical derivatives, are continuous at $z = -h$. An inherent bias to shallow boundary layers is ameliorated by making the coefficients at the $k_b - \frac{1}{2}$ interface linear combinations of the interior values at this interface, and the boundary layer values at both the interface and at the nearest higher level, $k_b - 1$ (LMD).

In convective (unstable) forcing situations the nonlocal term is non-zero, and evaluated as

$$\gamma_\varphi \;=\; C_\varphi \;\frac{\overline{w\varphi_o}}{u^*\;h}, \tag{195}$$

where the constant C_φ is prescribed as in LMD.

Fourth and finally, the viscosity is averaged from the tracer grid to velocity points.

6. Other Numerical Features

6.1. Advection Schemes

Third-Order Upwind Algorithm

Third-order upwind schemes and variants of this scheme were introduced by Leonard (1979). Our implementation of third-order upwinding simply involves the substitution of three-point interpolation for the two-point centered average in the determination of the tracer value at cell faces, and remains in leapfrog temporal implementation.

It should be noted that improved temporal implementations have been explored in Farrow and Stevens (1995), where they apply a predictor-corrector sequence, and in Holland et al. (1998), where in addition to documenting our implementation they also describe a reorganization of terms into two groups, one of which is entirely centered in space, the other having a dissipative form.

The finite-difference expression for the advection of tracers is

$$\begin{aligned} ADV(i,j,k) = \;& -\tfrac{1}{\mathrm{DXT}}(u_E\;T^*_E - u_W\;T^*_W) - \tfrac{1}{\mathrm{DYT}}(v_N\;T^*_N - v_S\;T^*_S) \\ & -\tfrac{1}{dz}(w_k\;T^*_T - w_{k+1}\;T^*_B) \end{aligned} \tag{196}$$

where u_W, u_E, v_S, v_N are the T-grid cell face centered velocity components and the directions W, E, S and N are with respect to the logical coordinates.

$$\begin{aligned} u_E(i) &= \frac{(u_{i,j}\mathrm{DYU}_{i,j} + u_{i,j-1}\mathrm{DYU}_{i,j-1})}{2\mathrm{DXT}_{i,j}} \\ u_W(i) &= u_E(i-1) \\ v_N(j) &= \frac{(v_{i,j}\mathrm{DXU}_{i,j} + v_{i-1,j}\mathrm{DXU}_{i-1,j})}{2\mathrm{DXT}_{i,j}} \\ v_S(j) &= \frac{(v_{i,j-1}\mathrm{DXU}_{i,j} + v_{i-1,j-1}\mathrm{DXU}_{i-1,j})}{2\mathrm{DXT}_{i,j}} \end{aligned} \tag{197}$$

and the T^* quantities are the cell face centered tracer concentration at the east, west, north, south, top, and bottom faces. In the standard second order scheme of MOM, these are taken as the arithmetic average of the tracer concentrations in adjacent cells, e.g.,

$$T^*_E = \frac{1}{2}(T_{i+1,j} + T_{i,j}) \tag{198}$$

The third point included in the three-point interpolation is taken to be in the upwind direction,

$$T^*_E(i) = \begin{cases} \gamma T_{i-1} \;+ \beta^+ T_i + \alpha^+ T_{i+1} & u_E > 0 \\ \beta^- T_i + \alpha^- T_{i+1} + \delta T_{i+2} & u_E < 0 \end{cases} \tag{199}$$

$$T_W^*(i) = T_E^*(i-1) \tag{200}$$

where the coefficients $\alpha, \beta, \gamma, \delta$ are given by

$$\begin{aligned}
\alpha^+ &= \frac{dx_i\ (2dx_i + dx_{i-1})}{(dx_i + dx_{i+1})\ (dx_{i+1} + dx_{i-1} + 2dx_i)} \\
\alpha^- &= \frac{dx_i(2dx_{i+1} + dx_{i+2})}{(dx_i + dx_{i+1})\ (dx_{i+1} + dx_{i+2})} \\
\beta^+ &= \frac{dx_{i+1}\ (2dx_i + dx_{i-1})}{(dx_i + dx_{i+1})\ (dx_i + dx_{i-1})} \\
\beta^- &= \frac{dx_{i+1}(2dx_{i+1} + dx_{i+2})}{(dx_i + dx_{i+1})\ (dx_i + dx_{i+2} + 2dx_{i+1})} \\
\gamma &= \frac{-dx_i * dx_{i+1}}{(dx_i + dx_{i-1})\ (dx_{i+1} + dx_{i-1} + 2dx_i)} \\
\delta &= \frac{-dx_i * dx_{i+1}}{(dx_{i+1} + dx_{i+2})\ (dx_i + dx_{i+2} + 2dx_{i+1})}
\end{aligned} \tag{201}$$

where $dx =$ DXT and the j-index is suppressed for clarity. The interface tracer values T_N^*, T_S^*, T_T^*, and T_B^* are obtained in an analogous manner.

6.2. Penetration of Solar Radiation

Setting the namelist flag *lshortwave* = *.true.* allows solar radiation to penetrate the water column. The solar absorption has the form of a double exponential, with the fraction of surface short-wave solar radiation, $SWFRAC$, that reaches a depth, $-z$, given by :

$$SWFRAC\ =\ coef\quad exp(\frac{z}{depth1})\ +\ (1 - coef)\ \ exp(\frac{z}{depth2}). \tag{202}$$

The first term on the right-hand side represents the rapid absorption of the longer wavelengths (reds), so the extinction coefficient $depth1$ is small (0.35 $\rightarrow$ 1.4m) and a fraction ($0.58 \leq coef \leq 0.78$) of the radiation is in this band. The extinction coefficient $depth2$ for the shorter wavelengths (blues) is much greater ($7.9m \rightarrow$ 23m). These parameters are set according to a specific Jerlov water type, and are varied neither spatially, nor in time. An important restriction is that no solar radiation passes through the bottom of the model ocean, lest it be lost to the system. Also, in order to avoid numerical underflows associated with vanishingly small exponentials, $SWFRAC = 0$ for depths deeper than $200m$. Jerlov water type Ib is used in the CCSM2.0.

6.3. Variable-Thickness Surface Layer

The tracer transport equations have the form

$$\frac{\partial}{\partial t}\varphi + \nabla \cdot \mathbf{u}\varphi + \frac{\partial}{\partial z}w\varphi = \nabla \cdot \mathbf{F} + \frac{\partial}{\partial z}F_V \tag{203}$$

where $\nabla\cdot\mathbf{F} = \mathcal{D}_H$ and $\frac{\partial}{\partial z}F_V = \mathcal{D}_V$ are, respectively, the horizontal and vertical diffusion operators. Integrating this equation over the model surface level (from $-h_1$ to η, where $h_1 = dz_1$ is the depth of the upper level) we find:

$$\frac{\partial}{\partial t}\int_{-h_1}^{\eta} dz\,\varphi + \nabla\cdot\int_{-h_1}^{0} dz\,\mathbf{u}\varphi - w(h_1)\varphi(h_1) = q_W\varphi(\eta) + \nabla\cdot\int_{-h_1}^{0} dz\,\mathbf{F} + F_V(\eta) - F_V(h_1) \quad (204)$$

In deriving (204) the same approximations used in the linear free surface model (Sec. 4.5.1) were employed: the boundary condition (118) was applied, and the advective and diffusive horizontal fluxes between $z = 0$ and $z = \eta$ were set to zero. Note that there is no advection of tracers through the surface with the vertical velocity $w(\eta)$.

Some care must be taken to specify the tracer fluxes through the air-sea interface. The total tracer flux seen by the ocean model at the surface is given by

$$F_T = q_W\varphi(\eta) + F_V(\eta)\ , \quad (205)$$

where $\varphi(\eta)$ is the tracer concentration at the sea surface and $q_W\varphi(\eta)$ represents the advection of tracers in the ocean relative to the sea surface due to the freshwater flux. The atmosphere (or sea ice model) sees a flux of tracers into the ocean given by

$$Q_T = q_W\varphi_W + Q_\varphi\ , \quad (206)$$

where φ_W is the tracer concentration in the freshwater being added or removed from the ocean, (for simplicity we assume here that locally there will be only one source of freshwater, and omit the sum over different types freshwater sources - see Eq.(42)), and Q_φ is any additional flux of tracers (e.g., for temperature in the open ocean it corresponds to the sum of sensible, latent, shortwave and longwave heat fluxes). In reality there is a very thin boundary layer where the tracer concentration varies continuously and $\varphi(\eta) = \varphi_W$ at the interface. The models do not resolve this boundary layer and so in general $\varphi(\eta) \neq \varphi_W$. However, the total flux must be conserved across the boundary layer, so that $F_T = Q_T$, and this can be used to eliminate $\varphi(\eta)$ in terms of φ_W. Substituting (206) for (205) in (204), we obtain

$$\frac{\partial}{\partial t}(h_1+\eta)\varphi + h_1\nabla\cdot\mathbf{u}_1\varphi_1 - w_1\varphi(h_1) = h_1\nabla\cdot\mathbf{F}_1 + Q_\varphi - F_V(h_1) + q_W\varphi_W \quad (207)$$

where now the flux Q_φ is applied as the surface boundary condition on $\mathcal{D}_V$. The integrals over the surface layer have been evaluated by assuming the model variables are constant within the surface layer. w_1 is the velocity at the bottom of the surface layer and $w_1\varphi(h_1)$ represents tracer advection through the bottom of the surface layer. Dividing (207) by h_1, we arrive at the tracer transport equation (40).

It remains to specify the freshwater tracer concentrations φ_W. In the case of salinity the default choice in the code is $\varphi_W = S_W = 0$, that is, the freshwater flux $q_W = P - E + R - F_{ice} + M_{ice}$ is assumed to have zero salinity (where P, E, R, F_{ice} and M_{ice} are the freshwater fluxes associated with precipitation, evaporation, river runoff, freezing and melting of sea ice). In the case of potential temperature, the default is to assume the freshwater has the same temperature as the model surface layer, so that $\varphi_W = \Theta_W = \Theta_1$. However, in general the tracer concentration in the freshwater may vary depending on its source (precipitation, evaporation, etc.), and the default assumption that these are all given by a single value φ_W as in (206) may be inadequate. When coupling to atmospheric or sea-ice models a more accurate accounting of the fluxes may be needed to ensure they are conserved between component models.

Combining (207) with the transport equations in the subsurface levels and integrating over volume, we find

$$\frac{d}{dt}\int dV\ \varphi = \int da\ (Q_\varphi + q_W \varphi_W)\ . \tag{208}$$

Thus tracers are conserved in the absence of surface fluxes. This is also true of the time-discretized equations in the model. In particular, if there are no surface fluxes of salinity, $Q_S = 0$, and if the freshwater flux has zero salinity ($\varphi_W = S_W = 0$), then total salt is conserved.

The time discretization of (207) or (40) is given by (75) for leapfrog steps and by (80) and (81) for forward and backward (Matsuno) timesteps. All of these discrete equations conserve global mean tracers exactly in the sense of (208).

There is also a small error in the time discretization associated with the fact that we have adopted a three-time-level leapfrog scheme for the tracer transport equations (75), but a two-time-level discretization for the barotropic continuity equation (122). Note that in the limit of a constant tracer $\varphi = 1$ the continuous transport equation (203), in the absence of surface forcing, reduces to the continuity equation $\nabla \cdot \mathbf{u} + \frac{\partial w}{\partial z} = 0$, and the vertically integrated transport equations also reduce to the barotropic continuity equation (120). This does not hold for the discrete equations because of the inconsistency between the two- and three-time-level schemes. The consequence of this is that a tracer which is initially constant in space may evolve to have some small spatial variations. However, this error is second-order in time and is typically extremely small. In a test problem involving the Goldsborough-Stommel circulation (see Griffies et al, 2001), which involves only freshwater forcing at the surface, the temperature field, which is initially constant, deviates from its inital value by only 1 part in 10^7 in a 100-year integration, and furthermore, the error does not accumulate in time. Therefore, we feel that this error is small enough that it will not cause problems in long-term climate simulations.

Finally, we note that there is a time discretization error when tracer acceleration is used (see Sec. 4.2.1) if the momentum timestep and the surface tracer timestep are not equal. This inconsistency can quickly lead to a computational instability, and for this reason we strongly advise that the surface tracer and momentum timesteps be equal (the momentum and tracer timestep may differ in subsurface levels).

6.3.1. Nullspace removal. The operator for the implicit solution of the barotropic equations for the surface height η in (126) contains the Laplacian-like term (127). On the B-grid this 9-point Laplacian-like operator contains two null eigenvectors: a constant field and a global checkerboard ($+/-$ or black/white) field (a checkerboard field has the value of $+1$ in black cells and -1 in white cells). The Laplacian operator on the B-grid annihilates both these types of null fields (i.e., they have zero eigenvalue). On the C-grid, the global checkerboard is not a null eigenvector; for this reason all the horizontal diffusive operators in POP have an approximate C-grid discretization, allowing grid-scale checkerboard noise to be damped by these operators. However, the presence of a checkerboard component in the surface height field does not significantly affect the dynamical solution in most cases. The dominant effect of the surface height is through its contribution to the surface pressure gradient, and on the B-grid the gradient operator also does not see the checkerboard field. There is only a small indirect effect on the vertical velocities in T-columns since they are determined by vertically-integrating the continuity equation with the surface boundary condition (45), which depends on η.

In the implicit free surface formulation, the extra term on the left in (126) is a diagonal term associated with the free surface (the $\partial_t \eta$ term in the barotropic continuity equation). Without this diagonal term (as is the case in the rigid-lid surface pressure formulation (Smith et al., 1992)), the null eigenvectors are not removed by the operator, and can creep into the iterative solution. The diagonal term shifts the eigenvalue of both constant and checkerboard fields away from zero so that they are damped. Therefore, in most cases it is not necessary to explicitly remove constant or checkerboard fields from the solution of the implicit free-surface system. However, experience has shown that the solution sometimes does develop large but localized regions where checkerboard noise appears in the surface height. The exact nature of such regional checkerboarding is not completely understood. It can even occur in the rigid-lid streamfunction formulation using an approximate 5-point operator for the barotropic operator which has no global checkerboard nullspace (a 5-point approximation is allowed in this case but not in the surface pressure or free-surface formulations, because the 5-point approximation to the 9-point B-grid operator violates energetic consistency (see Smith et al., 1992)). Experience has also shown that this regional checkerboarding does not affect the dynamical solution, again because it is almost completely invisible to the gradient operator. However, with the variable-thickness surface layer (Sec. 6.3), such regional checkerboarding can appear in the solution and then slowly accumulate over time, eventually leading to large checkerboard components that can affect the solution, either by contributing a large roundoff error to the pressure gradient, or by becoming so large that in some cells the surface height descends below the bottom of the surface layer.

When complicated lateral boundaries are included in the computational mesh, other global null eigenvectors besides the constant and checkerboard fields can also exist. These occur when the domain contains quasi-isolated bays that are only connected to the rest of the ocean through openings of one or two cell faces. These regions can have their own constant and checkerboard nullspaces. There are two types of quasi-isolated bays that need to be considered: 1) bays that are connected to the rest of the ocean at the surface by only one face of a T-cell, and 2) bays that are connected by two adjacent faces of the same T-cell, so that the mouth of the bay is across the diagonal of the T-cell. If either of these types of bay is in the grid, a checkerboard nullspace component of the solution can accumulate in the bay. In the first type of bay, a constant nullspace can also accumulate. In these smaller areas, the problems mentioned above (leading to roundoff error or complete emptying of surface cells) accumulate more quickly in time. Normally, bays of the first type do not appear, because it is the usual practice during construction of a grid to eliminate all points that are connected to the open ocean by only one cell face (such points have no influence on the flow except through diffusion across the open face). But when the variable-thickness surface layer is used, it is necessary to remove all isolated bays of both types in order for the model to run stably. Once these bays are eliminated from the grid, it is still necessary to remove the global constant and checkerboard nullspaces associated with the full domain, to prevent them from slowly accumulating over time. The null fields are projected out of the solution subject to two constraints: 1) the resulting field has no checkerboard component; and 2) the projection leaves the global mean sea-level unchanged. Denoting the null eigenvectors $\mathbf{P}_{const}$ and $\mathbf{P}_{check}$, which are normalised to 1 and ± 1, respectively, the nullspaces can be projected out with the following operation (here $\mathbf{X}$ is the surface height field considered as a one-dimensional vector, and $\mathbf{X}'$ is the same field after the projection):

$$\begin{aligned}
\mathbf{X}' &= \mathbf{X} + a\,\mathbf{P}_{const} - b\,\mathbf{P}_{check} \\
a &= \frac{X_{check} V_{check}}{N\,V_{const} - M\,V_{check}} \\
b &= \frac{X_{check} V_{const}}{N\,V_{const} - M\,V_{check}} \\
X_{check} &= \sum X \cdot \mathbf{P}_{check}\ , \\
N &= \sum \mathbf{P}_{const} \cdot \mathbf{P}_{const}\ , \\
M &= \sum \mathbf{P}_{check} \cdot \mathbf{P}_{check}\ , \\
V_{check} &= \sum d\mathbf{a} \cdot \mathbf{P}_{check}\ , \\
V_{const} &= \sum d\mathbf{a} \cdot \mathbf{P}_{const}\ , \qquad (209)
\end{aligned}$$

where $d\mathbf{a}$ is the one-dimensional vector of cell areas, and the sums are over all surface cells. It is straigtforward to show that $\sum \mathbf{X} \cdot \mathbf{P}_{check} = 0$ and $\sum \mathbf{X}' \cdot d\mathbf{a} = \sum \mathbf{X} \cdot d\mathbf{a}$. The latter constraint ensures the projection does not change the global mean sea-level. This projection is applied once each timestep immediately after the barotropic equations are solved for η.

6.4. Partial Bottom Cells

Note: Although PBC's are not in the initial CCSM2.0 release of the ocean model, we expect they will be in the next release and hence will include them in the manual.

6.4.1. Pressure error and spurious diffusion. To better represent the bottom topography, we employ the idea of partial bottom cells introduced by Adcroft et al., (1996) and later developed for the MOM model by Pacanowski and Gnanadesikan (1998), for which the thickness of bottom cells is allowed to vary in space. When we place tracer or pressure points at the center of partial bottom cells, as shown in Figure 6, the finite difference for horizontal derivatives between two adjacent points introduces an error since these points are located at different vertical positions. More specifically, when we compute the horizontal pressure gradient, we have

$$\left.\frac{\partial p}{\partial x}\right|_{z=-h(x)} = \frac{\partial}{\partial x}\left[p\,|_{z=-h(x)}\right] - g\rho\,|_{z=-h(x)}\,h'(x)\,, \qquad (210)$$

where $h(x)$ can be regarded as the position of tracer points. Without the second term on the right-hand side of (210), the pressure gradient does not vanish even for $\rho = \rho(z)$ for which we expect $p_x|_{z=-h(x)} = 0$. Even though we include the correction term, p_x vanishes only when $\rho(z)$

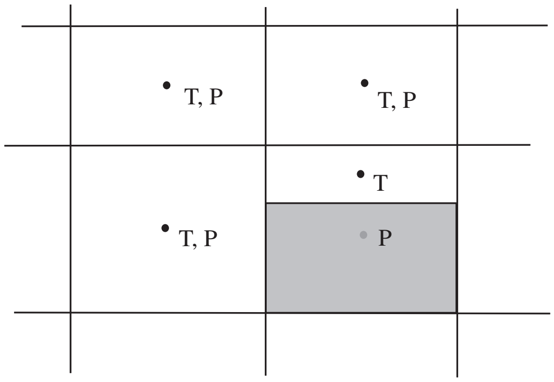

Figure 6: Pressure and tracer points in the x-z plane above partially-filled bottom cells.

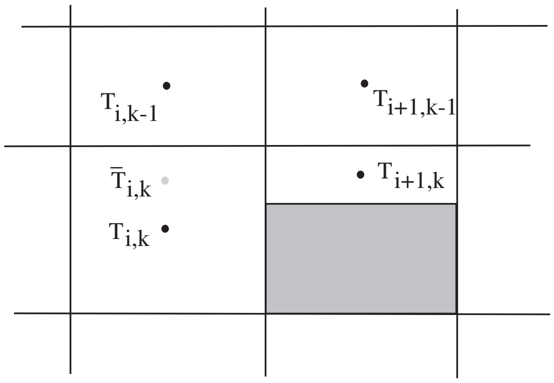

Figure 7: Interpolating tracer points to the same level before taking horizontal derivatives.

varies linearly in depth, since p is computed from the hydrostatic equation $p_z = -\rho g$ by using the trapezoidal rule. Likewise, when we compute the horizontal diffusion of tracers without taking the effects of different vertical positions of tracer points into account, we also introduce spurious diffusion errors. To avoid these errors, the interpolation between two vertical points has been used in z-coordinate ocean models. More specifically, after finding $\overline{T}_{i,k}$ by linear interpolation between $T_{i,k}$ and $T_{i,k-1}$,

$$\overline{T}_{i,k} = T_{i,k} + \frac{\min(h_{zi,k}, h_{zi+1,k}) - h_{zi,k}}{h_{zi,k} - h_{zi,k-1}} \left(T_{i,k} - T_{i,k-1}\right), \tag{211}$$

where h_z is the height of T-cells, the horizontal gradient is computed as $\Delta_x \overline{T} = T_{i+1,k} - \overline{T}_{i,k}$, as shown in Figure 7. This approach seems to be correct but it does not guarantee the negative definiteness of tracer variance: for example, for the Laplacian horizontal mixing,

$$\sum_{i,k} \left(\Delta_x^2 \overline{T}_{i,k}\right) T_{i,k}\,, \tag{212}$$

is not sign-definite! A correct approach is to work with the functional which can be written as

$$\mathcal{G}[T] = -\frac{1}{2} \sum \frac{h_y h_z}{h_x} \left(\Delta_x \overline{T}\right)^2, \tag{213}$$

where h_x and h_y are the horizontal grid sizes of T-cells in the x and y directions, respectively. Taking the functional derivative leads to the following discretization for the horizontal Laplacian diffusion:

$$\begin{aligned} dV_T\, D[T] = {} & \Delta_x \left[\left(\frac{h_y h_z}{h_x}\right) \overline{\kappa}^x \, \Delta_x \overline{T} \right] + \Delta_x \left[\left(\frac{h_y}{h_x}\right) \Delta_z \left(\gamma h_z \overline{\kappa}^x \Delta_x \overline{T}\right) \right] \\ & + \Delta_y \left[\left(\frac{h_x h_z}{h_y}\right) \overline{\kappa}^y \, \Delta_y \overline{T} \right] + \Delta_y \left[\left(\frac{h_x}{h_y}\right) \Delta_z \left(\gamma h_z \overline{\kappa}^y \Delta_y \overline{T}\right) \right]. \end{aligned} \tag{214}$$

Since the correct discretization even for the simple Laplacian mixing is too complicated and thus expensive to employ, we have made the partial-bottom-cell implementation in POP in the following ways: (1) we leave the pressure points at the same z-levels which introduces no pressure error and guarantees energetic consistency. Even though some pressure points are underneath the ocean floor, the pressure is computed by extrapolating tracers, therefore density, to pressure points. (2) when we compute the tracer gradients in diffusion operators, we make no interpolation for tracers. This approach introduces spurious diffusion errors but guarantees at least the sign-definite tracer variances, which has been given up in other OGCMs. Our test results show the error introduced by the second approach is so small that we can avoid any complication involving the interpolation.

6.4.2. Discretization. After reading the height of bottom T-cells, POP uses two additional three-dimensional arrays for the thickness of T-cells, $\mathrm{DZT}_{i,j,k}$, and the thickness of U-cells defined as

$$\mathrm{DZU_{i,j,k}} = \min(\mathrm{DZT}_{i,j,k}, \mathrm{DZT}_{i+1,j,k}, \mathrm{DZT}_{i,j+1,k}, \mathrm{DZT}_{i+1,j+1,k})\,. \tag{215}$$

Due to these variable cell heights, the spatial discretizations described in Sec. 3 need to be modified as shown in the following.

Tracer Advection:

$$\mathcal{L}_T(\varphi) = \frac{1}{H^T \Delta_y^T} \delta_x \left(\overline{H^U \Delta_y^U\, u_x}^y \overline{\varphi}^x \right) + \frac{1}{H^T \Delta_x^T} \delta_y \left(\overline{H^U \Delta_x^U\, u_y}^x \overline{\varphi}^y \right) + \delta_z \left(w \overline{\varphi}^z \right) \,, \tag{216}$$

where $\Delta_x^{(U,T)} = \mathrm{DX(U,T)}$, $\Delta_y^{(U,T)} = \mathrm{DY(U,T)}$, $H^{(U,T)} = \mathrm{DZ(U,T)}$, and the vertical velocity w at bottom T-cells is found as

$$w_{k+1} - w_k = \frac{1}{\Delta_y^T} \delta_x \left(\overline{\Delta_y^U H^U u_x}^y \right) + \frac{1}{\Delta_x^T} \delta_y \left(\overline{\Delta_x^U H^U u_y}^x \right) \,. \tag{217}$$

Laplacian Horizontal Tracer Diffusion:

$$\mathcal{D}_H(\varphi) = \frac{1}{H^T \Delta_y^T} \delta_x \left(\overline{A_H}^x \Delta_y \Delta_z^i \delta_x \varphi \right) + \frac{1}{H^T \Delta_x^T} \delta_y \left(\overline{A_H}^y \Delta_x \Delta_z^j \delta_y \varphi \right) \,, \tag{218}$$

where Δ_x=HTN, Δ_y=HTE, and

$$\Delta_z^i = \min(\mathrm{DZT}_{i,j,k}, \mathrm{DZT}_{i+1,j,k}), \quad \Delta_z^j = \min(\mathrm{DZT}_{i,j,k}, \mathrm{DZT}_{i,j+1,k}). \tag{219}$$

Vertical Tracer Diffusion:

$$\mathcal{D}_V(\varphi) = \frac{1}{H^T} \left[\frac{\kappa_{k+\frac{1}{2}}(\varphi_{k+1} - \varphi_k)}{({H^T}_{k+1} + {H^T}_k)/2} - \frac{\kappa_{k-\frac{1}{2}}(\varphi_k - \varphi_{k-1})}{(H_k^T + H_{k-1}^T)/2} \right] . \tag{220}$$

Momentum Advection:

$$\mathcal{L}_U(\alpha) = \frac{1}{H^U \Delta_y^U} \delta_x \left(\overline{\overline{H^U \Delta_y^U\, u_x}^y}^{xy} \overline{\alpha}^x \right) + \frac{1}{H^U \Delta_x^U} \delta_y \left(\overline{\overline{H^U \Delta_x^U\, u_y}^x}^{xy} \overline{\alpha}^y \right) + \delta_z \left(w \overline{\alpha}^z \right) . \tag{221}$$

Laplacian Horizontal Friction:

$$\mathcal{D}_H(\mathbf{u}) = \frac{1}{H^U \Delta_y^U} \delta_x \left(\overline{A_M}^x \Delta_y^i \Delta_z^i \delta_x \mathbf{u} \right) + \frac{1}{H^U \Delta_x^U} \delta_y \left(\overline{A_M}^y \Delta_x^j \Delta_z^j \delta_y \mathbf{u} \right)$$

$$- \frac{A_M}{H^U \Delta_x^U \Delta_y^U} \left[\overline{\frac{\Delta_y^i (H^U - \Delta_z^i)}{\Delta_x}}^x + \overline{\frac{\Delta_x^j (H^U - \Delta_z^i)}{\Delta_y}}^y \right] \mathbf{u} \,, \tag{222}$$

where Δ_x=HTN, Δ_y=HTE, Δ_x^j=HUS, Δ_y^i=HUW, and

$$\Delta_z^i = \min(\mathrm{DZU}_{i,j,k}, \mathrm{DZU}_{i+1,j,k}), \quad \Delta_z^j = \min(\mathrm{DZU}_{i,j,k}, \mathrm{DZU}_{i,j+1,k}). \tag{223}$$

The last term of (222) is the drag due to the no-slip boundary condition at lateral boundaries. Metric terms in the momentum advection and horizontal friction remain unaffected by partial bottom cells.

Vertical Friction:

$$\mathcal{D}_V(\mathbf{u}) = \frac{1}{H^U}\left[\frac{\kappa_{k+\frac{1}{2}}(\mathbf{u}_{k+1} - \mathbf{u}_k)}{(H^U_{k+1} + H^U_k)/2} - \frac{\kappa_{k-\frac{1}{2}}(\mathbf{u}_k - \mathbf{u}_{k-1})}{(H^U_k + H^U_{k-1})/2}\right]. \tag{224}$$

Hydrostatic Pressure: Since we leave the pressure points at the middle of full cells, the hydrostatic pressure is computed in the same way as for the full-cell case.

Biharmonic Horizontal Diffusion and Friction: The same operations for the Laplacian diffusion and friction given by (218) and (222) for φ and $\mathbf{u}$, respectively, are applied twice.

Isopycnal Diffusion:

$$\begin{aligned}
\mathcal{D}_H(\varphi) = {} & \frac{1}{H^T\,\Delta_y^T}\,\delta_x\left[\Delta_y\,\left(\overline{H^T\,\overline{\kappa}^z}^x\,\delta_x\varphi - \overline{H^T\,\overline{\kappa\,S_x\,\delta_z\varphi}^z}^x\right)\right] \\
& - \frac{1}{\Delta_x^T\,\Delta_y^T}\,\delta_z\left[\frac{1}{\overline{H^T}^z}\left(\overline{H^T\,\overline{\Delta_x^j\Delta_y\,\kappa\,S_x\,\delta_x\varphi}^x}^z - \overline{H^T\,\overline{\Delta_x^j\Delta_y\,\kappa\,S_x^2}^x}^z\,\delta_z\varphi\right)\right] \\
& + \frac{1}{H^T\,\Delta_x^T}\,\delta_y\left[\Delta_x\,\left(\overline{H^T\,\overline{\kappa}^z}^y\,\delta_y\varphi - \overline{H^T\,\overline{\kappa\,S_y\,\delta_z\varphi}^z}^y\right)\right] \\
& - \frac{1}{\Delta_x^T\,\Delta_y^T}\,\delta_z\left[\frac{1}{\overline{H^T}^z}\left(\overline{H^T\,\overline{\Delta_x\Delta_y^i\,\kappa\,S_y\,\delta_y\varphi}^y}^z - \overline{H^T\,\overline{\Delta_x\Delta_y^i\,\kappa\,S_y^2}^y}^z\,\delta_z\varphi\right)\right],
\end{aligned} \tag{225}$$

where Δ_x=HTN, Δ_y=HTE, Δ_x^j=HUS, Δ_y^i=HUW, $S_x = \delta_x\rho/\delta_z\rho$, and $S_y = \delta_y\rho/\delta_z\rho$. In (225), for example, in the (x,z)-plane, κ in the two quarter cells in either the east or west direction of T-cells right next to any partial cell is reduced to

$$\kappa^e_{i,j,k} = \frac{H^T_{i+1,j,k}}{dz_k}\,\kappa_{i,j,k}, \quad \kappa^w_{i,j,k} = \frac{H^T_{i-1,j,k}}{dz_k}\,\kappa_{i,j,k}\,, \tag{226}$$

where dz_k is the full-cell height. Similar reductions are made for κ in the (y,z)-plane, too. Also κ in the lower half of T-cells right above any partial cells and κ in the upper half of any partial cells are modified to

$$\kappa^{bottom}_{i,j,k-1} = \frac{H^T_{i,j,k}}{dz_k}\kappa_{i,j,k-1}, \quad \kappa^{top}_{i,j,k} = \frac{H^T{}_{i,j,k}}{dz_k}\kappa_{i,j,k}\,. \tag{227}$$

Remember that κ at the lower half of bottom T-cells is set to zero.

Anisotropic Viscosity:

$$\mathcal{D}^x_H(\mathbf{u}) = \frac{1}{\Delta^U_y}\delta_x(\overline{\Delta_y\gamma\sigma_{11}}^x) + \frac{1}{\Delta^U_x}\delta_y(\overline{\Delta_x\gamma\sigma_{12}}^y) - \overline{\Delta_x k_1\overline{\Delta_y\gamma\sigma_{22}}^x}^x + \overline{\Delta_y k_2\overline{\Delta_x\gamma\sigma_{12}}^y}^y\,, \tag{228}$$

$$\mathcal{D}^y_H(\mathbf{u}) = \frac{1}{\Delta^U_x}\delta_y(\overline{\Delta_x\gamma\sigma_{22}}^y) + \frac{1}{\Delta^U_y}\delta_x(\overline{\Delta_y\gamma\sigma_{12}}^x) - \overline{\Delta_y k_2\overline{\Delta_x\gamma\sigma_{11}}^y}^y + \overline{\Delta_x k_1\overline{\Delta_y\gamma\sigma_{12}}^x}^x\,, \tag{229}$$

where Δ_x=HTN, Δ_y=HTE, $\gamma = 1$ at four quarter cells at the centered U-cell and γ's are defined in two quarter cells adjacent to the U-cell in the east, west, north and south directions as

$$\gamma^e = \frac{\min(H^U_{i,j,k}, H^U_{i+1,j,k})}{H^U_{i,j,k}}\,, \quad \gamma^w = \frac{\min(H^U_{i,j,k}, H^U_{i-1,j,k})}{H^U_{i,j,k}}\,, \tag{230}$$

$$\gamma^n = \frac{\min(H^U_{i,j,k}, H^U_{i,j+1,k})}{H^U_{i,j,k}}\,, \quad \gamma^s = \frac{\min(H^U_{i,j,k}, H^U_{i,j-1,k})}{H^U_{i,j,k}}\,. \tag{231}$$

Due to no-slip boundary conditions imposed at lateral boundaries, the velocity gradients to compute the stress tensor σ are re-defined as

$$(\delta_x u)^e = \frac{1}{\Delta_x}\left(\gamma^e u_{i+1,j,k} - u_{i,j,k}\right)\,, \quad (\delta_x u)^w = \frac{1}{\Delta_x}\left(u_{i,j,k} - \gamma^w u_{i-1,j,k}\right)\,, \tag{232}$$

$$(\delta_x u)^n = \frac{1}{\Delta_y}\left(\gamma^n u_{i,j+1,k} - u_{i,j,k}\right)\,, \quad (\delta_x u)^s = \frac{1}{\Delta_y}\left(u_{i,j,k} - \gamma^s u_{i,j-1,k}\right)\,, \tag{233}$$

$$(k_1\mathbf{u})^e = k^e_1\,\frac{\mathbf{u}_{i,j,k} + \gamma^e\mathbf{u}_{i+1,j,k}}{2}\,, \quad (k_1\mathbf{u})^w = k^w_1\,\frac{\mathbf{u}_{i,j,k} + \gamma^w\mathbf{u}_{i-1,j,k}}{2}\,, \tag{234}$$

$$(k_2\mathbf{u})^n = k^n_2\,\frac{\mathbf{u}_{i,j,k} + \gamma^n\mathbf{u}_{i,j+1,k}}{2}\,, \quad (k_2\mathbf{u})^s = k^s_2\,\frac{\mathbf{u}_{i,j,k} + \gamma^s\mathbf{u}_{i,j-1,k}}{2}\,, \tag{235}$$

where k^e_1, k^w_1, k^n_2 and k^s_2 are the metric terms defined on the east, west, north and south faces.

7. Global Orthogonal Grids

As discussed in Sec. 3, the POP model is designed to handle any orthogonal curvilinear coordinate system for the horizontal grid. The only information the code needs locally at each grid point is the longitude and latitude of that point and the distances to neighboring points along the two coordinate directions. This information is computed offline and read in from a file. However, there are restrictions on the global connectivity of the grid. In the current (2002) model release, it is assumed that the computational grid can be mapped onto a cylinder, i.e., a 2-D array which is periodic in the x-direction. When such grids are mapped onto the sphere they necessarily involve two singularities, analogous to the north and south poles on a standard polar grid. In these "dipole" grids, originally developed independently by two groups (Madec and Imbard, 1996, and Smith et al., 1995), the two singularities can be placed inside land masses (e.g., North America, Greenland or Asia) to construct global grids. The next release of the code will also have the ability to use what we call "tripole" grids, which were orginally developed by R. Murray (1996, see his Figure 11). In these grids, there are two "poles" in the northern hemisphere (usually placed in North America and Asia). These grids can also be mapped onto a cylinder but special boundary communications are required in order to "sew up" the line between the two northern poles. In both dipole and tripole grids, the meshes are constructed so that there is a smooth transition moving north from the Equator, from the southern hemisphere polar grid (usually Mercator grid) to the northern "distorted" grid. Various factors enter into the design of a given grid, and different choices (pole locations, dipole vs. tripole) may be preferred for different problems. In global grids the Equator is typically chosen as a grid line so that the numerical dispersion relation for the B-grid linear Kelvin waves is accurate. At resolutions coarser than about $1/2°$, the meridional spacing near the Equator is enhanced (to about $1/2°$) in order to more accurately resolve equatorial waves and currents. Note: Software for generation of global dipole and tripole grids is not released with the code, but will eventually be made public.

7.1. Dipole Grids

The simplest orthogonal grid with two arbitrarily located poles can be constructed analytically in one of three ways: 1) Electrostatic dipole charge distribution on a sphere: solve Poisson's Equation on the sphere with two electric charges of opposite sign at the chosen pole points; the electric field lines and the lines of constant electrostatic potential form an orthogonal coordinate system; 2) Conformal map method (Murray, 1996): project a spherical polar grid onto a tangent plane at its equator from the point on the sphere opposite the tangent point, then reproject back onto a larger sphere tangent to the plane at the same point; the new pole points on the larger sphere will be shifted toward its equator and the tangent point; 3) Geometric method (Smith et al., 1995): choose two axes, one which intersects the surface of the sphere at the two pole points and one located at the intersection of the two planes tangent to the pole points, then the two families of planes that contain these axes intersect the surface of the sphere in two sets of circles which form the coordinates of an orthogonal coordinate system. All these methods produce equivalent orthogonal grids. The only problem with these simple analytic grids is that if one pole is located in Antarctica and the other is displaced away from the true North Pole, then the Equator will not be a grid line. To get around this problem, grids can be constructed (Madec and Imbard, 1996, Smith et al., 1995) by an iterative procedure, starting at the equator of a southern hemisphere polar grid, where the northern pole point is gradually shifted away from the true North Pole as successive latitude-like circles are constructed. In this manner it is possible to construct a semi-analytic dipole grid which smoothly matches onto the southern polar grid, and in which local changes in grid spacing are everywhere small and smooth. Composite grids that have sudden changes in grid spacing normal to matching boundaries suffer a loss of formal second-order accuracy of the spatial discretization scheme (Smith et al., 1995).

An example of a semi-analytic dipole grid is shown in Figure 8. In this grid the northern pole is in North America and the southern hemisphere is a Mercator grid with a pole at the true South Pole. Note that the meridional grid spacing near the Equator has been enhanced as discussed above. The grid spacing in the northern hemisphere is chosen so as to minimize the global mean aspect ratio, attempting to make each cell as nearly "square" as possible (as in a true Mercator grid). In both hemispheres the grid terminates in a circle enclosing the pole, and anything within this circle is excluded (e.g., Hudson Bay in Fig. 8).

7.2. Tripole Grids

Analytic grids can also be constructed which involve more than two singularities using the different methods described above for dipole grids. Various types of multipole grids are discussed by Murray (1996). One variation that is particularly useful for ocean models is shown in Figure 11 of Murray (1996). It has two singularities in the northern hemisphere and smoothly matches onto a southern hemisphere Mercator grid. A "tripole" grid of this type is shown in Figure 9. Note that the grid spacing in the Arctic is much more uniform and the cell aspect ratios are much closer to one than in the dipole grid in Fig. 8. This particular grid was constructed for the French OPA model (Madec et al., 1998), and was made available, courtesy of G. Madec, for testing in POP. Like the dipole grid, the tripole grid is mapped onto a cylinder (a logical 2-D array that is periodic in the x-direction). The difference is that in the dipole grid the northernmost ("upper") boundary of the grid corresponds to the circle surrounding the pole (e.g. the open circle containing Hudson bay in Fig. 8), whereas in the tripole grid the two northern poles both lie on the upper boundary, one is located at the two upper corners of the logical grid (which correspond to the same point since it is periodic) and the other at the midpoint of the upper boundary. When the 2-D cylindrical array is mapped onto the sphere (as in Fig. 9) there is a chopped pole in the southern hemisphere, but in the northern hemisphere there is a "cut" in the grid along the line between the poles, and flow across this line from one side of the Arctic to the other requires non-local communication along the upper boundary of the logical grid: it must be folded back along itself at its mid point in order to make neighboring cells in the physical grid be adjacent in the logical grid. The indexing for this nonlocal communication is different for quantities located at different places (i.e. U-points or T-points) on the grid. POP does assume that in the logical grid the upper boundary lies on the northern edge of the last row of T-cells, so that U-points lie exactly on the upper grid boundary, along the line between the two poles. Furthermore, it assumes that the middle pole point lies at a U-point exactly halfway across the top of the grid.

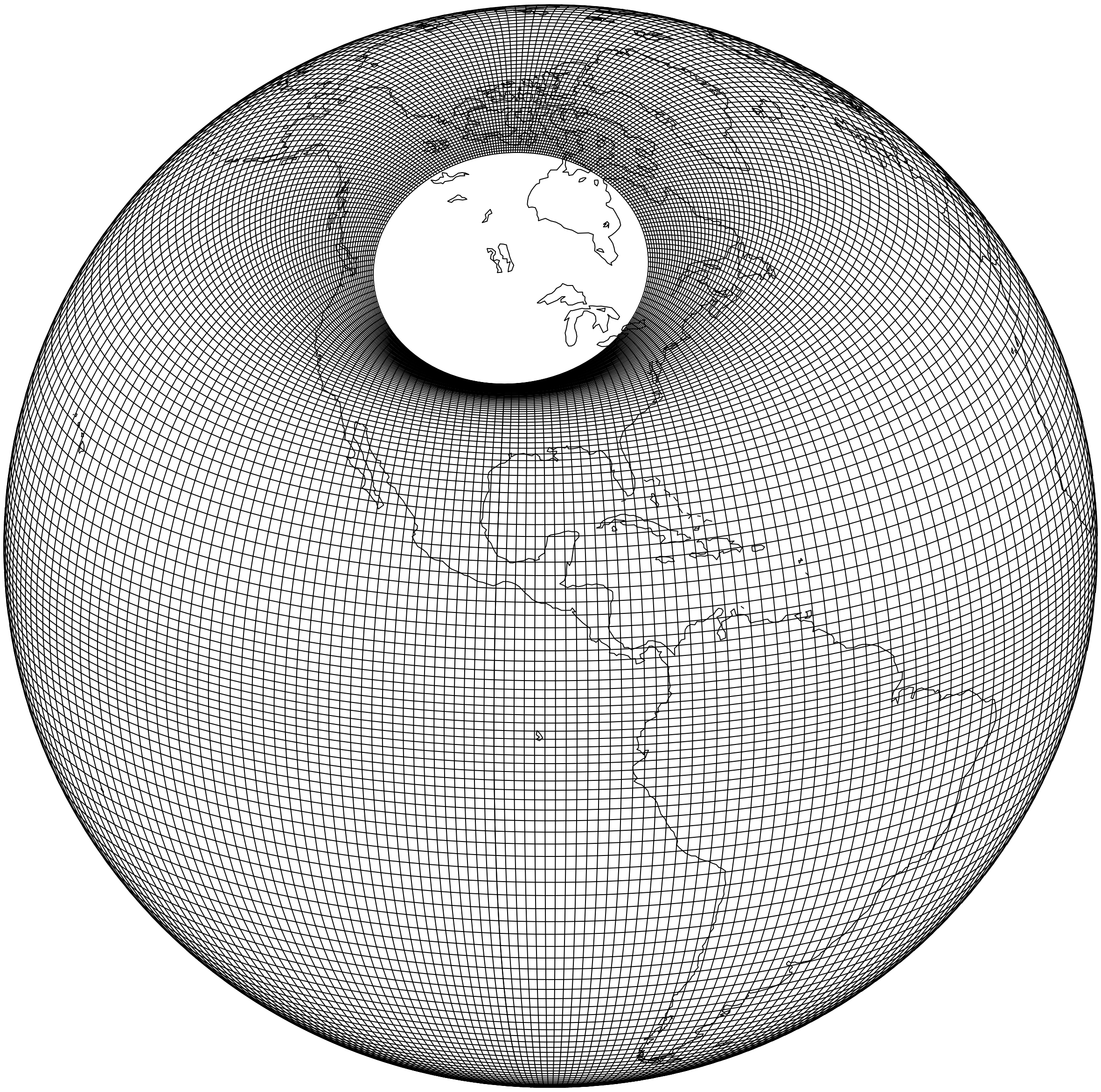

Figure 8: A dipole grid.

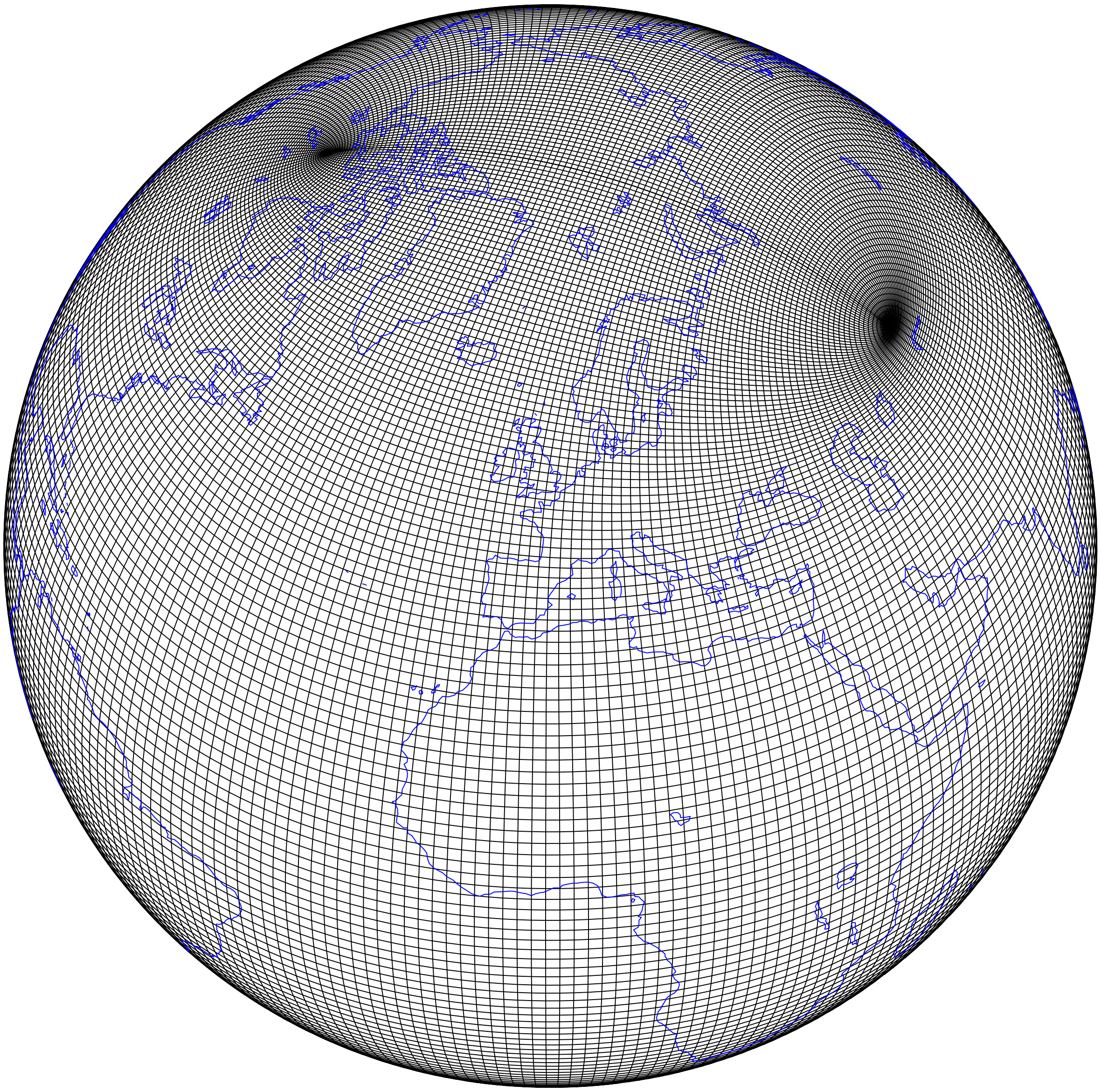

Figure 9: A tripole grid.

8. Forcing and Coupling Issues

8.1. Sea-Ice Formation and Melting

Air-sea heat fluxes, when applied over a time interval, can produce regions of subfreezing model upper-layer temperatures. This situation can be alleviated by adjusting the model Θ and S (and the first layer thickness, if the variable surface thickness formulation is used with freshwater fluxes) that is associated with frazil sea-ice formation. The choice of when to perform such adjustments is judiciously made to ensure tracer conservation with minimum adjustments. So, in coupled simulations with *averaging* as the time mixing option, such *ice formation time steps* are performed at both the coupling time step and the time step just prior to it so that both branches of the leap-frog time steps are brought to the freezing temperature which will then be used over the next coupling interval. If *Matsuno* time mixing option is selected, checking for ice formation only at the coupling time steps is adequate because the first time step of the next coupling interval is *Matsuno* and the unadjusted branch of the leap-frog time steps will be ignored. The following algorithm describes the procedure for the virtual salt flux formulation, default for the present integrations.

After Θ and S are updated, the upper `kmxice` model layers (denoted by index k) are scanned for Θ below the freezing point, Θ_f. Here, `kmxice` is a model namelist input in `ice_nml` and, at present, Θ_f is not a function of local salinity S_k, but $\Theta_f = -1.8°$ C. Although we describe the most general case here, in practice we do recommend setting `kmxice=1`, because unless a monotone advection scheme is in place, model advective errors could lead to Θ far below freezing in some deep locations and the scheme will form too much ice.

At each layer, the potential mass per unit area of ice formation ($\mathtt{POTICE}_k > 0$) or ice melt ($\mathtt{POTICE}_k < 0$) is computed as

$$\mathtt{POTICE}_k = \mathrm{MAX}\left[\frac{\rho_{sw}\ c_p}{L_f}\ \Delta z_k\ (\Theta_f - \Theta_k)\ \ ,\ \ \mathtt{QICE}_{k+1}\right]\ , \tag{236}$$

where ρ_{sw} and c_p are the density and heat capacity of sea water, respectively, L_f is the latent heat of fusion, Δz_k is the layer thickness, and Θ_k is the local potential temperature. Any ice that forms at depth is assumed to float towards the surface and this ice flux, $\mathtt{QICE}_k$ (defined positive downwards so $\mathtt{QICE}_k \leq 0$), is accumulated bottom to top as

$$\mathtt{QICE}_k = \sum_{kmxice}^{k} -\mathtt{POTICE}_k, \tag{237}$$

assuming no ice formation below the `kmxice` layer, i.e. $\mathtt{QICE}_{k=kmxice+1} = 0$. As ice floats to the surface, it can either partially or completely melt in upper layers whose temperatures are above freezing.

At each layer, Θ and S are both adjusted in accord with the ice formed or melted in the layer:

$$\Theta_k = \Theta_k + \frac{L_f}{\Delta z_k\ \rho_{sw}\ c_p}\quad \mathtt{POTICE}_k, \tag{238}$$

$$S_k = S_k + \frac{(S_o - S_i)}{\rho_{sw}\ \Delta z_k}\quad \mathtt{POTICE}_k\ . \tag{239}$$

For S, this is equivalent to replacing a volume of formed ice at salinity S_i with an identical volume of water at salinity S_o. Here S_i and S_o represent sea ice and ocean reference salinities, respectively, and they are set to constant values in order to ensure salt conservation. In addition, if POP is coupled to a sea-ice model, both must use the same value for S_i (presently in CCSM2.0, $S_i = 0$).

The ice flux is accumulated at each location over the number of ice formation time steps, N, during a coupling interval as

$$\texttt{AQICE} = \sum_{1}^{N} weight_1 \; \texttt{QICE}_{k=1}, \tag{240}$$

where $weight_1$ is either 1/2 or 1 depending on whether it is an *averaging* time step or not.

If, at the end of a vertical scan, the surface temperature Θ_1 remains greater than freezing, ($\texttt{QICE}_{k=1} = 0$), then the excess heat melts previously formed ice, and AQICE, Θ_1, and S_1 are adjusted accordingly. Thus, over a coupling interval, ice is assumed to remain where it was formed. In coupled simulations, the accumulated ice during a coupling interval is passed to an ice model via the flux coupler as an equivalent downward heat flux, QFLUX in W m^{-2}:

$$\texttt{QFLUX} = \frac{-L_f}{\Delta t^*} \; weight_2 \; \texttt{AQICE} \tag{241}$$

where AQICE includes any adjustment due to melting previously formed ice over the last timestep, $\Delta t^* = 1$ day is the coupling interval, and $weight_2$ is either 1/2 or 1 depending on *averaging* or *Matsuno* time steps, respectively. In the last two equations, the presence of $weight$s assures that Θ and S budgets will be conserved when the *averaging* option is selected where, after averaging, actual Θ and S changes become 1/2 of what is implied by the ice formation fluxes.

Thus, Θ and S adjustments corresponding to the total ice formed ($\texttt{QFLUX} > 0$) are made prior to the ice being passed to the sea-ice model. However, warm surface model temperatures result in $\texttt{QFLUX} < 0$, which is a potential to melt ice in the sea-ice model. Since the ocean does not know if sufficient ice is present at a given location, Θ and S adjustments are delayed until the appropriate heat and freshwater fluxes are received back from the sea-ice model through the coupler. If sufficient ice is present in the absence of all other fluxes, the heat flux will be equal to the cooling needed to make $\Theta_1 = \Theta_f$, when applied over the next coupling interval.

The ice formation option can also be activated in ocean-only (uncoupled) simulations subject to observational or reanalysis based air-sea heat fluxes. In this case, the accumulated ice is used internally for local ice formation and melt, and AQICE is saved in restart files for exact continuation. If the *Matsuno* time step mixing option is chosen, the ice time steps are set to be the time steps just before a mixing step by default. For the *averaging* time step option, the default is to form ice every time step.

The POP ice formation subroutine contains a modified algorithm when the freshwater flux formulation is chosen. Here, the volume of ice formed at layer k is replaced with an equal volume of water (because the thickness of the below surface layers cannot change) from the layer above with salinity S_{k-1}. The surface layer S can change both through exchanges of S with lower layers and change of thickness due to formed ice sent to the ice model. Because there still remain some consistencies and scientific issues with ice formation and melt when freshwater fluxes are used, this algorithm has not yet been tested, and is not recommended to be used at the present time.

8.2. Surface Freshwater Flux Balancing over Marginal Seas

Unlike the surface heat flux and sea surface temperature, there are no appreciable feed-backs between the surface freshwater flux and S. This is particularly so in isolated marginal sea regions where the freshwater fluxes can produce unphysical S values throughout the water column. The situation can be especially severe in coupled integrations when the marginal sea regions receive river run-off fluxes. In uncoupled integrations, one obvious remedy is to use restoring freshwater or salt fluxes instead in these regions where the surface S is restored to some climatological distribution with a relatively small restoring time scale. In this section, we describe a more novel approach, particularly designed for coupled integrations. Here, the amount of excess or deficit of freshwater flux over a marginal sea is transported to or from its associated active-ocean region, thus implicitly connecting marginal seas with active ocean and providing a means for marginal sea run-off to eventually discharge into the open ocean. This process assures that the volume-mean S stays constant throughout the integration within each marginal sea, eliminating any unphysical values in S.

When balancing is requested, the active-ocean regions corresponding to each marginal sea, ms, are determined at the beginning of each integration. For this computation, a longitude, a latitude (both in degrees), and a distribution active-ocean area size (in cm^2) are provided in the `region_info_filename` for each such region. Using the longitude and latitude values as starting points, a search window is created to find the active-ocean points associated with each ms. This iterative process continues until the cumulative active-ocean grid areas match the input distribution size. At each iteration, the search window is increased by 1° on all four sides. If the hard-coded values for either the maximum number of iterations or the maximum number of active-ocean grid points per ms are exceeded, POP will stop with an error message.

The final product of this initialization procedure is a global array of area fractions for each ms, $F_{i,j}^{ms}$. In F, the fractions are computed simply as the ratios of grid areas to the total computed distribution area. It is ensured that the sum of these area fractions for each ms is unity. The distribution regions for different ms can coincide. Note that, the zero elements of F represent the ocean points outside of the designated regions.

In fully coupled integrations in which the surface fluxes stay constant till the next coupling, balancing is performed once per coupling interval, immediately following the arrival of the new fluxes. If the fluxes do change during a coupling interval (e.g. `partially-coupled` option), then balancing is done every time step. For each ms, a transport term, T, is evaluated in Kg s^{-1} of freshwater:

$$T^{ms} = \sum_{ms} \left(\left[\mathtt{MAX}(0, \mathtt{QFLUX}_{i,j})\ c_q \ + \ (E_{i,j} \ + \ P_{i,j} \ + \ R_{i,j} \ + M_{i,j}) + F_{i,j}^{S}\ c_s \right] \Delta x_{i,j}\ \Delta y_{i,j} \right), \quad (242)$$

where only the fluxes over the marginal seas contribute to the sum which is performed over each ms individually. In the above equation, `QFLUX` > 0 is the frazil ice formation in W m^{-2} and E, P, R, and M represent the evaporation, precipitation, river run-off, and ice-melt freshwater fluxes from the coupler in Kg m^{-2} s^{-1}. F^S is the salt flux due to ice melt in Kg of salt m^{-2} s^{-1}; because $S_i = 0$ in the current CCSM2.0 simulations, F^S is also zero. Finally, Δx and Δy are the zonal and meridional grid spacings (in m) centered at Θ points, respectively, and c_q and c_s represent unit conversion factors given by

$$c_q = -\frac{1}{L_f} \; \frac{S_o \; - \; S_i}{S_o}, \qquad c_s = -\frac{1}{S_o} \; \frac{\rho_{fw}}{\rho_{sw}}. \tag{243}$$

T^{ms} represents the amount of excess or deficit freshwater flux for marginal sea ms that needs to be transported to or from its associated active-ocean region. How much of T^{ms} is transported to or from an active-ocean grid point is determined by

$$MSTF_{i,j} \;\; = \;\; \frac{F^{ms}_{i,j} \;\; T^{ms}}{\Delta x_{i,j} \;\; \Delta y_{i,j}}. \tag{244}$$

At these points, the surface freshwater or salt flux is modified as

$$SF^{new}_{i,j} \;\; = \;\; SF^{old}_{i,j} \;\; + \;\; c_t \; MSTF_{i,j}, \tag{245}$$

where $c_t = -S_o/\rho_{fw}$ or $c_t = 1$ depending on if POP is forced with virtual salt or freshwater flux boundary conditions, respectively.

In marginal seas where there is no frazil ice formation, the surface freshwater flux is set to zero. Otherwise, the surface flux is simply $-\mathtt{MAX}(0, \mathtt{QFLUX}_{i,j}) c_q c_t$ to undo the S adjustment that has been done in the previous time steps.

Because this algorithm has not yet been used with the freshwater flux formulation, its use is not recommended with that formulation.

References

Adcroft, A., C. Hill, and J. Marshall, 1997: Representation of topography by shaved cells in a height coordinate ocean model. *Mon. Wea. Rev.*, **125**, 2293–2315.

Arakawa, A., and V. R. Lamb, 1977: Computational design of the basic dynamical processes of the UCLA general circulation model, *Methods of Computational Physics* **17**, Academic, New York, p. 173.

Arfken, G., 1970: *Mathematical Methods for Physicists*, Academic, New York.

Brown, J. A., and K. A. Campana, 1978: An economical time-differencing scheme for numerical weather prediction. *Mon. Weather Rev.*, **106**, 1125–1136.

Bryan, K., 1969: A numerical method for the study of the world ocean *J. Comp. Phys.*, **4**, 347–376.

Bryan, K., 1984: Accelerating the convergence to equilibrium of ocean-climate models. *J. Phys. Oceanogr.* **14**, 666–673.

Bryan, K., and M. D. Cox, 1972: An approximate equation of state for numerical models of 0cean circulation. *J. Phys. Oceanogr.*, **2**, 510–514.

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

Danabasoglu, G., and J. C. McWilliams, 1995: Sensitivity of the global ocean circulation to parameterizations of mesoscale tracer transports. *J. Climate*, **8**, 2967–2987.

Dewar, W. K., Y. Hsueh, T. J. McDougall, and D. Yuan, 1998: Calculation of pressure in ocean simulations. *J. Phys.. Oceanogr.*, **28**, 577–588.

Dukowicz, J. K., 2001: Reduction of density and pressure gradient errors in ocean simulations. *J. Phys. Oceanogr.* **31**, 1915–1921.

Dukowicz, J. K., and R. D. Smith, 1994: Implicit free-surface formulation of the Bryan-Cox-Semtner ocean model. *J. Geophys. Res.* **99**, 7991–8014.

Dukowicz, J. K., R. D. Smith and R. C. Malone, 1993: A reformulation and implementation of the Bryan-Cox-Semtner ocean model on the Connection Machine. *J. Atmos. Ocean Tech.*, **14**, 294-317.

Farrow, D.E., and D.P. Stevens, 1995: A new tracer advection scheme for Bryan and Cox type ocean general circulation models. *J. Phys. Oceanogr.* **25**, 1731–1741.

Feistel, R. and E. Hagen, 1995: On the GIBBS thermodynamic potential of seawater. *Prog. in Oceanogr.*, **36**, 249–327.

Fofonoff, N. P. and R. C. Millard, 1983: Algorithms for computation of fundamental properties of seawater. UNESCO Technical Papers in Marine Science, **44**, UNESCO, 53pp.

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

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

Griffies, S. M., 1998: The Gent-McWilliams skew flux. *J. Phys. Oceanogr.* **28**, 831–841.

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

Griffies, S. M, and R. W. Hallberg, 2000: Biharmonic friction with a Smagorinsky-like viscosity for use in large-scale eddy-permitting ocean models, *Monthly Weather Review*, **128**, 2935–2946.

Griffies, S. M., R. C. Pacanowski, M. Schmidt, and V. Balaji, 2001: Tracer conservation with an explicit free surface method for z-coordinate ocean models. *Monthly Weather Review*, **129**, 1081–1098.

Haltiner, G. J., and R. T. Williams, 1980: *Numerical Prediction and Dynamic Meteorology*, 2nd Ed., John Wiley and Sons, New York.

Holland, W.R. and F.O. Bryan and J.C. Chow, 1998: Application of a third-order upwind scheme in the NCAR Ocean Model. *J. Climate* , **11**, 1487–1493.

Jackett, D. R. and T. J. McDougall, 1995: Minimal adjustment of hydrographic profiles to achieve static stability. *J. of Atmos. and Oceanic Tech.* **12**, 381–389.

Killworth, P. D., D. Stainforth, D. J. Webb, and S. M. Paterson, 1991: The development of a free-surface Bryan-Cox-Semtner ocean model, *J. Phys. Oceanogr.* **21**, 1333–1348.

Large, W.G., J.C. McWilliams and S.C. Doney, 1994: Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization, *Reviews of Geophysics*, **32**, 363–403.

Large, W. G., G. Danabasoglu, S. C. Doney, and J. C. McWilliams, 1997: Sensitivity to surface forcing and boundary layer mixing in the NCAR CSM ocean model: Annual-mean climatology. *J. Phys. Oceanogr.* **27**, 2418–2447.

Large, W. G., G. Danabasoglu, J. C. McWilliams, P. R. Gent and F. O. Bryan, 2001: Equatorial circulation of a global ocean climate model with anisotropic horizontal viscosity. *J. Phys. Oceanogr.*, **31**, 518-536.

Leonard, B.P., 1979: A stable and accurate convective modeling procedure based on quadratic upstream interpolation. *Comp. Meth. in Appl. Eng.*, **19**, 59–98.

Madec, G., and M. Imbard, 1996: A global ocean mesh to overcome the North Pole singularity. *Climate Dyn.*, **12**, 381–388.

Madec, G., M. Imbard, and C. Levy, 1998: OPA 8.1 Ocean General Circulation Model Reference Manual. *Note du Pole der modelisation*, Institute Pierre-Simon Laplace (IPSL), France, n°11, 91pp.

Maltrud, M., R. D. Smith, A. J. Semtner, and R. C. Malone, 1998: Global eddy-resolving ocean simulations driven by 1985-1995 atmospheric winds. *J. Geophys. Res.*, **103**, 30,825–30,853.

McDougall, T. J., 1987: Neutral Surfaces. *J. Phys. Oceanogr.* **17**, 1950–1964.

McDougall, T. J., D. R. Jackett, D. G. Wright, and R. Feistel, 2002: Accurate and computationally efficient algorithms for potential temperature and density of seawater. *J. of Atmos. and Oceanic Tech.* Submitted.

Murray, R. J., 1996: Explicit generation of orthogonal grids for ocean models. *J. Comp. Phys.*, **126**, 251.

Pacanowski, R. C. and S. M. Griffies, 2000: The MOM3 Manual. Geophysical Fluid Dynamics Laboratory/NOAA, Princeton, USA.

Pacanowski, R. C. and Gnanadesikan A. 1998: Transient response in a z-level ocean model that resolves topography with partial cells. *J. Phys. Oceanogr.* **126**, 3248–3270.

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

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

Schopf, P. S. and A. Loughe, 1995: A reduced-gravity isopycnal ocean model: hindcasts of El Nino. *Monthly Weather Review*, **123**, 2839-2863.

Semtner, Jr, A. J.,1986: Finite-difference formulation of a world ocean model, in *Advanced Physical Oceanographic Numerical Modelling*, J. J. O'Brien, Ed., D. Reidel Publishing Company, Norwell, Mass., p. 187.

Smagorinsky, J., 1993: Some hisorical remarks on the use of nonlinear viscosities, in *Large Eddy Simulation of Complex Engineering and Geophysical Flows*, B. Galperin and S. A. Orszag, Eds. Cambridge University Press.

Smith, R. D., J. K. Dukowicz, and R. C. Malone, 1992: Parallel ocean general circulation modeling. *Physica D.*, **60**, 38–61.

Smith, R. D., S. Kortas, and B. Meltz, 1995: Curvilinear coordinates for global ocean models, Los Alamos Technical Report LA-UR-95-1146.

Smith, R. D., 1999: The primitive equations in the stochastic theory of adiabatic stratified turbulence. *J. Phys. Oceanogr.* **29**, 1865–1880.

Smith, R. D. and J. C. McWilliams, 2002: Anisotropic horizontal viscosity for ocean models, Submitted to *Ocean Modelling.*

Visbeck M., Marshall, J. , Haine, T. and M. Spall, 1997: Specification of eddy transfer coefficients in coarse- resolution ocean circulation models. *J. Phys. Oceanogr.* **27**, 381–402.

Wajsowicz, R. C., 1993: A consistent formulation of the anisotropic stress tensor for use in models of the large-scale ocean circulation, *J. Comp. Phys.* **10**, 333-338.

Webb, D. J., 1995: The vertical advection of momentum in Bryan-Cox-Semtner ocean general circulation models. *J. Phys. Oceanogr.* **25**, 3186–3195.

Williams, G. P., 1972: Friction term formulation and convective instability in a shallow atmosphere, *J. Atmos. Sci.* **29**, 870-876.

This preprint was prepared with AGU's LaTeX macros v4. File manual.final formatted May 13, 2002.